Chapter 3

Deconvolution

- The objective of deconvolution is to increase the vertical resolution of the seismic data by:

    o Compressing the source wavelet (to a zero-phase spike, if possible)

    o Removing multiples

The seismic convolutional model

- The seismic convolutional model is used to explain how the seismic trace is formed.

- The seismic convolutional model approximates the earth by a linear system.

- A <u>linear system</u> is one whose output o(t) is given by the convolution of its input i(t) with its impulse response r(t): $\boxed{o(t) = i(t) * r(t)}$.

- The impulse response of a system is the system's output when the input is an impulse (delta) function. It is a series of impulses whose distribution and magnitude depend on the system's properties.

- In the seismic convolutional model:

    o The system's output is o(t) = s(t), where s(t) is the recorded seismic trace.

    o The system's input i(t) = w(t), where w(t) is the wavelet generated by the seismic source.

    o The system's impulse response r(t) = e(t), where e(t) is a series of impulses corresponding, in time and amplitude, to the reflection coefficients at layer boundaries. e(t) is also known as the reflectivity (series).

- According to this model, the <u>seismic trace s(t)</u> is given by:

$$s(t) = w(t) * e(t). \qquad (3.1)$$

- A random noise component n(t), if present, is additive; hence, the <u>noisy seismic trace</u> becomes:

$$s_n(t) = w(t) * e(t) + n(t). \qquad (3.2)$$

- The <u>forward seismic convolutional model</u> is used to compute synthetic seismograms s(t) given the source wavelet w(t) and earth's reflectivity e(t) (equations 3.1 and 3.2). See example in <u>Figure 3.1</u>.

- The <u>deconvolution (inverse seismic convolutional model)</u> is used to:

  o Compute the earth's reflectivity e(t) given the seismic trace s(t) and the source wavelet w(t). This is the most common objective of deconvolution.

    ➢ If the source <u>wavelet is known</u>, the deconvolution becomes <u>deterministic</u>.

    ➢ If the source <u>wavelet is not known</u>, the deconvolution becomes <u>statistical</u>.

  o Compute the source wavelet w(t) given the seismic trace s(t) and the earth's reflectivity e(t). This is used if a seismic trace is recorded near a borehole.

- The seismic convolutional model is widely accepted because it agrees well with the observed seismic traces.

- The <u>seismic convolutional model commonly assumes the following</u>:

  (1) The earth is made up of horizontal layers of constant velocity.

  (2) The source generates only a P-wave, which is reflected on layer boundaries at normal incidence.

  (3) The source wavelet is stationary. That is, it does not change its shape as it travels in the subsurface.

(4) The noise component n(t) is zero.

(5) The Earth's reflectivity e(t) is a white random series of impulses.

(6) The source wavelet w(t) is a minimum-phase wavelet.

- Often, one or more of these assumptions might not be satisfied; in which case, advanced techniques of deconvolution have to be implemented.

Inverse filtering (spiking deconvolution)

- The aim of this process is to compress the source wavelet w(t) into a zero-phase spike of zero width (i.e., δ(t)). This means that we are eliminating the effect of the source wavelet and leaving only the effect of the Earth's reflectivity in the seismogram.

- We can achieve this by convolving the seismic trace by the <u>inverse filter</u>, f(t), of the source wavelet defined as:

$$w(t) * f(t) = \delta(t). \tag{3.3}$$

- Taking the FT of equation (3.3):

$$W(f) \, F(f) = 1, \tag{3.4}$$

where W(f), F(f), and 1 are the FTs of w(t), f(t), and δ(t), respectively.

- From equation (3.4):

$F(f) = 1/W(f) = [1/|A_w(f)|] \exp[-i \, \phi_w(f)],$

$\boxed{\therefore |A_f(f)| = 1/|A_w(f)|}$, and

$\boxed{\phi_f(f) = -\phi_w(f)}$,

where $|A_w(f)|$ and $\phi_w(f)$ are the amplitude and phase spectra of w(t) and $|A_f(f)|$ and $\phi_f(f)$ are the amplitude and phase spectra of f(t).

