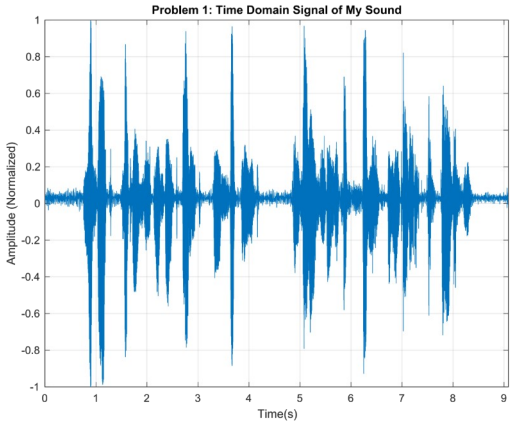
National Taiwan University

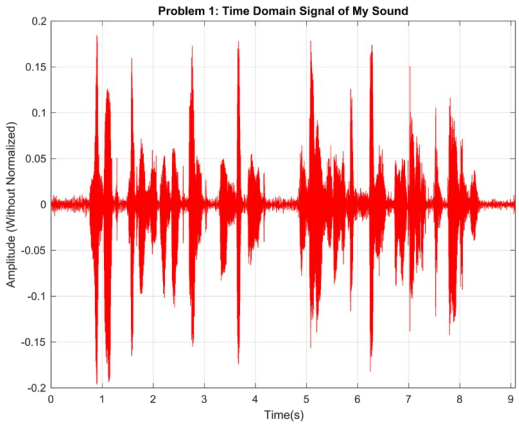
Fourier transform and Fourier optics

Project: Audio Signal Processing (Spring 2025)

B10901151 林祐群

Problem 1: Record your own voice and create an audio file about 10 seconds in length. Plot the signal in the time domain. Be sure to label the axes with units.

