ADVANCED SIGNAL ANALYSIS AND PROCESSING LAB

Lab sheet. No: 04

NAME: Pedasingu Rajesh **Roll. No:** EE21M019

Q1. Verify the All properties of Fourier transform using function of MATLAB fft. You may use following sequences–

x1[n] = [0.59 0.95 0.95 0.59 0.00 0.59 0.95 0.95 0.59 0.00]

x2[n] = [0.16 0.97 0.96 0.49 0.80 0.14 0.42 0.92 0.79 0.96]

AIM: To verify all the fourier transform properties using given x1[n] and x2[n]

Short Theory:

Linearity:

A system S is

1.**homogeneity**: if, for any input x(n) and any number a,

$$S\{ax(n)\} = aS\{x(n)\}$$

2.**additive**: if for any two inputs x1(n) and x2(n),

$$S\{x1(n) + x1(n)\} = S\{x1(n)\} + S\{x2(n)\}$$

A system that is both additive and homogeneous is called linear. In other words, S is linear if, for any two inputs x1(n) and x2(n) and any two numbers a1 and a2,

$$S\{a1x1(n) + \mathrm{a2}x1(n)\} = \mathrm{a1}S\{x1(n)\} + \mathrm{a2}S\{x2(n)\}$$

convolution

x[n] and h[n] are two finite sequences of length N with DFTs denoted by X[k] and H[k], respectively. Let us form the product W[k] = X[k]H[k], and determine the sequence w[n] of length N for which the DFT is W[k].

$$x_1(n) \textcircled{N} x_2(n) \xrightarrow{DFT} X_1(k).X_2(k)$$

(Note that this is NOT the same as the convolution property.)

Time shifting property

Shifting the sequence in time domain by ‘l’ samples is equivalent to multiplying the sequence in frequency domain by the twiddle factor.

$$x((n-l))_N = x(n-l) \xleftrightarrow{DFT} X(k)e^{-2\pi jlk/N}$$

time reversal property

$$x(N-n) \xleftrightarrow{DFT} X(N-k)$$

Parseval’s theorem

$$\sum_{n=0}^{N-1} x(n).y^*(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k).Y^*(k)$$

Key Commands:

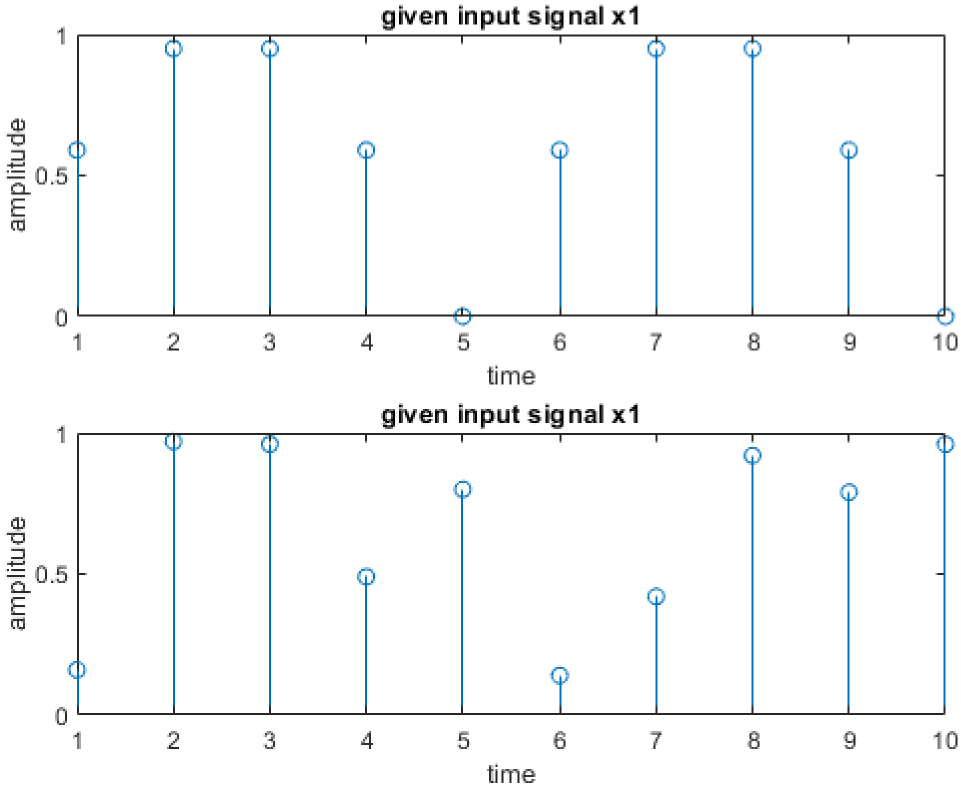
cconv % cconv(a,b) convolves vectors a and b.

fft(x)   % fft(X) computes the discrete Fourier transform (DFT) of X using a fast Fourier transform (FFT) algorithm.

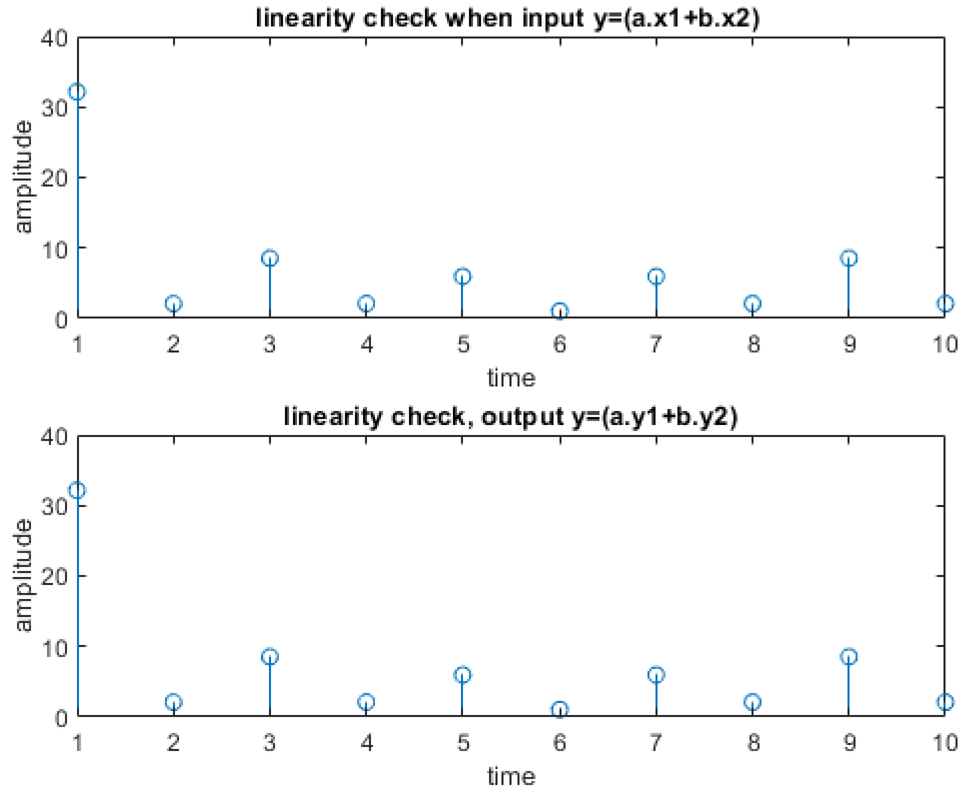
Abs() % abs(x) gives the magnitude of x.

Circshift() % circshift(A,K) circularly shifts the elements in array A by K positions.

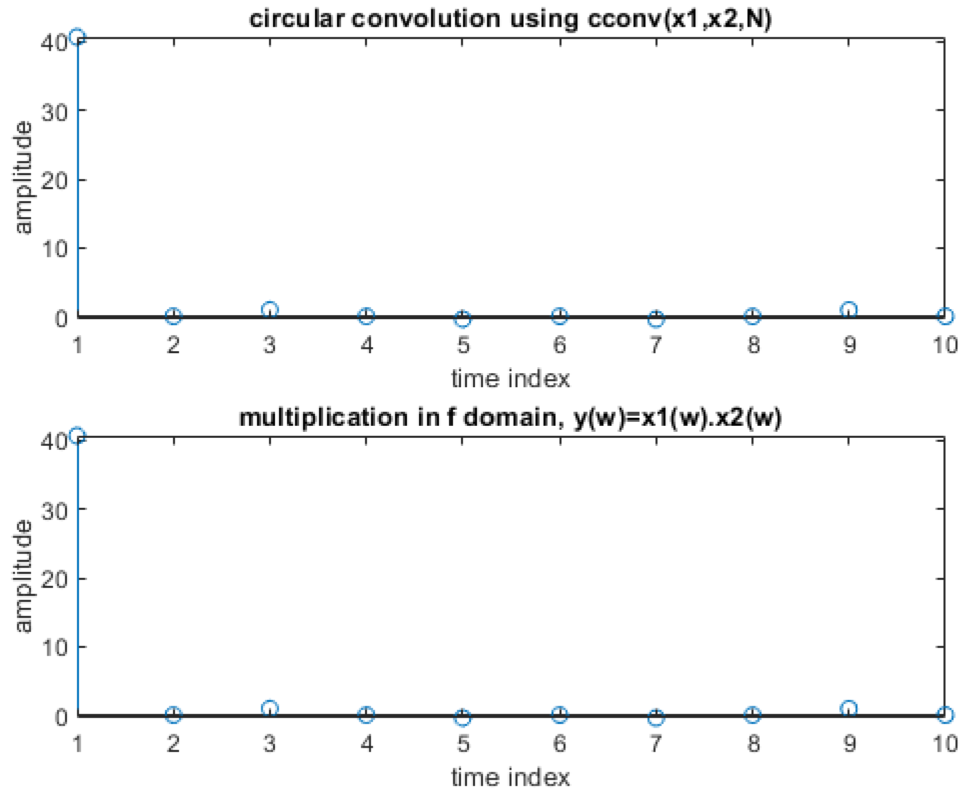
Plots:

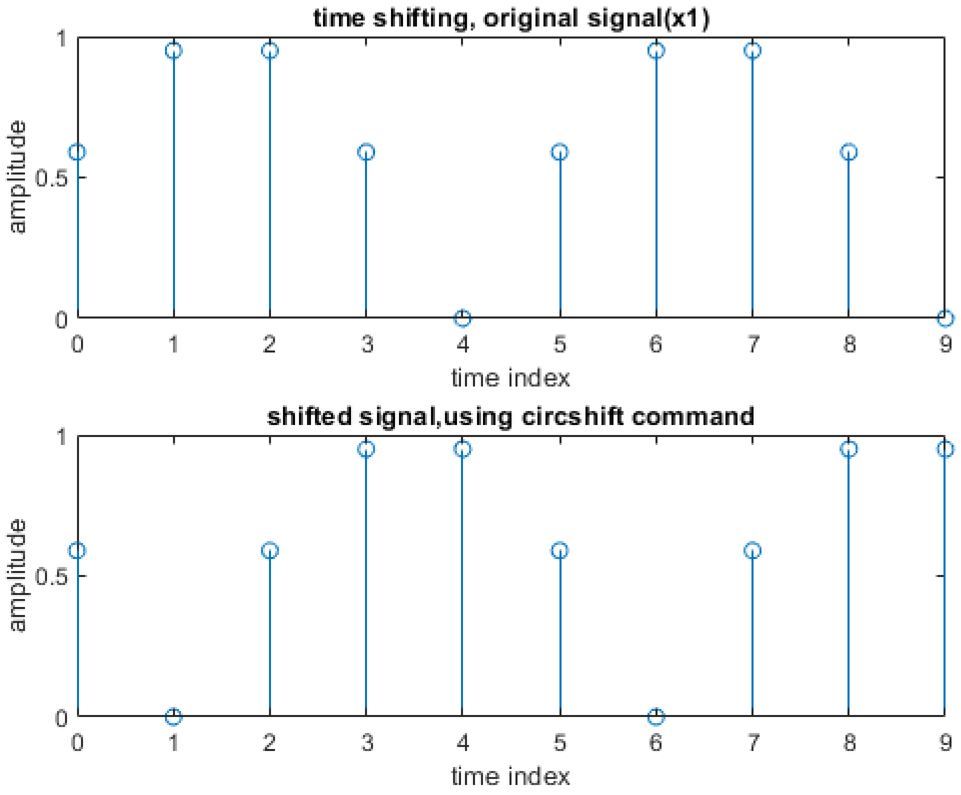


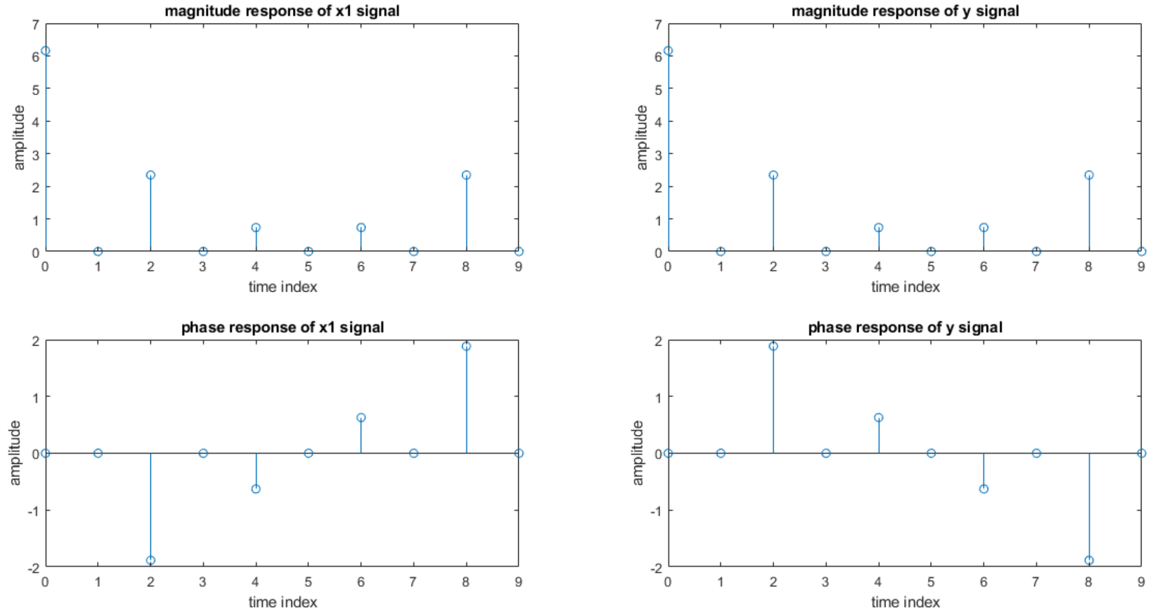
given input signal x1
amplitude
time
given input signal x1
amplitude
time

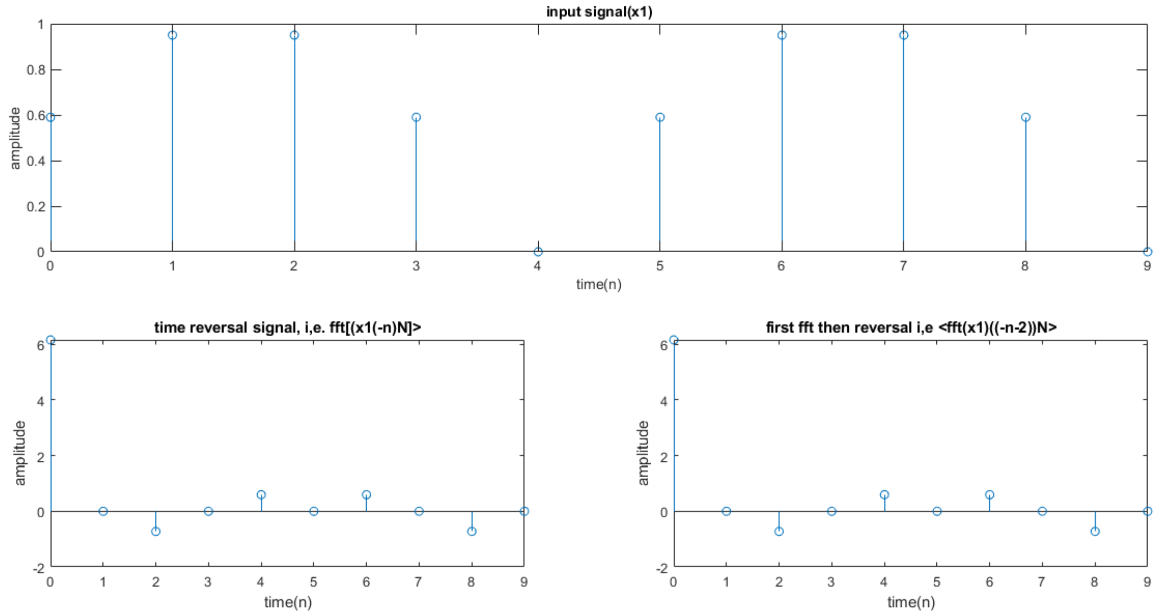


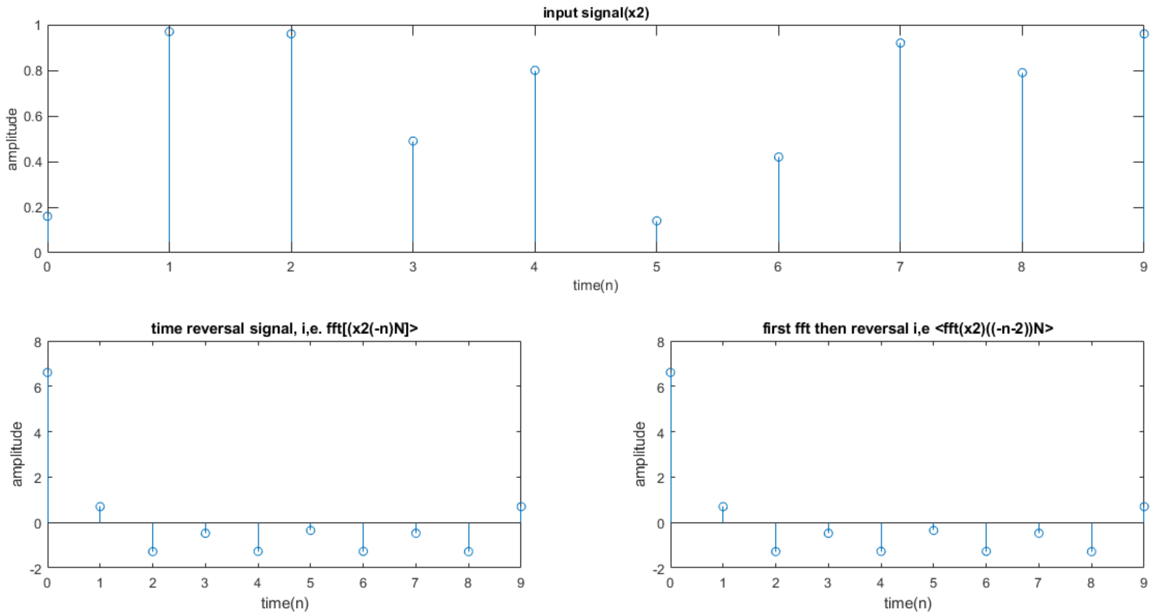
linearity check when input y=(a.x1+b.x2)
amplitude
time
linearity check, output y=(a.y1+b.y2)
amplitude
time


