Enhancing the peakmap with a low-pass filter

Andrew Miller

8 March 2017

1 Summary

1. Create a simulated signal that is an ideal filter.
2. Take 2D FFT of all signals added together. This creates a power spectrum of the ideal signal.
3. Cube the power spectrum, then multiply it by the 2D FFT of an image from real data that contains an r-mode injection (white noise was used here).
4. Inverse FFT2 the product to create enhanced t/f map.
5. Normalize and smooth these new amplitudes, then multiply by the original peakmap to obtain enhanced peakmap image.

Results: 3-5x increase in maximum amplitude of the r-mode signal around the injection in white noise compared to the mean critical ratios. We obtain different enhancement factors depending on which r-mode signal we use ($f_0 =$ 1000 Hz and $f_0 =$ 900 Hz result in different improvements). However the r-mode signal in the data was more of a line with a negative slope than a real r-mode signal, so we may see more of an improvement when the injection better matches the simulated signals in the filter (when we look at higher frequencies).

2 Justification

The r-mode waveform is a continuous, low frequency signal in the context of an image. The noise that surrounds the waveform is quite discontinuous, and is a high frequency signal in the context of an image. Hence, a low-pass filter would allow us to reduce the effect of the noise by filtering out the high frequency noise and enhancing the low-frequency r-mode signal. The r-mode frequency decreases over time in a nonlinear way that is distinct from the time-evolution behavior of a continuous wave signal, which is essentially a constant frequency over time. This low-pass filter does not work with horizontal lines or with increasing frequencies (spinups).

3 Creating the low-pass filter

The low pass filter is created from simulated r-mode signal data. In theory, it is good to explore the whole parameter space of r-modes, but for now we have just chosen a strong r-mode signal. The frequency and amplitude evolution of r-modes were simulated by following the Owen et al. 1998 model:

$$f(t) = \frac{f_0}{(1 + \lambda\alpha^2 f_0^6 (t - t_0))^{1/6}} \tag{1}$$

$$\lambda = (2\pi)^6 \frac{12Q}{\tau(\pi G\rho)^3} \approx 10^{-20} \tag{2}$$

λ is a constant related to the equation of statement parameter Q, average density of neutron star ρ; α is the amplitude at which the r-mode saturates and is a measure of how much rotational energy of the neutron star goes into the r-mode gravitational waves. λ has units of Hz^{-5}. f_0 is the initial frequency of the neutron star, the frequency at time t_0. The strain on the detector is:

$$h(t) = 1.8 \times 10^{-24} \left(\frac{20}{d}\right) \left(\frac{f(t)}{1000}\right)^3 \alpha \tag{3}$$

To simulate the signals, we inputted amplitudes and frequencies that followed the above behavior, and then interpolated these values based on dt and n. The phase evolution was calculated by:

$$\phi = \cos(\Sigma(f) * dt * 2\pi) \tag{4}$$

where Σ is a cumulative sum of a vector of frequencies f.

Once we simulated a signal, we created a time/frequency spectrum for by taking a one-dimensional FFT of the amplitudes. See fig. 1. We will define the 2D DFT as:

$$A(u, v) = F(a) = \Sigma\Sigma a(t, f) e^{-i(ut+vf)} \tag{5}$$

where $a(t, f)$ are the amplitudes of the signal, t is the time, and f is the frequency. u and v are the 'new frequencies' in Fourier space and don't have any physical meaning. The notation $F(a)$ denotes the two dimensional FFT of an image a, which is a matrix of critical ratios at each time/frequency.

We find the power spectrum of the r-mode signal by y taking the two-dimensional FFT of the time/frequency map (that is colored with amplitudes) of this simulated data. See fig. 2. In the future, when we have more r-mode signals injected in data, we will simulate correctly spaced r-mode signals in the power spectrum, so that fig. 1 will contain more r-mode signals, and fig. 2 will approach an 'ellipse' in Fourier space.

