भारतीय प्रौद्योगिकी संस्थान तिरुपति

TIRUPATI

ADVANCED SIGNAL ANALYSIS AND PROCESSING LAB

Lab sheet. No: 05

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

Q1. Following are the signals of different nature like complex, real, even, odd, etc. Take the Fourier transform of those signals and identify the nature of the transformed signals.

(a) x1 = sin(200πn/8000); −100 ≤ n ≤ 100

(b) x2 = cos(200πn/8000); −100 ≤ n ≤ 100

(c) x3 = e^0.02n; −100 ≤ n ≤ 100

AIM: To Take the Fourier transform of those signals and identify the nature of the transformed signals.

(a) x1 = sin(200πn/8000); −100 ≤ n ≤ 100

(b) x2 = cos(200πn/8000); −100 ≤ n ≤ 100

(c) x3 = e^0.02n; −100 ≤ n ≤ 100

Short Theory:

If f(x) is a complex function, f(x) = re(x) + i im(x), the Fourier integral splits into four components

F(w)=$\int_{-\infty}^{\infty} re(x)\cos(wx)\,dx + i\int_{-\infty}^{\infty} re(x)\sin(wx)\,dx + \int_{-\infty}^{\infty} im(x)\cos(wx)\,dx + i\int_{-\infty}^{\infty} im(x)\sin(wx)\,dx$

so that integrals 1 and 3constitute the real part of the transform, and integrals 2 and 4 its imaginary part.

If *f(x)* is...	And F(w) is...
real and even	real and even
Real and odd	Imaginary and odd
imaginary and even	imaginary and even
imaginary and odd	real and odd

Key Commands:

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

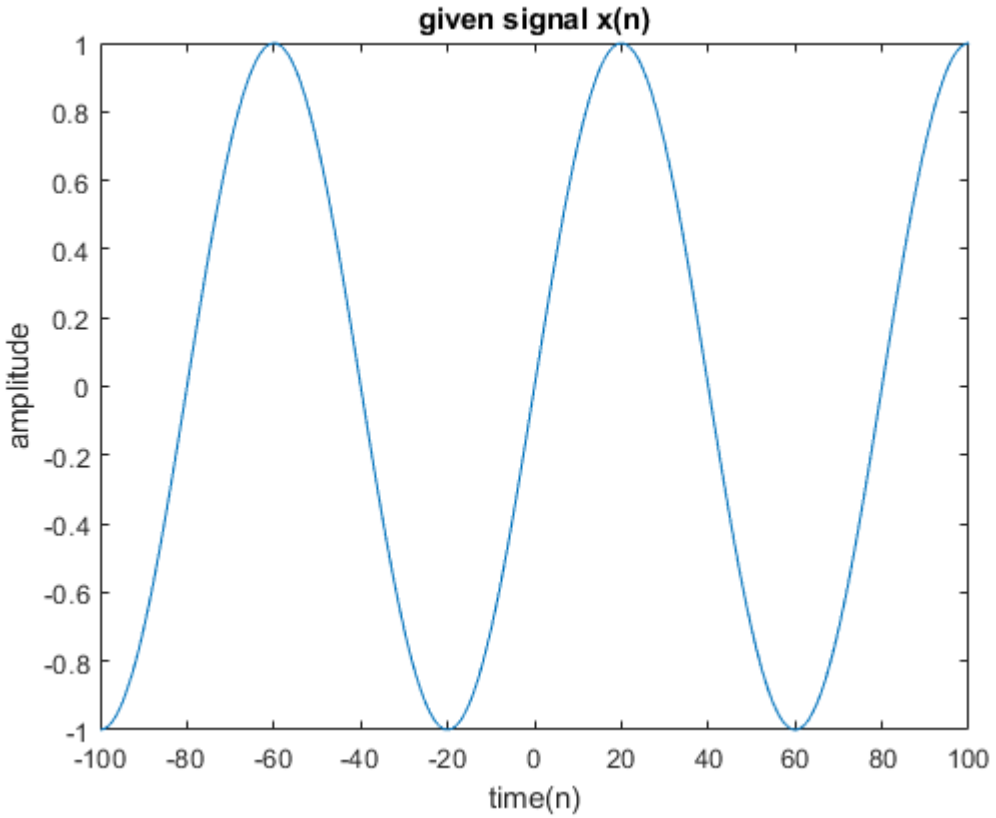
Abs() % abs(x) gives the magnitude of x.

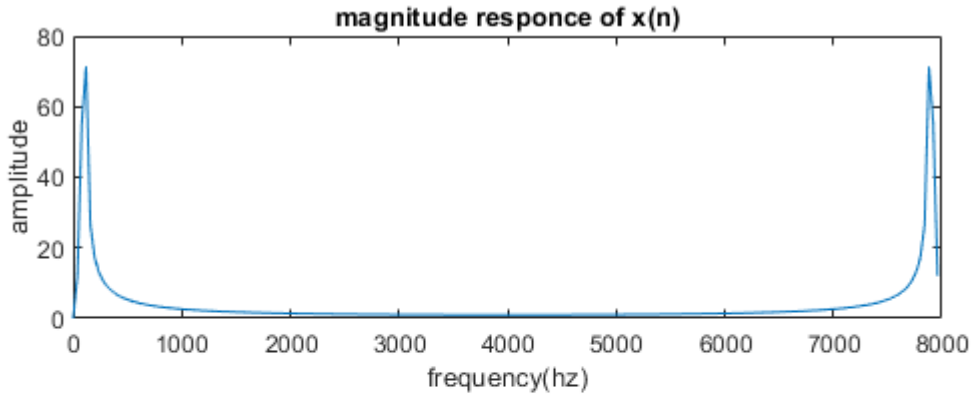
Angle % angle(z) returns the phase angle in the interval [-π,π] for each element of a complex array z.

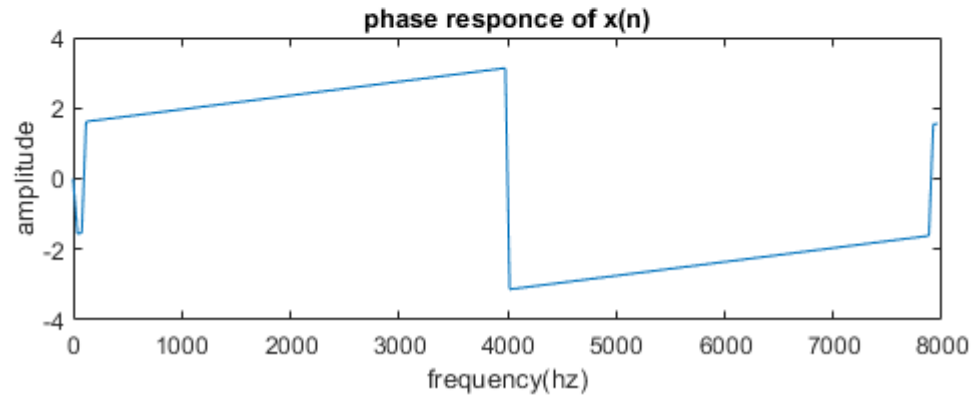
sin % it gives the values of sin(x)

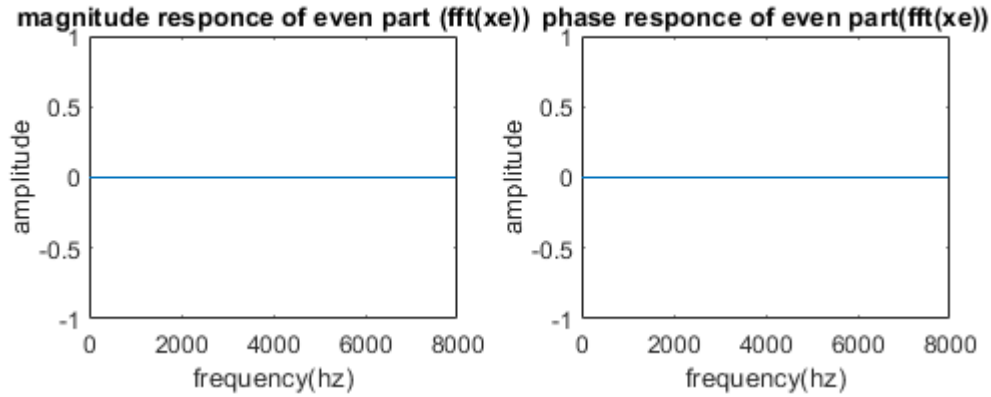
cos % it gives the values of cos(x)

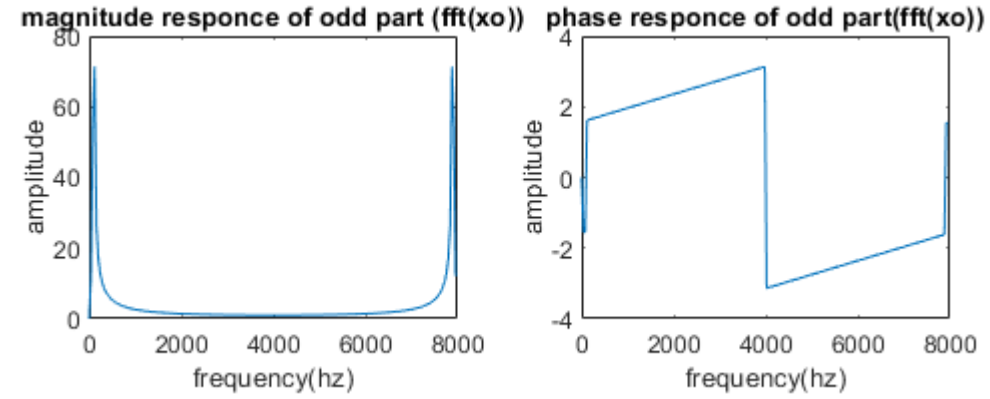
Plots:

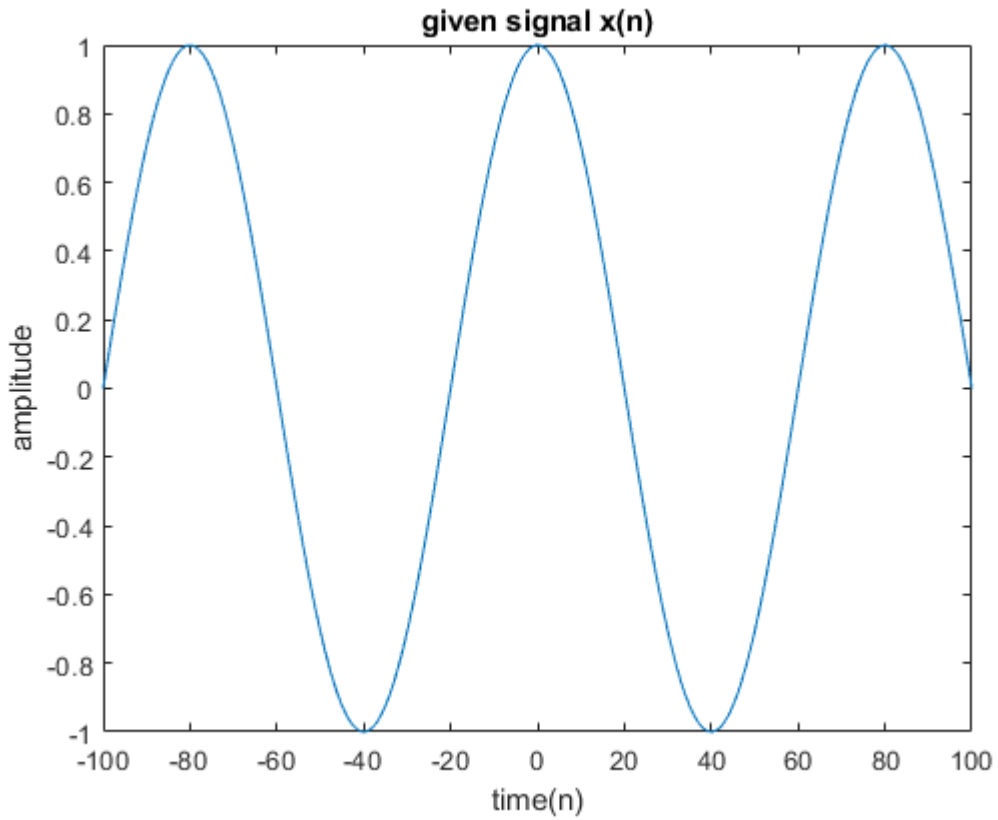

given signal x(n)
amplitude
time(n)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-100
-80
-60
-40
-20
0
20
40
60
80
100

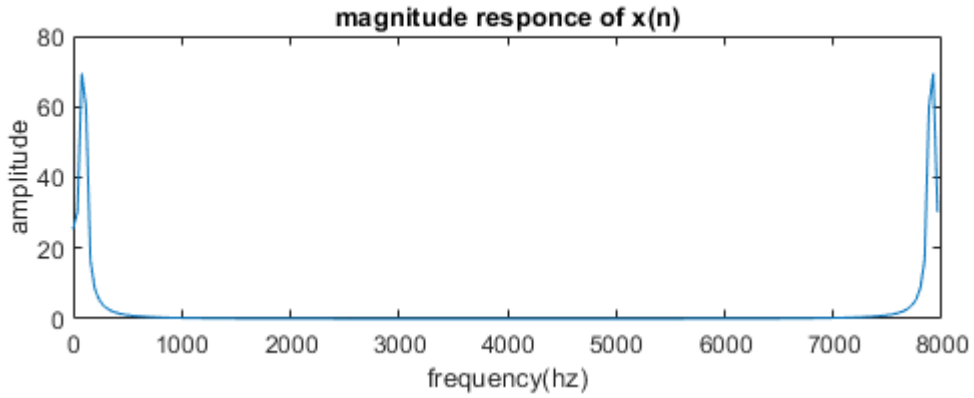

magnitude responce of x(n)
amplitude
frequency(hz)
80
60
40
20
0
0
1000
2000
3000
4000
5000
6000
7000
8000

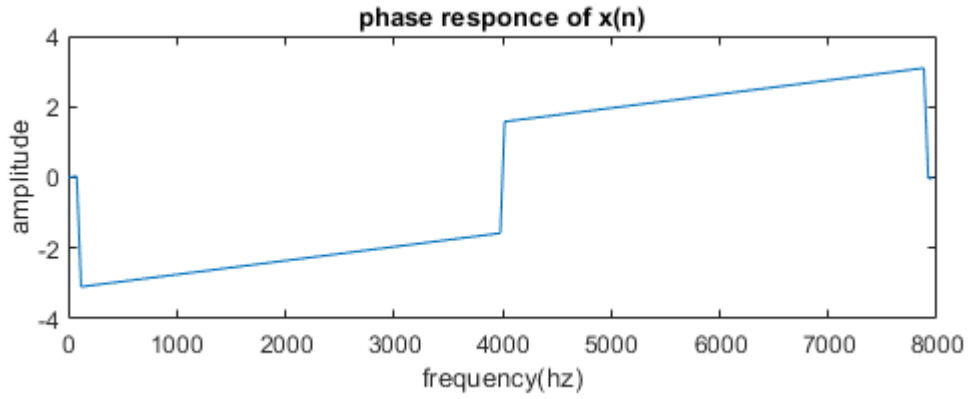

phase responce of x(n)
amplitude
frequency(hz)
4
2
0
-2
-4
0
1000
2000
3000
4000
5000
6000
7000
8000

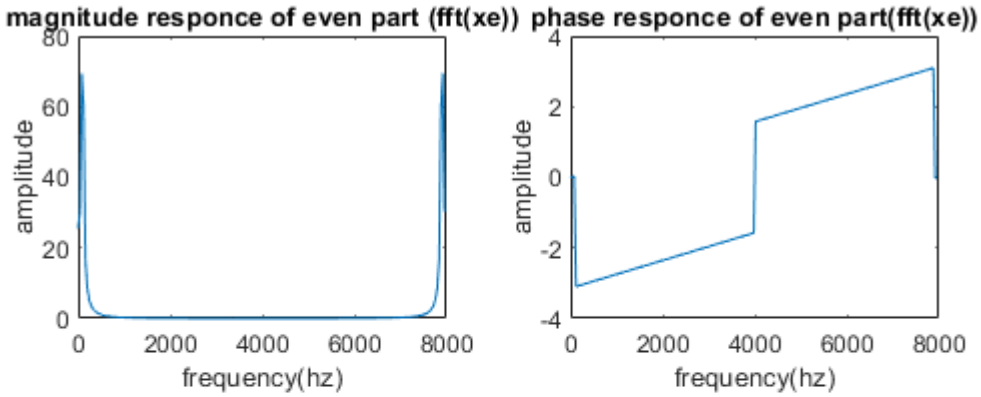

magnitude responce of even part (fft(xe))
amplitude
frequency(hz)
phase responce of even part(fft(xe))
amplitude
frequency(hz)


magnitude responce of odd part (fft(xo))
amplitude
frequency(hz)
phase responce of odd part(fft(xo))
amplitude
frequency(hz)