circular convolution using cconv(x1,x2,N)
amplitude
time index
multiplication in f domain, y(w)=x1(w).x2(w)
amplitude
time index


time shifting, original signal(x1)
amplitude
time index
shifted signal,using circshift command
amplitude
time index


magnitude response of x1 signal
magnitude response of y signal
phase response of x1 signal
phase response of y signal
amplitude
time index


input signal(x1)
amplitude
time(n)
time reversal signal, i,e. fft[(x1(-n)N]>
first fft then reversal i,e <fft(x1)((-n-2))N>



Command Window

```
parsevel theorem in time domain
    4.0387

parsevel theorem in frequency domain
    4.0387

fx >>
```

Inferences/comments:

1) Circular convolution and linear convolution both are different. In case of circular convolution the length of the convolution sequence is maximum of both the sequences.

2) We can observe from the Parseval theorem property, energy of a signal in the time domain equals the energy of the transformed signal in the frequency domain.

Q2. Generate the rectangular pulse signal of appropriate size. Use MATLAB function fft to find the Fourier transform. Write, in brief, about your observations.

AIM: To generate and plot the rectangular pulse signal and find the Fourier transform using fft .

Short Theory:

The rectangular pulse is defined as Π(t)={ A ,-T2≤t≤T2

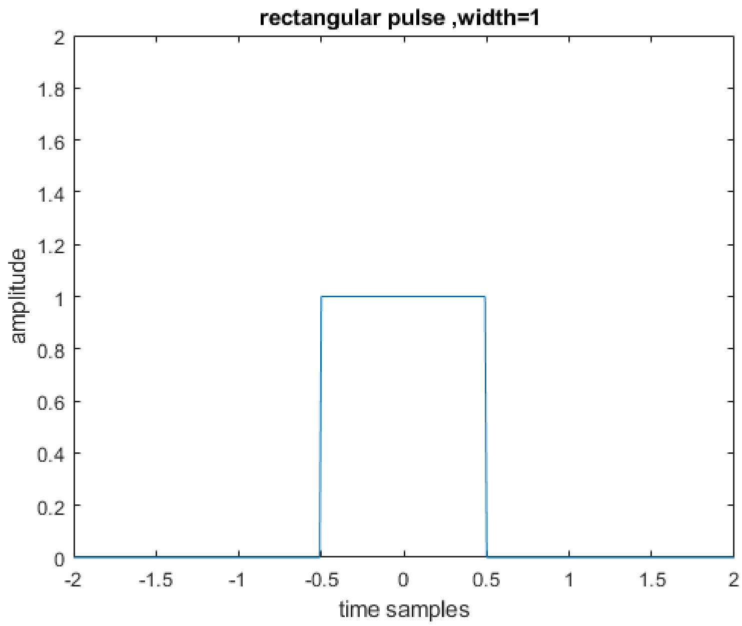
= 0 ,otherwise.

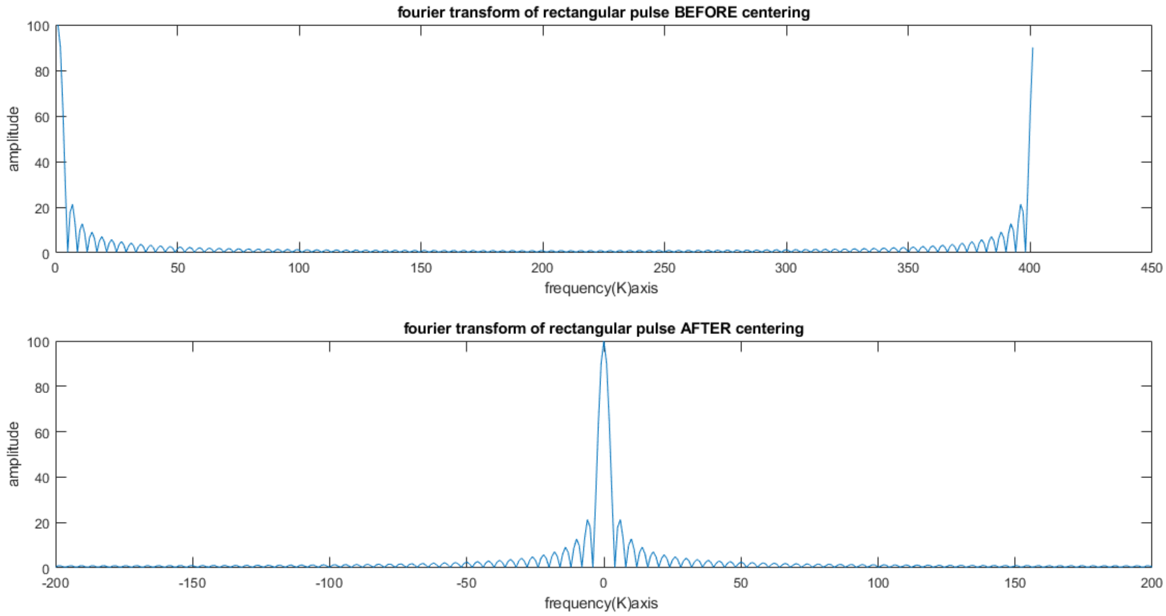
where A is the amplitude of the pulse and L is an integer.

$$\mathrm{DFT}_{\text{rect.function}} = \frac{\sin(x)}{\sin(x/N)}, \text{ or } \frac{\sin(x)}{x}, \text{ or } \frac{\sin(Nx/2)}{\sin(x/2)}.$$

Key Commands:

Plots:





Inferences/comments:

1) Fourier transform of rectangular pulse gives the sinc function in frequency domain.
2) Fft command in matlab gives the output without centering the frequencies to zero. But fftshift command rearranges a Fourier transform X by shifting the zero-frequency component to the center of the array.

Q3. Generate a sinusoidal signal of length 0.5 seconds with frequency 100Hz, sampled at 8000Hz. Plot the magnitude and phase spectrum of the Fourier transform. Write, in brief, about your observations.

AIM:
To generate and plot the sinusoidal signal of length 0.5 sec with f=100Hz, Fs=8000Hz. And plot the phase spectrum of the Fourier transform.

Short Theory:

Sinusoidal signal with a given frequency can generate by using y=sin(2πft). fft command gives the Fourier transform of the signal y . Fourier transform contains the complex numbers so we plot the Fourier transform as magnitude and phase response.

Key Commands:

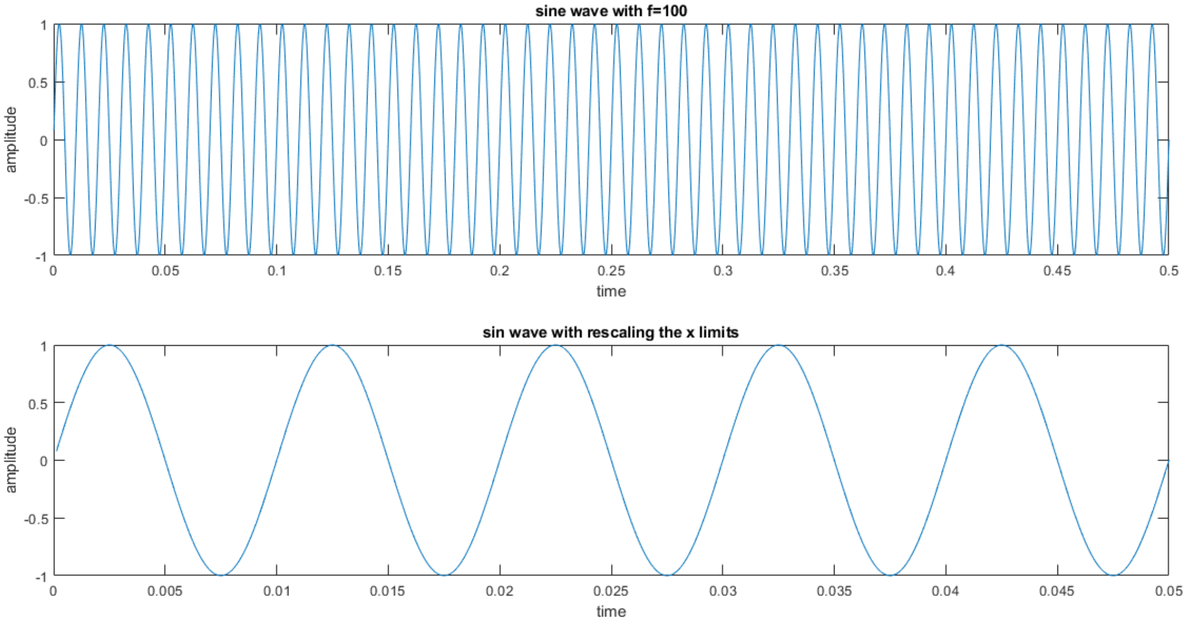
fft % fft(X) computes the discrete Fourier transform (DFT) of X using a fast Fourier transform (FFT) algorithm.

abs % Abs() % abs(x) gives the magnitude of x.

