# Spectral Balancing

by Jon F. Claerbout

A collection of seismograms may be said to be spectrally balanced if they have been filtered so that they all have about the same spectrum. Spectral balancing is often useful when timing relationships among the traces are important. For example, spectral balancing usually preceeds velocity estimation. The spectral balancing is often done by deconvolution (whitening) followed by bandpass filtering. This results in the balanced traces having the spectrum of the bandpass filter. In the balancing algorithm to be described here, the final spectrum is not predetermined but is the geometric mean of the spectra of the input traces. One advantage of this balancing is that there is no danger of pulling up "noisy holes" in the spectrum. The geometric mean is the exponential of the arithmetic mean of the logarithms of the spectra. Since spectra are positive, the geometric mean is a natural average. It does cost more to compute so we should have some reason for selecting it. When a collection of spectra are already very similar to one another, there is very little difference between the two averages. For a common depth point gather, i.e., seismograms at various shot to geophone offsets, the spectra are often quite different. This gives us a reason to think about the comparison of the geometric mean to the arithmetic mean. Actually we are not planning to use CDP gathers, but slant stacks for various seismic ray parameters $p$ . With a slant stack of some $p$ there is a shot antenna response, a geophone antenna response, and a mud layer resonance, all of which are constant over travel time. For each $p$ there is a different collection of responses. It is the geometric average response which we feel is appropriate to use before velocity analysis. The geometric average discounts extremely large values which can occur at certain natural resonance frequencies and angles in stratified sediments. The geometric mean will be zero if any element is zero. Thus, if a frequency component is completely missing from one trace, it will be removed from all the traces.

Let $X_k(Z)$ denote the $k^{th}$ original seismogram and $Y_k(Z)$ denote the balanced seismogram. The most straightforward calculation is

$$Y_k(Z) = X_k(Z) \frac{\frac{1}{N}\left[ \sum_{k=1}^{N} \bar{X}_k(Z^{-1})\, X_k(Z) \right]^{1/2}}{\left[ \bar{X}_k(Z^{-1})\, X_k(Z) \right]^{1/2}} \tag{1}$$

The use of the geometric mean converts this to

$$Y_k(Z) = X_k(Z) \frac{\left[ \prod_{k=1}^{N} |X_k(Z)| \right]^{1/N}}{|X_k(Z)|} \tag{2}$$

These are readily accomplished with fast Fourier transforms. In the time domain, this amounts to convolving each trace by a symmetrical filter. Commonly we prefer to use causal filters. Causal filters give the correct result with synthetic seismograms. Thus, (2) may be modified to use the $k^{th}$ prediction error filter $A_k(Z)$, and the prediction effor filter, $A_{ave}(Z)$ of the geometric mean spectrum giving

$$Y_k(Z) = X_k(Z) \frac{A_k(Z)}{A_{ave}(Z)} \tag{3}$$

Although Fourier transformation could be used to compute the deconvolution filter, $A_k(Z)$ , it is commonly done by the Levinson solution to Toeplitz equations. Perhaps the reason for preferring the Toeplitz approach is that it is easy to restrain the filters to be short. Although the spectra are then not completely balanced, it is probably preferable that the original data should be modified only with short filters. The question is now whether we can carry these advantages over to the geometric mean balancing filter, $A_{ave}(Z)$ . Obviously, we could compute the geometric mean spectra with Fourier transforms and then return to the time domain Toeplitz formulation to get $A_{ave}(Z)$ . Actually, the whole calculation can be quickly done in the time domain. First, note that the average of the logarithms of some spectra may be computed by the average of the logarithms of the prediction effor filters, since

$$\Sigma \ \ell n \ \bar{X} X = \Sigma \ \ell n \frac{1}{\bar{A} A} = - \Sigma \ \ell n \ \bar{A} - \Sigma \ \ell n \ A \tag{4}$$