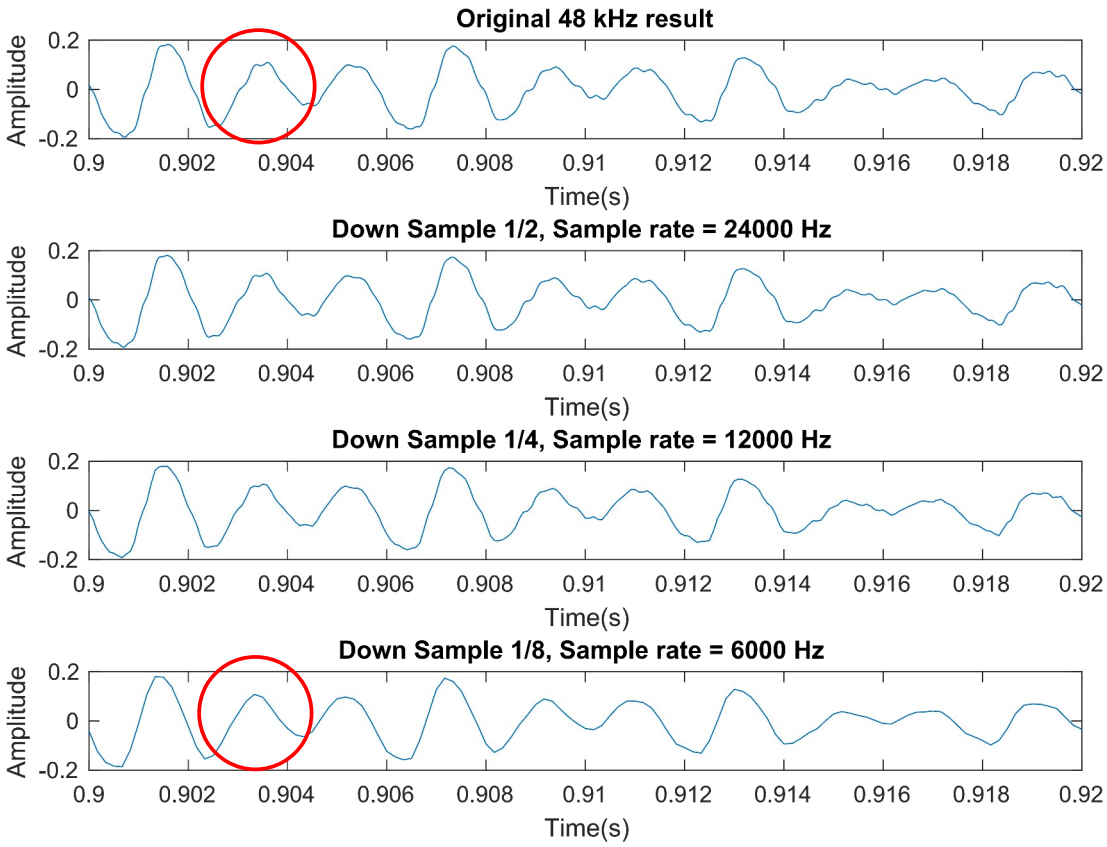




Problem 2: What was the sampling rate? Reduce the sampling rate numerically and compare the signals with different sampling rates in the time domain. Describe what you observe.

The original sampling rate of the audio is **<u>48000Hz</u>**.

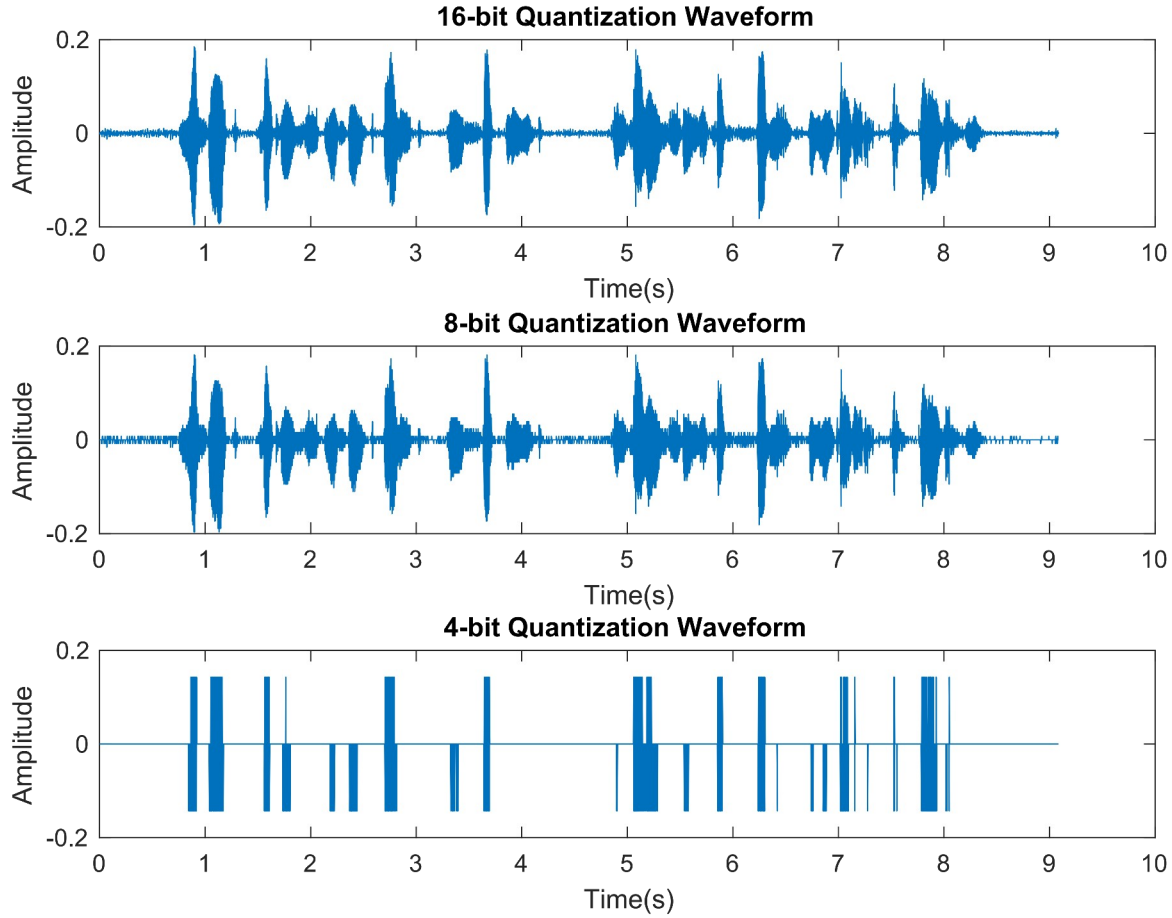
I show the result



without normalization and the x axis is set to the range [0.9, 0.92] for clear comparison between different sampling rate. It's is obviously demonstrated that after down sampling with reducing sampling rate, the waveform becomes "sparse". The high-frequency information is lost and cost the waveform to be different from the original signal. As the two red circles highlighted, we can discover the difference and deterioration of downsample the signal.

