NMO and beating the central limit theorem

Joe Dellinger and *Francis Muir*

INTRODUCTION

Motivation for this paper is provided by an ongoing study into a new, incremental NMO removal scheme. Although quite incomplete, we are sufficiently far along to discuss this method in outline.

NMO removal is usually based on a simple hyperbolic time-shift equation:

$$\text{output}(\text{Tout}) = \text{input}(\text{Tin}),$$

where the two times are related:

$$\text{Tout}^2 = \text{Tin}^2 - S(t)x^2.$$

$S(t)$ is Sloth (inverse velocity squared), perhaps a function of time, and x is the offset from source to receiver. In case we are interested in correcting for a range of n sloths, as for example in velocity analysis using multiple panels, then the equation becomes

$$\text{Tout}(j)^2 = \text{Tin}(j)^2 - S(j)x^2,$$

where j runs from 1 to n. (Note that when we write Tout(j) the time is implicit in Tout. The j refers to which velocity panel.) Here we are assuming the sloth does not vary with time, but only from one panel to the next.

If we select our Sloth set to be evenly incremented and beginning from zero, the same equation becomes

$$\text{Tout}(j)^2 = \text{Tin}(j)^2 - (j-1)\Delta S x^2$$

where ΔS is the incremental Sloth. This can in turn be written as a recursion:

$$\text{Tout}(1) = \text{Tin}(1),$$
$$\text{Tout}(j)^2 = \text{Tout}(j-1)^2 - \Delta S x^2,$$

where j runs from 2 to n.

This is better written in its more usual computational form

$$\text{Tout}(j-1) = \sqrt{(\text{Tout}(j)^2 + \Delta S x^2)}.$$

So far this looks like a lot of work for not much, but we can achieve several gains, which may turn out to be important on some fast but dumb computing devices:

(1) The time shifting equations need be computed but once, and can then be kept in a table look-up for each sloth iteration.

(2) If ΔS is chosen sufficiently small, then the differential shift at each increment might be kept below a sample interval, so that NMO removal becomes a straight-forward filtering operation, without the usual large integer time shift.

(3) There is a particularly simple way of calculating the weights of the 4-point scheme - even for the spectral interpolation case (see Appendix).

The negative to all this is that these repeated filtering operations would seem to bring in their wake some serious diffusion problems created by Central Limit Theorem considerations.

It is, then, the purpose of this paper to explore just what these Limit Theories really say, and how they can be "beat".

THE CENTRAL LIMIT THEOREM

In doing repeated shifting, we will repeatedly convolve our shifting convolutional operator with itself. The central limit theorem says that almost any convolutional operator convolved with itself enough times tends towards a Gaussian. This is *not* what we want. We must construct our operator so that the central limit theorem does not apply.

Repeated convolution in the time domain is equivalent to repeated multiplication in the frequency domain. In figure 1 we have plotted with a thick line the amplitude spectrum of some arbitrary filter, normalized so that the largest amplitude is 1. Below this curve are others showing how the amplitude spectrum "decays" with repeated autoconvolution. The only point which does not decay is the highest one, with a value of 1. The immediate neighborhood of this point entirely determines the behavior of this filter in the limit as the number of autoconvolutions becomes large.

Since this is the case, we can expand about this point (which we will call ω_0) in a power series in frequency:

