# Audio signal processing: Project2

Meili Liu

November 18, 2022

**Abstract**

Your abstract.

## 1 Introduction

Linear predictive coding (LPC) is a widely used technique in audio signal compression. The philosophy of LPC is related to the human voice production. Then vocal fold generate the sound source $e(n)$, which then go through a linear, time-varying filter $h(n)$. The output signal can be represented as:

$$x(n) = e(n) * h(n)$$

The source signal could be an quasi-periodic pulses (voiced speech) or random noise (unvoiced speech). The filter is modeled as all-pole filter, which can be represented as:

$$H(z) = \frac{1}{1 - \Sigma a_k Z^{-k}}$$

If we could extract the coefficients $a_k, k = 1...p$ of the all pole filter, then we can reconstruct the signal $x(n)$ by using the current input e(n) and past samples $x(n-k), k = 1...p$ using the following equation:

$$x(n) = \Sigma_{k=1}^{p} a_k x(n-k) + e(n)$$

In this report, we implement a Vocoder based on LPC method above, to achieve signal compression and reconstruction. To be more specific, we first cut the signal into small frames and then for each frame, we add a window function and then encoded the windowed signal into a few coefficients based on LPC method. These coefficients include:

1. Pitch
2. Gain
3. Poles coefficients $a_k, k = 1...p$

Then, we try to reconstruct each frame signals based on these coefficients and overlap add them to recover the original signal. Finally, we evaluate the quality of the synthesized signal by ear.

There is a trade-off between the compression rate and the quality of the synthesized speech. Our goal is to use the least number of coefficients to reconstruct the original signal with the best quality. After experiment, we found that, for a 16 kHz frame rate speech signal, we only need to store 454 samples per second to resynthesize the original signal in a pretty high standard.

### 1.1 Description of code

*vocoder.m*: the main LPC vocoder program
*vocoder_triangle.m*: the LPC vocoder based on triangle pulse excitation
*triangle_pulse.m*: function for generating triangle pulse
*pitchDetector_xcorr.m*: function for pitch detection based on auto-correlation method
*pitchDetector_FFT.m*: function for pitch detection based on FFT method

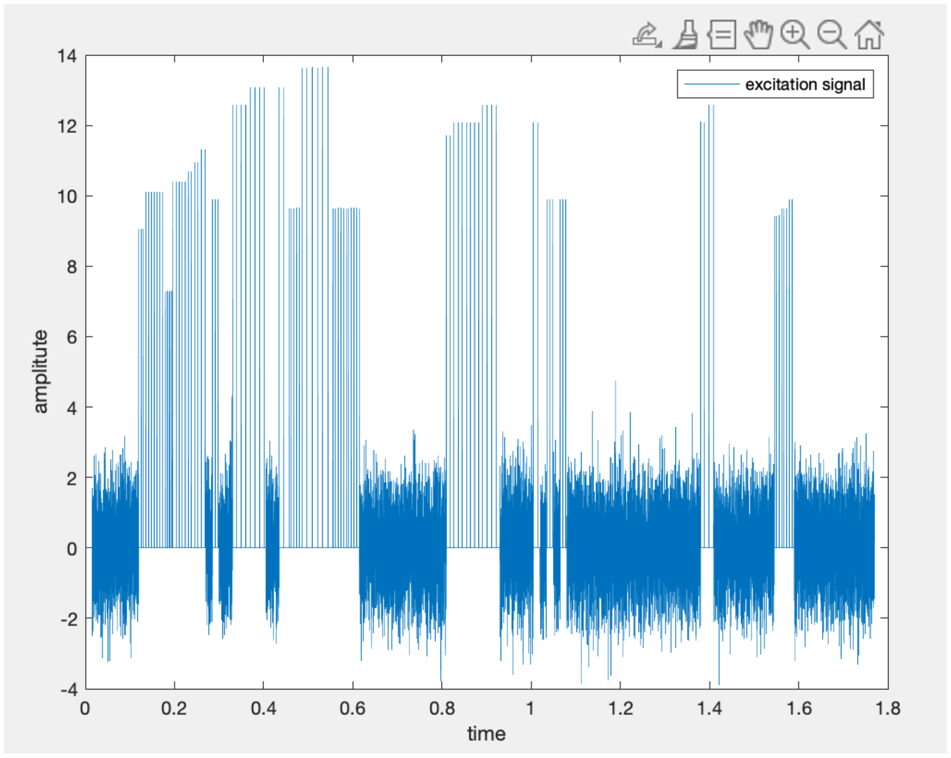


Figure 1: The excitation signal for Sample1.

