

LOW-VELOCITY DECONVOLUTION

Jon Claerbout 3-26-2013

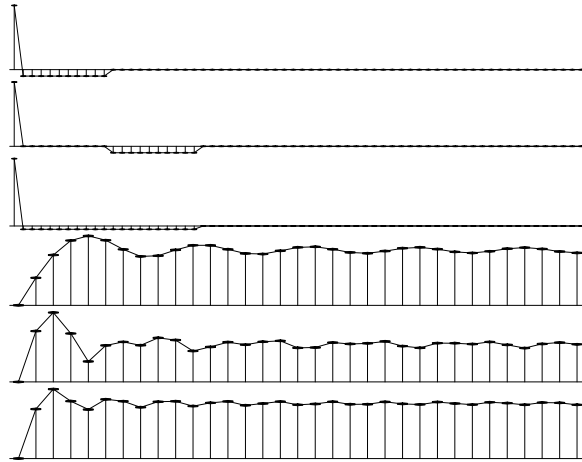
ABSTRACT

Decon destroys zero frequency. Zero frequency may be regarded as zero velocity. Here we extend the idea of gapped deconvolution to a gapped low-velocity destructor filter.

INTRODUCTION

Figure 1 compares three low-cut toy filters and their amplitude spectra. Each filter does a perfect job of eliminating zero frequency. (Imagine sea swell, for example.) Filters F2 and F3 have the same cutoff bandwidth about zero frequency whereas F1 is wider. Filter F2 has a more extreme Gibbs phenomena. All three filters have fairly flat spectra at mid to high frequencies; and their Gibbs oscillations could be flattened further by a design using triangles instead of rectangles.

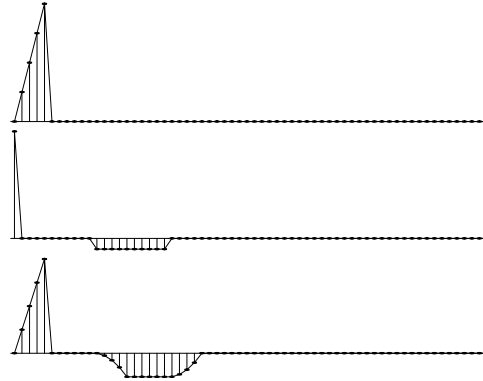
Figure 1: Three filters (F1,F2,F3) and their amplitude spectra.



What is the advantage of the gap? Figure 2 shows a toy example. Notice the input signal, a triangle, contains both high and low frequencies, but the triangle is perfectly preserved in the output (at a cost of a low-frequency blob behind it).

A decon filter with a very long gap, say 1/2 sec, would mainly destroy sea swell because that's all that can be predicted at such long gaps. I found a decon filter with a 60ms gap did an outstanding job of suppressing bubble while leaving signals with their ghosts perfectly untouched. The low-cut filter F2 has some resemblance to these decon filters. The difference being here we have a box shape with constants in it, while there the filter lived in the same box neighborhood but had adjustable coefficients designed to minimize output. That goal would minimize the trailing negative "blob" we see in Figure 2 while doing nothing to interfere with perfect preservation of signal.

Figure 2: (top) Triangle function input, (center) filter F2, and (bottom) output.



Notice that sea-swell is often treated as a low-frequency problem whereas in reality it is a low-velocity problem. We wish to treat it as a velocity problem to better preserve low frequencies in our signals. Our goal now is to design low-velocity rejection filters as we earlier designed debubble filters. These should destroy not only bubble and sea swell, but intermediate velocities associated with cable tug. Although I carry forward the discussion for the marine case, the greatest applicability may turn out to be with land data because of the dominance of ground roll.

This velocity filter will have the novel property that a low velocity suddenly appearing in the data will not be suppressed until after the gap time. With marine data most low velocities begin long before any signals arrive, so we may expect to be rid of them all. With land data we expect to be suppressing ground roll only gap-time after it first appears.