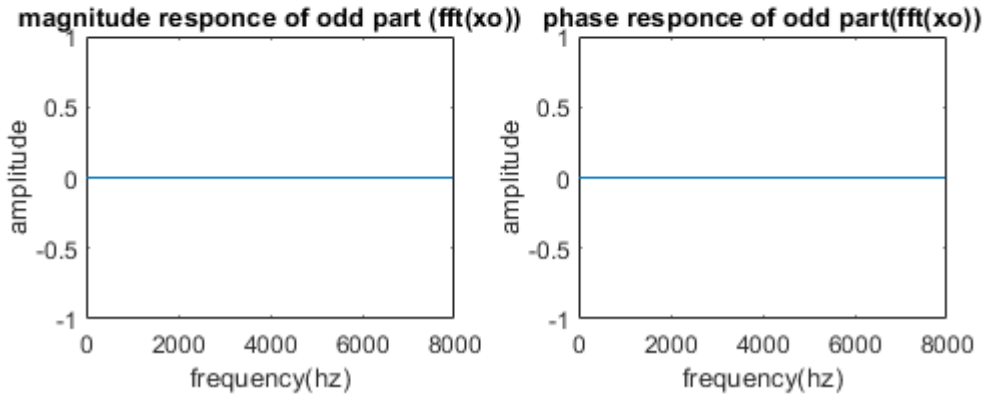

given signal x(n)
amplitude
time(n)


magnitude responce of x(n)
amplitude
frequency(hz)
0
20
40
60
80
0
1000
2000
3000
4000
5000
6000
7000
8000

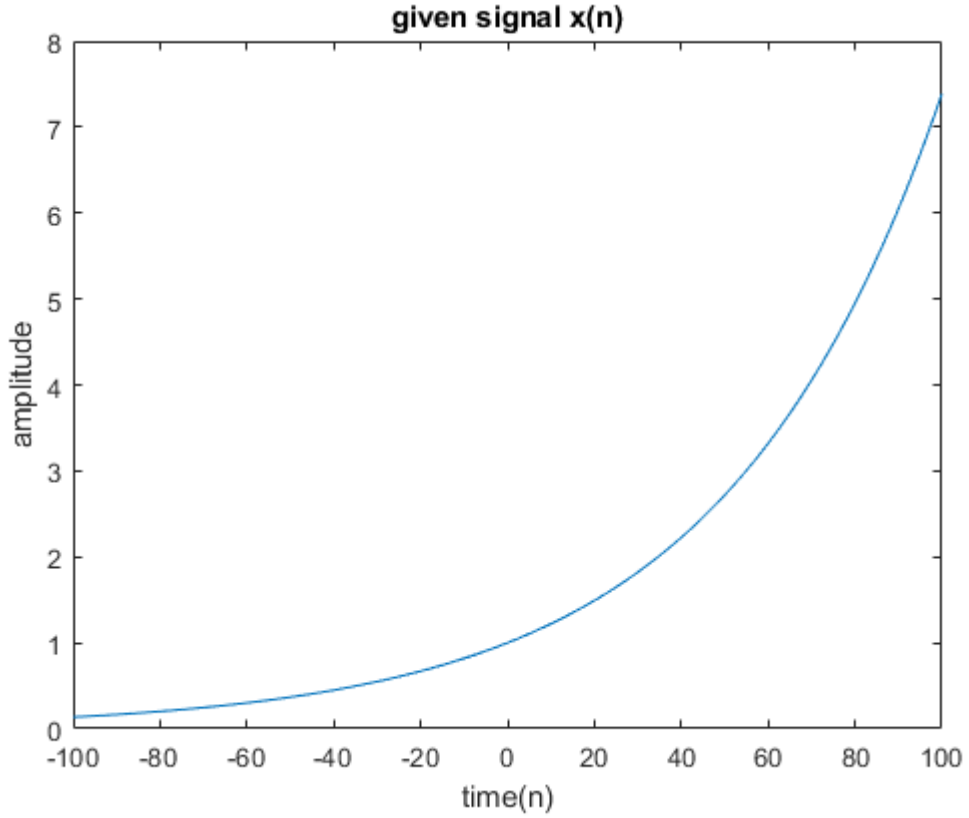

phase responce of x(n)
amplitude
frequency(hz)
-4
-2
0
2
4
0
1000
2000
3000
4000
5000
6000
7000
8000

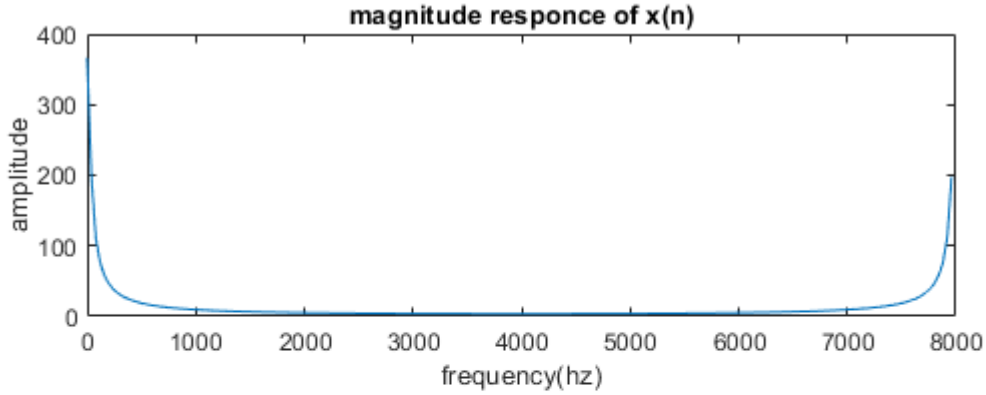

magnitude responce of even part (fft(xe))
amplitude
frequency(hz)
0
20
40
60
80
0
2000
4000
6000
8000

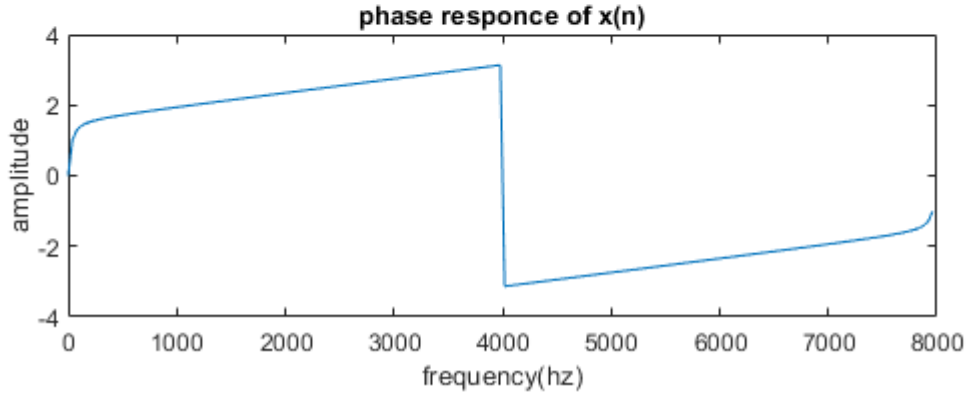
phase responce of even part(fft(xe))
amplitude
frequency(hz)
-4
-2
0
2
4
0
2000
4000
6000
8000

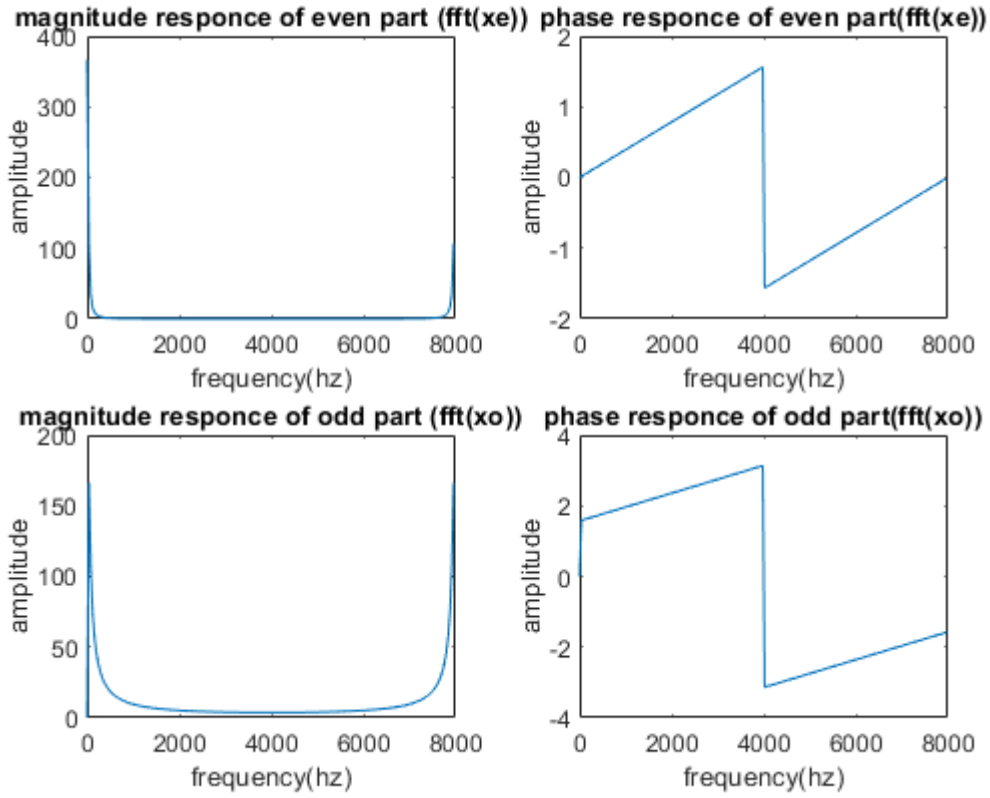

magnitude responce of odd part (fft(xo))
amplitude
frequency(hz)
-1
-0.5
0
0.5
1
0
2000
4000
6000
8000

phase responce of odd part(fft(xo))
amplitude
frequency(hz)
-1
-0.5
0
0.5
1
0
2000
4000
6000
8000


given signal x(n)
amplitude
time(n)
8
7
6
5
4
3
2
1
0
-100
-80
-60
-40
-20
0
20
40
60
80
100


magnitude responce of x(n)
amplitude
frequency(hz)
400
300
200
100
0
0
1000
2000
3000
4000
5000
6000
7000
8000


phase responce of x(n)
amplitude
frequency(hz)
4
2
0
-2
-4
0
1000
2000
3000
4000
5000
6000
7000
8000



Inferences/comments:

(a) given x1 = sin(200πn/8000); is a real and odd signal then the fourier transform of the x1 is a imaginary and odd signal.

(b) x2 = cos(200πn/8000); is a real and even signal then the fourier transform of the x2 is a real and even signal.

(c) x3 = e^0.02n; −100 ≤ n ≤ 100; is a neither even nor odd signal

Q2. Generate the signal x(t) = 2cos(2000πt) with a sampling rate of F s = 8000Hz. Use the Matlab DFT to compute the signal spectrum (amplitude and power spectrum) with the frequency resolution to be equal to (a) 1 Hz, (b) 8 Hz, (c) 16 Hz. Explain, in brief, about your observations.

AIM: To generate the signal x(t) = 2cos(2000πt) with a sampling rate of Fs = 8000Hz and compute the spectrum with the frequency resolution to be equal to (a) 1 Hz, (b) 8 Hz, (c) 16 Hz.

Short Theory:

The frequency resolution in a DFT is given by sampling rate upon the total number of samples.

$$\Delta f = \frac{f_s}{N}$$

The result of an N-point DFT is an N element complex vector. For strictly real data, only the first 1+N/2 bins matter, as the others are a redundant complex conjugate image. For properly bandlimited data sampled at Fs, the frequency range is 0 to Fs/2. Divide Fs/2 by N/2, and you get Fs/N as the frequency spacing of 1+N/2 equally spaced DFT results, including both end points, DC and Fs/2. So Fs/N is the DFT result bin spacing of an N-point DFT.

Key Commands:

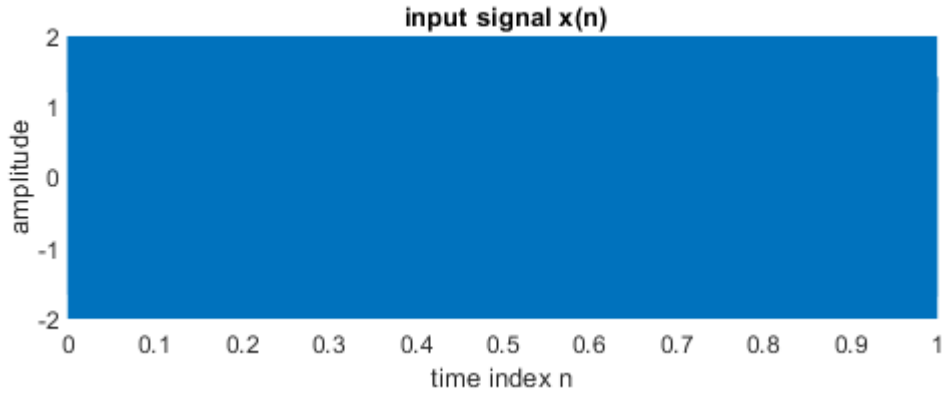
fft(x) % fft(X) computes the discrete Fourier transform (DFT) of X using a fast Fourier transform (FFT) algorithm.

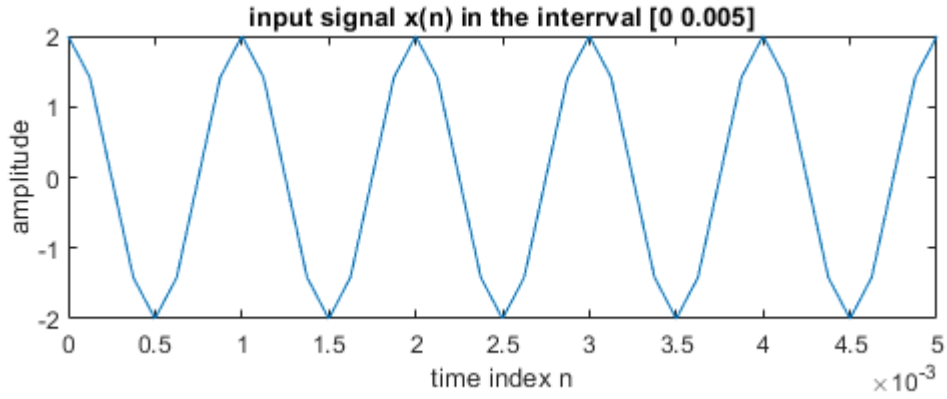
Abs() % abs(x) gives the magnitude of x.

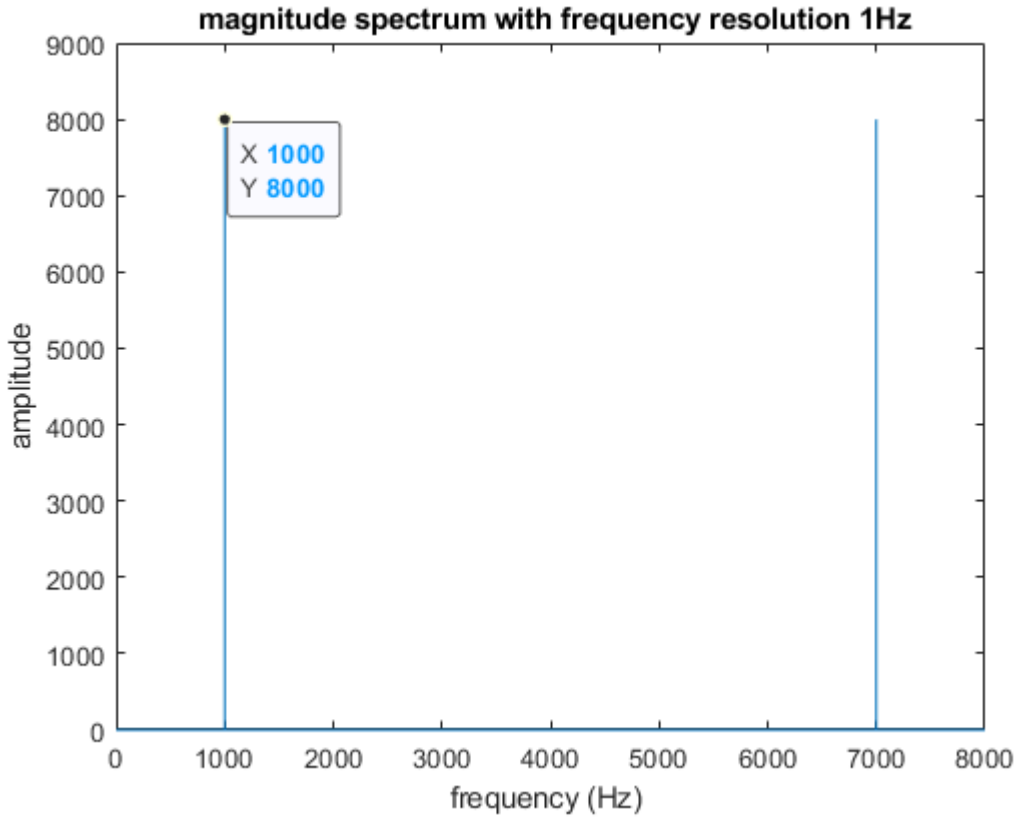
Angle % angle(z) returns the phase angle in the interval [-π,π] for each element of a complex array z.

