EE125 Project: Spectral analysis of SWellEx96 Data

(Revision 11/9; changes in RED in Part 1 question 2), 2nd paragraph of Appendix A, and 2nd to last paragraph in Appendix B)

Goal:

In this project you will use perform spectral analysis on measured data from an ocean acoustics experiment.

Background:

In 1996, the Scripps Institute of Oceanography carried out the SWellEx96 experiment near San Diego. In this experiment, a research ship towed shallow and deep sources that broadcast a series of tones at different frequencies. Data were recorded on several different arrays of hydrophones (underwater microphones) and oceanographic data was also collected. The data are available on-line (http://www.mpl.ucsd.edu/swellex96/index.htm) and have been used to test a variety of different signal processing approaches.

In this project you will process data from the first sensor in the vertical line array (VLA), from event J59. At the beginning of this event, only the research ship was near the array, but after about 20 minutes a merchant ship travelled past the array. This gives a chance to look at different types of signals.

Data:

The two Matlab data files are on Trunk. The main data file is Event59Data.mat. The other file, Event59Data_20min.mat, is something you may want to look at for the last part of the project (if you are working on a computer with a reasonable amount of memory).

For the analysis, we will focus on a pair of transmitted tones, which were roughly at 198 Hz and 201 Hz. The sampling rate for the data is 1500 Hz.

In the analysis, we will use just two basic window types: rectangular and Hanning. In practice, the rectangular window would rarely be used for reasons we have discussed in class, but it is a useful point of comparison. The Hanning window is a reasonable starting point, though the Kaiser window is more flexible.

Part 1: DFT-based spectral analysis

1) Start by reading in the file 'Event59Data'. You should see a variable xstart; this is a time series from the array for the start of the event, and is the data for this part of the project.
2) Use the FFT to perform spectral analysis for this signal. Start by looking at relatively short rectangular and Hanning windows, as follows:
    a. Calculate a 512-point FFT for the first 512 points of the data, with a rectangular window. Do the same for a 512-point Hanning window (useful Matlab routines: fft, boxcar, hanning).

b. Make a single plot showing the magnitude-squared**, in dB,** of the two resulting spectra vs. frequency in Hz. Plot only the 256 non-negative frequencies. Remember that the frequency spacing in Hz can be found from the sampling frequency and the FFT length.
c. For the 512-sample window length, can you see there are two tones present near 200 Hz (**at 198 and 201 Hz**)? Do a calculation that shows whether or not you should expect to be able to resolve these tones, and include it in your report. Remembering that the mainlobe width is a conservative estimate of frequency resolution, calculate the resolution (in Hz) you should expect for a 512 point rectangular window and Fs=1500 Hz.

3) Repeat the analysis and plots above, but increase the window length to 2048 points. How do your results change?

4) In lecture we talked briefly about how using finer frequency resolution can help to improve SNR on tonal signals (this is related to the concept of noise-equivalent bandwidth in the Harris paper). The concept is that the FFT bin corresponds to a range of frequencies, so that when noise is present, a wider mainlobe (wider noise-equivalent bandwidth) means that more noise energy will be summed into each bin. Because the tone's energy is concentrated into a very narrow bandwidth, the signal contribution will remain roughly constant as the resolution is improved.

   Redo the step above, but using a longer window length (say, 8192). Do you see evidence that SNR is improved as compared to a 2048 point window?

5) When we defined the periodogram, we carefully considered normalizations so that the estimate signal amplitude would *not* change when we changed the window type. As discussed in class notes and textbook, we did this by normalizing by a factor L*U = sum(w(n).^2), where w(n) are the values of the L-length window.

   When we do simple FFT-based spectrum estimation, as in the plots you have generated so far, we do not have any normalization. Redo your plots for the rectangular and Hanning windows, for 8192 points, and make a correction so the amplitudes estimated from the two windows should match.

Part 2: Periodogram-based spectral analysis

As we discussed in class, the simple DFT/FFT-based approach above is mostly appropriate for deterministic signals. While the projected tones in the SWellEx experiment are deterministic, the background noise is definitely random. You may have noticed a fair amount of fluctuation in the plots below. In this section of the project, you will use averaged periodograms to estimate the signal.

When doing so, you can use the Matlab function 'pwelch', which implements Welch's method for modified periodogram averaging. Read the help on this function carefully. The main input arguments are 1) the data vector, 2) the window for a single periodogram, for example hanning(512); 3) overlap in samples, 4) FFT length, and 5) sample rate.

