Assignment 3

Batuhan Faik Derinbay derinbay18@itu.edu.tr
150180705

Question 1

(a)

$$
\begin{aligned}
y[n] &= \sum_{k=0}^{M} b_k x[n-k] \Rightarrow b_k = 1,1,1,1,1 \\
H(\hat{w}) &= \sum_{k=0}^{M} b_k e^{-j\hat{w}k}
\end{aligned}
$$

With the properties given above, frequency response of the system can be obtained as follows:

$$
\begin{aligned}
H(e^{-j\hat{w}k}) &= 1 + e^{-j\hat{w}} + e^{-j\hat{w}2} + e^{-j\hat{w}3} + e^{-j\hat{w}4} \\
&= e^{-j\hat{w}2}(e^{j\hat{w}2} + e^{j\hat{w}} + 1 + e^{-j\hat{w}} + e^{-j\hat{w}2}) \\
&= e^{-j\hat{w}2}(1 + 2cos(\hat{w}) + 2cos(2\hat{w}))
\end{aligned}
$$

(b) The frequency response is periodic and the period of the frequency response is 2π since period of the cosine terms is 2π.

(c) The magnitude and the phase plots can be observed in Figures 1 and 2. Note that all important points are labeled via grids on the plots.

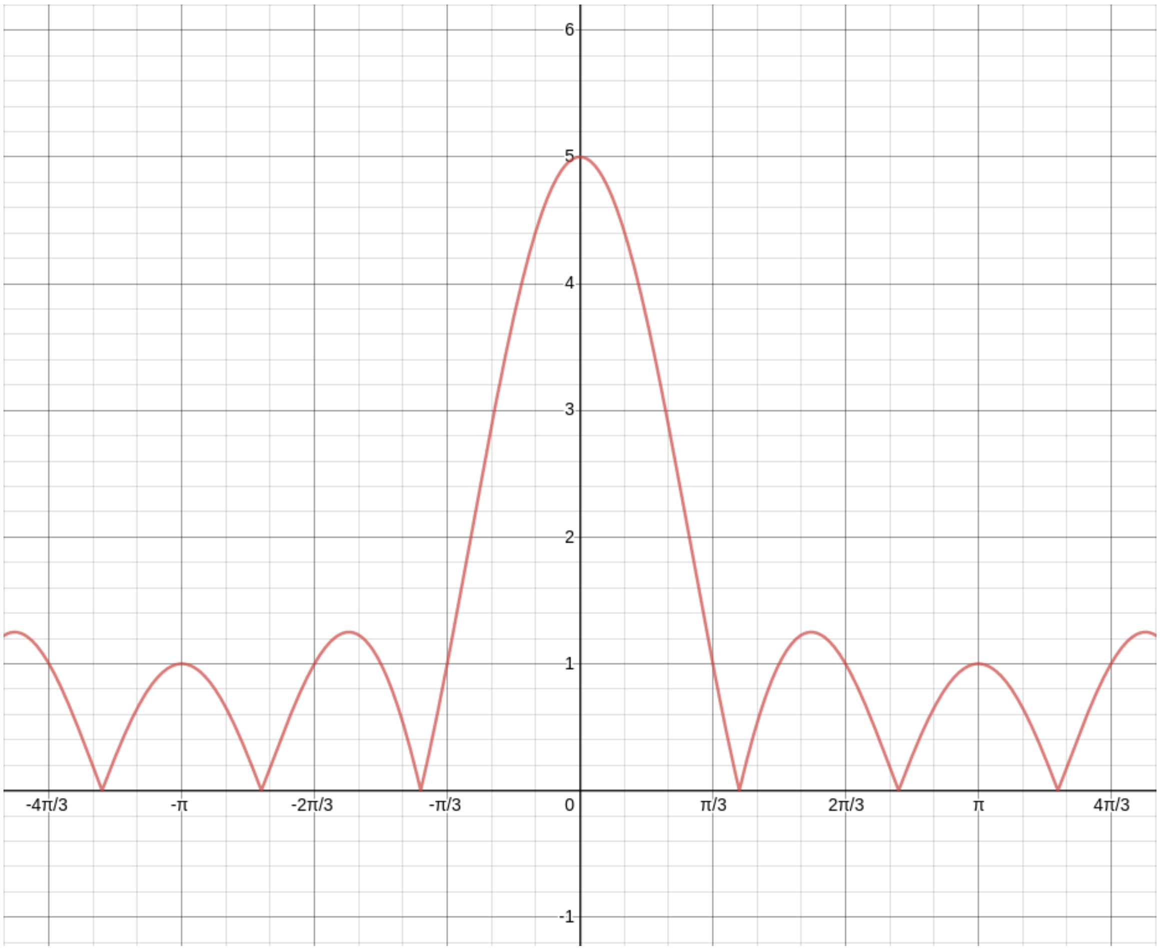

Figure 1: Plot of Magnitude of Frequency Response Between $-\pi \leq \hat{w} \leq \pi$