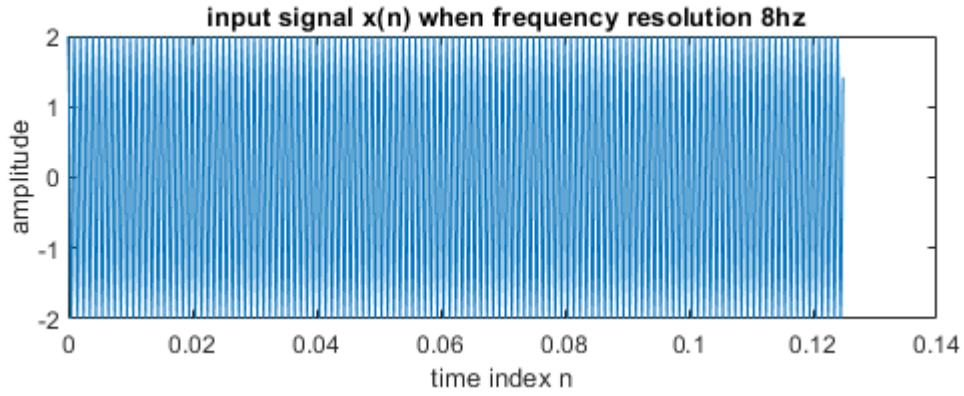
cos % it gives the values of cos(x)

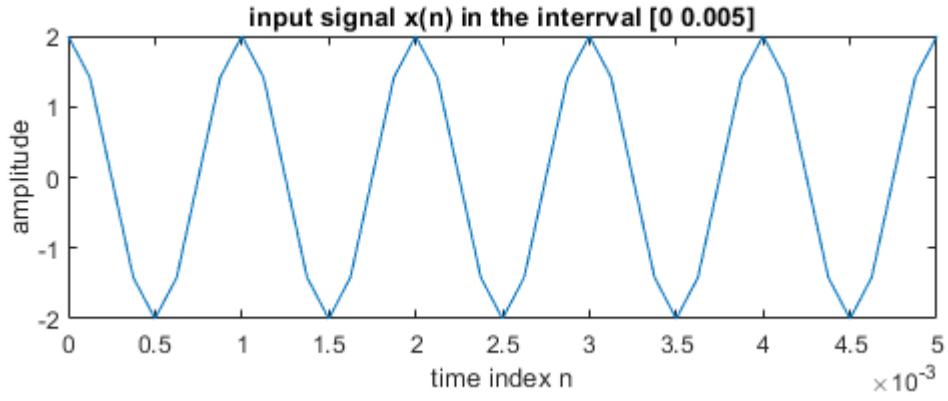
Plots:

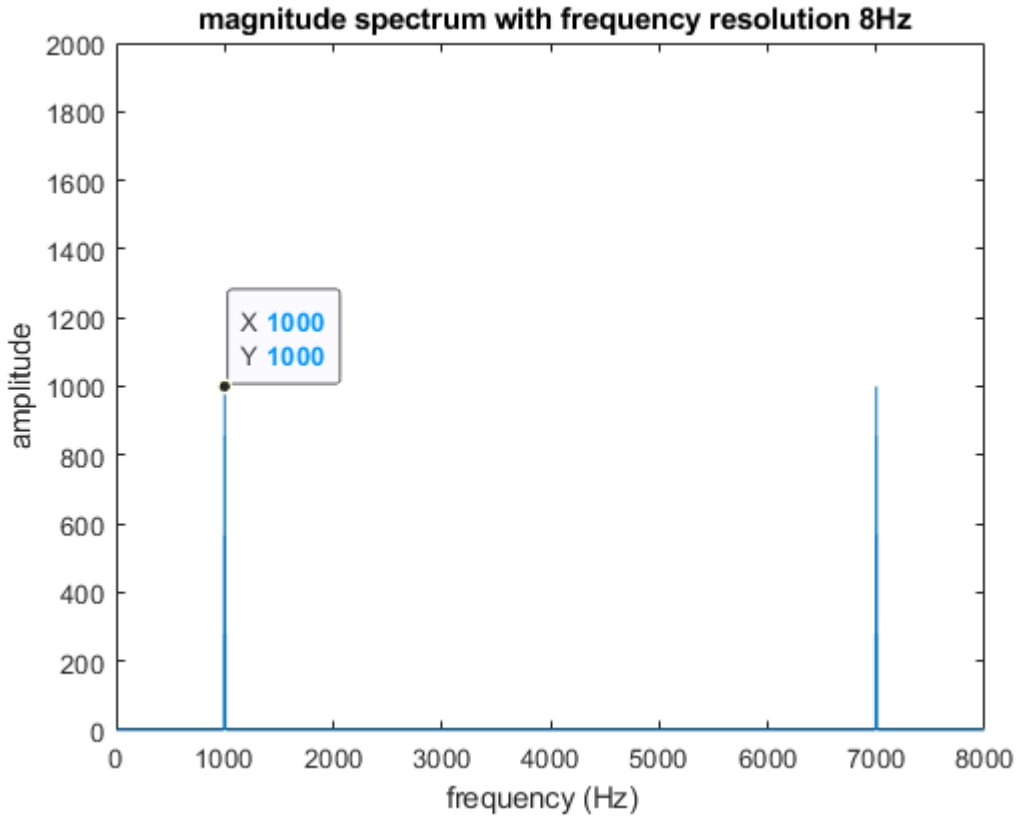

input signal x(n)
amplitude
time index n

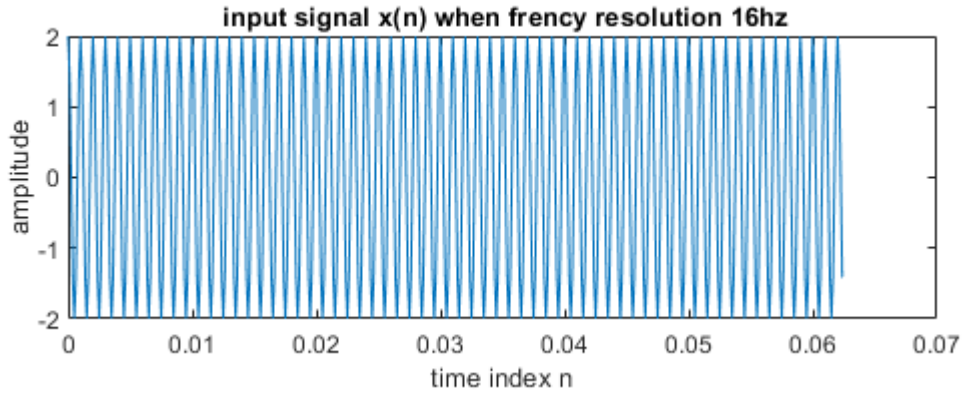

input signal x(n) in the interrval [0 0.005]
amplitude
time index n
×10^-3

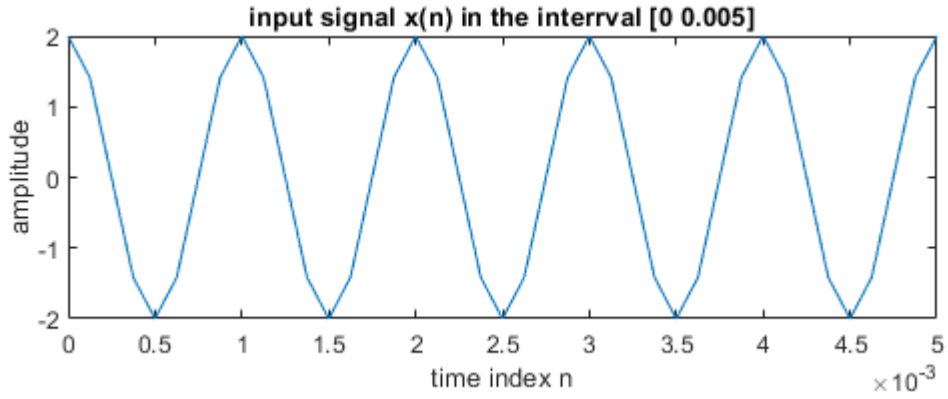

magnitude spectrum with frequency resolution 1Hz
X 1000
Y 8000
amplitude
frequency (Hz)

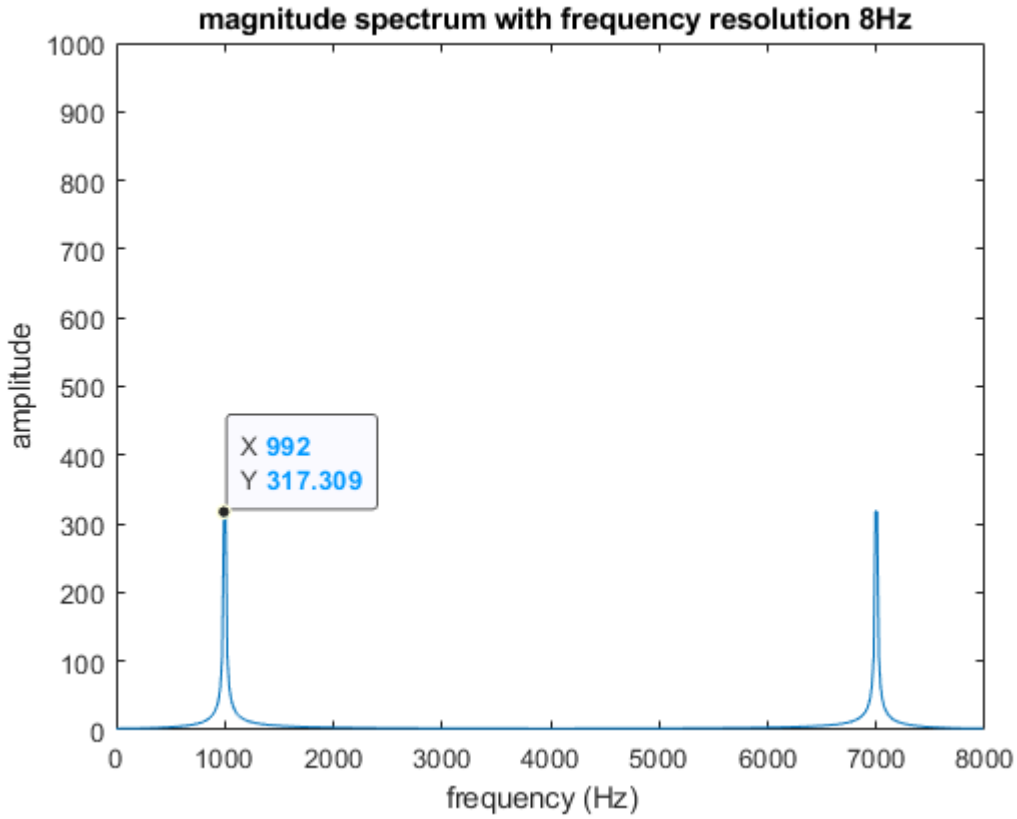

input signal x(n) when frequency resolution 8hz
amplitude
time index n


input signal x(n) in the interrval [0 0.005]
amplitude
time index n
×10^-3


magnitude spectrum with frequency resolution 8Hz
X 1000
Y 1000
amplitude
frequency (Hz)







Inferences/comments:

1) If the frequency resolution increases then the number of samples decreases
2) If the number of sample points decreases beyond length of the signal points then we can not reconstruct the signal.
3) If the frequency resolution increases number of sidelobes will increase in Fourier transform.

Q3. A 1 kHz sinusoid is sampled at 8 kHz. The 128-point FFT is performed to compute X(k). What is the computational resolution? At what frequency indices k we expect to observe peaks in |X(k)|? Can we observe the line spectrum?

AIM: To generate the sinusoid with frequency f=1Khz and sampling frequency fs=8Khz and compute the 128 point FFT.

Short Theory:

DFT of an N -point sequence xn , n = 0,1,2,...,N −1 is defined as

$$A_k = \sum_{n=0}^{N-1} e^{-i\frac{2\pi}{N}kn} a_n$$

$$A_k = \sum_{n=0}^{N-1} W_N^{kn} a_n$$

$$W_N = e^{-i\frac{2\pi}{N}}$$

and W kN for k =0 ...N −1 are called the Nth roots of unity.

Key Commands:

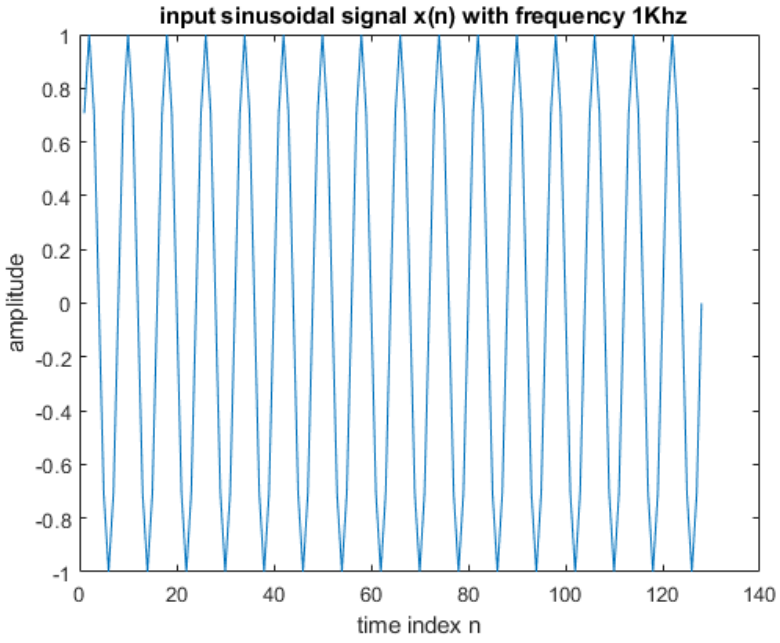
fft(x) % fft(X) computes the discrete Fourier transform (DFT) of X using a fast Fourier transform (FFT) algorithm.

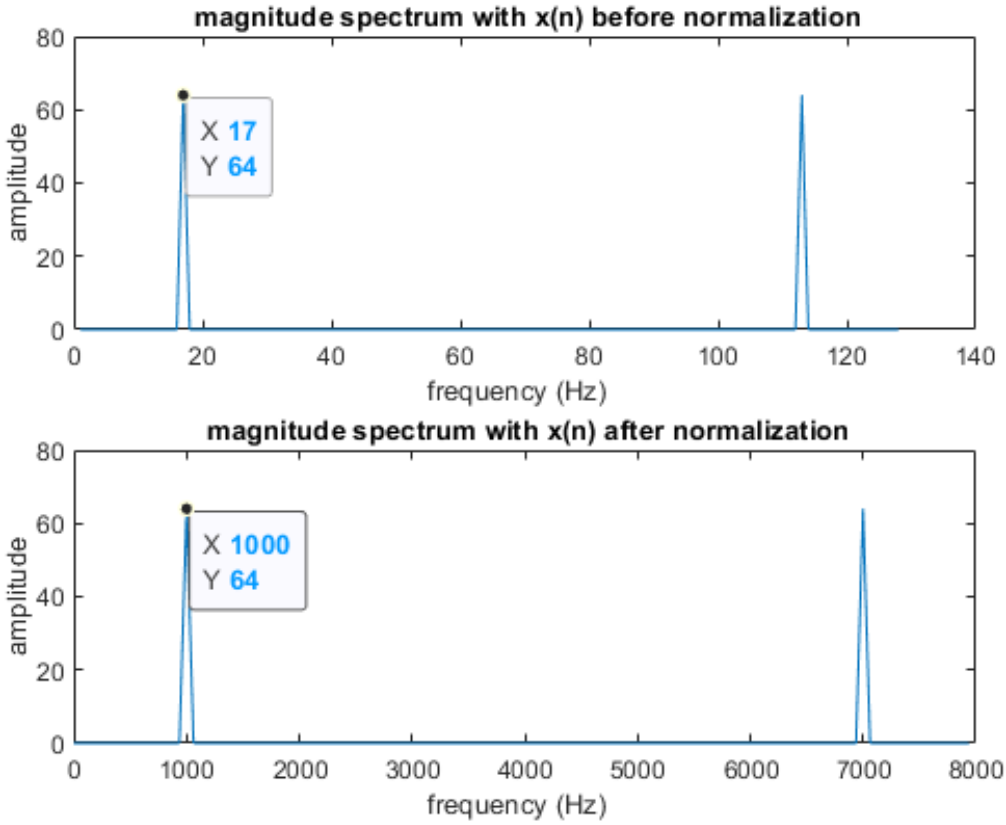
Abs() % abs(x) gives the magnitude of x.

