EECE 340: Signals and Systems

Signal Processing Project with MATLAB

Abed Karim Nasreddine & Mohammad Itani

Introduction: In this project, we will be examining principal concepts from signal processing and applying them on three audio samples. For this purpose, we will be using MATLAB to configure several algorithms that will help us achieve this purpose in addition to some visualization that will be beneficial in interpreting the results obtained.

I) Time Representation

(a) In this part we will construct a time vector *t* that starts from -5 up till 5. This vector will be constructed using the command **linspace(-5,5,N)** where N refers to the number of sample points. Therefore, we have constructed a time vector *t* with N elements in it. We shall take N=1024 for our analysis.

(b) For this part, we will be examining four signals:

$x_1(t) = u(t)$.

$x_2(t) = u(t) - u(t-1)$.

$x_3(t) = e^{-2t}u(t)$.

$x_4(t) = \cos(2\pi t)$.

By using the commands **figure** and **plot** on MATLAB, we will obtain the following figures:

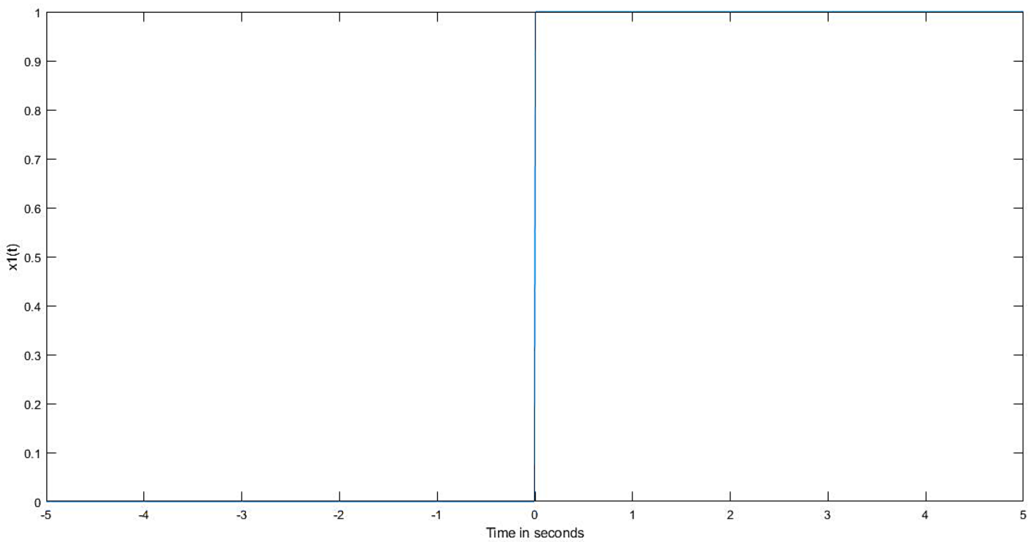

