



Using a 1D FX Predictive Filter for 3D Seismic Data Random Noise Attenuation

Mike Galbraith*

Geophysical Exploration & Development Corp. ,Calgary

mgalbraith@gedco.com

and

Zhengsheng Yao, Randy Kolesar

Geophysical Exploration & Development Corp. ,Calgary

Summary

For some time now, FXY has been used as a standard algorithm for random noise attenuation on 3D seismic data. Recently, however, users noticed that, when signal/noise ratio is low, FXY is not as successful. In this paper, we will demonstrate that a series of 1D FKxY predictive filters is superior to the 2D FXY filter in the presence of significant random noise.

Introduction

As a standard seismic processing algorithm, the 2D FXY predictive filter is extensively and routinely applied for seismic data random noise attenuation. FXY is a 2D extension of the 1D FX. FX works independently on each frequency slice of a 2D seismic data set (i.e. there is a different FX filter for each frequency slice) and it is actually a 1D predictive operator. This 1D operator has an excellent computational advantage which arises due to the fact that the convolution matrix is in the form of a Toeplitz or Hankel matrix. In the 2D case, the frequency slices of the traces form a vector (or “trace” of complex values). When the prediction filter is applied to 3D data, each frequency slice becomes a 2D matrix of complex values and FX is extended to FXY.) In FXY, the pure Toeplitz or Hankel matrix form no longer holds true and therefore, the computational cost increases. Recently, it has been noticed that unlike FX, FXY does not work well for the case when S/N level is low (e.g. Trickett, 2008), which has prompted us to re-examine FX and FXY. Since FXY is a 2D filter, it can be directly applied to a 2D seismic data set for testing. The result shows that the difference between the two filters is significant. When S/N level is low, it is readily apparent that FX is superior to FXY. Based on this observation, we propose a different form of predictive filter for 3D seismic data, namely FKxY.

FX vs FXY ---- a synthetic example with some discussions

In order to compare the effects of FX and FXY (here Y refers to the time variable, “t”), Figure 1 shows pure data and random noise to be used as test input data. First, noise free data is used as input. With a proper choice of filter lengths for FX and FXY the results and differences are shown in Figure 2a and 2b; and 2c and 2d. Even where both of them recover data very well, FXY produces blurry results on the signal. When noise is added to data, as shown in Figure 1c, and using exactly the same filter lengths, FX gives a much better result than that from FXY as shown in Figure 3.

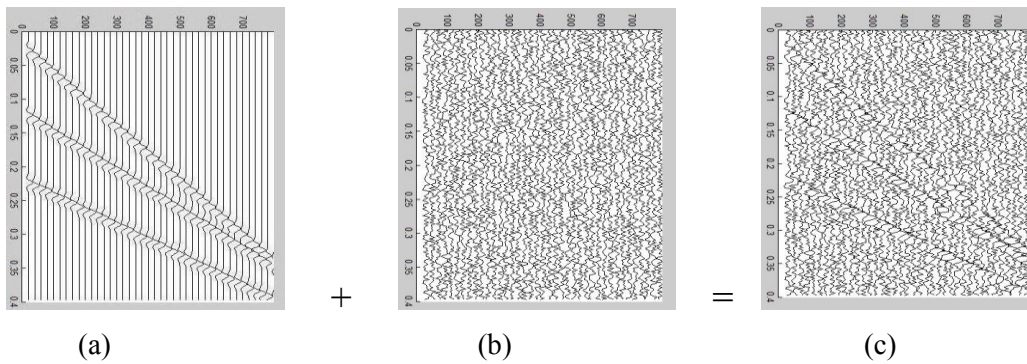


Figure 1. Synthetic data (a) pure data, (b) random noise and (c) noisy data = pure data+noise.