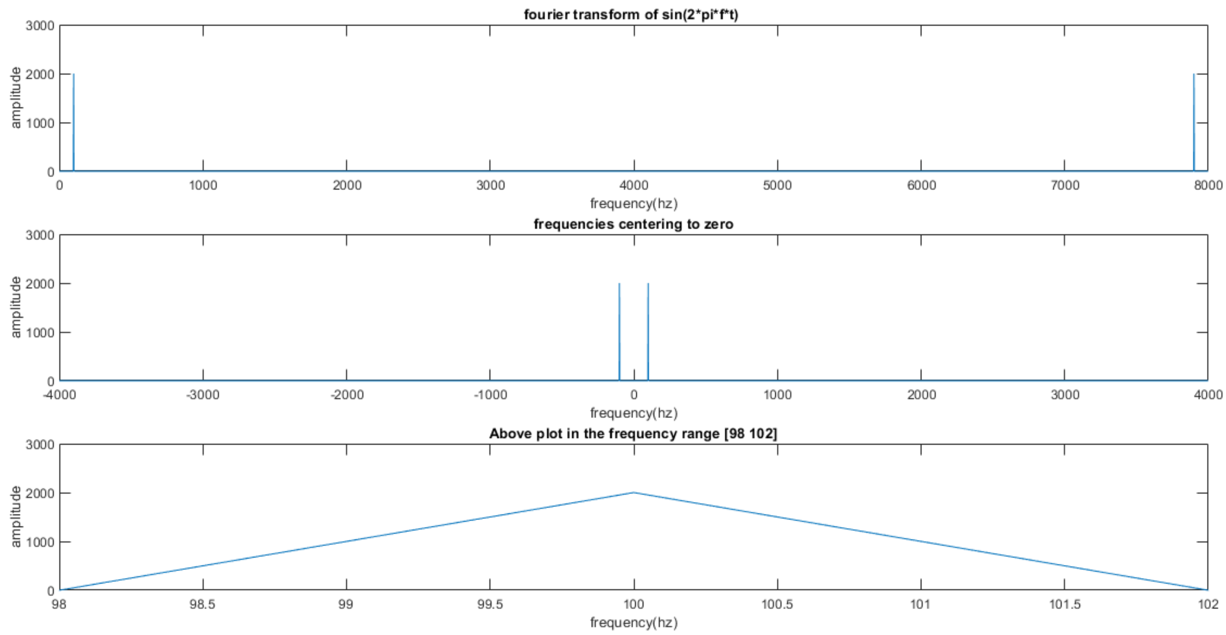
Angle % angle(z) returns the phase angle in the interval [-π,π] for each element of a complex array z.

Fftshift % fftshift(X) rearranges a Fourier transform X by shifting the zero-frequency component to the center of the array.

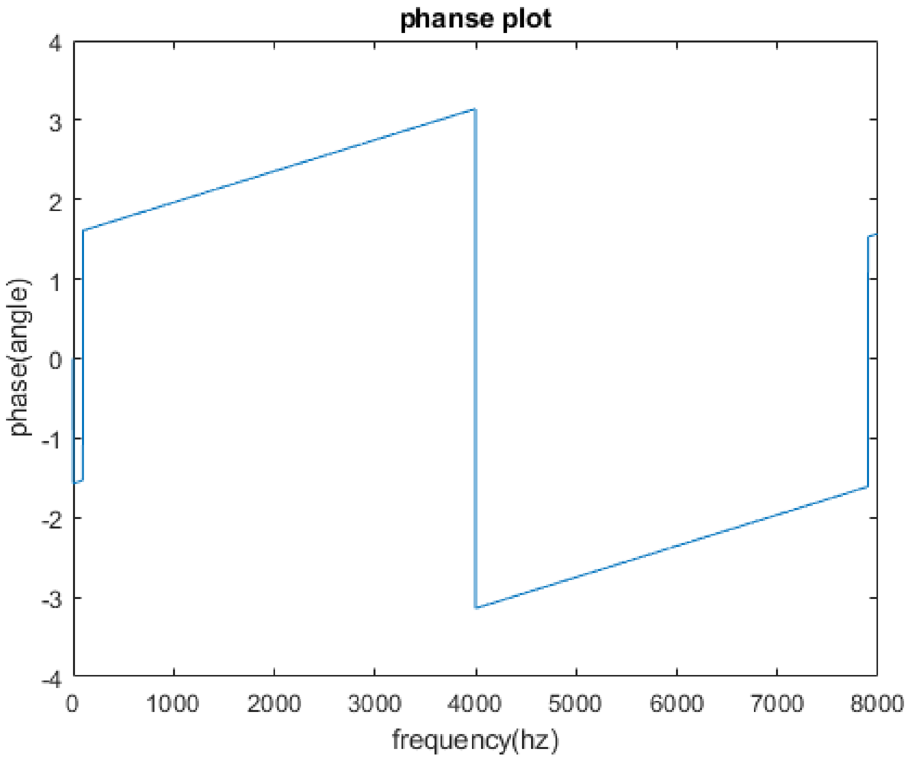
Plots:



Magnitude plot



Phase plot



Inferences/comments:

1) fourier transform of a sinusoidal function gives the 2 impulses in the frequency domain.
2) Fourier transform gives the complex numbers so we plot the Fourier transform of a signal as magnitude and phase response.
3) Fft command in matlab gives the output without centering the frequencies to zero. But fftshift command rearranges a Fourier transform X by shifting the zero-frequency component to the center of the array.

Q4. Generate a Gaussian function with zero mean and variance 1. Plot the magnitude spectrum of the Fourier transform of the signal.

AIM: To generate the gaussian function with mean=0 and variance =1 . And plot the magnitude spectrum of the Fourier transform of the signal.

Short Theory:

Gaussian probability distribution is perhaps the most used distribution in all of science. It can also called “bell shaped curve” or normal distribution. By using central limit theorem gaussian random variables are generated.

The probability density of the normal distribution can be computed as

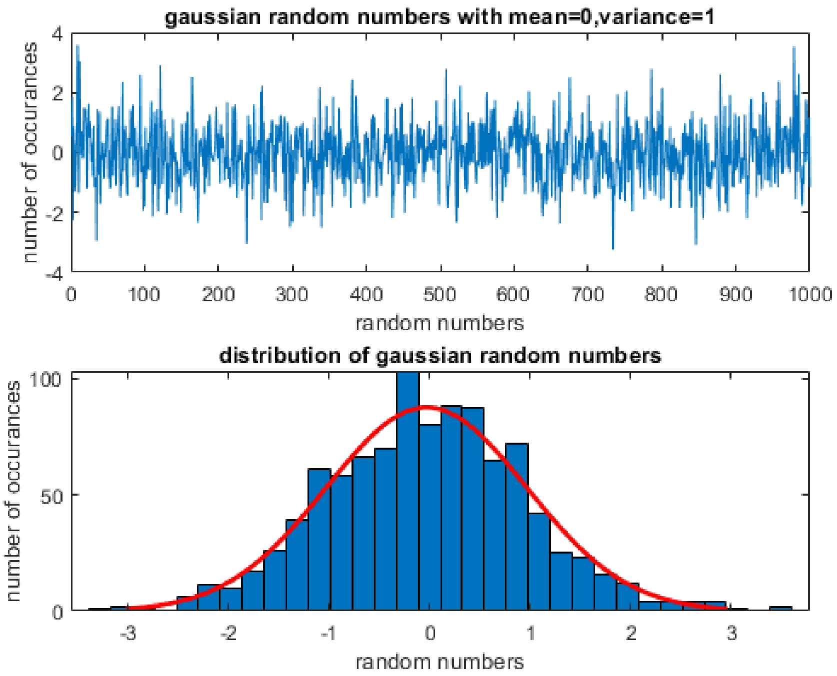
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