Problem 3: How many bits were used to quantize the time-domain signal? Reduce the bits for quantization numerically and compare the signals with different quantization in the time domain. Describe what you observe.

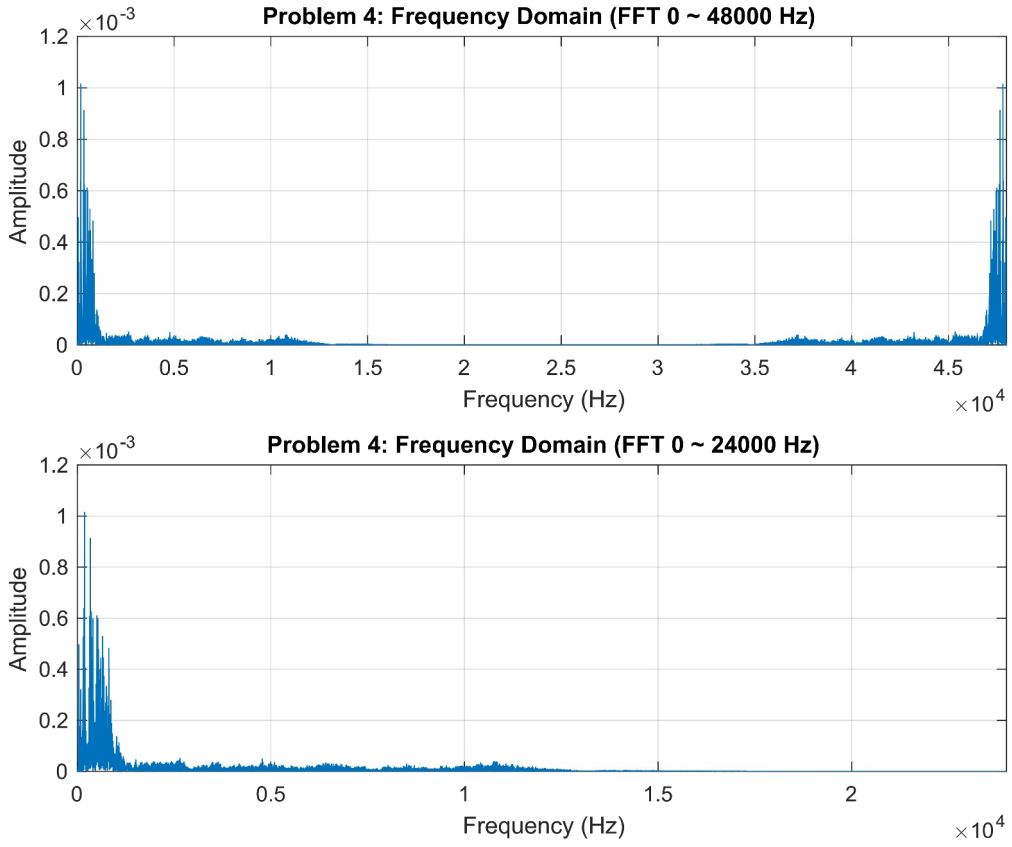
The original number of bits used to quantize the time-domain signal is **16-bits**.
From the above graphs, it's clearly shown that as the bit depth becomes lower, the waveform



starts to suffer from step-shaped distortion. This is due to the capacity of representation; where 16-bit representation has almost none-distortion, while the 4-bit quantization has extreme distortion. The change in amplitude should be large enough so that it's possible for the 4-bit representation to properly quantized the signal, which can also be observed from the long-existing zeros.

Problem 4: Perform fast Fourier transform and plot the signal in the frequency domain. Be sure to label the axes with units. Describe the energy distribution in the spectrum. What frequency components are the signal? What are the noise?