Figure-1: Graph of $x_1(t)$

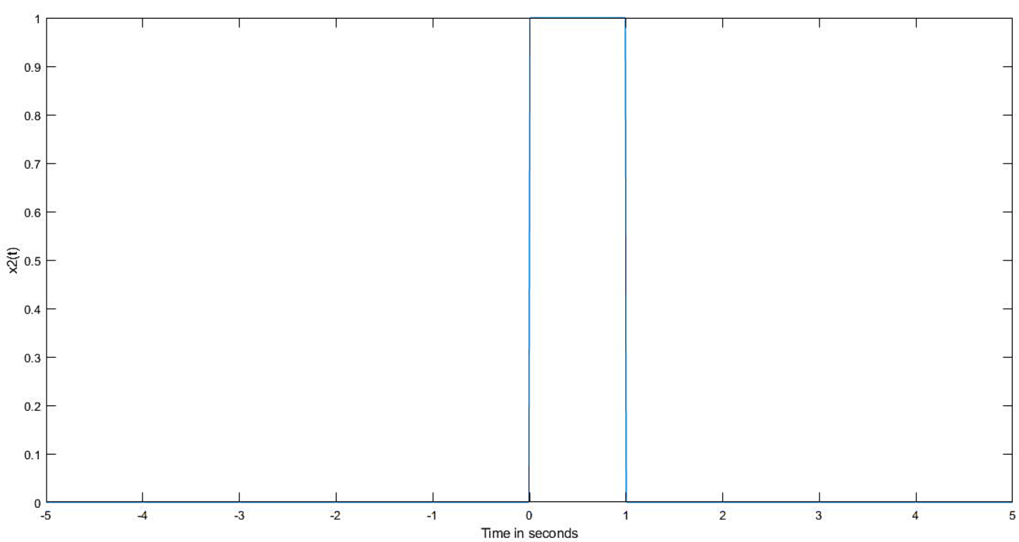

Figure-2: Graph of $x_2(t)$

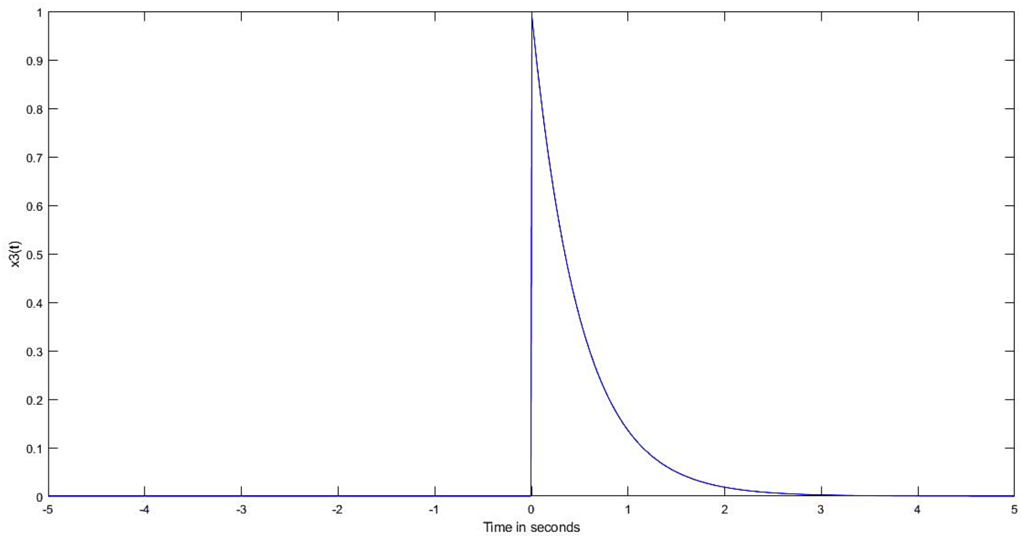

Figure-3: Graph of $x_3(t)$

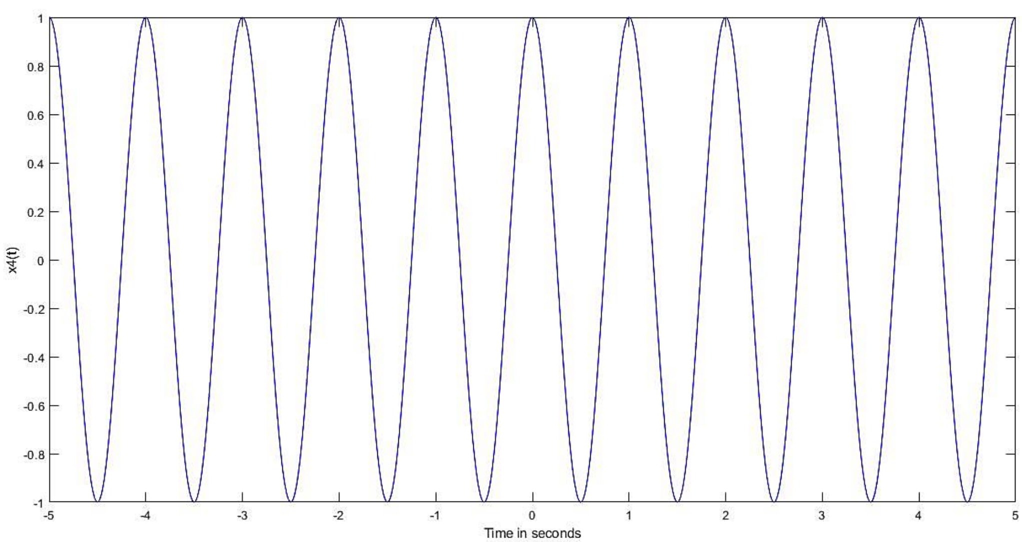

Figure-4: Graph of $x_4(t)$

- **Analysis**:

- The graph of $x_1(t), x_2(t), x_3(t)$, and $x_4(t)$ are all in accordance with their characteristics.
- The graph of $x_1(t)$ and $x_2(t)$ both visualize the unit-step function and a rectangular pulse formed by subtracting two unit-step functions.
- The graph of $x_3(t)$ describes a decaying exponential and this is because the unit step function forces the signal to start at $t = 0$.
- The graph of $x_4(t)$ describes a cosine wave with period $T = \frac{2\pi}{w} = \frac{2\pi}{2\pi} = 1$ sec.

(c) In this part, we will be repeating the previous two parts only that we will be taking N=50 (decreasing number of sampled points) and we will be plotting the graphs in their discrete time representation using the command **stem**.

- We will generate the new vector $n =$ **linspace(-5,5,50)** For which we obtain the following graphs:

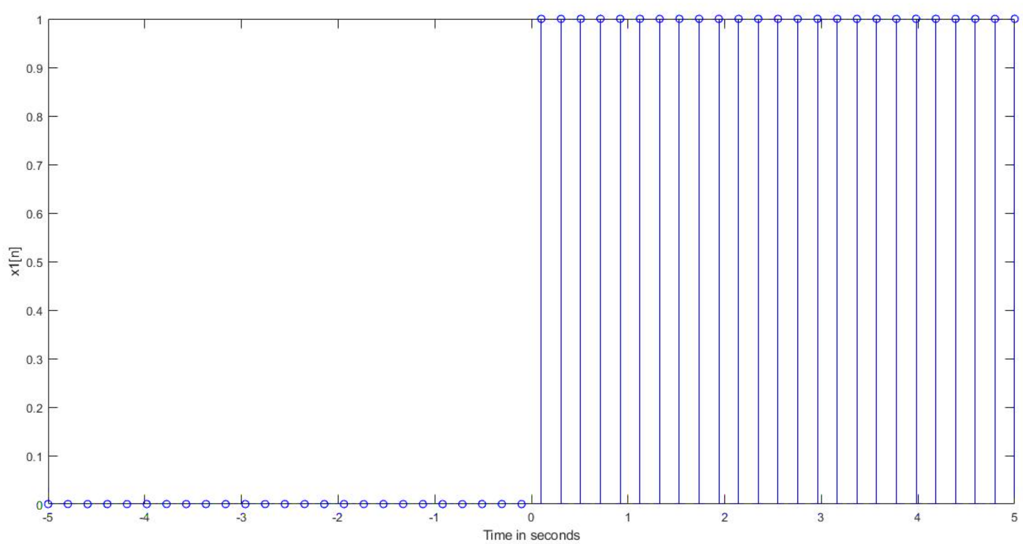

Figure-5: Graph of $x_1[n]$

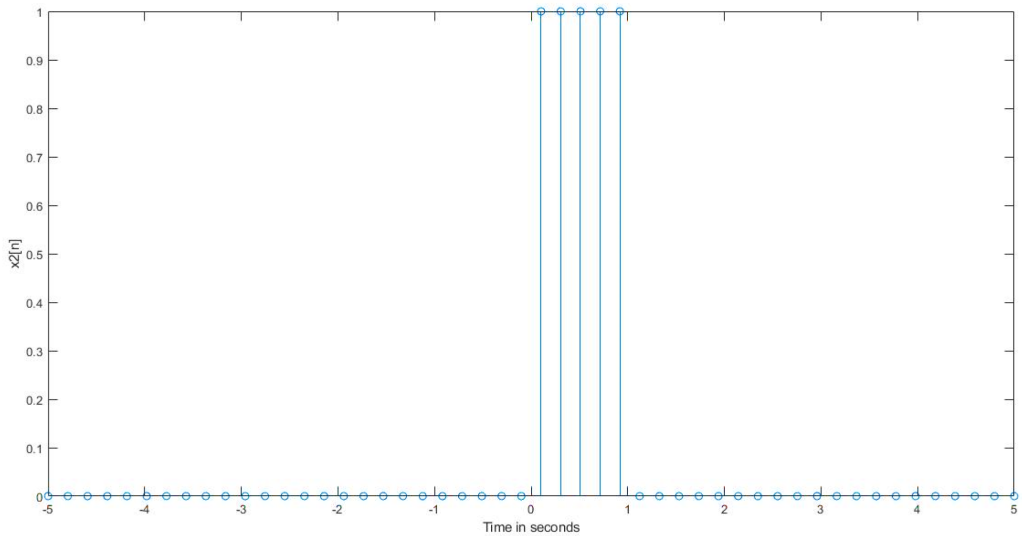

Figure-6: Graph of $x_2[n]$

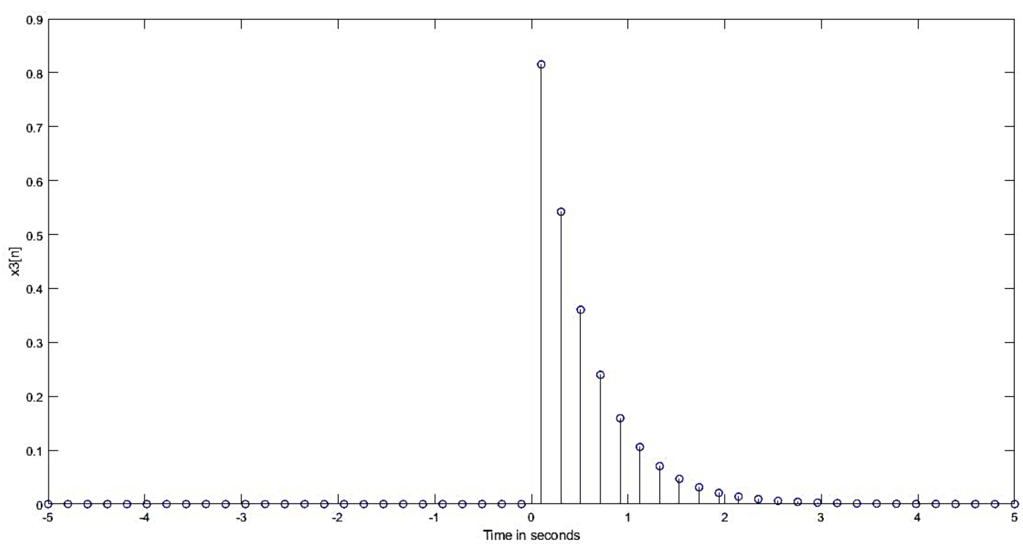

Figure-7: Graph of $x_3[n]$

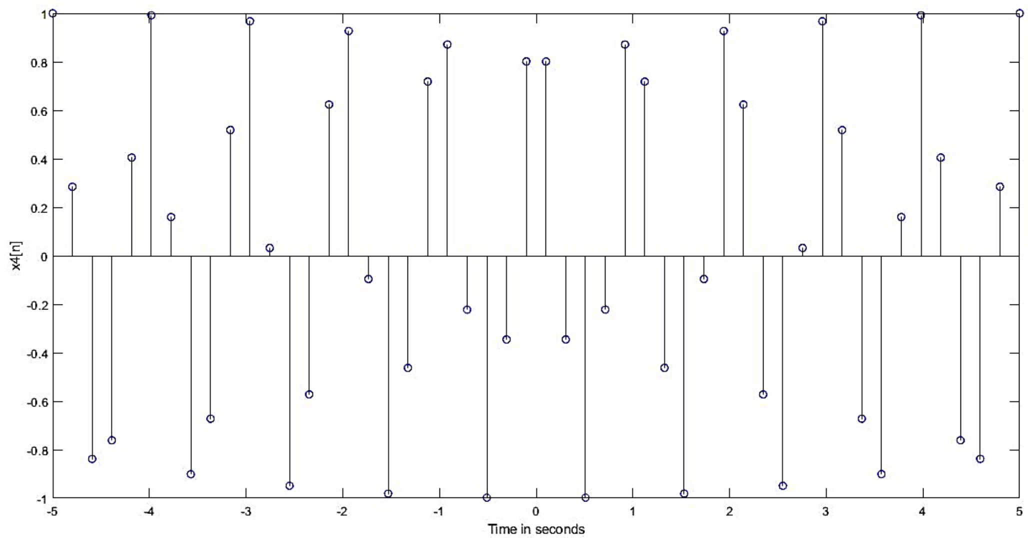

Figure-8: Graph of $x_3[n]$

- **Analysis**:

- Unlike part **(b)**, this time we observe that our graphs are plotted in discrete time values.
- All 50 sampled points are separated equally.
- This explains why $x_1[n]$ contains 25 equally distant impulses.
- This also justifies why $x_2[n]$ contains 4 equally distant impulses.

