Assignment: Homework 4

What to Hand In

1. Your code with appropriate comments and a readme file. You are responsible for making us understand and know how to run your code.
2. A report with answers, figures, and explanations.
3. Any sound files asked for in the problems.

How to Hand It In

1. Put all your files in one folder. Compress this folder and name it <firstname>_<lastname>_HW4 .zip. For example, "Zhiyao_Duan_HW4.zip".
2. Submit via blackboard to the appropriate entry. You can make multiple attempts but only the last attempt will be graded.

When to Hand It In

It is due immediately at the beginning of the day (i.e., 0:00AM) specified on the course calendar. **Late assignments will receive a grade deduction of 20% of the full grade each day.**

Problems (10 points in total)

1. (2.5 points) Linear Predictive Coding
   a. (0.5 points) Segment "male.wav" into frames with frame length of 512 points (about 32ms), hop size of 128 points, and hamming window. Perform a 20-order LPC analysis of the 125th frame of the time domain signal. Write down the LPC coefficients. You can use the "lpc" function in Matlab.
   b. (0.5 points) Calculate the error signal (i.e., the estimated excitation signal). Plot the waveform and the magnitude spectrum of the error signal. By looking at the figure, do you think that the spectral envelope is flat? Explain why.
   c. (1 point) Plot the magnitude spectrum (dB vs. Hz) of the 125th frame of the original signal, and the reconstructed spectral envelope of the LPC coefficients, using the "energy matching criterion" in the same plot. Label your figure appropriately.
   d. (0.5 points) Vary the LPC order (e.g., 10, 20, 50, 100, 200, and 500). Plot the magnitude spectrum together with the spectral envelope on the same plot for each order. Label your figure appropriately. How would you describe the change of the spectral envelope when the order is changed?

2. (3 points) Choose one of the following problems (Vowel Combination or Voice Conversion) to solve.

2a. Vowel Combination. In this problem, you will try to create a diphthong from two monophthongs, i.e., combining two vowels into a single syllable using LPC-based cross synthesis.

a. (0.5 point) Speak two vowels, /i:/ (eve) and /a:/ (father), and record them as separate files. Segment each signal into frames with an appropriate frame size and hop size (for speech, a frame size around 20 ms is often good). Choose one frame in the middle range of each vowel and perform LPC with an appropriate order. Plot the magnitude spectrum of the spectrum together with the spectral envelope calculated from the LPC coefficients (with energy matching criterion), as well as the magnitude spectrum of the excitation signal. Compare figures of the two vowels.

b. (2 points) Could you combine these two vowels to generate a diphthong /ai/ (fine)? Try your best to make it sound smooth. Comment on your generated diphthong. If you are satisfied and curious, you can repeat the process and try some other combinations, e.g., /au/ (round).

Hint: Step 1: LPC analysis on each frame of your vowels to get two sequences of excitation signals and two sequences of LPC coefficients. The LPC coefficients in different frames of the same vowel should be pretty similar and you may simply use their average in the following steps.
Step 2: For each frame, generate a weighted average between the LPC coefficients of /a:/ and those of /i:/. These will be the LPC coefficients of your generated diphthong in that frame. The weights should change over time as the diphthong changes over time.
Step 3: Use the generated LPC coefficients to filter the excitation signal of one vowel that you choose to generate the diphthong.
Step 4: Overlap-add different frames of the generated diphthong signal. Use COLA to avoid clicks.

c. (0.5 points) The generated /ai/ should sound more natural if its pitch goes down. Try to define a pitch contour and modify the pitch of the excitation signal of each frame following the contour before generating the diphthong.

Hint: Use a time-varying phase vocoder idea: Resampling the excitation signal at the frame and its previous frame to change the pitch to the target pitch; calculate the phase advance between adjacent frames; use this phase advance to calculate the phase of the current frame in the finally reconstructed excitation signal; filter the excitation signal with the time-varying filter.

2b. Voice Conversion. In this problem, you will try to convert your voice to Donald Duck's voice. Record 10 seconds of your voice speaking some funny content and name it as "voice.wav". Perform STFT and then LPC analysis of each frame of your voice. Stretch the spectral envelope of your voice along the frequency axis in each frame by a factor "alpha" (i.e., f Hz is mapped to f*alpha Hz), but do not change the excitation signal. Save your modified voice as "voice_donaldduck.wav". Does your voice sound more like Donald Duck now? Choose

appropriate parameters including sampling rate, bit depth, frame length, hop size, zero-padding factor, window function, and LPC order when you do this experiment.

Hint: Step 0: You can use Audacity (audacity.sourceforge.net) to do the recording.

(0.5 points) Step 1: LPC analysis on each frame of your voice to get the excitation signal e1(t).

(1 point) Step 2: Resample your signal with fs/alpha so the spectrum is stretched along the frequency axis (i.e., pitch shifted) with the factor of alpha, and then perform LPC in each frame to get the LPC coefficients of the modified spectral envelope H2(f). **You need to use a different frame length and hop size here, to make the frames of the resampled signal match with the frames of the original signal.**

(1 point) Step 3: Filter e1(t) with H2(f) to achieve the Donald Duck effect in the modified signal.

(0.5 points) Step 4: Overlap-add different frames of the modified signal. Use COLA to avoid clicks.

3. (4.5 points) Implement a function to calculate the cepstral coefficients of an audio file. You will then use your function to analyze the timbre of note samples of three instruments.

a. (1 point) Write a function named "myCeps" to calculate the first several cepstral coefficients of a frame of audio. The input/output specifications of the function are as follows:

```
function cc = myCeps(x, p, fftLen)
% Calculate the linear frequency cepstral coefficients by
performing discrete cosine transform (DCT) on the positive-
frequency log-magnitude spectrum or performing FFT on the full-
frequency log-magnitude spectrum of the input waveform.
%
% Input
%   - x      : one frame of audio waveform (windowed signal)
%   - p      : the number (order) of cepstral coefficients
%   - fftLen: fft length when calculating the spectrum of x
% Output
% - cc    : the calculated first p cepstral coefficients
```

b. (1 point) Take a frame in the middle part of the provided file "NoteSamples\BbClar\BbClar_mf_C4B4_1.wav", and calculate its full cepstrum (i.e. p=fftLen/2) using your myCeps function. Use frame length of 2048 points, hamming window, fft length of 2048 points in spectrum calculation. Plot the cepstrum you get. The x axis should be quefrency measured in seconds. Label the axis's appropriately. In this figure, you should be able to see several significant peaks besides the peak around 0 quefrency. The first peak tells us the fundamental period of the note. From this you can calculate the pitch of the note. Write down the x coordinate of the first peak, and calculate the pitch.

c. (2.5 points) In the NoteSamples folder, there are note samples of the full pitch range of three instruments. We can see the pitch ranges of the three instruments overlap significantly. Use your myCeps function to calculate the first 21 cepstral coefficients

of all the note samples. For each note sample, only analyze one frame in the middle of the note. Use frame length of 2048 points, hamming window, fft length of 2048 points in spectrum calculation. Then plot a figure to show the 2nd cepstral coefficient (horizontal) and the 3rd cepstral coefficient (vertical axis) of all the notes. Each note is hence represented as a dot on the plane. Use different colors and markers for different instruments. By looking at this figure, do you think cepstral coefficients are good at representing the timbre of musical instruments? Are the first three cepstral coefficients enough to classify the three instruments?
Hint: You may find “dir” function useful when processing a bunch of files.