The audio energy is concentrated mostly in the region of 0-4kHz, especially in lower



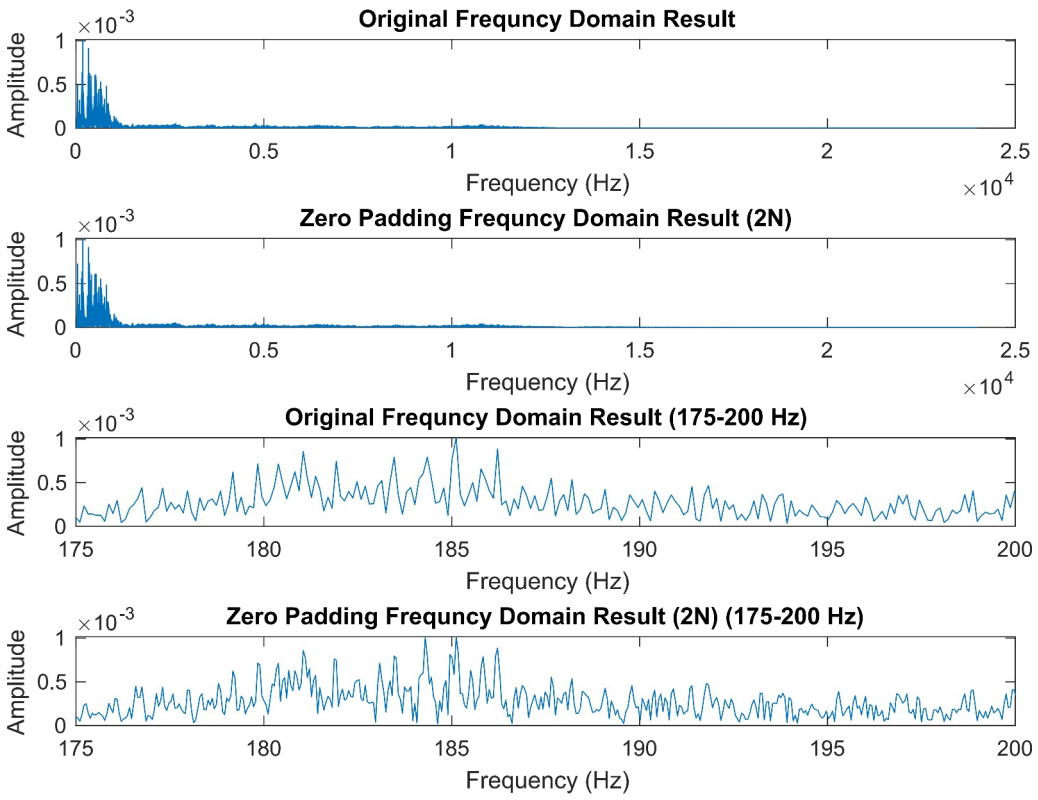
frequency. Though there are fluctuation between 2000-12500Hz, the frequency components of frequency 12500-24000Hz are very small. Beyond that, the energy higher than 24000Hz are generally more possible to be recording noise or environmental noise. The above graph is the result of original FFT; thus, we can discover the (circular) shift property with the replication of waveform.

Problem 5: Double the length of the signal by the zero padding technique in the time domain.

Compare the signal spectrum with and without zero-padding. Describe what you observe.

Is the frequency resolution increased?

From the first two graphs below, we can observed that the functionality of zero padding is to do interpolation in frequency domain, such that the waveform will be more dense. Thus, the general, or large-scale view of the two waveform are similar. As for the difference, I demonstrate it through the bottom two graphs where the one with interpolation is obviously equipped with more data point; as a result, “seemingly” finer graph is given. In fact, the frequency resolution still remains the same since no additional information is given. The only difference is to let the diagram more appealing and visually attending for our objectives.



Problem 6: Perform short-time Fourier Transform and show the signal in a spectrogram (a frequency vs time plot). Describe how you window the signal and estimate the corresponding time resolution and frequency resolution.

