In the Name of God

University of Tehran

School of Electrical and
Computer Engineering

Digital Signal Processing

Computer Assignment #1

Amirhossein Mohammadi

81019689

Spring 2021

Table of Contents

Abstract

Goal of the computer assignment is to get familiar with digital processing of signal in both time and frequency domain. This computer assignment takes a glance look at frequency analysis of picture and music and shows the relation between phase and domain of frequency response of signal. It also shows properties of up-sampled and down-sampled signal in both time and frequency domain, at last it tries to interpolate up-sampled signal.

Question one

Part A

Odd and even part of signal are separated from original one and shown in figure 1.

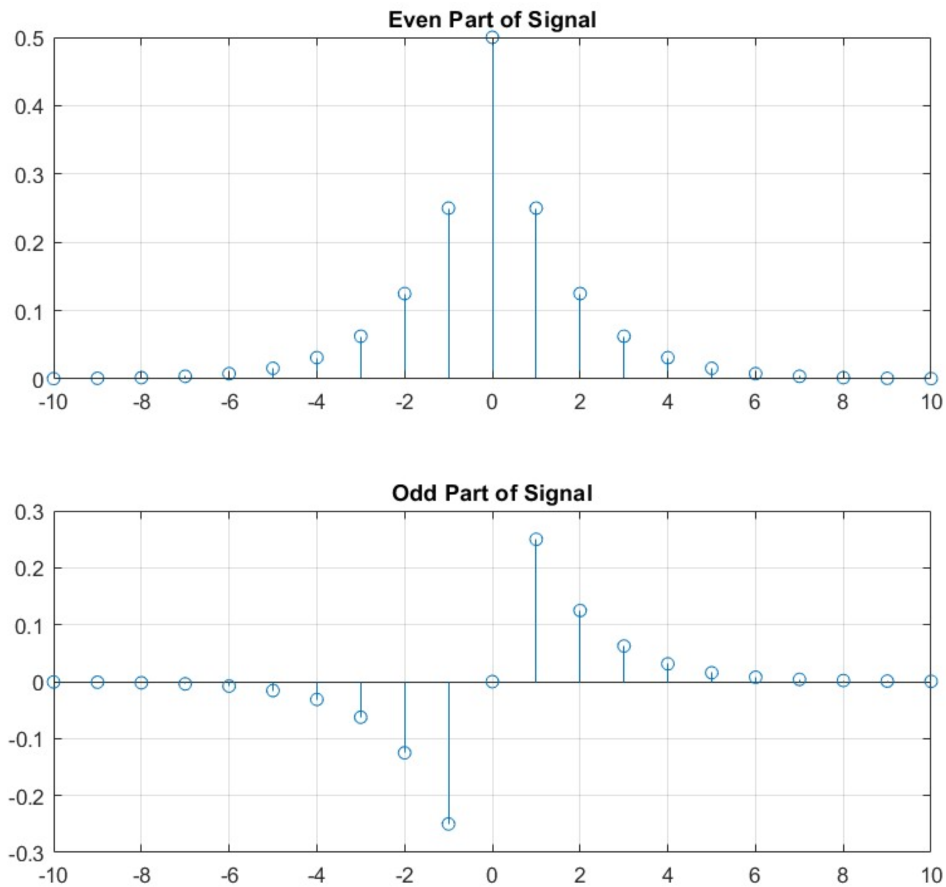


Figure 1, Even and Odd Part of x[n]

Part B

For finding **DFT** of signal FFT algorithm has been used. Signal has been plotted in $[-\pi, \pi]$ since DFT of signal is periodic with fundamental period of 2π. Since number of samples is not power of two, in FFT algorithm sampled has been padded to next power two. Therefore, result in figure two has shown FFT in 32 samples [FFT algorithm perform better if number of samples are power of two].

