A TERRAIN NOISE SUPPRESSION FILTER FOR MAGNETIC PROFILES

by W. Scott Dunbar

Terrain noise is often a confusing factor in most potential field surveys. For magnetic surveys, the correction can be made through the use of Fourier transform techniques [1, 2]. These methods can also be used to correct gravity surveys. The advantage of the technique in [1] is that it includes non-linear terms.

It is also desirable to handle non-stationarity of the terrain noise. One approach to this problem is through the use of communication theory. A terrain noise suppression filter was developed by Clarke [3]. It assumes that the terrain noise is a linear function of the topography. However, through the use of short data series analysis techniques, it can be modified to handle non-stationarity in the spectrum of the topographic and magnetic profile. More study of the problem would undoubtedly reveal ways of merging the non-linear analysis with the non-stationary analysis.

We can regard the magnetic profile $m(x)$ as the sum of a signal component, $s(x)$, due to the "target source" and a noise component, $n(x)$, due to the shallower sources in the topography.

$$m(x) = s(x) + n(x) \tag{1}$$

The noise model is linear in the topography, $z(x)$:

$$n(x) = \int_{-\infty}^{\infty} g(x')\, z(x-x')\, dx' \tag{2}$$

The kernel, $g(x)$, is to be estimated from the available data $m(x)$ and $z(x)$.

The cross-correlation between $z(x)$ and $m(x)$ is

$$\phi_{zm}(x_0) = \int_{-\infty}^{\infty} z(x+x_0)\, m(x)\, dx$$

Substituting from (1)

$$\phi_{zm}(x_0) = \int_{-\infty}^{\infty} z(x+x_0)\, [\, s(x) + n(x)\,]\, dx$$

$$= \phi_{zs}(x_0) + \phi_{zn}(x_0)$$

Since the target is deeper than the noise sources, we can assume that $\phi_{zs}(x_0) = 0$, i.e., there is no correlation between signal and noise. Therefore, we only need to compute $\phi_{zn}(x_0)$ using (2):

$$\phi_{zn}(x_0) = \int_{-\infty}^{\infty} z(x+x_0)\, dx \int_{-\infty}^{\infty} g(x')\, z(x-x')\, dx'$$

$$= \int_{-\infty}^{\infty} g(x')\, dx' \int_{-\infty}^{\infty} z(x+x_0)\, z(x-x')\, dx$$

$$= \int_{-\infty}^{\infty} g(x')\, \phi_{zz}(x_0+x')\, dx'$$

Fourier transforming, $g(x)$ is given in the wavenumber domain

$$G(k) = \left[\frac{P_{zm}(k)}{P_{zz}(k)} \right]^*$$

where $P_{zm}(k)$ is the cross-power spectrum of $z(x)$ and $m(x)$ and $P_{zz}(k)$ is the auto-power spectrum of $z(x)$. The star denotes the complex conjugate. k is the wavenumber.

Substituting (2) into (1) and Fourier transforming, the terrain noise suppression is done according to

$$S(k) = M(k) - G(k)\, Z(k) \tag{3}$$

Assuming that the noise is uncorrelated with the signal amounts to limiting the analysis to geometries such as that in figure 1. Probably the most serious assumption is that of stationarity in the spectral estimates. A solution to this problem is to individually analyze pre-selected portions of the profile over which the topographic noise can be assumed stationary. These portions would probably be rather short and therefore the spectral estimates and Fourier transforms would not have much resolution in the wavenumber domain. However, the maximum entropy spectral estimator [4] can be used to estimate $P_{zz}(k)$ according to

$$P_{zz}(k) = \frac{P_{N+1}/W}{\left| 1 + \sum_{j=1}^{n-1} PEF_{j+1}\, e^{-i2\pi fj\Delta x} \right|^2} \tag{4}$$

where $PEF = (1, PEF_1, \ldots, PEF_N)$ is the prediction error filter of length N and with power output P_{N+1}. W is the Nyquist frequency of the data and Δx is the sampling interval. The cross power spectrum $P_{zm}(k)$ can also be estimated by the maximum entropy method according to

$$P_{zm}(k) = C_{zm}(k) - i\, Q_{zm}(k)$$

where $C_{zm}(k)$ is the cospectrum and $Q_{zm}(k)$ is the quadrature spectrum. $C_{zm}(k)$ and $Q_{zm}(k)$ can be estimated according to

