FFT Homework (#9)
C. DiMarzio, Northeastern University
EECE2150, Fall 2018

Assignment

Here we will learn about signals and noise in the time domain and frequency domain, along with issues like sampling aliasing, and windowing. You will need to download files I wrote; FFTAXIS, FFTAXISSHIFT, and CXPLOT, and maybe PLOTPAIR.

Download the file for quiz 10, and become familiar with it.

Submit a single .pdf file with the results. You can generate a .pdf file by typing at the Matlab command line,

Make use of CXPLOT, and/or PLOTPAIR, to produce results that are easy to read.

$publish('YourFileName.m','pdf');$

This will generate a file called YourFileName.pdf in a directory ./hmtl. Don't ask why that directory.

1. First, apply a filter to remove the noise. Look at the filtered signal in the time domain. After you build a filter that works, then increase the level of the noise (the value that muliplies *randn* and see how far you can go before the filter doesn't work. Look at the input and output signals of the filter in both time and frequency domains.

2. The signal has a "drift" toward higher voltages with time that is probably not real. Try to remove it with a filter.

3. Very near the fetal heartrate of 2 Hz you can see the second harmonic of the maternal heartrate. It is quite likely that the fetal signal will be weaker than the maternal one, so finding the fetal heartrate could be difficult. The issue is that signals at these two frequencies look very similar up to a time of 10 seconds. Extend the time window and see if you can more clearly separate these two signals.

Hints: Housekeeping

You will collect a lot of figures doing this, and it will get very confusing. You can do the following at the command line.

$clear; close\ all;$

The above line clears all variables in the workspace and closes all figures. Clearing is a good idea because it removes any variables you may have set and forgotten. Your script will run “clean.” Closing the figures cleans up a lot of mess. I usually run my script from the command line as follows, but you can just hit the “run” button if you like.

$clear; close\ all; YourFileName;$

Hints: FFT

Matlab’s fft produces the spectrum so that the first point is at zero frequency, and the last point is at the inverse of the spacing of the time points. Remember that the spectrum is periodic. The highest–frequency half of the spectrum “belongs” at negative frequencies.

fftshift does the required shift to put the negative frequency contributions in the right place.

Matlab doesn’t provide a way to generate the frequency axis from the time axis. I wrote fftaxis to do this. The axis also needs to be shifted. fftaxisshift does this. So, a Fourier Transform looks like.

$V = fftshift(fft(v)) * max(t)/length(t);$
$f = fftaxisshift(fftaxis(t));$

Hints: Picking out Data

You can get the index of a particular value (or values) in an array of data using the FIND function in Matlab.

$moose = find(f == 400);$

will return the index of all f that are equal to 400. You can write complicated logic statements if you want.

Having done the above,

$V400 = V(moose)$

will return the V associated with $f = 400$.

If there isn't an exact match, an empty list, [], will be returned. You can get pretty close by using

$moose = find(f > 400, 1);$

which finds the first time f is greater than 400. This is likely close enough. If not, you could pick $moose$ or $moose-1$, whichever is closer, or even better, you could interpolate between the two.