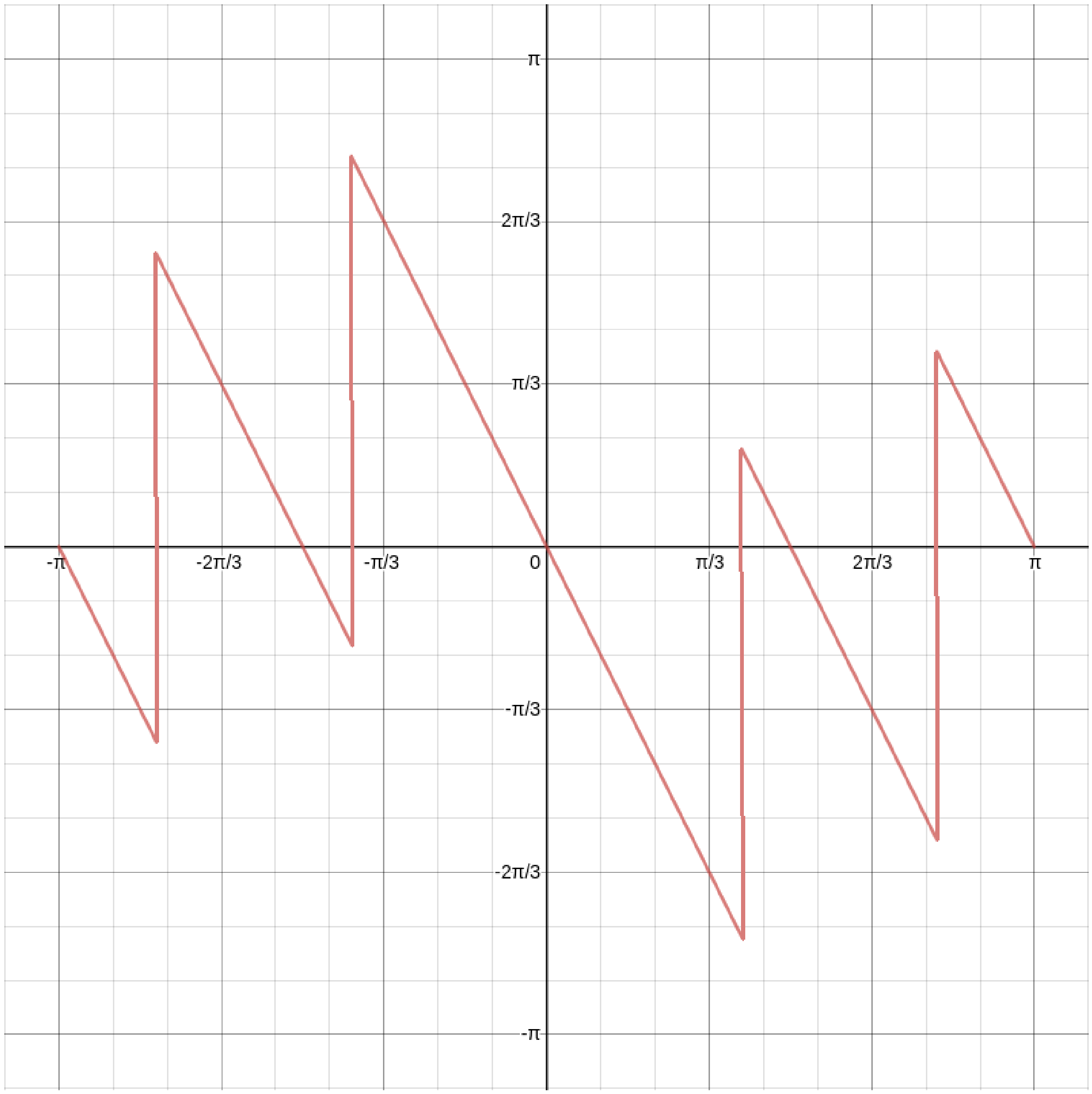

Figure 2: Plot of Phase of Frequency Response Between $-\pi \leq \hat{w} \leq \pi$

