UCLA
Dept. of Electrical and Electrical Engineering
ECE 114, Fall 2019
Computer Assignment 4: Spectral Analysis of Speech

Introduction: This assignment studies the use of time-frequency representations, or spectrograms, during speech processing. Furthermore, the assignment explores the role of linear prediction in spectral analysis of speech. Specifically, this assignment focusses on the effect of prediction order on resulting spectral approximations.

Spectrograms:

Time-frequency representations of speech are commonly referred to as spectrograms. MATLAB includes the function `specgram`, which divides data into a series of overlapping windows, and computes the FFT for each signal segment. In the resulting 2D signal, the vertical axis represents frequency, and the horizontal axis represents time. The log-magnitude of each FFT is mapped to color, with red corresponding to large values and blue corresponding to small values. The variables loaded into your workspace includes "tone," which is a sinusoid of constant frequency. You can listen to "tone" by typing:

```
soundsc(tone,8000);
```

You can then view the spectrogram of "tone" by typing:

```
specgram(tone);
```

Repeat the previous steps for the signal "sweep," which is a sweep of frequencies between 0 Hz, and 4000 Hz.

The full `specgram` function takes the form:

```
specgram(signal,NFFT,Fs,L,NOL);
```

The default settings for the `specgram` function are set to $L = 256$-point analysis windows, $N_{\text{FFT}} = 256$-point FFTs, and $N_{\text{OL}} = 128$ sample overlaps between successive frames. Note that $N_{\text{FFT}} \geq L$ must hold in order to avoid aliasing in the time domain. Also, N_{OL} must be less than L.

Experiment with the above parameters for the signals “tone” and “sweep” to see the effects on the resulting spectrograms. Also, experiment with the signal “noise,” which contains white noise.

Questions:

1. Describe the spectrograms of the signals “tone,” “sweep,” and “noise” using the default settings.
2. If $L = 256$-point analysis windows are used with $N_{\mathrm{OL}} = 240$ sample overlaps, what is the length of the window increment (“hop”) in seconds? As a function of the variables N_{OL} and L, determine the rate at which successive FFTs are calculated in Hz (FFTs/second).

Analyze the signal “male_sentence” using the `specgram` function. With the FFT size set to $N_{\mathrm{FFT}} = 512$, view the spectrogram for the following parameter sets (L, N_{OL}):

$$(480, 360); \quad (240, 180); \quad (120, 90); \quad (60, 45);$$

Questions:

1. When the window length is long (i.e., $L = 480$), what do the horizontal striations during voiced sounds correspond to?
2. When the window length is short (i.e., $L = 60$), why do vertical striations occur in the spectrogram?
3. Of the previous four parameter sets, which set results in the spectrogram which is best for estimating the formant frequencies for the male sentence?

Linear Prediction Order:
We now focus on LPC analysis with specific attention to LPC order. Load the necessary speech data into your workspace by typing:

```
load_in;
```

at the prompt. Analyze the male /a/ segment using the `zpfft` script:

```
zpfft(male_a, 8000, 1);
```

Questions:

1. Does the spectral analysis provided by the `zpfft` script correspond to narrowband or wideband analysis?
2. Approximate the spectral locations of the first 3 formant frequencies.

MATLAB provides a function `lpc` which, given a signal x and predictive order p, will output the coefficients of the filter $A(z)$:

```
a=lpc(x,p);
```

Recall that $A(z)$ is the same expression that is used to approximate the spectrum in linear predictive analysis:

$$H(z) = \frac{G}{A(z)}.$$

As can be interpreted from the formulation of $A(z)$, the output of the `lpc` function is structured in the following way:

$$a = [1 \quad -a_1 \quad -a_2 \quad \ldots \quad -a_p],$$

where a_i is the ith prediction coefficient. The estimated spectral representation using LPC can be plotted by:

```
figure; [h,w]=freqz(1,a);
plot(w,20*log10(abs(h)));
```

MATLAB also provides a function to display the pole-zero plot for a given transfer function. If `num` and `den` are the vector-form numerator and denominator of a transfer function, the pole-zero plot is displayed by:

```
zplane(num,den);
```

Perform linear predictive analysis (using MATLAB) and plot the corresponding pole-zero plots for predictive orders of $p = (4, 6, 8, 10, 12)$.
Questions:

1. What is the lowest of the previously mentioned prediction orders that provides adequate information for accurate formant estimation? Estimate the spectral location of the first three formant frequencies.

2. Perform LPC analysis with $p = 100$. Comment on the resulting spectral representation. What do the fine ripples in the spectral representation correspond to?