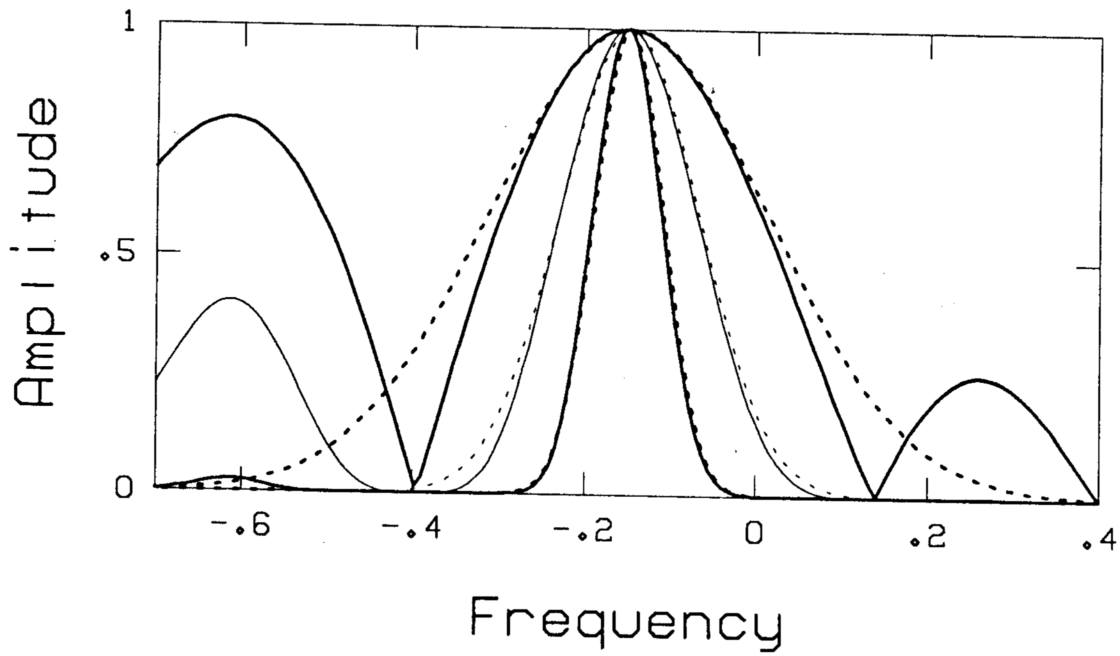

Figure 1: The effect upon the amplitude spectrum of repeated autoconvolution in the time domain. The original function is drawn solid. The Gaussian which fits the value and the second derivative at the maximum point is drawn dotted. Both spectra are also shown raised to the 4^{th} and 16^{th} powers, with the first and last drawn thick. The behavior of the amplitude spectrum in the immediate neighborhood of the largest amplitude determines the character of the filter in the limit, and in this region it approaches the form of a Gaussian.

$$F(\omega) = 1 - A_1(\omega - \omega_0)^1 - A_2(\omega - \omega_0)^2 - A_3(\omega - \omega_0)^3 + \cdots .$$

Given that the A's are well behaved, in a small enough neighborhood of ω_0 only the first nonzero term after the constant term is significant, since by choosing a small enough neighborhood we can make the higher order term's contribution proportionally as small as we like. This leading term cannot be one of odd power, or else $F(\omega_0)$ wasn't a maximum. In general the significant term should be the next one, which has power two.

To analyze what this implies, we can choose any other function which also has a peak at ω_0, is well behaved, and has the same first two terms in its power expansion. Furthermore, we wish to choose this function to have the special property that its general form does not change when it is squared. Of course, the function which fills the bill is the Gaussian

$$F(\omega) = e^{-A_2\omega^2}.$$

Since the Gaussian maps onto itself (although A_2 changes, of course), this must be the limiting form that most functions will tend towards under repeated autoconvolution. This is the essence of the central limit theorem. Since the inverse transform of a Gaussian is also a Gaussian, in the time domain we find that under repeated autoconvolution this filter tends more and more towards a Gaussian.

To break this theorem, we have only to make sure that A_2 is zero. In fact, we zero as many terms as possible. If this function is then raised to any positive power, these terms *stay* zero... there is no way to multiply other higher order terms together to regenerate these terms. The flatness in the spectrum near the peak does not go away upon autoconvolution, even though the base of the peak becomes narrower. (See figure 3.)

Analogously with the usual case in which the squared term is the significant one, if the first nonzero term is of the form $-A_n\omega^n$, the limiting form is

$$F(\omega) = \mathrm{e}^{-A_n\omega^n}.$$

Like the Gaussian this function has the property that when raised to any power it retains the same functional form.

Note that the larger n is, the more this approximates a "boxcar" function which is 1 in some interval containing ω_0 and 0 elsewhere. A filter with such a frequency spectrum has the interesting property that the effect is the same when it is applied any (positive) number of times.

HOW TO BEAT THE CENTRAL LIMIT THEOREM

Now to the construction of our special convolutional operators.

