Bilkent University

Electrical and Electronics Engineering Department

EEE-424 Coding Assignment 2 Report

11/05/2025

Fatih Mehmet Çetin - 22201689

Introduction

This assignment consists of two questions. In the first part, we design a linear phase FIR bandpass filter with a length 10. The filter is aimed to have a passband in the range of human voice which is between 300Hz and 3kHz. In the second part of the assignment, we test the filter we created in part 1. We record our own voice and use the filter on that audio file. We also carry out manual resampling operation in part 2. We use Matlab for the assignment.

"codass2.m" is the Matlab file that contains all the matlab code for this assignment. **(Appendix 1)**

Assignment Work

Question 1

The question asks us to create a linear phase bandpass filter with a length 10. The filter must have a narrow passband such that for audio recordings sampled at 20 kHz, it must only pass frequencies within the specified portion of the human voice range which is between 300 Hz to 3 kHz.

Since the filter had a length, it was an FIR filter. We used the windowing method to create an FIR filter. To create the filter, we first create the IIR impulse response for the desired ideal passband filter. Then we truncate it with a length of 10. Here is each step how we obtained the FIR filter impulse response **(4)**:

Ideal LPF with cutoff frequency 3kHz:

$$h_{ideal2}[n] = 2 \cdot \omega_2 \cdot sync(2 \cdot \omega_2 \cdot (n)) \qquad \textbf{(1)}$$

Ideal LPF with cutoff frequency 300Hz:

$$h_{ideal1}[n] = 2 \cdot \omega_1 \cdot sync(2 \cdot \omega_1 \cdot (n)) \qquad \textbf{(2)}$$

Ideal bandpass filter with cutoffs 300Hz and 3kHz:

$$h_{idealBPF}[n] = 2 \cdot \omega_2 \cdot sync(2 \cdot \omega_2 \cdot (n)) - 2 \cdot \omega_1 \cdot sync(2 \cdot \omega_1 \cdot (n)) \quad \textbf{(3)}$$

Windowed FIR bandpass filter with length 10:

$$h_{FIRBPF}[n] = 2 \cdot \omega_2 \cdot sync(2 \cdot \omega_2 \cdot (n - M)) - 2 \cdot \omega_1 \cdot sync(2 \cdot \omega_1 \cdot (n - M)) \quad \textbf{(4)}$$

; where n is from 0 to 9 (makes filter length 10)

; where M is the symmetric center of the filter (M = 4.5)

The term M ensures a symmetric impulse response, therefore a linear phase for the impulse response. Here is the calculated impulse response from **(4) (Figure 1)**:

```
Command Window
New to MATLAB? See resources for Getting Started.
>> h = 2 * w2 * sinc(2 * w2 * (n - M)) - 2 * w1 * sinc(2 * w1 * (n - M))

h =

   -0.0921   -0.0437    0.0603    0.1797    0.2590    0.2590    0.1797    0.0603   -0.0437   -0.0921

fx >>
```

Figure 1: Impulse Response of the filter

Here is the plot of the impulse response **(Figure 2)**:

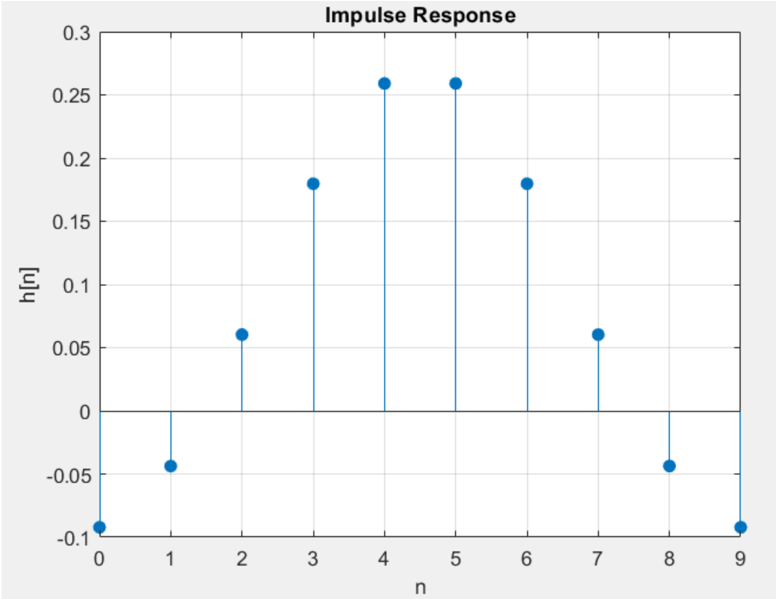

Figure 2: Impulse Response of the filter

The z-transform of the impulse response is pretty straightforward to calculate:

$$H(z) = h[n] \cdot z^{-n}$$

$$H(z) = -0.0921 - 0.0437 \cdot z^{-1} + 0.0603 \cdot z^{-2} + 0.1797 \cdot z^{-3} + 0.2590 \cdot z^{-4} + 0.2590 \cdot z^{-5} + 0.1797 \cdot z^{-6} + 0.0603 \cdot z^{-7} - 0.0437 \cdot z^{-8} - 0.0921 \cdot z^{-9}$$