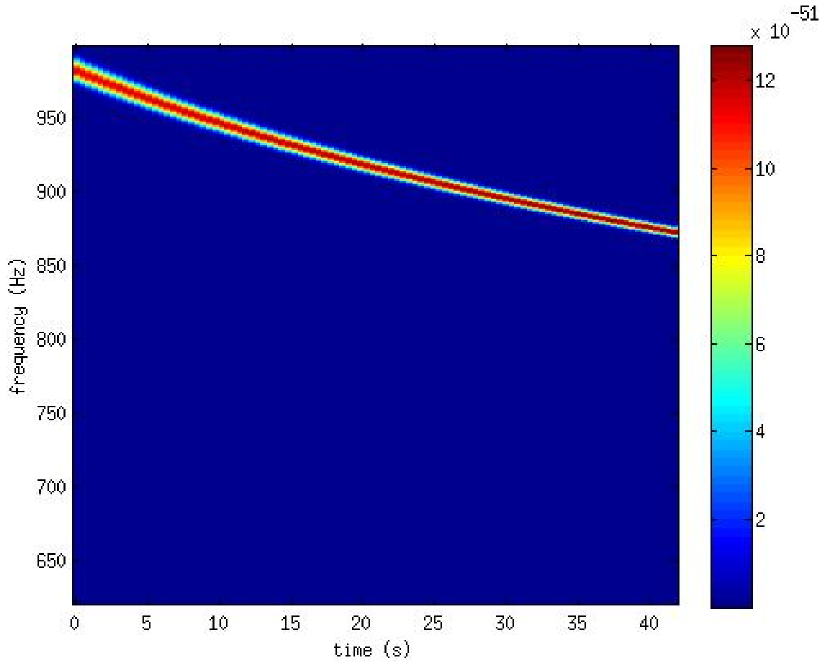

Figure 1: time/frequency map, simulated data

3.1 Simulation parameters

1. n: number of samples- 100,000
2. dt: sampling time- 1/(2*fmax) = 1/2000 s
3. nfft: number of ffts in samples- calculated so that there would be 8190 frequencies
4. nover: number of samples to overlap- calculated so that there would be 99 times. Lower means more times, raise means less times

99x8190 is size(t/f image) that the filter will be multiplied to.

r-mode injection parameters:

1. f_0: 1000 Hz
2. d: 20 m
3. $\alpha = 0.1$

The filter, in Fourier space, is shown in fig. 2 (when it is centered-otherwise the low-pass information will be in the corners of the image). Note that we have only simulated 45 seconds worth of data.

Additionally, since we were working with data up to 128 Hz, we arbitrarily made the amplitude (eqn. 3) of the r-mode signal a factor of 1000 stronger,

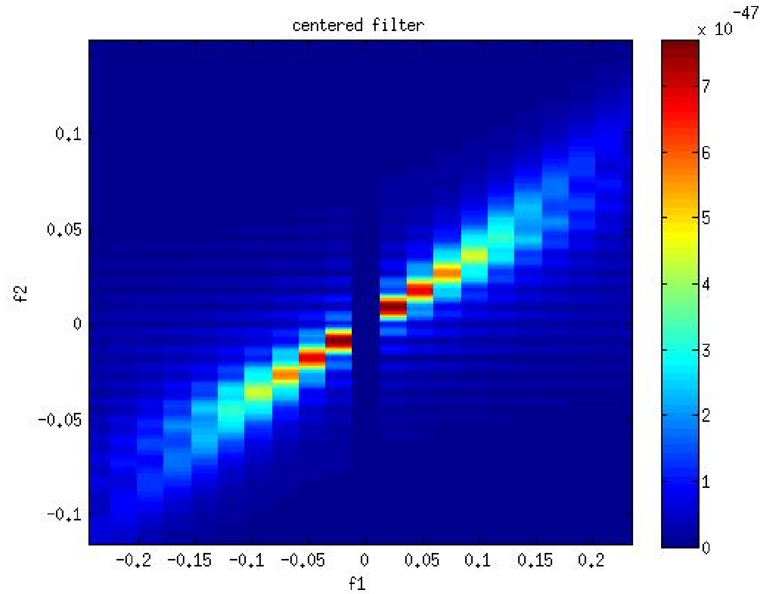

Figure 2: Power spectrum of simulated signal. Forms an 'ellipse' in this space. This is the 'filter.'

