Stanford Exploration Project, Report 111, June 9, 2002, pages 347–361

Implementing non-stationary filtering in time and in Fourier domain

Gabriel Alvarez[1]

ABSTRACT

Non-stationary filtering of seismic data can be accomplished in time or in Fourier domain by the theory of non-stationary convolution (Margrave, 1998). Here I show the results of implementing this theory for time-variant filtering of seismic data with an arbitrary number of filters and for forward and inverse NMO correction in the frequency domain. In the first case I show that the filters may be made to change sample-by-sample down the trace without artifacts being introduced and in the second case that the accuracy of the implied fractional sample interpolation can be controlled as an input parameter.

INTRODUCTION

The frequency content of seismic data decreases with time as a result of absorption as the wavefield travels through the earth. Therefore, for seismic interpretation it is desirable to bandpass-filter the seismic traces with different filters at different times. Usual practice consists of applying a few filters in predefined time windows which are made to overlap to provide a smooth transition between them. There are at least two problems with this approach: on the one hand, the overlap zones are rather arbitrary and phase distortion can be expected in them. On the other hand, with this approach we are limited to a few filters corresponding to the few chosen time windows.

A well-known alternative is the theory of non-stationary convolution and combination (Margrave, 1998; Rickett, 1999). This theory allows the design of arbitrary filters that can be made to change in a sample-by-sample manner. The design of the filters themselves is done in the frequency domain and its application to the data can be done in either the time domain, the frequency domain or a mixed time-frequency domain. In the time domain the process is similar to stationary filtering, with the columns of the convolutional matrix representing the delayed impulse responses of the filters applied to each sample, rather than the more familiar Toeplitz matrix of the stationary case. In the frequency domain, the convolutional matrix is nearly diagonal with the departure from diagonal being a direct indication of the degree of non-stationarity of the filters. In the mixed domain, the non-stationary filtering is performed

[1] **email:** gabriel@sep.stanford.edu

via a slow generalized Fourier transform.

Aside from the flexibility in choosing the domain of computation, we can also choose between non-stationary convolution and combination. The former is more appropriate when the spectra of the filters vary slowly and the latter when the change is sudden. Both non-stationary convolution and combination, however, reduce to stationary convolution in the limit of stationarity. This of course means that stationary filtering is a particular case of non-stationary filtering when the filters are kept constant for all samples.

In this paper I show the implementation of this algorithm for seismic trace filtering and for forward and inverse NMO correction. For the first application I used a set of randomly-generated seismic traces as well as a few traces of an actual seismic line. It will be shown by a time-frequency analysis of the data before and after the filter that it is indeed possible to change the spectrum of the seismic trace in a sample-by-sample basis without noticeable frequency distortions. For the NMO application I used a few CMP gathers consisting of five hyperbolic reflections and background Gaussian noise. It will be shown that we can pose the NMO-correction problem as a time-variant filtering problem and that we can control the accuracy of the underlying fractional sample interpolation as an input parameter.

THEORY OVERVIEW

Time-variant Filtering

In a linear time-invariant (stationary) filter the output $g(t)$ is related to the input $h(t)$ by the convolution

$$g(t) = a(t) * h(t) = \int_{-\infty}^{\infty} a(t-\tau)h(\tau)d\tau$$

where $a(t)$ is the impulse response of the filter. In order to extend the applicability of this simple expression for the response of a non-stationary filter, we could replace $a(t-\tau)$ in the previous equation with the more general expression $a(t,\tau)$, indicating that the impulse response itself is now a function of the input time τ. This expression, however, is too general and gives little insight into what the response of such a time-variant filter would be. Margrave, (Margrave, 1998) propose to maintain the convolutional nature of the impulse response by adding an explicit time dependence to it, that is, replacing $a(t-\tau)$ with either $a(t-\tau,\tau)$ (convolution) or $a(t-\tau,t)$ (combination). The formal definitions of non-stationary convolution and combination are, respectively (Margrave, 1998)

$$g(t) = \int_{-\infty}^{\infty} a(t-\tau,\tau)h(\tau)d\tau \quad (1)$$

$$\bar{g}(t) = \int_{-\infty}^{\infty} a(t-\tau,t)h(\tau)d\tau \quad (2)$$

These equations are clearly straightforward extensions of the stationary convolution concept. The introduction of the second index accounts for the non-stationarity. Comparing the two equations we see that the difference between non-stationary convolution and combination lies in the way the impulse responses are considered in the convolutional matrix. In non-stationary convolution the filter impulse responses (as a function of input time τ) correspond to the columns of the matrix, whereas in the non-stationary convolution case they correspond (as a function of the output time t) to the rows of the matrix (time reversed).

Just as with stationary filtering, it is convenient to find equivalent expressions in the frequency domain. These expressions are (Margrave, 1998)

$$G(f) = \int_{-\infty}^{\infty} H(F)A(f, f-F)dF \tag{3}$$

$$\bar{G}(f) = \int_{-\infty}^{\infty} H(F)A(F, f-F)dF \tag{4}$$

where f and F are the Fourier duals of t and τ respectively and $H(F)$ and $G(f)$ are the Fourier transforms of $h(\tau)$ and $g(t)$. It is interesting to notice from these equations that non-stationary convolution in time domain translates into non-stationary combination in the frequency domain and vice versa. This is as opposed to the stationary case in which convolution in time domain corresponds to multiplication in the frequency domain.