Here is the pole-zero diagram of the system **(Figure 3)**:

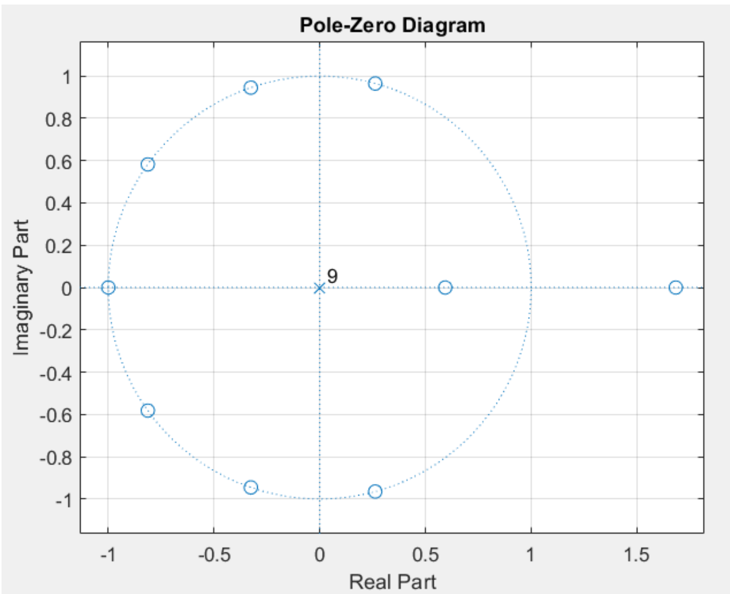

Figure 3: Pole-Zero Diagram

Here is the magnitude and the phase plots of the filter **(Figure 4)**:

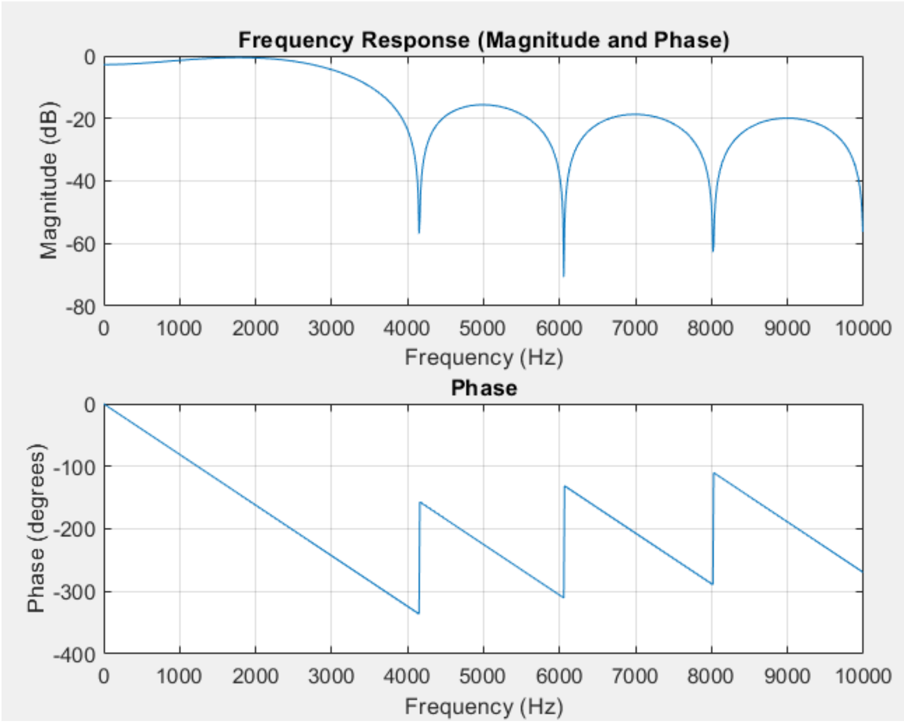

Figure 4: Magnitude and Phase Graphs

Our aim was to create a linear phase length-10 FIR bandpass filter with cutoff frequencies 300Hz and 3kHz. The resulting filter has a linear phase due to the symmetric impulse response. It is length 10. However, it is not an ideal bandpass filter due to length limitations. The passband roughly aligns with the desired shape, but it is not perfect.

