Chapter II
A Theory for Iterative Source Waveform Deconvolution

A type of iterative deconvolution that extracts the source waveform and reflectivity from a seismogram through the use of zero memory, non-linear estimators of reflection coefficient amplitude is developed in this chapter. Estimators of this type are implicit in iterative deconvolution techniques that optimize norm ratios. Wiggins (1977) pioneered the development of these algorithms, and Gray (1979) summarized current alternatives.

Here, we present a theory for iterative deconvolution that is based upon the specification of a stochastic model describing reflectivity. Such a model can be developed using the stochastic model of impedance discussed in the last chapter. The resulting parametric algorithm deconvolves the seismogram by forcing a filtered version of the seismogram to resemble an estimated reflection coefficient sequence. This latter time series is itself obtained from the filtered seismogram, and so a degree of iteration is required. One advantage of using a parametric algorithm is that, besides the deconvolution, optimal estimates of reflection coefficient amplitude, free of convolutional noise, are obtained.

We ignore the dependence between reflection coefficients, inherent in the Markov model, in this particular application. The next chapter includes such dependence when formulating a model for seismogram inversion. The intent of the present chapter is to provide an intuitive, parametric theory for iterative deconvolution. We start by reviewing the objective of iterative deconvolution.

2.1. Iterative Deconvolution - The Objective

The object of iterative deconvolution algorithms is to compute an inverse filter, denoted by f_t^{k+1}, such that the convolution of f_t^{k+1} with the data (or seismogram y_t) results in the complete or partial removal of the forward wavelet b_t, where superscripts refer to iteration.

Assuming that an estimate of the reflection coefficient sequence exists, denoted by $\tilde{r}_t^k$, the following set of overdetermined equations must be solved for f_t^{k+1}:

$$(y * f^{k+1})_t \simeq \tilde{r}_t^k \tag{2.1}$$

The least squares solution to equation (2.1) gives the familiar Toeplitz system of equations:

$$R_{yy} f^{k+1} = \tilde{r}^k \infty y \tag{2.2}$$

R_{yy} = co-variance matrix of y

∞ = crosscorrelation symbol

Equation (2.2) is valid for every channel of data, and if f_t^{k+1} is assumed channel-independent, we merely add together the co-variance matrices and crosscorrelation vectors and solve for the filter. All the current techniques of iterative deconvolution differ in their approach to obtaining the estimated reflection sequence $\tilde{r}_t^k$. Our approach is to compute $\tilde{r}_t^k$ so as to minimize a function of the error $r_t - \tilde{r}_t^k$, where r_t is the "true" reflection sequence. Both square and absolute value functions are considered.

2.2. *The Deconvolution Sequence*

In order to get $\tilde{r}_t^k$, a time series which contain r_t must be available. Of course, this time series contains noise and hence only an estimate $\tilde{r}_t$ can be obtained. The deconvolution sequence, denoted by x_t^k, and obtained from y_t via

$$x_t^k \triangleq (y * f^k)_t \tag{2.3}$$

is such a time series. Equation (2.3) can be rewritten in a way that makes apparent the idea that x_t^k contains r_t:

$$x_t^k = (y * f)_t + \Big[y * (f^k - f)\Big]_t$$

$$x_t^k = r_t + n_t^k \tag{2.4}$$

Equation (2.4) sets the stage for discussing optimal estimates of reflectivity.

2.3. *Optimal Reflection Coefficient Estimators*

Given a deconvolution, $\underline{x} = (x_1,x_2,\ldots,x_N)$, the optimal least squares estimate of the reflection coefficient r_i is the expected value of r_i given $\underline{x}$. Kailath (1976) gives a nice derivation of the result, and it is reproduced in Appendix A. [Throughout this section, we often refer to $\underline{x}$ as the vector of observations. In point of fact they are filtered observations, but from an estimation viewpoint, such a qualification is unnecessary.] For instance, if two Random Variables (RVs) R_i and X (recall that upper case letters refer to RVs and lower case to their values) are known to be related and only one is observed, e.g. $X = 5$, then an intuitive estimate of R_i , call it $\tilde{r}_i$, is just the expected value of R_i given the observation $E(R_i|X=5)$. In our particular case, the situation is slightly more general in that a vector of observations $\underline{x}$ are available to estimate r_i:

$$
\begin{aligned}
{}_2\tilde{r}_i(\underline{x}) &= E\left[R_i|\underline{X} = \underline{x}\right] &(2.5)\\
&= \int r_i f_{R_i|\underline{X}}(r_i|\underline{x})\, dr_i
\end{aligned}
$$

In equation (2.5), the subscript "2" distinguishes the least squares (or L-2) estimator from subsequent estimators. Also, the statement $\underline{X} = \underline{x}$ is equivalent to $(X_1,X_2,\ldots,X_N) = (x_1,x_2,\ldots,x_N)$, i.e. $\underline{X}$ is a random vector which itself is a member of the process $\{X_t\}$.

An alternative estimator to the least squares one is that obtained using a least absolute value error function. The optimal least absolute value estimate of r_i is the conditional median of r_i given $\underline{x}$. Mathematically, this means solving the following equation for ${}_1\tilde{r}_i$:

$$\int_{-\infty}^{{}_1\tilde{r}_i(x)} f_{R_i|\underline{X}}(r_i|\underline{x})dr_i = 0.5 \tag{2.6}$$

The function ${}_1\tilde{r}_i(\underline{x})$ refers to the least absolute or L-1 estimator. The derivation of equation (2.6) is found in Van Trees (1968), p. 56.

From equations (2.5) and (2.6), it is evident that to determine ${}_2\tilde{r}_1(\underline{x})$ and ${}_1\tilde{r}_1(\underline{x})$, we need to solve these formidable expressions, dealing simultaneously with all the x_i's (i.e. $\underline{x}$). This is difficult in the general case in which noise and signal are neither Gaussian nor white nor even independent. The goal of the next section will be to introduce reasonable approximations that will allow the determination of the estimators.

2.4. A Probabilistic Model for Deconvolutions

If the joint pdf for $\underline{R}$ and $\underline{N}$, and therefore their correlation (or lack of), is specified, then the pdf for $\underline{X}$ is completely described. First, we construct a model for $\underline{R}$.

Consider the vector of reflection coefficients $\underline{r}$ and associated RVs $\underline{R}$. Assume each RV in $\underline{R}$ is independent and identically distributed. The RVs independence means that the joint pdf for $\underline{R}$ can be written in terms of each component pdf, and their identical distribution means each component pdf is identical:

$$\begin{aligned} f_{\underline{R}}(\underline{r}) &\triangleq f_{R_1,R_2,\cdots,R_N}(r_1,r_2,\cdots,r_N) \\ &= f_{R_1}(r_1)f_{R_2}(r_2)\cdots f_{R_N}(r_N), \text{ independent} \\ &= \prod_i f_R(r_i), \text{ identically distributed} \end{aligned} \qquad (2.7a)$$

The stochastic model for impedance developed in the last chapter suggests the following pdf to describe each RV in $\underline{R}$:

$$f_R(r) = \lambda\delta(r) + (1-\lambda)G(\sigma_r^2,r^2) \qquad (2.7b)$$

$$G(\sigma_r^2,r^2) = \text{Gaussian pdf of zero mean and variance } \sigma_r^2$$

The parameter λ represents the blockiness of the impedance log or, equivalently, sets the mean waiting time for reflections at $1/(1-\lambda)$. In summary, we propose a non-Gaussian, independent model for reflection coefficients.

A description of the noise and its correlation with $\underline{R}$ is motivated by considering equation (2.4):

$$x^k_t = r_t + \left[y * (f^k - f)\right]_t$$

Substituting $(r*b)_t$ for y_t, where b is the actual bubble (unknown) in the above equation, gives

$$x^k_t = r_t + (r * \nabla^k)_t$$

$$\nabla^k_t \triangleq \left[b * (f^k - f)\right]_t$$

The waveform ∇^k_t is the residual wavelet present in the deconvolution at the k-th iteration. We imagine that ∇^k_t is a long and oscillatory waveform which produces a noisy output when convolved with r_t:

$$x^k_t = r_t + n^k_t \qquad (2.8a)$$

$$n^k_t \triangleq (r * \nabla^k)_t = \sum_l r_l \nabla^k_{t-l} \qquad (2.8b)$$

Some properties of the noise can be derived from equation (2.8b). Again, dropping superscripts, we have

$$EN_iN_j = E\left(\sum_l R_l \nabla_{i-l} \sum_m R_m \nabla_{j-m}\right) \qquad (2.9)$$

But from equation (2.7), $\underline{R}$ is white, hence

$$ER_iR_j = (1-\lambda)\sigma_r^2\delta_{ij}$$

Substituting this expression in (2.9) gives

$$EN_iN_j = (1-\lambda)\sigma_r^2 \sum_l \nabla_{i-l}\nabla_{j-l}$$

Since ∇_t is long and oscillatory, the sum in the above equation contributes a non-zero value only when i=j:

$$EN_iN_j = \sigma_N^2\delta_{ij} \tag{2.10}$$

$$\sigma_N^2 \triangleq (1-\lambda)\sigma_r^2 \sum_l \nabla_l^2$$

From equation (2.10), we conclude that the noise is white. If in addition its pdf is taken as Gaussian, the result is a white, Gaussian model for the noise.

All that remains is to consider the correlation between $\underline{R}$ and $\underline{N}$. These two sequences are certainly correlated [equation (2.8b)], but when compared to the magnitude of the noise variance, the correlation is negligible. To understand this point, consider the autocorrelation of $\underline{X}$:

$$\begin{aligned} EX_iX_j &= E(R_i + N_i)(R_j + N_j) \\ &= \left[(1-\lambda)\sigma_r^2 + \sigma_N^2\right]\delta_{ij} + ER_iN_j + N_iR_j \end{aligned} \tag{2.11}$$

The cross-terms can be calculated using equation (2.8b):

$$ER_iN_j = (1-\lambda)\sigma_r^2\nabla_{j-i}$$

Again, using the assumption that ∇_t is long and oscillatory, we conclude that $\sigma_N^2 >> ER_iN_j$. From equation (2.11) this means that $EX_i^2 >> EX_iX_j$, $j \neq i$, or in effect that $\underline{X}$ is white. This suggests that $\underline{R}$ and $\underline{N}$ can be taken to be independent (the relation uncorrelated ≠ independent is ignored). Hence we have the model

$$X_i = R_i + N_i \tag{2.12a}$$

$$ER_iN_j = 0$$

The pdf for each RV in $\underline{X}$ is the convolution of the pdfs for R and N. Using the fact that N is Gaussianly distributed,

$$f_N(n) = G(\sigma_N^2, n^2) \tag{2.12b}$$

we have

$$f_X(x) = f_N(\ ) * f_R(\ )$$

$$f_X(x) = \lambda G(\sigma_N^2, x^2) + (1-\lambda) G(\sigma^2, x^2) \tag{2.12c}$$

$$\sigma^2 \triangleq \sigma_N^2 + \sigma_r^2$$

The pdf in equation (2.12c) will be referred to as a normal mixture. Such a model has been used to characterize impulsive noise, superposed in background noise, of telephone circuits (Miller and Thomas, 1976).