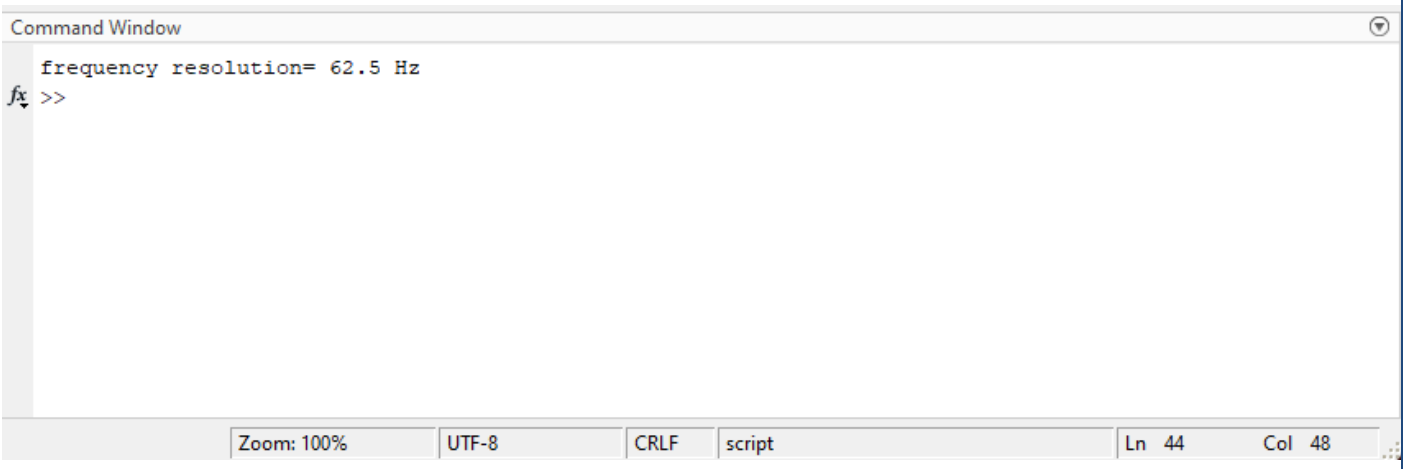
Angle % angle(z) returns the phase angle in the interval [-π,π] for each element of a complex array z.

sin % it gives the values of sin(x) .

Plots:







Inferences/comments:

1) If we perform the 128-point fft to a given signal, it takes first 128 points of a signal and perform the Fourier transform.
2) If we perform the 128-point fft to a given signal, then the length of the output signal is 128 points.
3) If the length of the given signal is less than the 128 points, interpolation occurs in frequency domain.
4) If the length of the given signal is more than the 128 points , we can not re construct the original signal from the spectrum.
5) For a given question the frequency resolution = fs/N=62.5 Hz.

Q4. Compute the 16-point and 32-point DFTs of the 4-point sequence x(n) = {1, 0.5, 0, −0.5}. Plot their magnitudes and compare them.

AIM: To compute and plot the 16-point and 32-point DFT of 4-point sequence of x(n)= {1, 0.5, 0, −0.5}.

Short Theory:

DFT of an N -point sequence xn , n = 0,1,2,...,N −1 is defined as

$$A_k = \sum_{n=0}^{N-1} e^{-i\frac{2\pi}{N}kn} a_n$$

$$A_k = \sum_{n=0}^{N-1} W_N^{kn} a_n$$

$$W_N = e^{-i\frac{2\pi}{N}}$$

and W kN for k =0 ...N −1 are called the Nth roots of unity.

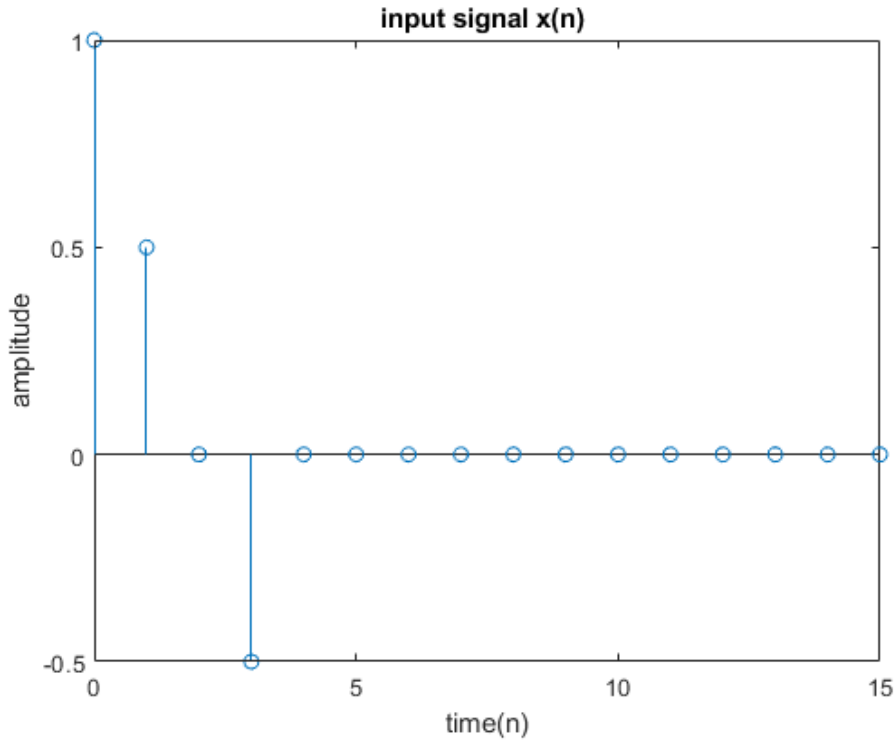
Key Commands:

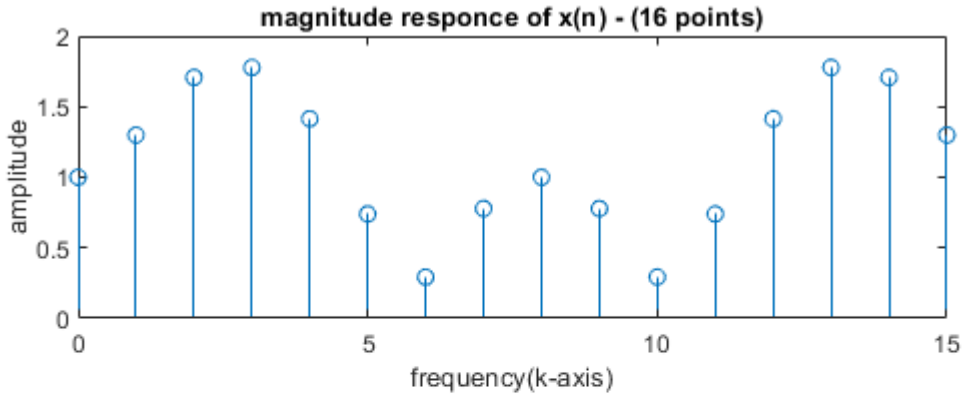
fft(x) % fft(X) computes the discrete Fourier transform (DFT) of X using a fast Fourier transform (FFT) algorithm.

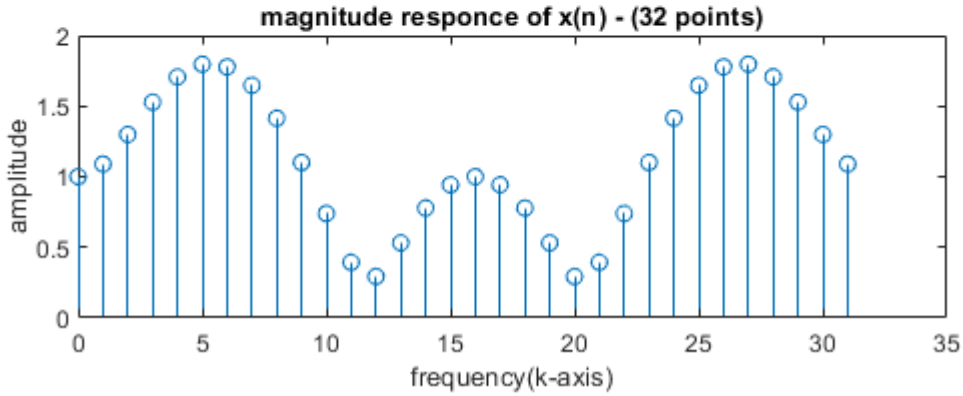
Abs() % abs(x) gives the magnitude of x.

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

Plots:







Inferences/comments:

1)As the number of points in fft increases from 16 to 32 then number of samples between two points increases in the Fourier transform.

2) If the length of the given signal is less than the 16 and 32 points, interpolation occurs in frequency domain.

Q5. Compute the 24-point DFT of the sequence in Problem above , plot the magnitude of this DFT. Now compute 24-point IDFT of this DFT and compare it with x(n) given above question.

AIM: To compute and plot the magnitude spectrum for the 24-oint DFT of the sequence x(n)= {1, 0.5, 0, −0.5}.

Short Theory:

DFT of an N -point sequence xn , n = 0,1,2,...,N −1 is defined as

$$A_k = \sum_{n=0}^{N-1} e^{-i\frac{2\pi}{N}kn} a_n$$

$$A_k = \sum_{n=0}^{N-1} W_N^{kn} a_n$$

$$W_N = e^{-i\frac{2\pi}{N}}$$

and W kN for k =0 ...N −1 are called the Nth roots of unity.

Key Commands:

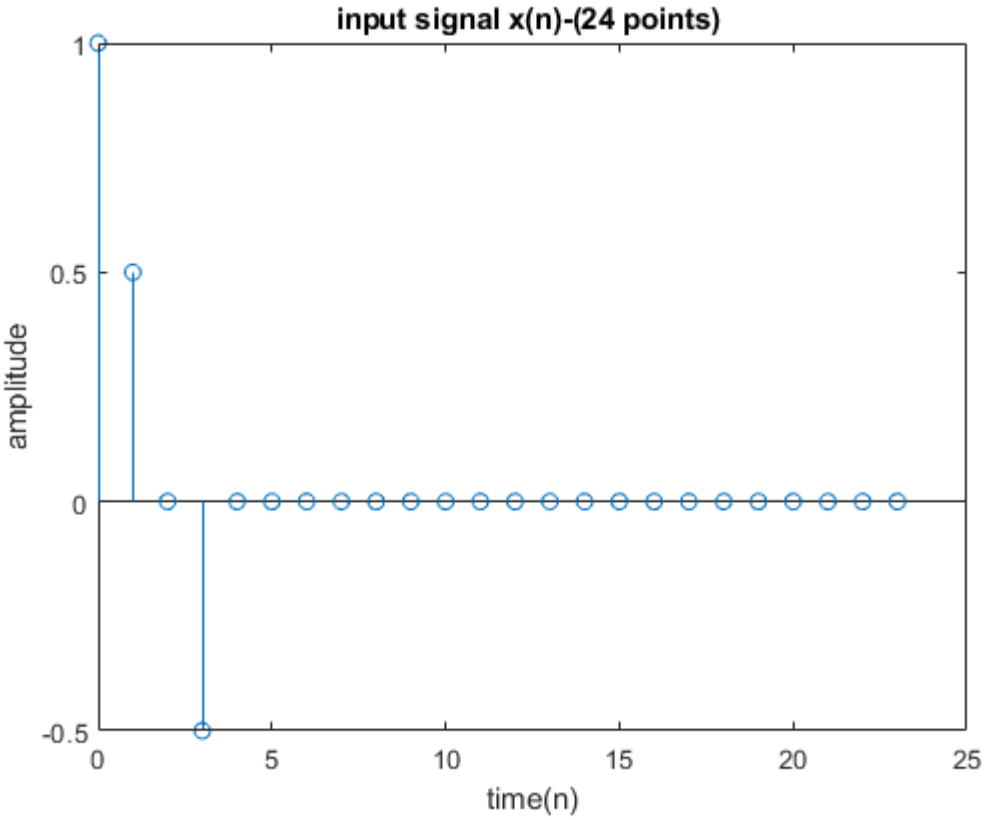
fft(x) % fft(X) computes the discrete Fourier transform (DFT) of X using a fast Fourier transform (FFT) algorithm.

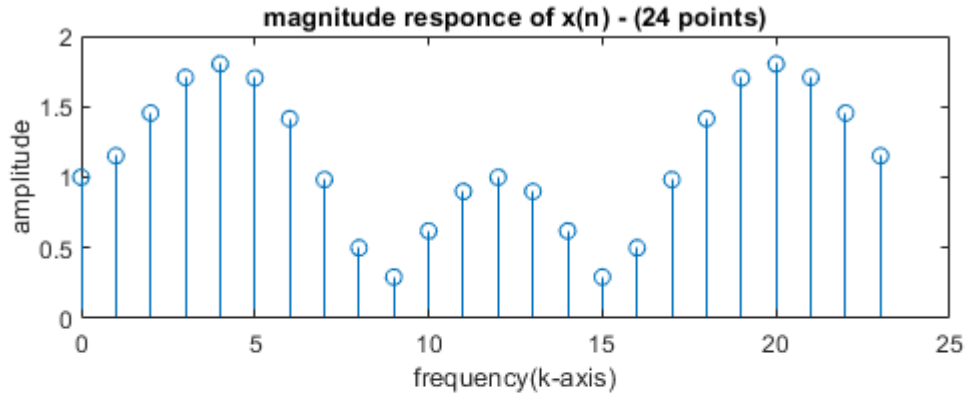
Abs() % abs(x) gives the magnitude of x.

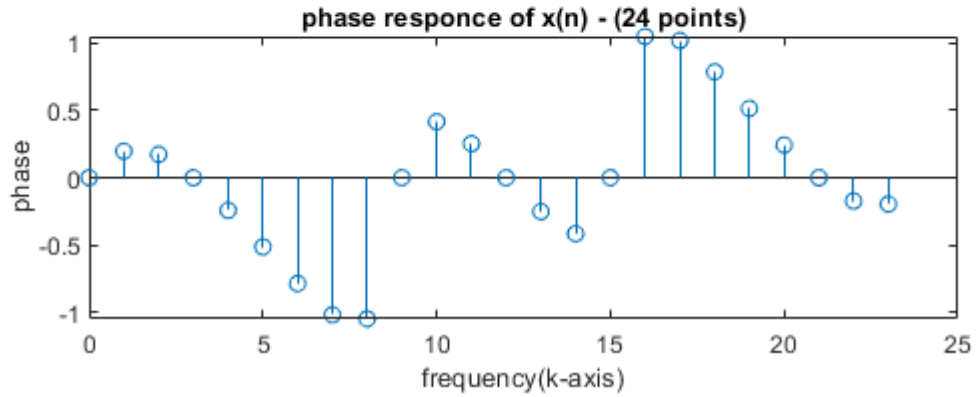
Ifft % ifft(Y) computes the inverse discrete Fourier transform of Y using a fast Fourier transform algorithm. X is the same size as Y.

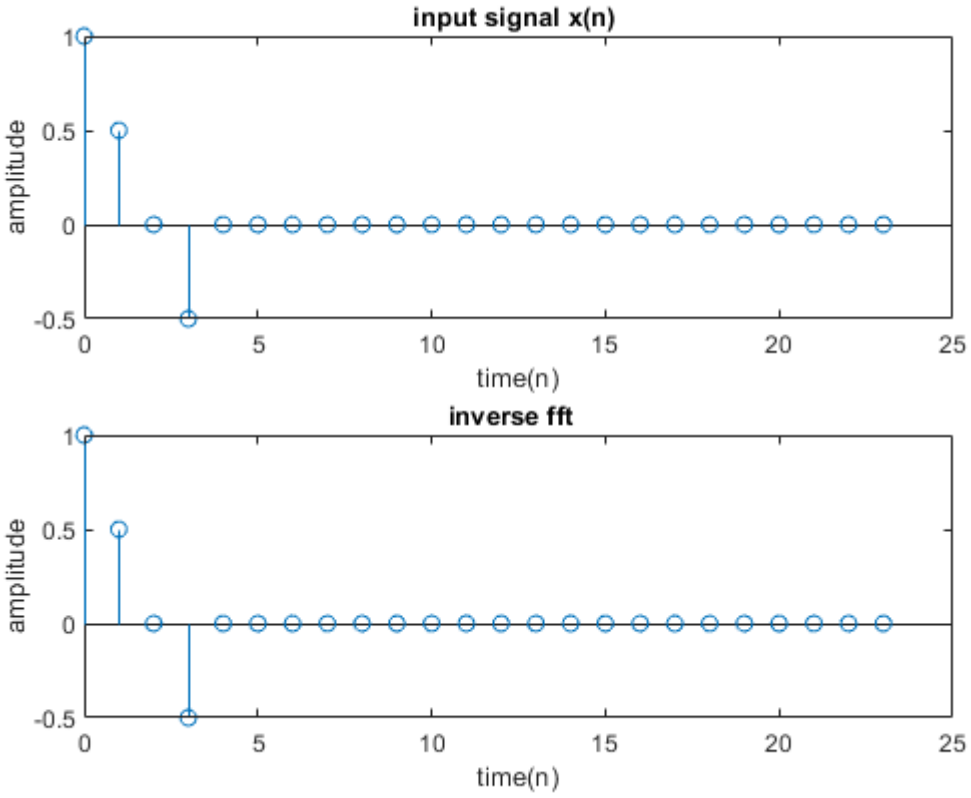
Angle % angle(z) returns the phase angle in the interval [-π,π] for each element of a complex array z.