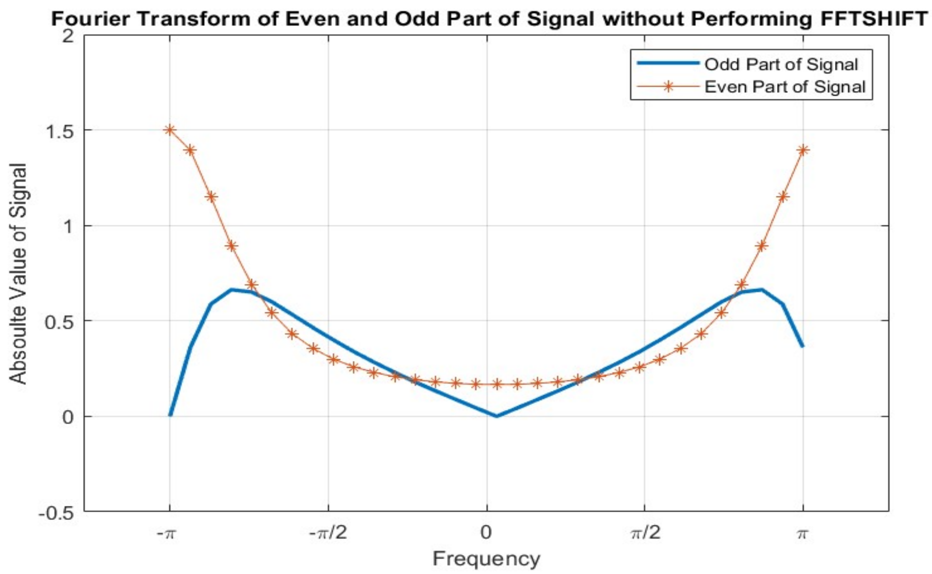


Figure 2, Fourier Transform of Even and Odd Part of Signal

As it can see in figure 2, FFT algorithm need a modification in indexing after calculating its result. Zero frequency indexes should shift to center of spectrum in order to get correct result.

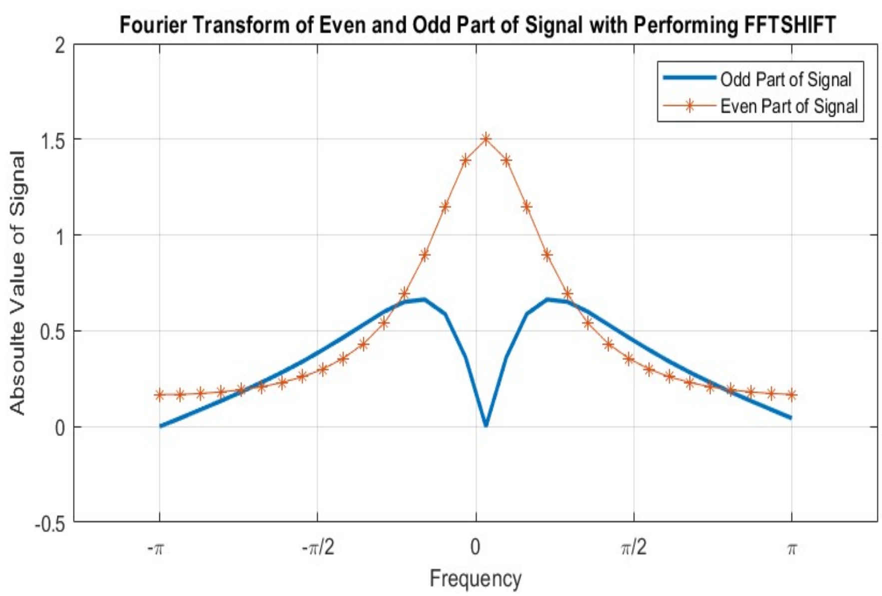


Figure 3, Fourier Transform after performing FFT shift

FFT is a function that computes Fourier of a function numerically. It's actually summation of every single index:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N} \qquad k = 0, \ldots, N-1,$$

But faster way is to decompose it to two halves and calculate them as shown in figure 1:

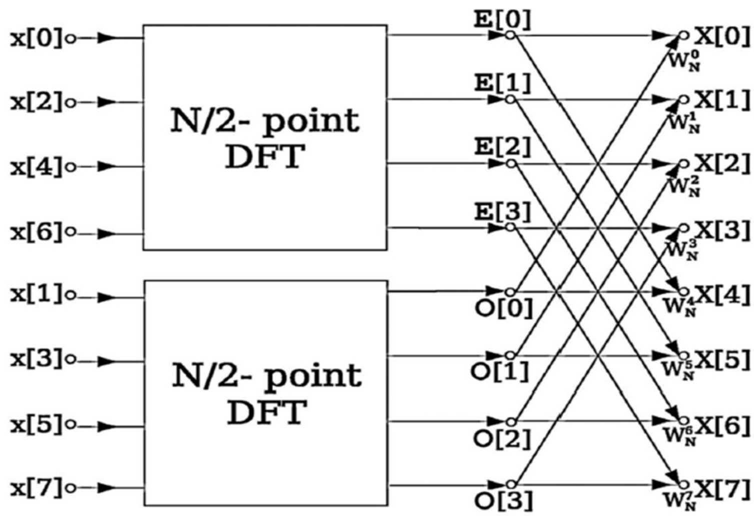


Figure 4, Structure of FFT Algorithm[1]

Figure 4 can tell why "FFTSHIFT" function is needed. Vector's indexes need to rearranged by shifting the zero-frequency to the center.

