Project2 report

Qihua Gong

For the part 1 of the project, we need to design a speech communication channel using the LPC compression scheme. I separate this part in to two sections. The first part will compress the audio signal into as few coefficients so that it can be transmitted over the communication channel. Then in part two I will transmit the signal and the receiver needs to be able to re-synthesize the audio signal back.

In the compression part, first we need to load the audio and analysis it. I set a part of parameters as the changeable input that I can adjust it freely. They are the frame length, frame offset, filter order and noise volume. I first use a filter to preprocessing of input sample before LPC analysis. I will normalize it to [-1,1] and filter it. The filter chooses here is the butterworth filter. It will return the transfer function coefficients of an n order lowpass digital butterworth filter with normalized cutoff frequency Wn. Then limit the input and initial the array we need to use later. They are the output array, excitation array, gain array and pitch period array. In order to achieve the goal of compressed transmission, we will complete the transmission through analysis, encode, transmit and decode.

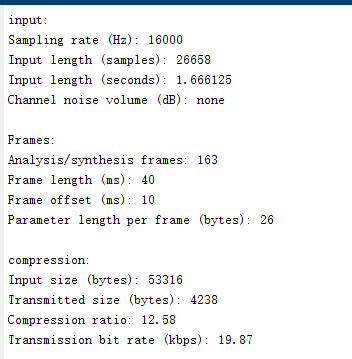
For the Lpc analysis, I limit the pitch and set the window as Hamming. Then use the Lpc function to get the coefficients and error variance. Also write a function to find the pitch and voice/unvoice parameter. I write the bonus part for the Pitchdetector, it can check the input sample X to determine whether X is periodic and estimate its fundamental pitch period in samples within the range [pitchmin, pitchmax]. If X is periodic, the voiced/unvoiced parameter V will be set to 1. This way also detect the voiced or unvoiced situation.

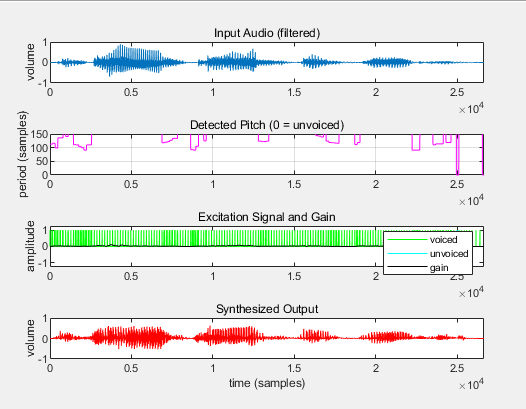
Next, I go to the part of encode. For the both part of encode and decode, I use the Matlab build function poly2lsf and lsf2poly. They will convert prediction filter coefficients to line spectral frequencies and finally convert it back.

In the transmit part. It will transmit of LPC parameters over a simulated channel with random noise. Then returns the input vector params with noise of amplitude volume.

Before the synthesis, we also need to prepare the output sample and excitation frame. The frame will separate the train of period pitch and white noise.

Finally, we will use the overlap-added way to synthesis the audio signal. I input frames and excite source to the output arrays we build before with samples of frame offset. As stated in the project requirements, I need to reconstruct waveforms with the best possible quality. Although I don’t know how to achieve the best results, I need to set up some monitoring points for the parameters used and the resynthesized audio. The inspection index we set up here is origin input audio information, output frames and parameter lengths, and data compression information.

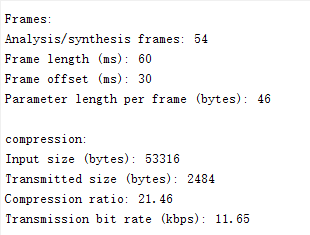
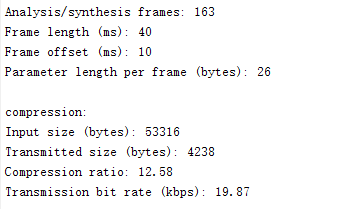


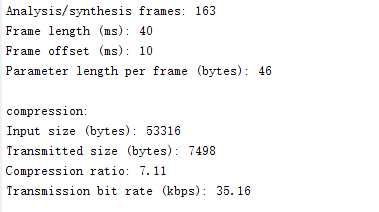
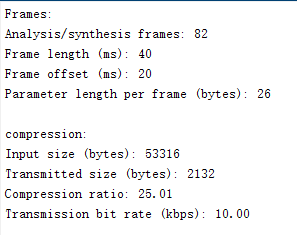


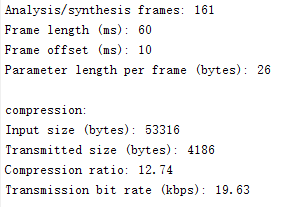
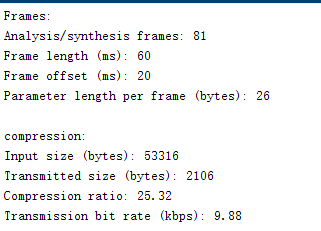
Here is a test sample for the Sample1.wav. I repeat the process ten times and save the result of Sample wave 1- 10.

The three parameters of Transmitted size, Compression ratio, and Transmission bit rate are the main statistical indicators, and they affect the quality of resynthesized speech. The quality of the synthesis can be discerned by my ears. And these three parameters will change with the change of frame parameters and filter order. The specific comparison is as follows:

The below shows the compare when I change the parameters in the frame.