In summary, the probabilistic model for the deconvolutions is given by equations (2.7b) and (2.12). Using this model, expressions for ${}_2r_i$ and ${}_1r_i$ can be derived. Because of the approximation involved in deriving the model, these expressions give sub-optimal estimators. During the latter stages of iteration (i.e. the algorithm is approaching convergence), the model is applicable since $\underline{R}$ and $\underline{N}$ are independent and $\underline{N}$ is Gaussianly distributed. In the early stages of iteration, however, $\underline{R}$ and $\underline{N}$ are heavily correlated and $\underline{N}$ is more uniform than Gaussian. This point should be kept in mind when selecting parameters for the algorithm.

The next section gives the derivation of the sub-optimal estimators.

2.5. *Sub-Optimal Estimators*

The independence of $\underline{X}$ means that ${}_2\tilde{r}_i$ and ${}_1\tilde{r}_i$ will depend only on x_i. Hence equation (2.5) becomes

$$ {}_{2}\tilde{r}_i(x_i) = \int r_i f_{R_i|X_i}(r_i|x_i)dr_i $$

As a notational convenience, the subscript i is henceforth dropped and ${}_{2}\tilde{r}_1(x_1)$ becomes $\tilde{r}_2(x)$.

Rewriting the above equation in terms of the joint density between R and N gives

$$ \tilde{r}_2(x) = \frac{\int rf_R(r)f_N(x-r)dr}{f_X(x)} \qquad (2.13) $$

It is a simple matter to calculate the integral in equation (2.13) using the model of the previous section, the result being

$$ \tilde{r}_2(x) = xh_2(x) \qquad (2.14) $$

$$ h_2(x) = \frac{\sigma_r^2}{\sigma^2}\left\{1 + c\exp\left[-\frac{x^2}{2}\left(\frac{1}{\sigma_N^2} - \frac{1}{\sigma^2}\right)\right]\right\}^{-1} $$

$$ c = \frac{\lambda \quad \sigma}{(1-\lambda)\sigma_N} $$

The estimator is composed of two parts, a linear term x weighted by a gain function $h_2(x)$. The gain function has two intuitively appealing properties: (i) $h_2(x)\geq 0$ and (ii) $h_2(|x|\rightarrow\infty) = \sigma_r^2/\sigma^2$. The first means that given a sample $\underline{x}$, composed of reflectivity and noise, independent and both zero mean, the estimates of the amplitude of the reflection coefficients have the same sign as $\underline{x}$. The second property means the asymptotic behavior of the estimator is linear.

The derivation of the least absolute value estimator $r_1(x)$ is considerably more detailed than the least squares estimator. The calculation is presented in Appendix B, where it will be evident that a closed form solution is not possible. For a restricted range of parameters, however, $\tilde{r}_1(x)$ can be approximated by a threshold device:

$$\tilde{r}_1(x) = x\, h_1(x) \tag{2.15}$$

$$h_1(x) = 0, \; |x| < x_c$$

$$= \frac{\sigma_r^2}{\sigma^2}, \; |x| \geq x_c$$

$$x_c \simeq \sqrt{2}\,\sigma_N \left\{ \ln \left[\frac{\sigma_r \lambda}{\sigma_N (1-\lambda)} \right] \right\}^{\frac{1}{2}}$$

In this case, the gain function h_1 is a step function. Note that the asymptotes of both $\tilde{r}_1(x)$ and $\tilde{r}_2(x)$ are identical.

The implementation of either equation (2.14) or (2.15) requires the specification of three parameters $(\sigma_r^2, \sigma_N^2, \lambda)$. The next section deals with their selection.

2.6. Parameter Selection and Convergence

If the ZNLs of the last section are implemented, each iteration tries to better identify the reflection coefficients and to then remove the forward wavelet with a proper inverse filtering. The objective of this section is to justify the former statement with regard to parameter selection. We start the analysis by introducing some necessary notation.

Previously, it was assumed that the parameters $(\sigma_r^2, \sigma_N^2, \lambda)$ describing the deconvolution are known ***a priori***. Actually, they are unknown, and a set denoted by $(\tilde{\sigma}_r^2, \tilde{\sigma}_N^2, \tilde{\lambda})$ must be chosen for use in the algorithm. One constraint placed on their selection is that the algorithm be data-dependent. Equation (2.12) provides the necessary condition for this constraint, for it relates the variance of X^k to the algorithm parameters:

$$(\sigma_x^2)^k = (1-\tilde{\lambda})(\tilde{\sigma}_r^2) + (\tilde{\sigma}_N^2)^k$$

or

$$(\tilde{\sigma}_N^2)^k = \frac{(\sigma_x^2)^k}{(1-\tilde{\lambda})\tilde{S}^k + 1} \qquad (2.16)$$

$$\tilde{S}^k \triangleq \frac{\tilde{\sigma}_r^2}{(\tilde{\sigma}_N^2)^k}$$

As a consequence of equation (2.16), only two parameters must be selected- $(\tilde{S}^k,\tilde{\lambda})$- with the data fixing the remaining parameter.

Say $(\tilde{S}^k,\tilde{\lambda})$ are chosen constant, independent of deconvolution and iteration; how much penalty is incurred for mismatching the true parameters with these constants? To answer this question, a penalty function must be defined. The mean square error (MSE), defined by

$$(\text{MSE})^k \triangleq E\left[\tilde{r}(X^k) - R\right]$$

is a good choice because it is inversely proportional to the iteration gain, IG, defined as

$$(\text{IG})^{k+1} \triangleq \frac{(\sigma_N^2)^k}{(\sigma_N^2)^{k+1}} \qquad (2.17)$$

To prove the proportionality, first recall the definition of n_t from equation (2.4):

$$n_t^{k+1} = \left[y * (f^{k+1} - f)\right]_t$$

$$= x_t^{k+1} - r_t$$

Hence,

$$\frac{1}{N}\sum_1^N (n_t^{k+1})^2 = \frac{1}{N}\Sigma\,(x_t^{k+1} - r_t)^2$$

Equating the time averages with ensemble averages in the above equation gives the equivalent expression

$$(\sigma_N^2)^{k+1} = E(X^{k+1} - R)^2 \tag{2.18}$$

If the system of equations in equation (2.1) wasn't over-determined (i.e. if we had one channel of data and an infinite length filter), then $X^{k+1} = \tilde{r}(X^k)$ and the term on the RHS of equation (2.18) would be the mean square error at the k-th iteration. The over-determination, however, reduces the variance of the uncorrelated noise in X^{k+1} with respect to that present in $\tilde{r}(X^k)$. Hence the inequality:

$$(\sigma_N^2)^{k+1} \leq E\left[\tilde{r}(X^k) - R\right]^2 = (MSE)^k \tag{2.19}$$

The MSE can be factored into two parts (see Appendix C):

$$(MSE)^k = P(S^k,\tilde{S}^k,\lambda,\tilde{\lambda})\ (\sigma_N^2)^k$$

where P is mnemonic for penalty function. With this substitution, equation (2.19) becomes

$$\frac{(\sigma_N^2)^{k+1}}{(\sigma_N^2)^k} \leq P$$

Comparing the above equation with equation (2.17) gives the result

$$(IG)^{k+1} \geq \frac{1}{P(S^k,\tilde{S}^k,\lambda,\tilde{\lambda})} \tag{2.20}$$

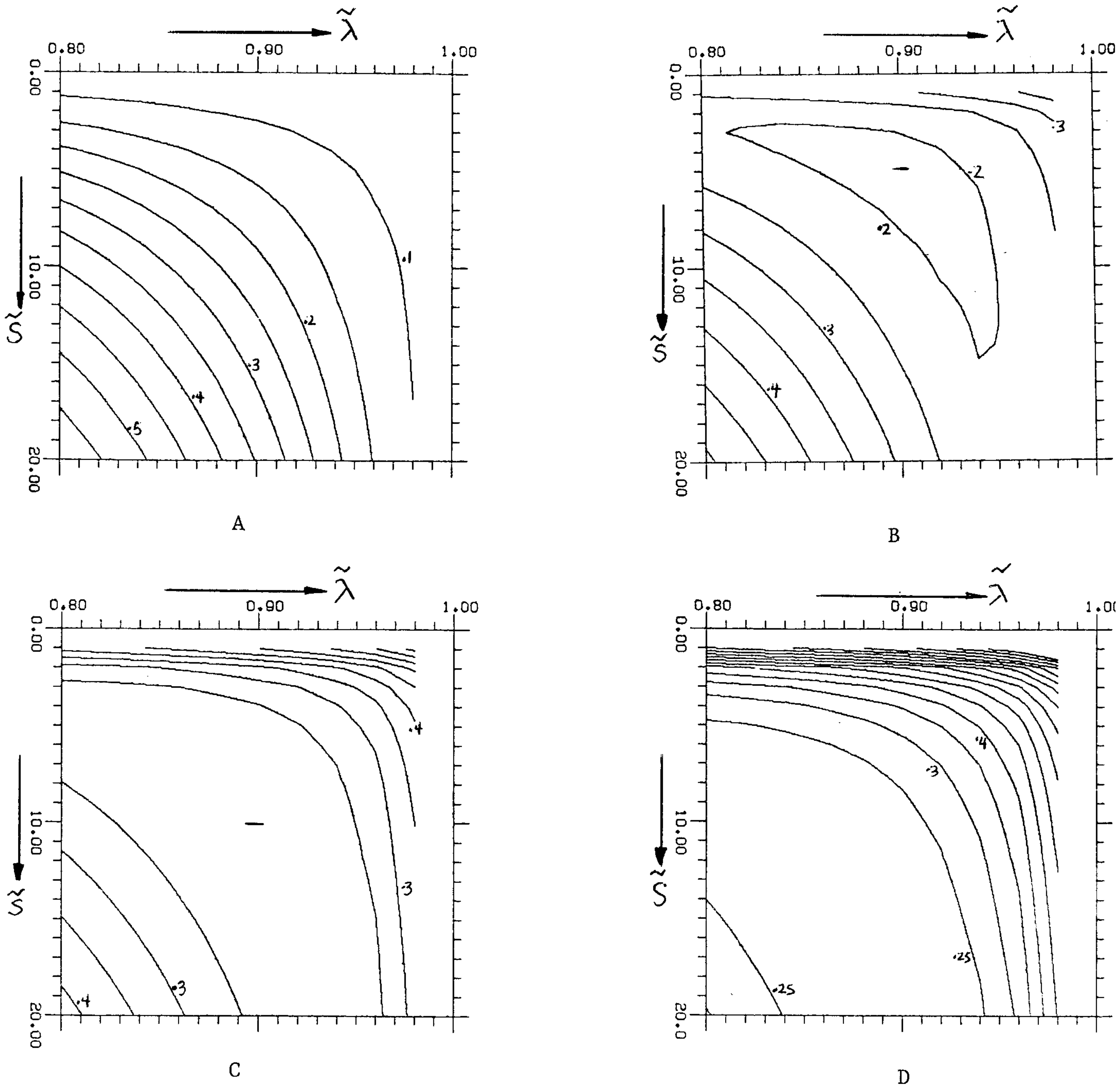
Note that the penalty function provides only a lower bound for the total iteration gain, which is composed of two parts: (1) the gain derived from P, and (2) the over-determination gain.

