CS578: Project 1: Linear Predictive Coding

February 28th 2020

Delivery: March 13th 2020

Questions: yannis@csd.uoc.gr, kafentz@csd.uoc.gr

During this project you will explore the Linear Prediction theory and an implementation in Matlab of a Linear Prediction based Analysis and Synthesis system for speech. In the provided Matlab code there are some empty command lines that are waiting for you to fill in. Once you do this, you can play with the code to do various speech modifications in an input speech signal.

In this project you will use the code in the Matlab file: `lpc_as_toyou.m`. You will play with a speech signal in a wav format `H.22.16k.wav`.

More specifically:

1. Analysis-Synthesis based on Linear Prediction

   Download the Matlab code `lpc_as_toyou.m`

   At this stage, the lines require your interventions are 44-47.

   - In 44, you require to compute the autocorrelation function of a given speech frame defined in line 40. You may use the Matlab function **xcorr**.
   - In line 45 you must compute the autocorrelation coefficients using the Levinson recursion. In Matlab, there are the functions **lpc** and **levinson**. Please develop your own functions.
   - In line 46 you must compute the gain of the LP filter, as discussed in lectures.
   - In 47, you must compute the linear prediction error, by inverse filtering the speech signal through the estimated linear prediction filter. You may use the Matlab function **filter**.

   Once done, you must be able to load the wav file, perform analysis and synthesis frame-by-frame. At the end you can save your computed speech signal (variable **out** which is the output from the above Matlab file. To save it, you may use the command **wavwrite**). Listen to the original and the processed speech signal using the Matlab command **soundsc** or **sound**.

   Write down very briefly your listening impressions.

2. Follow frame-by-frame the analysis procedure

   In each analysis frame, compare the magnitude of the frequency response of the estimated linear prediction filter (using Matlab command **freqz**) with the magnitude of the Fourier Transform (Matlab command **fft**) of the speech frame.

   - Plot and compare on the same figure (use **hold on** and **hold off** - use different colors) the two magnitude spectra. Do the above in both voiced and unvoiced areas.
   - Change the order of the Linear Prediction towards both directions (by increasing and decreasing the initial value (line 24: 24(!)))

   Save some interesting - according to you - plots and write down briefly your comments/observations.

3. Make modifications in the excitation signal

   - Create a whisper voice

     Change the excitation signal (line 50) so that the synthesized output sounds like a whisper voice. For this, you may want to use the Matlab command **randn**. Do the above for a lower and a higher linear prediction order than the one provided (24).

     Save the generated whispered speech files and write down briefly your comments/observations while explain shortly your choices.

   - Create a robot voice

     Change the excitation signal (line 50) so that the synthesized output sounds like a robot voice that uses a constant pitch period (it is up to you to define that specific pitch period). For this, you may want to use the Matlab command **ones** and **zeros**. Do the above for a lower and a higher linear prediction order than the one provided (24).

     Save the generated robot speech files and write down briefly your comments/observations while explain shortly your choices.

4. Make modifications in the vocal tract

   In each frame, you can compute the poles of the estimated linear prediction filter using the Matlab command **roots**. Select the most significant (according to their magnitude) three poles which we may assume they correspond to the first three formants. Check that these selected poles have frequencies close to the expected formants (for instance, by viewing the magnitude spectrum of the Fourier transform of the speech frame). Modify accordingly these formants so that you can get an elderly and a younger voice than the input speaker. To get back the filter once you have modified the poles, you may use the Matlab command **conv**.

Save the generated speech files and write down briefly your comments/observations while explain shortly your choices.

5. Give us your voice

   Record a speech signal using your voice and choose one of the above modifications to be applied on your speech signal. You may use $f_s = 16000$ as sampling frequency during your recording.

   Save both signals and include them in your deliverables.

Answers may be given in Greek or in English. Return the functions you wrote by yourself plus the original (initial) Matlab file with the requested lines filled in, along with a report that discusses your observations.