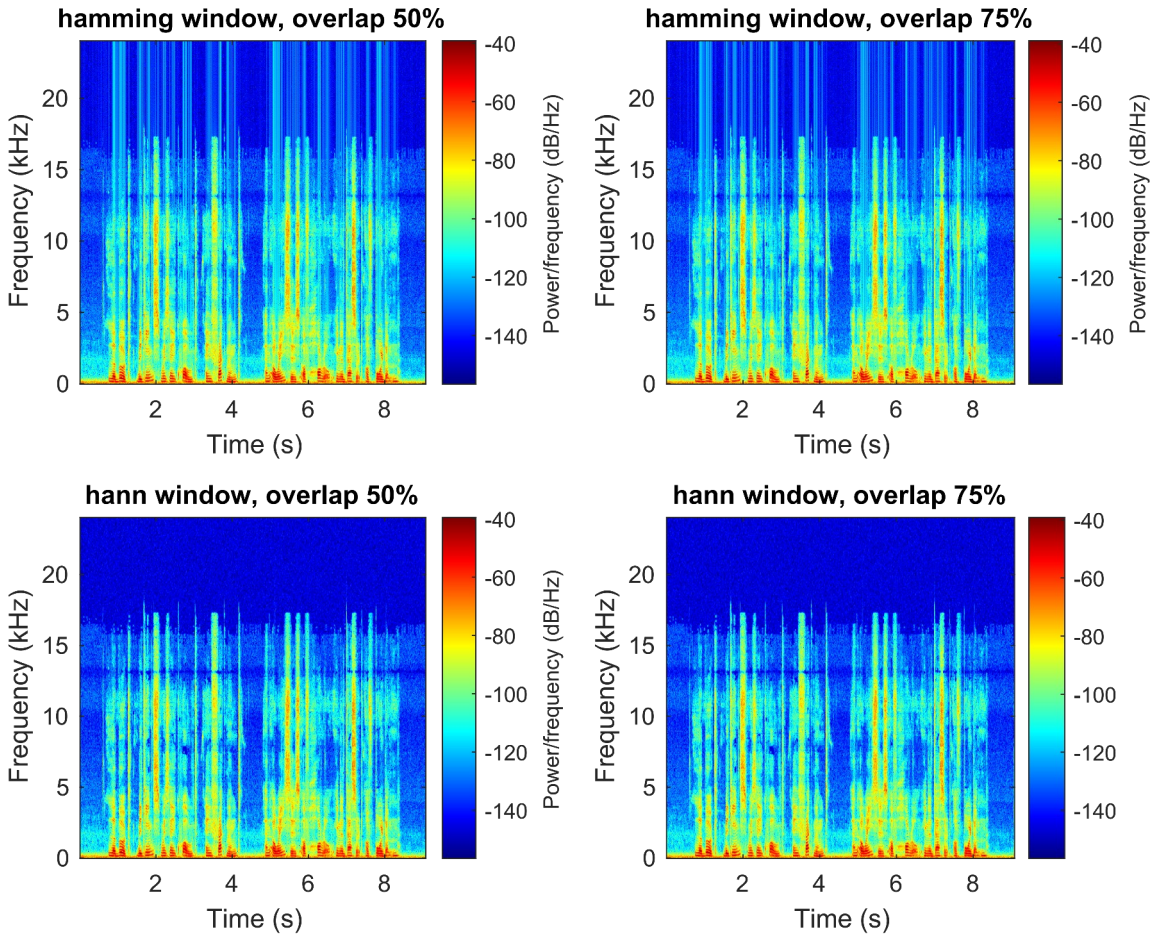
In this question, I implement can compare 4 cases, which consists of two window techniques (hamming, hann window) and two overlap are ratio (50% and 75%). For frame length (window size), I split the 9 seconds mono signal into successive frames of L = 1024 samples each. At a sampling rate of 48000Hz, each frame spans 1024/48000 ≈ 21.3 ms.

With multiplying each frame by a tapering window to reduce spectral leakage, we can discover some common results: Hamming window leads to lower sidelobes, but slightly wider main lobe. On the other hands, Hann window results in narrower main lobe, but slightly higher sidelobes.

As we advance the window by a hop size H between frames, overlapping successive windows: (75% overlap is more computational exhausting!)