- Therefore, the amplitude spectrum of the inverse filter is the reciprocal of that of the source wavelet whereas its phase spectrum is the negative of that of the wavelet.

- Taking the IFT of the F(f), we get the desired inverse filter f(t).

- The <u>spiking deconvolution</u> is accomplished then by convolving the inverse filter f(t) with the seismic trace s(t):

$$f(t) * s(t) = f(t) * [w(t) * e(t)] = [f(t) * w(t)] * e(t) = \delta(t) * e(t) = e(t), \qquad (3.5a)$$

  which is the earth's response that we want to extract from the seismic trace.

- We can also use the z-transform (ZT) in equation (3.4), which results in

  $F(z) = 1 / W(z)$.

- F(z) is an <u>infinite polynomial</u> of z that is convergent only if w(t) is a minimum-phase wavelet.

- For practical reasons, the infinite polynomial F(z) has to be <u>truncated</u> to n terms (i.e., $F_n(z)$).

- Truncation generates less error if:

  o We include more terms of $F_n(z)$.

  o w(t) is a minimum-phase wavelet

- The truncated filter $f_n(t)$ is calculated by taking the inverse z-transform of $F_n(z)$.

- Because of truncation, convolution of the truncated filter $f_n(t)$ with the wavelet will *not* give the desired output $d(t) = \delta(t) = (1,0,0,\ldots)$.

- Instead, an <u>actual output</u> y(t) will be given as:

$$y(t) = f_n(t) * w(t) \qquad (3.5b)$$

- Apparently, $y(t) \neq d(t)$ and there will be a <u>truncation error</u> E defined as:

$$E = \Sigma_i(d_i - y_i)^2, \qquad (3.5c)$$

  where d_i and y_i are the ith samples of the desired and actual outputs, respectively.

- Example (1): Given a seismic trace $s(t) = (1/3, -17/30, 1/5)$ and a source wavelet $w(t) = (1, -1/2)$, use inverse filtering to calculate the deconvolved trace $s_{d2}(t)$ and compare it to the earth response $e(t) = (1/3, -2/5, 0, 0)$. *Note: $s_{d2}(t)$ indicates the deconvolved trace we get by using an inverse filter $f_2(t)$.*

Step (1): Take the z-transform of $w(t)$:

$$W(z) = 1 - (1/2)z.$$

Step (2): Use polynomial division to find $F(z)$:

$$F(z) = 1 / W(z) = 1 / [1 - (1/2)z] = 1 + (1/2)z + (1/4)z^2 + (1/8)z^3 + \ldots .$$

For a 2-sample wavelet (w_0, w_1), you can use the geometric power series as:

$$1 / [w_0(1 - a.z)] = (1/w_0)(1 + a.z + a^2.z^2 + a^3.z^3 + \ldots), \text{ where } a = w_1/w_0,$$

which converges only when $|a| < 1$ (i.e., $W(t)$ is minimum phase.).

Step (3): For $f_2(t)$, we truncate at the second term $(n = 2)$, giving:

$$F_2(z) = 1 + (1/2)z.$$

Step (4): Find the truncated inverse filter by taking the inverse z-transform of $F_2(z)$, which results in:

$$f_2(t) = (1, \tfrac{1}{2}).$$

Step (5): Deconvolve $s(t)$ by convolving it with $f_2(t)$ to get:

$$s_{d2}(t) = f_2(t) * s(t) = (1, \tfrac{1}{2}) * (1/3, -17/30, 1/5) = (1/3, -2/5, -1/12, 1/10).$$

Step (6): The error between $s_{d2}(t)$ and $e(t)$ can be calculated as:

$$L = \Sigma_i(s_{d2i} - e_i)^2 = (1/3-1/3)^2 + (-2/5+2/5)^2 + (-1/12-0)^2 + (1/10-0)^2 = 0.01694.$$

Step (7): The actual output is:

$$y(t) = f_2(t) * w(t) = (1, \tfrac{1}{2}) * (1, -1/2) = (1, 0, -\tfrac{1}{4}).$$

Step (8): Knowing that $d(t) = \delta(t) = (1, 0, 0)$, we calculate E as:

$$E = (1-1)^2 + (0-0)^2 + (0+1/4)^2 = 1/16 = 0.0625.$$

- Figure 3.2 shows the effect of wavelet phase on deconvolution.