We imagine that our sample points are samples from a continuous, band-limited function. We wish to evaluate this function at a new set of sample points shifted by Δ from the old set. We let our operator have coefficients

$$W_{-N+1},\ W_{-N+2},\ \ldots,\ W_0,\ W_1,\ \ldots,\ W_N.$$

We pick the origin to be Δ to the right of sample W_0, and set the sample interval to be 1. The Fourier transform of this function is then

$$F(\omega) = \sum_{j=-N+1}^{N} W_j \mathrm{e}^{-i\omega(j-\Delta)}.$$

We will assume that the maximum value of this transform occurs at $\omega = 0$, and solve for the W's given this assumption. The assumption then turns out to be true for the filters we construct, so things are OK. As before, we expand in a power series about the maximal point:

$$F(\omega) = \sum_{j=-N+1}^{N} W_j \Big[\sum_{k=0}^{\infty} \frac{(-i\omega(j-\Delta))^k}{k!} \Big],$$

which can be rearranged to

$$F(\omega) = \sum_{k=0}^{\infty} \Big[\sum_{j=-N+1}^{N} W_j \frac{(-i(j-\Delta))^k}{k!} \Big] \omega^k.$$

We will refer to the coefficient of ω^k as A_k:

$$A_k = \Big[\sum_{j=-N+1}^{N} W_j \frac{(-i(j-\Delta))^k}{k!} \Big].$$

To beat the central limit theorem, we require that $A_0 = 1$ and as many $A_k = 0$ as we have degrees of freedom to set, which is for $k = 1,\ 2N-1$. Expressed in matrix form, this condition on the W's gives the following matrix equation:

$$\begin{pmatrix} 1 & 1 & \dots & 1 & \dots & 1 & 1 \\ (-N+1-\Delta) & (-N+2-\Delta) & \dots & (-\Delta) & \dots & (N-1-\Delta) & (N-\Delta) \\ (-N+1-\Delta)^2 & (-N+2-\Delta)^2 & \dots & (-\Delta)^2 & \dots & (N-1-\Delta)^2 & (N-\Delta)^2 \\ (-N+1-\Delta)^3 & (-N+2-\Delta)^3 & \dots & (-\Delta)^3 & \dots & (N-1-\Delta)^3 & (N-\Delta)^3 \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} W_{-N+1} \\ W_{-N+2} \\ \vdots \\ W_0 \\ \vdots \\ W_{N-1} \\ W_N \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}.$$

The $k!$ and $(-i)^k$ terms are constants across a given row of the matrix and have been canceled from the equations. This is a special form of a Vandermonde matrix. Efficient methods exist to solve such matrix systems in general (Golub, 1985). Note that these matrices become very ill-conditioned for large values of N.

RESULTS

In figure 2 we show how well these filters actually work. Using the 8 point filter, we can shift half a sample 65536 times, and smear over only about 10 samples.

If you look down the diagonal of this figure, you will notice that 16 applications of the 2 pointer smears about as much as 16^2 of the 4 pointer, 16^3 of the 6 pointer, or 16^4 of the 8 pointer. We know from the previous section that we can approximate any of these filters in the frequency domain with a function of the form