## 2 How to programmed the LPC Vocoder

In this section, we are going to discuss some details about implementation of the LPC Vocoder that we use, including excitation generation and pitch detection.

### 2.1 How to generate excitation

In the process of human speech signal production, there are two possible sources: an voiced glottal flow with fixed pitch period for voiced sound and noise for plosive/fricatives. In our experiment, we simulate these sources by impulse train and random white noise, respectively.

The files X.txt provided contains estimates of pitch frequencies information. In these files, if the frequency equals to zeros, then the current frame is unvoiced, and we can use the *rand* function in Matlab to generate a random signal. The generated white noise follows a normal distribution with a mean value 0 and a standard deviation 1. This ensures that the energy power of a random signal equals to 1.

For voiced signal, we simply generate impulse train frame-by-frame to simulate the glottal wave with pitch period equals to $round(fs/frequency)$, where $fs$ is the sample rate and frequency is the fundamental frequency of the current frame. An important issue is that the current frame are not independent of previous frame. More precisely, we need to know the last position of previous impulse and use it as the start point of current impulse. In addition, to keep the power of impulse train at 1, we need to compensate the size of impulse use $sqrt(period)$.

The excitation signal simulated for sample1 is shown in Figure 1

### 2.2 How to detect pitch frequency

In addition to use the given pitch frequency files, we can also design a automatical pitch detection program, which can help us to detect pitch in part 2. Pitch detection is an extensively researched topic. In our project, we implemented two methods, auto-correlation method and the frequency spectrum method.

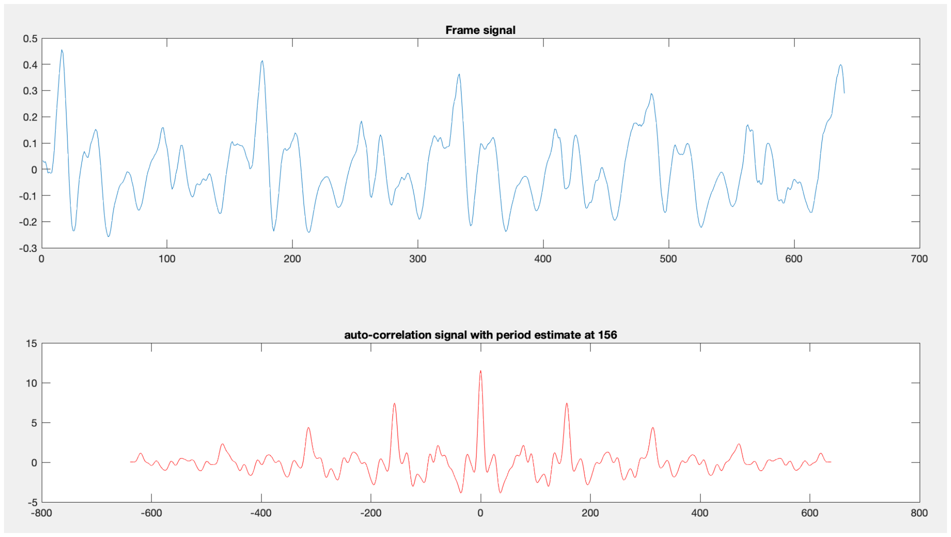


Figure 2: The frame signal(up) and its auto-correlation function(down).

#### 2.2.1 Detect pitch frequency based on auto-correlation

Auto-correlation is the correlation of signal with a delayed copy of itself as a function of delay. For a period signal with period P, the auto-correlation function has peaks at $0, +/-P, +/-2P$. Thus, by finding the difference between peaks of auto-correlation signal, we can get the fundamental frequency of the original signal. As shown in Figure 2, for the auto-correlation signal, the distance between the central peak and second highest peak is 156, thus we can compute that the $period = 156$ and the fundamental frequency is $fs/period = 102.56Hz$ for this frame signal. For those frames with frequency higher than 140Hz and lower than 90Hz, we set them as 0. This is how we implemented the pitch detection at $pitchDetectoin\_xcorr.m$ function.

