Project Report for EC516 Project Assignment 02 (Fall 2018)

Author: Yuzhou Cui (cuicyz@bu.edu)
Coworker: Kaiyu Yang, Yangdi Li
Date: 2018. Dec. 6

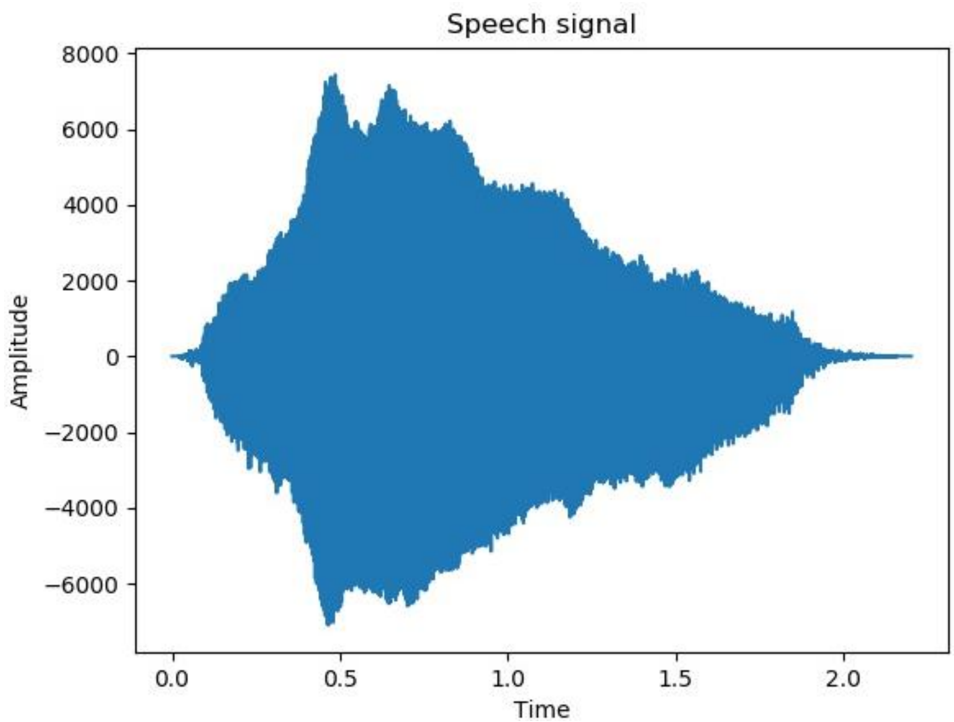
1. Assignment

1.1 (3 points) Record and display a speech signal. Select a windowed interval of approximately 20 ms within the speech signal such that the speech signal within the interval has an approximately periodic structure. Compute a DFT of the interval and display the log-magnitude of the DFT of the selected interval.
1.2 (4 points) For the interval selected in the previous part, compute an order 10 indirect least-squares model. Plot the log-magnitude of the frequency response (appropriately sampled in frequency) corresponding to the model you computed for the selected interval.
1.3 (3 points) Discuss the similarities and differences between the log-magnitude plots you obtained in the project.

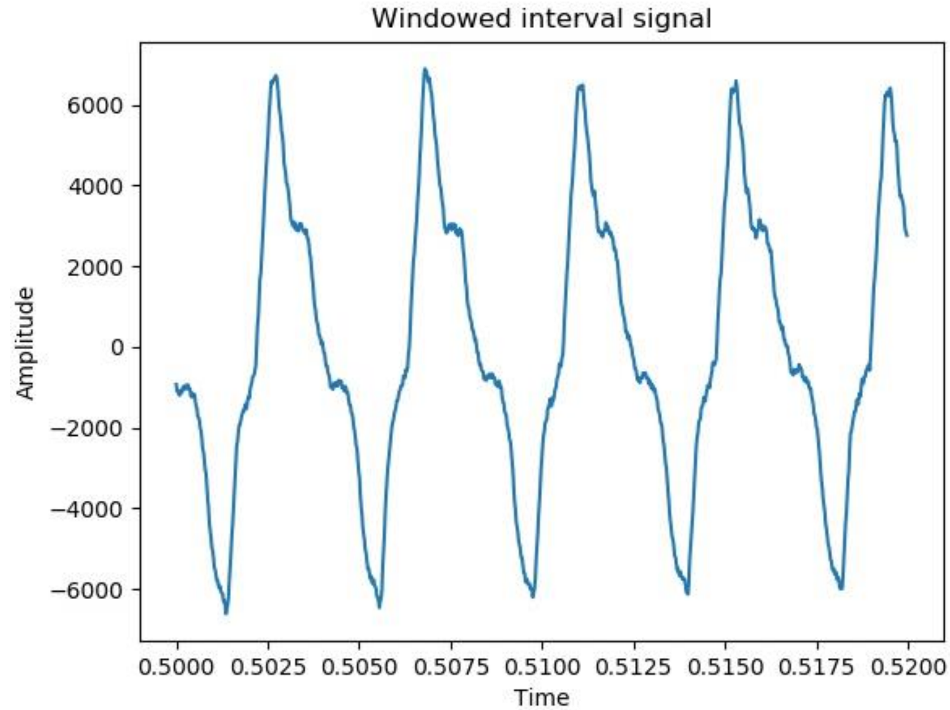
2. The process and the result of the project