Our objective is to select $(\tilde{S},\tilde{\lambda})$ by "minimizing" P over a range of S and λ. This is difficult in the general case when S and λ vary over their full range, $0 < S < \infty$ and $0 \leq \lambda \leq 1$. The frequency of reflection coefficients, however, rarely exceeds 1 per 5 time units, and for this reason λ is restricted to the range $0.8 \leq \lambda \leq 0.98$. The upper limit was arbitrarily set at 1 reflection in 50 time units. The ratio of reflection to noise variance, S, can be as low as 1 at the first iteration, but rapidly increases to values of 15-20 as iteration proceeds, hence $1 \le S \le 20$. Using $\tilde{r}(x) = \tilde{r}_2(x)$, Figure 2.1 shows the surface $P(S,\tilde{S};\lambda,\tilde{\lambda})$ [a function of $(\tilde{S},\tilde{\lambda})$] for four combinations of (S,λ). See Appendix C for details of the computation. Note that the minimal value of P occurs when $(\tilde{S},\tilde{\lambda}) = (S,\lambda)$, as expected. The values of P on all plots in Figure 2.1 are generally less that 1.0, regardless of $(\tilde{S},\tilde{\lambda})$, implying an iteration gain of at least 1.0. The exception is when $S \to 20$. In other words, the surfaces $P(S,\tilde{S};\lambda,\tilde{\lambda})$ are moderately flat for a wide range of $(\tilde{S},\tilde{\lambda})$ and an estimation scheme to ensure that $(\tilde{S},\tilde{\lambda}) \simeq (S,\lambda)$ is not required.

Different approaches are available for selection of the "best" pair of $(\tilde{S},\tilde{\lambda})$. Our approach, empirical in nature, was to select $(\tilde{S},\tilde{\lambda}) = (5.0,0.9)$ on the basis of deconvolving synthetic data using three criteria: (1) quality of deconvolution; (2) rapid rate of convergence; and (3) maximum penalty =.93 for $1 \le \lambda \le 20$ and $0.8 \le \lambda \le 0.98$. The maximum penalty occurred at $(S,\lambda) = (20.,0.8)$. In Figure 2.2a, $\tilde{r}_2(x)$ has been plotted using various combinations of $(\tilde{S},\tilde{\lambda})$.

The effect of choosing $\tilde{S}$ independent of iteration is that as iteration proceeds, S increases and surpasses $\tilde{S}$. Depending on the combination (S,λ), a point is reached when the iteration gain becomes < 1. The next iteration results in a reduction in S but now the penalty < 1 and algorithm oscillation begins. The value of S at which oscillation is initiated, call it S_{max}, is quite high generally and represents the signal/noise ratio the algorithm is capable of achieving. One could then change the pair of constants $(\tilde{S},\tilde{\lambda})$ in order to achieve a better signal/noise ratio. The real power of the algorithm, however, is exhibited during the first few iterations, when deconvolutions of low S are correctly "transformed" into deconvolutions of higher S. Our choice of $(\tilde{S},\tilde{\lambda})$ is made to guarantee that we move correctly out of low S regions.

The MSE is a measure of the iteration gain of any ZNL and the same analysis was used in selecting $(\tilde{S},\tilde{\lambda}) = (5.0,0.9)$ for the absolute value non-linearity. In this case the maximum penalty occurred at $(S,\lambda) = (20.,0.8)$ and equaled 0.95. Figure 2.2 also includes plots of $\tilde{r}_1(x)$ for various parameter sets.



A

B

C

D

Figure 2.1.

(A): Effect of mismatching algorithm parameters $(\tilde{S},\tilde{\lambda})$ with the actual parameters (S,λ) is illustrated by contouring the mean square error surface for the L-2 estimator. Here $(S,\lambda) = (1.,0.9)$.

(B),(C),(D): Corresponding surfaces as S increases through 5.,10., and 20. keeping λ fixed at 0.9. Note the minimal mean square error occuring when $(\tilde{S},\tilde{\lambda}) = (S,\lambda)$.

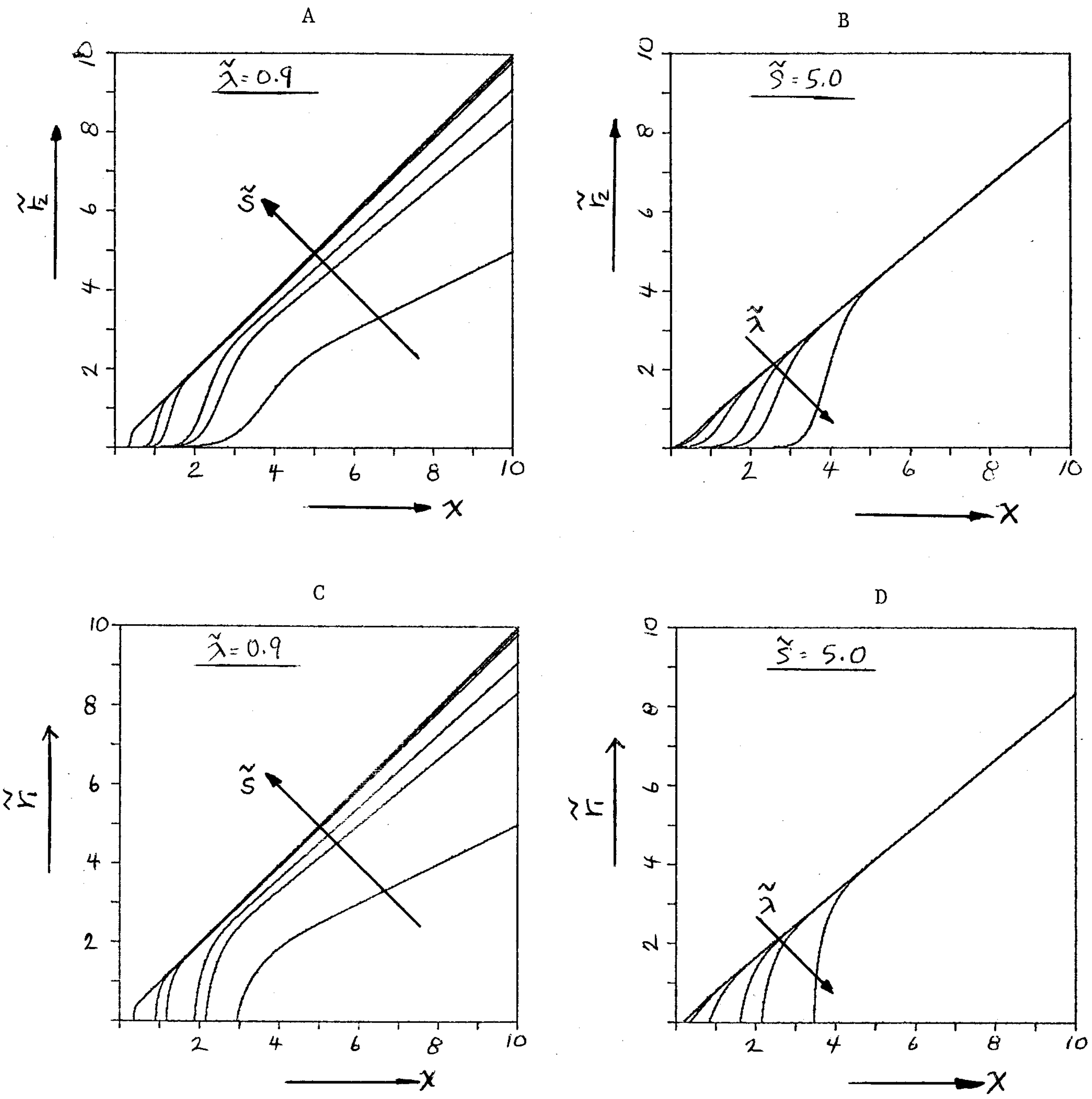


Figure 2.2.

(A),(B): The L-2 estimator plotted for $\tilde{\lambda}$ and $\tilde{S}$ fixed. In (A), $\tilde{S}$ ranges through 1.,5.,10.,50.,100., and 1000. In (B) $\tilde{\lambda}$ ranges through 0.1,.2,.5,.8,.9, and 0.99. Both abscissa and ordinate are normalized by the variance σ_X.

(C),(D): Corresponding plots for the L-1 estimator.

In summary, based on the concept of iteration gain, the set of algorithm parameters $(\tilde{S}^k, \tilde{\lambda}) = (5.0, 0.9)$ can be used at each iteration, independent of the deconvolution, and convergence will be attained. Our analysis, however, assumed that the squared error had zero variance and could be completely characterized by its mean (i.e. assumed an infinite number of samples). Convergence of a particular dataset is certainly enhanced, therefore, when a large amount of independent data constitutes the dataset.

2.7. *Bussgang Convergence*

The convergence properties discussed in the last section have tacitly assumed that the only convergent sequence is the original reflection sequence. As a preliminary to discussing this assumption, the family of processes to which any iterative algorithm utilizing a ZNL converges is now defined.

Gray (1979) discovered that the variable norm deconvolution algorithm converged to sequences that were Bussgang [see Barrett and Lampard (1955)]. This is true of any iterative algorithm utilizing a ZNL. To show this, begin by writing equation (2.2) in the frequency domain (temporarily, upper case letters refer to frequency domain variables and not RVs):

$$Y\bar{Y}F^{k+1} = Z^k\bar{Y}$$

In the above equation, Z refers to any ZNL of which $\tilde{r}_1$ and $\tilde{r}_2$ are two particular cases. If F^{k+1} is proportional to F^k, the algorithm has converged:

$$\alpha Y\bar{Y}F^k = Z^k\bar{Y}$$

Multiplying the above through by $\bar{F}^k$ gives

$$\frac{(X\bar{X})^k}{(Z\bar{X})^k} = \alpha \tag{2.21}$$

Equating $(X\bar{X})^k$ with the power spectral density of the process $\{X_t^k\}$, equation (2.21) is the definition for $\{X_t^k\}$ to be Bussgang. Once the process $\{X_t^k\}$ has converged to a Bussgang process, any ZNL can be used without affecting the result. Of course $\{X_t^k\}$ could converge to a number of different Bussgang processes, depending on the particular ZNL used. A similar situation occurs when a function is being optimized. There, the possibility of the function having many local minimum (maximum) points is analogous, in the present context, to the process $\{X_t^k\}$ converging to different Bussgang processes.

Next, it will be demonstrated that an independent, non-Gaussian sequence convolved with a wavelet is not Bussgang. Begin by hypothesizing that reflection coefficients are independent and are described by a normal mixture pdf. The data results from the convolution of a forward wavelet with the reflection sequence. If after iterating k times, there is a residual wavelet of the form (1,a) present in X^k, the following theorem, proved in Appendix D, shows that X^k is not Bussgang:

> ***Theorem***: If a stationary, independent process, characterized by a normal mixture pdf, is input into a two-term filter, the new sequence is not Bussgang.