For example, you can implement the Bartlett method (no overlap, rectangular windows):

```
pwelch(data,boxcar(NFFT),0,NFFT,Fs);
```

Now, do the following:

1) Calculate and plot the results for zero-overlap with both the rectangular window (i.e., Bartlett method) and zero-overlap with Hanning window, for a window length of L=4192. Make sure the two plots show power on the same dB scale.

   Comment on the plots in terms of a) differences between the rectangular and Hanning windows and b) any reduction in variance you see as compared to the un-averaged plots you generated in Part 1.

2) As we discussed at length, overlapping windows during periodogram averaging can help to further improve the variance. For the Hanning window case and length of L=4192, repeat the calculation and plots you did above, but with overlaps of 50% and 75%. Can you visually observe a reduction in variance? How does the change compare to the change you saw when you moved from no averaging to non-overlapped averaging?

   The book describes a quality factor which relates the mean value of the spectrum to the variance. Because averaging does not change the mean, this quality factor basically measures variance reduction. Using formulas from the text, estimate the quality factor improvement you would expect when moving from no averaging, to non-overlapped averaging, to 50% averaging, to 75% averaging. This will require you to estimate the number of windows that can fit into the available data; remember to account for overlapping when doing the calculation.

   Compute what you expect the quality factor / variance reduction to be for the various periodograms you have generated (see Appendix A).

   Then, roughly estimate the variance in your plots (see Appendix B). A common mistake in estimating noise is to just compute the variance of the entire spectrum; please explain why this is not a good idea. Are your variance estimates consistent with the calculated quality factors? An exact numerical match is unlikely, but the trends should at least match each other.

3) As noted above, about 20 minutes into the event a large merchant ship steamed by the array (if you are interested, radar tracks are on the website listed above). In the data file you have loaded, there is a variable 'xlater' with data from this time period. Pick one of the periodogram settings from above and calculate the periodogram for this new time period.

What new features do you see? As background, noise from large ships contains both broadband energy (turbulence over the propeller) and tonal components (vibration from rotating machinery). These tonal components are often harmonics in frequency or otherwise linked (for example, a large motor may drive another machine via a gear).

Part 3: Spectrogram

Since the scene is changing over time, use of the spectrogram (time-varying Fourier transform) is clearly interesting.

Here you can use the Matlab function 'spectrogram'. Read the help on this function carefully. The main input arguments are actually very similar to those for pwelch. Remember that for display purposes, a high overlap is often useful. When plotting spectrogram results, you will want to use the 'colorbar' command to show the dB range of the signal. Furthermore, it will be useful to adjust the various plots to have the same dB range, for easy comparison; this can be done using the 'caxis' command (for example, the commands "caxis([*a* *b*]); colorbar" will set the dB scale to run from *a* to *b* , where *a* and *b* are the lower and upper range you specify.

For similar parameters as your spectrogram (M=4192, Hanning window, 75% overlap), calculate and plot the spectrograms for the data samples 'xstart' and 'xlater'. What differences do you observe? Zoom in on some of the new, higher frequencies seen when the ship is present. Describe how the what new information is provided by the time-dependent transform. If you zoom in on some of the spectral lines, the spectrogram should get more interesting. In that experiment, there are some tones that were transmitted by a source that are quite loud and stable, but there are others which are the ship's own radiated noise that wobble a bit more.

Communications channels, like the underwater channel, often have multipath which can cause "fading", or destructive interference as the signal arrives along different paths. We discussed this a long time ago, in the first lecture on $H(\omega)$. Can you see examples of multipath fading in this data?

If your memory has a reasonable amount of memory, load the larger data file 'Event59Data_20min.mat' and form the spectrogram of the data contained in it. This data basically covers the full time period from the start of the track up to the closest approach of the commercial ship. Comment on what you see happening over this longer time scale.

APPENDIX A: Calculating periodogram quality Factor using results from Harris' paper

In the textbook, Proakis and Manolakis define a quality factor Q for periodograms, which is the ratio of the mean of the spectrum estimate (squared) to its variance (eq. 14.2.41). For reasons discussed in the text and below, the quality factor basically measures how much the variance is reduced for a particular amount averaging and window type. P&M present results for several windows, but not the Hanning window you are using in this project.