Part C

For finding P_{eror} relation below is used.

$$P_e = |\, X(e^{jw}) - X_e(e^{jw}) - X_o(e^{jw})\, |^2$$

It is expected to P_e be zero, as we can see in figure below it is near zero. summation of resulting vector also shows that.

$$P_e = \quad 4.7455 \times 10^{31}$$

Wikipedia [1]

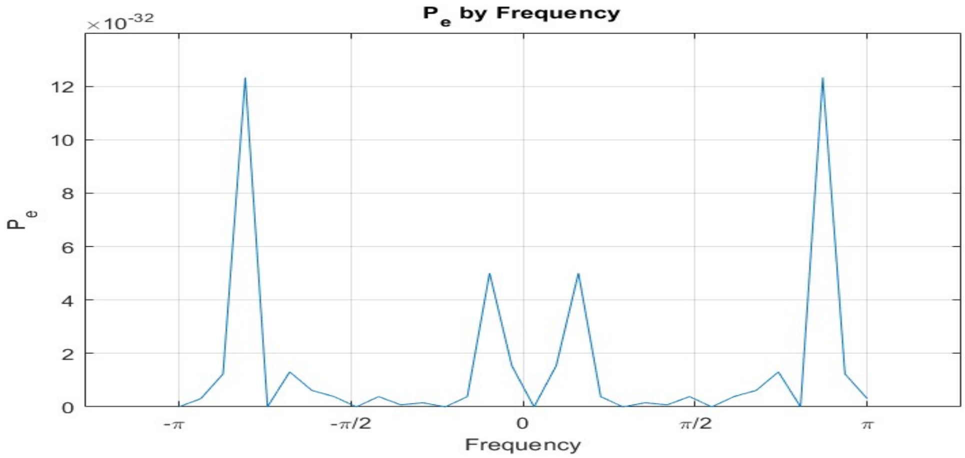


Figure 5, Power of error

Part D

It can be concluded that Fourier transform of even part of signal is also even and if signal be real to its Fourier Transform will be Purely Real and Even. On the other hand if signal be odd its Fourier Transform will also be odd either and in addition if signal be odd and real its Fourier Transform will be Purely Imaginary and odd.

Also for finding Fourier transform, signal can decomposed to two parts it real and its even part. Fourier transform of real part yields real signal and odd part, odd. From super position, Fourier transform of main signal is summation of these two.

Question 2

Part A

For finding Zero-Pole Diagram in Z-Plane '*zplane"* is used and for absolute value of signal and its phase by frequency "*freqz*" function is used.

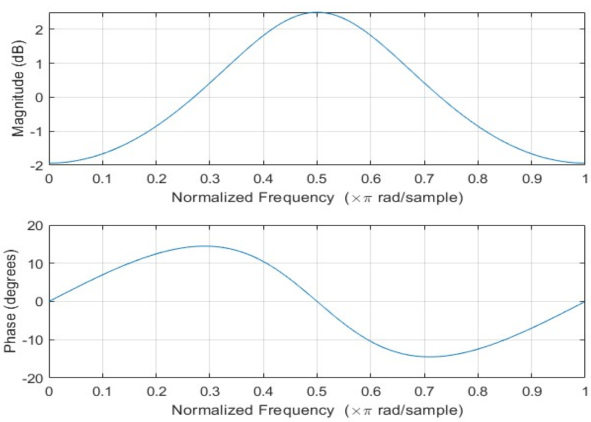


Figure 6, Phase and Magnitude of H(z)

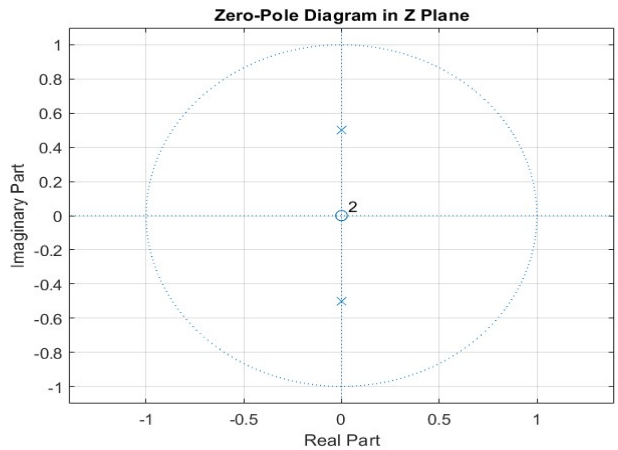


Figure 7, Zero-Pole Diagram

Part B

For finding impulse respond, firstly H(z) is decomposed into fraction of two polynomials then Z-Inverse of function will be evaluated.