Another theorem, also proved in Appendix D, deals with arbitrary non-Gaussian pdfs:

> ***Theorem***: An independent stochastic process convolved with a delta-like wavelet is Bussgang if and only if the original process was Gaussian.

Note that if the observation sequence becomes Gaussian (not necessarily independent), the algorithm has reached a convergent point since any Gaussian process is necessarily Bussgang.

Summarizing the results, we have that (1) the equilibrium points of iterative deconvolution algorithms are Bussgang processes, (2) an independent non-Gaussian sequence convolved with a wavelet is not Bussgang, and 3) any Gaussian process is Bussgang. Therefore, iterative deconvolution algorithms have two equilibrium points - either $\{X_t^k\}$ is

Gaussian or it is independent. The ZNL used in the algorithm controls the ***direction*** of convergence taken. In addition, since the algorithm has access to only a sample of $\{X_t^k\}$, the ZNL also affects the rate of convergence, or equivalently, the efficiency of the algorithm.

Actually, random telegraph waves are also Bussgang and are therefore equilibrium points of iterative deconvolution algorithms. One might conclude from this that a possible output of iterative deconvolution algorithms would be integrated reflectivity or impedance sequences. We conjecture that such sequences, although they are equilibrium points, are not ***stable***. To test this conjecture, an impedance function was input to the algorithm. After iterating 15 times, a differentiated version was observed as output, thus supporting the conjecture.

The final two sections of the paper present the results of applying the algorithm to synthetic and real data.

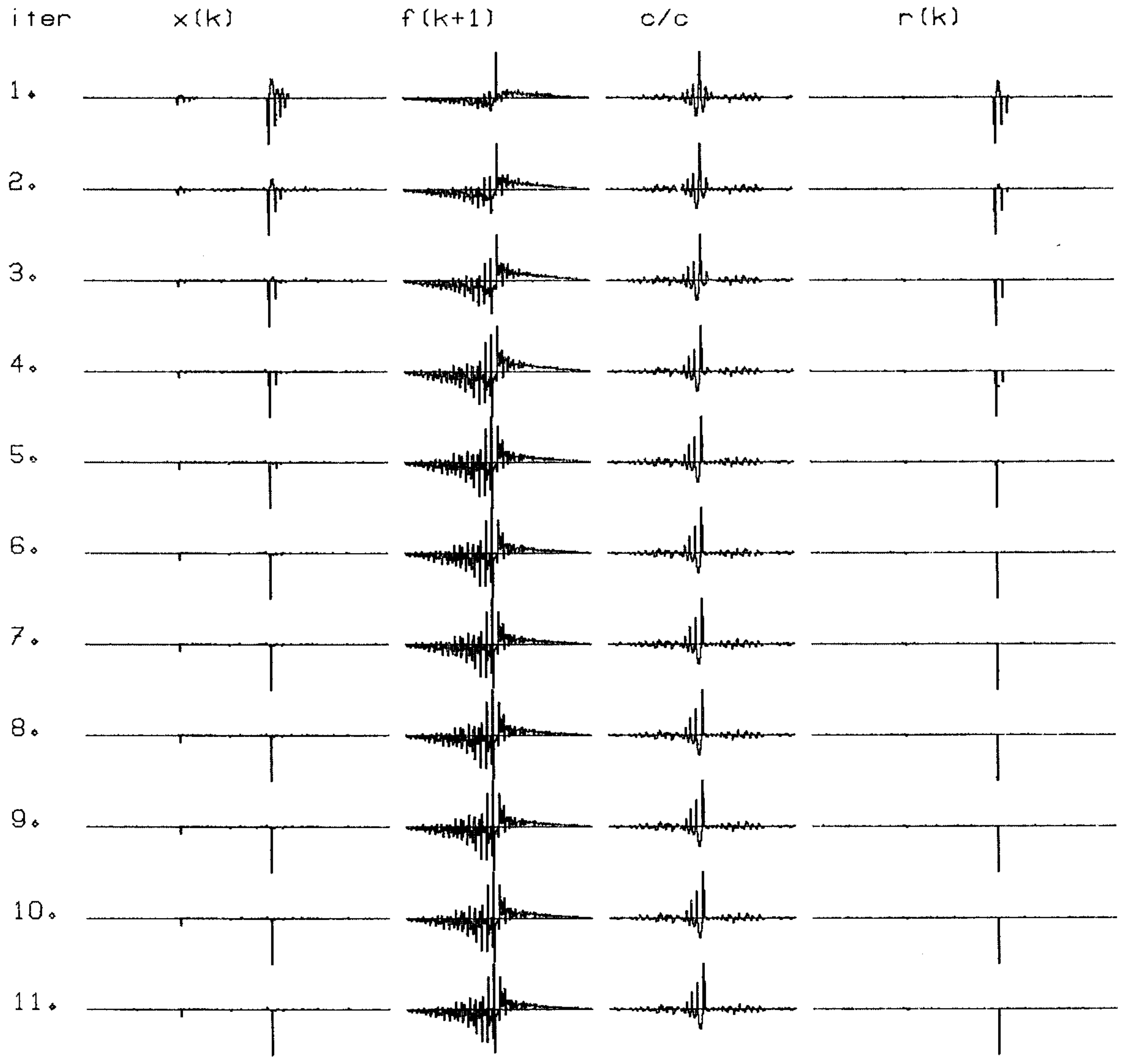
2.8. Synthetic Data

Using the model in equation (2.7b), reflectivity sequences were synthesized. The density of reflections varies according to the value of λ used. The series of Figures 2.3-2.5 shows the effect of increasing the density of reflections from 1 in 20 points (λ=0.95) to 1 in 10 points (λ=0.90), and finally to 1 in 7 points (λ=0.85). In all cases, the reflectivity sequence r_t was 100 points long, the bubble b_t=23 points, and the resulting seismograms were padded with zeros to create a 128-point trace. Five data channels were generated. The inverse filter f_t^k had a length of 201 points, and the starting filter $f_t^1 = \delta(t-100)$.

The top plots (a) in each figure shows the progress of the first channel as iteration proceeds. Along with the deconvolution x_t^k, the inverse filter f_t^k, the crosscorrelation of data with estimated reflection sequence $y\infty\tilde{r}_2^k$, and the estimated reflection sequence are plotted. The bottom plot (b) summarizes the final result by showing the original sequence r_t, the data y_t, and the deconvolved result x_t^6.

Convergence in all three cases is obtained after 5 iterations. As the reflection density increases, the algorithm fails to distinguish the

A



B

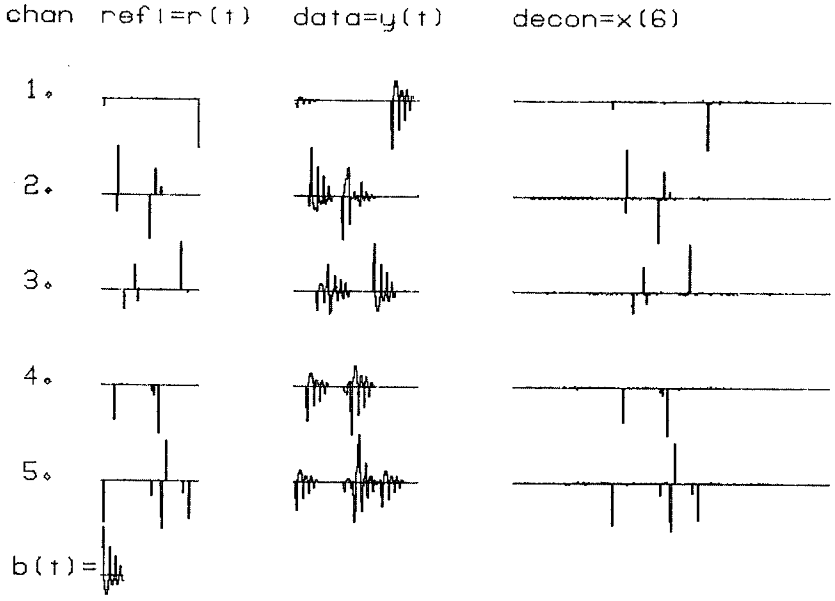
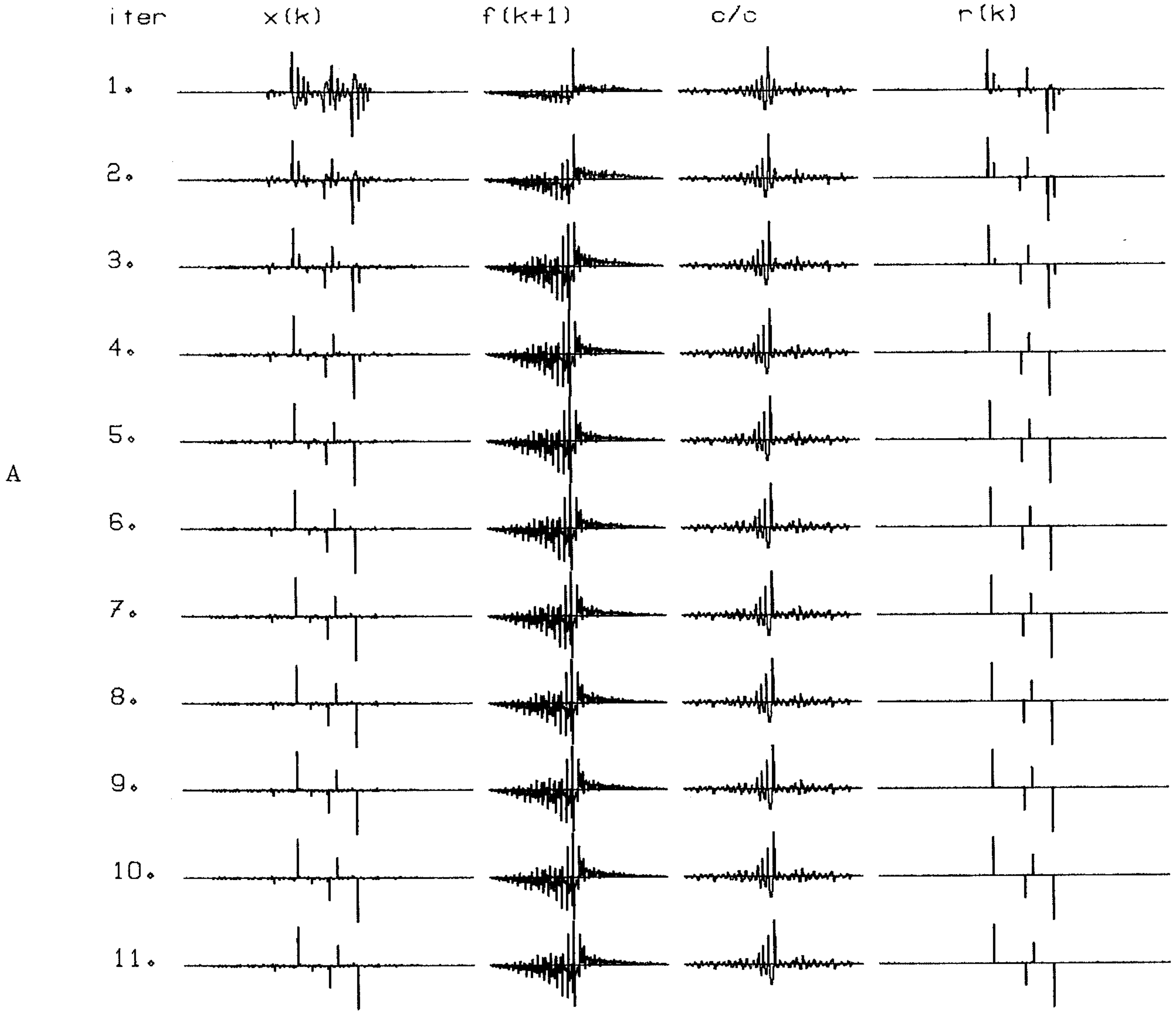


Figure 2.3a. Progress of channel 1 as iteration proceeds. The index k refers to iteration and crosscorrelation = y ∞ $\tilde{r}_2$ (k). Convergence is obtained after 5 iterations.