Exercise (1):

a. Calculate $s_{d3}(t)$ for the above example and compare the resulting errors to those you got using $s_{d2}(t)$.

b. Carry out the steps of the above example using the FT instead of the ZT.

c. If we have random noise in the seismic trace, where will you include it in the above example?

Optimum Wiener filters

- Wiener optimum filtering involves designing a filter f(t) so that the error E between the desired output d(t) and the actual output y(t) is minimum:

$$E = \sum_i (d_i - y_i)^2, \qquad (3.6)$$

where d_i and y_i are the ith samples of the desired and actual outputs, respectively.

- Remember that the actual output y(t) was given as:

$$y(t) = f(t) * x(t), \qquad (3.7)$$

where x(t) is the input.

- Substituting equation (3.7) into (3.6):

$$E = \sum_i (d_i - \{f(t) * x(t)\}_i)^2. \qquad (3.8)$$

where $\{f(t)*x(t)\}_i = y_i$ is the ith sample of y(t).

- The goal is to compute the filter coefficients (f_0, f_1, ..., f_{n-1}) so that the error E is minimum; where the filter length, n, must be predefined.

- This is a typical least-squares problem, and the minimum error is attained by setting the partial derivative of E with respect to f_j to zero:

$$\partial E/\partial f_j = 0, \qquad j = 0, 1, 2, ..., (n-1). \qquad (3.9)$$

- Applying equation (3.9) on (3.8) and simplifying, the result can be expressed as:

$$\sum_{i=0}^{n-1} r_{j-i} f_i = g_j, \qquad j = 0, 1, 2, ..., (n-1). \quad (3.10a)$$

- In matrix format, equation (3.10a) can be written as:

$$\begin{bmatrix} r_0 & r_1 & r_2 & . & . & r_{n-1} \\ r_1 & r_0 & r_1 & . & . & r_{n-2} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ r_{n-2} & r_{n-3} & r_{n-4} & . & . & r_1 \\ r_{n-1} & r_{n-2} & r_{n-3} & . & . & r_0 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ . \\ . \\ f_{n-2} \\ f_{n-1} \end{bmatrix} = \begin{bmatrix} g_0 \\ g_1 \\ . \\ . \\ g_{n-2} \\ g_{n-1} \end{bmatrix}. \qquad (3.10b)$$

- In equations (3.10a) and (3.10b), g_j is the jth term of the crosscorrelation between

   d(t) and x(t):

$$g_j = \{d(t) \otimes x(t)\}_j, \qquad j = 0, 1, 2, ..., (n-1).$$

$$(3.11a)$$

   (*Note that g(t) = d(t) $\otimes$ x(t) $\neq$ x(t) $\otimes$ d(t) and that we* <u>*only*</u> *need lags 0, 1, ...(n-1)*).

- In equations (3.10a) and (3.10b), r_{j-i} is the jth term of the autocorrelation of x(t):

$$r_j = \{x(t) \otimes x(t)\}_j, \qquad j = 0, 1, 2, ..., (n-1). \qquad (3.12)$$

   (*Note that r(t) = x(t) $\otimes$ x(t) = r(-t) and we* <u>*only*</u> *need lags 0, 1, ...(n-1)*).

- If g(t) and r(t) are known, equation (3.10) can be solved uniquely to find the filter

   f(t).

- Equations (3.10a) and (3.10b) are known as <u>the normal equations</u>.

- The autocorrelation (r_j) matrix of equation (3.10b) is a Toeplitz matrix that can be

   inverted efficiently using the Levinson recursion procedure.

- See Figure 3.3 for the derivation of the normal equations.

Inverse filtering using the normal equations

- As an application of the normal equations, we will use them to design the inverse filter of the source wavelet such that the desired output is a zero-lag spike.