because we expect to be searching for r-modes at high initial frequencies, 900 Hz, 1000 Hz, etc. So this factor of 1000 comes from the fact that the amplitude of r-mode oscillations is related to the cube of the frequency. However, the time/frequency behavior depends on the initial frequency, so increasing the amplitudes does not capture the true variation of the signal's frequency in time.

4 Applying the filter to all peaks

As stated before, the filter will work best on continuous data, therefore we will apply it to the time/frequency map that contains all peaks (t/f points with certain amplitudes). That is, we are not selecting local maxes above a certain threshold; we are using all the data. Let X represent the original t/f image with all peaks, and M represent the model of the signal (fig. 1). The following equations from Sergio's 'Analisi dei Segnali' (p.230-236) are applied:

$$Y = F^{-1}(F(X) \times F(M)^*) \tag{6}$$

$$X' = F^{-1}(F(Y \times |Y|^3)F(M)) \tag{7}$$

X is the image on the left-hand side of figure 3; X' is the on the right-hand side. * denotes the complex conjugated of $F(M)$

Note that the injection does not need to have the same parameters as the r-mode injections in the simulated data.

4.1 r-mode injection parameters in real data

1. f_0=94.3377 Hz
2. α=1;
3. $\dot{f}_0 = 1\times10^{-7}$ Hz/s (initial spindown, but changes very little for this signal)
4. t_0= 0.094814 days
5. t_f=4.5511 days

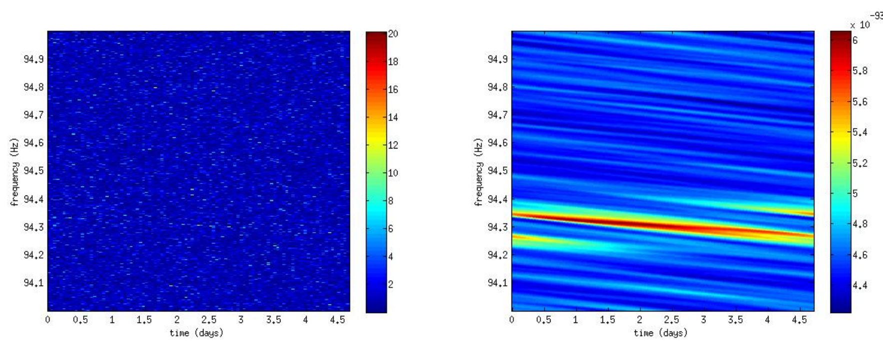

Figure 3: t/f map before (left) and after (right) filter is applied. The red region of the image on the right is where the signal was injected.

5 Creating the new peakmap

The enhanced t/f map contains a lot of unnecessary data, so we would still like to use the 'peakmap' in further analysis. We will hence multiply the enhanced t/f map with the original peakmap, but first we need to put the amplitudes of each map on the same level. So we normalize the amplitudes of the t/f map such that the maximum amplitude in the peakmap is the same as the maximum amplitude in the t/f map. Additionally, we smooth the normalized amplitudes from the enhanced t/f map with a sigmoid function to make the signal more continuous. Now, we multiply the enhanced, normalized, smoothed t/f map with the original peakmap (adjusting the simulation parameters nfft and nover such that the sizes of these two maps are equal). See figures 4 and 5 for a comparison of the original peakmap to the new peakmap and a comparison of critical ratio projections onto the frequency axis respectively.