Figure 2.3b. Comparison of reflectivity, data and deconvolutions for the 5 channels. Reflectivity was synthesized using $\lambda = 0.95$.



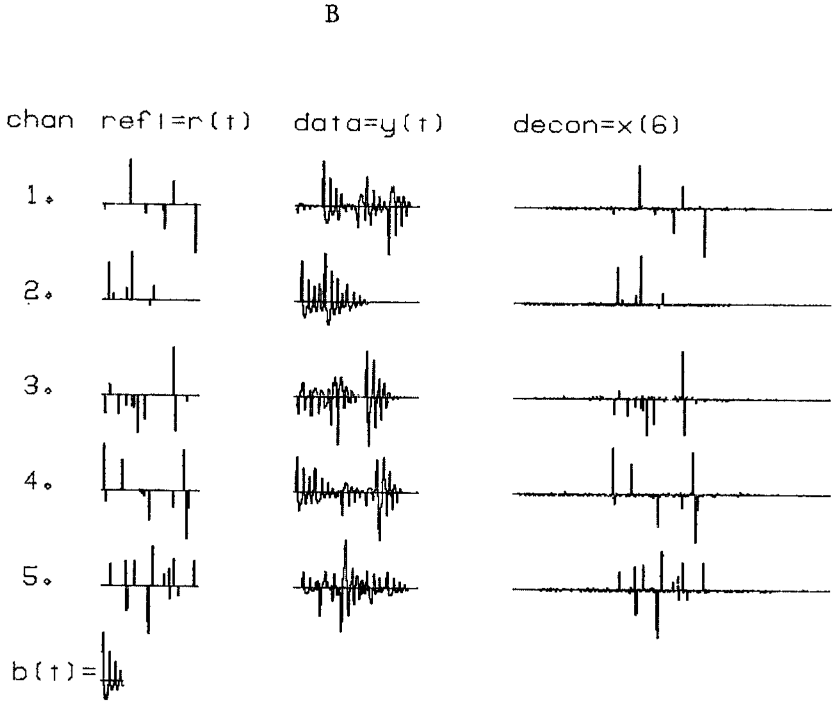


Figure 2.4. Same as Figure 2.3 except λ = 0.90. Note that small reflection coefficients are difficult to distinguish from noise in the deconvolved data.

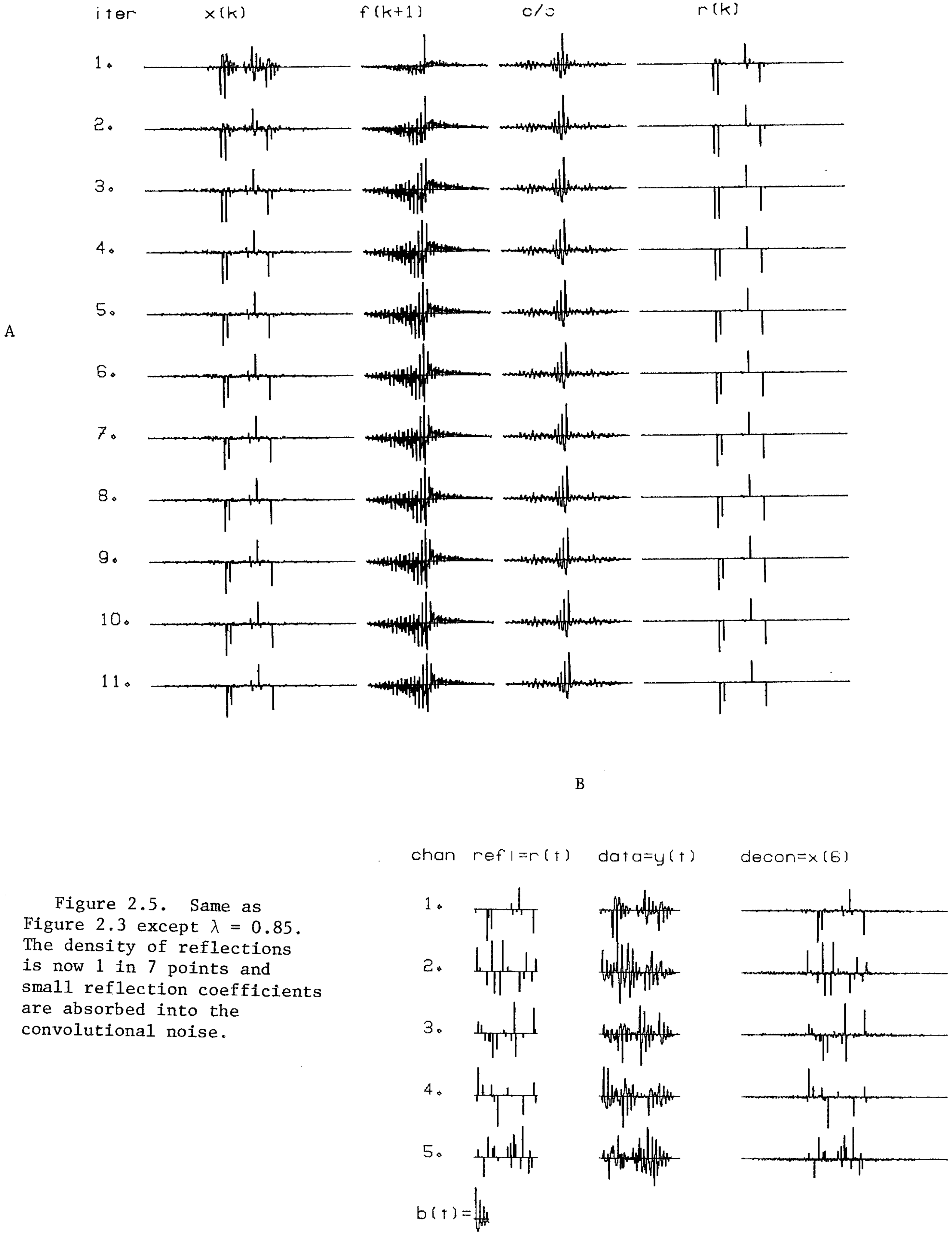


Figure 2.5. Same as Figure 2.3 except λ = 0.85. The density of reflections is now 1 in 7 points and small reflection coefficients are absorbed into the convolutional noise.

small reflection coefficients from convolutional noise. We have performed the same experiments after adding Gaussian and uniform noise to the data y_t, and the algorithm is stable, with an expected degradation in the deconvolved results.

These synthetics are presented to demonstrate that selecting $(\tilde{S}^k,\tilde{\lambda}) = (5.0,0.9)$ independent of the actual data gives rapid convergence and acceptable deconvolutions. It should be emphasized that the estimated reflection sequence is not the deconvolved result. The latter can be obtained only by filtering the data y_t.

2.9. Real Data

The first real data case consists of 16 channels of a common shot gather (CSG) recorded using an aqua-pulse source. This same dataset has been used extensively by Gray (1979). Figure 2.6a shows the gather, along with the average amplitude spectrum and autocorrelation function. Each channel is 512 points long. Figures 2.6b and 2.6c show the deconvolution using both a prediction-type algorithm (i.e. minimum-phase wavelet) and six iterations of our algorithm. Note that the correlation function gives a misleading impression of the quality of deconvolution. In our judgment, Figure 2.6c is cleaner that Figure 2.6b, but the correlation function would suggest otherwise. The inverse filter was 401 points long in our deconvolution. This raises the question of why the sea-floor multiple is not removed by the filter. The answer: since the separation of seafloor primary and multiple changes as the offset changes, the algorithm treats multiples as primaries.

The last example is a CSG supplied by Digicon. The gather, Figure 2.7a, consists of 48 channels, each 1000 points long. The source was a maxi-pulse bubble, which was field-recorded and is shown on channels 25 and 48. Spiking the recorded bubble by frequency domain division resulted in Figure 2.7b. Applying a predictive-type algorithm resulted in Figure 2.7c and six iterations of our algorithm gave Figure 2.7d. The difference in the last two results is indicative of the role the phase of the wavelet has in the deconvolution. In the case of the aqua-pulse bubble, a minimum-phase approximation is actually quite good, and the

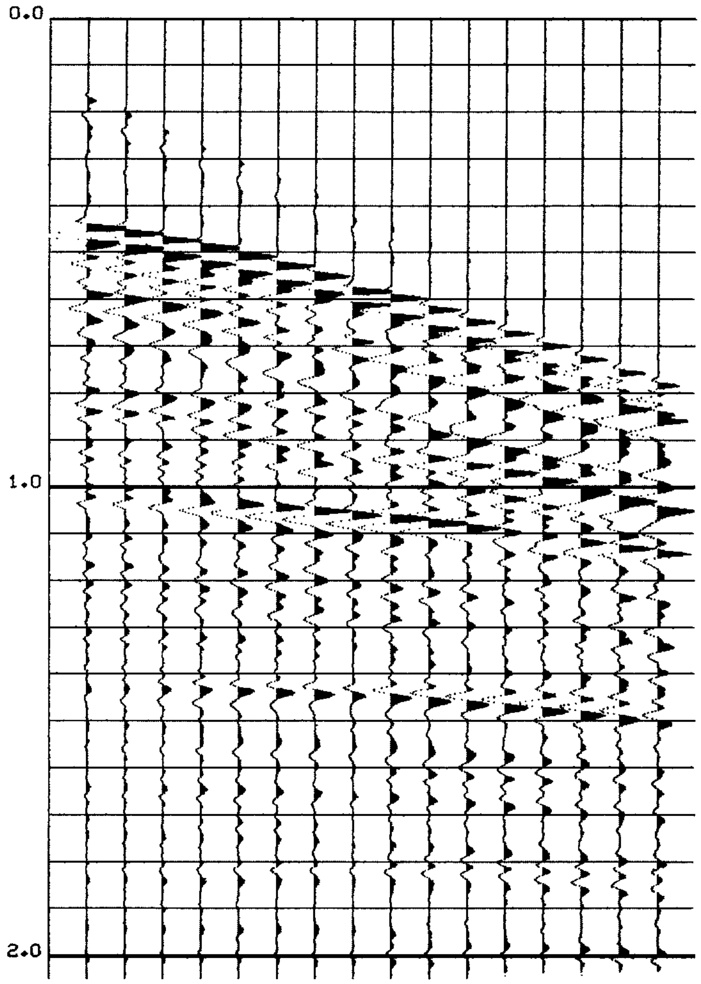
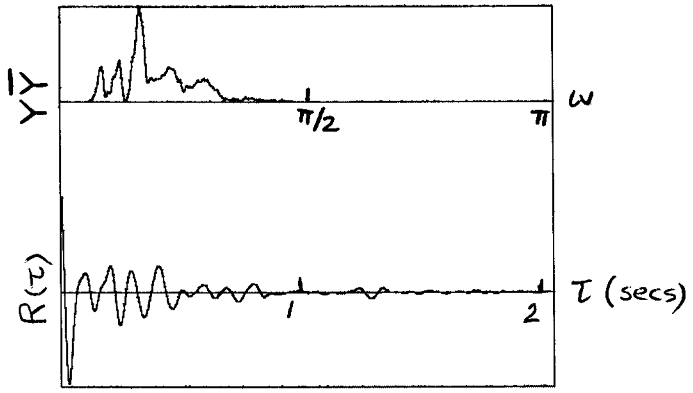


Figure 2.6a. Top plot is a common shot gather (CSG) using an aquapulse source, 16 channels, each 512 points long. Bottom plot is the average power spectrum $Y\overline{Y}(\omega)$ and autocorrelation function $R(\tau)$ of the CSG.