(d)

$$y[n] = \sum_{k=0}^{M} b_k x[n-k] = \sum_{k=0}^{M} b_k \delta[n-k] \Rightarrow b_k = 1, 1, 1, 1, 1$$

$$H(\hat{w}) = \sum_{k=0}^{M} b_k e^{-j\hat{w}k} = \sum_{k=0}^{M} h[k] e^{-j\hat{w}k}$$

With the properties given above, impulse response of the system can be obtained as follows:

$$h[n] = \delta[n] + \delta[n-1] + \delta[n-2] + \delta[n-3] + \delta[n-4]$$

It should be noted that linear time invariant (LTI) systems' impulse responses can be denoted in terms of its frequency response.

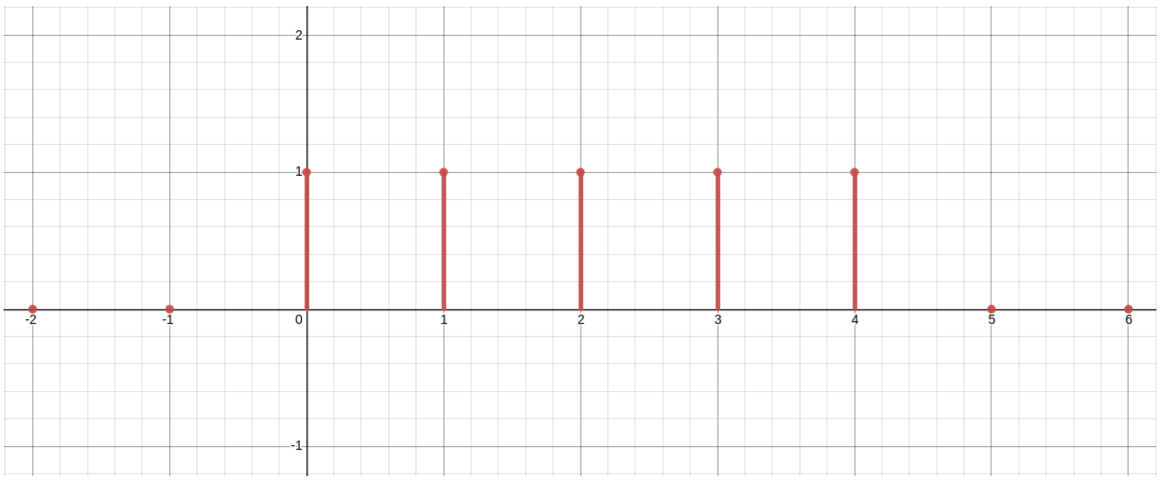

Figure 3: Plot of the Impulse Response

(e)

$$H(e^{j0}) = 5$$
$$H(e^{j\frac{\pi}{2}}) = e^{-j\pi}(1 + 0 + 2cos(-\pi)) = -1 \times -1 = 1$$
$$H(e^{j\frac{3\pi}{10}}) = e^{-j\frac{3\pi}{5}}(1 + 2cos(\frac{3\pi}{10}) + 2cos(\frac{3\pi}{5})) = 1.558e^{-j\frac{3\pi}{5}}$$

Frequency response of the system is calculated as shown above for different frequencies in the input signal. The output $y[n]$ can then be given as follows:

$$y[n] = 4 \times 5 + 2cos(\frac{\pi}{2}n - \frac{\pi}{2}) - 1.558 \times 3cos(\frac{3\pi}{10}n - \frac{6\pi}{10})$$
$$= 20 + 2cos(\frac{\pi}{2}(n-1)) - 4.674cos(\frac{3\pi}{10}(n-2))$$

(f) The image will get blurred because even though far from ideal, this is a low pass filter and low pass filters smoothens out abrupt changes, in the case of images these are RGB values of the pixels. This smoothness is a a result of the low pass filter that attenuates the higher frequencies in the signal, which correspond to edge like sudden changes in the image.