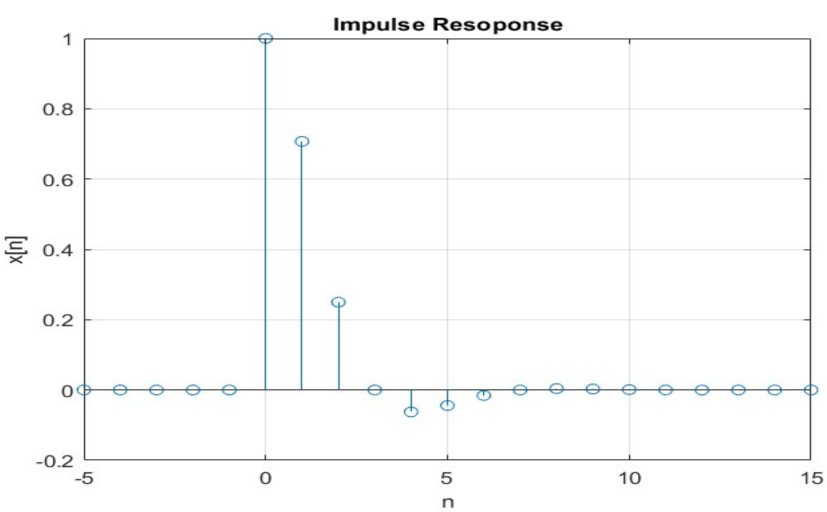


Figure 7, Impulse Response

Question 3

Part A

Fundumental period of signal is $\frac{2\pi}{0.02\pi} = 100$ and it has been plotted for 4 period in figure 8.

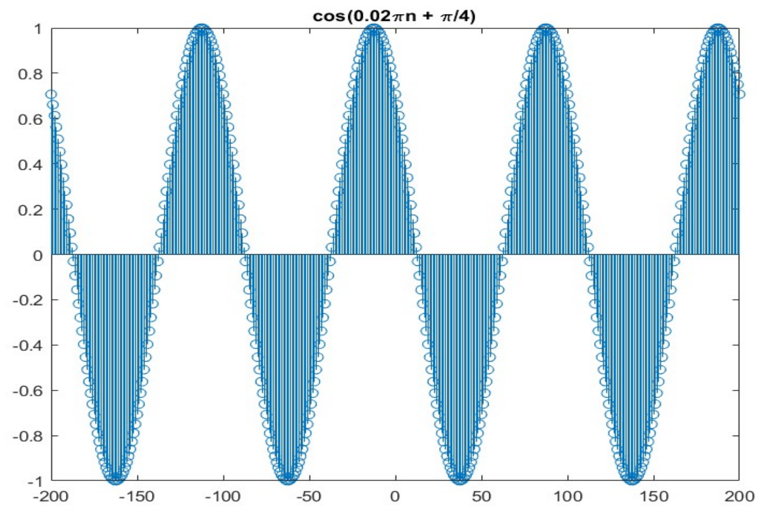


Figure 8, Cos($0.02\boldsymbol{\pi}\, \boldsymbol{n} + \frac{\boldsymbol{\pi}}{\boldsymbol{4}}$)

Part B

Plot of y[n] and $y[n] = x[n] + 3w[n]$.

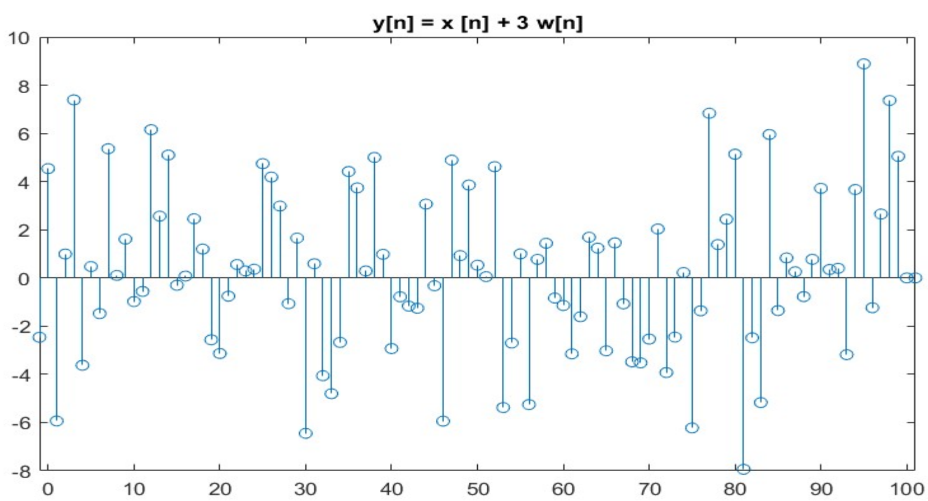


Figure 9, y[n]