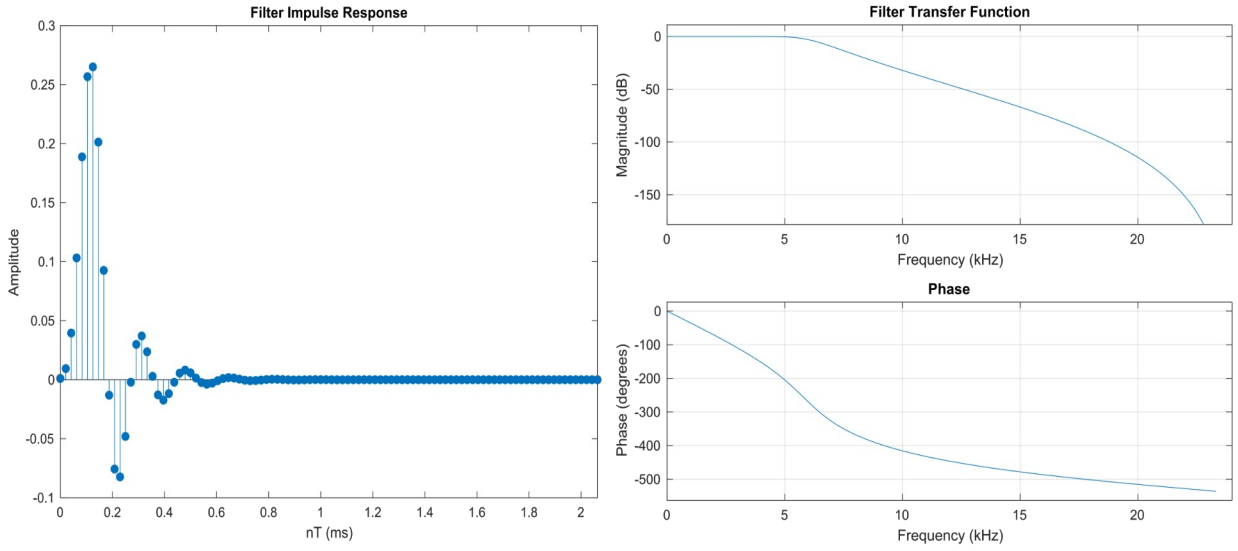
- **50 % overlap**: H = 0.5 L = 512 samples ⇒ frame-to-frame shift of 512/48000 ≈ 10.7 ms
- **75 % overlap**: H = 0.25 L = 256 samples ⇒ shift of 256/48000 ≈ 5.3 ms

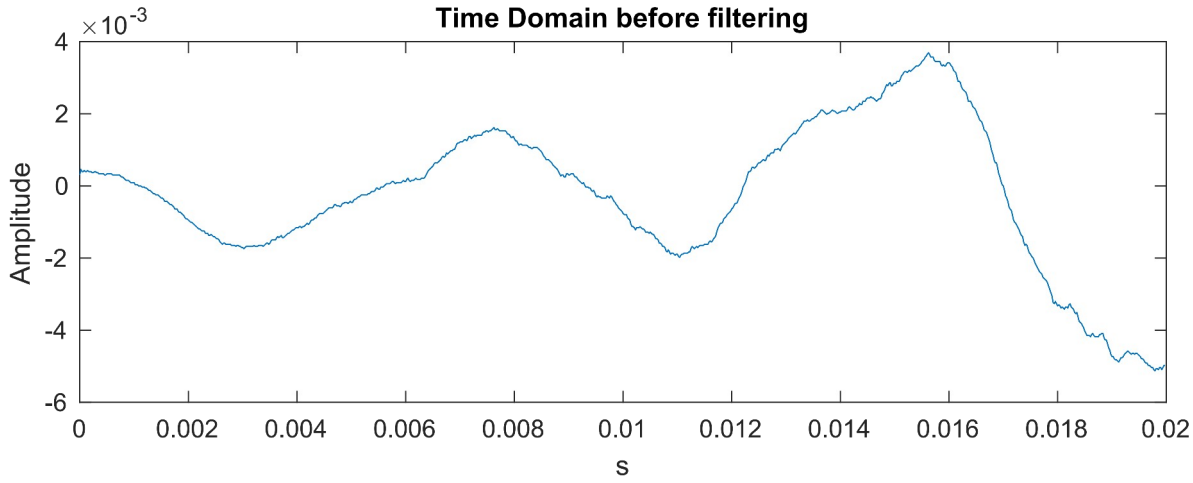
Thus, the time resolution is 21.3 ms and the frequency resolution is 48000/1024 ≈ 46.875 Hz. We can see the reciprocal between the resolution in time domain and frequency domain. The trade-off and the focused aspect are of importance for us to determine and apply, as a result.