Question 2

i) For this question the LTI system given in Figure 4 will be designed. Low pass filter (LPF) $h_1[n]$ and high pass filter (HPF) $h_2[n]$ are chosen as follows:

$$h_1[n] = \frac{1}{2}\delta[n] + \frac{1}{2}\delta[n-1] + \delta[n-2] + \frac{1}{2}\delta[n-3] + \frac{1}{2}\delta[n-4]$$
$$h_2[n] = \frac{1}{2}\delta[n] - \frac{1}{2}\delta[n-1] + \delta[n-2] - \frac{1}{2}\delta[n-3] + \frac{1}{2}\delta[n-4]$$

$$H_1(e^{j\hat{w}}) = e^{-j2\hat{w}}(1 + cos(x) + cos(2x))$$
$$H_2(e^{j\hat{w}}) = e^{-j2\hat{w}}(1 - cos(x) + cos(2x))$$

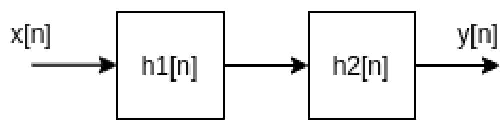

Figure 4: Designed LTI System

ii)

$$\begin{aligned}
h_1[n] * h_2[n] &= (\frac{1}{2}\delta[n] + \frac{1}{2}\delta[n-1] + \delta[n-2] + \frac{1}{2}\delta[n-3] + \frac{1}{2}\delta[n-4]) \\
&* (\frac{1}{2}\delta[n] - \frac{1}{2}\delta[n-1] + \delta[n-2] - \frac{1}{2}\delta[n-3] + \frac{1}{2}\delta[n-4]) \\
&= \frac{1}{4}(\delta[n] - \delta[n-1] + 2\delta[n-2] - \delta[n-3] + \delta[n-4] + \delta[n-1] - \delta[n-2] \\
&+ 2\delta[n-3] - \delta[n-4] + \delta[n-5] + 2\delta[n-2] - 2\delta[n-3] + 4\delta[n-4] - 2\delta[n-5] \\
&+ 2\delta[n-6] + \delta[n-3] - \delta[n-4] + 2\delta[n-5] - \delta[n-6] + \delta[n-7] + \delta[n-4] \\
&- \delta[n-5] + 2\delta[n-6] - \delta[n-7] + \delta[n-8]) \\
&= \frac{1}{4}\delta[n] + \frac{3}{4}\delta[n-2] + \delta[n-4] + \frac{3}{4}\delta[n-6] + \frac{1}{4}\delta[n-8] \\
y_1[n] &= \frac{1}{4}x[n] + \frac{3}{4}x[n-2] + x[n-4] + \frac{3}{4}x[n-6] + \frac{1}{4}x[n-8]
\end{aligned}$$

$$
\begin{aligned}
H_1(e^{j\hat{w}})H_2(e^{j\hat{w}}) &= e^{-j2\hat{w}}(1+cos(x)+cos(2x))e^{-j2\hat{w}}(1-cos(x)+cos(2x)) \\
&= e^{-j4\hat{w}}(1+\frac{1}{2}e^{2j\hat{w}}+\frac{1}{2}e^{j\hat{w}}+\frac{1}{2}e^{-j\hat{w}}+\frac{1}{2}e^{-2j\hat{w}}) \\
&\times(1+\frac{1}{2}e^{2j\hat{w}}-\frac{1}{2}e^{j\hat{w}}-\frac{1}{2}e^{-j\hat{w}}+\frac{1}{2}e^{-2j\hat{w}}) \\
&= \frac{1}{4}e^{-j4\hat{w}}(e^{j4\hat{w}}+3e^{j2\hat{w}}+4+e^{-j2\hat{w}}+e^{-j4\hat{w}}) \\
&= \frac{1}{4}+\frac{3}{4}e^{-2j\hat{w}}+e^{-4j\hat{w}}+\frac{3}{4}e^{-6j\hat{w}}+\frac{1}{4}e^{-8j\hat{w}} \\
y_2[n] &= \frac{1}{4}x[n]+\frac{3}{4}x[n-2]+x[n-4]+\frac{3}{4}x[n-6]+\frac{1}{4}x[n-8]
\end{aligned}
$$

Since $y_1[n] = y_2[n]$, it is obvious that convolution in the time domain is equivalent to multiplication in the frequency domain.

iii) The system designed is a poor band pass (nullifying) filter. Since cutoff frequency of the LPF is approximately $\frac{\pi}{3}$ and cutoff frequency of the HPF is approximately $\frac{2\pi}{3}$, the expected signal should have frequencies with their magnitudes around 0. In other words, outcome of this filter should be null. However, since the filters are not ideal, outcome of this filter has peaks at frequencies around $-\pi, 0, \pi$ as expected. Magnitudes of the filters and the output can be observed in Figure 5. Red dashed line represents the LPF, blue dashed line represents the HPF and green line represents the output.