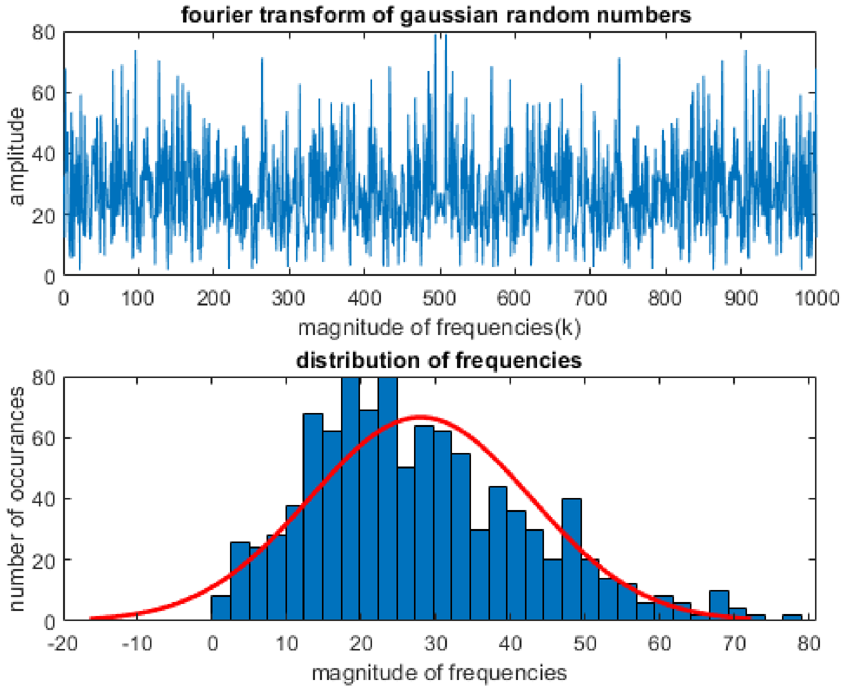
By using fft command we can get the Fourier transform of given gaussian random numbers.

Key Commands:

histogram % creates bar plot for numeric data that group the data into bins.
randn % generate random number that follows gaussian distribution.
fft % fft(X) computes the discrete Fourier transform (DFT) of X using a fast Fourier transform (FFT) algorithm.
Abs % abs(x) gives the magnitude of x.
Angle % angle(z) returns the phase angle in the interval [-π,π] for each element of a complex array z.

Plots:





Inferences/comments:

1) If the N value(number of samples) increases the distribution become more likely standard gaussian distribution.
2) fft command gives the complex numbers so, we plot the magnitude and phase response, even though we generated random numbers with mean zero and variance 1, the magnitude response of the Fourier transform mean changes to positive value, that is depend on the number of samples we generate.
3) The Fourier transform of the gaussian random numbers also a gaussian distribution.

Q5. Generate a dual tone signal by adding two sinusoidal signals of length 0.5 seconds and different frequencies. Normalize the frequency axes to find out the frequencies present in the given signal.

AIM: To generate and plot a dual tone signal by adding 2 sinusoidal signals of length 0.5 sec and find the frequencies present in the given signal.

Short Theory:

Dual tone signal can be created by adding to signals with different frequencies. If f1, f2 are the two different frequencies then dual tone signal can be generated by using the following equation

$$Y = A1 \sin(2\pi f1\ t) + A2 \sin(2\pi f2\ t).$$

The fft command gives the fourier transform of the dual tone signal in K axis not in frequency(hz). We need to normalise the frequency axis from K axis to Hz axis. So for that we need to multiply the (fs/N) to the K axis. Where fs=sampling frequency and N is the length of the signal Y.

Key Commands:

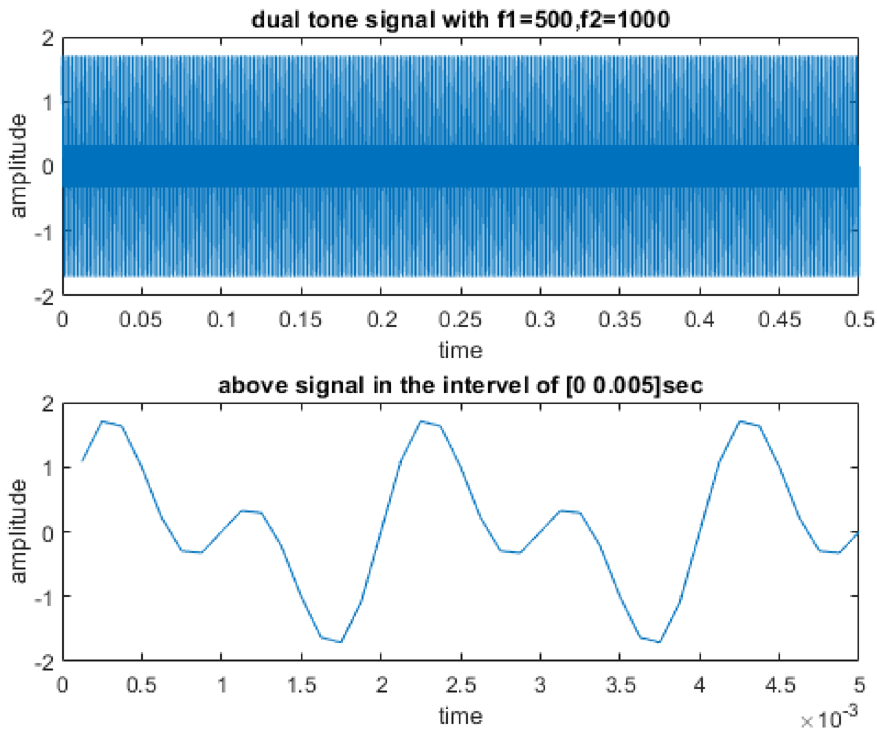
sin % it gives the values of sin(x)

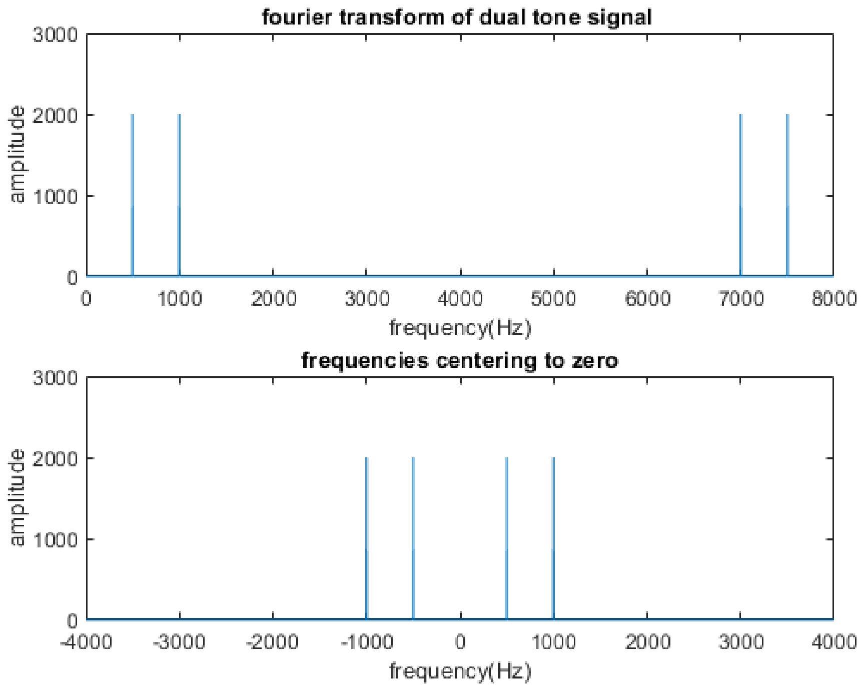
fft % fft(X) computes the discrete Fourier transform (DFT) of X using a fast Fourier transform (FFT) algorithm.

Abs % abs(x) gives the magnitude of x.

fftshift % fftshift(X) rearranges a Fourier transform X by shifting the zero-frequency component to the center of the array.

Plots:





Inferences/comments:

1) The fourier transform of a dual tone signal contains 2 impulses.
2) fftshift rearranges a Fourier transform of signal by shifting the zero-frequency component to the center of the array.
3) normalising the frequency axis by multiply the (fs/N) to the K axis. Where fs=sampling frequency and N is the length of the signal , then we can find the frequency components in a dual tone signal from the magnitude response(peak location of the impulses gives the frequencies)

Experimental Exercises:

Q1. Download the audio file from the URL provided during lab. Find the frequency content of the given signal.

AIM: To find the frequency contents of the given signals.

Short Theory:

Procedure to find the frequency components in a given audio signals

- Kept all the audio files and correspond .m file in one folder.
- Extract the signal values and sampling frequency by using audioread command
- Apply the Fourier transform by using fft command
- Normalise the frequency axis by multiply the (fs/N) to the K axis. Where fs=sampling frequency and N is the length of the signal.
- Then read the frequency values from magnitude response.