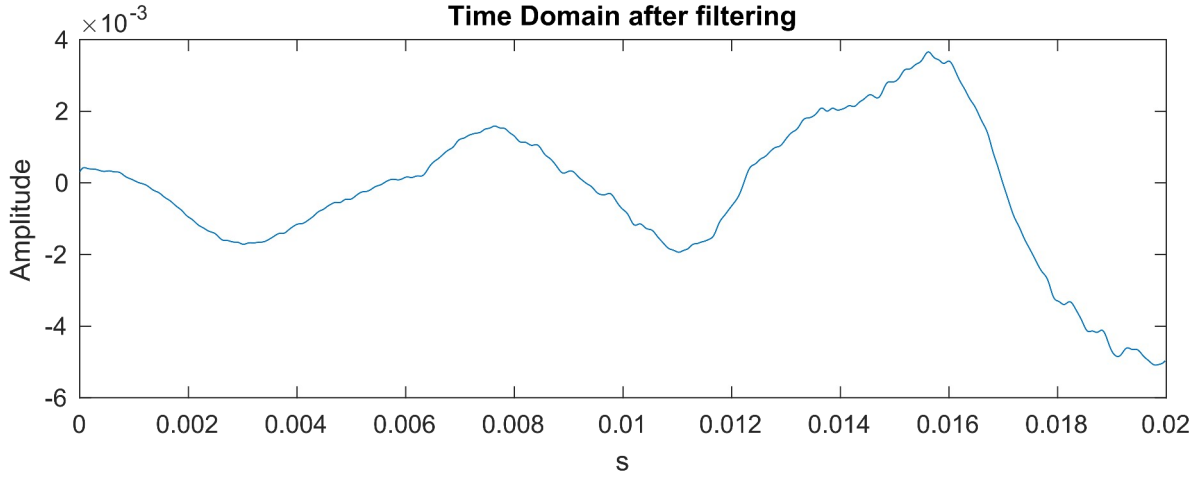


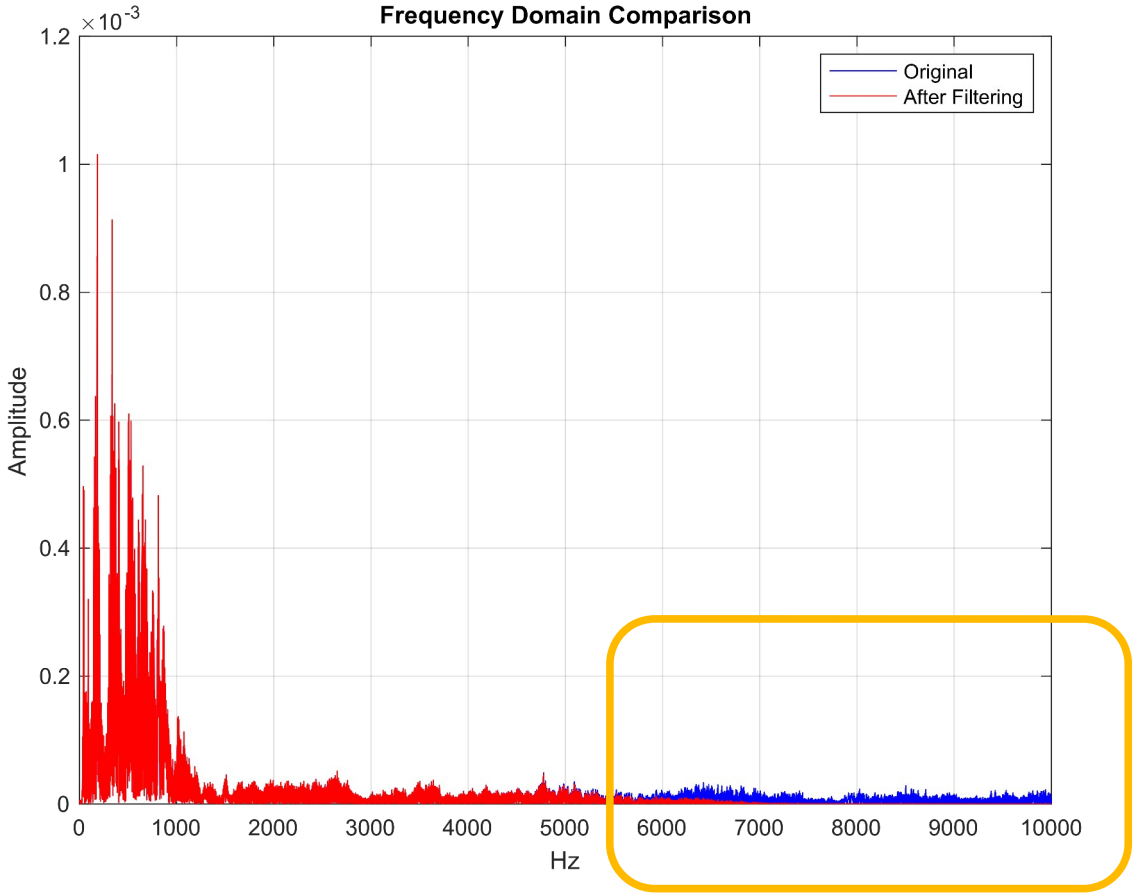
Problem 7: Try to apply a filter to reduce the noise. Describe your filter design. Plot the impulse response and transfer function of your filter. Compare the signal before and after filtering in the time domain and the frequency domain. Play the filtered signal and describe what you observe.