Since we now have two time indexes (t representing the filter samples and τ to keep track of the sample index to which each filter is applied), it is possible to have a third domain of computation, the so-called mixed domain in which the impulse response of the filters in the convolutional matrix are replaced with their corresponding frequency spectra. The equations to apply non-stationary filtering in the mixed domain are slow generalized Fourier transforms given by (Margrave, 1998)

$$G(f) = \int_{-\infty}^{\infty} \alpha(f,\tau)h(\tau)e^{-2\pi i f\tau}d\tau \tag{5}$$

$$\bar{g}(t) = \int_{-\infty}^{\infty} \alpha(F,t)H(F)e^{2\pi i Ft}dF \tag{6}$$

where $\alpha(p,v)$ is the so-called non-stationary transfer function which is the basic matrix in which the horizontal axis is time and the vertical axis is frequency (an example of this matrix will be given below). The transfer function is given by

$$\alpha(p,v) = \int_{-\infty}^{\infty} a(u,v)e^{-2\pi i pu}du \tag{7}$$

where $a(u,v)$ is the matrix of impulse responses, that is, the matrix with τ as its horizontal axis and t as its vertical axis (again, an example will be shown below).

Forward and Inverse NMO

The NMO correction time for the small offset-spread approximation is given by the well-known hyperbolic equation (Yilmaz, 1987):

$$\Delta t_{NMO} = t_x - t_0 = \sqrt{t_0^2 + x^2/V_s^2} - t_0 \tag{8}$$

where x is the trace offset, t_x is the two-way travel time at offset x, t_0 is the two-way travel time at zero offset (normal incidence trace) and V_s is the stacking velocity. Clearly, for a given trace different samples will have different NMO correction times even if the velocity is constant. Shallow events on the farthest trace with the slowest velocity have the maximum NMO-correction time whereas deep events on the near traces with the fastest velocity will have the minimum NMO-correction time. It is also important to note that in general some fractional sample interpolation will be required since we cannot expect the values of Δt_{NMO} to be integer multiples of the sampling interval.

In order to apply the non-stationary filtering algorithm we need to recast the NMO equation as an all-pass non-stationary filter that will simply shift each sample by the given value of Δt_{NMO}. This can easily be achieved in the frequency domain by a linear phase shift with slope proportional to the value of Δt_{NMO}. In principle, any value of Δt_{NMO} can be handled, so no fractional interpolation is required. For the sake of efficiency, however, it is convenient to precompute a given number of Δt_{NMO} values. The accuracy of the implicit fractional interpolation is determined by the number of precomputed Δt_{NMO} values and so can be controlled as an input parameter. Clearly, this parameter controls the trade-off between accuracy and speed of computation.

DESCRIPTION OF THE ALGORITHM

Time-variant Filtering

I will now summarize the steps necessary to perform non-stationary filtering in each of the three domains. In every case I will refer to non-stationary convolution but it is straightforward to change it to perform non-stationary combination instead.

Time Domain

The algorithm in the time-domain is:

1. Design the filters in the frequency domain. These may be trapezoidal tapered filters or some other suitable bandpass filter. We can consider these filters as making up a matrix whose horizontal coordinate is time τ (that is, the sample time of application of every filter) and whose vertical component is frequency as shown in the left-hand side of Figure 1.

2. Get the filter impulse responses in time domain. This essentially means taking an inverse Fourier transform of each column of the left panel in Figure 1.

3. Form the non-stationary impulse response matrix in time domain. The right panel in Figure 1 shows an example of this matrix for the case of three different filters to be applied in three windows of data. The wavelet is zero phase and the impulse responses are shifted so that the wavelet is centered along the diagonal.

4. Apply the non-stationary convolution. This is done by matrix multiplication between the matrix in the right panel of Figure 1 and the seismic trace to be filtered.