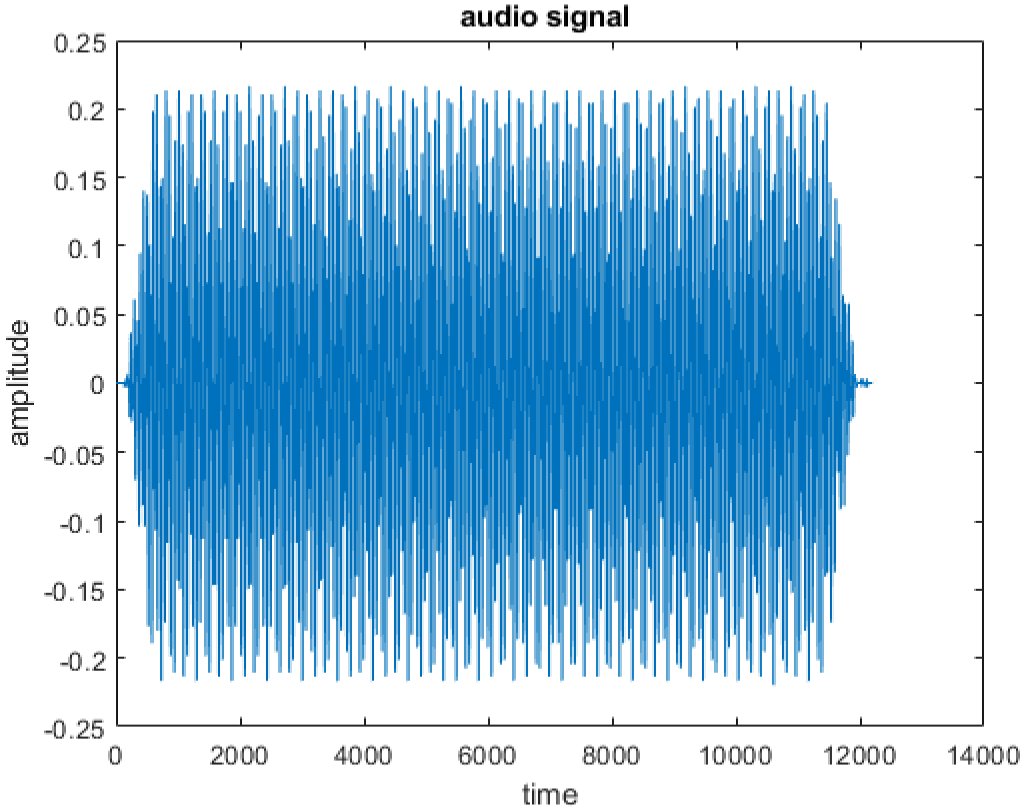
Key Commands:

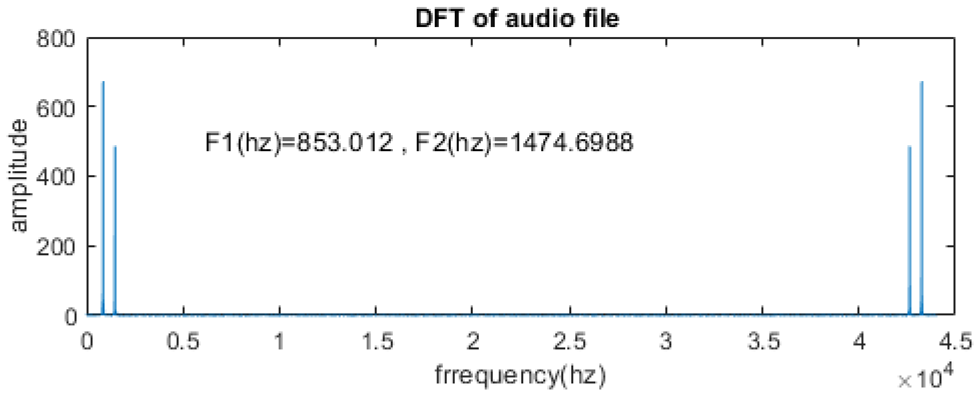
fftshift % fftshift(X) rearranges a Fourier transform X by shifting the zero-frequency component to the center of the array

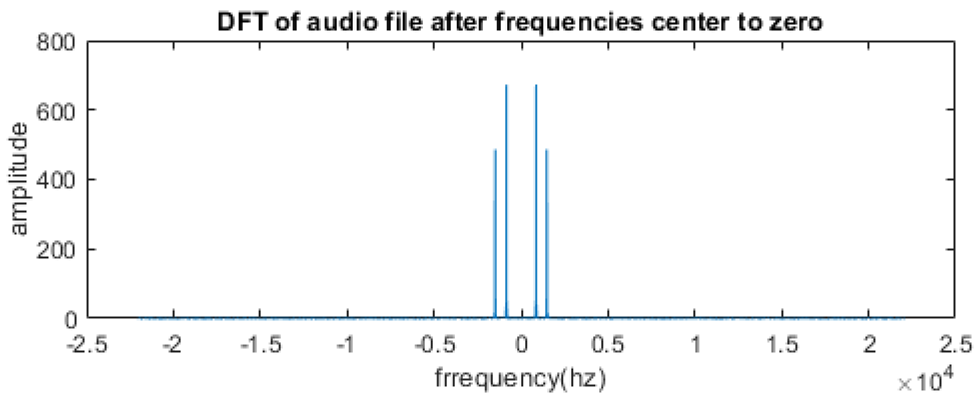
Abs % abs(x) gives the magnitude of x.
Audioread() % [y,Fs] = audioread(filename) reads data from the file named filename, and returns sampled data, y, and a sample rate for that data, Fs.

Plots:

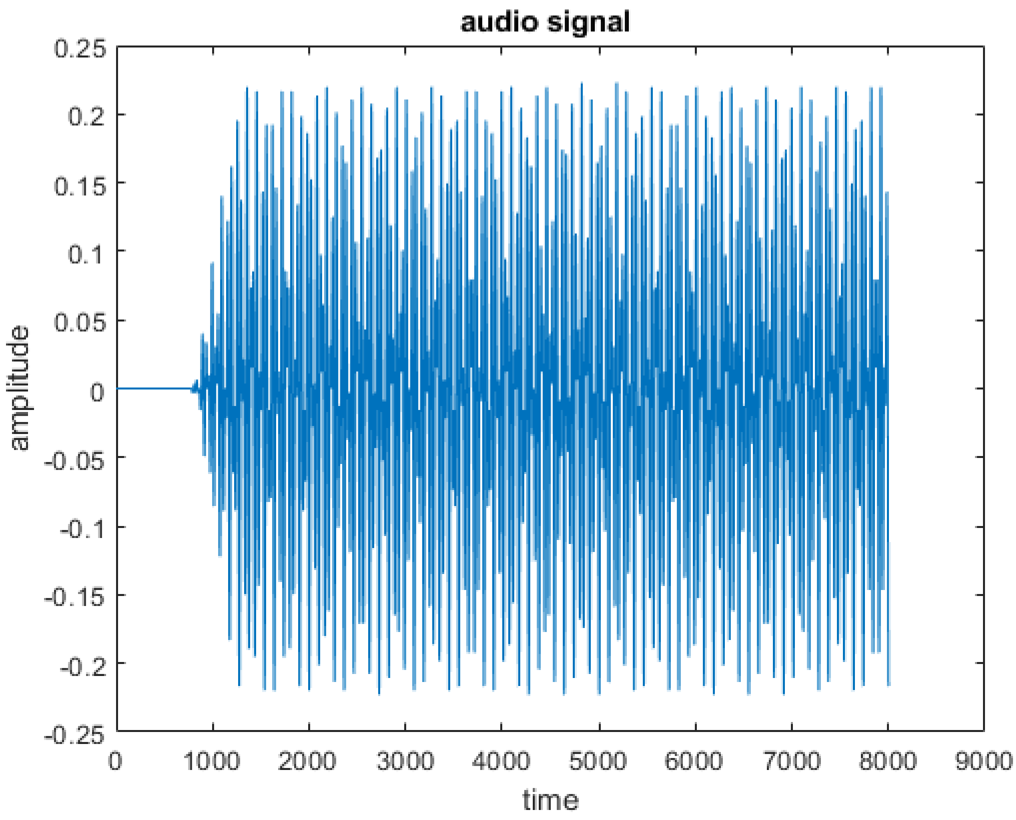
Audio signal : '1.wav'