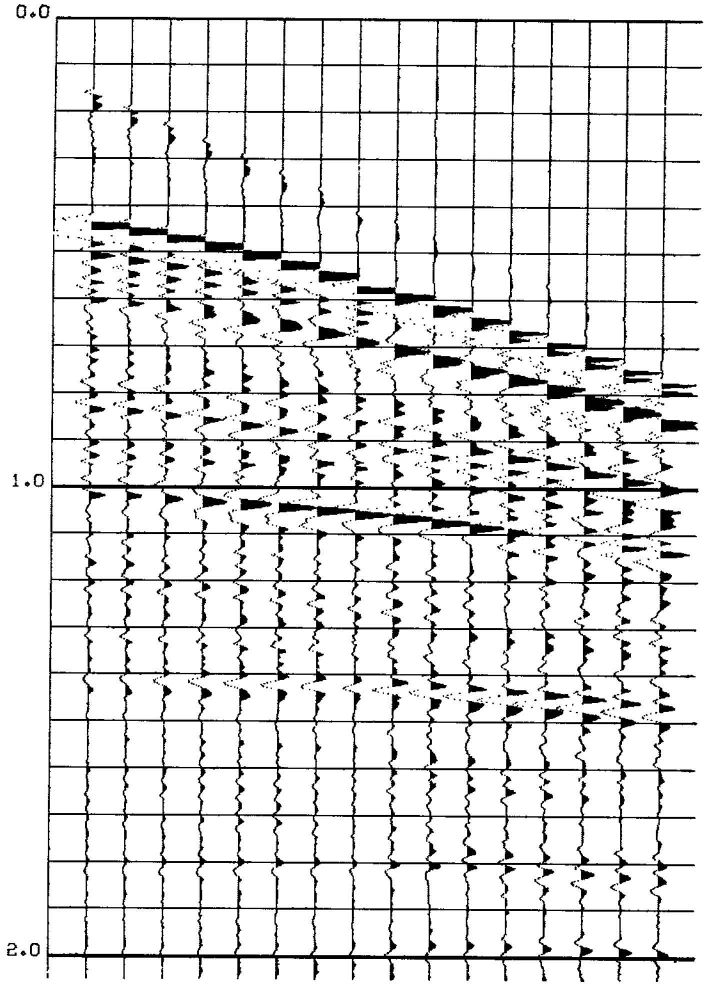


Figure 2.6b. Deconvolution of Figure 2.5a using minimum phase prediction filter.

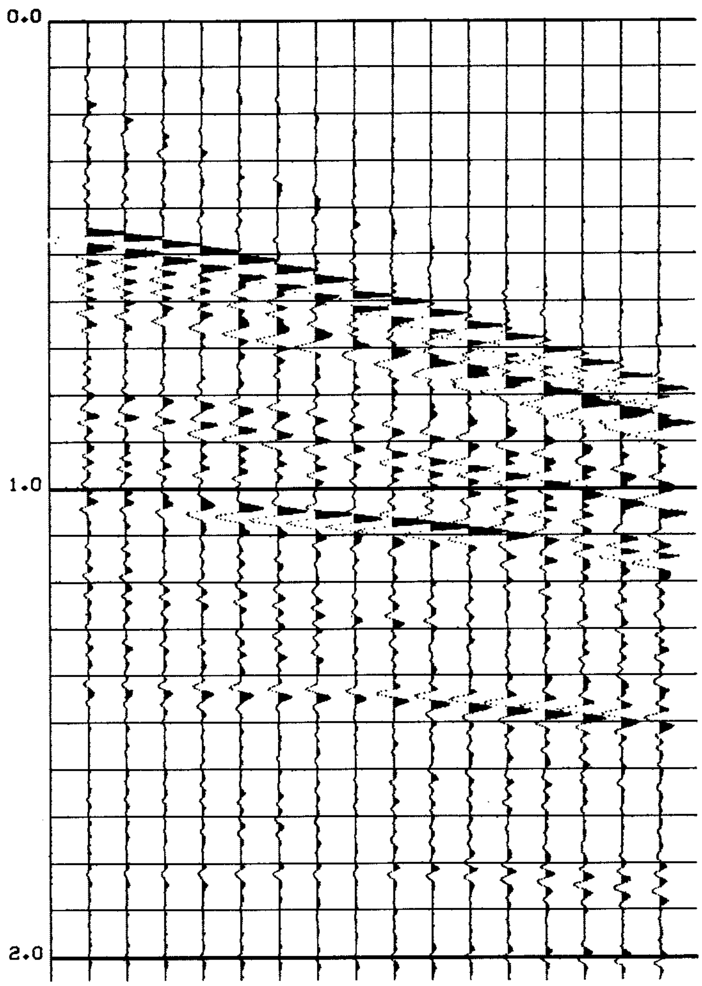
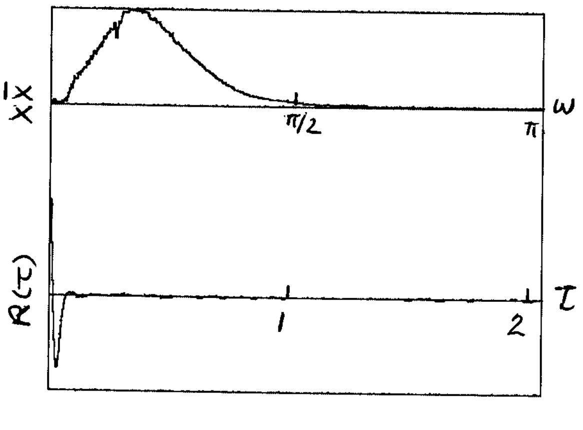
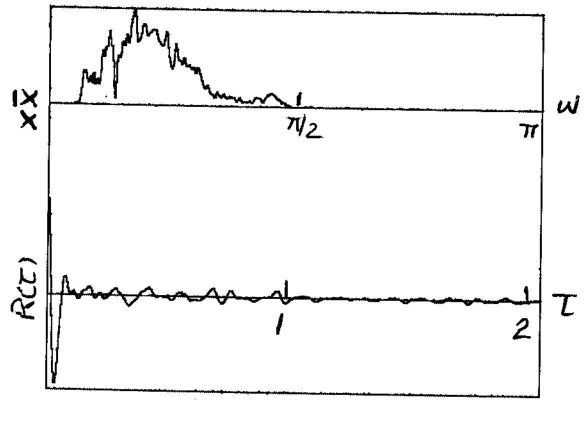


Figure 2.6c. Deconvolution of Figure 2.5a after 6 iterations. There is a slight precursor present caused by the non-minimum-phase characteristic of the inverse filter.





differences between Figures 2.6b and 2.6c are negligible. In the present case, however, there is no comparison between the two, and the additional cost of our algorithm is justified. Finally, as an experiment, we performed predictive deconvolution on Figure 2.7d to give Figure 2.7e. Even though the spectrum is whitened, the process has degraded the result by inserting erroneous events and collapsing valid pairs of events to a spike. Figure 2.8 is a summary of the power spectrums and correlation functions of Figures 2.7a-2.7e.

2.10. Conclusions

The theory proposed in this paper gives an intuitive feeling for what iterative deconvolution algorithms utilizing a ZNL can accomplish. Our approach is parametric, in the sense that a probabilistic model describing reflection coefficients must be prescribed. The particular model used was based on earlier work that modeled impedance as a Markov chain and treated reflectivity as a derived process (i.e. differential of logarithm of impedance). Based on this model, two classes of ZNLs were derived: (1) those obtained by minimizing a squared error function and (2) those obtained by minimizing an absolute value error function. In order to select parameters for the ZNLs, the term penalty function and iteration gain were defined. We demonstrated that for a wide range of observations, only one set of parameters need be used in the algorithm. The choice of parameters was made under the constraint that the iteration gain be greater that one.

An old result, that the convergent points of iterative deconvolution algorithms are Bussgang processes, can be reinterpreted using the theory. Under the hypothesis that reflection sequences are independent and non-Gaussian, the algorithm converges to either the original reflection sequence or to a Gaussian sequence. The direction of convergence is determined solely from the ZNL.

In summary, by utilizing the non-Gaussianness of a particular time series, it is possible to extract useful information. This estimation can be obtained, however, only by using non-linear estimators. In our case, the estimators were particularly simple because we assumed