#### 2.2.2 Detect pitch frequency based on FFT

Another method we used to detect pitch of a signal is based on the frequency domain approach, and we implemented this method at $pitchDetector\_FFT.m$ function. We first convert the frame signal to an estimate of frequency spectrum utilizing the Short-Time Fourier Transform. and then select the first peak of spectrum as the fundamental frequency. After that, we need to check silence frame and noise frame, and set their fundamental frequency as 0, which is similar with auto-correlation method.

In Figure 3 4, we compare these two methods by the quality of reconstructed signal Sample1. As we can see from the graph, both these two methods can detect the pitch frequency well. In the following report, we choose the second method to compute pitch information.

## 3 Evaluation of synthesis performance based on different parameters

In this section, we tried to optimize the choices of parameters, such as window shape, frame length, frame overlap, p-value, and glottal shapes, to achieve a balance between the quality of reconstruction and compression rate.

### 3.1 Window, frame length and frame overlaps

The principle of LPC is to assume that the speech signal is generated by a source-filter model. For different phones, the coefficients of the filter are different . Therefore, the choice of frame length is crucial. For example, if the length of the frame is too large, the frame will contain two or even more phones. If the length of the frame is too small, we will only get part of the phones. Both cases will lead to inaccurate coefficient calculations.

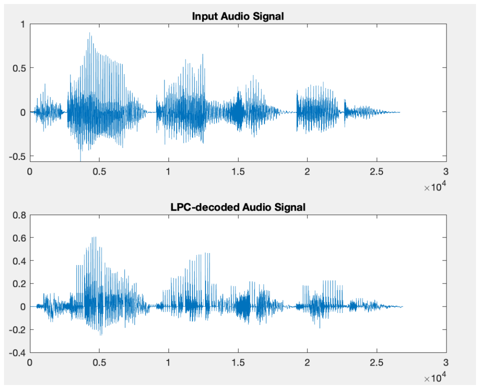


Figure 3: the original signal (up) and the synthesised signal based on the auto-correlation method (down).

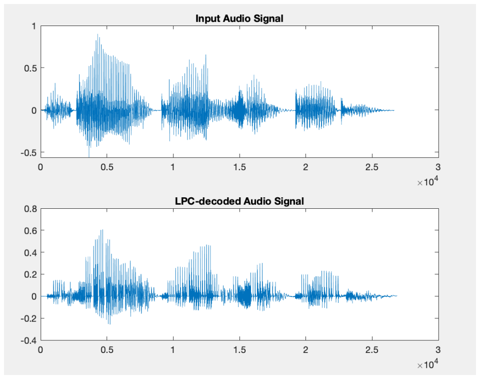


Figure 4: the original signal (up) and the synthesised signal based on the FFT method (down).

In additional, the frame overlaps and window shape would also influence the quality of reconstructed signal.

In our experiment, we choose three kinds of window option: the hanning window, hamming window and no window. The frame length is selected from 20ms, 30ms and 40ms, and the overlap is from 10ms, 15ms, 20ms, 25ms and 30ms. After comparison, we finally choose hanning window, with frame length at 30ms, with 10ms overlap. The original signal and synthesis signal are shown on Figure 5

## 3.2 P value

The selection of P value is also very important. LPC assumes the filter is a p-th order all-pole model, larger p can improve the quality of reconstructed signal, but may also lead to low compression rate and high computational burden. After evaluating the synthesised signal by our ear, we find that $p = 16$ allows for a reasonable balance between quality with compression rate.

## 3.3 vary the number of coefficients for voiced vs. unvoiced frames

To further improve our compression ratio, we tried to reduce the number of coefficients for unvoiced frames. This is because for unvoiced phonemes, their frequency spectrum only contains high frequency noise part. Therefore, the formant information is not very important for them. After experiment, we found that we can only keep 6 coefficient for unvoiced frames without affecting the quality of decoded signal.