The logarithm of a prediction error filter (actually any filter, with convergence on the unit circle only for minimum phase filters) may be computed from the following identities

$$U(Z) = \ell n \ B(Z) \tag{5}$$

$$\frac{dU}{dZ} = \frac{1}{B(Z)} \frac{dB}{dZ}$$

$$\frac{dB}{dZ} = \frac{dU}{dZ} B(Z)$$

$$b_1 + 2b_2 + 3b_3 Z^2 + \cdots = ( u_1 + 2u_2 Z + 3u_3 Z^2 + \cdots )( b_0 + b_1 Z + b_2 Z^2 + \cdots ) \quad (6)$$

Identifying coefficients of successive powers of $Z$ we obtain

$$\begin{aligned} b_1 &= u_1 \; b_0 \qquad (7) \\ 2\, b_2 &= u_1 \; b_1 + 2\, u_2 \, b_0 \\ 3\, b_3 &= u_1 \; b_2 + 2\, u_2 \, b_1 + 3\, u_3 \, b_0 \end{aligned}$$

Along with the initial condition at $Z = 0$ , namely

$$\begin{aligned} \ell n \; b_0 &= u_0 \\ b_0 &= e^{u_0} \end{aligned} \qquad (8)$$

the relations (7) enable the recursive calculation of logarithms or exponentials of polynomials. For exponentiation we have

$$b_k = \frac{1}{k} \sum_{i=1}^{k} i \, u_i \, b_{k-i} \qquad (9)$$

For logarithms we have

$$u_k = \frac{1}{b_0} \left( b_k - \frac{1}{k} \sum_{i=1}^{k-1} i \, u_i \, b_{k-i} \right) \qquad (10)$$

Programs and examples are found in Figures 1 through 6.

It may be noted that it is entirely optional whether to use a numerator or a denominator representation for the balanced filter. There is no proof, however, that truncated denominators will be stable.

```
      SUBROUTINE AUTO(N,X,LAGS,R)
C     COMPUTE AUTOCORRELATION R OF SEISMOGRAM X
      DIMENSION X(N),R(LAGS)
      DO 20 LAG=1,LAGS
      R(LAG)=0.
      NSUM=N-LAG+1
      DO 10 ISUM=1,NSUM
   10 R(LAG)=R(LAG)+X(ISUM)*X(ISUM+LAG-1)
   20 R(LAG)=R(LAG)/N
      RETURN
      END

      SUBROUTINE LEV(N,R,A,SPACE)
C     FIND PREDICTION ERROR WAVELET A FROM AUTOCOR R
      DIMENSION R(N),A(N),SPACE(N)
      V=R(1)
      A(1)=1.
      DO 30 J=2,N
      A(J)=0.
      E=0.
      DO 10 I=2,J
   10 E=E+R(I)*A(J-I+1)
      C=E/V
      V=V-E*C
      DO 20 I=1,J
   20 SPACE(I)=A(I)-C*A(J-I+1)
      DO 30 I=1,J
   30 A(I)=SPACE(I)
      V=SQRT(V)
      DO 40 I=1,N
   40 A(I)=A(I)/V
      RETURN
      END

      SUBROUTINE PMULT(NF,F,N,X)
C     POLYNOMIAL MULTIPLICATION X=X*F
      DIMENSION X(N),F(NF)
      DO 20 KR=1,N
      K=N-KR+1
      SUM=0.
      J=1+MAX0(0,K-NF)
      DO 10 I=J,K
   10 SUM=SUM+F(K-I+1)*X(I)
   20 X(K)=SUM
      RETURN
      END
```