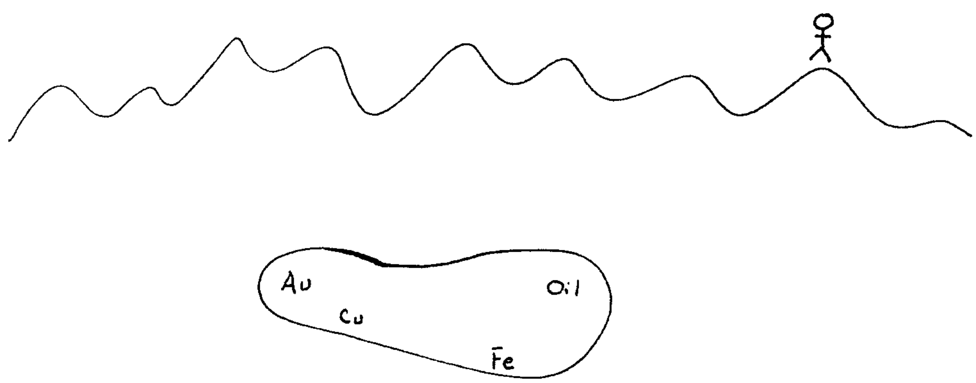

Figure 1. Geometry suitable for terrain filter.

$$C_{zm}(k) = \frac{1}{M\Delta x} [(P_{bb}(k) - P_{zz}(k) - P_{mm}(k)) / 2]$$

$$Q_{zm}(k) = \frac{1}{M\Delta x} [(P_{aa}(k) - P_{zz}(k) - P_{mm}(k)) / 2] \tag{5}$$

where M is the number of data samples and

$$a(t) = x(t) + i\,y(t) \qquad b(t) = x(t) + y(t)$$

Each of the quantities on the right hand side of (5) can be estimated by (4). $P_{aa}(k)$ is computed by a complex form of the algorithm. See [5] for a derivation of the preceding formula.

The prediction error filter is estimated by the Burg algorithm [6] from the data alone. The algorithm is a tricky little devil to write and its theory is probably not in the mathematical armory of most potential field analysts. The algorithm for complex time series is given in the Appendix.

Now that we have good resolution in the spectral estimates, we need some way of obtaining comparable resolution in the Fourier transforms $M(k)$ and $Z(k)$. Since the unit distance prediction error filter PEF is related to the unit prediction filter, PF, according to

$$PF_i = PEF_{i+1} \qquad i = 1, 2, \ldots, N-1$$

we can use PF to predict the samples of $m(x)$ and $z(x)$ out to greater lengths. More points in the space domain means more resolution in the wavenumber domain and thus the problem is solved. Since PEF is derived from the data alone, it can be seen that no information will be added to the data by doing this.

The prediction to distance $j\Delta x$ is done either by constructing the prediction filter for distance $j\Delta x$ and convolving it with the data or by convolving the unit distance prediction filter with the data j times each time incorporating a new point into the prediction scheme. The original data is extended in both directions, leaving the profile in the middle of the extended set. After applying the filter and transforming back to the space domain, the filtered profile, $S(k)$, is retrieved from the middle of the transform. Further details and proof of the above statements are given in [4] and [7]. A listing of a subroutine which does the prediction is given in the Appendix.

