520.445 - Audio Signal Processing
Fall 2015

Project 1
Due: 20 October 2015, 11:59pm

Part 1

The speech file *'vowel.wav'* represents a quasi-periodic vowel.

1. Plot the log-magnitude of the Fourier transform of the signal over the interval $[0, \pi]$, using a 1024-FFT. The signal should be windowed with a Hamming window of two different durations, $25ms$ and $10ms$, with the window placed, in each case at the signal center. Show the log magnitude plot for each duration.

2. From the speech waveform, estimate the period in seconds of the quasi-periodic waveform. Which spectral estimate from part (a), i.e., from a $25ms$ or from a $10ms$ window, is more suitable to pitch estimation. Propose a method of pitch estimation using the log magnitude of the Fourier transform.

Part 2

Vowels in the English language are distinctive sounds with unique spectro-temporal features. Figure 1 below shows a classic plot from Peterson and Barney (1952) that highlights the variability in vowel sounds produced by the human vocal tract. Figure 2 shows the identity of these vowels.

Write a MATLAB program to synthesize a database of the variants within the 10 vowel classes shown in Figures 1 and 2. All vowels should be at a pitch of a female voice (as close as possible to 220Hz). You are not allowed to extract vowel sounds from any database of continuous or phonemic speech. Your approach should follow the all-pole vocal tract model concept presented in class. Clearly explain your choice of approach, and show an analysis of your results demonstrating that your vowel synthesizer can generate the variability within each vowel class. Clearly reference any resources you used.

Return a MATLAB code that can generate any instance of a given vowel, in addition to a zip file with at least 20 different examples of each of the 10 vowel classes in .wav file format. Write a detailed report summarizing your approach and showing analysis of the vowels that you generated (your code should not be part of the report; but should be submitted separately)

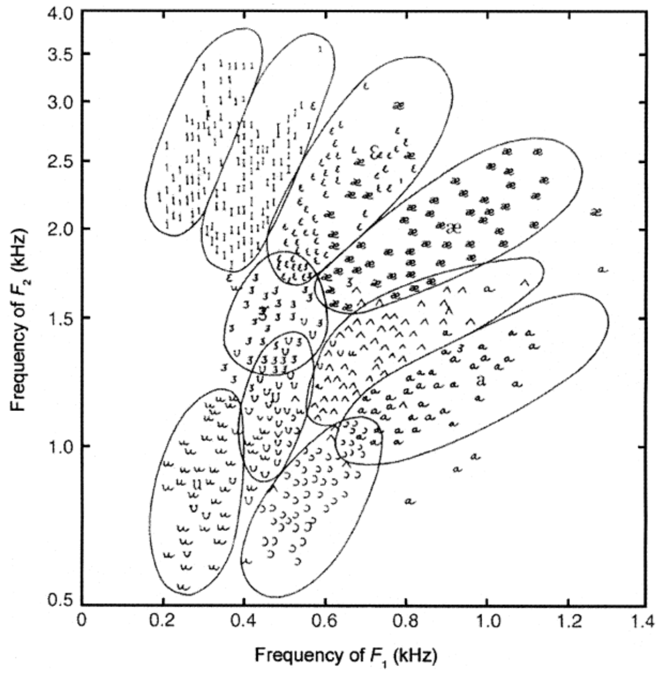

Figure 1: First and second format frequencies derived by Peterson and Barney (1952) for the vowels produced by 76 speakers. The frequency of F2 is plotted with respect to the ordinate for each vowel token, the frequency of F1 with respect to the abscissa

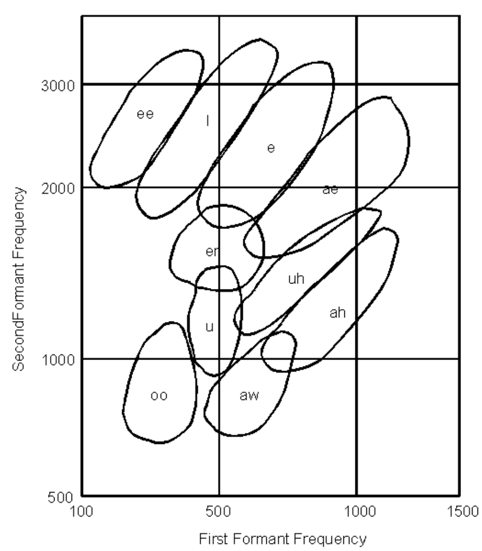

Figure 2: F2 vs. F1 formant frequency for some common vowels

Instructions

a) **NOTE:** You should make sure that your MATLAB functions are working properly and that you send the TA all the functions necessary to properly test your system. If your code crashes for any reason, you will not get credit for that part of the project.

b) You are expected to work on the project individually. You can discuss with your classmates; but each of you is expected to submit their own MATLAB code and report.

c) Please email your project to the TA <abellur1@jhu.edu>

Relevant matlab functions

Some MATLAB functions that you might find useful are:

- `help` or `doc`: find out more details about each function. e.g. Type `help cos`
- `wavread`: to read a *'.wav'* file into MATLAB (also check `audioread`)
- `wavwrite`: to write a signal into a *'.wav'* file. Be careful when using this function. If your signal is not bounded between 1 and -1; it will be clipped and therefore corrupted
- `sound`, `soundsc` or `audioplayer`: to listen to the sound audio
- `conv`: to perform the convolution operation. Make sure your input signals are aligned in time before calling the function
- `fft`: computes the Fourier-domain spectrum $X(f)$
- `ifft`: computes the inverse Fourier transform to get back a time-domain signal
- `hamming`: generates a Hamming window
- `spectrogram`: displays a time-frequency spectrogram of a signal
- `poly`: to determine the coefficients of a polynomial (all pole filter in this case) when provided with the roots.
- `filter`: filters a signal according to parameters given in the function arguments