Fig. 1. Some subroutines required for time domain spectral balancing. Autocorrelation, Levinson recursion and polynomial multiplication (filtering).

```
      SUBROUTINE PDIV(L,DIV,N,X)
C     POLYNOMIAL DIVISION X=X/DIV
      DIMENSION DIV(L),X(N)
      X(1)=X(1)/DIV(1)
      DO 20 J=2,N
      LL=MINO(L,J)
      SUM=0.
      DO 10 I=2,LL
 10   SUM=SUM+DIV(I)*X(J-I+1)
 20   X(J)=(X(J)-SUM)/DIV(1)
      RETURN
      END
```

```
      SUBROUTINE PEXP(N,U,B)
C     POLYNOMIAL EXPONENTIATION B=EXP(U)
      DIMENSION B(N),U(N)
      B(1)=EXP(U(1))
      DO 20 K=2,N
      SUM=0.
      DO 10 I=2,K
 10   SUM=SUM+(I-1)*U(I)*B(K-I+1)
 20   B(K)=SUM/(K-1.)
      RETURN
      END
```

```
      SUBROUTINE PLOG(N,B,U)
C     POLYNOMIAL LOGARITHM U=LN(B)
      DIMENSION B(N),U(N)
      U(1)=ALOG(B(1))
      U(2)=B(2)/B(1)
      DO 20 K=3,N
      SUM=0.
      KM=K-1
      DO 10 I=2,KM
 10   SUM=SUM+(I-1)*U(I)*B(K-I+1)
 20   U(K)=(B(K)-SUM/(K-1.))/B(1)
      RETURN
      END
```

Fig. 2. More subroutines required for time domain spectral balancing. Polynomial division, exponentiation and logarithm.

```
C     TEST POLYNOMIAL SUBROUTINES
      DIMENSION SPACE(10),R(5),X(5),A(5),B(5),U(5),ONE(5)
      DATA X,ONE/1.,2.,0.,0.,0.,1.,0.,0.,0.,0./
      N=5
      PRINT 10, X
 10   FORMAT (5F13.5)
      CALL AUTO(N,X,N,R)
      PRINT 10, R
      CALL LEV(N,R,A,SPACE)
      PRINT 10,A
      CALL PDIV(N,A,N,ONE)
      PRINT 10,ONE
      CALL PMULT(N,A,N,ONE)
      PRINT 10,ONE
      CALL PLOG(N,A,U)
      PRINT 10,U
      CALL PEXP(N,U,A)
      PRINT 10,A
      STOP
      END
```

```
COMPILE TIME =      0.39 SECONDS, OBJECT CODE=     4,672 BYTES, ..
      1.00000      2.00000      0.00000      0.00000      0.00000
      1.00000      0.40000      0.00000      0.00000      0.00000
      1.11762     -0.55717      0.27531     -0.13110      0.05244
      0.89476      0.44607      0.00197     -0.00394      0.00789
      1.00000     -0.00000     -0.00000      0.00000      0.00000
      0.11121     -0.49853      0.12207     -0.03580      0.00388
      1.11762     -0.55717      0.27531     -0.13110      0.05244
STOP IN LINE    23.
EXECUTION TIME =      0.07 SECONDS
```

Fig. 3. Check-out of subroutines required for spectral balancing.

```
      SUBROUTINE BALANS(NT,NX,X,L)
C     FILTER SEISMOGRAMS X TO BALANCE THEIR SPECTRA
      DIMENSION X(NT,NX),A(9),R(9),S(9),U(9)
      DO 10 I=1,L
10    S(I)=0.
      DO 20 IX=1,NX
      CALL AUTO(NT,X(1,IX),L,R)
      CALL LEV(L,R,A,U)
      CALL PMULT(L,A,NT,X(1,IX))
      CALL PLOG(L,A,U)
      DO 20 I=1,L
20    S(I)=S(I)+U(I)/NX
      CALL PEXP(L,S,A)
      DO 30 IX=1,NX
30    CALL PDIV(L,A,NT,X(1,IX))
      RETURN
      END
```

Fig. 4. Subroutine for balancing spectra of NX seismograms X of NT points each with filters of L lags. Sample results in Figure 5 for L=5 .