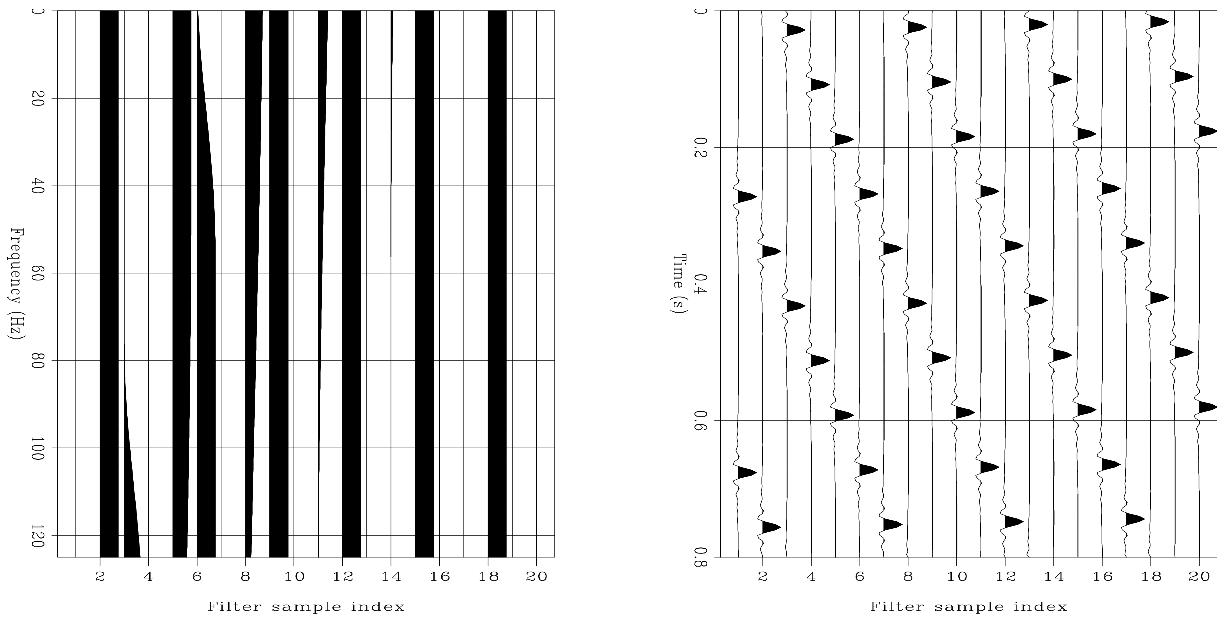

Figure 1: Filter design in the time-frequency domain. On the left, filter spectra as a function of time. On the right, impulse responses on time gabriel1-tvf_td1 [ER]

Frequency Domain

To develop an algorithm in the frequency domain, we basically have to take a Fourier transform in the horizontal direction of the data in the left panel of Figure 1. The algorithm is therefore:

1. Design the filters in the frequency domain, as before.
2. Take a Fourier transform in the horizontal direction (that is, a Fourier transform for each row of the matrix on the left panel of Figure 1) and form the corresponding frequency-domain convolutional matrix. Figure 2 shows the resultant matrix (amplitude spectrum only). This matrix is called the frequency connection matrix. On the left is a horizontally-shifted version of the matrix. The center "trace" corresponds to the stationary response and the "traces" away from it represent the departure from stationarity. Only a few "traces" are shown. On the right panel we have the complete dataset shifted so that the "stationary trace" is along the diagonal, which means that the off-diagonal energy represents again the departure from stationarity.
3. Take the Fourier transform of the input trace.
4. Multiply the frequency connection matrix (right panel of Figure 2) with the Fourier transform of the input trace to get the filtered trace in the frequency domain.
5. Take the inverse Fourier transform of the filtered trace to get it in the time domain.

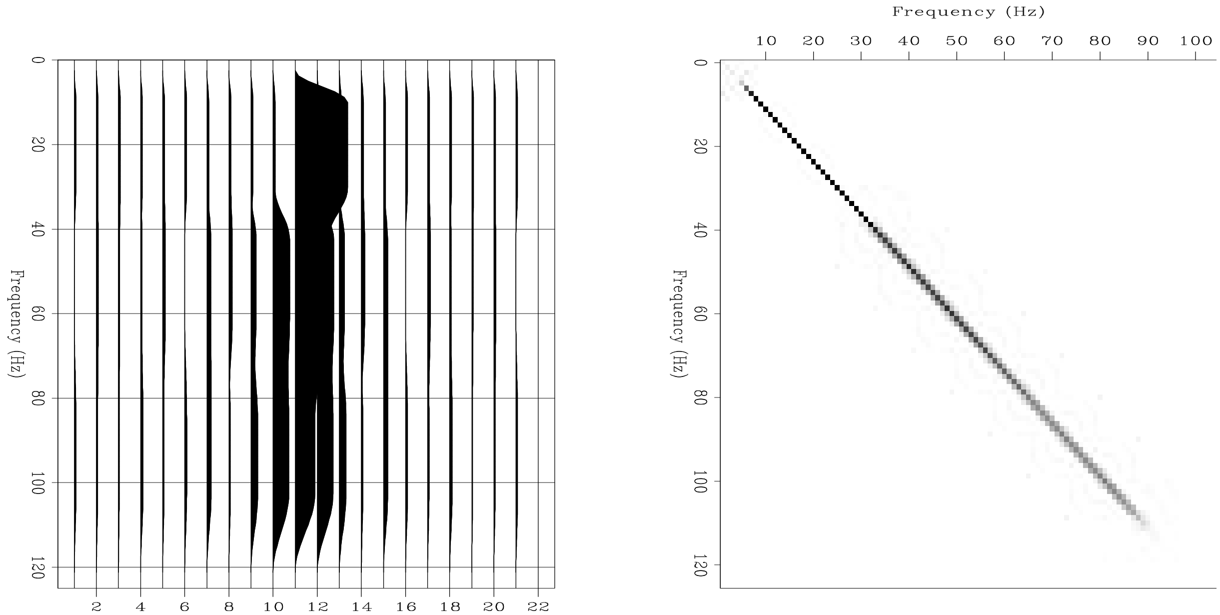

Figure 2: Frequency domain convolutional matrix. On the left, the filter spectra (amplitude only). The center "trace" represents the stationary response, the traces to the right positive frequencies and the traces to the left negative frequencies (only a few "traces" are shown). On the right the complete matrix shifted so that the stationary "trace" is on the diagonal of the matrix gabriel1-tvf_fd1 [ER]

Mixed Domain

In a sense this is the most natural domain for time-variant filtering because we design and apply the filters in the same domain. Hence, the algorithm is simpler:

1. Design the filters in the frequency domain as before and form the time frequency matrix (the so-called non-stationary transfer matrix). This is just the matrix shown on the left panel of Figure 1.
2. Apply the filter via the (slow) generalized Fourier transform given by equation (6).
3. Inverse Fourier transform the filtered data to the time domain.

Forward and Inverse NMO

The algorithm for both forward and inverse NMO with the non-stationary all-pass filtering approach is the following:

1. Compute the maximum and minimum NMO correction required in the data according to the offsets, reflection times and velocities.