We'll be designing a filter in (ω, k_x) -space. Although it should simultaneously suppress bubble and slow noise, we'll test it in three stages, first the sea swell, then the slow noise, and finally the bubble.

EXTENDING THE DEBUBBLE THEORY

Now let us bring in the debubble theory and see where extensions need to be made.

Theory review

Let $W(\omega)$ be a wavelet in frequency domain. Let $U(\omega) = \ln W(\omega)$ so the wavelet is $W(\omega) = e^{U(\omega)}$. With the definition $Z = e^{i\omega\Delta t}$ fourier transforms become polynomials (Z -transforms). Thus $U(\omega)$ relates to the time function u_τ by the fourier sum $U = \sum_{\tau=0}^{2048} u_\tau Z^\tau$. The u_τ values will be our parameterization of the wavelet W . The τ axis may be called the "quefreny" axis or the "lag-log" axis.

The property of exponentials that $e^{A+B+C} = e^A e^B e^C$ has an interesting meaning when we exponentiate a Z -transform $\exp(A + B + C) = \exp(\sum_{\tau=1}^{2048} u_\tau Z^\tau)$. The Z -transform sum may be split up into small lags, medium lags, and large lags. This

decomposes a wavelet into a sequence of three wavelets, each with its own meaning where for example in common marine seismology:

$$e^{A+B+C} = e^A e^B e^C \quad (1)$$

$$e^{\sum_{\tau=1}^{2048} u_{\tau} Z^{\tau}} = e^{\sum_1^2} e^{\sum_3^{15}} e^{\sum_{16}^{2048}} \quad (2)$$

$$\text{(wavelet)} = \text{(continuity)}(\text{Ricker})(\text{bubble}) \quad (3)$$

Equation (2) defines the boundaries of the three regions abruptly although in practice we blend them smoothly. Changing the sign of $(A + B + C)$ changes a wavelet to its inverse, the decon filter.

Commonly we begin from Kolmogoroff spectral factorization (Appendix) giving us all the u_{τ} . We may design a wavelet e^{A+B+C} by over-riding Kolmogoroff with $A = 0$ and $B = 0$. To see what happens, consider the wavelet $e^C = 1 + C + C^2/2! + \dots \approx 1 + C$. For 4ms data and a 60ms gap the leading wavelet coefficients are: $(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, u_{16}, u_{17}, u_{18}, \dots)$. My paper ‘‘Ricker-compliant and pseudo-unitary decon’’ illustrates the application of the inverse of this wavelet. This operation on the data is called ‘‘debubbling’’. The result was outstanding. Bubbles were lifted off the data leaving primaries untouched.

Range of applicability

Start with a spectrum $S(\omega, k_x)$. The log spectrum is called the Cepstrum which might be denoted $\Xi = \Xi(\omega, k_x) = \log(S(\omega, k_x))$. I’ve been leaping off from Ξ with 2-D FT into a new place I need a name for. I’m thinking of calling it the ‘‘lag-log’’ space λ . It is analogous to autocorrelation with a lag τ axis, but what about its space axis? I am contemplating $\chi = \chi$ for this purpose. Thus, the lag-log space I’ll be talking about could be $\lambda(\tau, \chi)$.

The next question is when it makes sense to think of lag-log space $\lambda(\tau, \chi)$ instead of autocorrelations $C(\tau, \chi)$. Dropping down to one-dimensional space $\lambda(\tau)$ it made good sense to decompose source functions by lag because the bubble following an impulsive source convolves with the surface causing ghost. The spectra of these gapped filters multiply. On the other hand, often phenomena are additive in spectra, not multiplicative. For example the sea-swell spectra adds to the reflectivity spectra. This leads to the question of what, if any, examples might I find where the methods of this paper are appropriate? Unfortunately, my initial goal of separating signals from miscellaneous noises according to their velocities is clearly wrong.

Is there any hope to find more multiplicative processes? Perhaps the generation of shear waves is one such process. Events of differing slopes are linked.

Extending the theory to slow velocities

Fourier transforms will now go over both time and space. The sum of ω functions you see now in the exponent of equation (2) becomes a double sum of ω and k_x functions.