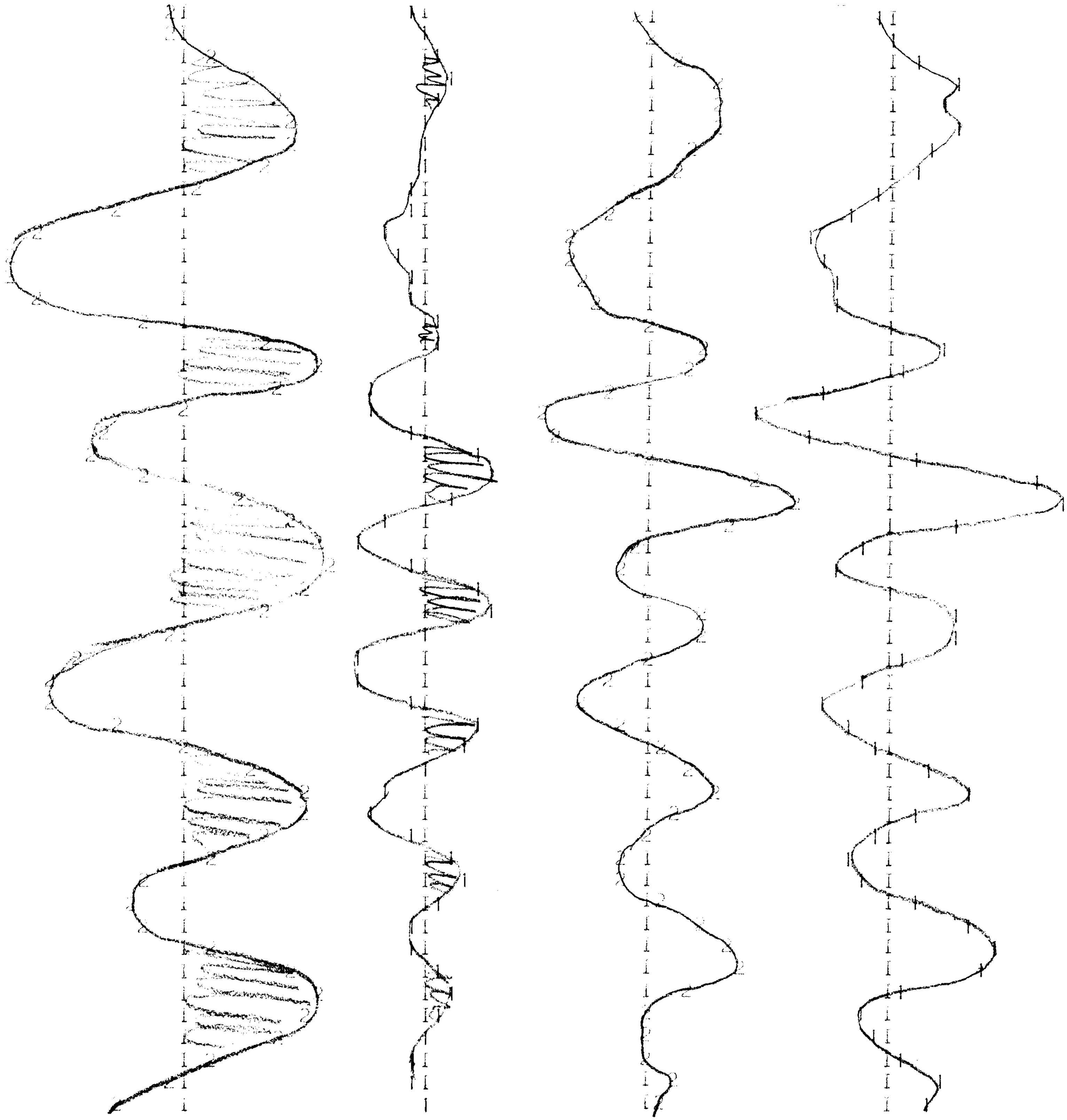

Fig. 5. Two traces (left) were generated with random noise into two different filters. Observe amplitude and frequency imbalance. Output traces (right) were balanced by the program of Figure 4 to the geometric mean spectra.