- $x(t) = w(t) = (w_0, w_1, ..., w_{n-1}).$

- $f(t) = (f_0, f_1, ..., f_{n-1}).$

  both of length = n.

- $d(t) = \delta(t) = (1, 0, ..., 0),$ of length $2n-1$.

- $r_0 = w_0{}^2 + w_1{}^2 + ... + w_{n-1}{}^2$

  $r_1 = w_0\,w_1 + w_1\,w_2 + ... + w_{n-2}\,w_{n-1}$

  ... ...

  $r_{n-1} = w_0\,w_{n-1}$

- $g_0 = w_0$

  $g_1 = 0$

  ... ...

  $g_{n-1} = 0.$

- Therefore, the normal equations become:

$$
\begin{bmatrix}
r_0 & r_1 & r_2 & . & . & r_{n-1} \\
r_1 & r_0 & r_1 & . & . & r_{n-2} \\
. & . & . & . & . & . \\
. & . & . & . & . & . \\
r_{n-2} & r_{n-3} & r_{n-4} & . & . & r_1 \\
r_{n-1} & r_{n-2} & r_{n-3} & . & . & r_0
\end{bmatrix}
\begin{bmatrix}
f_0 \\
f_1 \\
. \\
. \\
f_{n-2} \\
f_{n-1}
\end{bmatrix}
=
\begin{bmatrix}
w_0 \\
0 \\
. \\
. \\
0 \\
0
\end{bmatrix}.
$$

(3.13)

- <u>Example (2)</u>: Use the normal equations to find the inverse filter of $w(t) = (1,-1/2)$.

  o $d(t) = \delta(t) = (1,0,0).$

  o $r_w(t) = (5/4,-1/2).$

  o $g(t) = (1,0).$

  o Therefore the normal equations become: $\begin{bmatrix} 5/4 & -1/2 \\ -1/2 & 5/4 \end{bmatrix}\begin{bmatrix} f_0 \\ f_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$

- o Solution yields the filter: $f(t) = (20/21, 8/21)$.

- o $y(t) = f(t) * w(t) = (20/21, -2/21, -4/21)$.

- o $E = (1-20/21)^2 + (0+2/21)^2 + (0+4/21)^2 = 21/441 = 1/21 = 0.048$.

- o Comparison of this error to the error we got using inverse filtering ($E = 0.0625$), we see that optimum Weiner filtering performs better than inverse filtering.

- Exercise (2): deconvolve $s(t) = (1/3, -17/30, 1/5)$ using $f(t) = (20/21, 8/21)$ and compare the error in $e(t)$ for the cases of inverse filtering versus optimum Weiner filtering.

The trace-wavelet relation

- We just saw that we need to know the amplitude spectrum or autocorrelation of the source wavelet in order to perform spiking deconvolution or optimum filtering.

- However, for most impulsive sources, the source wavelet is not known and deconvolution or filtering cannot be performed directly.

- We need to either estimate the source wavelet or find relations between the amplitude spectra or autocorrelations of the seismic trace and source wavelet. Here we investigate the latter case.

- Amplitude spectra:

  - o The earth response $e(t)$ can safely be approximated as a <u>white random series</u> of impulses (Robinson and Treitel, 2000; Yilmaz, 2001).

  - o If this is the case, the amplitude spectrum of $e(t)$ will be constant. That is:

$$|A_e(f)| \Leftrightarrow E_0. \tag{3.14}$$

- o Using equation (3.1) to find the trace amplitude spectrum and substituting equation (3.14) yields:

$$|A_s(f)| = |A_w(f)| \, |A_e(f)| = E_0 \, |A_w(f)|. \tag{3.15}$$

- o Equation (3.15) means that <u>the amplitude spectrum of the seismic trace is a scaled version of the amplitude spectrum of the source wavelet.</u>

- o <u>More internet resources about colored random series</u>.