II) Spectral Representation

(a) In this part, we will focus on the frequency domain representation of our signals. We will start by generating a vector frequency f that starts from $-fs/2$ to $fs/2$ where $f_s = N/10$ is the sampling frequency. Since N=1024, then we have $f_s = 102.4$ Hz which our frequency vector starts from -51.2 and ends at 51.2 so we will define $f =$ **linspace (-51.2,51.2,1024)**.

(b) In this part, we will be evaluating the Fourier transform of $x_2(t)$, $x_3(t)$, and $x_4(t)$ after which we plot their corresponding magnitudes.

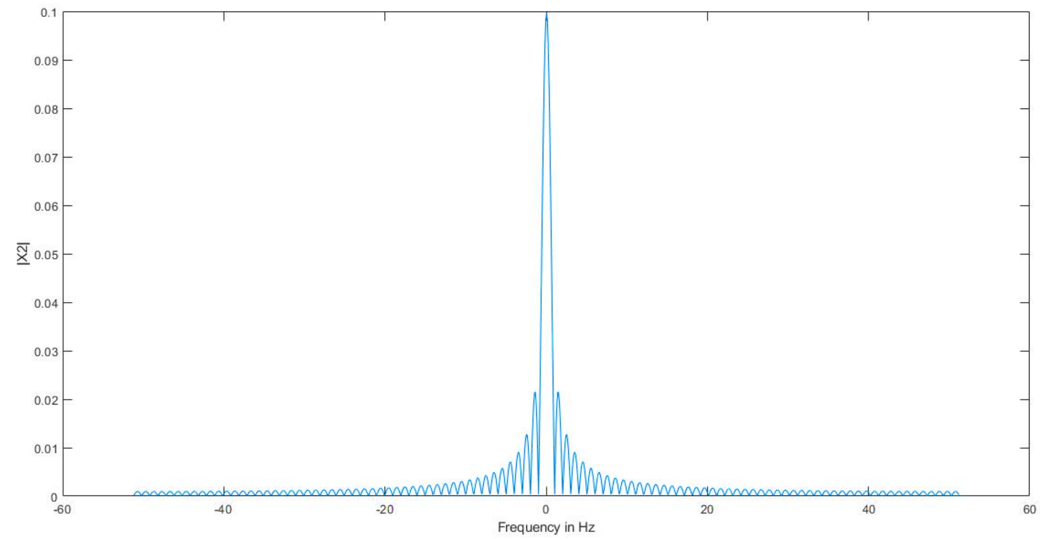

Figure-9: Graph of $|X_2|$

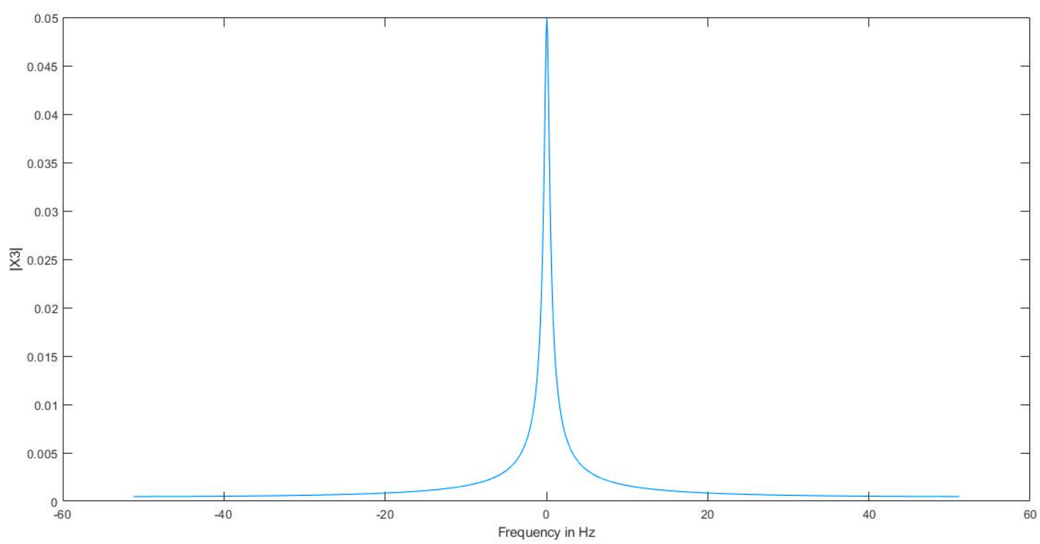

Figure-10: Graph of $|X_3|$

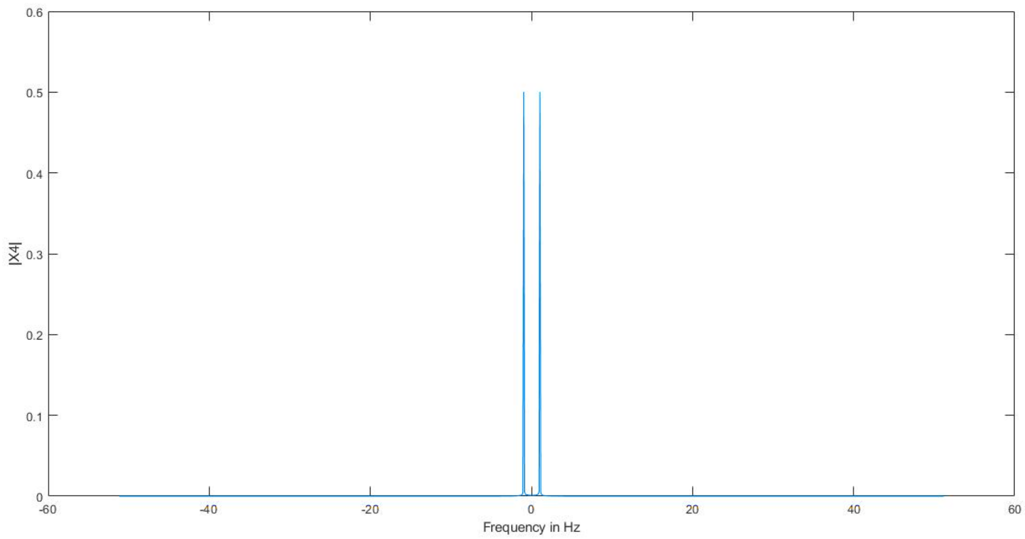

Figure-11: Graph of $|X_4|$

- Analysis:

- The three graphs correspond to the magnitude of their Fourier transform.
- As we know, the Fourier transform of a rectangular pulse is the *sinc* function which is what we observe in figure-10.
- The Fourier transform of $\cos(2\pi t)$ is two impulses located equidistantly from each other at $f = -1$ and at $f = 1$. This is because the Fourier transform of $\cos(2\pi a t)$ is $\frac{1}{2}\left(\delta(f-a) + \delta(f+a)\right)$ where here in this case $a = 1$. This what is correctly described in figure 11.

III) Ideal Low pass Filtering

In this part, we will be introducing a low pass filter to study its effect on our given signals. A low pass filter is a filter that blocks high frequencies from a signal and protects the low frequency components.

The blockage of high frequencies occurs starting at a threshold called the **cutoff frequency**. The Fourier transform of low pass filter is described the following way:

$$H = \begin{cases} 1, & |f| \leq f_c \\ 0, & |f| > f_c \end{cases}$$

(a) To create a vector that describes H the Fourier transform of the Lowpass filter h, we will consider the cutoff frequency $f_c = 10$ and consider out frequency vector f = **linspace(-51.2,51.2,1024)** and our main analysis will be completely centered around $x_1(t) = \cos(2\pi t)$. To construct H, we will thus use the command H=**heaviside(f+10)-heaviside(f-10)** which creates the rectangular pulse.

(c) For this part, we will be filtering the signal $x_4(t)$. We refer to **II** were we determined the Fourier transform of $x_4(t)$. However, we will not be taking its magnitude but the signal as a whole. We define the resulting output frequency Y such that $Y = H \cdot X$. We obtain the following graphical representations of $X, H,$ and Y as follow:

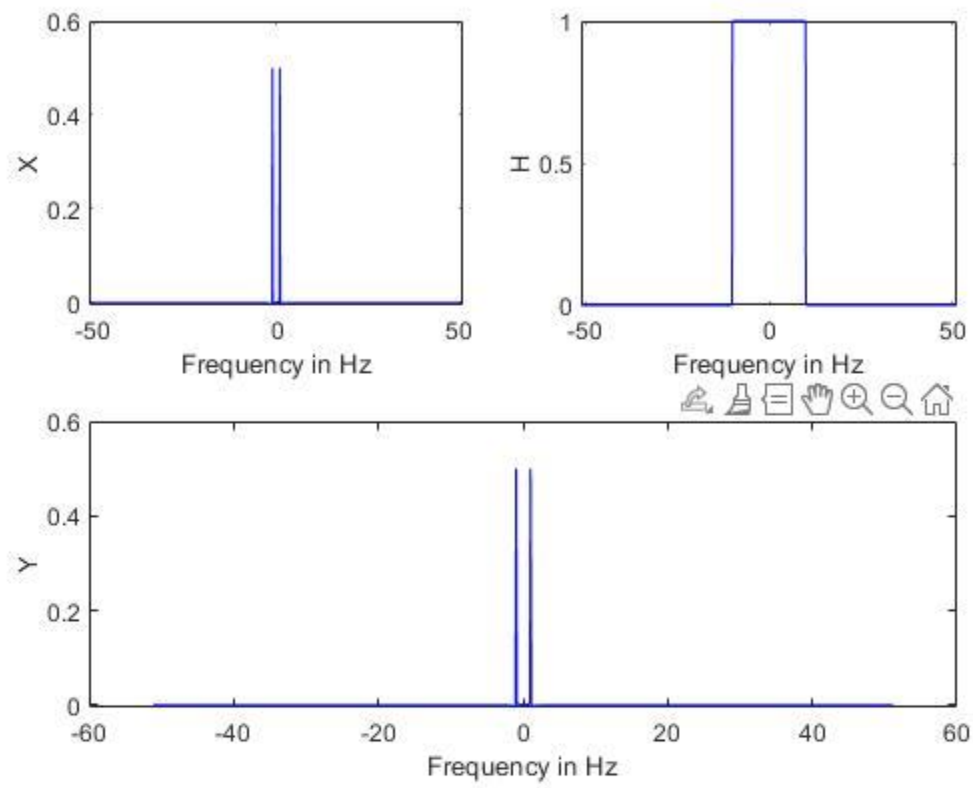

Figure-12: Graph of X,H, and Y for $f_c = 10$

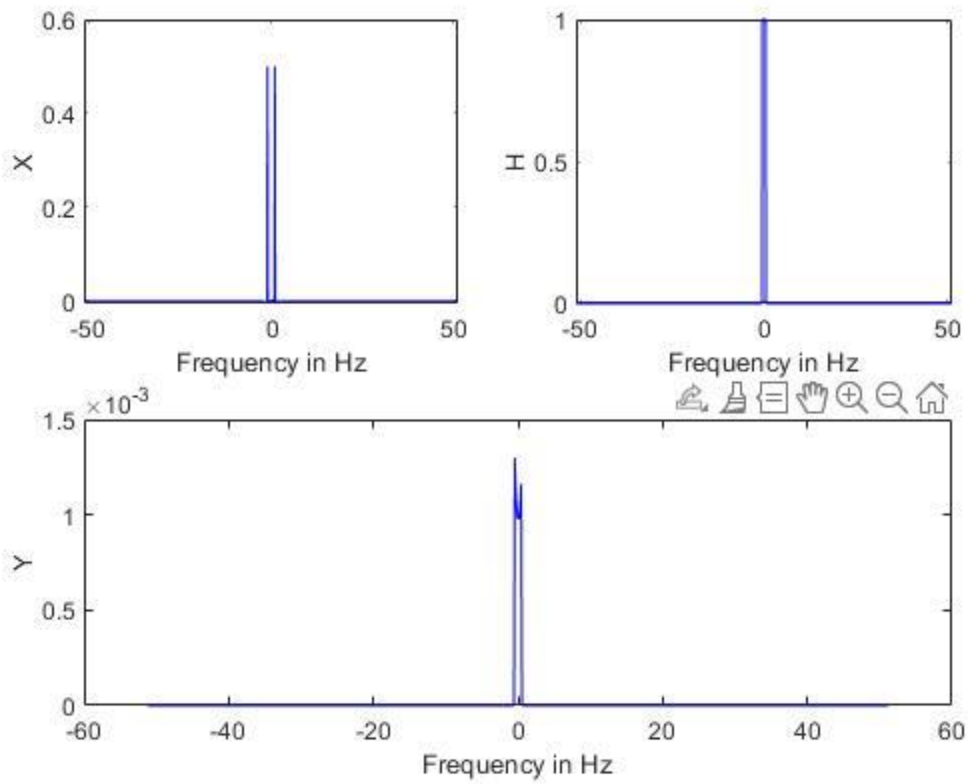

Figure-13: Graph of X,H, and Y for $f_c = 0.5$

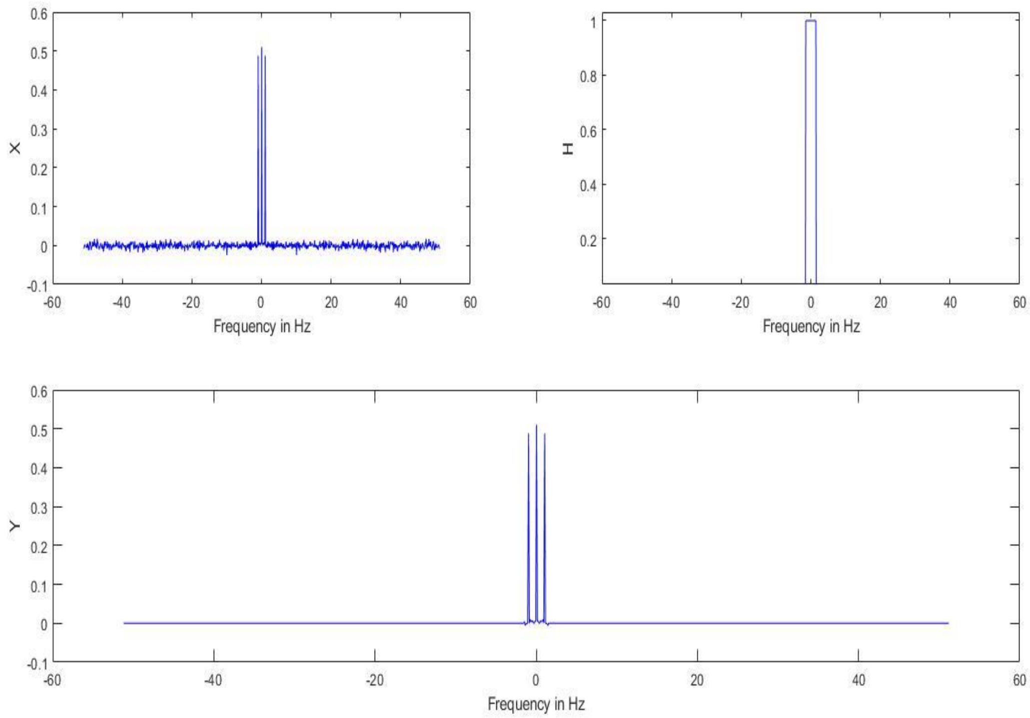

Figure-14: Graph of X,H, and Y for $f_c = 1.5$ with Noise.

We will also plot the inverse Fourier transform of the above signals for which we get the following figures:

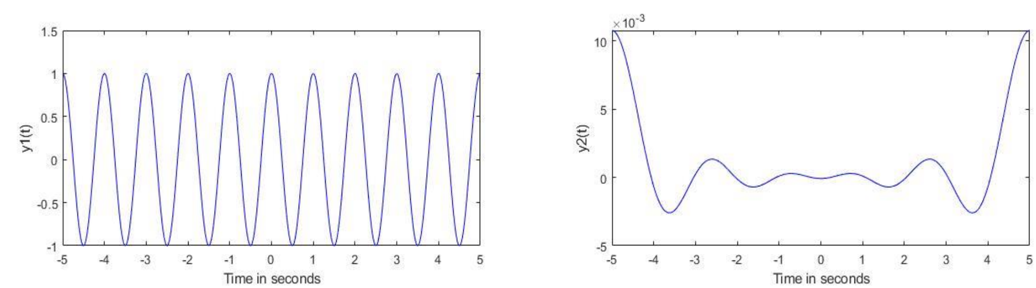

Figure-15: Graph of $y_1(t)$ the inverse Fourier of Y ($f_c = 10$) and $y_2(t)$ the inverse Fourier of Y ($f_c = 0.5$)