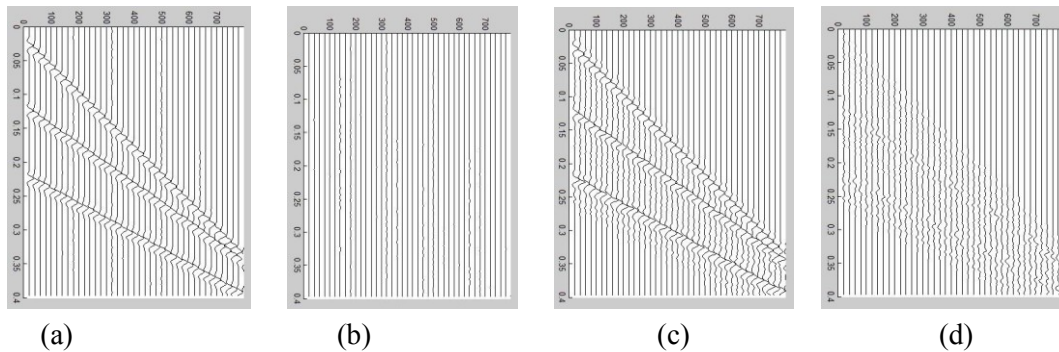


Figure 2, (a) FX applied to data in Figure 1(a); (b) Difference between Figure 2(a) and Figure 1(a); (c) FXY applied to data in Figure 1(a) and (d) Difference between Figure 2(c) and Figure 1(a)

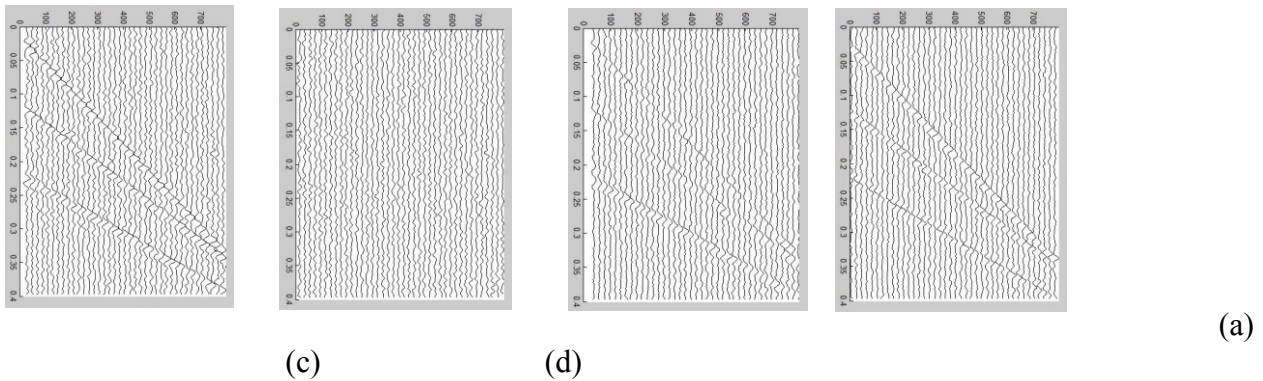


Figure 3. (a) FX applied to data in Figure 1(c); (b) Difference between Figure 3a and Figure 1a; (c) FXY applied to data in Figure 1c and (d) Difference between figure 3c and Figure 1a.

Both FX and FXY come from a linear invariant dynamic model and can be formulated as the same convolution equation except that the convolution operator for FX is a vector and for FXY is a matrix. By looking at the input data as in Figure 1, the data is not a function of variables of time and has X coordinates that are linearly time-space invariant. Therefore, there should be an inherent problem in using FXY. However, when we transfer the filter computation to the frequency domain, then within each frequency slice, such a condition for linear-invariance can be improved. For example, if there is only one linear event, then any neighbouring traces are different by a simple constant phase shift (e.g. Sacchi, 2009). Therefore, data in the f-x domain fits the model better than that in the t-x domain. Moreover, usually signal has its own

frequency band and within that band, the ratio of signal/noise is generally higher than the average shown in the t-x domain. For all of the above reasons, we can conclude that FX will always work better than FXY.

FKxY vs FXY – a 3D data approach

Each frequency slice from a 3D dataset can be viewed as a 2D matrix. Usually the signal events in the slice may not be a linear-invariant function because the seismic waves are traveling through geological structures. With the same idea as FX for 2D data, we can apply the Fourier transform to one of the spatial coordinates, say X. Now the frequency slice is in what we may call the FKxY domain. This transform works as a partial plane-wave decomposition and therefore, each Kx component is a 1D vector that better fits the linear model as discussed previously. Therefore, our approach is:

Input a 2D frequency slice (X,Y) \rightarrow fft to Kx,Y \rightarrow 1D filter for each Kx component

All the computational advantages of FX now exist because the 2D filtering problem has been separated to a series of 1D independent problems.