1. Form the (ω, k_x) amplitude spectrum of the data and take its logarithm.
2. Inverse transform the logarithm to (τ, x) space. (Here both τ and x are lag variables.)
3. Mask off negative time lags and double the positive ones. (This is the traditional Kolmogoroff step.) Subsequent “masking operations” are not masks but tapers, meaning like a smoothed step function.
4. Mask off inner lags. (This is the “gapped wavelet” step, the step that ignores ghosts and limits the process to bubbles.)
5. Here is the new step: Mask off coefficients outside the range $|x| > v|\tau|$ leaving a narrow (low-velocity) vertical wedge. Because the masks have smoothed off edges, this step is likely integrated with the previous one.
6. Forward FT to (ω, k_x) space.
7. Exponentiate your function of (ω, k_x) to get your wavelet. This wavelet models the noises you wish to eliminate.
8. While still in FT space, divide your data by your wavelet.
9. Return to (t, x) space. Try this on shot gathers and try it on sections.

STOP HERE?

Let us think about what we have when in FT space we divide our data by our wavelet. Our data FT2D is rough because the data spans long distances in (t, x) . Our wavelet FT2D is smooth because it spans a short distance in x and τ lags. Before we bounded the range of the wavelet in lag space it represented an average of the entire data set. It was an impulse response for the entire data set, signals with noises. After bounding of the range it is an impulse response for noises only. Dividing them out pushes them down, but does not make them zero.

Hyperbolas in lag space

While pouring over unsuccessful tests I explained to Martin Morf that I believed my problem was overfitting — too many adjustable parameters — and that I needed to think carefully about the appropriate shaped masks on $\lambda(\tau, \chi)$ separately for sections

and for gathers. He suggested an intriguing possibility. “Try hyperbolic boundaries on the masks.” What makes it intriguing is that passive seismology suggests hyperbolic events are natural.

Hopes for success

Many geophysicists have cleaned up shot gathers only to find final stacks unchanged. They had removed noises that would later be stacked away anyway.

That could happen here too. How then might velocity decon become a winner? It might remove aliased velocities, something stacking does not do well. How does it remove spatially aliasing? It should be good at noticing very slow velocities. It senses and destroys any correlation between $d(t, x)$ and $d(t + 20\Delta t, x + \Delta x)$ which is a strong sign of aliasing.

The main weakness of this approach was mentioned earlier. It is not intended for situations where noises are additive. It presumes the spectrum is built up in a multiplicative manner.

Coding tips

The directory `/homes/sep/jon/res/qdecon` contains a program `Dipfilter.rst` that does basic steps you’ll need to repeat. It references (ω, k_x) space using the indexing inherent to FFT. It extends both axes lengths to 10% above some power of 2. It does not pad time with zeros, instead with a linear interpolation from the last data value wrapping back around to the first. It defines a transitional ramp function between zero and unity as $1/2$ plus $1/2$ a sine wave from $-\pi/2$ to $+\pi/2$ over a range given by some low frequency (`flow`) defined in field units. For example, on 2 ms data, the maximum frequency is 250 Hz. Setting `flow=2.5` means a step function becomes a ramp function over 1% of the available frequency axis. Should this parameter be specified in lag units instead of frequency units? Perhaps.

The 1-D decon program you will be extending to 2-D lies in `/homes/sep/jon/res/qdecon/Rickerdecon.rst` It contains additional features for deconvolving ghosts that may be passed over.

DATA SUITABLE FOR TESTS

1. `/data1/chevron2msec/cvxAus.H` Slices from the new Chevron Australia marine data cube.
2. `/data/wz?from?around?the?world/wz.10.H` where “?” matches underscore. Land data in GEE Figure 9-10, known locally as `wz10`.

3. /net/server/book/gee/Data/gravel2D.H Figure 9.9 in GEE, "Gravel plain".