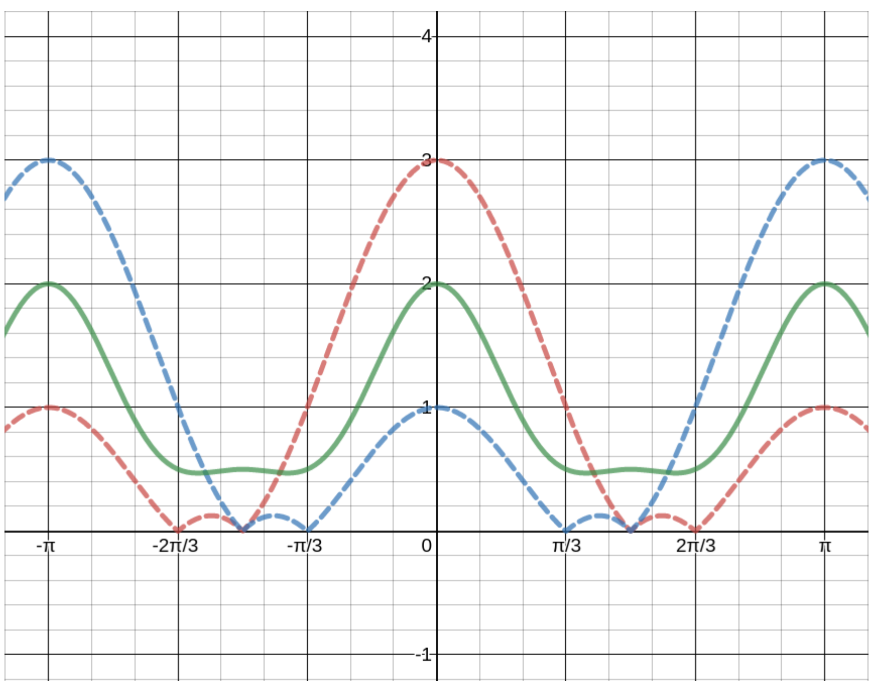

Figure 5: Magnitudes of Filters $H_1(e^{j\hat{w}})$, $H_2(e^{j\hat{w}})$ and the Output

Question 3

i)

$$|X(e^{j\hat{w}})| \leq \sum_{n=-\infty}^{\infty} |x[n]| < \infty$$

By using the sufficient condition for existence of the discrete time fourier transform (DTFT) given above, existence of DTFT for $x_1[n]$ is checked. Then, using the linearity and differentiation in frequency properties of DTFT, following expression is obtained.

$$X_1(e^{j\hat{w}}) = n(\frac{1}{3})^n u[n] e^{-j\hat{w}n} + e^{j\frac{3\pi}{10}n} e^{j\frac{\pi}{4}} e^{-j\hat{w}n} \wedge (\frac{1}{3})^n u[n] \Leftrightarrow \frac{1}{1-\frac{1}{3}e^{-j\hat{w}}}$$

$$= j\frac{\mathrm{d}\frac{1}{1-\frac{1}{3}e^{-j\hat{w}}}}{\mathrm{d}\hat{w}} + e^{j\frac{\pi}{4}} e^{j\frac{3\pi}{10}n} = \frac{3e^{j\hat{w}}}{(1-3e^{j\hat{w}})^2} + e^{j\frac{\pi}{4}} 2\pi\delta(\hat{w} - \frac{3\pi}{10}) \text{ with } -\pi < \hat{w},\ \frac{3\pi}{10} \leq \pi$$

ii) Existence of DTFT check is applied to $x_2[n]$. DTFT exists, so it can be obtained. $x_2[n]$ is periodic with every seven samples. Therefore by using this knowledge of the signal and the forward property, DTFT of $x_2[n]$ can be calculated as follows:

$$X_2(e^{j\hat{w}}) = \sum_{n=-\infty}^{\infty} (3-|n|)e^{-j\hat{w}n} \rightarrow \sum_{n=-3}^{3} (3-|n|)e^{-j\hat{w}n}$$

$$= e^{j\hat{w}2} + 2e^{j\hat{w}} + 3 + 2e^{-j\hat{w}} + e^{-j\hat{w}2} = e^{j\hat{w}2}(1 + e^{-j\hat{w}} + e^{-j\hat{w}2} + e^{-j\hat{w}3} + e^{-j\hat{w}4})$$

$$= 3 + 4cos(\hat{w}) + 2cos(2\hat{w})$$

iii) Existence of DTFT check is applied to $x_3[n]$. DTFT exists, so it can be obtained. By using the linearity property of DTFT, following expression is obtained.