As stated before, the non-linearity can be handled by a somewhat different approach using Fourier transform techniques. Parker and Hueshs [1] compute the Fourier transform of the magnetization $T(x)$ of a magnetic layer of thickness h_0 according to

$$T(k) = \frac{M(k)\exp(|k|\,z_0)}{(\frac{1}{2}\mu_0)[1-\exp(-|k|h_0)]V(k)} - \sum_{n=1}^{\infty} \frac{|k|^{n-2}}{n!} C_n(k)$$

where $C_n(x)$ $T(x)$ $(Z(x))^n$, z_0 is the measurement height of the anomaly $m(x)$, μ_0 is the dipole moment of the earth's magnetic field and

$$V(k) = \hat{B}_0 \cdot (\hat{z} + i\hat{x}k/|k|)\,\hat{M}_0 \cdot (\hat{z} + i\hat{x}k/|k|)$$

Here M_0 is a constant unit vector in the direction of magnetization and B_0 is a unit vector in the direction of the ambient field. Only a few terms in the series are required for convergence. M_0 will undoubtedly vary over the length of the profile. However, the data extension methods could be used to analyze shorter portions of the data for which $\hat{M}_0$ can be assumed constant.

References

[1] Parker, R. L. and S. P. Hueshs, 1974. "The Inversion of Magnetic Anomalies in the Presence of Topography", JGR, v. 79, p. 1587.

[2] Courtillot, V., DuCruix, J., and J. L. LeMouël, 1973. Le Prolongement d'un champ de potentiel d'un contour quel conque sur un contour horizontal: une application de la methode de Backus et Gilbert", Ann. Geophys., v. 29, p. 361.

[3] Clarke, G. K. C., 1971. "Linear Filters to Suppress Terrain Effects on Geophysical Maps", Geophysics, v. 36, p. 963.

[4] Smylie, D. E., Clarke, G. K. C., and T. J. Ulrych, 1973. "Analysis of Irregularities in the Earth's Rotation", in Meth. of Comp. Physics, v. 13.

[5] Ulrych, Tad and Oliver Jensen, 1974. "Cross-Spectral Analysis Using Maximum Entropy", Geophysics, v. 39, p. 353.

[6] Burg, J. P., 1968. "A New Analysis Technique for Time Series Data", presented at NATO Advanced Study Institute on Signal Processing, Enschede, Netherlands.

[7] Ulrych, T. J., Smylie, D. E., Jensen, O. G. and G. K. C. Clarke, 1973. "Predictive Filtering and Smoothing of Short Records by Using Maximum Entropy", JGR, v. 78, p. 4959.

Appendix

Subroutine CBURG computes the prediction error filter (PEF) for a complex data series x . The inputs are:

X - complex data series with LX points

LPEF - desired length of PEF (< LX) .

The outputs are

PEF - the prediction error filter

ED - the error power of PEF

FE, RE - arrays of forward and reverse prediction errors

NP - print option (= 0 , no printed output)

The parameter LPEF can be chosen by looking at the curve of EP versus the recursion number N for N greater than about 4 . A drop in EP indicates a possible length of PEF . Several of these drops will be found and each one could be tried. LPEF is still an arbitrary parameter and reflects the underdetermined nature of the problem of estimating a power spectrum of a process from a finite length of data.

With this routine, all the parameters to compute spectra according to (4) is given. For real data series all the complex operations reduce to real operations and PEF is real.

Subroutine EXTEND extends a data set in both the forward and reverse directions using the prediction filter (PF) derived from PEF . The inputs are

X - real data series with LX points

PEF - prediction error filter with LPEF points. This array is changed to a prediction filter upon exit

NP - printoption (= 0 , no printed output)

LEX - desired number of points in the extended data set. This must be some integer multiple of LX greater than 1 .