Part C

$z[n]$ is defined as equation below. Since cross correlation between two signals indicate how much two signals are similar to each other, and z[n] is y[n] itself but shifted 20 times it can concluded cross correlation has its maximum value at n = 20 as it shown in figure 10.

$$z\,[n] = y[n-20] + w[n]$$

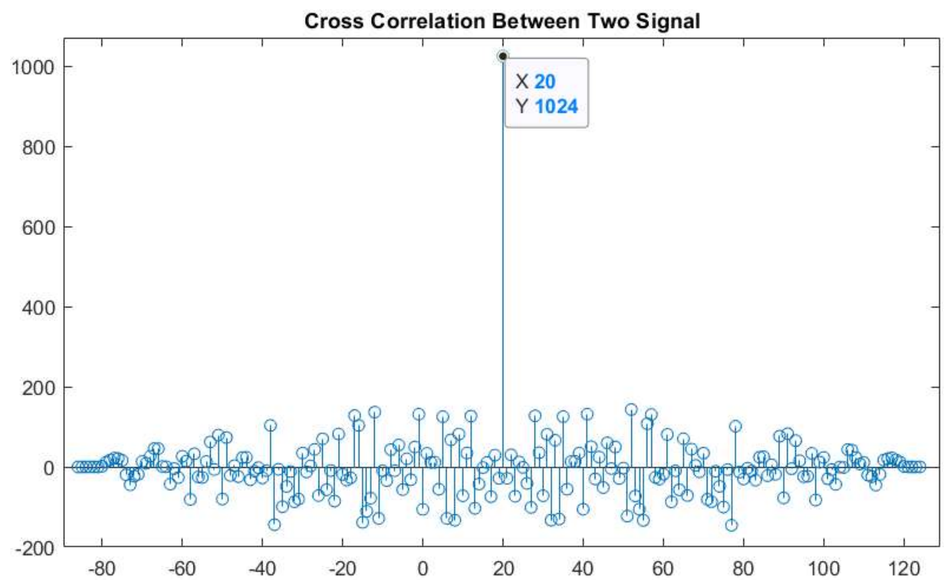


Figure 10, Cross Correlation Between two signals

Question 4

Part A, B

After saving image in MATLAB, its frequency response has shown below.

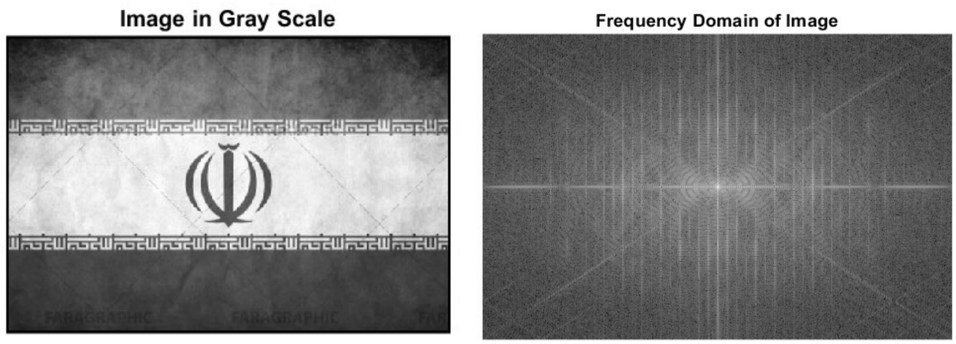


Figure 11, Picture and its Frequency Response

Part C, D E F

After adding noise to the picture we can see white and black dots appear in original picture and picture in frequency domain gets blurry. Noise has been added in three different power values: 0.01, 1, 10 dB. As we can see by increasing noise power, SNR and resolution of picture decreases.

