Gravitational-Wave Data Analysis
Exercises – Day 5

Peter S. Shawhan
CGWA Summer School, June 1, 2012

Today you will need four data files: `day5dataA.txt`, `day5dataB.txt`, `day5dataC.txt`, and `day5dataD.txt` . Once again, the sampling rate for these is 16384 Hz.

1. Burst search using excess power

The data files `day5dataA.txt` and `day5dataB.txt` contain LIGO-like noise with some number of simulated burst signals embedded in them and also some glitches. The two files are supposed to represent data streams from two different detectors, but for simplicity there is no relative time delay for the signals. Your job is to find the bursts! First, let's use an "excess power" method.

a. Load `day5dataA.txt` into Matlab.

b. Create an 8^{th}-order Butterworth band-pass filter to select the frequency band 64 Hz to 128 Hz. Plot the output. Do you see any sign of a signal? Write down the approximate time(s) of any signal(s) you see, for later comparison.

c. Repeat this using the `day5dataB.txt` file. If you see a noticeable spike at the same time in both data streams, then that is evidence for it being a real GW signal.

d. Now apply a *set* of Butterworth band-pass filters to your data, covering a sequence of 64 Hz bands. That is, set up a loop so that the first band is 128–192 Hz, the second band is 192–256 Hz, etc., going up to 1024 Hz. In each case, apply the filter to the *original* data (not the previously filtered data) and plot the output. Do you see any more coincident signals? Note that some signals appear in more than one frequency band.

2. Burst search using cross-correlation

Now let's search the same data using a cross-correlation method.

a. Load both `day5dataA.txt` and `day5dataB.txt` into Matlab.

b. Calculate the cross-correlation between chunks of "A" data and "B" data over a short time window, 20 msec long. To search all the data, step through the data streams, each time calculating the cross-correlation for a 20 msec chunk of "A" with the corresponding chunk of "B". To make this go faster, you can step through the streams by offset increments of 10 msec, say, instead of one sample at a time. Store all the cross-correlation values in an array.

c. Plot the array of cross-correlation values. Do you see evidence for one or more signals?

d. Repeat this, but using time windows of 100 msec and 500 msec instead of 20 msec. Now what do you see?

e. In this cross-correlation analysis, we have not yet done anything about the fact that the noise is non-white. This is significant since the large noise at low and high frequencies is contributing to our calculated cross-correlations. To help alleviate this, apply a Butterworth band-pass filter, 64 to 2048 Hz, to the "A" and "B" data streams, and then re-do parts b, c, and d using the filtered data. What do you see now?

3. Stochastic searches

Now get copies of two more data files, `day5dataC.txt` and `day5dataD.txt`, which contain LIGO-like noise with a small common stochastic signal.

a. Load both `day5dataC.txt` and `day5dataD.txt` into Matlab.

b. Calculate the simple time-domain cross-correlation, i.e. the dot product of the two time series (without any relative time-shift). The common signal should contribute to this number – but so should the instrumental noise. To get an estimate of what size number you would expect from noise alone, calculate the *time-shifted* dot product using at least ten different time shifts of at least 100 msec. You should do a "circular shift" that wraps the data around so that you're cross-correlating the same amount of data. There is a `circshift` function which you may be able to use, or else write your own code. The RMS of those time-shifted correlation values tells you what noise alone will give on average. The power signal-to-noise ratio (SNR), then, is the ratio of the zero-lag correlation value to the RMS of the time-lagged correlation values.

c. You know the stochastic signal should be strongest at low frequencies, but the cross-correlation you calculated in part b also includes contributions from the instrumental noise at very low frequencies and at high frequencies. To help the stochastic signal stand out better, apply a Butterworth band-pass filter, with the band 40 to 200 Hz, to the "A" and "B" data streams, and then re-do part b using the filtered data. Does this improve the signal to noise ratio?

d. You can also do an optimal filter, using the knowledge that the stochastic signal (in this example, at least) has a PSD proportional to f^{-3}. You'll have to do this in the frequency domain, of course, and use the detector noise PSD like you did for the inspiral filtering. If you have time, see if you can figure out how to do this!