The short length 10, gives poor frequency resolution. The transition band is not sharp near the cutoff frequencies, instead it is quite horizontal and does not show ideal characteristics. Some gain leakage outside 300Hz-3kHz band is unstoppable with the requirements. The filter broadly satisfies the passband conditions; however, it is far from ideal.

It is possible to make the filter closer to ideal. We chose a rectangular window to truncate the infinite impulse response of the ideal bandpass filter. We could try to choose another window (i.e. Hamming window or Blackman window) to improve the filter within the given constraints.

Also increasing the length of the filter would be a significant improvement to the passband characteristics of the filter. To illustrate the importance of length, here is the magnitude and phase responses of the same filter with a truncated length of 1000 **(Figure 5)**:

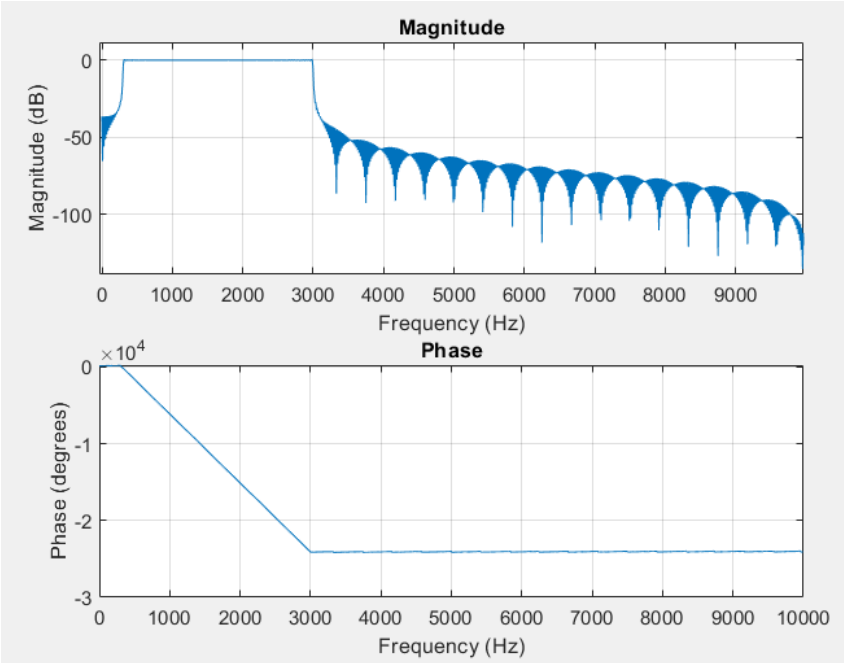

Figure 5: Magnitude and Phase Graphs for the Filter with Length 1000

If you compare figures 4 and 5, it is very obvious that the length of the filter is the most significant factor for the non-ideal nature of the filter. Increased length provides better filtering, but it also increases the computational costs significantly.

Question 2

The question asks us to record our own voice and do some audio processing on our voice. I recorded my voice in my computer, which recorded the voice with a sampling rate of 48kHz. I recorded my voice for 11.3 seconds. Using matlab, I truncated the number of bits to 480 000, in order to obtain exactly 10 seconds of my speech.

Then, it was time to resample my voice to 20kHz. The resampling rate for my voice was 5/12. I first upsampled the original audio file by 5. This means to append 4 zero bits after every single bit in the original audio. After upsampling, I obtained an array consisting of 2.4 million bits. Then, it was time for part 2. I downsampled the 2.4 million bits vector in Matlab to a new vector consisting of 200000 bits. To do this, I took the arithmetic average of each 12 bits in the upsampled audio file, then I have rounded the arithmetic average to the nearest integer value and written it to each bit in the new 200000 bits file. This was the resampled file which has a sampling rate of 20kHz and lasted 10 seconds.

Then pretty straightforwardly, I did a one-line convolution operation to obtain the filtered audio. I convolved the impulse response from part 1 with the resampled 20kHz audio file.

I had three audio files. File 1 was my recording with 48kHz sampling rate. File 2 was the resampled audio file. File 3 was the filtered version of the resampled audio file. Here you can see the FFT of the three audio files **(Figure 6)**.