$$x_3[n] = x_{3_1} + x_{3_2} = \frac{sin(0.8\pi n)}{\pi n} - \frac{sin(0.5\pi n)}{\pi n}$$

$$X_{3_1}(e^{j\hat{w}}) = \begin{cases} 1, & |\hat{w}| \leq 0.8\pi \\ 0, & 0.8\pi < |\hat{w}| \leq \pi \end{cases}$$

$$X_{3_2}(e^{j\hat{w}}) = \begin{cases} -1, & |\hat{w}| \leq 0.5\pi \\ 0, & 0.5\pi < |\hat{w}| \leq \pi \end{cases}$$

$$X_3(e^{j\hat{w}}) = X_{3_1}(e^{j\hat{w}}) + X_{3_2}(e^{j\hat{w}}) = \begin{cases} 1, & 0.5 < |\hat{w}| \leq 0.8\pi \\ 0, & |\hat{w}| \leq 0.5\pi \wedge 0.8\pi < |\hat{w}| \leq \pi \end{cases}$$

Question 4

(a)

$$X(e^{j\hat{w}}) = \frac{2+\frac{1}{4}e^{-j\hat{w}}}{-(\frac{e^{-j\hat{w}}}{4}-1)(\frac{e^{-j\hat{w}}}{2}+1)} = -\left(\frac{A}{(\frac{e^{-j\hat{w}}}{4}-1)} + \frac{B}{(\frac{e^{-j\hat{w}}}{2}+1)}\right) \Rightarrow A = 1 \wedge B = -1$$

$$= \frac{-1}{\frac{e^{-j\hat{w}}}{4}-1} + \frac{1}{\frac{e^{-j\hat{w}}}{2}+1} = \frac{1}{1-\frac{1}{4}e^{-j\hat{w}}} + \frac{1}{1-\frac{-1}{2}e^{-j\hat{w}}}$$

$$x[n] = (\frac{1}{4})^n u[n] + (\frac{-1}{2})^n u[n] = (4^{-n} + (-2)^{-n})u[n]$$

(b)

$$X(e^{j\hat{w}}) = \begin{cases} 1, & 0.25 < |\hat{w}| \leq 0.75\pi \\ 0, & |\hat{w}| \leq 0.25\pi \wedge 0.75\pi < |\hat{w}| \leq \pi \end{cases} = X_1(e^{j\hat{w}}) + X_2(e^{j\hat{w}})$$

$$X_1(e^{j\hat{w}}) = \begin{cases} 1, & |\hat{w}| \leq 0.75\pi \\ 0, & 0.25\pi < |\hat{w}| \leq \pi \end{cases} \Rightarrow x_1[n] = \frac{sin(0.75\pi n)}{\pi n}$$

$$X_2(e^{j\hat{w}}) = \begin{cases} -1, & |\hat{w}| \leq 0.25\pi \\ 0, & 0.25\pi < |\hat{w}| \leq \pi \end{cases} \Rightarrow x_2[n] = -\frac{sin(0.25\pi n)}{\pi n}$$

$$x[n] = x_1[n] + x_2[n] = \frac{sin(0.75\pi n)}{\pi n} - \frac{sin(0.25\pi n)}{\pi n} = \frac{2cos(0.5\pi n)sin(0.25\pi n)}{\pi n}$$

Question 5

a) Code below reads the "ilber_04.wav" and "esg_00.wav" files. Then extracts the left channel audios and plots their FFT magnitudes as shown in Figures 6 and 7. Please note that in order for the given code snippet to work, data given with the homework should be in the same root directory and ".wav" audio files should be under their corresponding "ilber", "esg" and "test" folders.