2. Use the previous information to precompute all the NMO time shifts in the frequency domain by application of an all-pass non-stationary linear phase filter.

3. Take the inverse Fourier transform to get the precomputed impulse responses in time domain.

4. For each trace, form the impulse response matrix selecting from the precomputed ones those required according to the NMO shifts appropriate to that trace.

5. Carry out the filtering by multiplying the trace by the the impulse response matrix.

COMPUTER IMPLEMENTATION

Although the computer implementation of the above algorithms seems straightforward enough, I will mention some specific details here that, simple as they may be, are important when dealing with these algorithms

1. When using non-causal wavelets it is convenient to apply a time shift to the whole matrix so that the complete wavelet can be recovered. This can be easily done at the time of the computation of the filters in the frequency domain. This time shift needs to be compensated for after the filtering, of course.

2. In the time domain implementation only a few samples of the impulse responses are required to get a satisfactory result. This allows us to speed up the computation enormously because we need to multiply by a matrix that is non zero only near the diagonal as opposed to a dense matrix. We don't even need to store the complete impulse responses.

3. In the frequency domain similar, even more pronounced savings in computation, can be achieved by realizing that the frequency connection matrix is very nearly diagonal except for wildly varying filters. I found that good results could be obtained with only a few "traces" (perhaps 7 or 9) in the frequency domain.

4. After taking the horizontal Fourier transform in the frequency domain algorithm, it may be necessary to unscramble the traces to get both the positive and negative frequencies, otherwise only the amplitudes above or below the diagonal in Figure 2 will be present and the results will not be satisfactory.

RESULTS AND DISCUSSION

I will now illustrate the results of doing time-variant filtering and NMO correction with the algorithms described above.

Time-variant Filtering

For time-variant filtering I will show impulse responses as well as results with synthetic and real data.

Impulse Responses

Figure 3 shows a synthetic dataset comprising six spikes and their impulse responses when the dataset is filtered with three different filters. Convolution of a seismic trace with this dataset will filter it in the time-variant manner implied by the right panel of Figure 3.

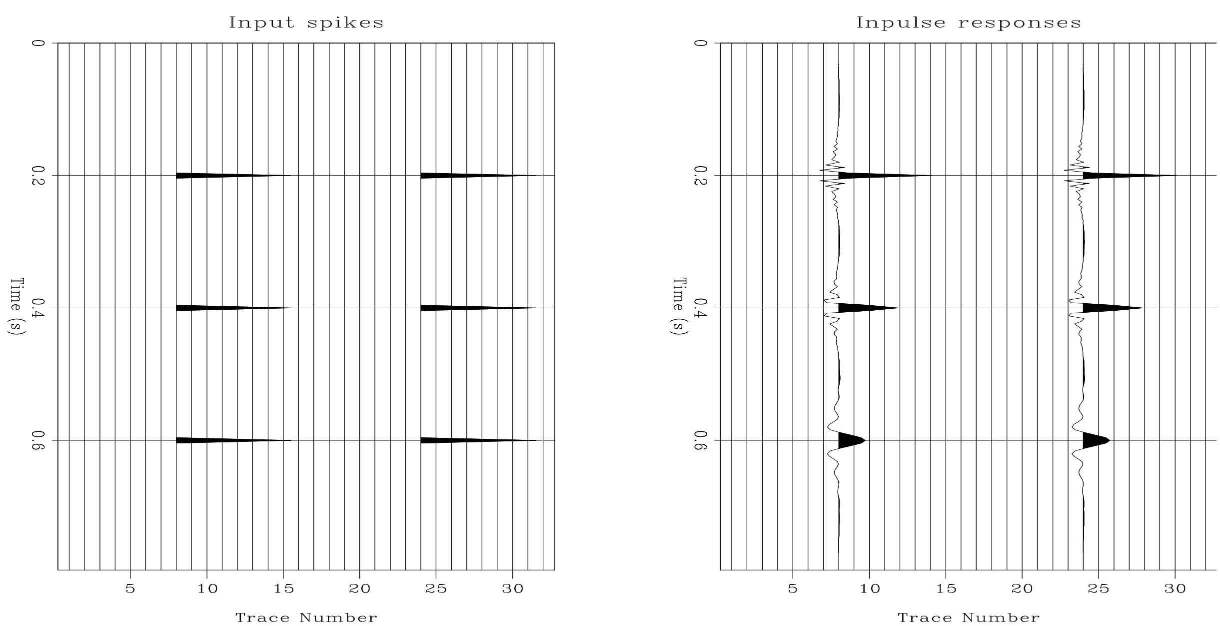

Figure 3: Impulse responses. On the left, the input spikes and on the right their corresponding impulse responses. gabriel1-tvf_ir1 [ER]

Random Traces

The top left-side of Figure 4 shows a dataset composed of randomly generated seismic traces. The top right-hand-side shows the result of filtering this dataset with constant filters applied in three time windows corresponding to the top third, the middle third and the bottom third of the