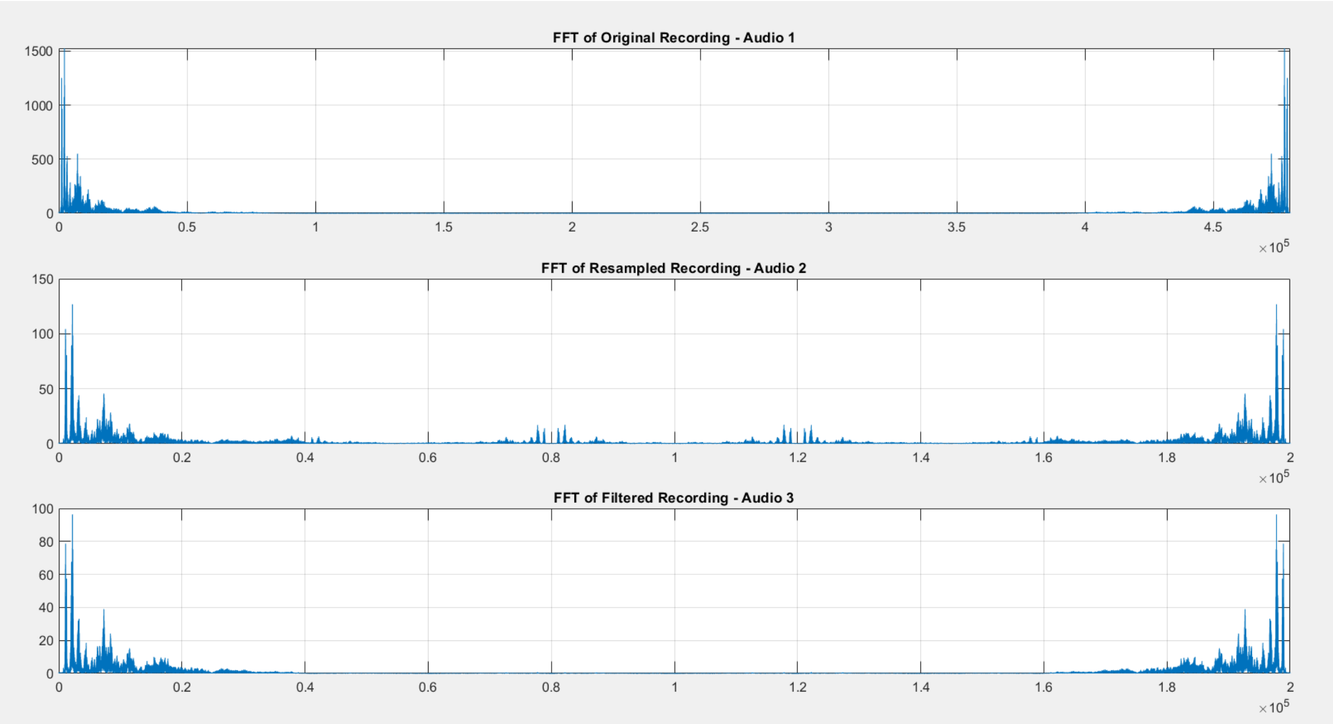

Figure 6: The FFT of Three Audio Files

As it can be seen in the above figure, 2nd audio had sharp and disturbing voices when I listened to it. This is because the resampling operation we conducted was very conducted blindly. Taking the arithmetic average of some bits in the original file and rewriting them to a new file definitely messed up the frequency characteristics of the 2nd audio. Also, we can observe that our filter had done very well with removing the higher frequencies in the 2nd audio file.

Another comment I want to make is that 1st and 3rd audio felt very alike when I listened to them. Indeed the 1st audio has more precision when digitalizing sound because of a higher sampling rate. However, the nyquist frequency for human voice is maximum 6kHz. 1st and 3rd audio's sampling rates are much above the nyquist frequency. Therefore, the low sampling rate did not make much of a difference in terms of aliasing human voice.

Conclusion & Comments

This assignment consisted of two questions. In the first part, we designed a linear phase FIR bandpass filter with a length 10. The filter was aimed to have a passband in the range of human voice which is between 300Hz and 3kHz. In the second part of the assignment, we tested the filter we created in part 1. We recorded our own voice and used the filter on that audio file. We also carried out manual resampling operation in part 2.

The assignment was a total success, all the requirements were met, constraints were followed, and observations were made. I believe this lab was very helpful for the course purposes. We understood significant characteristics of linear phase FIR filtering and also audio processing in Matlab. I think we should have more than 2 coding assignments throughout the semester. These coding assignments really help us learn a lot of things.

Appendices

1. https://github.com/fmcetin7/Bilkent-EEE-424/blob/main/Coding%20Assignment%202/codass2.m