Reading the Audio Files and Plotting The FFT Magnitudes

```python
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal
from scipy.io import wavfile
import scipy.fftpack as fftpack

sampling_freq, ilber_sample = wavfile.read("ilber/ilber_4.wav")
sampling_freq, esg_sample = wavfile.read("esg/esg_0.wav")
ilber_sample_L = ilber_sample[:,0]; ilber_sample_R = ilber_sample[:,1]
esg_sample_L = esg_sample[:,0]; esg_sample_L = esg_sample[:,1]

freqs = fftpack.fftfreq(len(ilber_sample_L))*sampling_freq

```

```
ilber_sample_L_norm = ilber_sample_L/max(abs(ilber_sample_L))
ilber_fft = fftpack.fft(ilber_sample_L_norm)
# ilber_fft_magn = np.sqrt(np.square(ilber_fft.real)+np.square(ilber_fft.imag))
ilber_fft_magn = np.abs(ilber_fft)

plt.figure(figsize=(12, 5))
plt.title("Ilber 4 FFT Magnitude")
plt.xlabel('Frequency [Hz]')
plt.ylabel('Magnitude')
plt.xlim(-sampling_freq/2, sampling_freq/2)
plt.plot(freqs, ilber_fft_magn)

esg_sample_L_norm = esg_sample_L/max(abs(esg_sample_L))
esg_fft = fftpack.fft(esg_sample_L_norm)
# esg_fft_magn = np.sqrt(np.square(esg_fft.real)+np.square(esg_fft.imag))
esg_fft_magn = np.abs(esg_fft)

plt.figure(figsize=(12, 5))
plt.title("ESG 0 FFT Magnitude")
plt.xlabel('Frequency [Hz]')
plt.ylabel('Magnitude')
plt.xlim(-sampling_freq/2, sampling_freq/2)
plt.plot(freqs, esg_fft_magn)
```

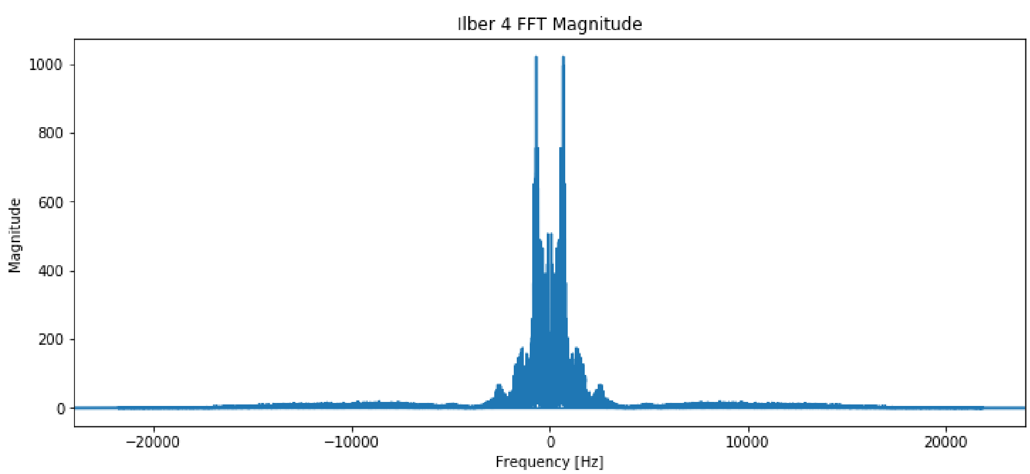

Figure 6: FFT Magnitude Plot of ”ilber_04.wav”

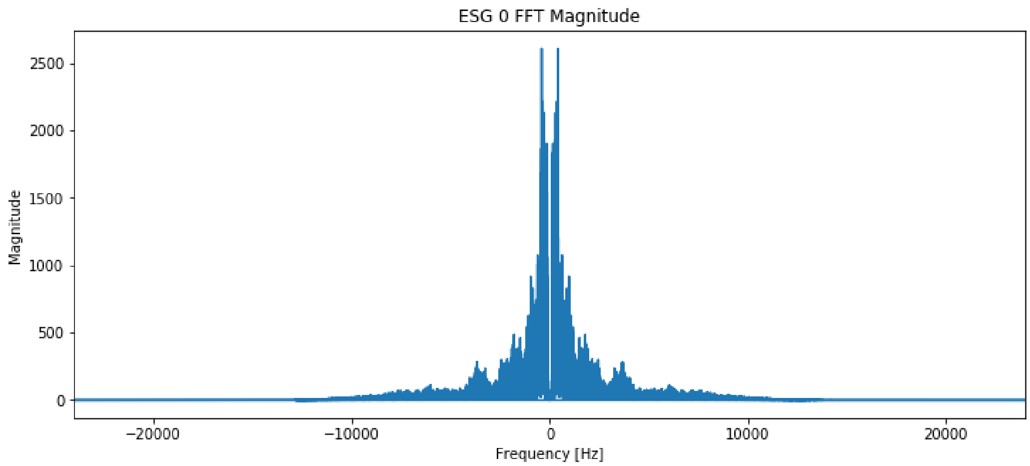

Figure 7: FFT Magnitude Plot of ”esg_0.wav”

b) Classifying audio files belonging to İlber Ortaylı and Emrah Sefa Gürkan is an easy task due to magnitudes of the FFTs. After normalizing the audio signals, means of the amplitudes are calculated for each person and tresholds are determined accordingly. Then, test signals below the obtained threshold are determined to belong to the first person, others determined to belong to the second person.