2.1 Record and display a speech signal. Display the log-magnitude of the DFT of the selected interval.

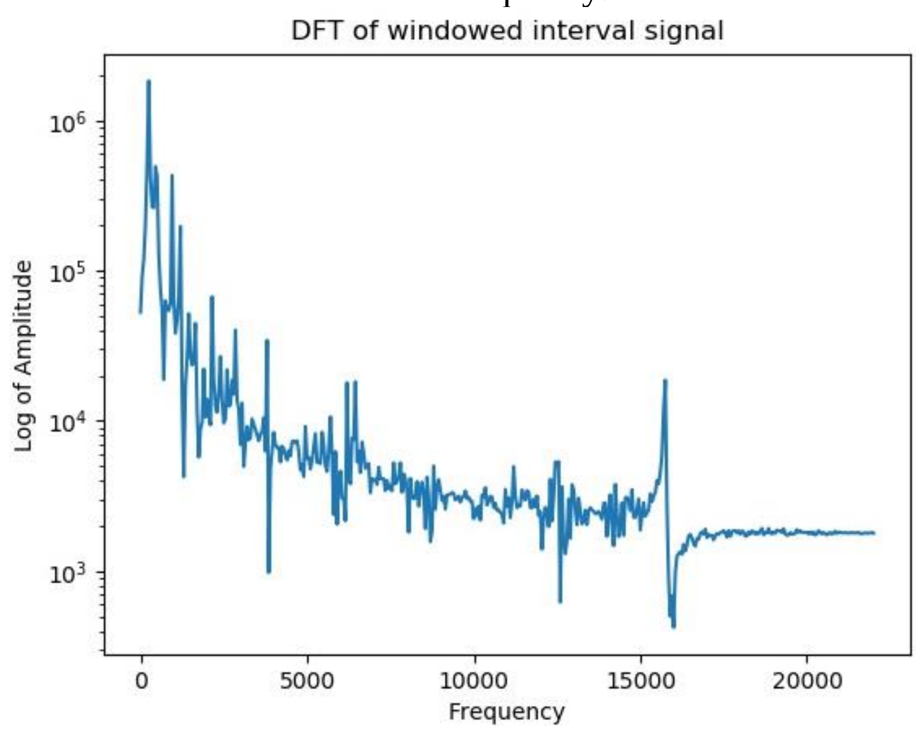
First, I read the sample rate and the amplitude of the wav file using scipy.io in python. The sample rate for this project is 44100Hz which is very important in the project.



Then, I intercepted a 20 ms signal with great periodic character showed in the picture below.



After that, I did DFT using function fft in numpy. The result is showed below. The figure in the frequency domain is symmetry so I just displayed the first half of it. We can see that the main part of the window interval signal located in 0Hz to 5000Hz which conform to human voice frequency.



