COMP ENG 4TL4 – DIGITAL SIGNAL PROCESSING

Lab #5: Spectral Analysis and Estimation

Objective:

To gain experience in spectral analysis of time-varying signals and spectral estimation of stationary random signals within MATLAB.

Assessment:

- Your grade for this lab will be based on your ability to use spectral analysis and estimation tools from the MATLAB Signal Processing Toolbox to analyze time-varying speech signals and estimate the power spectral density of stationary random signals, and on your reporting of the results. The report should contain any mathematical calculations or derivations carried out, MATLAB plots of results with brief descriptions, the MATLAB code with which you obtained your results, and answers to specific questions below.
- Clearly label all plots and their axes (points for style will be deducted otherwise)
- DON'T COPY OTHERS BLINDLY!!
- Please attend the lab section to which you have been assigned.
- You should complete this lab with one lab partner. If there are an odd number of students, then one group of three will be created by the TA.
- Each pair of students should complete one lab report together, which is to be submitted one week from the date of the lab. Those students in the "At Home" section must submit their reports one week from the day on which they demonstrated their lab to Jeff Bondy.

Pre-lab:

- Carefully read through this lab description, so that you know what is required.
- Read through the lecture notes (and bring them with you) so that you know how to answer the questions.
- Familiarize yourself with the MATLAB commands that may be required for this lab – see the list at the end of this lab description for some hints.

1. Spectrogram analysis of speech signals:

a. Load the speech signals `oilyrag.wav` and `defineit.wav`.

b. Given the sampling frequency for `oilyrag.wav` and `defineit.wav`, work out the window lengths required to create a *wideband spectrogram* with a window duration of 4 ms and a *narrowband spectrogram* with a window duration of 32 ms.

c. Generate wideband *and* narrowband spectrograms for both `oilyrag.wav` and `defineit.wav` using the MATLAB Signal Processing Toolbox function `specgram()`.

d. Some parameters of `specgram()` that you will need to play around with are `NFFT` and `NOVERLAP`. See what effect these parameters have on the spectrograms, and see if you can find suitable values that give a spectrogram for `defineit.wav` that looks like the spectrogram on slide #18 in the notes from Lecture #27.
Note that you can use the functions `caxis()` and `colormap()` to adjust the appearance of your spectrogram.

e. From the spectrograms, work out and note down approximate start and end times for *each phoneme* in the sentences.
Hint #1: You should be able to work out the type of excitation source (voiced, fricative or plosive) from the spectrograms.
Hint #2: Vowels (*a*, *e*, *i*, *o* and *u*) have a fairly stationary spectrum over their duration, whereas other voiced sounds (such as *m*, *n, w*, *r* and *l*) will have a changing spectrum over their duration.
Hint #3: When you think you have the correct start and stop time for a phoneme, listen to that segment of the speech *alone* using `soundsc()`. Note, however, that some phonemes will be hard to identify when taken out of context, i.e., when not heard in their correct place within the entire sentence.

f. Compare and contrast the representation of the different phonemes in the wideband and narrowband spectrograms. Which features are better represented in the wideband spectrogram, and which are better represented in the narrowband spectrogram?

2. Spectral estimation of stationary random signals:

a. Load the random signals `bandnoise.wav`.

b. Use the MATLAB Signal Processing Toolbox function `periodogram()` to compute and plot an estimate of the power spectral density (PSD) of the random process that generated this random discrete-time sequence.

c. From the periodogram generated in part b above, try to estimate the center frequency (in units of `radians`) of this bandpass noise sequence.

d. Now, use the MATLAB Signal Processing Toolbox function `pwelch()` to compute and plot an estimate of the PSD. Try varying the window length and the amount of overlap of the windows, and show what effect it has on the estimate.

e. From the Welch periodograms generated in part d above, can you improve your estimate of the center frequency of the bandpass noise (from part c)?

3. Spectral estimation of fricative speech sounds:

a. From section 1e above, extract the waveform segments corresponding to the 's' from 'ask' in `oilyrag.wav` and the 'f' from 'define' in `defineit.wav`. Plot the waveforms to check that they are indeed fricative-looking waveforms and that they are fairly stationary over their durations.

b. Plot the wideband spectrograms of each of these extracted phonemes. From the spectrograms of the two fricatives, 's' and 'f', how well can you tell them apart?

c. Because fricatives are noisy signals, we can use periodograms to get a better prediction of the frequency response of the vocal tract for the two different fricatives, 's' and 'f'. First compute and plot the standard periodograms of each of these fricatives, then try to estimate the frequencies of the spectral peaks (resonances) in the vocal tract frequency response for each.

d. Next, compute and plot the Welch periodograms for each fricative. How do your estimates of the vocal tract resonance frequencies using this method compare to those from the standard periodogram?

Potentially useful MATLAB commands

Note that this is not an exhaustive list! You are not required to incorporate all of these in your scripts.

help <topic>	helpwin	figure	plot	stem
subplot	hold	xlabel	ylabel	legend
title	function	clear	close	clc
specgram	periodogram	pwelch	colormap	caxis
wavread	soundsc	zoom	ginput	max