The only output of this routine is EX , the extended data set. The length of EX , LEX , depends on the resolution desired. Usually LEX = 5 LX is sufficient.

```
      SUBROUTINE CBURG (X,LX,PEF,LPEF,EP,FE,RE,NP)
      COMPLEX X(LX),PEF(LPEF),FE(LX),RE(LX),CONJG,SN,PEFH,FEH
      LPF=LPEF-1
C
C  INITIALIZE ERRORS,ERROR POWER AND PEF(1)
C
      EP=0.
      DO 1 I=1,LX
      FE(I)=X(I)
      RE(I)=X(I)
    1 EP=EP+X(I)*CONJG(X(I))
      EP=EP/LX
      PEF(1)=(1.,0.)
C
C  COMPUTE PEF RECURSIVELY
C
      DO 6 N=2,LPEF
      SN=(0.,0.)
      SD=0.
      JS=LX-N+1
C
C  COMPUTE PEF(N),UPDATE ERROR POWER
C
      DO 2 J=1,JS
      SN=SN+FE(J+N-1)*CONJG(RE(J))
    2 SD=SD+FE(J+N-1)*CONJG(FE(J+N-1))+RE(J)*CONJG(RE(J))
      PEF(N)=-2.*SN/SD
      EP=EP*(1.-PEF(N)*CONJG(PEF(N)))
      IF(N.EQ.2) GO TO 4
C
C  UPDATE PREVIOUS COEFFICIENTS
C
      KUP=(N-1)/2+1
      DO 3 K=2,KUP
      L=N-K+1
      PEFH=PEF(K)+PEF(N)*CONJG(PEF(L))
      PEF(L)=PEF(L)+PEF(N)*CONJG(PEF(K))
    3 PEF(K)=PEFH
C
C  UPDATE ERRORS
C
    4 DO 5 J=1,JS
      FEH=FE(J+N-1)
      FE(J+N-1)=FE(J+N-1)+PEF(N)*RE(J)
    5 RE(J)=RE(J)+CONJG(PEF(N))*FEH
    6 CONTINUE
```

```
C
C END OF RECURSION,PRINT ALL DATA
C
      IF (NP.EQ.0) RETURN
      WRITE (6,10) LX,(X(I),I=1,LX)
      WRITE (6,15) LPEF,(PEF(I),I=1,LPEF)
      WRITE (6,20) EP
      NWR=LX-LPEF+1
      WRITE (6,25) (FE(I),I=LPEF,LX)
      WRITE (6,30) (RE(I),I=1,NWR)
C*****
   10 FORMAT ('1',' INPUT DATA SERIES: LENGTH=',I5//(2X,10F12.6))
   15 FORMAT (///' PREDICTION ERROR FILTER COEFFICIENTS: FILTER LENGTH=
     $  ',I5//(2X,10F12.6))
   20 FORMAT (///' ERROR POWER=',1PE10.3)
   25 FORMAT (///' FORWARD PREDICTION ERRORS'//(2X,10F12.6))
   30 FORMAT (///' REVERSE PREDICTION ERRORS'//(2X,10F12.6))
C*****
      RETURN
      END
```

```
      SUBROUTINE EXTEND (X,LX,PEF,LPEF,EX,LEX,NP)
      REAL X(LX),PEF(LPEF),EX(LEX)
C
C BURG PREDICTION FILTER
C
      LPF=LPEF-1
      DO 1 I=1,LPF
    1 PEF(I)=-PEF(I+1)
C
C PLACE X IN MIDDLE OF EX
C
      LEND=(LEX-LX)/2
      IX1=LEND+1
      IXL=LX+LEND
      J=1
      DO 2 I=IX1,IXL
      EX(I)=X(J)
      J=J+1
    2 CONTINUE
C
C FORWARD PREDICTION
C
      IF1=IXL+1
      IFL=IXL+LEND
      DO 4 I=IF1,IFL
      EX(I)=0.
      DO 3 J=1,LPF
    3 EX(I)=EX(I)+PEF(J)*EX(I-J)
    4 CONTINUE
C
C REVERSE PREDICTION
C
      IR=IX1
      DO 6 I=1,LEND
      IR=IR-1
      EX(IR)=0.
      DO 5 J=1,LPF
    5 EX(IR)=EX(IR)+PEF(J)*EX(IR+J)
    6 CONTINUE
C
C WRITE LEX,EX
C
      IF (NP.EQ.0) RETURN
      WRITE (6,10) LEX,(EX(I),I=1,LEX)
   10 FORMAT (///' EXTENDED DATA SERIES:LENGTH= ',I5//
     $(2X,10F12.6))
      RETURN
      END
```