- Autocorrelations:

  - o Because of the random nature of $e(t)$, its autocorrelation $r_e(t)$ is generally zero anywhere except at $t = 0$, where it is equal to the energy in $e(t)$ given by:

$$r_{e0} = \sum_i e_i^2 = e_0^2 + e_1^2 + \cdots. \tag{3.16a}$$

  - o Therefore, $r_e(t)$ can be approximated as:

$$r_e(t) = r_{e0} \, \delta(t). \tag{3.16b}$$

  - o Using this fact about $r_e(t)$, we can derive the relation between the trace and wavelet autocorrelations ($r_s(t)$ and $r_w(t)$) using equations (2.14) and (3.16b) as follows:

$r_s(t) = s(t) \otimes s(t)$

$\quad = s(t) * s(-t) \hfill$ (eq. 2.14)

$\quad = [w(t) * e(t)] * [w(-t) * e(-t)] \hfill$ (eq. 3.1)

$\quad = [w(t) * w(-t)] * [e(t) * e(-t)]$

$\quad = [w(t) \otimes w(t)] * [e(t) \otimes e(t)] \hfill$ (eq. 2.14)

$\quad = r_w(t) * r_e(t)$

$\quad = r_{e0} \, \delta(t) * r_w(t) \hfill$ (eq. 3.16b)

$\quad = r_{e0} \, r_w(t). \hfill$ (3.17)

    o   Equation (3.17) means that <u>the autocorrelation of the seismic trace is a scaled version of the autocorrelation of the source wavelet</u>.

- The benefit of these relations is that we can use $|A_s(f)|$ or $r_s(t)$ whenever $|A_w(f)|$ or $r_w(t)$ are needed in deconvolution or filtering.

- Provided the randomness and whiteness assumptions about $e(t)$ are satisfied, all we miss by using $|A_s(f)|$ or $r_s(t)$ instead of $|A_w(f)|$ or $r_w(t)$ is a scaling factor.

Predictive deconvolution

- The goal of predictive deconvolution is to generalize the normal equations to any type of deconvolution (spiking or multiple suppression).

- For the autocorrelation coefficients, r_j, of the source wavelet on the left-hand side of the normal equations (3.10), we can use the corresponding autocorrelation coefficients of the trace (i.e., the trace-wavelet relation).

- But we still need the source wavelet in constructing the right-hand side of the normal equations (i.e., g_j in equation (3.10)); otherwise, we cannot use the normal equations.

- One way to work around this problem is to replace the crosscorrelations g_j on the right-hand side by the autocorrelations of the source wavelet r_j.

- We can achieve this replacement if we require the desired output $d(t)$ to be a time-advanced (left-shifted) version $x(t+\alpha)$ of the input $x(t)$, where α is a parameter that we select.

- So basically we will be solving the normal equations for the filter $p(t)$ that will transform the input signal to the same signal but only advanced α samples.

- This is the same as trying to *predict* the $(n-1+\alpha)$ time sample of the input signal. Therefore, we call <u>$p(t)$ the prediction filter</u> and <u>α the prediction lag</u>.

- For example, if $x(t) = (x_0, x_1, x_2, x_3, x_4)$ and $\alpha = 1$, then: $d(t) = x(t+1) = (x_1, x_2, x_3, x_4, x_5)$.

- Since $d(t) = x(t+\alpha)$, then $g(t) = d(t) \otimes x(t) = x(t+\alpha) \otimes x(t) = r(t+\alpha)$. Therefore, $g_0 = r_\alpha$, $g_1 = r_{\alpha+1}$, ..., $g_{n-1} = r_{\alpha+n-1}$.

- Inserting these values for $g(t)$ into the normal equations (3.10b), we get:

$$\begin{bmatrix} r_0 & r_1 & r_2 & . & . & r_{n-1} \\ r_1 & r_0 & r_1 & . & . & r_{n-2} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ r_{n-2} & r_{n-3} & r_{n-4} & . & . & r_1 \\ r_{n-1} & r_{n-2} & r_{n-3} & . & . & r_0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ . \\ . \\ p_{n-2} \\ p_{n-1} \end{bmatrix} = \begin{bmatrix} r_\alpha \\ r_{\alpha+1} \\ . \\ . \\ r_{\alpha+n-2} \\ r_{\alpha+n-1} \end{bmatrix}. \qquad (3.18)$$

- When $p(t)$ is convolved with the input $x(t)$, it will give us an actual output $y(t)$ that is the best least-squares fit to the desired output $d(t) = x(t+\alpha)$.