| | | |
|---|---|---|
| 0.97405 | 1.02669 | 1 |
| 0.00000 | 0.00000 | 2 |
| -0.38960 | -0.61801 | 3 |
| -0.64934 | -0.34223 | 4 |
| -0.05844 | -0.18480 | 5 |
| 0.25973 | 0.20534 | 6 |
| -0.21645 | -0.11088 | 7 |
| 0.03896 | 0.06160 | 8 |
| 0.11288 | -0.08316 | 9 |
| -0.14401 | 0.11332 | 10 |
| 0.04490 | -0.08279 | 11 |
| 0.03559 | -0.02282 | 12 |
| 0.03367 | 0.02905 | 13 |
| -0.02499 | 0.00178 | 14 |
| -0.02772 | 0.02149 | 15 |
| 0.00587 | -0.01817 | 16 |
| -0.00071 | 0.00390 | 17 |
| 0.00753 | -0.00155 | 18 |
| -0.00575 | 0.0009[illegible] | 19 |
| 0.00005 | -0.00[illegible] | 20 |

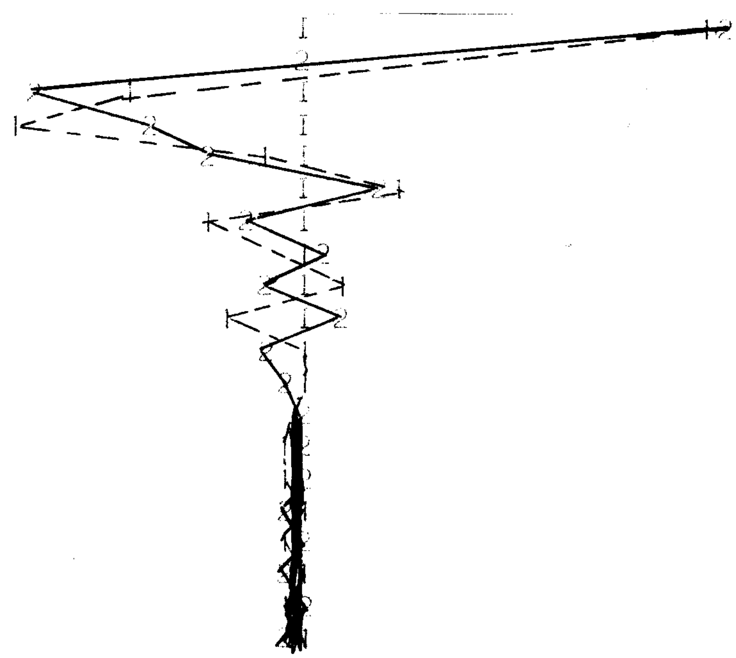

Fig. 6. Spectral balance of the ghosting filters $1 - Z^3$ and $1 - Z^2$ . These filters have multiple zeros on the unit circle. Balancing was done with 9 term filters. Plotted above are the balanced traces for an input delta function.

An application of spectral balancing in the conventional processing environment (no slant stacks) would be to estimate both shot and receiver waveforms. Let $A_{ij}(Z)$ denote the prediction error wavelet of the trace from the $i^{th}$ shotpoint and the $j^{th}$ receiver point. Let $U_{ij}(Z)$ be the logarithm computed by PLOG on Figure 2. We would then do a least squares problem of the type

$$U_{ij}(Z) \approx S_i(Z) + G_j(Z) \tag{11}$$

for shot waveforms $S_i(Z)$ and geophone waveforms $G_j(Z)$ . Our physical model is that we are determining the filtering response near to the shots and geophones which results from things like irregular weathering layers or space variable shallow water resonance. The computational effort in solving (11) is rather modest since each time lag may be computed separately. The number of variables is equal to the "fold" of the coverage squared, which makes a rather large set of simultaneous equations. However, it is so well conditioned and so sparse that iterative techniques converge extremely rapidly. After removal of the shot and geophone wavelets, their geometric mean could be restored to the data before statics are estimated by crosscorrelation and decomposition of a time residual matrix $t_{ij}$ into $s_i + g_j$ .