$$e^{-(\frac{\omega}{\omega_c})^n},$$

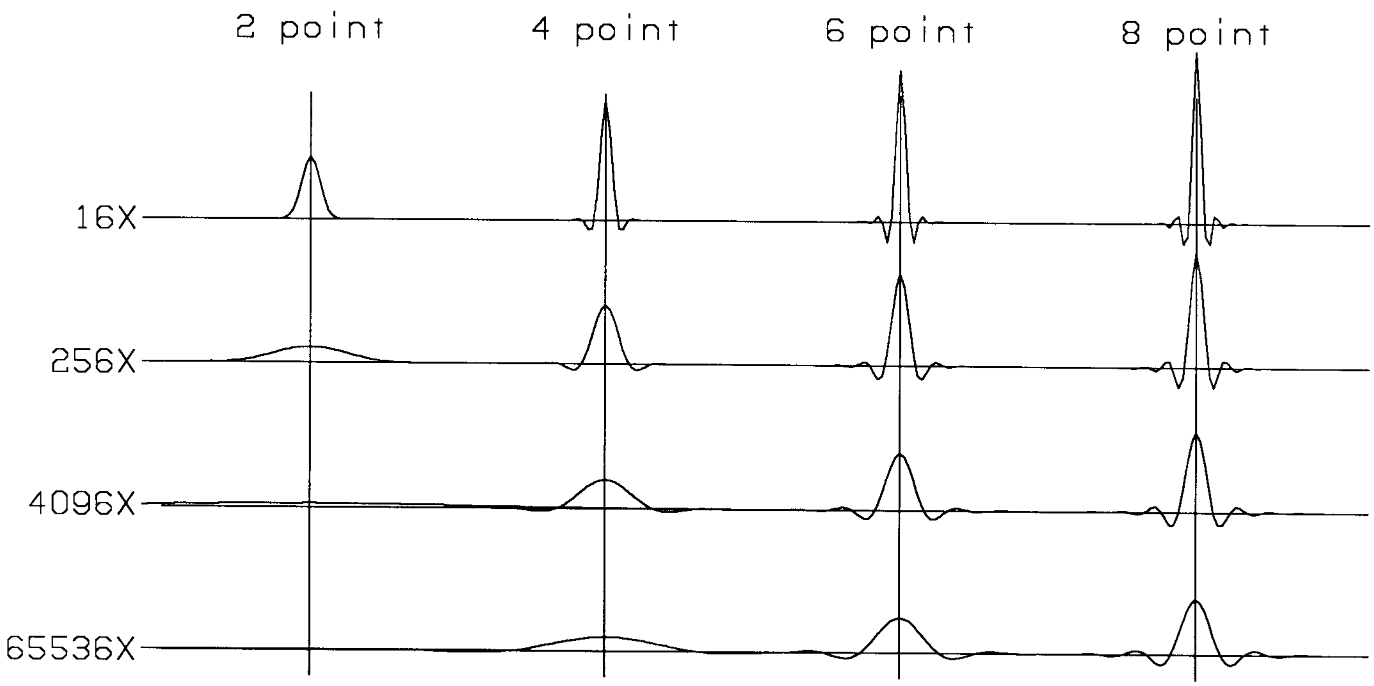

Figure 2: Results of repeated autoconvolution of our 2, 4, 6, and 8 point filters with $\Delta = .5$, which is the "worst case". The filters are shown after autoconvolution 16, 256, 4096, and 65536 times. The zero crossings of the lower-rightmost plot are 5 samples from the peak, to give scale. Note that 65536 convolutions with the 8 pointer smears only about as much as 16 convolutions with the 2 pointer.

where ω_c is the "corner frequency". For ω less than ω_c, $\frac{\omega}{\omega_c}$ is less than one and $-\frac{\omega}{\omega_c}^n$ tends towards 0 for n large. For ω greater than ω_c, $-\frac{\omega}{\omega_c}^n$ tends towards $-\infty$. Thus ω_c is the cutoff frequency in the limit for our approximate boxcar filter. The amplitude spectra of our filters are shown in figure 3.

Numerically, the amplitude spectrum of the two point filter is approximately $e^{-\frac{\omega}{2.9}^2}$, and after 16 passes this becomes $e^{-16\frac{\omega}{2.9}^2} = e^{-\frac{\omega}{.71}^2}$. Thus, $\omega_c = .71$. For the 4 point filter, we get $e^{-\frac{\omega}{2.6}^4}$, which becomes after 256 passes $e^{-256\frac{\omega}{2.6}^4} = e^{-\frac{\omega}{.64}^4}$ giving $\omega_c = .64$. Similarly for the 6 point filter after 4096 passes we get $\omega_c = .61$ and for the 8 pointer after 65536 passes $\omega_c = .59$. It is expected that this trend continues for higher order operators, but as yet no analytic results about the size of the first non-zero coefficient in the power series as a function of filter order are known.

CONCLUSIONS

Work on NMO removal will continue, but the theory developed in this paper is useful and instructive in its own right, and may find wide application.

APPENDIX

There are particularly simple rules for calculating the weights of the 4-point interpolator:

(1) Set the weights equal to the inverse of their distance from the location of the desired value.

(2) Multiply the two outer weights by -1/3.

(3) Scale the weights so that their sum is unity.

In case of frequency interpolation for, say, Stolt migration then:

(4) Multiply the first and third weights by -1.

(5) Scale all the weights by (COS(phi) - i*SIN(phi)), where phi is pi*del, and where del is the distance in sample units of the desired location from the second towards the third input points.