2.2 Compute an order 10 indirect least-squares model and plot the log-magnitude of the frequency response.

To compute frequency response, we first need r[n]. In python, we can figure out r[n] using a simple loop function based on the equation showed below.

$$r[m]=\sum_{n=0}^{\infty} g[n]g[n+m]$$

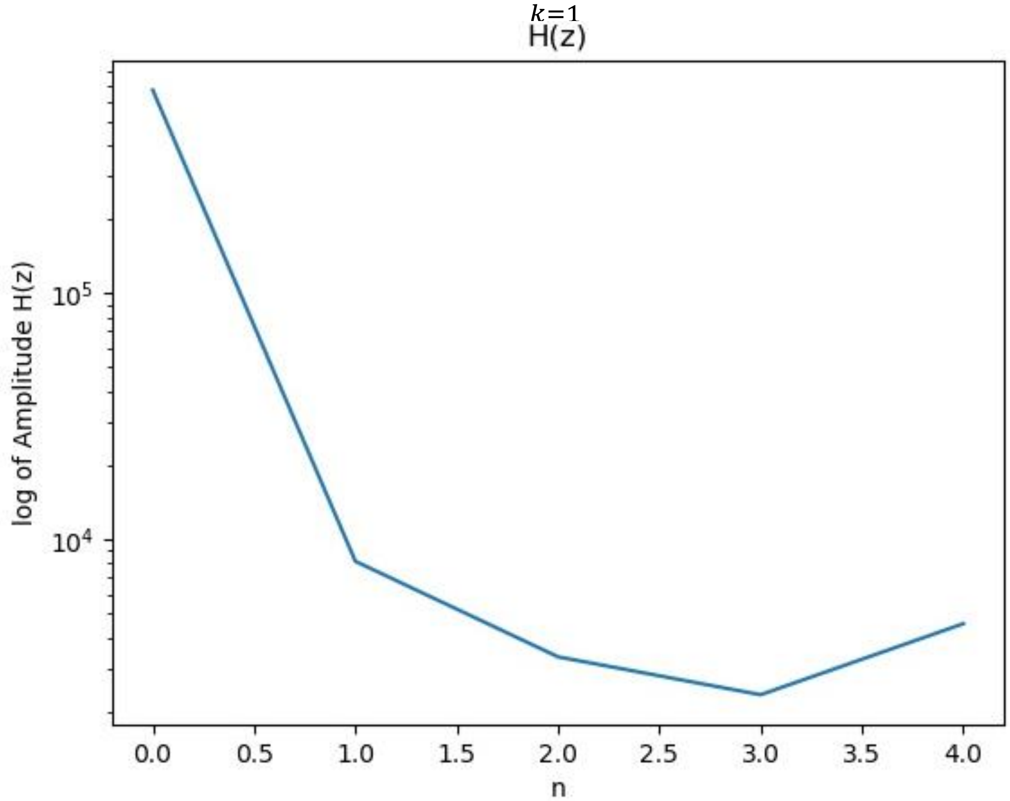
Now, we have r[n] matrix. The next step is solving a_n and G with the equation below.

$$\begin{bmatrix} r[0] & r[1] & \cdots & r[p-1] \\ r[1] & r[2] & & r[p-2] \\ \vdots & & \ddots & \vdots \\ r[p-1] & r[p-2] & \cdots & r[0] \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} r[1] \\ r[2] \\ \vdots \\ r[p] \end{bmatrix}$$

$$G^2 = r[0] + \sum_{k=1}^{p} a_k r[k]$$

Since we have a_n and G, we can figure out x[n]. x[n] is a eleven point vector in this project. By making reciprocal to the DFT of x[n], we get H(z) as showed. Also, H(z) is symmetry, so I just showed the first half of it.

$$\mathrm{x[n]} = \frac{1}{G}\delta[n] + \sum_{k=1}^{p} \frac{a_k}{G}\delta[n-k]$$