> Impulse Response and Transfer Function of the filter:




Time Domain before filtering
×10^-3
4
2
0
-2
-4
-6
Amplitude
0
0.002
0.004
0.006
0.008
0.01
0.012
0.014
0.016
0.018
0.02
s


Time Domain after filtering
×10^-3
4
2
0
-2
-4
-6
Amplitude
0
0.002
0.004
0.006
0.008
0.01
0.012
0.014
0.016
0.018
0.02
s


Frequency Domain Comparison
×10^-3
1.2
1
0.8
0.6
0.4
0.2
0
Amplitude
Original
After Filtering
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
Hz

I choose to utilize 6th-order butter-worth filter as a low pass filter to eliminate high-frequency noise at first. Then, I use the filtfilt function in Matlab to apply two-directional filtering to the signal to keep thee phase information from distortion. The impulse response and transfer function are given above. We can see that the transfer function is retain at roughly 0 dB even at 5-6 kilohertz.

As for comparing the signal before and after the filtering in time-domain, it's quite clear that the noisy fluctuation is alleviated and the waveform becomes more smooth. On the other hand, for comparing the signal before and after the filtering in frequency-domain, the filter significantly gets rid of the undesired high-frequency components (enclosed by yellow marker). This directly points out the effectiveness of the butter-worth filter for low pass design.

After the filtering, the audio has lesser noise compared with before. However, the difference is not significant mainly due to the reason that my original recording is already clean enough and that the recording studio I used may apply certain degree of filtering in advance. With these consideration, the difference from hearing is not as obvious as in frequency domain (or in time domain) plot as discussed above.

Problem 8: Try to change the pitch of your voice. Describe how you do it and what you observe.

I utilize the resample function in Matlab to modify the pitch of my voice. Resample function (resample(x, p, q)) will do interpolation and resample at the same time. It first up sample by factor p, followed by a low-pass filter, and then down sample for every q distance. That is, it turns the sampling rate from Fs into Fs*(p/q). However, if I force the sample rate of the sound function to be Fs = 48000Hz as the same with the original sampling rate when playing the audio file, the play speed and frequency are changed along with the factor (p/q). As a result, when I set $p/q = 0.8 < 1$, the pitch of my voice becomes higher, while pitch becomes lower for $p/q = 1.25 > 1$.

Despite the method I apply here is simple and effective, the method change play speed and frequency all at once. That is to say, I can't modify neither the speed nor frequency independently without affecting the other one. To further deal with the problem, we may need to include phase-vocoder or other algorithm to achieve such application.

Appendix:

I. 音檔資料:

Audio Format: MPEG-4 AAC, 48000Hz

Data Rate: 127.44 kbit/s

Channels: Stereo (L R)

Bit Depth: 16 bit