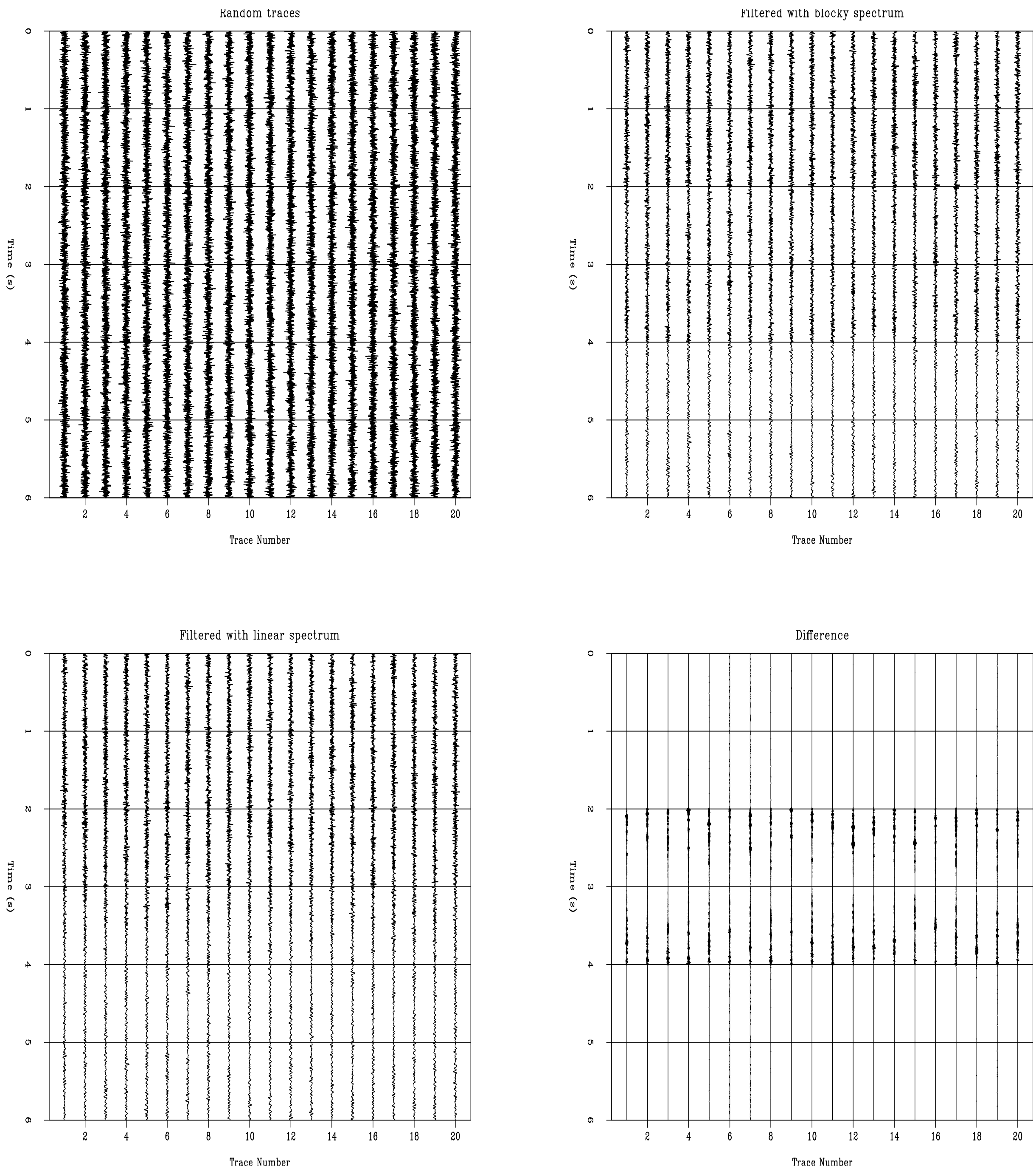

Figure 4: Random traces. On the top left, the input data. On the top right the filtered data with a "blocky spectrum," that is, constant spectrum in each of three time windows corresponding to one third of the trace length. On the bottom left data filtered with a linear spectrum in the middle third and constant spectrum in the top and bottom third. On the bottom right is the difference of the two filtered datasets. gabriel1-tvf_rt1 [ER]

traces. The applied filters were: in the top window (4-12-90-125) Hz, in the middle window (4-12-60-90) Hz and in the bottom window (4-12-30-50) Hz. The bottom left-hand-side shows the same dataset filtered a different way: the top third and the bottom third were filtered as before, but the middle third was filtered with a linearly changing spectrum that matched the filters above and below at the window limits. The bottom right-hand side shows the difference between the two filtered datasets. As expected, there is no difference in either the top or bottom third of the datasets. In the middle dataset, however, there is a difference which is greatest at the limits between the three zones where the difference in the filters is greatest.

Figure 5 shows a time-frequency analysis of the original data on the left and the filtered data with the linear spectra on the right. Since the data are random traces, they have all the frequencies from 0 to Nyquist (250 Hz in this case). After the filtering the spectrum is shaped by the applied filters. The bold line represents approximately the high pass frequency of the filters applied at each sample. There is no evidence of distortion in the spectrum even though the spectrum was made to change sample-by-sample in the middle third of the dataset.