Plots:


input signal x(n)-(24 points)
amplitude
time(n)
1
0.5
0
-0.5
0
5
10
15
20
25


magnitude responce of x(n) - (24 points)
amplitude
frequency(k-axis)
2
1.5
1
0.5
0
0
5
10
15
20
25


phase responce of x(n) - (24 points)
phase
frequency(k-axis)
1
0.5
0
-0.5
-1
0
5
10
15
20
25



Inferences/comments:

1) If the length of the given signal is less than the 24 points, interpolation occurs in frequency domain.
2) Number of fft points are greater than equal to The length of the given signal, then we can re construct the original signal from the Fourier transformed signal.
3) If we perform the ifft for given 24 point fft signal, we we exactly get the original signal.

Q6. From the real sequence x(n) = {1, −1, 2, 0.5, 0, −1, 2, 1},

(a) show that the DFT of [xe(n)] = Re[X(k)], where the even part xe(n) = [x(n) +x(−n)N] /2.

(b) obtain its odd part and show that its DFT of [xo(n)] = j Im [X(k)].

AIM: To show that the DFT of [xe(n)] = Re[X(k)] , and [xo(n)] = j Im [X(k)]

where the even part xe(n) = [x(n) +x(−n)N] /2 and xo(n) = [x(n) - x(−n)N] /2.

Short Theory:

Every function can be decomposed into a sum of its even part fe(n) and odd part fo(n), where

$$f_e(n) \triangleq \frac{f(n)+f(-n)}{2}$$
$$f_o(n) \triangleq \frac{f(n)-f(-n)}{2}.$$

Proof: In the above definitions, fe(n) is even and fo(n) is odd by construction. Summing, we have

$$f_e(n)+f_o(n) = \frac{f(n)+f(-n)}{2} + \frac{f(n)-f(-n)}{2} = f(n).$$

Key Commands:

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

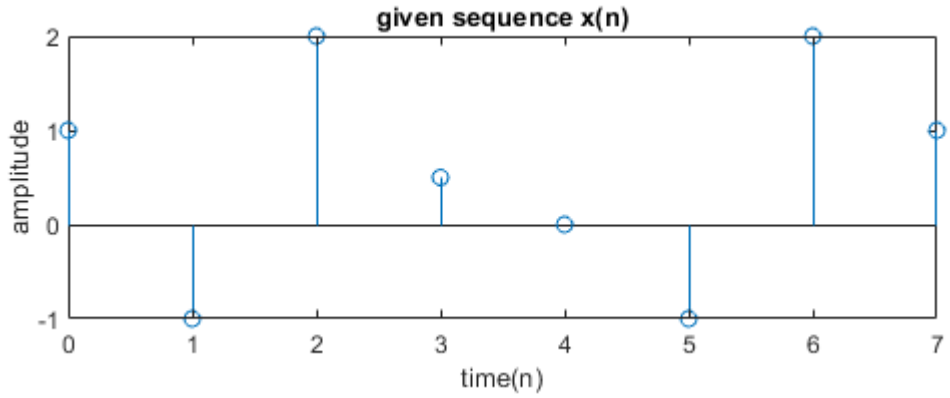
abs(x) % returns the absolute value of each element in array X.

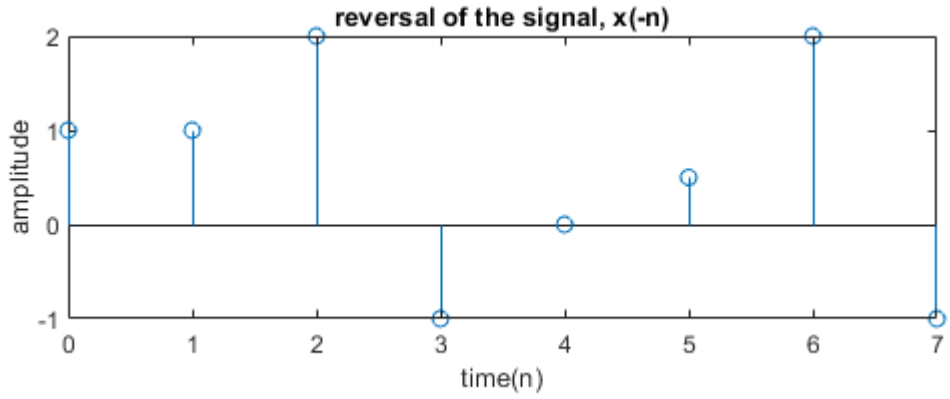
real % real(Z) returns the real part of each element in array Z.

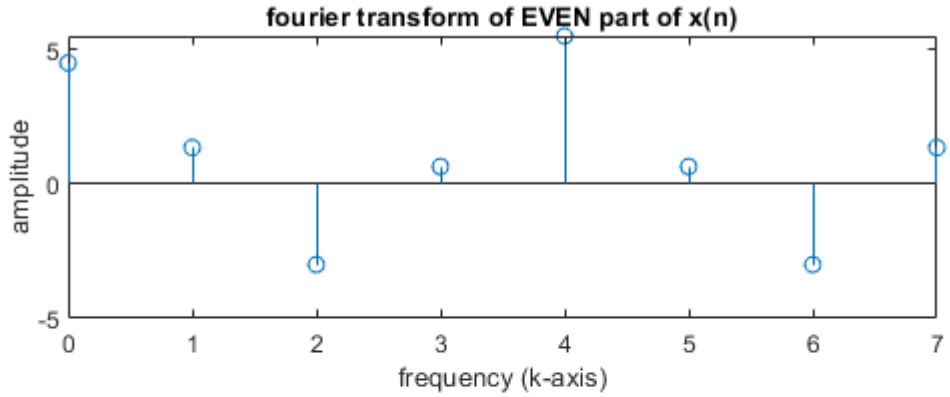
imagg %imag(Z) returns the imaginary part of each element in array Z.

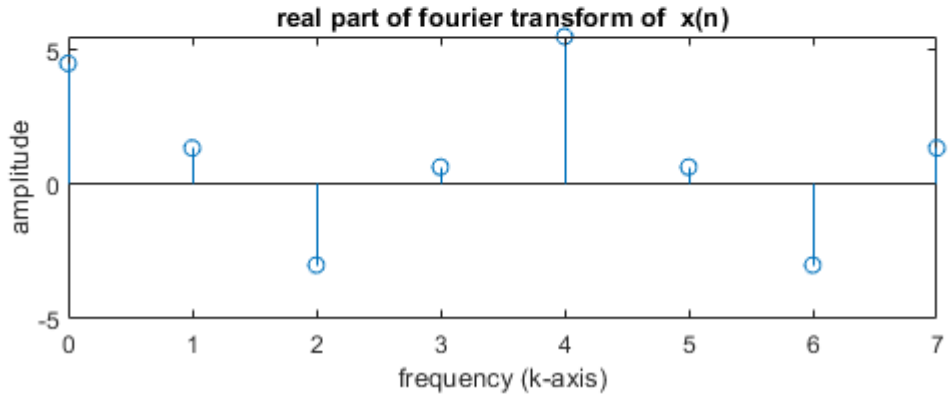
mod() %mod(a,m) returns the remainder after division of a by m, where a is the dividend and m is the divisor

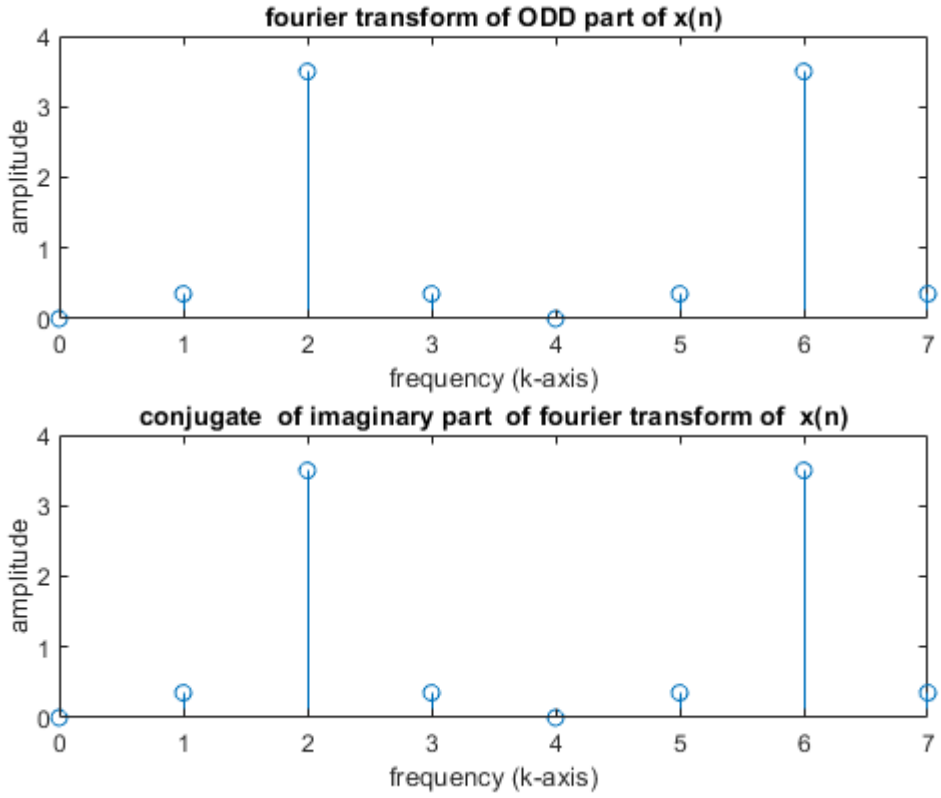
Plots:


given sequence x(n)
amplitude
time(n)


reversal of the signal, x(-n)
amplitude
time(n)


fourier transform of EVEN part of x(n)
amplitude
frequency (k-axis)


real part of fourier transform of x(n)
amplitude
frequency (k-axis)



Inferences/comments:

1) The DFT of Even part of the x(n) is equal to the real part of the Fourier transform of the x(n).

DFT of [xe(n)] = Re[X(k)], where the even part xe(n) = [x(n) +x(−n)N] /2.

2) The DFT of Odd part of the x(n) is equal to the j * the imaginary part of the Fourier transform of the x(n).

DFT of [xo(n)] = j*imag[X(k)], where the even part xo(n) = [x(n) -x(−n)N] /2.

Q7. Using MATLAB verify the effect of i) circular time shift ii) circular frequency shift on the sequence x(n) of problem 6.

AIM: To verify the effect of circular time shift and circular frequency shift properties for the x(n)= {1, −1, 2, 0.5, 0, −1, 2, 1}.

Short Theory:

The Circular Time shift states that if

$$X(n) \xleftrightarrow[N]{DFT} x(k) \text{ And}$$

$$x((n-l))_N \xleftrightarrow[N]{DFT} x(k)\, e^{-j2\pi k l / N}$$

Thus shifting the sequence circularly by „l samples is equivalent to multiplying its DFT by $e^{-j2\pi k1/N}$

The Circular frequency shift states that if

$$X(n) \xleftrightarrow[N]{DFT} x(k) \text{ then}$$

$$x^*(n) \xleftrightarrow[N]{DFT} x^*((-k))_N = x^*(N-k) \text{ And}$$

$$x^*((-n))_N = x^*(N-k) \xleftrightarrow[N]{DFT} x^*(k)$$

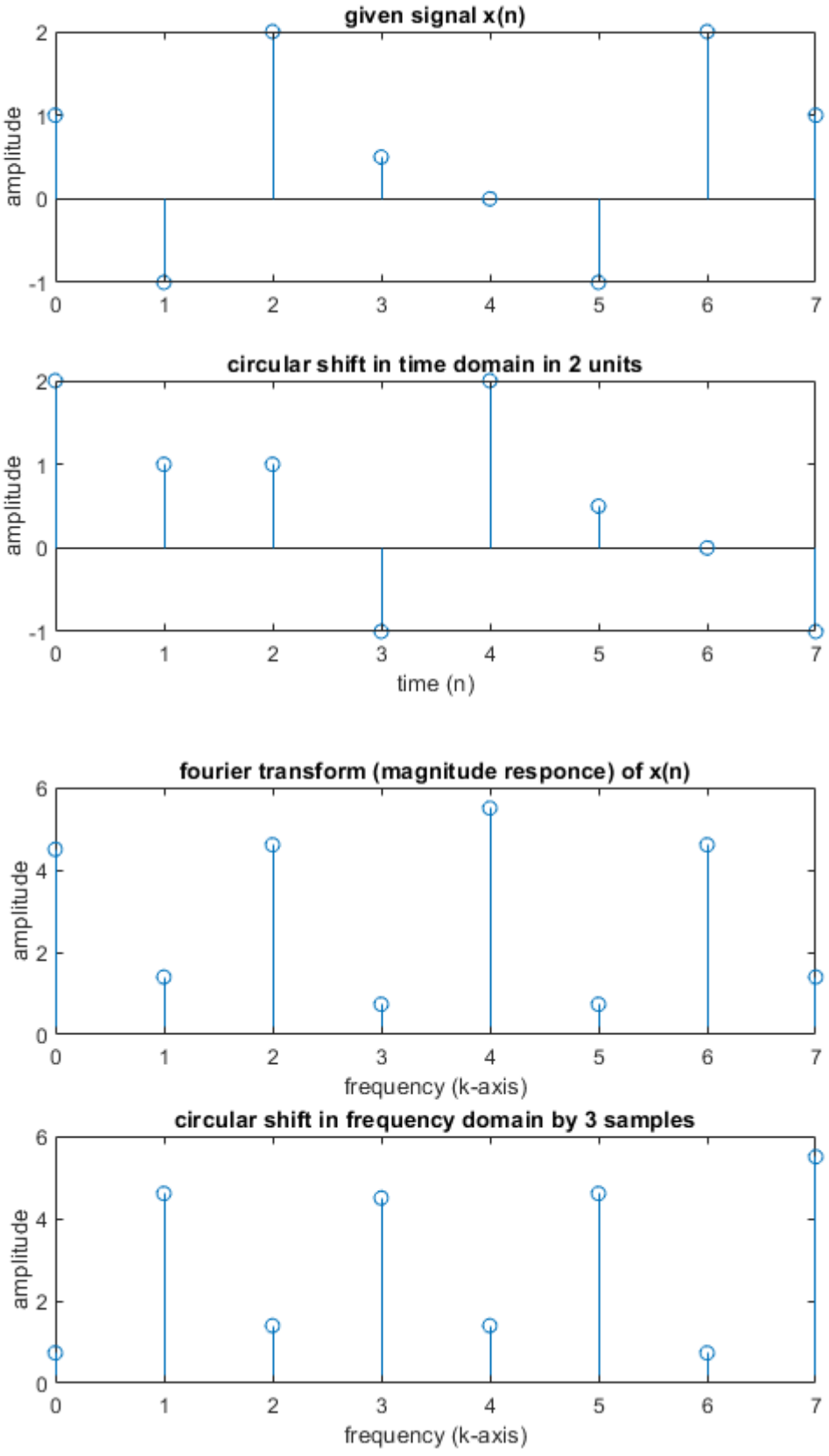
Key Commands:

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

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

abs(x) % returns the absolute value of each element in array X.

Plots:



Inferences/comments:

1) Magnitude response of circular time shift by 2 units is not equal to magnitude response of circular frequency shift by same units.

Experimental Exercises:

Q1. Decode the mobile number from DTMF encoded tone by using manual segmentation and Fourier transform. You will be provided with the *.wav file. use the DTMF table provided below.

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

AIM: Decode the the mobile number from DTMF encoded tone by using manual segmentation and Fourier transform for given audio signals provided in the **URL**

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.