In a classic paper, Harris tabulated a variety of metrics for many different windows used in spectrum analysis. After translating the notation, these results can used to find Q. The results from Harris we will use are:

- Eq. 18, which shows how variance is reduced due to averaging non-overlapped windows (valid for any window type).
- Eq. 19, which gives results for 50% and 75% overlap
- Table 1, which lists (last 2 columns) the overlap correlations needed in Eq. 19, **i.e. c(0.75) and c(0.5)** Note that the standard Matlab 'hanning' window corresponds to Harris' Hanning window with alpha=**2**

To use Harris' results, we need to translate the notation. Eq. 18 and 19 are in terms of $\frac{\sigma_{AVG}^2}{\sigma_{MEAS}^2}$, which for convenience we can call 'F'. σ_{AVG}^2 is the variance of the averaged periodogram, and is the same as the textbook's quantity $var[P_{xx}^A(f)]$. σ_{MEAS}^2 is the variance of a single (unaveraged) window, or what the textbook would call $var[P_{xx}(f)]$, as in Eq. 14.2.45.

We are normalizing the periodograms so the windowing function doesn't change the energy. This means that, for any window type the expected value of the periodogram goes to the true value as we increase the window length (for example, P&M Eq. 14.2.57):

$$E[P_{xx}^A(f)] \to \Gamma_{xx}(f)$$

For the case of no averaging, the textbook states (Eq.14.2.45) that for long windows

$$var[P_{xx}(f)] = \sigma_{MEAS}^2 \to \Gamma_{xx}^2(f)$$

Putting this all together, we can find that in the limit of increasing window length,

$$Q = \frac{E[P_{xx}^A(f)]^2}{var[P_{xx}^A(f)]} \to \frac{\Gamma_{xx}^2(f)}{F\Gamma_{xx}^2(f)} = \frac{1}{F}$$

Based on this equation, Q is just 1/ (the result from Eq. 18 or 19).

We can verify this by comparing to a few examples from the book:

- For no overlap, Harris Eq. 18 would give F=1/K, where K is the number of windows. This matches P&M Eq. 14.2.58, where 'L' is the number of windows (P&M use 'L' for denote the number of windows for the Welch method, where windows can be overlapped, and 'K' for methods where there is no overlap).
- For 50% overlap and triangular windows, Harris Eq. 19 gives F = 9/(8K) (ignoring the 1/K^2 terms, as discussed in the text below Eq. 19). This matches P&M Eq. 14.2.58.

A few more notes about the Harris paper:

- Section IV A describes the equivalent noise bandwidth metric; this is a more detailed analysis of why longer FFT windows improve SNR for tonal signals.
- Section IV D describes scalloping loss, which is something we didn't explore in class but is a useful concept.

APPENDIX B: Estimating noise in the spectrum

Estimating variance is quite easy in simulated data (since noise power is characterized by its variance, we can just us a command to compute variance). In real data, we get a mix of both random noise and deterministic signals, so we somehow need to exclude the deterministic signals. Even if you think you have picked a portion of the spectrum where there is only random noise, may be some deterministic signals buried in the "noise" portion that make the statistics skewed. Those give outliers which will cause simple calculations like 'var' or 'mean' to be distorted.

The project asks you to estimate the variance in your periodograms from the data. Doing this accurately can be quite challenging, but a quick method which will get you most of the credit is:

- find a portion of the spectrum where there aren't tones - so the data appears to be mostly broad band noise.

- estimate by eye the amount of variation you see. Remember that >99% of the variation you see will fall within +/- 3 standard deviations; you may be able to visually estimate the variance by remembering that. Also, remember that you are probably looking at plots in dB, and the statement above about standard deviations makes sense only for linear values. Thus, for this task only, it may make more sense to plot the spectra on linear scale.

For a more quantitative answer, you can read up on robust statistics. Measures like the median and interquartile range (IQR) are much more robustly estimated than the mean and standard deviation; Wikipedia's page **on IQR** shows a conversion from IQR to standard deviation, for Gaussian noise. Thus, a more robust approach to finding variance is to a) find IQR, and b) map that to variance.

I'd prefer to see you write your own version, but the code below is a very nice example of this kind of approach. may be interesting to just read the description of what it does.

https://www.mathworks.com/matlabcentral/fileexchange/16683-estimatenoise/content/estimatenoise.m