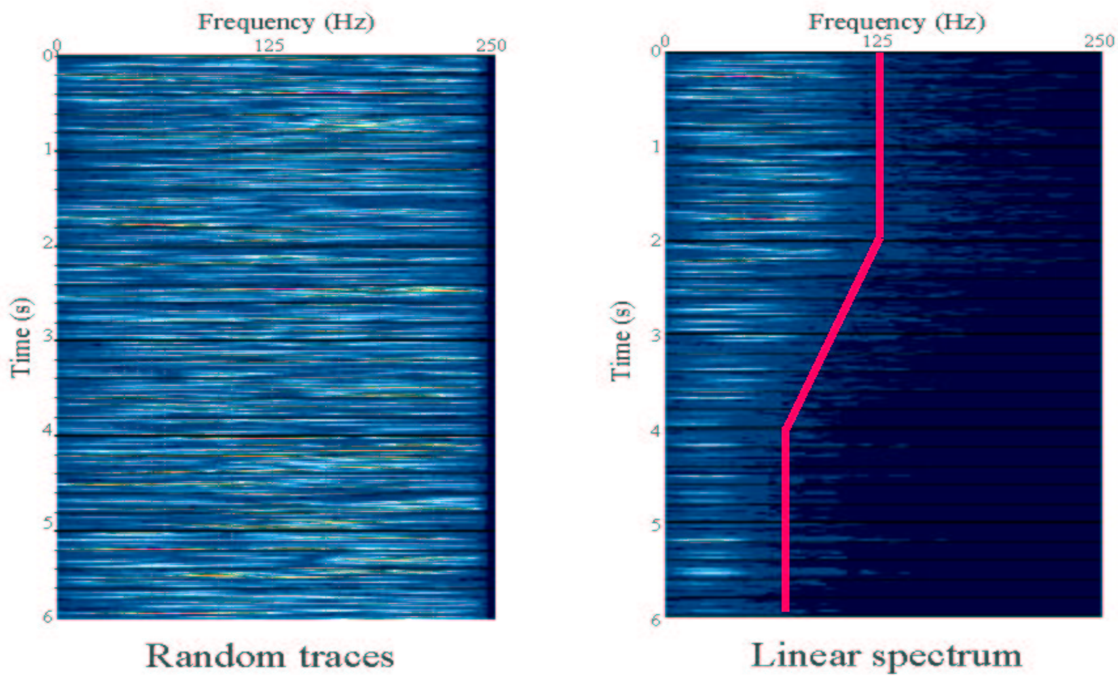

Figure 5: Time-frequency analysis for random data. On the left, the time-frequency display of the input data and on the right the result of the filter with the linear spectrum. The white areas represent large amplitudes. The thick solid line represents approximately the high cut frequency of the filters in every window gabriel1-tvf_tfa1 [NR]

Real Data

The top left-hand side of Figure 6 shows a few stacked traces from a real 2-D seismic line. The top right-hand side shows the result of filtering the data with a "blocky" spectrum in which