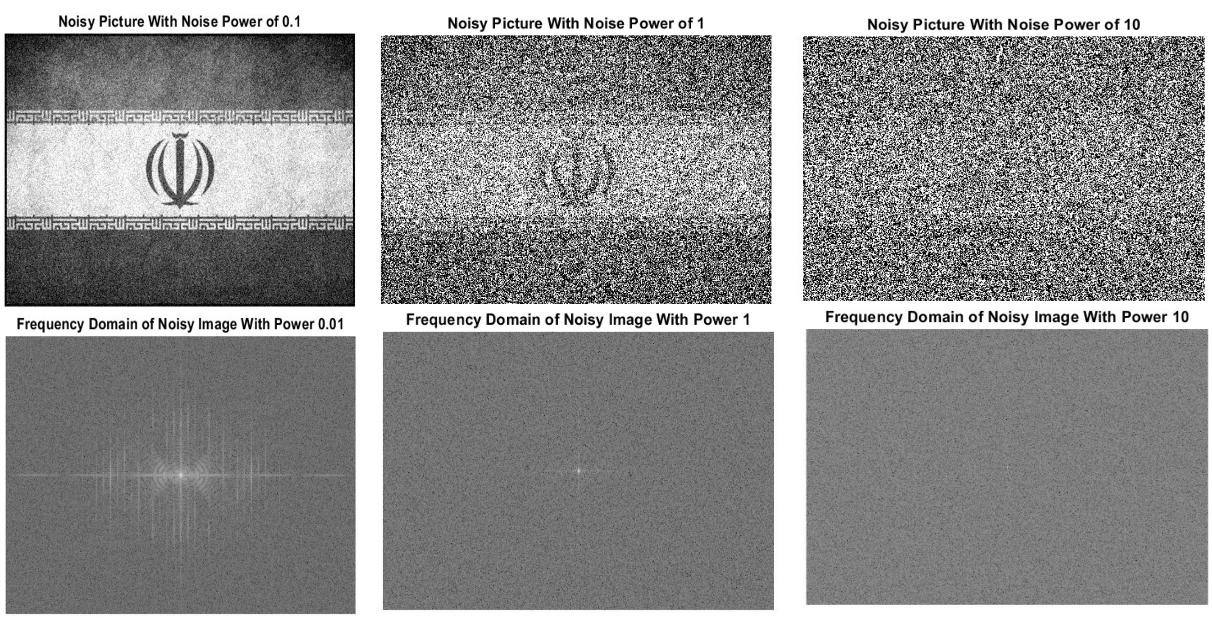


Figure 12, Noisy Picture And its Frequency Response

Question 5

After finding frequency response of both images, its amplitude and phase will be separated and switched between two pictures. As we can see in figure 15, The picture with phase first face and amplitude of second can recognized as face one. It can be concluded that phase in pictures has much important information than amplitude.