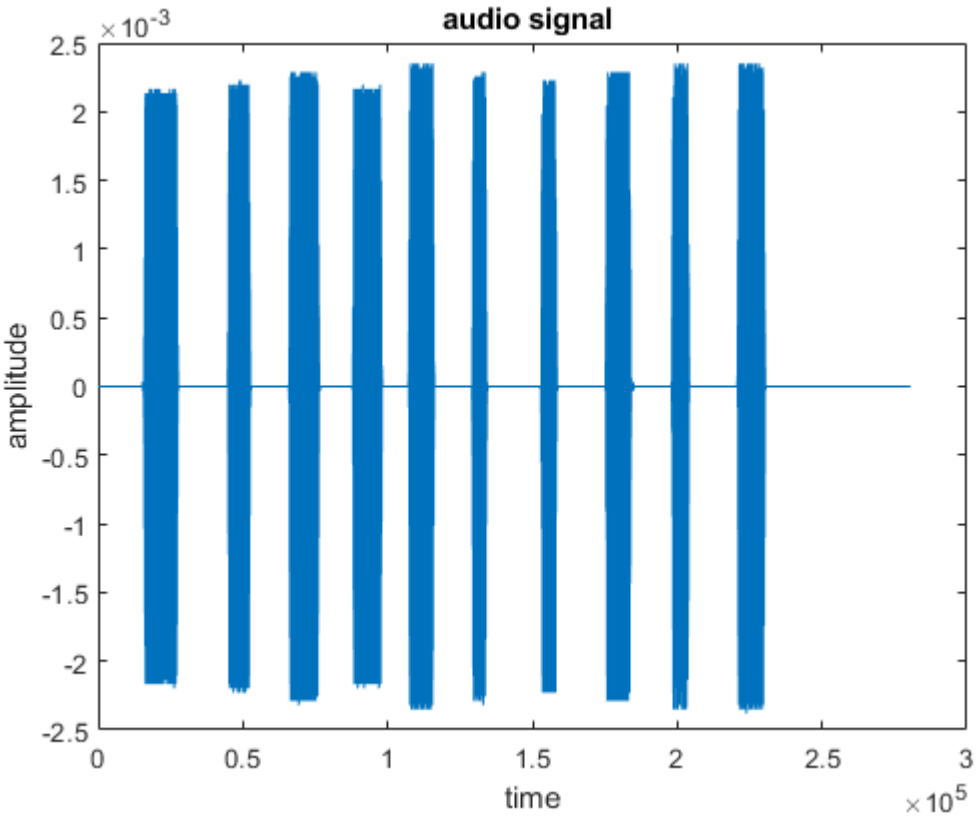
Key Commands:

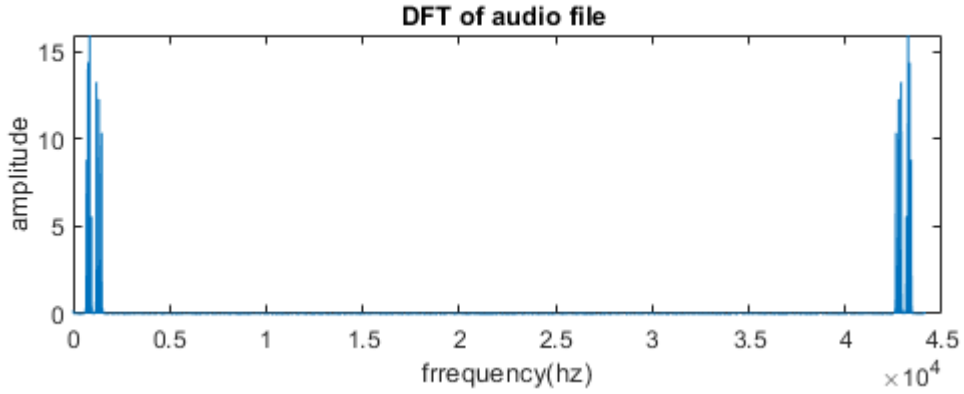
fft(x) %computes the discrete Fourier transform (DFT) of X using a fast Fourier transform (FFT) algorithm.

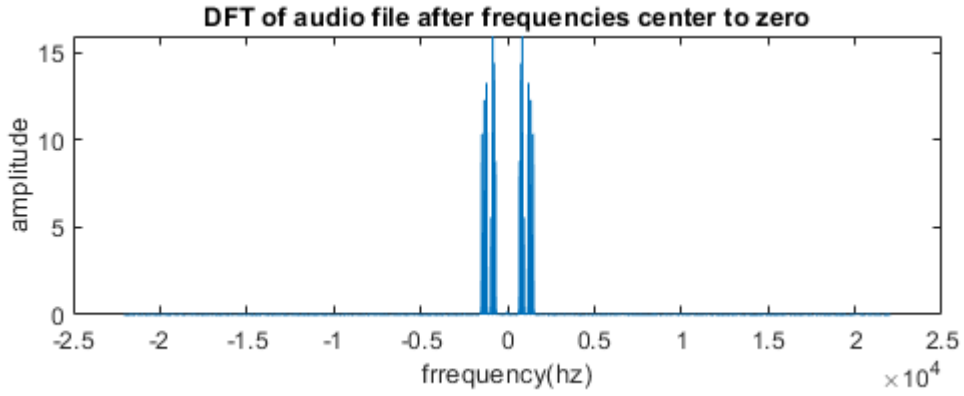
abs(x) % returns the absolute value of each element in array X.

Plots:

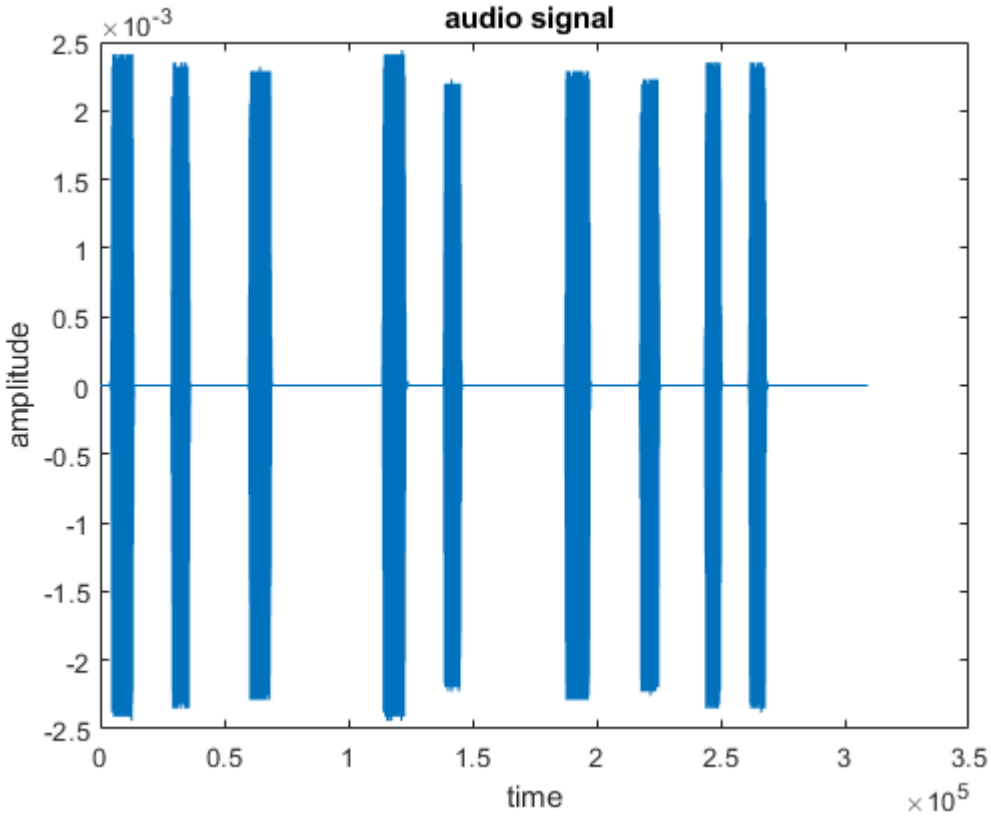
For audio signal - exp1.wav

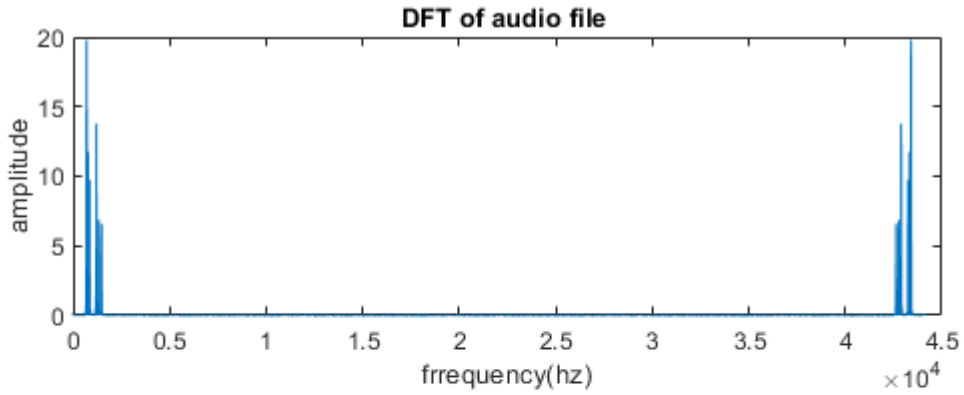


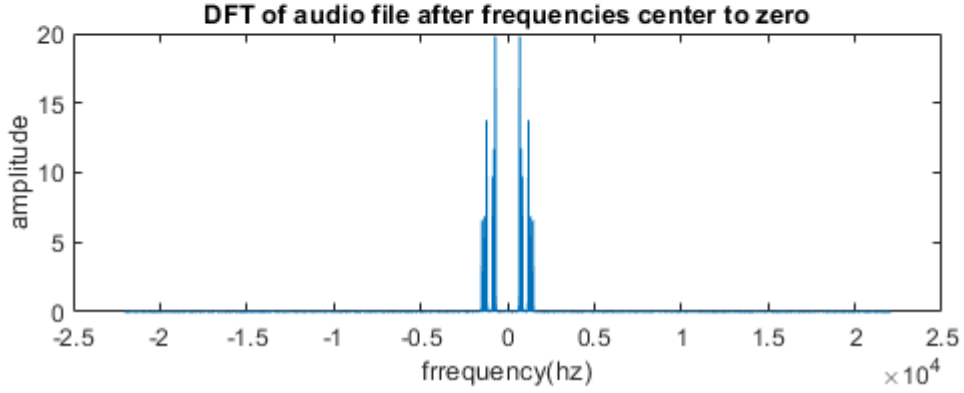




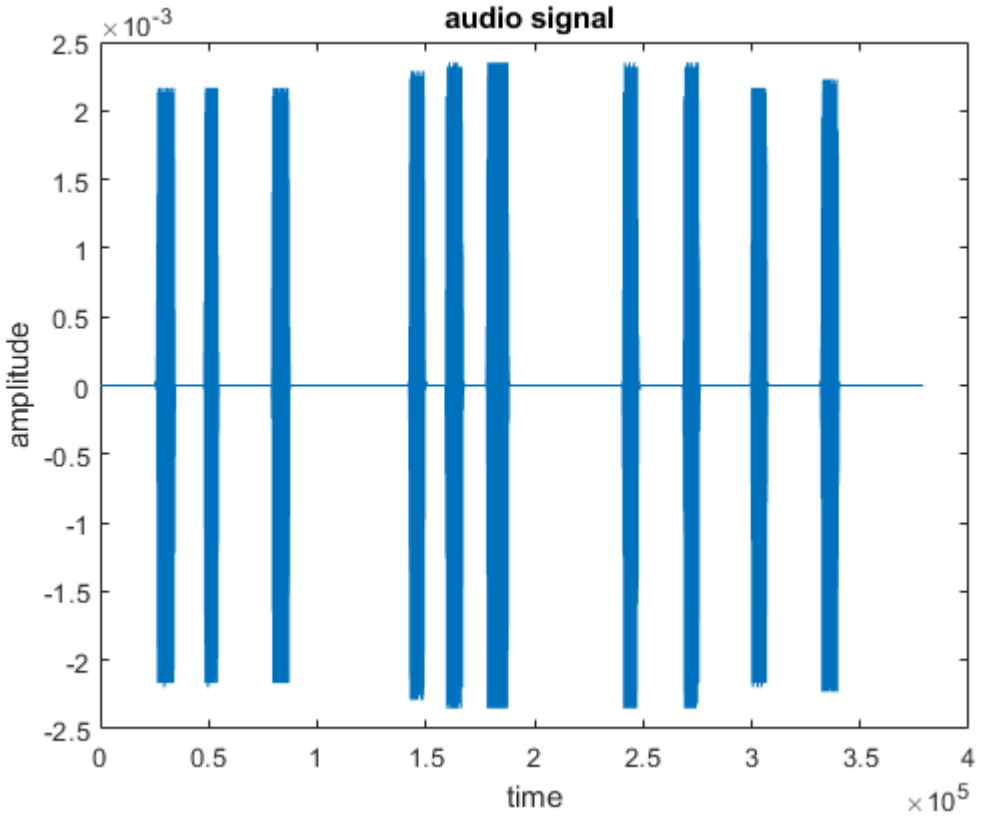
exp2.wav

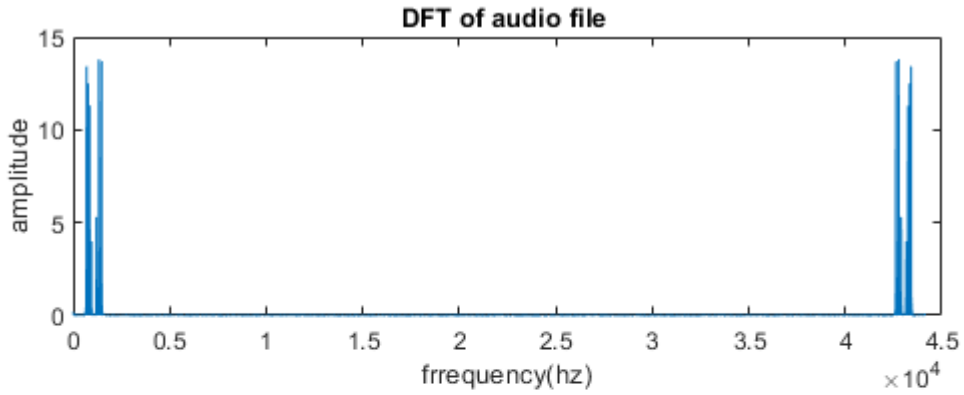

audio signal
×10^-3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
amplitude
0
0.5
1
1.5
2
2.5
3
3.5
time
×10^5

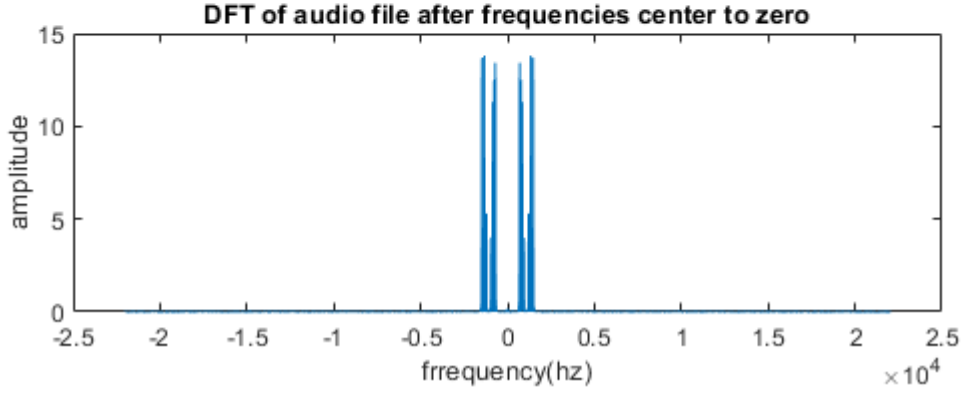

DFT of audio file
20
15
10
5
0
amplitude
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
frrequency(hz)
×10^4


DFT of audio file after frequencies center to zero
20
15
10
5
0
amplitude
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
frrequency(hz)
×10^4