In the case of this homework, mean magnitude of İlber Ortaylı's FFT is approximately 12.286 and mean magnitude of Emrah Sefa Gürkan's FFT is approximately 33.151. So the threshold is chosen as 22.719, mean of means of FFTs. Test audio files that are below this threshold are classified as audio snippets belonging to İlber Ortaylı and other are classified as audio snippets belonging to Emrah Sefa Gürkan. Python code of this algorithm is given below.

Reading the Audio Files and Plotting The FFT Magnitudes

```
from os import listdir
from os.path import isfile, join

ilber_data = [f for f in listdir("ilber") if isfile(join("ilber",↩
     f))]
esg_data = [f for f in listdir("esg") if isfile(join("esg", f))]
test_data = [f for f in listdir("test") if isfile(join("test", f)↩
    )]

sampling_freq, ilber_sample = wavfile.read(join("ilber", ↩
    ilber_data[0]))
freqs = fftpack.fftfreq(len(ilber_sample[:,0]))*sampling_freq

ilber_total = 0
for audio_name in ilber_data:
    sampling_freq, ilber_audio = wavfile.read(join("ilber", ↩
        audio_name))
```

```
    ilber_audio_L = ilber_audio[:,0]
    ilber_audio_L_norm = ilber_audio_L/max(abs(ilber_audio_L))
    ilber_fft = fftpack.fft(ilber_audio_L_norm)
    ilber_fft_magn = np.abs(ilber_fft)

    ilber_total += np.mean(ilber_fft_magn)

ilber_mean = ilber_total/len(ilber_data)
print("Mean Magnitude of Ilber's FFT: {}".format(ilber_mean))

esg_total = 0
for audio_name in esg_data:
    sampling_freq, esg_audio = wavfile.read(join("esg", audio_name))
    esg_audio_L = esg_audio[:,0]
    esg_audio_L_norm = esg_audio_L/max(abs(esg_audio_L))
    esg_fft = fftpack.fft(esg_audio_L_norm)
    esg_fft_magn = np.abs(esg_fft)

    esg_total += np.mean(esg_fft_magn)

esg_mean = esg_total/len(esg_data)
print("Mean Magnitude of ESG's FFT: {}".format(esg_mean))

ilber_test_data_amount = 0
esg_test_data_amount = 0
total_test_data_amount = len(test_data)

for audio_name in test_data:
    if "ilber" in audio_name:
        ilber_test_data_amount += 1
    else:
        esg_test_data_amount += 1

correct_pred = 0
wrong_pred = 0

for audio_name in test_data:
    mean = (ilber_mean + esg_mean) / 2

    sampling_freq, test_audio = wavfile.read(join("test", audio_name))
    test_audio_L = test_audio[:,0]
    test_audio_L_norm = test_audio_L/max(abs(test_audio_L))
    test_fft = fftpack.fft(test_audio_L_norm)
    test_fft_magn = np.abs(test_fft)

    if np.mean(test_fft_magn) > mean:
        print("Audio file \"{}\" belongs to Emrah Sefa Gurkan".format(audio_name))
        if not "esg" in audio_name:
```

```
                wrong_pred += 1
            else:
                correct_pred +=1
        else:
            if not "ilber" in audio_name:
                wrong_pred += 1
            else:
                correct_pred +=1
            print("Audio file \"{}\" belongs to Ilber Ortayli".format(audio_name))

print("\n\nResults:\nCorrect Predictions:\t{}\nWrong Predictions:\t{}\nAmount of Test Data:\t{}\nAccuracy:\t{}%".format(correct_pred, wrong_pred, total_test_data_amount, 100*correct_pred/total_test_data_amount))
```

Results of this algorithm is summed up by the following terminal output.

```
    Results:
    Correct Predictions:    40
    Wrong Predictions:  0
    Amount of Test Data:    40
    Accuracy:   100.0\%
```