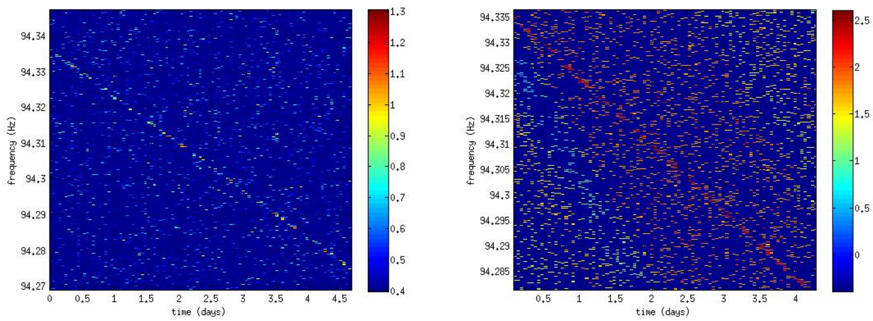

Figure 4: Peakmaps before (left) and after (right) the filter. These are log plots of the critical ratios.

6 Another way to filter

Instead of combining the noise and signals, and creating fig. 1, we can also take the 2D FFT of the signals t/f map and noise t/f map separately, then divide the power spectrum of the signals by the power spectrum of the noise. We can then multiply this ratio by the t/f map containing all the peaks, and subsequently multiply this enhanced t/f map with the original peakmap.

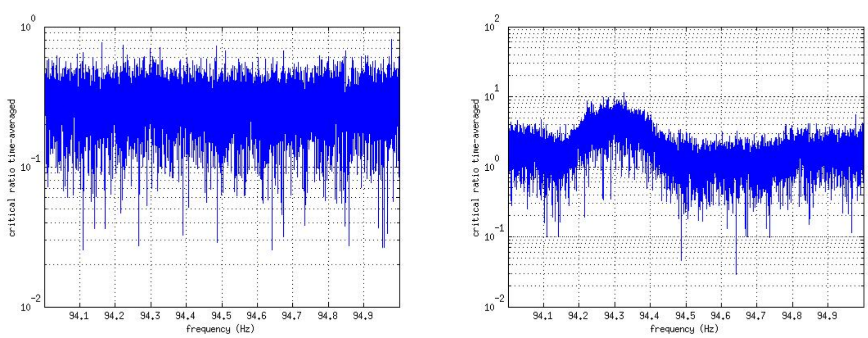

Figure 5: Projections of critical ratios onto frequencies before (left) and after (right) the filter. The injected signal began at 94.3364 Hz and spun down to as little as 94.2778 Hz, and lasted for 4.45 days. This is just to visualize the enhancement.

7 Quantifying effectiveness of the filter

Unfortunately, the power due to the injected signal is spread across many frequency bins, so it is not clear yet how we can estimate an improvement in SNR. However, we can determine how 'enhanced' the image is, meaning how do the critical ratios around the injection compare to critical ratios due to noise before and after the filter? For the peakmaps before and after the filter, we extract a submatrix that contains the critical ratios corresponding to the duration of the signal and the maximum change in frequency. We then find the maximum value of the critical ratio in each matrix. Then, we zero the corresponding entries in the original peakmap matrix that correspond to the injection, and then calculate the mean of the matrix that only contains critical ratios due to noise. Let S be either the old or new submatrix, and P the old or new peakmap after the injection has been taken out. Formulas follow:

$$CR_{max} = max(max(S)) \tag{8}$$

$$CR_{mean} = mean(mean(P)) \tag{9}$$

$$R = \frac{CR_{max}}{CR_{mean}} \tag{10}$$

$$E = \frac{R_{new}}{R_{old}} \tag{11}$$

E is the enhancement and is the ratio of two ratios.

The enhancement is thus the ratio of the new peakmap's max/noise value to the old peakmap's max/noise value. For this injection, we calculated this enhancement factor to be between 3-5, depending on which 'template' for the r-mode signal we use. For the injection described here, $E = 3.07$.