- In the case of seismic trace deconvolution, $x(t)$ is the seismic trace $s(t)$, which includes a predictable part (e.g., multiples) and a non-predictable (random) part which consists of the primaries: $s(t) = m(t) + e(t)$.

- Therefore, $y(t)$ will only contain an estimate $\hat{m}(t + \alpha)$ of the shifted predictable part $m(t+\alpha)$ of the desired output $d(t) = s(t+\alpha) = m(t+\alpha) + e(t+\alpha)$, where $e(t+\alpha)$ represents the random part of $s(t+\alpha)$ generated by shifting the earth response $e(t)$.

- Therefore, $e(t+\alpha)$ can be found as:

$$e(t+\alpha) = d(t) - y(t) = s(t+\alpha) - p(t)*s(t). \qquad (3.19)$$

- Equation (3.19) can be manipulated to yield the earth response $e(t)$:

$$e(t) = s(t)*[\delta(t) - p(t-\alpha)] = s(t)*a(t), \qquad (3.20a)$$

$$\text{where } a(t) = \delta(t) - p(t-\alpha). \qquad (3.20b)$$

- Equation (3.20) states that the earth response $e(t)$ can be retrieved by convolving the filter $a(t)$ with the seismic trace $s(t)$.

- The filter a(t) is called the <u>prediction error filter</u> and is given by:

$$
\overbrace{\phantom{(\alpha-1)\ \text{zeros}}}^{(\alpha-1)\ \text{zeros}}
$$

$$
a(t) = (1,\ \underbrace{0,\ 0,\ ...,\ 0,}\ -p_0,\ -p_1,\ ...,\ -p_{n-1}). \tag{3.21}
$$

$$
\underbrace{}_{n\ \text{filter coefficients}}
$$

- For <u>α = 1</u>, <u>a(t) = (1, -p_0, ..., -p_{n-1})</u> and the normal equations can be manipulated to have the same structure as equation (3.13) except for a scaling factor (k):

$$
\begin{bmatrix}
r_0 & r_1 & r_2 & . & . & r_n \\
r_1 & r_0 & r_1 & . & . & r_{n-1} \\
. & . & . & . & . & . \\
. & . & . & . & . & . \\
r_{n-1} & r_{n-2} & r_{n-3} & . & . & r_1 \\
r_n & r_{n-1} & r_{n-2} & . & . & r_0
\end{bmatrix}
\begin{bmatrix}
1 \\ -p_0 \\ . \\ . \\ . \\ -p_{n-1}
\end{bmatrix}
=
\begin{bmatrix}
k \\ 0 \\ . \\ . \\ 0 \\ 0
\end{bmatrix}. \tag{3.22}
$$

- Hence, <u>the prediction error filter a(t) = (1, -p_0, ..., -p_{n-1}) is the inverse filter of the source wavelet of our input seismic trace s(t) except for a scaling factor</u>.

- Here are the steps for using predictive deconvolution to perform spiking deconvolution to a seismic trace:

1. Select n and α = 1.

2. Autocorrelate the trace.

3. Construct the predictive normal equations (3.18) using the trace autocorrelations.

4. Solve equation (3.18) to find the prediction filter p(t) = (p_0, ..., p_{n-1}).

5. Construct the prediction error filter a(t) = (1, -p_0, ..., -p_{n-1}).

6. Convolve a(t) with the trace s(t) to get the earth response e(t).

- <u>Prediction error filters with prediction lags of α > 1 are used for multiple suppression.</u>

- Because of the general nature of the predictive deconvolution, we normally use it for most of our deconvolution needs.