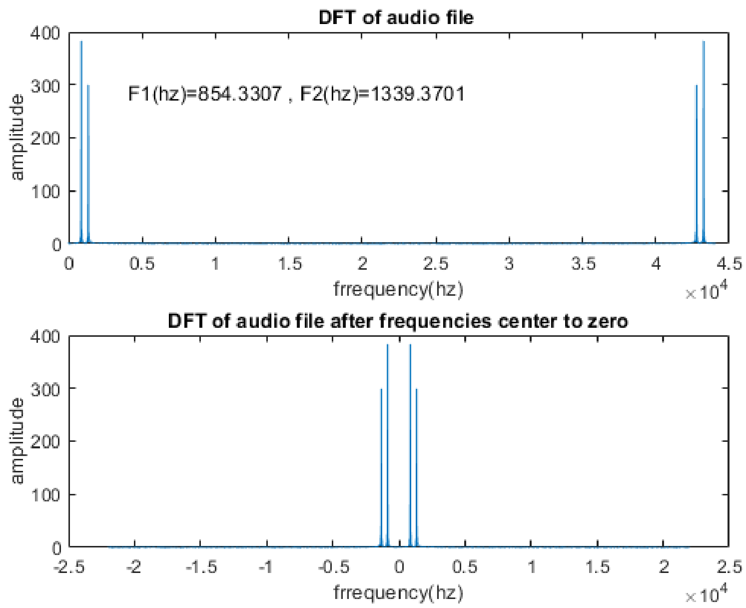




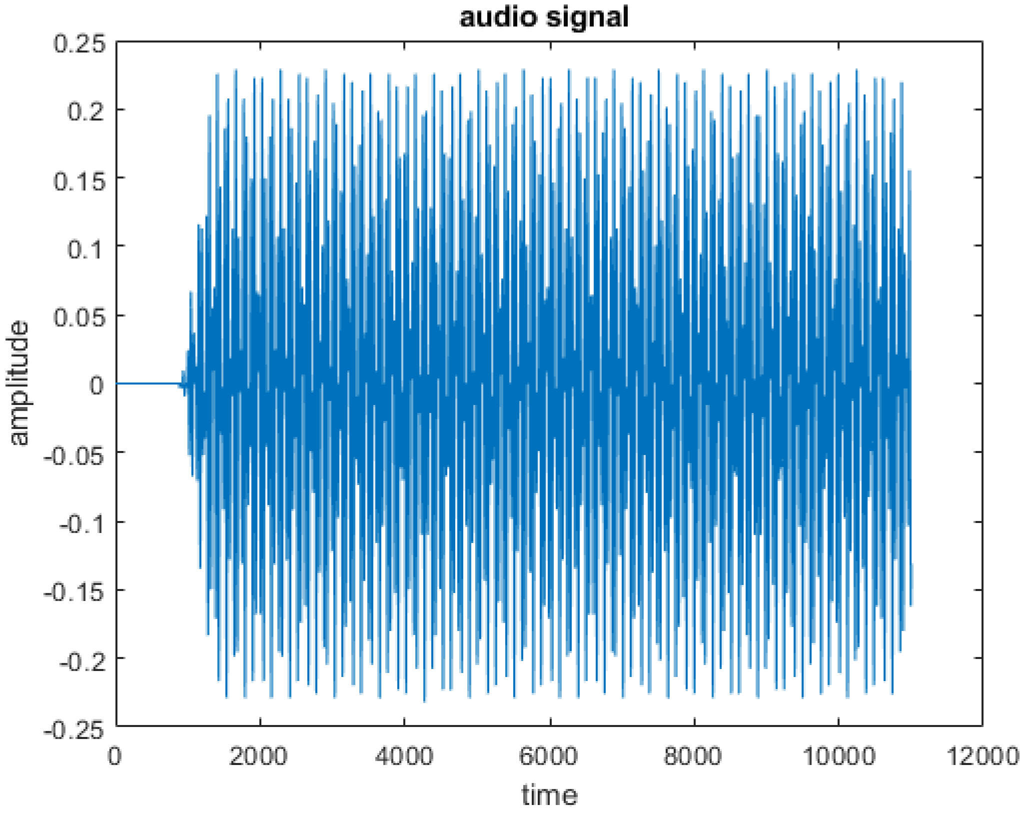


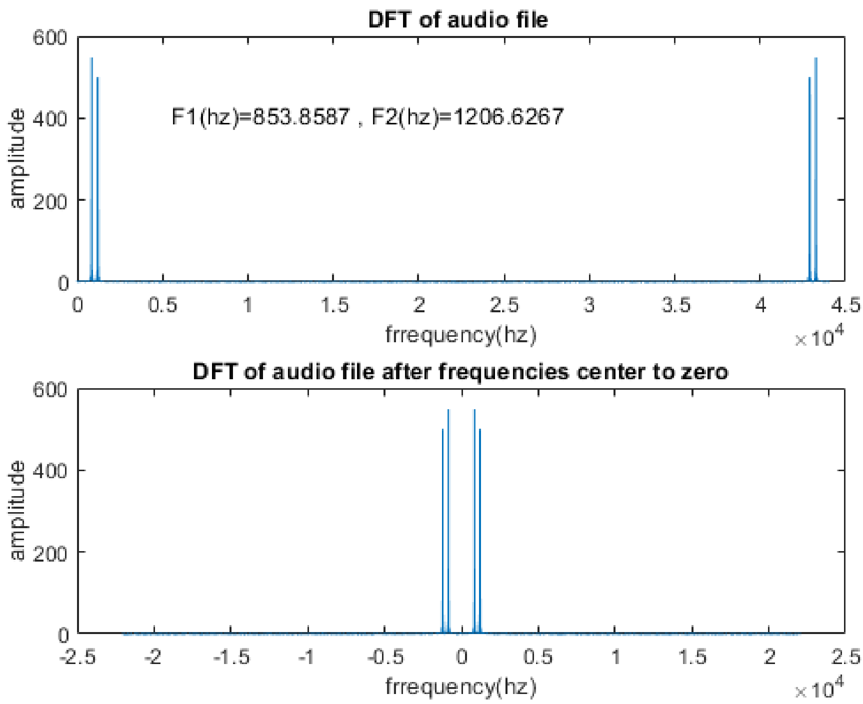
Audio signal : '2.wav'



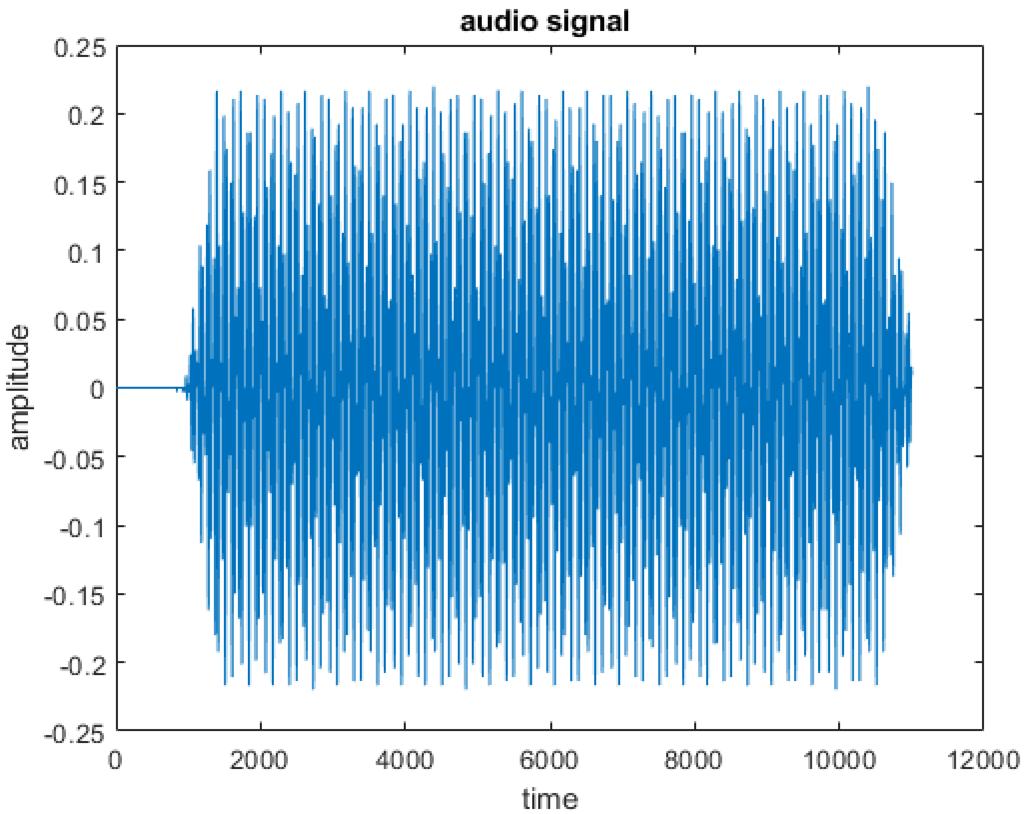


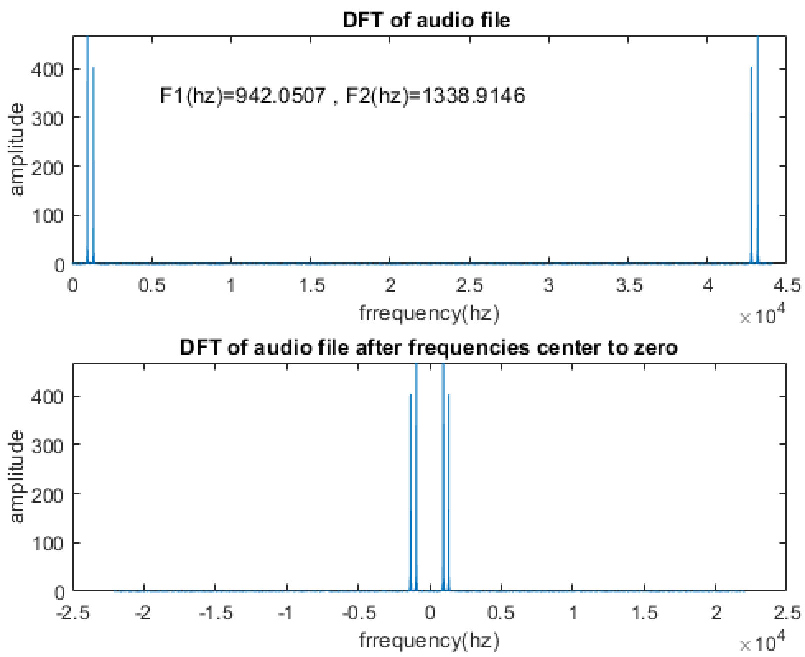
Audio signal : '3.wav'