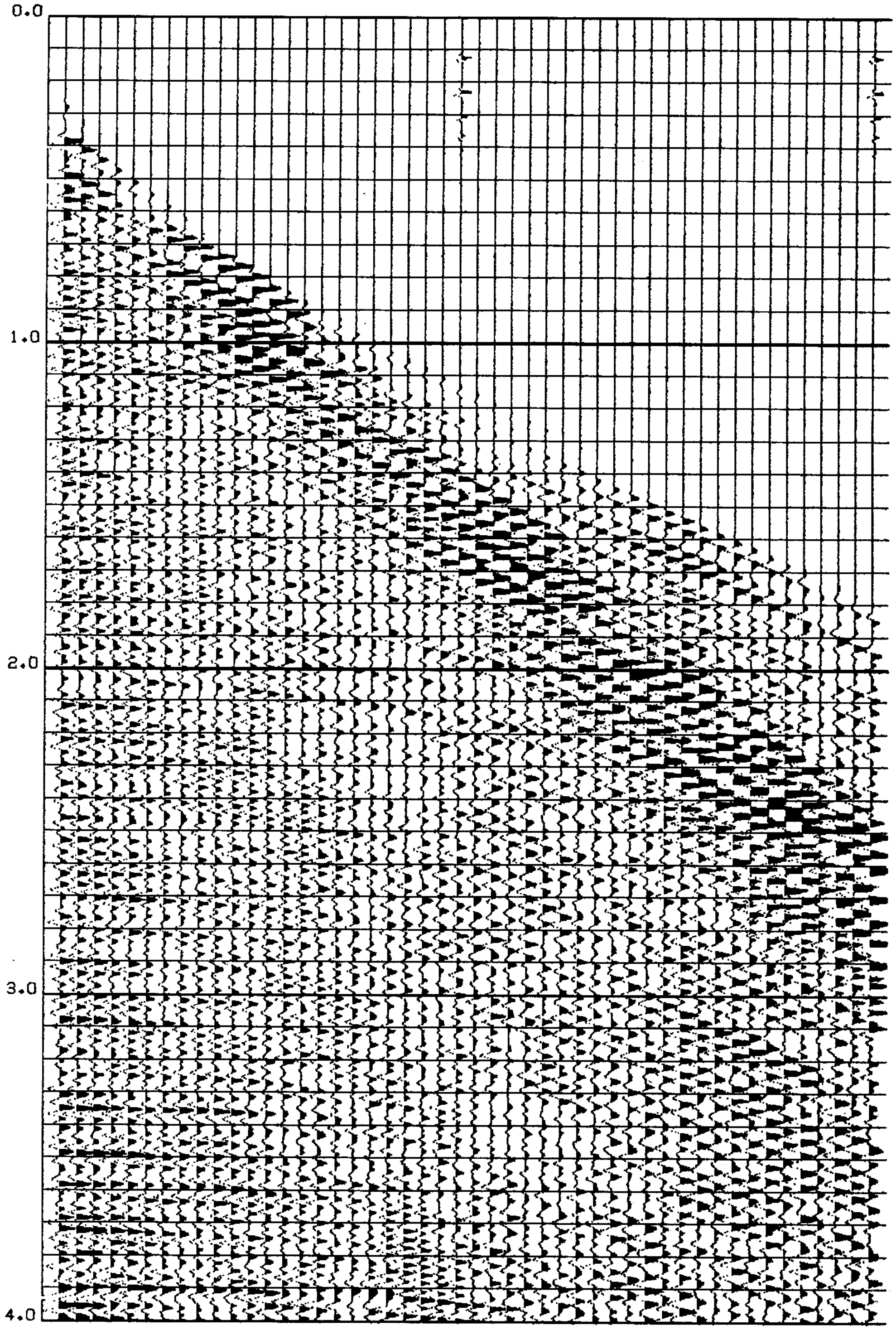


Figure 2.7a. A CSG using maxi-pulse source. 48 channels, 1000 points per channel. The recorded bubble is overlaid on channels 25 and 48. Data supplied by Digicon.

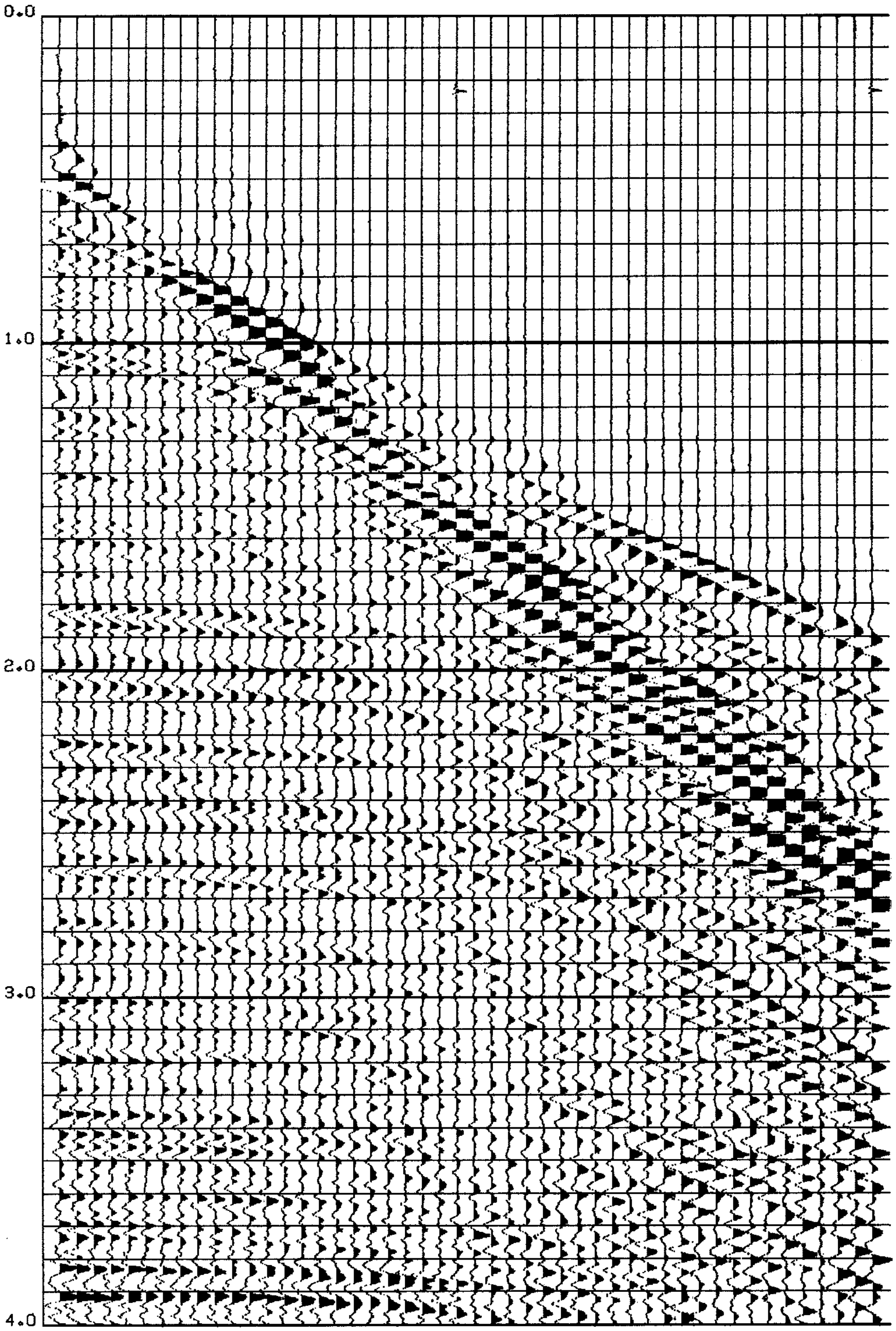


Figure 2.7b. Deconvolution of Figure 2.7a by frequency domain division using the recorded bubble. A slight precursor is caused by using a divisor of the type $(B\bar{B} + \varepsilon)$.

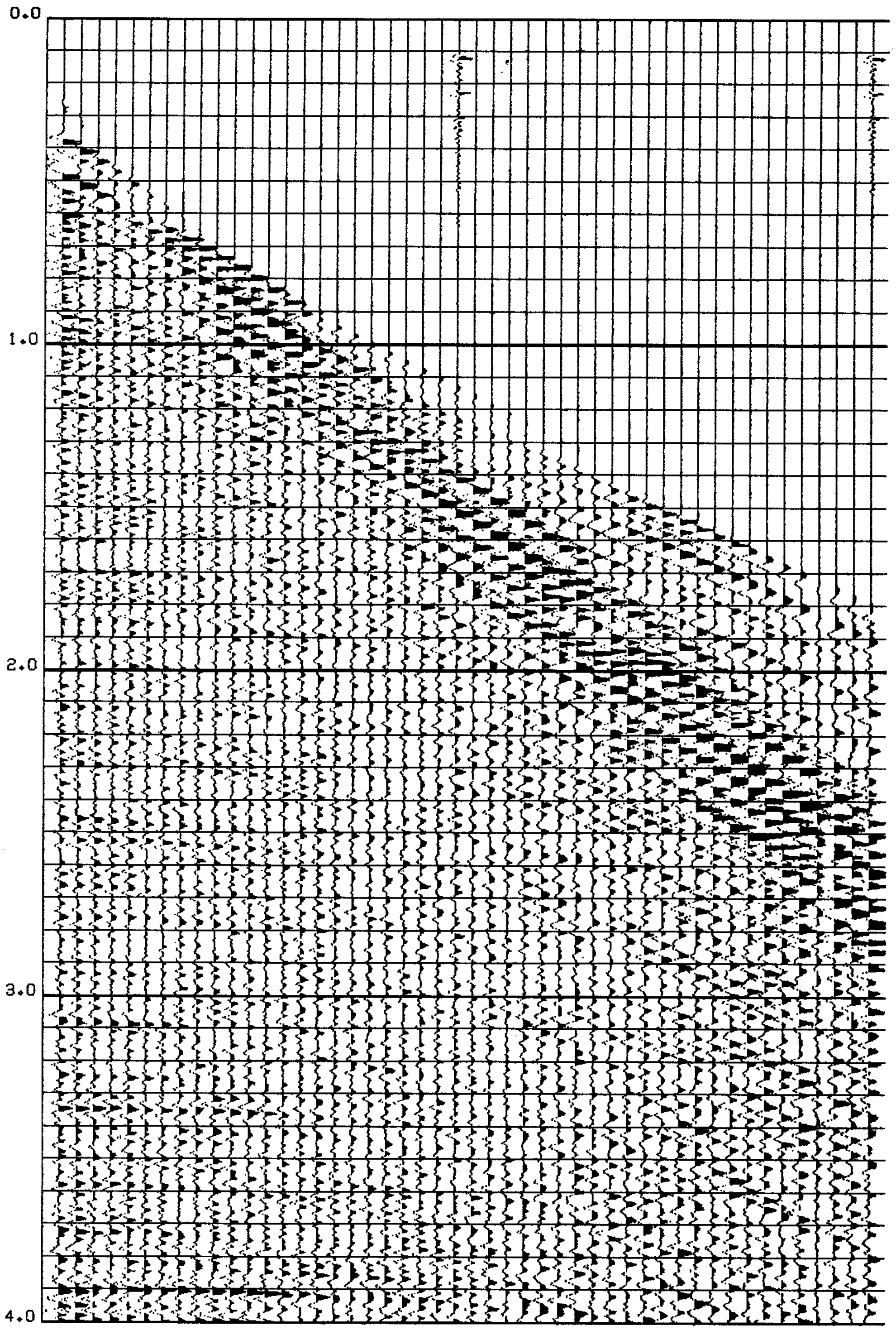


Figure 2.7c. Deconvolution of Figure 2.7a by minimum phase prediction filter. Compared to the "answer" of Figure 2.7b, it is evident that the minimum-phase assumption prevents a successful deconvolution.