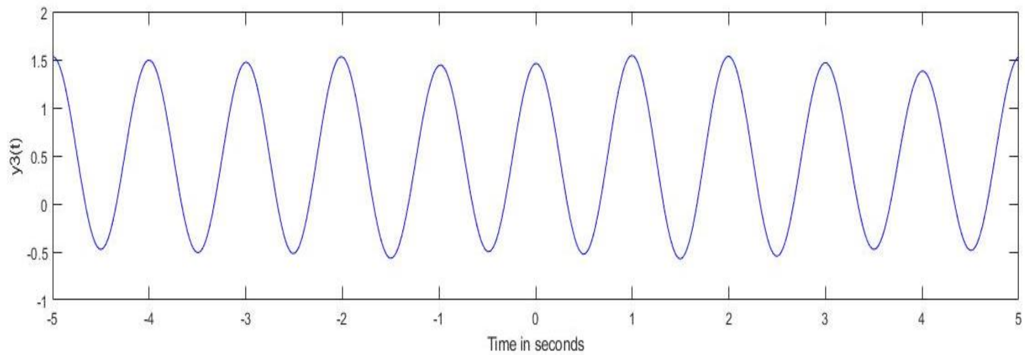

Figure-16: Graph of $y_3(t)$ the inverse Fourier of Y ($f_c = 1.5$) that contains Noise.

Analysis:

- For $f_c = 10$, the filter did not have any effect since the two impulses lie at $f = -1$ and $f = 1$ which is covered by the low pass filter.
- For $f_c = 0.5$, the filter removed all the frequencies outside the range of cutoff frequency and preserved a very low frequency component with maximum amplitude 1.3×10^{-3} Hz as seen in **figure-13**
- For $f_c = 1.5$, with the noise added to the signal, the filter was able to remove the noise from the frequency spectrum not covered by the lowpass filter, but this process did not eliminate the impulse at the middle of the two impulses in **figure-14**.
- After finding the inverse Fourier transform, we can conclude that the signal is almost negligible $f_c = 0.5$ because most information of the signal was lost as seen in **figure-15(2)**.

IV) Filtering of Audio Signals

In this part, we will be loading the sample audio1 into MATLAB and perform the necessary analysis.

(b) Using the command **audioread** we retrieved the acquisition frequency which is determined to be $f_s = 16{,}000$ Hz. Moreover, the length of the audio is computed and is found to be 144000.

(c) By computing the Fourier transform of the audio signal, we plot its magnitude below:

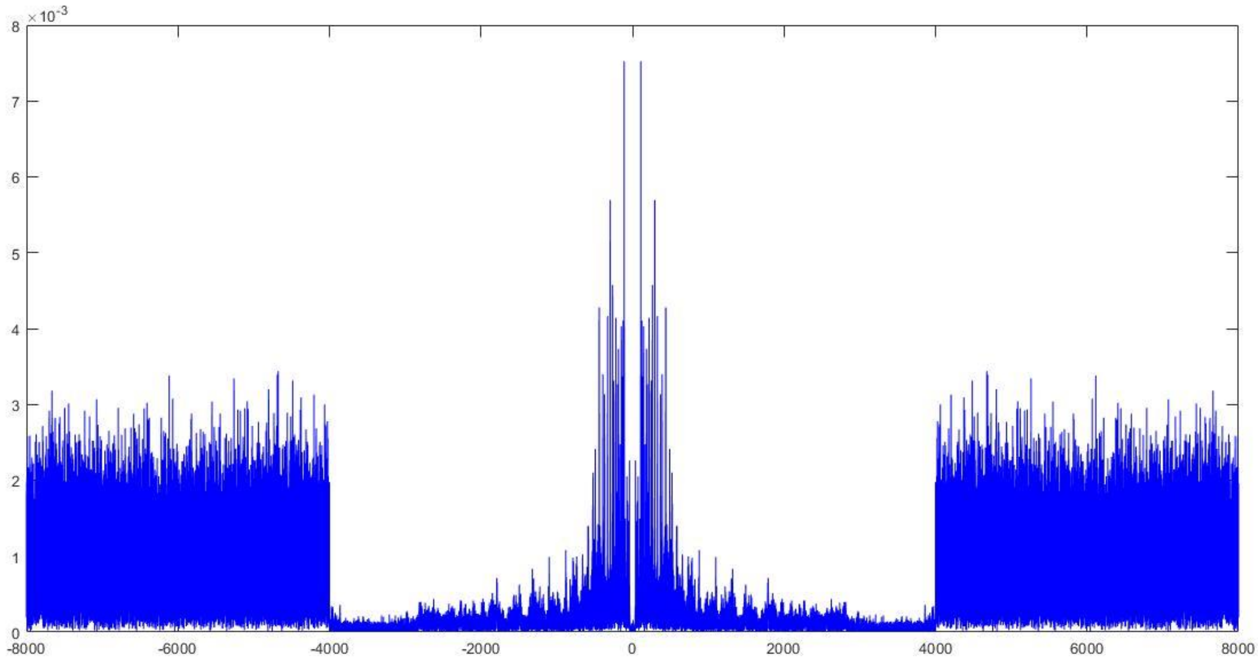

Figure-17: Magnitude of FFT of Audio1.

It's clear from **figure-17** that the frequency that separates the high frequency component from the original frequency component is $f = \pm 4000$ Hz.