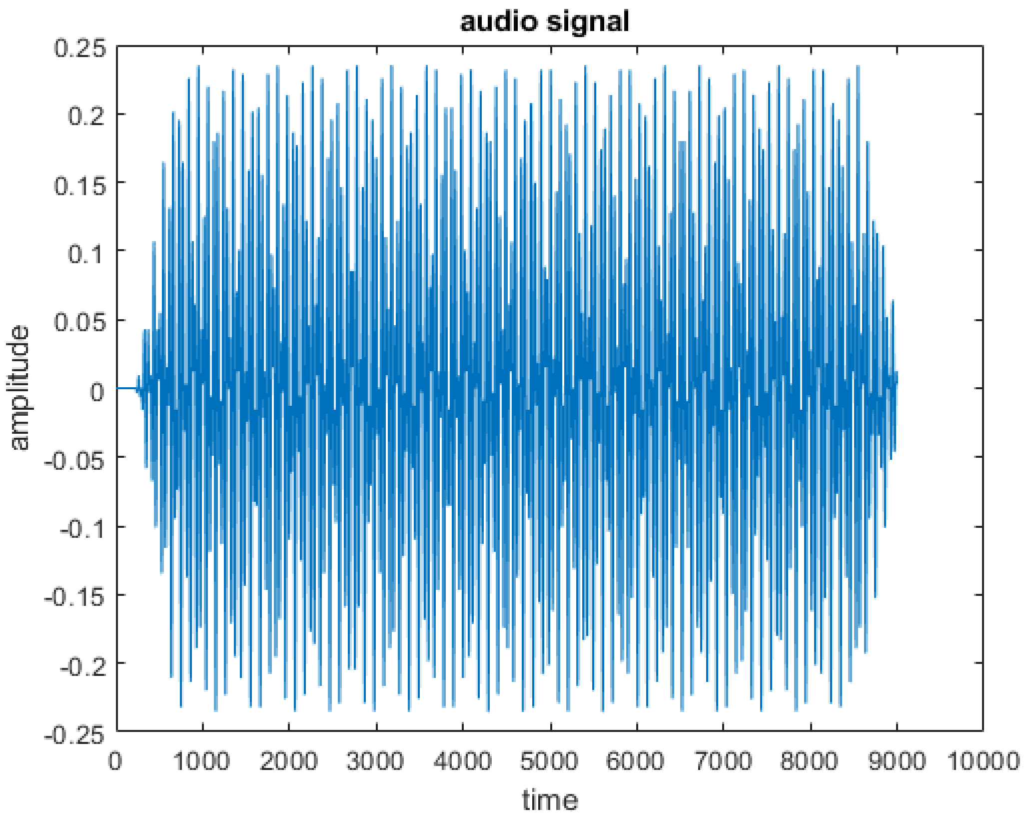


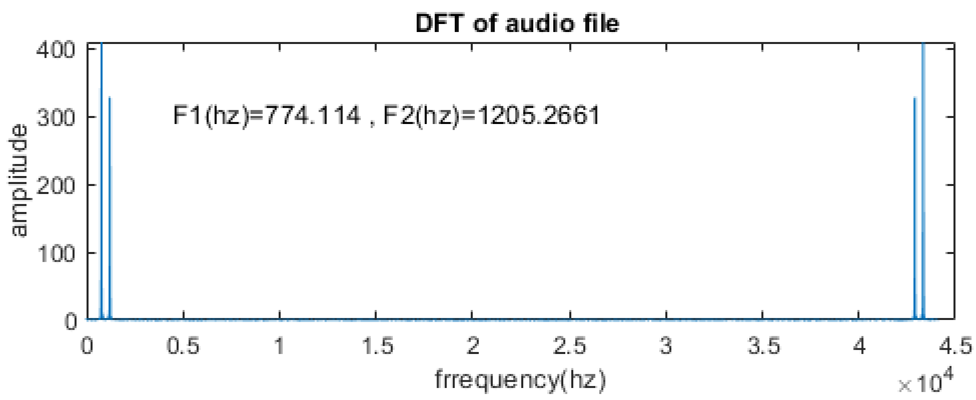
Audio signal : '4.wav'

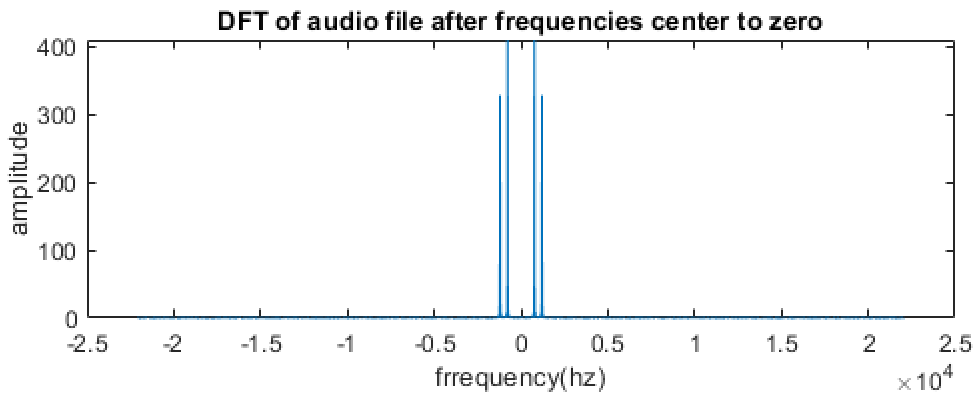




Audio signal : '5.wav'







Editor - Q6.m | Command Window

```
1.wav audio has frequencies  f1= 853.012 Hz &  f2=1474.6988 Hz
2.wav audio has frequencies  f1= 854.3307 Hz &  f2=1339.3701 Hz
3.wav audio has frequencies  f1= 853.8587 Hz &  f2=1206.6267 Hz
4.wav audio has frequencies  f1= 942.0507 Hz &  f2=1338.9146 Hz
5.wav audio has frequencies  f1= 774.114 Hz &  f2=1205.2661 Hz
6.wav audio has frequencies  f1= 776.0048 Hz &  f2=1340.3719 Hz
7.wav audio has frequencies  f1= 771.6214 Hz &  f2=1477.1038 Hz
8.wav audio has frequencies  f1= 850 Hz &  f2=1480 Hz
9.wav audio has frequencies  f1= 1338.3271 Hz &  f2=848.9388 Hz
10.wav audio has frequencies  f1= 769.0387 Hz &  f2=1478.1523 Hz
11.wav audio has frequencies  f1= 938.8265 Hz &  f2=1338.3271 Hz
12.wav audio has frequencies  f1= 1208.4894 Hz &  f2=769.0387 Hz
13.wav audio has frequencies  f1= 769.0387 Hz &  f2=1338.3271 Hz
14.wav audio has frequencies  f1= 1338.3271 Hz &  f2=848.9388 Hz
15.wav audio has frequencies  f1= 699.1261 Hz &  f2=1338.3271 Hz
16.wav audio has frequencies  f1= 1338.3271 Hz &  f2=848.9388 Hz
17.wav audio has frequencies  f1= 1208.4894 Hz &  f2=848.9388 Hz
18.wav audio has frequencies  f1= 850 Hz &  f2=1340 Hz
19.wav audio has frequencies  f1= 1208.4894 Hz &  f2=699.1261 Hz
20.wav audio has frequencies  f1= 699.1261 Hz &  f2=1338.3271 Hz
21.wav audio has frequencies  f1= 1478.1523 Hz &  f2=848.9388 Hz
22.wav audio has frequencies  f1= 769.0387 Hz &  f2=1478.1523 Hz
23.wav audio has frequencies  f1= 769.0387 Hz &  f2=1478.1523 Hz
24.wav audio has frequencies  f1= 1208.4894 Hz &  f2=699.1261 Hz
25.wav audio has frequencies  f1= 1208.4894 Hz &  f2=699.1261 Hz
26.wav audio has frequencies  f1= 1208.4894 Hz &  f2=769.0387 Hz
27.wav audio has frequencies  f1= 1338.3271 Hz &  f2=848.9388 Hz
fx >>
```

Inferences/comments:

1) Normalising the frequency axis by multiply the (fs/N) to the K axis. Where fs=sampling frequency and N is the length of the signal , then we can find the frequency components in a audio signal from the magnitude response(peak location of the impulses gives the frequencies)

2) by matching the given audio signals to DTMF signal then

	column 1 1209 Hz	column 2 1336 Hz	column 3 1477 Hz	column 4 1633 Hz
row 1 697 Hz	1	2	3	A
row 2 770 Hz	4	5	6	B
row 3 852 Hz	7	8	9	C
row 4 941 Hz	*	0	#	D

Audio signal match to DTMF signal	
1.wav---------→ 9	14.wav-------→ 8
2.wav---------→ 8	15.wav-------→ 2
3.wav---------→ 7	16.wav-------→ 8
4.wav---------→ 0	17.wav-------→ 7
5.wav---------→ 4	18.wav-------→ 8
6.wav---------→ 5	19.wav-------→ 1
7.wav---------→ 6	20.wav-------→ 2
8.wav---------→ 9	21.wav-------→ 9
9.wav---------→ 8	22.wav-------→ 6
10.wav-------→ 3	23.wav-------→ 6
11.wav-------→ 0	24.wav-------→ 1
12.wav-------→ 4	25.wav-------→ 1
13.wav-------→ 5	26.wav-------→ 4
	27.wav-------→ 8