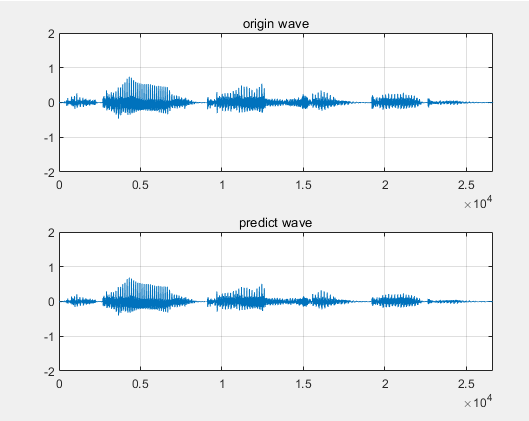
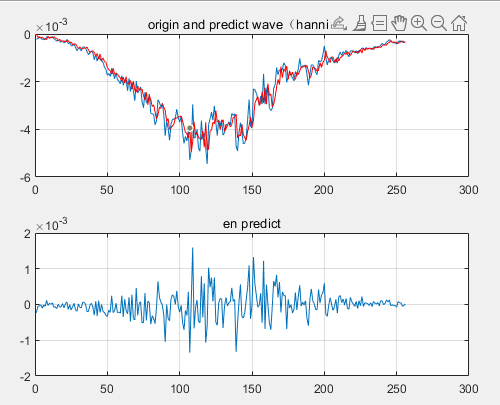
 

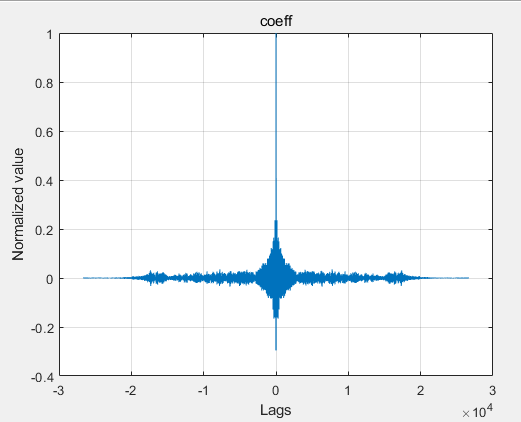
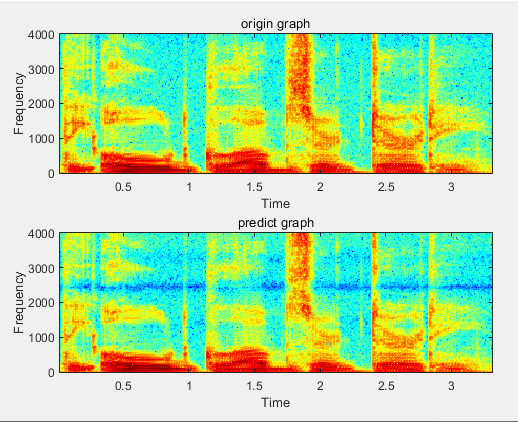
 

After experimenting, I found that the length and offset of the frame and the filter order mainly affect the ratio of the number of compressions. So write these as variable variables at the top. Then change and compare them. I think the best possible quality, using the least amount of coefficients, does not actually exist, because after listening to the audio, I feel that the greater the amount of data transmitted by the channel, the less the compression, the better the sound. The difference lies in the small parameter change, which is intuitively reflected in the compression rate and transmit rate, but I can not feel it. In other words, human ears cannot tell the difference. If this standard is used to meet better compression and quality, I will choose the first set of data in the picture above. You can see that the compression ratio has come to 20%, but I sounded better than 7% compression. It may be that 7% of the sound has more noise that makes me think that the audio is not good. The noise I chose has a higher degree of compression, but the human voice is better restored. In fact, you can still clearly hear which audio is more compressed, but I subjectively think his reconstruction is better.

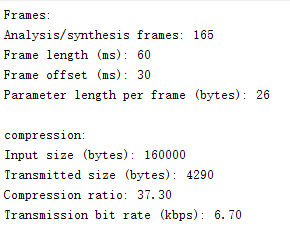
LPC is to model the speech signal into a sequence of processes. Voiced sound excitation is a quasi-periodic impulse sequence, and unvoiced sound excitation is a white noise sequence. The clustering effect of vowels is much better, and the same vowel syllables have relatively close formants even for different people, which means that they have similar LP coefficients.

Also I write a small function follow the official documentation of the lpc function in MATLAB - test1.m. This test is used to analyze the formants of vowel phonemes, and the prediction coefficients and error functions of different samples are used to synthesize sound effects. A comparison chart of synthesized audio and original audio is drawn.





For the part 2 of the project, I wrote a small program recorder.m to record my voice, and imported this audio into the step of part1 and synthesis the voice. The following is a comparison of the results of several parameters. I use my ears to judge the quality of the final synthesized sound to determine the final selected parameters. The finally selected parameters are:



This sentence has a large number of fricatives, specially the phoneme /s/. These fricatives sounds are more pronounced in the noisy restored audio. As we said above, if the compression ratio is large, the noise will have a smaller impact on the audio perception, but the compression ratio will be large, and the loss of the original audio will be large. What I did here was to constantly change the parameters to find a balance point, and finally chose this group of parameters. We can see that his compression ratio is as high as 40%, but it is the clearest group I hear.