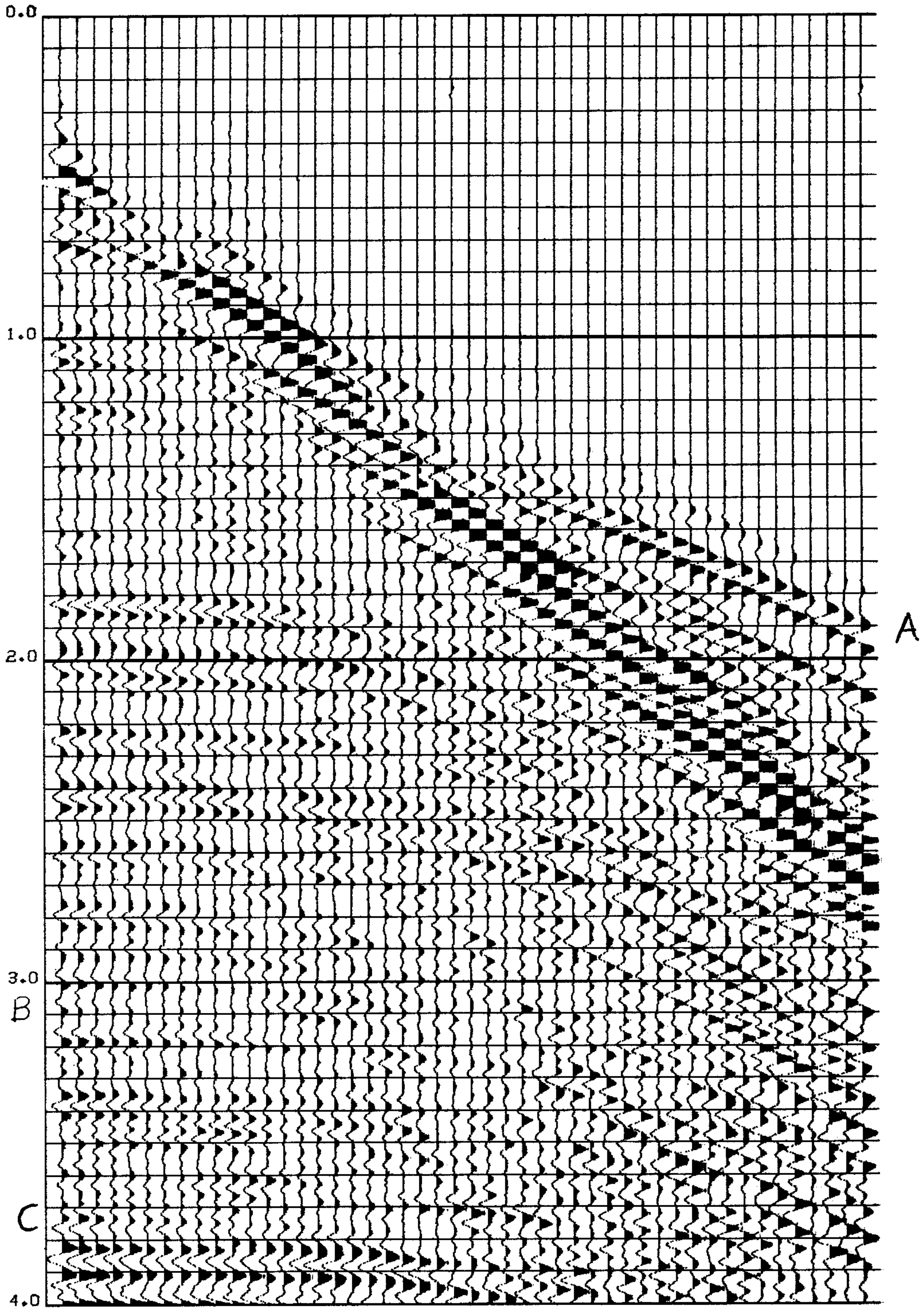


Figure 2.7d. Deconvolution of Figure 2.7a after six iterations. The frequency content is lower than Figure 2.7b, but compared to predictive deconvolution, the result is superior.

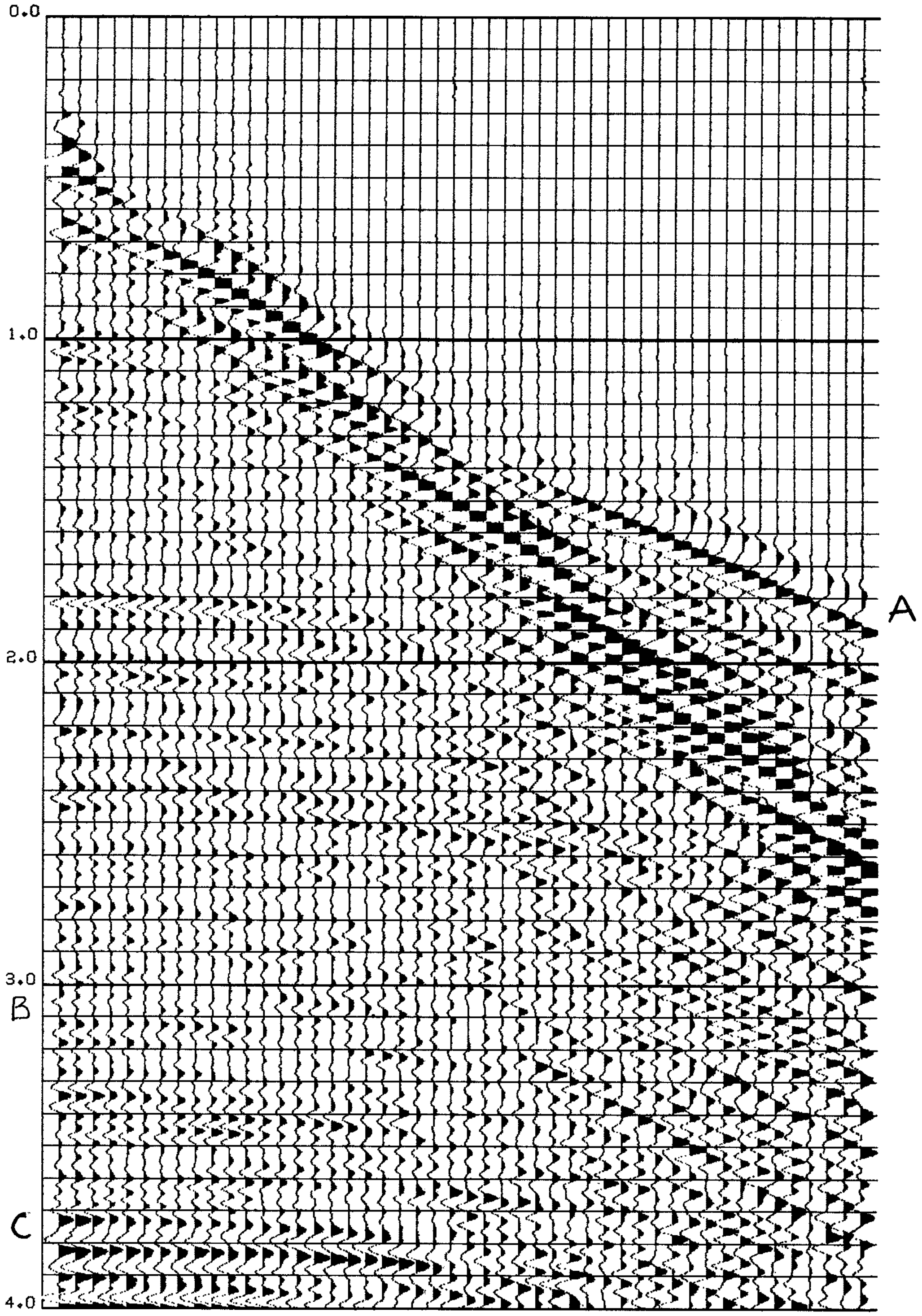


Figure 2.7e. Predictive deconvolution of Figure 2.7d. This process actually degrades the result. Compare, in particular, areas A, B and C on the two figures.

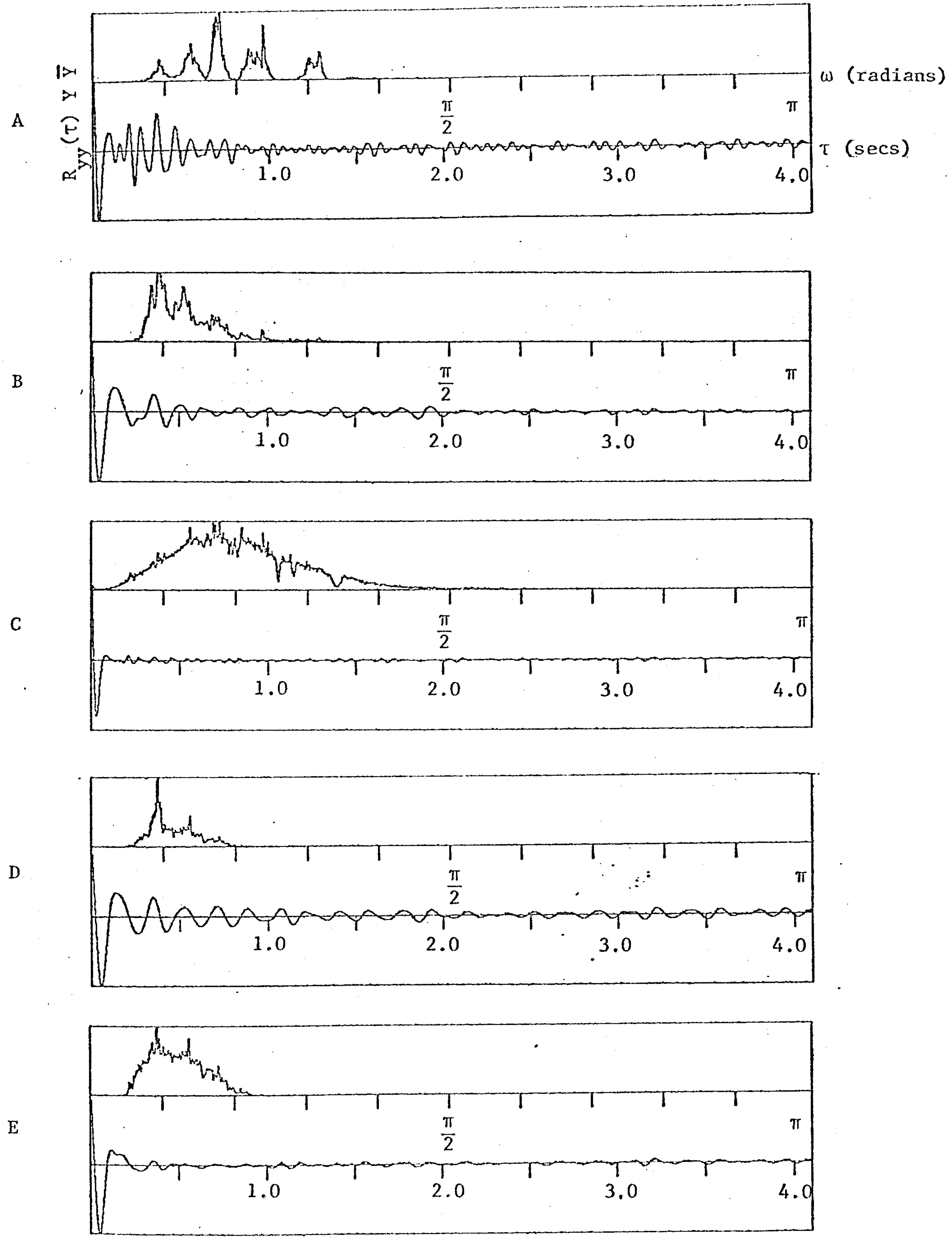


Figure 8. Each plot contains average power spectrum (top) and correlation function (bottom) for the corresponding data in Figures 2.7a -2.7e.

independent reflection coefficients.