- Example (3): Given the seismic trace $s(t) = (1/3, -17/30, 1/5)$. Use predictive deconvolution to find the prediction error filter $a(t)$ needed to spike $s(t)$ into $s_{d2}(t)$ and compare $s_{d2}(t)$ with the earth response $e(t) = (1/3, -2/5)$.

  1. Set $n = 2$ and $\alpha=1$ (for spiking deconvolution).

  2. $r_s(t) = (17/36, -68/225, 1/15)$.

  3. The normal equations become: $\begin{bmatrix} 17/36 & -68/225 \\ -68/225 & 17/36 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \end{bmatrix} = \begin{bmatrix} -68/225 \\ 1/15 \end{bmatrix}$.

  4. Solution yields the filter: $p(t) = (-0.93, -0.45)$.

  5. $a(t) = (1, 0.93, 0.45)$.

  6. $s_{d2}(t) = a(t) * s(t) = (0.33, -0.26, -0.18, -0.07, 0.09)$.

  7. $E = \Sigma_i(e_i - s_{d2i})^2 = 0.065$, which is larger than that we got using previous methods due to the fact that $e(t)$ might not be a white random series.

- Exercise (3): Starting with the known wavelet $w(t) = (1, -1/2)$, use predictive deconvolution to find the prediction error filter $a(t)$ needed to deconvolve $s(t)$ into $s_{d2}(t)$ and compare $s_{d2}(t)$ with $e(t)$. How does the error between $s_{d2}(t)$ and $e(t)$ compare in this case to the case when the wavelet is unknown?

Predictive deconvolution in practice

Parameter selection criteria

- To perform predictive deconvolution in practice, we need to set the following parameters:

> ➢ Autocorrelation window (w): this sets up the part of seismic trace from which we will select the elements of the autocorrelation matrix in the normal equations.

> ➢ Operator length (n): this sets up the length of the prediction filter p(t).

> ➢ Prediction lag (α): this sets up the type of deconvolution: spiking or multiple suppression.

> ➢ Prewhitening (ε): this sets up the amount of white random noise we want to include into our autocorrelation matrix to stabilize the solution of the normal equations.

Autocorrelation window (w)

- The choice of deconvolution parameters depends largely on the characteristics of the autocorrelation of the seismic trace.

- Therefore, it is important to choose a suitable autocorrelation window (gate) that will be used to calculate the deconvolution parameters.

- The autocorrelation window should include the part of the record that contains useful reflection signal and should exclude coherent (e. g., ground roll) or incoherent noise (e. g., later parts of the record).

- The length of the autocorrelation window should be as large as possible and must be greater than eight times the largest operator length that will be used for that dataset.

Operator length (n)

- The operator (filter) length should be equal to the wavelet length.

- The first transient zone of the trace autocorrelation is the part that mostly represents the autocorrelation of the source wavelet.

- The first transient zone is the first part of the autocorrelation that contains high amplitudes.

- <u>The operator length should be as large as possible but not less than the length of the first transient zone of the trace autocorrelation</u>.

- The optimum operator length should not leave considerable amount of energy in the trace autocorrelation.

Prediction lag (α)

- <u>For spiking deconvolution, $\alpha = 1$ should be selected</u>.

- If prediction lag is increased, the output from predictive deconvolution becomes less spiky.

- The application of spiking deconvolution to some field data might boost high-frequency noise in the data. In this case, we can use α slightly greater than 1.

- For <u>suppressing multiples</u>, the selection of α depends on the type of multiple:

  - *Long-path multiples:* <u>α should be selected equal to the beginning of first multiple on the trace autocorrelation</u> ([Figure 3.4A](#)).

  - *Short-path multiples:* <u>α should be selected equal to the first or second zero crossing on the trace autocorrelation</u> ([Figure 3.4B](#)).

- In case of multiples, we deconvolve the trace in the following sequence:

  1. Multiple suppression deconvolution.
  2. Spiking deconvolution.

Prewhitening (ε)

- A numerical instability might be encountered when inverting the trace autocorrelation matrix in the normal equations if the determinant of the autocorrelation matrix is zero.