This strategy can dramatically improve the compression ratio, because we only need 8 number (6 LPC coefficients + 1 gain value + 1 pitch information) to encoded unvoiced frame. For voiced frames, we still use 18 numbers (16 LPC coefficients + 1 gain value + 1 pitch information). In table below, we list the number of samples needed for reconstructing each audio signal. The average compression rate is 454 samples/second.

| signal | duration | number of samples after compression | comression rate (samples/second) |
|---|---|---|---|
| Sample1 | 1.67s | 825 | 494 |
| Sample2 | 1.68s | 715 | 426 |
| Sample3 | 1.91s | 887 | 464 |
| Sample4 | 1.83s | 780 | 426 |
| Sample5 | 1.86s | 899 | 483 |
| Sample6 | 1.94s | 956 | 493 |
| Sample7 | 1.58s | 628 | 397 |
| Sample8 | 1.88s | 838 | 446 |
| Sample9 | 1.32s | 657 | 498 |
| Sample10 | 1.78s | 742 | 417 |
| total | 17.45s | 7927 | 454 |

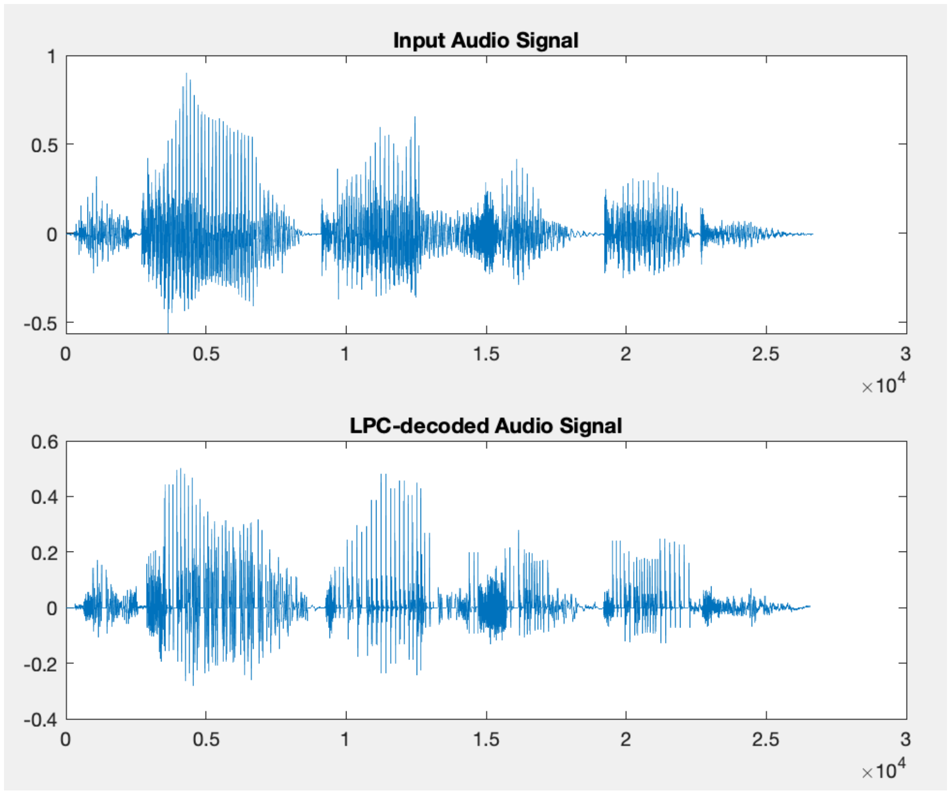


Figure 5: The original signal (up) and the synthesises signal (down) with p=12, frame length = 30ms, and overlap = 10ms.

### 3.4 Glottal shapes

In this section, we want to explore the influence of different pulse shapes for the quality of synthesised signal. Considering the shape of glottal wave, shown on the left of Figure 6, We try to simulate it use a skewed triangle pulse, which has a closed phase, open phase and a return phase, shown in the right of Figure 6. The first 10 frame of generated excitation signal is shown on Figure 7.

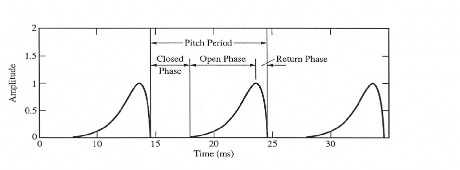
The Vocoder based on triangle pulse is implemented at *vocoder_triangle.m*. we also show the decoded signal for Sample1 on Figure 8. Compared to the impulse excitation, the quality of signal synthesized by triangle pulse is a little bit worse. Therefore, we still choose impulse signal as the excitation signal.