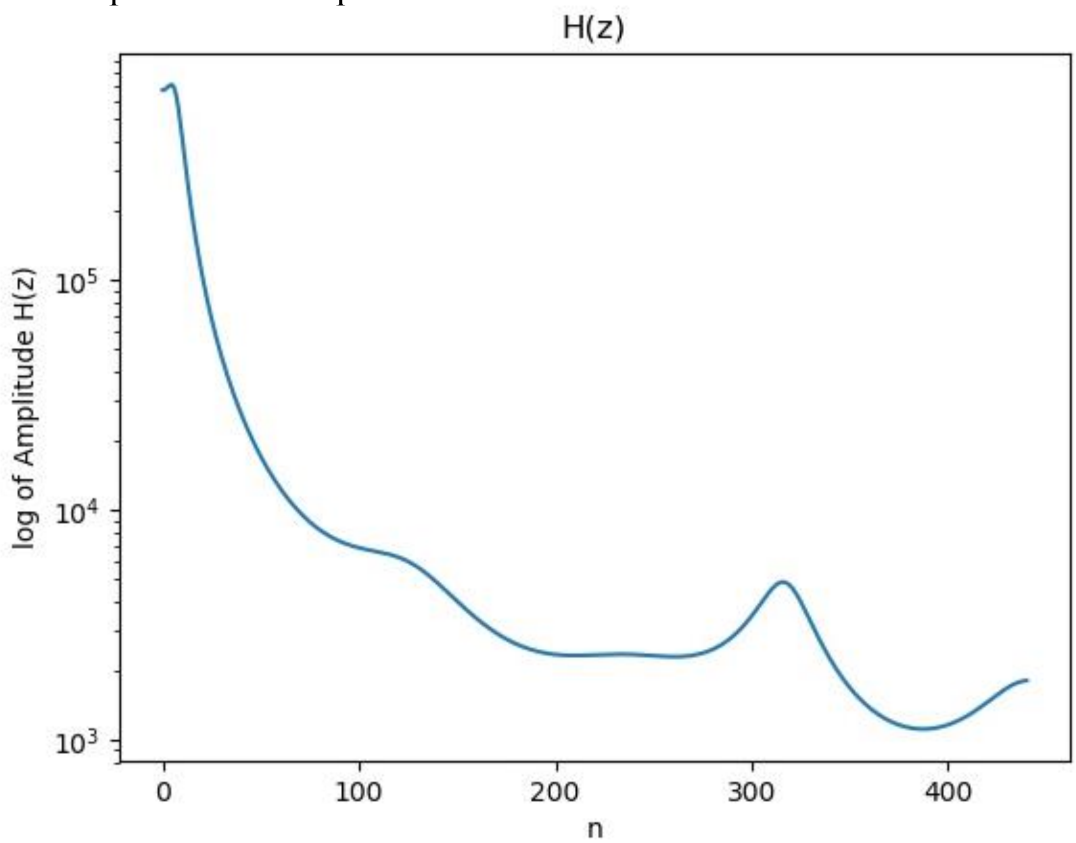


2.3 Discuss the similarities and differences between the log-magnitude plots you obtained in the project.

From what I learned, how a video sound like is mainly determined by two things. The first one is the shape of the whole figure in the frequency domain. The second one is the frequency and the amplitude of the "impulses" in the frequency domain.

As we can see, the DFT of the window interval signal is abundant with such information. But the figure of H(z) is made of just 5 points of information. In this way, the information in the original signal cannot be fully presented and the figure of H(z) itself is also rough.

To make a more delicate figure, we add 0 to the end of x[n] to make x[n] as a 1024 points signal. Then, we do 1024 points DFT to x[n] and make H(z) as the reciprocal of the DFT of x[n]. In this way, the figure of H(z) looks more smoothly and much more like the DFT of the window interval signal. However, it still cannot present the "impluses".



I suppose that since we are doing Indirect least-squares approach to PSM, it is never possible to make the H(z) looks exactly the same as the DFT of the original Signal. Although they may look alike, but there will be information loss however. This is the biggest difference between H(z) and the DFT of the original signal.

3. The challenges I faced in the project

I used the same wav file as I used in the project 1 at the beginning. However, this wav file has a sample rate of 11025Hz which is only quarter of the sample rate as the wav file I use now. The low sample rate causes many problems. The sample point I got in the windowed interval is just a little comparing to what I got now. In this way, both the figure of window interval and the figure of its DFT is rough. I believe this will also affect the quality of frequency response, so I give up this wav file and find another one with sample rate of 44100 Hz instead. In this way, the result I got is believed to be much better.

From the DFT of window interval signal, we found an irregular “impulse” around 16000Hz. I’m not sure why this happened. I asked some other classmates. Actually, we used different signal and different programming software as well, but nearly all of us have the same problem. I guess there might be something to do with the interception of the signal because our signal is intercepted from the movie, video or something else. I will continue working on that.

4. The code and Github link

I put my Github link here in case there is something I didn’t clarify.
Github link: https://github.com/cuicyz/EC516-project2
You can find all things you need in my Github including the wav file, the code and all the pictures.