REFERENCES

Golub, G. H., 1985, Matrix Computations, The John Hopkins University Press, p. 119-123

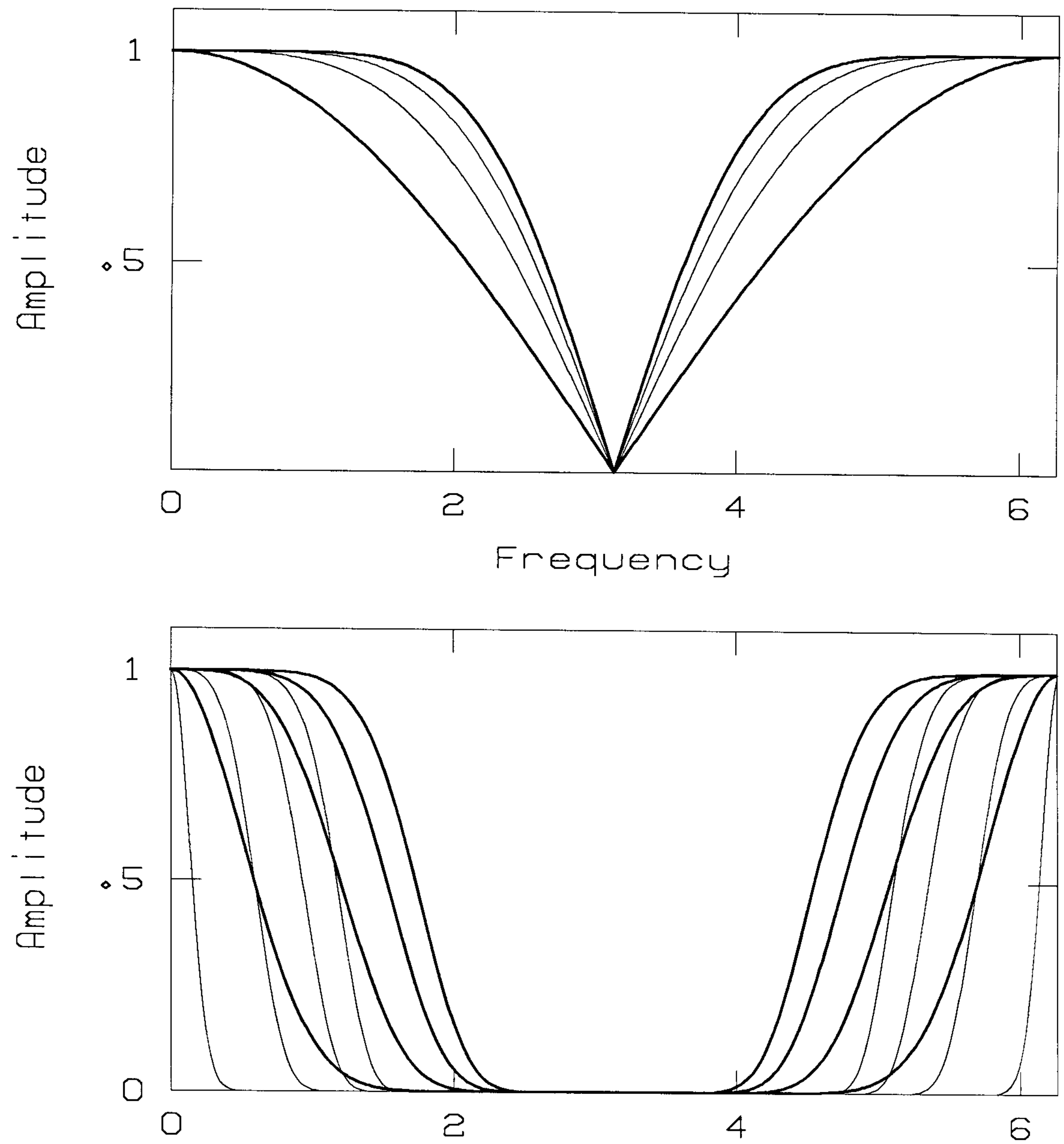

Figure 3: The upper plot shows the fourier transforms of the our 2, 4, 6, and 8 point shifting filters, for the case $\Delta = .5$. For Δ nearer to 0 or 1, the curve does not reach 0 at the nyquist and the amplitude spectrum is flatter all the way across. The higher the order of the filter, the flatter the curve is at $\omega = 0$. (Also note that the peak is indeed at 0, as promised.) In the lower plot, the amplitude spectrum after autoconvolution 16 (thick) and 256 (thin) times is shown.