### 3.5 Analysis of main sources of distortion

There could be several reasons for the distortion of decoded signal. First, the estimation of pitch frequency for voiced signal may have some errors. some voiced frames may be incorrectly identified as unvoiced, while some unvoiced signal may be identified as voiced. Second, for voiced fricatives and plosive, they actually mix up excitation signal with noise signal. However, in our experiment, we doesn't consider this situation. Lastly, the setting of $p = 16$ losing lot of information of vocal tract, after changing $p = 48$, the quality has improved a lot.

## 4 Part2

In the part2 of this project, we are required to record our own voice of a sentence "The synthesized signal signal is supposed to be of high standard". Then we tried to optimize frame length, overlap, window choice, and p-value specifically for this sentence. After experiment, we choose $p = 16$ for voiced frames, $p = 6$ for unvoiced frames, window length = 30ms, overlap=10ms. The comparison of



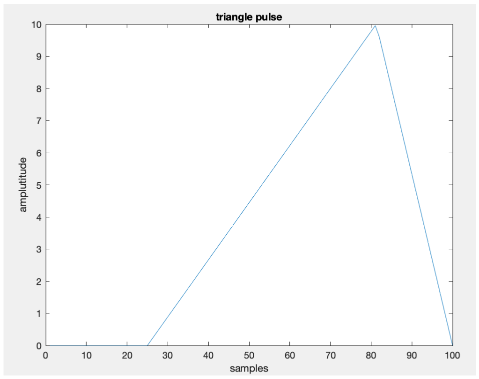