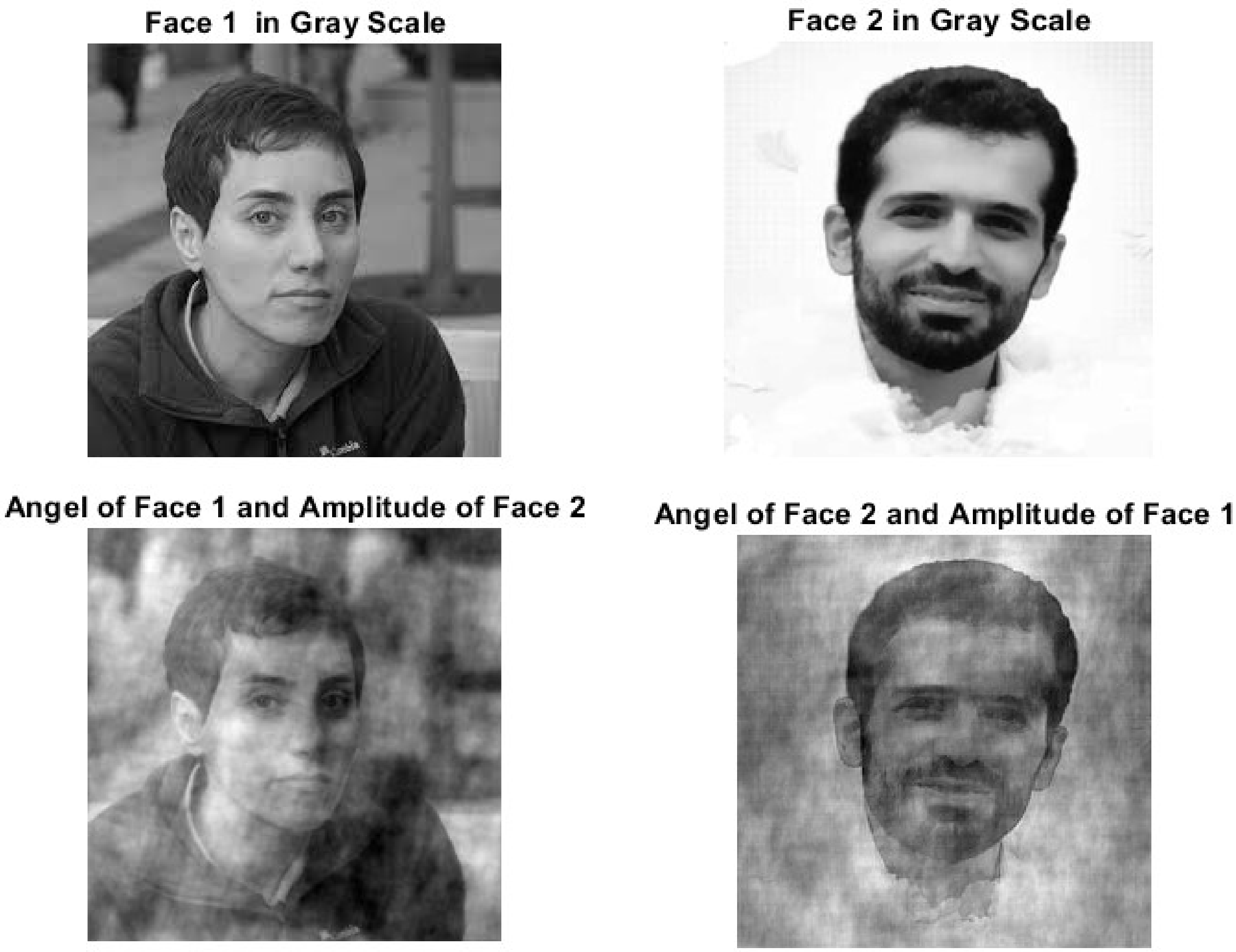


Figure 13, Pictures shown after swiching amplitutde and phase

Question 6

Part A, B, C

After reading file with MATLAB and playing it with different sampling rates it can be understood as sampling rate increases voice will be thinner (more treble) and lower sampling rate, voice will be bolder(more bass).

Part D

Performing FFT to signal result spectrum below.

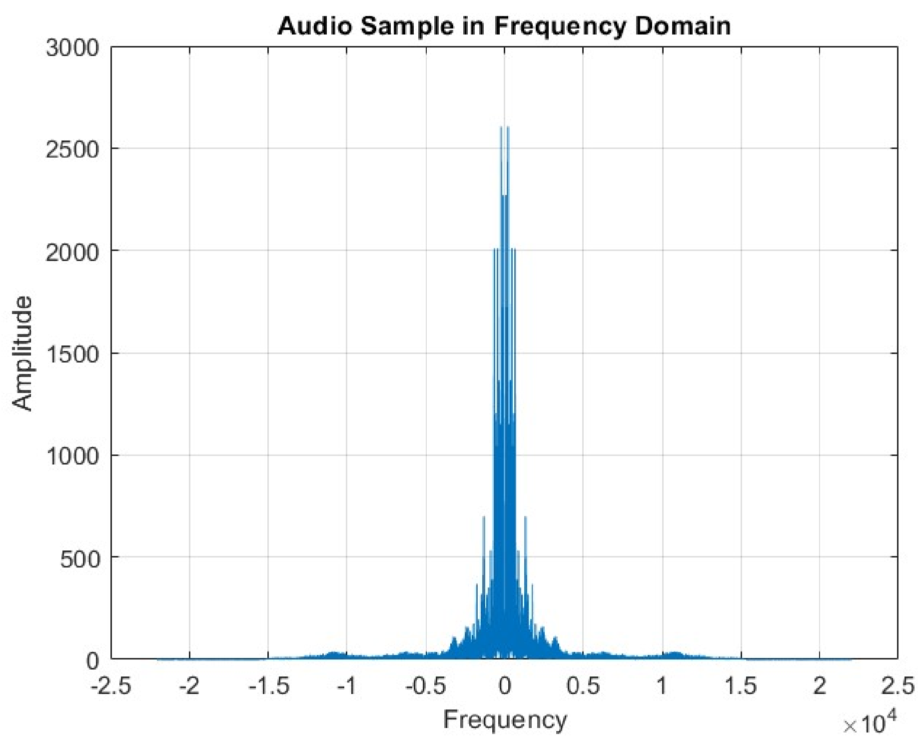
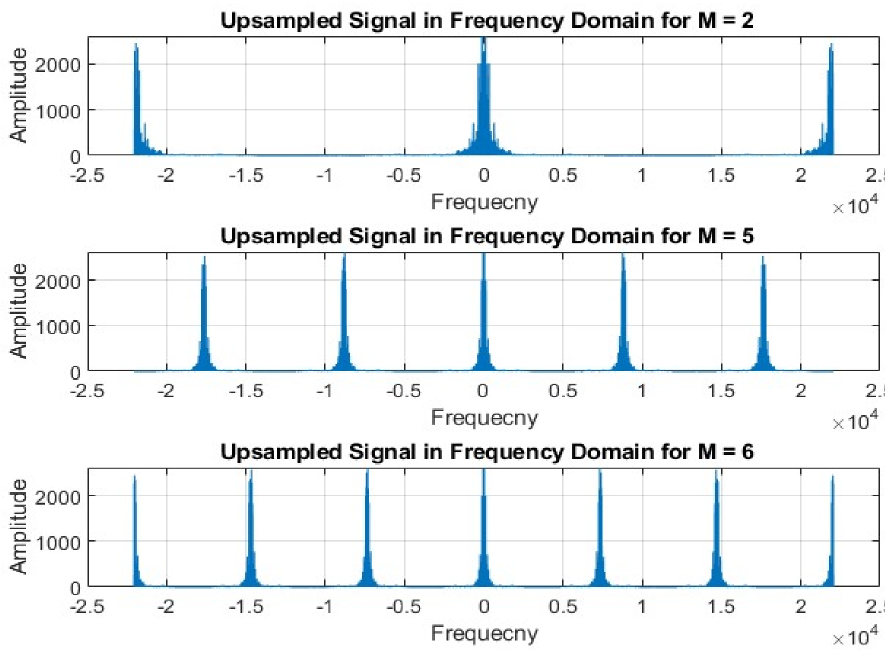