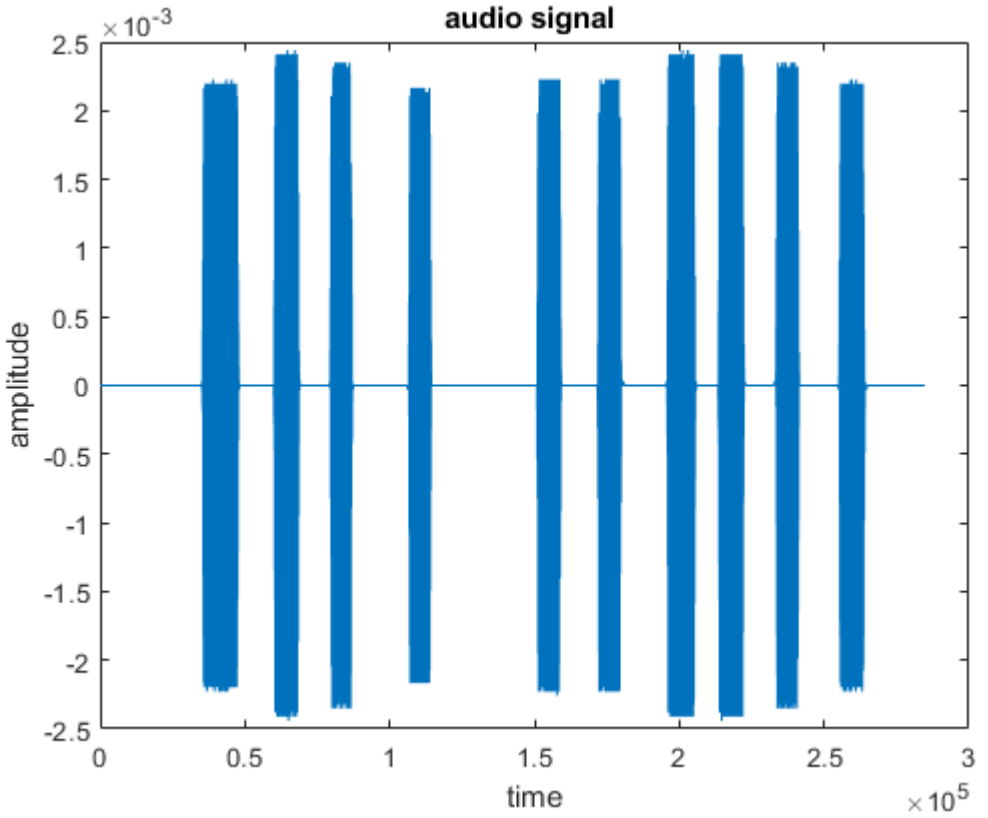
exp3.wav

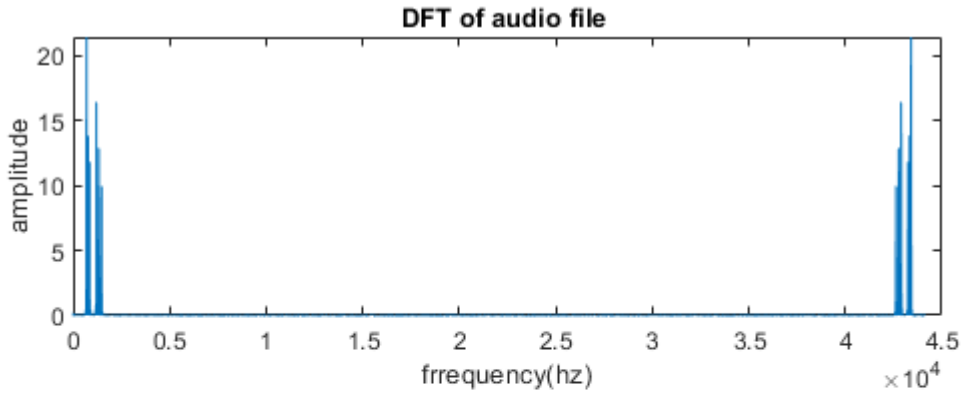

audio signal
×10^-3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
amplitude
0
0.5
1
1.5
2
2.5
3
3.5
4
time
×10^5

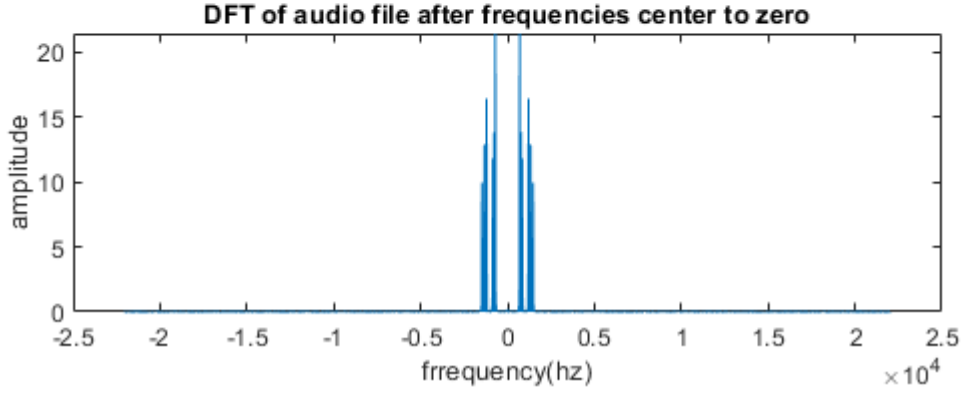

DFT of audio file
15
10
5
0
amplitude
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
frrequency(hz)
×10^4


DFT of audio file after frequencies center to zero
15
10
5
0
amplitude
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
frrequency(hz)
×10^4

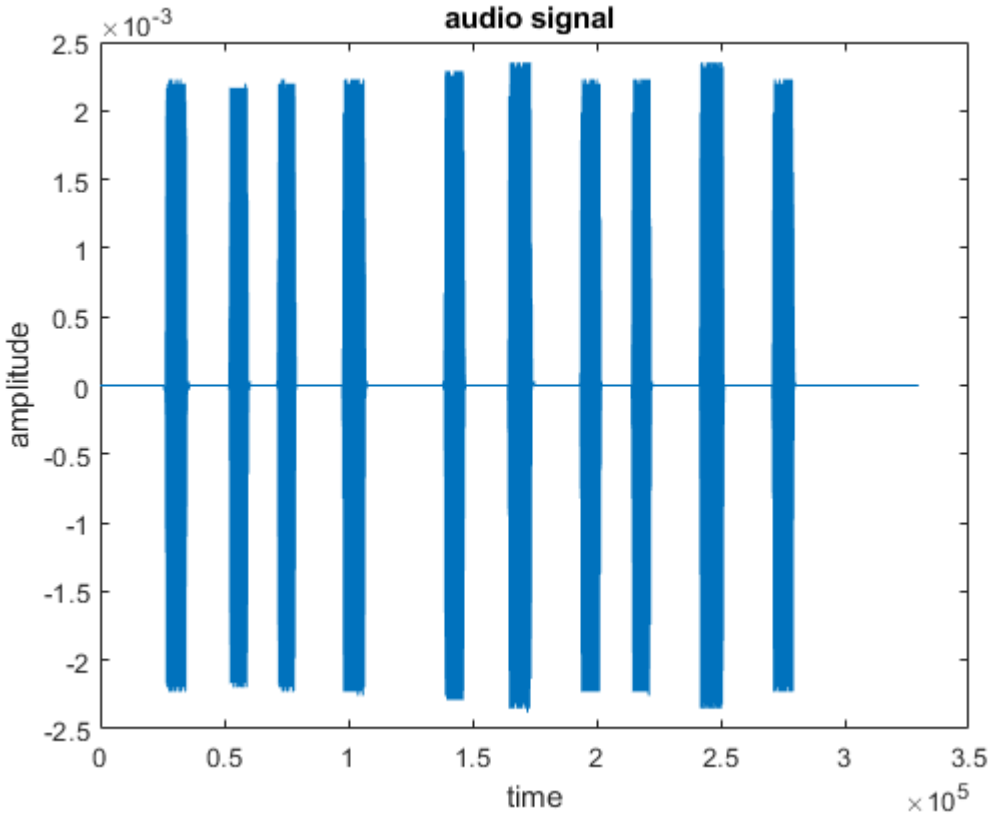
exp4.wav

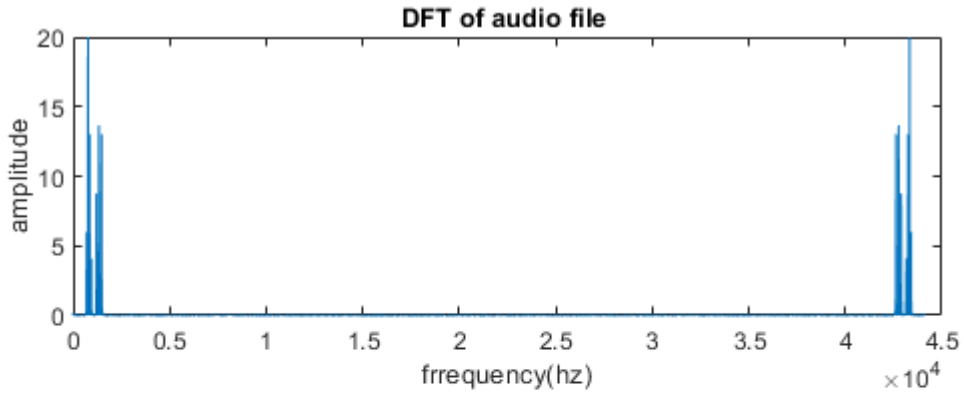

audio signal
×10^-3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
amplitude
0
0.5
1
1.5
2
2.5
3
time
×10^5

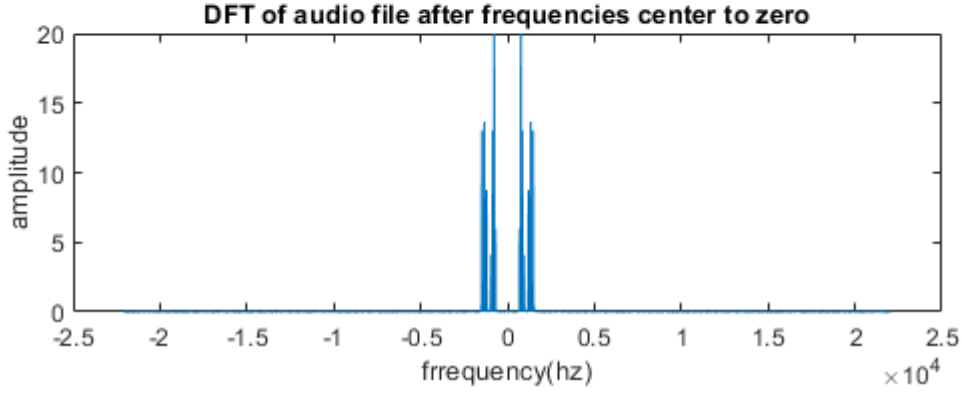

DFT of audio file
20
15
10
5
0
amplitude
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
frrequency(hz)
×10^4


DFT of audio file after frequencies center to zero
20
15
10
5
0
amplitude
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
frrequency(hz)
×10^4

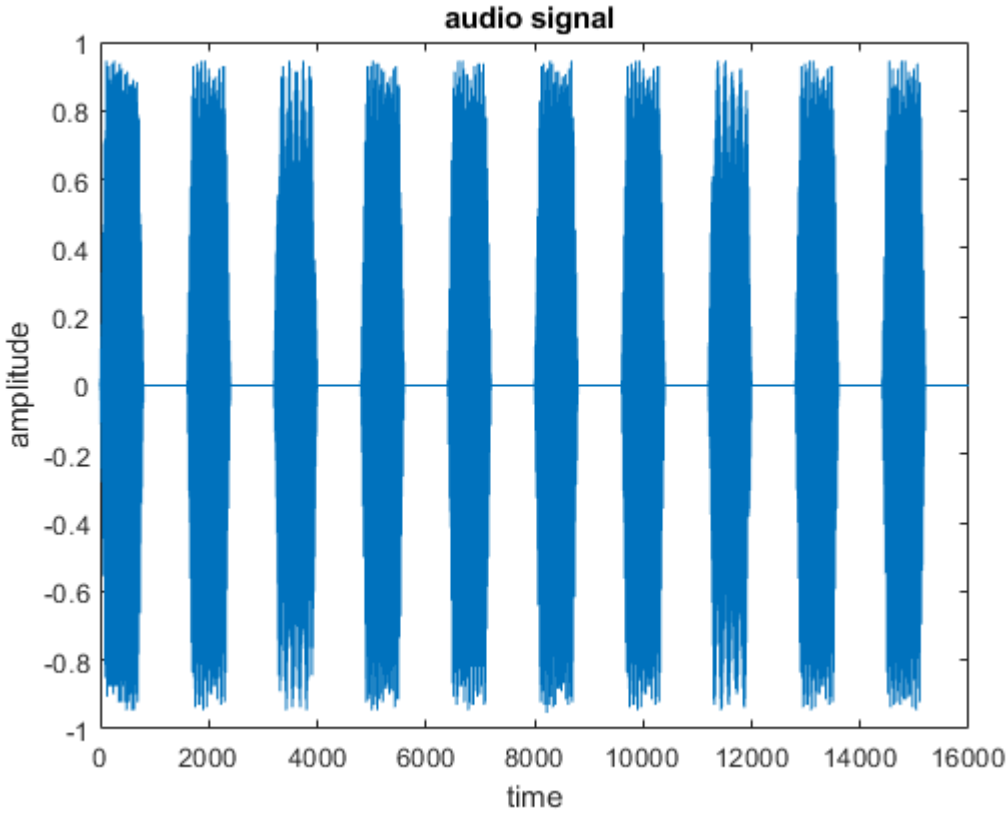
exp5.wav

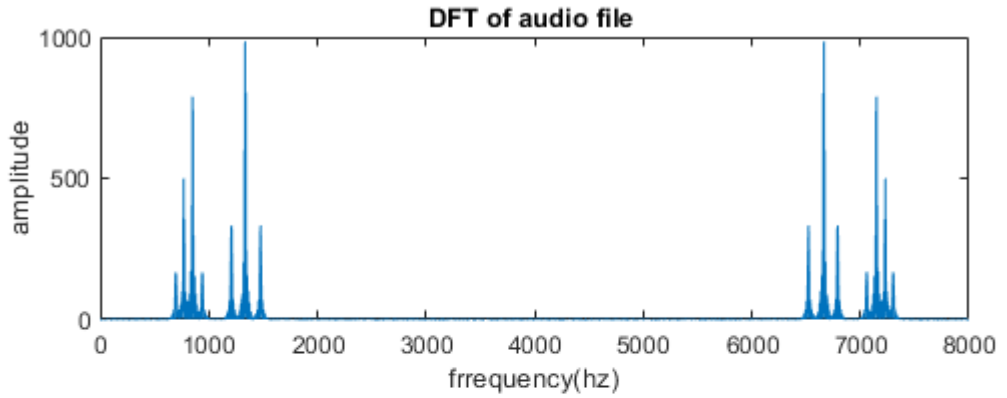

audio signal
×10^-3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
amplitude
0
0.5
1
1.5
2
2.5
3
3.5
time
×10^5

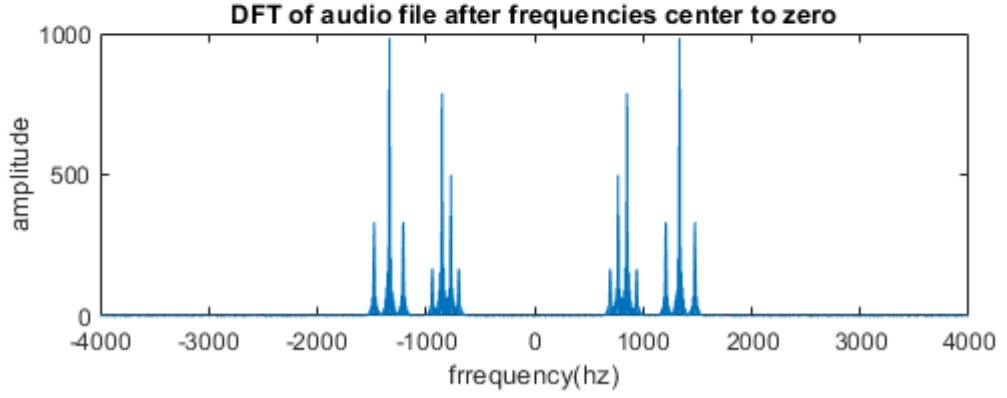

DFT of audio file
20
15
10
5
0
amplitude
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
frrequency(hz)
×10^4


DFT of audio file after frequencies center to zero
20
15
10
5
0
amplitude
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
frrequency(hz)
×10^4