(d) To filter this audio, we will construct **H** with a cutoff frequency based on the observation and analysis performed in **(c)**.

$$H = \begin{cases} 1, & |f_{audio}| \leq 4000 \\ 0, & \text{else} \end{cases}$$

The output audio frequency will be denoted by Y and its magnitude is plotted below:

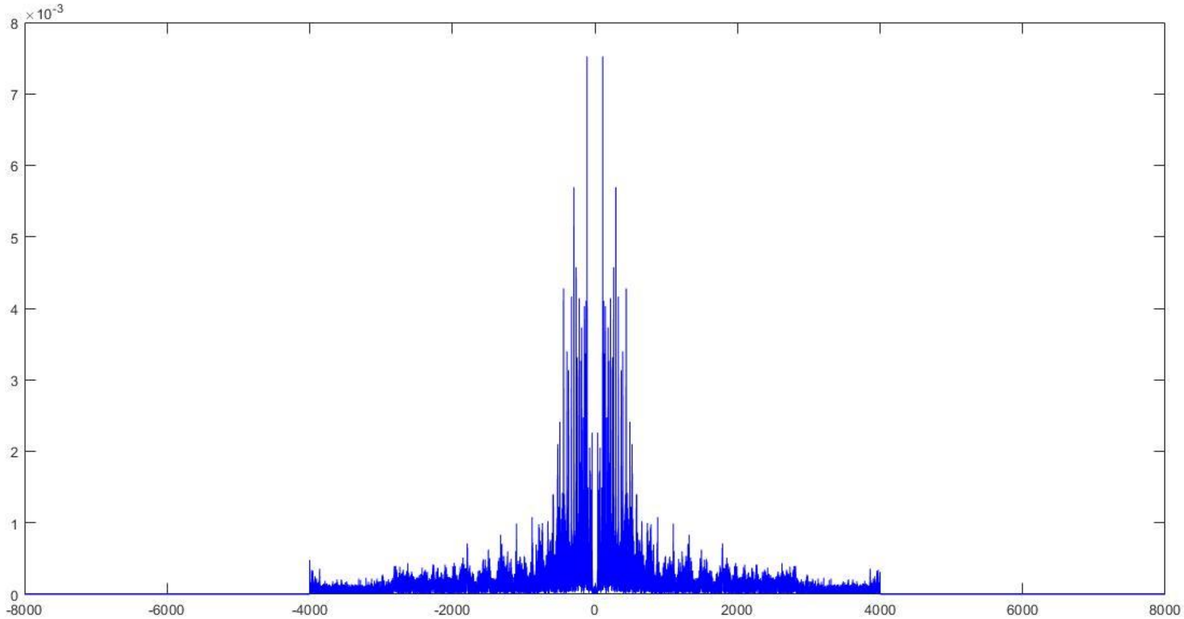

Figure-18: Magnitude of FFT of the filtered Audio1

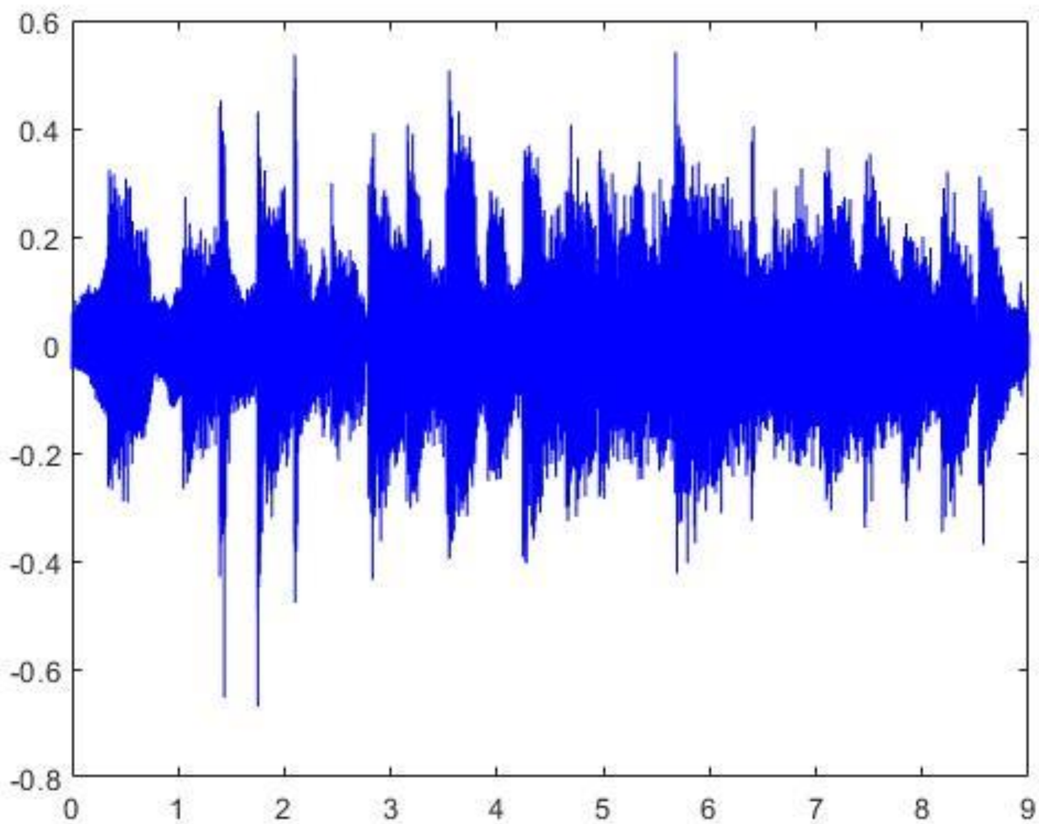

Figure-19: Magnitude of FFT of the filtered Audio1 in time-domain

V) Audio Applications

1) For this part, we are mainly interested in inserting an echo to audio signal2. This can be done as follow:

- Step 1: Read the audio2 file and extract fs and y.

- Step 2: assign the input signal to be y.

- Step 3: Define $h[n] = \delta[n] + 0.5 * \delta[n - 16000]$

- Step 4: Compute the convolution using command **filter(h,1,x)**

After writing the saved audio file, and listening to it, we can confirm the presence of echo in the signal.

2) In the last part of the project, we will be working on separating the bass drum audio from the entire audio signal.

- After plotting the graph of the Fourier transform of Audio3 it was determined that at $f_c = 900$, the Low pass filter would act in removing other frequency components and keeping the bass.

- Furthermore, we have subtracted the output signal from the original audio and obtained the audio without bass.