Figure 14, Audio Sample in Frequency Domain

Part E, F, G, H

For up-sampling/down-sampling signal '*my_upsample/my_downsample'* function has been written. After giving sampled audio to function and setting up-sampling/downsampling rate, FFT is performed on up-sampled signal and result is shown in figure 15 and 16.

As it can be seen up-sampling expands signal in time domain, but up-sampling shirinks signal and make it narrower in frequency domain, it also repeats signal with period of $\frac{2\pi}{m}$. In the other hand down-sampling make signal narrower in time domain and expand it in frequency domain..



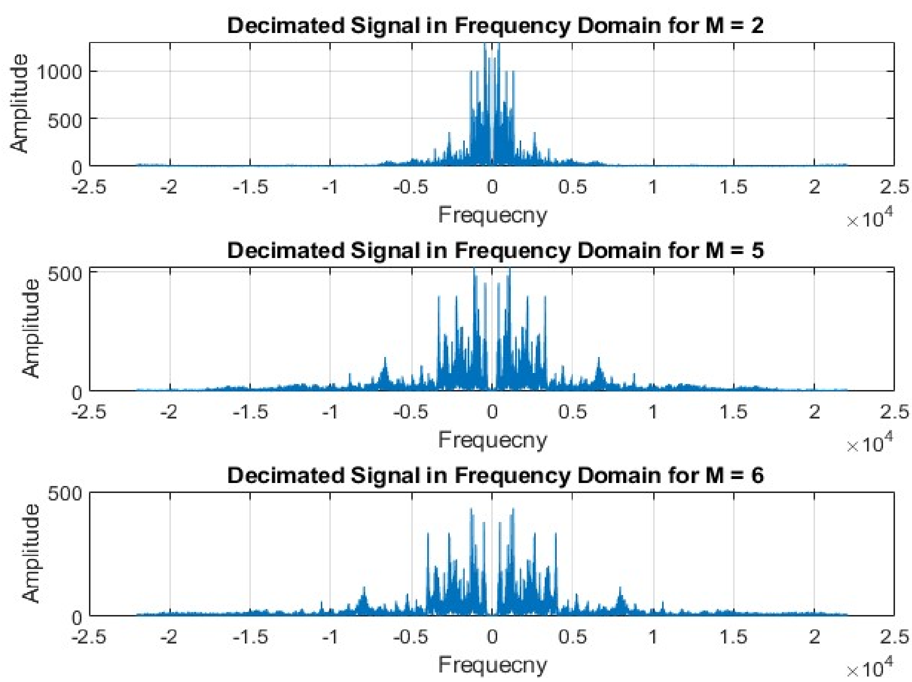


Figure 15, 16: Decimated And Upsampled Signal in Frequency Domain

Part I:

Up-sampled Vector has zeros between original elements and the goal is to fill the zeros with proper value. There are several ways to create interpolated signal like convolving with *sinc* or passing it through a low pass filter.

The method is used here is replacing zeros by evaluation of mean of every consecutive sequence in original signal and swapping in up-sampled signal.

Result shows that this method is effective and our sound is much like first sound.

Appendix: Instruction in Running Code

Each Question has its own name in code documentary.

For last question sound of each part will be played after running code, for finding which sound is playing follow the code flow.