As an example, Figure 4 shows an inline section of a 3D synthetic data set with (a) pure signal; (b) random noise; (c) is the sum of (a) and (b). Data sets 4(a) and 4(c) are now used as input data for testing. Figure 5 shows the results from the proposed FKxY filter and FXY filter.

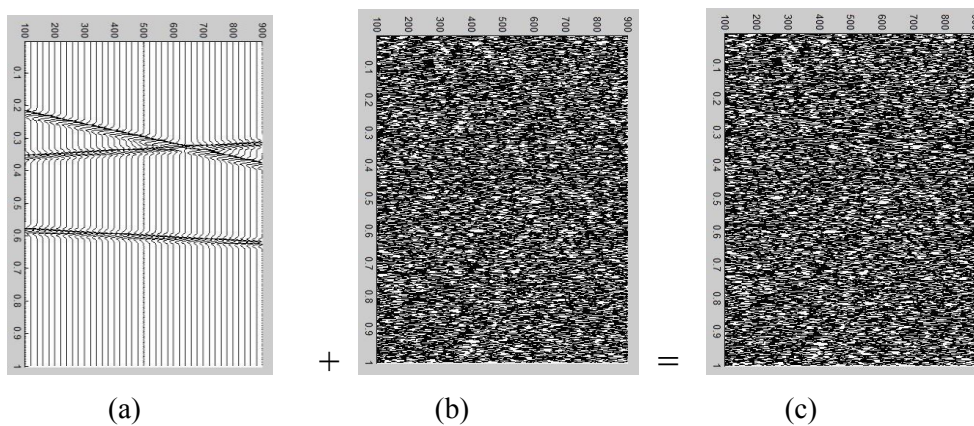


Figure 4. An inline section: (a) pure data, (b) random noise and (c) noise + data.

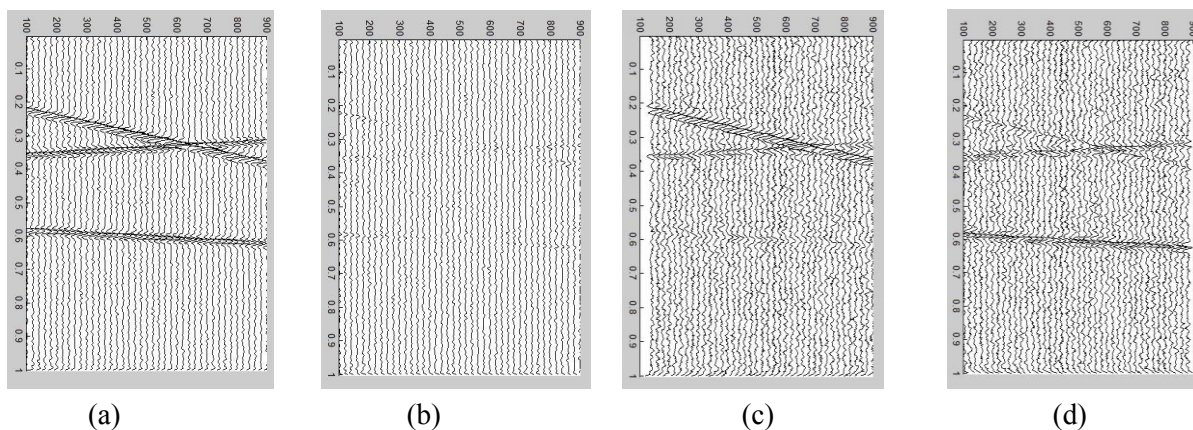


Figure 5. (a) FKxX applied to data in Figure 4c; (b) Difference between Figure 5a and figure 4a; (c) FXY applied to data in Figure 4c and (d) Difference between Figure 5c and Figure 4a.

Conclusions and discussions

It has been demonstrated that, compared to FXY, the FK_xY filter can significantly improve noise reduction while also having much greater computational efficiency.

References

Trickett, F-xy Cadzow Noise Suppression, CSEG Convention Abstracts, 2008

Sacchi, FX Singular Spectrum Analysis CSEG Convention Abstracts, 2009