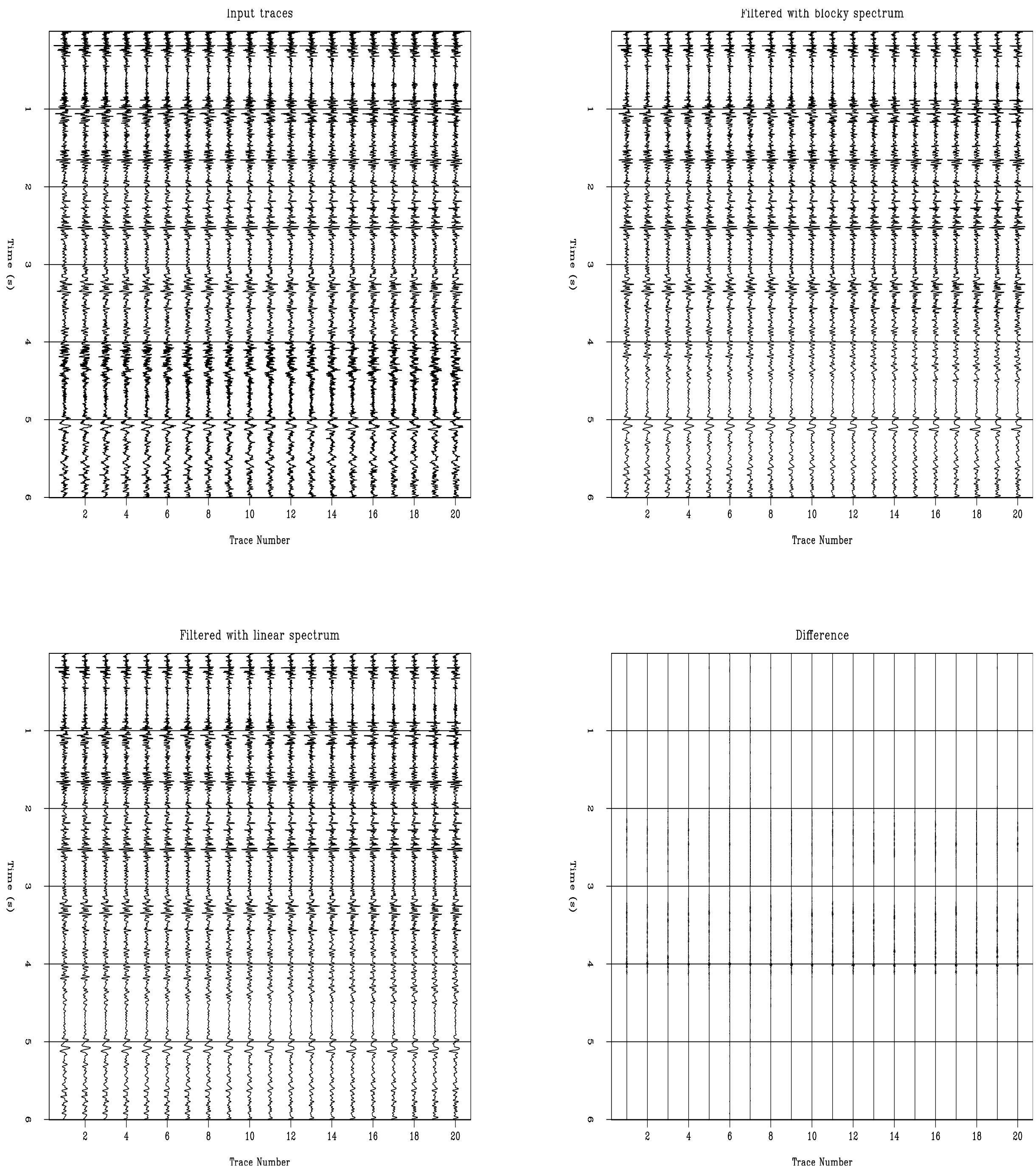

Figure 6: Real traces. On the top left, the input data. On the top right the filtered data with a "blocky spectrum," that is, constant spectrum in each third of the data. On the bottom left data filtered with a linear spectrum in the middle third and constant spectrum in the top and bottom third. On the bottom right is the difference of the two filtered datasets. gabriel1-tvf_rd1 [ER]

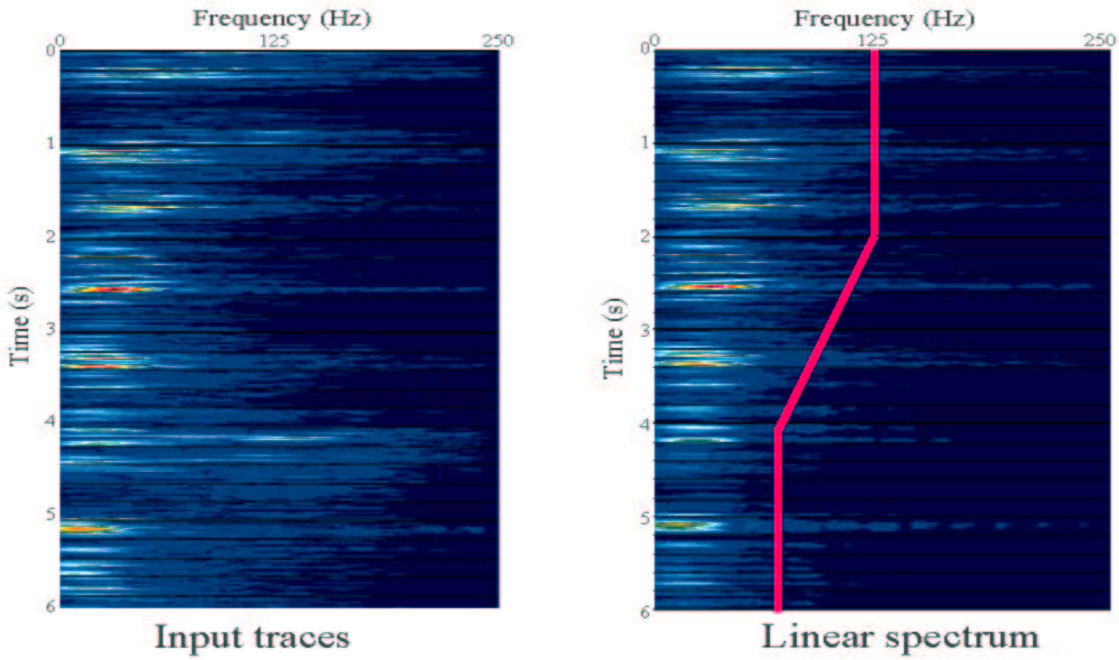

Figure 7: Time-frequency analysis for real data. On the left, the time-frequency display of the input data and on the right the result of the filter with the linear spectrum. The white areas represent large amplitudes. The thick solid line represents approximately the high cut frequency of the filters in every window gabriel1-tvf_tfa2 [NR]

again the top third of the data are filtered with one filter, the middle third with a narrower filter and the bottom third with an even narrower filter. The bottom left panel corresponds to the result of filtering the dataset with a linearly changing spectra in the middle third of the trace. The bottom right shows the difference between the filtered datasets. This time the difference is small even in the middle third of the trace because the original data spectrum is not very broad as shown in Figure 7. In this figure the left-hand side corresponds to the time-frequency spectrum of the data and the right-hand-side corresponds to the equivalent plot for the dataset filtered with the linearly-changing spectrum. As noted before, there is no apparent frequency distortion arising from the sample-to-sample change in the trace spectrum.

NMO correction

The top left-hand side of Figure 8 shows a modeled CMP gather consisting of five reflections in a Gaussian noise background. The top right hand-side shows the result of applying NMO correction using the time-variant filtering algorithm described above. The fractional sample interpolation parameter was set to two, meaning that a nearest neighbor interpolation was implicitly performed in the frequency domain to half the sampling interval. Changes in the fractional interpolation parameter would allow for better interpolation at the expense of increased run-time. The result shows a very good correction with very little distortion even