- To ensure numerical stability, we introduce an artificial level of white random noise into the trace autocorrelation before deconvolution. This process is called <u>prewhitening</u>.

- Prewhitening is achieved by adding a white random noise, with a very small variance (ε), to the trace amplitudes at every time sample.

- This is equivalent to adding a very small positive constant to the zero-lag autocorrelation (r_0) of the trace (Prove!).

- The magnitude of prewhitening is measured as a percentage of the zero-lag autocorrelation value r_0.

- In practice, <u>typically 0.1% to 0.3% (i.e., 0.001r_0 to 0.003r_0) prewhitening is standard</u> in processing.

- See Figure 3.5 for a summary of selecting deconvolution parameters.

Practical aspects of real seismic data

Effect of random noise on deconvolution

- Field data practically contain a noise component. If this component is small, then we can ignore its effect.

- Alternatively, we can delay the deconvolution until we stack the data, so that the S/N ratio is improved.

- However, if the noise is considerable, then we must test to see if deconvolution can enhance the resolution of the data without destroying its interpretive content.

- Exercise (4): Generate Gaussian random noise with a zero mean and 0.1 standard deviation and add it to s(t) in Exercise (3) and carry out the predictive deconvolution. How does the error between $s_{d2}(t)$ and e(t) compare in this case to

the case in Exercise (3)? What happens if you increase the noise standard deviation?

Nonstationarity and deconvolution

- Because the Earth filters high frequencies, the shape of the source wavelet changes as it travels in the earth.

- If the wavelet changes its shape considerably with depth, a time-variant deconvolution (TVD) should be used.

- Time variant deconvolution consists of segmenting the seismic record into 3 - 4 parts and each part is used to construct its own deconvolution parameters.

- To see if TVD is required, we look at the amplitude spectrum of few adjacent traces using a sliding time window along the traces. If the peak frequency becomes considerably lower with increasing TWTT time, we need TVD.

- See Figure 3.6.

Multiple suppression by predictive deconvolution

- As an example, we will see how predictive deconvolution is used to suppress the water multiple (Figure 3.7).

- The time series representing the water-multiple is:

$$m(t) = (1, 0, ..., 0, -R, 0, ..., 0, R^2, 0, ..., 0, -R^3, ...), \qquad (3.23)$$

which is an infinite series, but with decreasing magnitude because $R < 1$ and the separation between the nonzero elements is c-1 zeros.

- For simplicity, we will use the Z-transform approach to estimate the α parameter for this m(t) series.

- The Z-transform of m(t) is:

$$M(z) = 1 - R.z^c + (R.z^c)^2 - (R.z^c)^3 + ... = \sum_{i=0}^{\infty} (-R.z^c)^i = 1/(1+R.z^c), \qquad (3.24)$$

where the convergence is due to the minimum-phase of the m(t) series.

- The inverse filter of m(t) is f(t) such that:

$$f(t) * m(t) = \delta(t). \tag{3.25}$$

- The Z-transform of equation (3.25) is:

$$F(z) \, M(z) = 1. \tag{3.26}$$

- Therefore, $F(z) = 1/M(z) = (1 + R.z^c)$.

- Taking the inverse Z-transform of F(z), we get f(t) = (1, 0, ..., 0, R), where the separation between 1 and R is c-1 zeros.

- This means that the α parameter for this m(t) series is equal to c samples = $c.\Delta t$ (sec).

- Therefore, the predictive deconvolution parameters in this case are:

  1. <u>For multiple suppression</u>: The optimum parameters are: w = trace length, ε = 0.1% = 0.001, n = T_{z1} and α = c, where T_{z1} is the length of the first transient zone of the trace autocorrelation.

  2. <u>For spiking deconvolution</u>: The optimum parameters are: w = trace length, ε = 0.1% = 0.001, n = T_{z1} and α = 1.

- Exercise (5): Use the normal equations to estimate the inverse filter of m(t). What is the α parameter you estimate in this case? Find the error between the desired and actual outputs up to the first 4 terms and compare to that found from the Z-transform.