Figure 6: The glottal wave (left) and the simulated triangle pulse

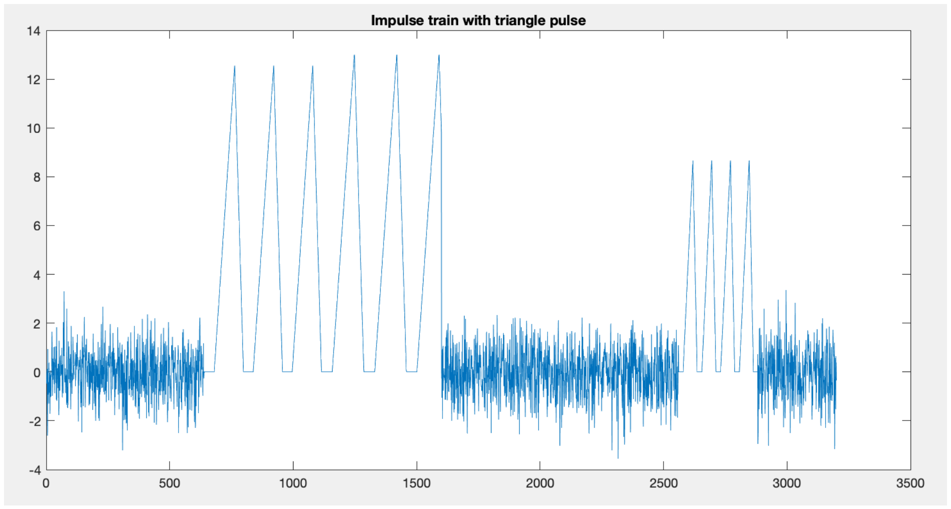


Figure 7: The first 10 frame of excitation signal of Sample1

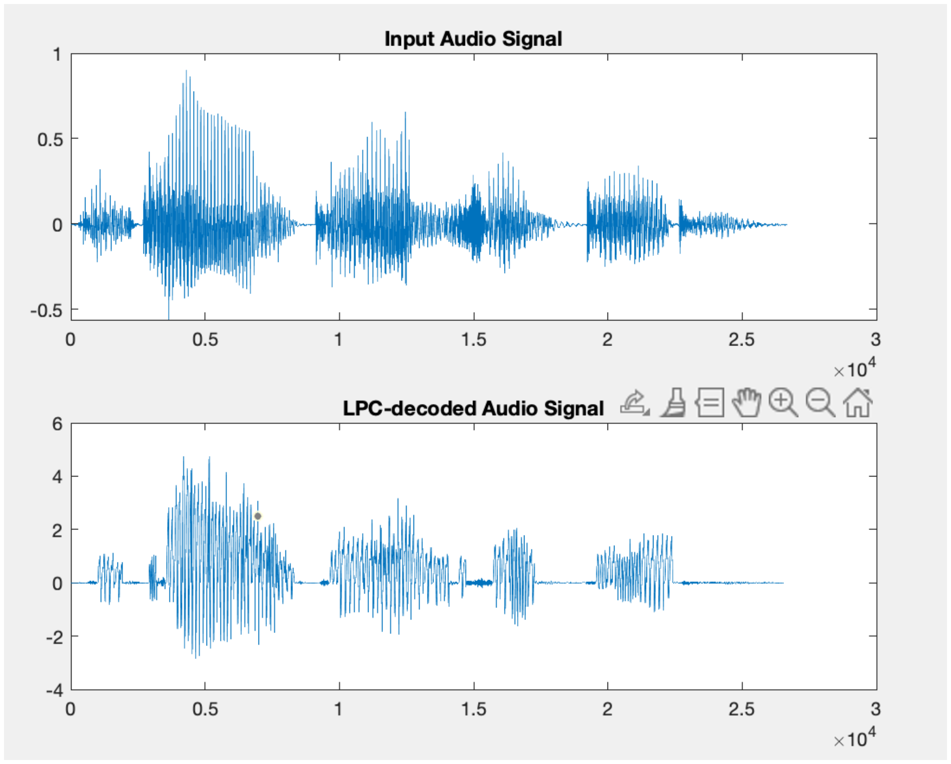


Figure 8: The compare of original Sample1 (up) with decoded signal using triangle pulse (down)

original signal and reconstructed signal in time-domain and frequency domain are shown on Figure 9 and Figure 10, respectively.

As we can see from Figure 10, most of the formant information are well preserved and restored, which guarantee the quality of synthesised signal. Besides, because there are a large number of constants, which only need 8 number to decode. Thus The compression rate is only 401 samples/second for this sentence, lower than other test sentences.

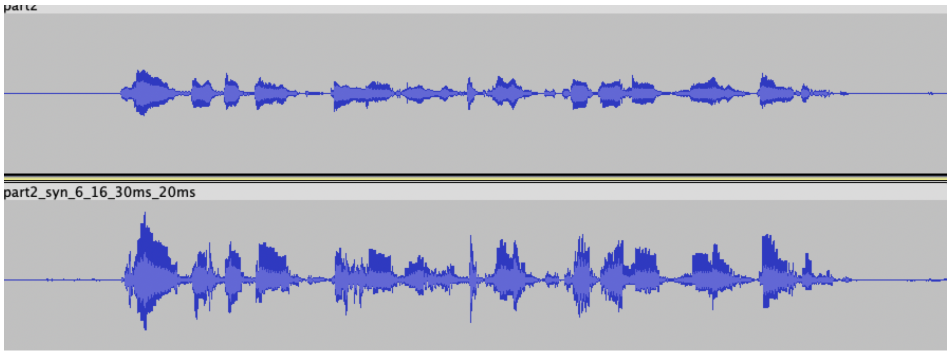


Figure 9: original signal (up) and reconstructed signal (down) on time-domain

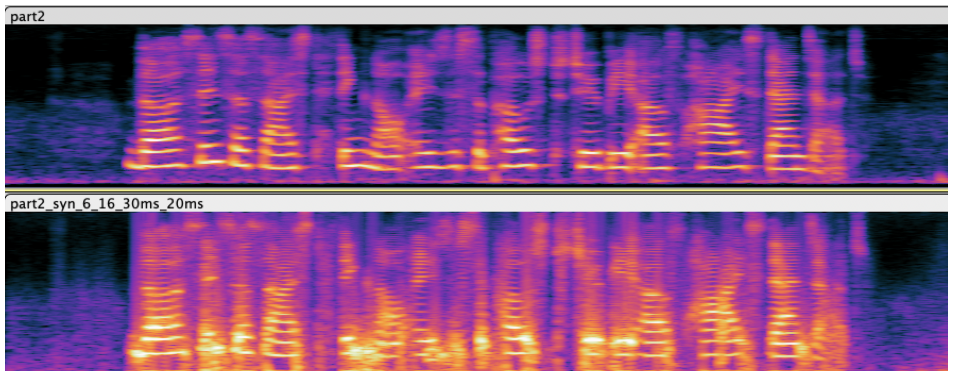


Figure 10: original signal (up) and reconstructed signal (down) on frequency domain