TESTING

Testing was frustrating. An old problem cropped up, a problem we have often encountered in many contexts, a problem we have no general solution for, a problem where we generally throw together an ad hoc approach. That's what I do next.

The essence of the problem here is that the data is strongly made of two different models. It is an oversimplification to say the data spectrum is a sum of two spectra, the sea-swell spectrum and the reflectivity spectrum, because the latter has a more complicated model. Making matters worse, the stronger of the two models is the one we need to get rid of.

To progress with the problem at hand, I'm simply going to try make a good model of the sea-swell; subtract it off; and never return.

The Bandpass program we have is too crude. At the $t = 0$ onset for all x it gives a large unwanted transient. (When the data is copied to a 2^N size mesh there is another one at the end.)

The best way to live with the Bandpass program is to be sure to apply it before any windowing is done. Then be sure to window off at least the first 100ms before beginning other data analysis.

APPENDIX

Kolmogoroff spectral factorization

When a time function such as c_t vanishes at all negative time lags it is said to be causal. Its Z transform is $C(Z) = c_0 + c_1Z + c_2Z^2 + c_3Z^3 + \dots$. Observe that $C(Z)^2$ is also causal because it has no negative powers of Z , alternately, because the convolution of a causal with a causal is causal. Likewise e^C is causal because it is a sum of causals.

$$e^C = 1 + C + C^2/2! + C^3/3! + \dots \quad (4)$$

Happily, this infinite series always converges because of the strong influence of the denominator factorials. The time-domain coefficients for e^C could be computed the hard way, putting polynomials into power series, or e^C may be computed by Fourier transforms. To do so, we would evaluate $\exp(C(Z = e^{i\omega}))$ for many real ω , and then invoke an inverse Fourier transform program to uncover the time-domain coefficients.

Let $r = r(\omega)$, $\phi = \phi(\omega)$, and $Z^\tau = e^{i\omega\tau}$. Let us investigate the consequence of

exponentiating a causal filter.

$$|r|e^{i\phi} = e^{\ln |r|+i\phi} = e^{\sum_{\tau} c_{\tau}Z^{\tau}} = \exp\left(\sum_{\tau} c_{\tau}Z^{\tau}\right) \quad (5)$$

Notice a pair of filters, both causal and inverse to each other.

$$|r|e^{i\phi} = e^{+\sum_{\tau} c_{\tau}Z^{\tau}} \quad (6)$$

$$|r|^{-1}e^{-i\phi} = e^{-\sum_{\tau} c_{\tau}Z^{\tau}} \quad (7)$$

A filter from any such pair is said to be “minimum phase”. Many filters are not minimum phase because they have no causal inverse. For example the delay filter Z . It’s inverse, Z^{-1} is not causal. Such filters do not relate to a casual complex logarithm. If they have a logarithm, it must be non-causal.

Given a spectrum $r(\omega)$ we can construct a minimum-phase filter with that spectrum. Since $r(\omega)$ is a real even function of ω , the same may be said of its logarithm. Let the inverse Fourier transform of $\ln |r(\omega)|$ be u_{τ} , where u_{τ} is a real even function of time. Imagine a real odd function of time v_{τ} .

$$|r|e^{i\phi} = e^{\ln |r|+i\phi} = e^{\sum_{\tau}(u_{\tau}+v_{\tau})Z^{\tau}} \quad (8)$$

The phase $\phi(\omega)$ transforms to v_{τ} . We can assert causality by choosing v_{τ} so that $u_{\tau} + v_{\tau} = 0$ for all negative τ . This defines v_{τ} at negative τ . Since v_{τ} is odd, we also know its values at positive lags. This creates a causal exponent which creates a causal minimum-phase filter with the specified spectrum. The code does this by multiplying u_{τ} by a step function of height 2, the doubling accounting for the zeroing of half the axis. This computation is called Kolmogoroff spectral factoring. The word “factoring” enters because in applications one begins with an energy spectrum $|r|^2$ and factors it into an $re^{i\phi}$ times its conjugate (time reverse).

Step Dip Decon, by Jon Claerbout SEP-77