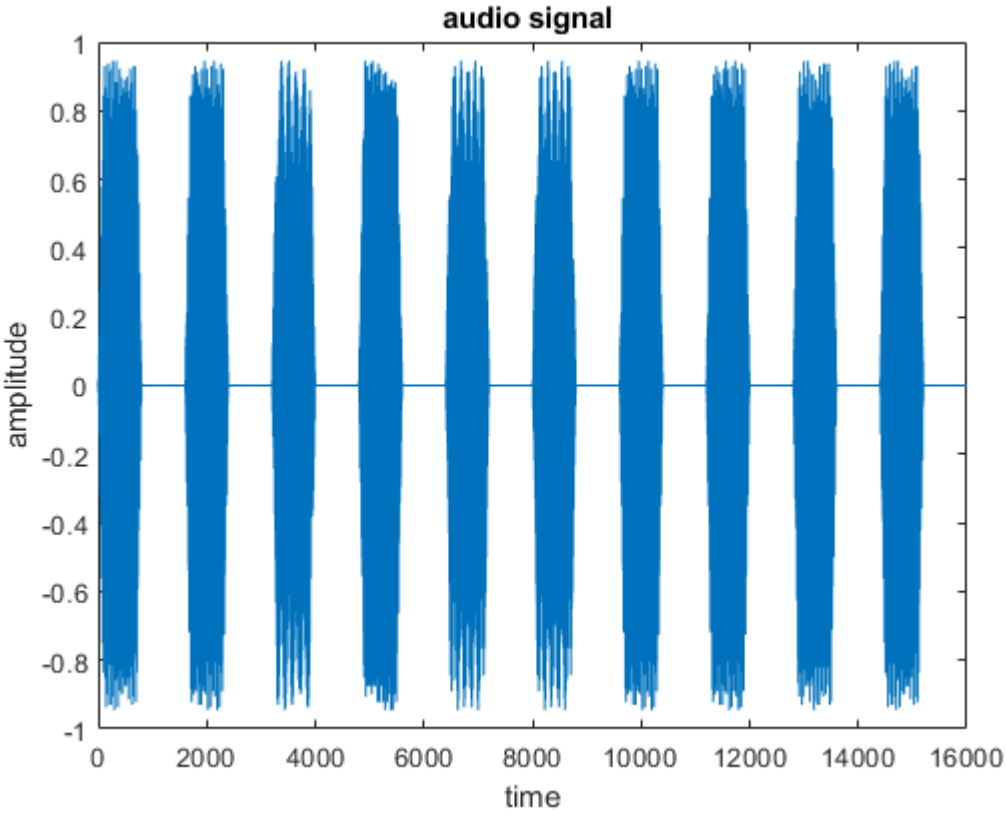
file1.wav

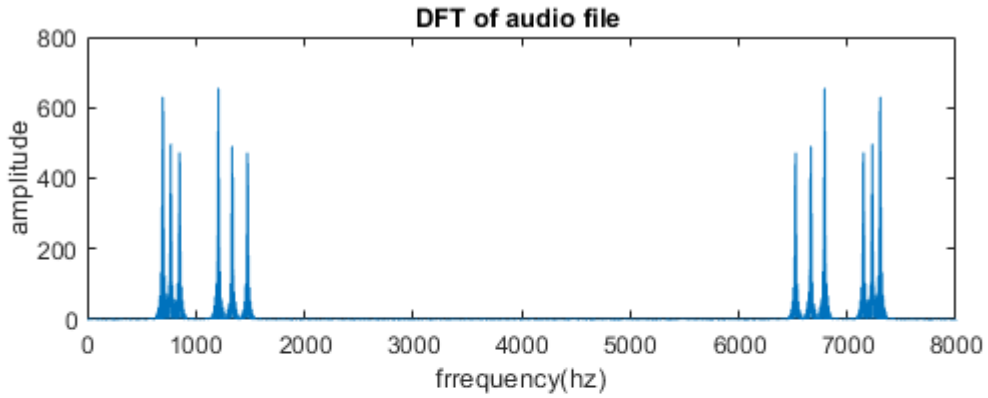

audio signal
amplitude
time
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
2000
4000
6000
8000
10000
12000
14000
16000

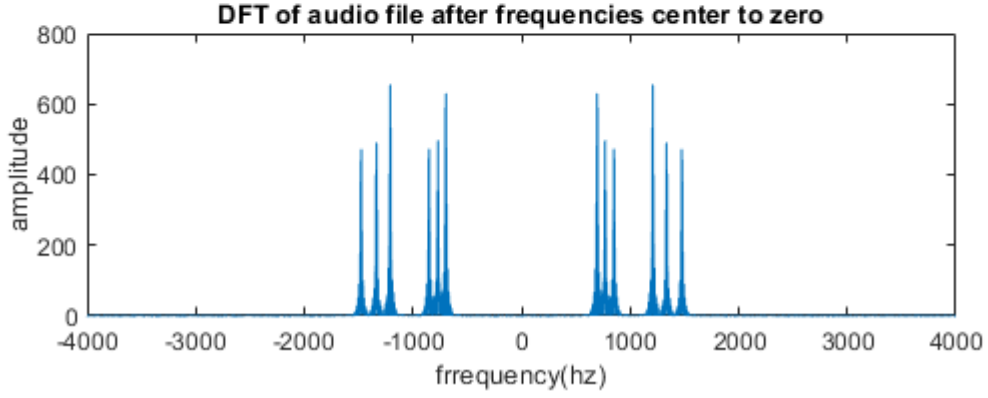

DFT of audio file
amplitude
frrequency(hz)
1000
500
0
0
1000
2000
3000
4000
5000
6000
7000
8000


DFT of audio file after frequencies center to zero
amplitude
frrequency(hz)
1000
500
0
-4000
-3000
-2000
-1000
0
1000
2000
3000
4000

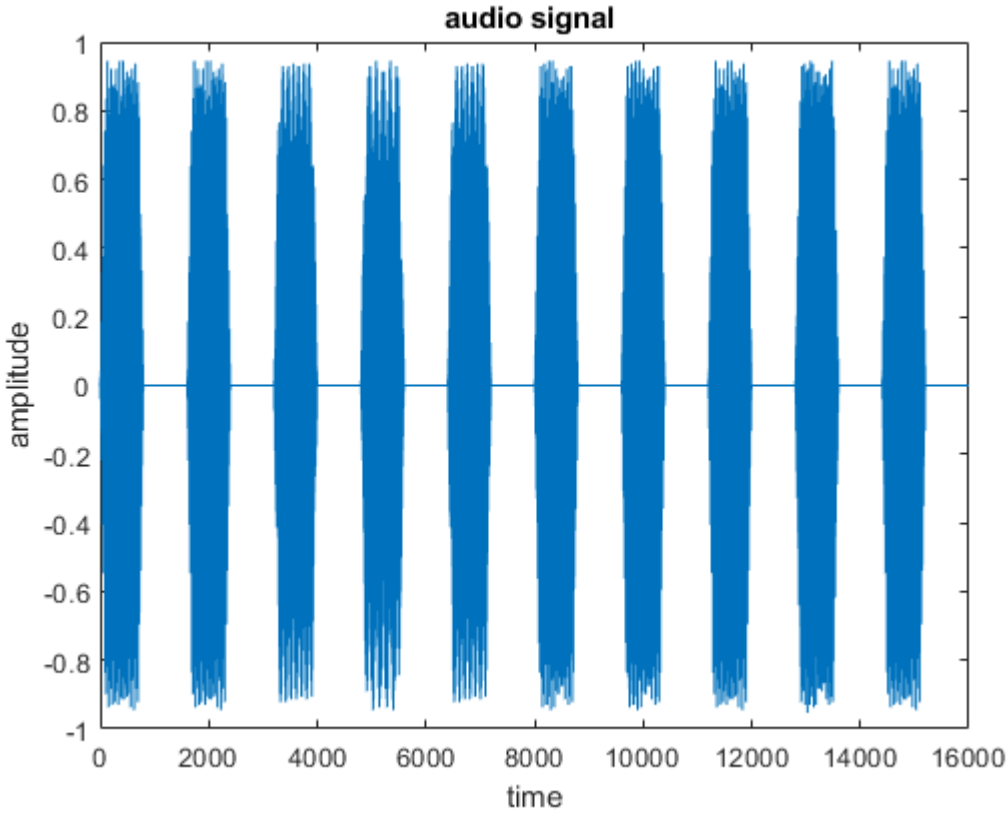
file2.wav

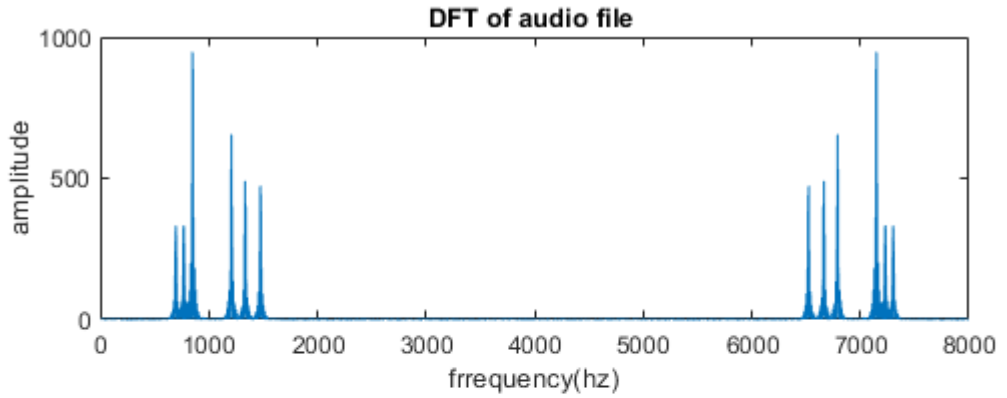

audio signal
amplitude
time
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
2000
4000
6000
8000
10000
12000
14000
16000

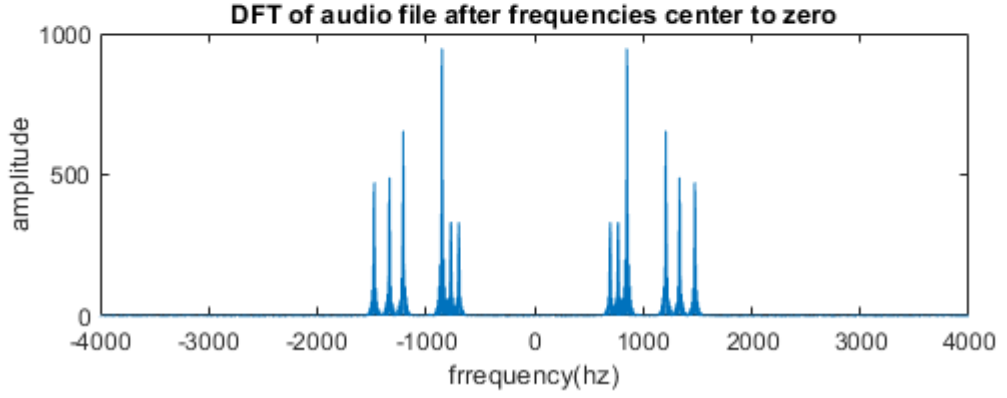

DFT of audio file
amplitude
frrequency(hz)
800
600
400
200
0
0
1000
2000
3000
4000
5000
6000
7000
8000


DFT of audio file after frequencies center to zero
amplitude
frrequency(hz)
800
600
400
200
0
-4000
-3000
-2000
-1000
0
1000
2000
3000
4000

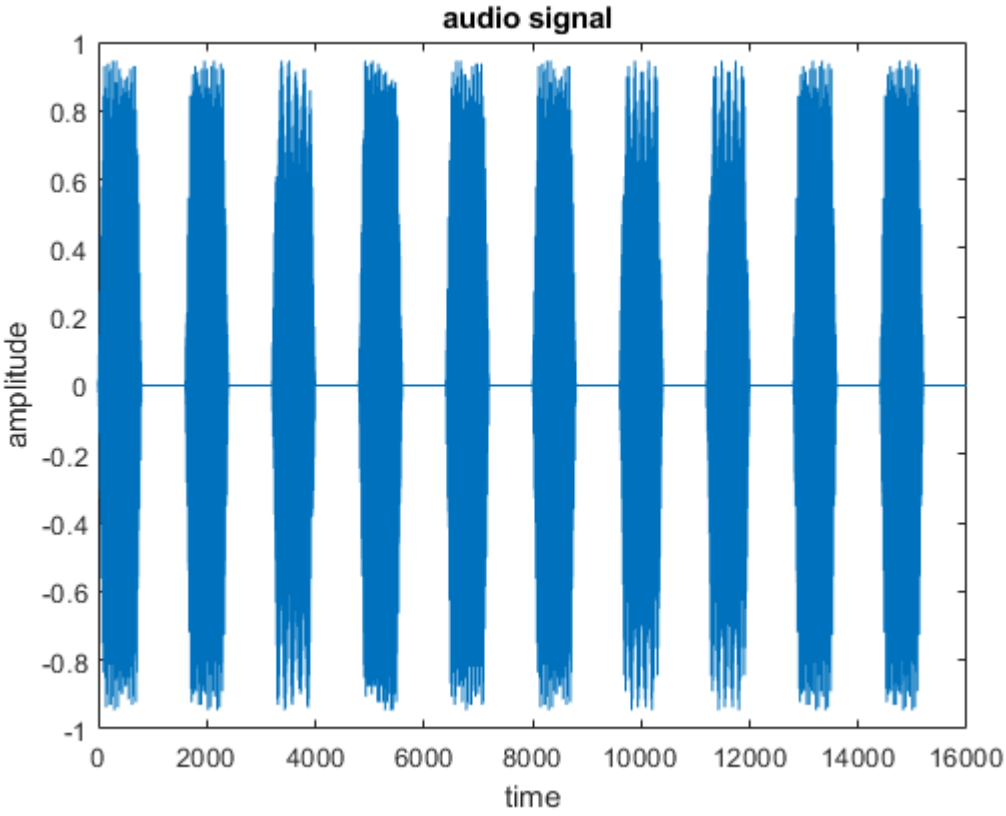
file3.wav

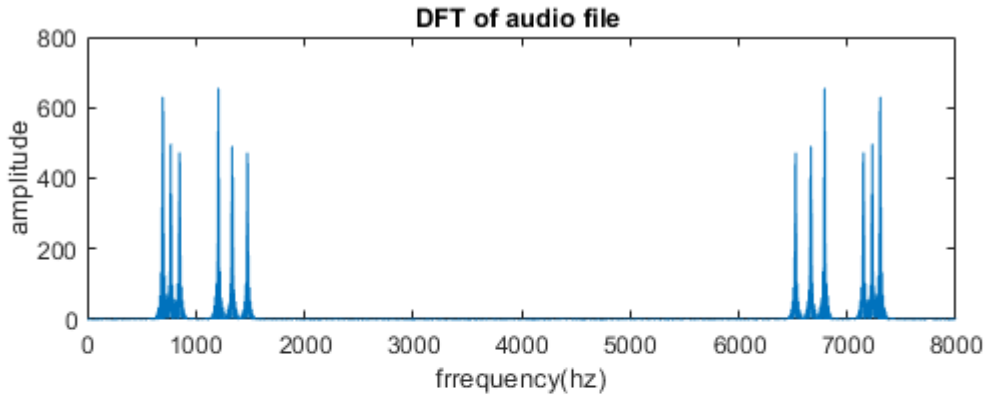

audio signal
amplitude
time
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
2000
4000
6000
8000
10000
12000
14000
16000

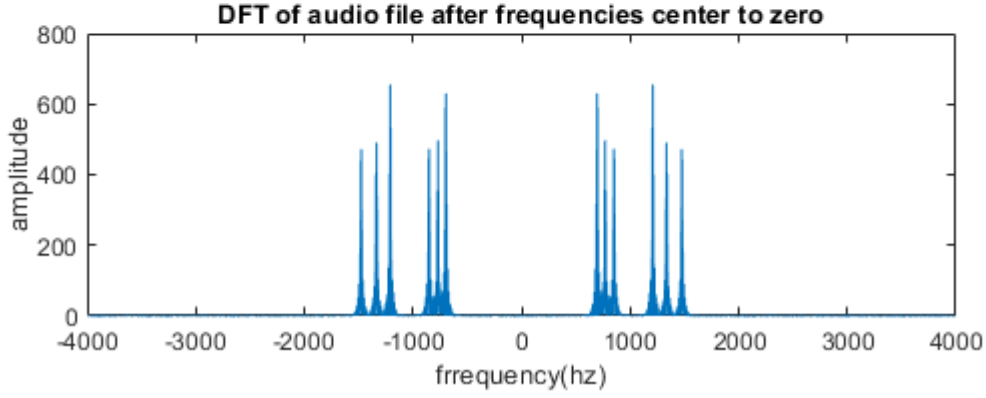

DFT of audio file
amplitude
frrequency(hz)
1000
500
0
0
1000
2000
3000
4000
5000
6000
7000
8000


DFT of audio file after frequencies center to zero
amplitude
frrequency(hz)
1000
500
0
-4000
-3000
-2000
-1000
0
1000
2000
3000
4000

file4.wav


audio signal
amplitude
time
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
2000
4000
6000
8000
10000
12000
14000
16000


DFT of audio file
amplitude
frrequency(hz)
800
600
400
200
0
0
1000
2000
3000
4000
5000
6000
7000
8000


DFT of audio file after frequencies center to zero
amplitude
frrequency(hz)
800
600
400
200
0
-4000
-3000
-2000
-1000
0
1000
2000
3000
4000

Inferences/comments:

1).

Audio signal name	All Frequencies present in Hz	Corresponding DTMF Number
exp1.wav	**699.12, 771.71, 852.8, 940.7 1208.06, 1337.26, 1472.59**	**9870456324**
exp2.wav	**700.32, 772.05, 853.05, 1207.14, 1336.63, 1472.67**	**123187644**
exp3.wav	**699.07, 771.82, 853.64, 940.36 1207.37, 1336.92, 1474.5**	**9995242206**
exp4.wav	**699.75, 772.4, 851.7, 1207.25, 1337.07, 1473.25**	**8129661148**
exp5.wav	**699.6, 772.52, 852.6, 941.75 1206.8, 1337.2, 1474.56**	**8086726646**
file1.wav	**699.9, 770, 850, 941.5, 1209, 1335, 1477.5**	**9860458287**
file2.wav	**695, 770, 850 1210, 1335, 1475**	**8129661148**
file3.wav	**695, 770, 850 1210, 1335, 1475**	**7736388751**
file4.wav	**695, 770, 850, 1210, 1335, 1475**	**8129486611**

2) For the audio file " exp2.wav" has only 9 digits in it and the remaining all the audio files has 10 digits DTMF signals.