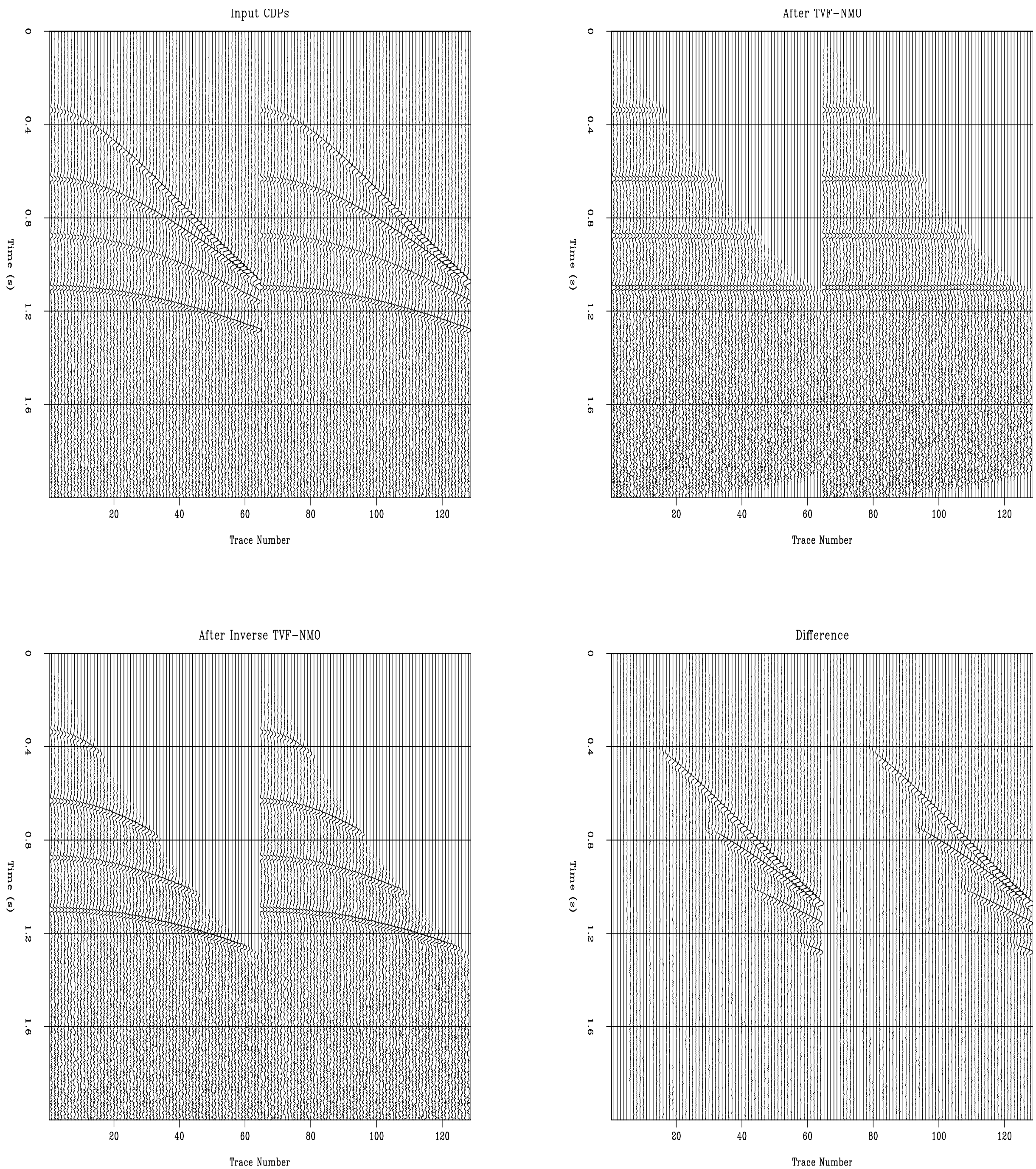

Figure 8: Frequency domain NMO correction. On the top left, the input synthetic data. On the top right the NMO-corrected data. On the bottom left inverse NMO-corrected data and on the bottom right the difference of the inverse NMO-corrected and the input data. Clearly the difference is due to NMO stretch gabriel1-nmo1 [ER]

for this small interpolation value.

The same algorithm used for forward NMO correction was also applied for inverse NMO correction. The bottom left panel of Figure 8 shows the inverse-NMO corrected CMP gather. Finally, the bottom right panel of Figure 8 shows the difference between the original data and the inverse-NMO result. Except for the obvious effect of the NMO-stretch mute, the inverse NMO-corrected CMP is nearly identical to the original CMP.

SUMMARY AND CONCLUSIONS

The theory of nonstationary filtering allows great flexibility in the design and application of time-variant filters. We can choose among three different domains of application: time, frequency and mixed and we can also choose between convolution and combination. From the results shown and from many other tests that I performed I drew the following conclusions:

1. Time domain should be preferred when filtering with "blocky" spectra, that is, when the filters are kept constant in each window and change abruptly from window to window.
2. Frequency domain should be preferred when using filters that change from sample-to-sample.
3. The mixed domain should only be used when filtering a few traces because of its large run-time.
4. As a general rule, non-stationary combination is preferable to non-stationary convolution when the filter spectrum changes abruptly from one sample to the next. For slowly changing spectrum, non-stationary convolution is probably better.

ACKNOWLEDGMENTS

I wish to thank Conoco for sponsoring me as an intern during the Summer of 2001 at its Seismic Imaging Technology Center (SITC) in Ponca City where I performed the work reported on here. In particular, I would like to thank Dr. Chuck I. Burch, my mentor at the time, and Phil Anno for helpful discussions on non-stationary convolution.

REFERENCES

Margrave, G., 1998, Theory of nonstationary linear filtering in the Fourier domain with application to time-variant filtering: Geophysics, **64**, 244–221.

Rickett, J., 1999, On non-stationary convolution and inverse convolution: SEP–**102**, 129–136.

Yilmaz, O., 1987, Seismic data processing: Society of Exploration Geophysicists.