Experiment Name:

Consider a continuous time Signal , x(t)=sin(2\*pi\*1000\*t)+1/2\*sin(2 \*pi\*2000\*t+3\*pi/4),

Determine sampled signal x(n\*Ts)=x(n) , using The Matlab taking n=8 samples at sampling rate,

Fs=8000Hz (sample/second).The Sample signal , x(n)=x(n\*Ts)=sin(2\*pi\*1000\*n\*Ts)+1/2\*sin(2\*pi\*2000\*n\*Ts+3\*pi/4)  
where Ts=1/Fs=1/8000 sec is in the sampling period.

**Fourier transform:**

A Fourier transform converts a signal in the time domain to the frequency domain(spectrum). An inverse Fourier transform converts the frequency domain components back into the original time domain signal.

Continuous‐Time Fourier Transform:

**Discrete‐Time Fourier Transfor(DTFT):**

**Discrete Fourier Transform(DFT):**

Using the Fourier series representation we have Discrete Fourier Transform(DFT) Transform(DFT) for finite length signal • DFT can convert time‐domain discrete signal into frequency‐ domain discrete spectrum.

Assume that we have a signal .

Then the DFT of the signal is a sequence for X(k) for k=0,1,2,3, …

The Inverse Discrete Fourier Transform(IDFT):

,n=0,2,..,N-1;

**Fast Fourier Transform(FFT):**

The Fast Fourier Transform does not refer to a new or different type of Fourier transform. It refers to a very efficient algorithm for computing the DFT .The time taken to evaluate a DFT on a computer depends principally on the number of multiplications involved. DFT needs N2 multiplications. FFT only needs Nlog2(N) . The central insight which leads to this algorithm is the realization that a discrete Fourier transform of a sequence of N points can be written in terms of two discrete Fourier transforms of length N/2 .Thus if N is a power of two, it is possible to recursively apply this decomposition until we are left with discrete Fourier transforms of single points.

**Experiment No: 01**

MATLAB Code:

x=[1 0 0 1];

y=fft(x);

subplot(2,1,1)

stem(abs(y),'k')

xlabel('m')

ylabel('X(m)')

title('Absolute Value Of DFT Sequence')

subplot(2,1,2)

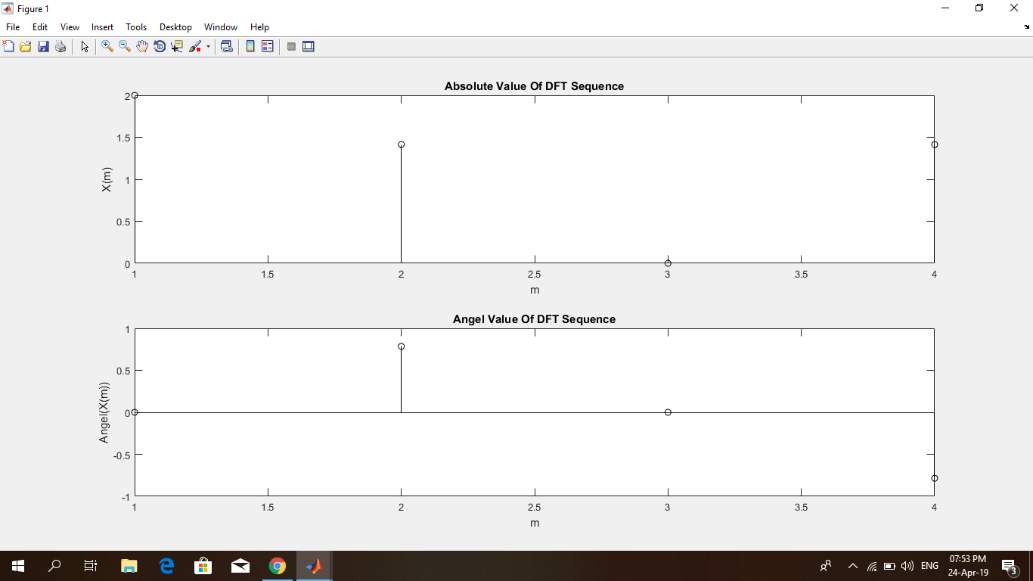
stem(angle(y),'k')

xlabel('m')

ylabel('Angel(X(m))')

title('Angel Value Of DFT Sequence')

Figure:



**Experiment No:02**

**Rectangular Pulse:**

**MATLAB Code:**

**fs=500;**

**t=-1:1/fs:1;**

**x=rectpuls(t,0.12)**

**subplot(2,2,1)**

**plot(t,x)**

**grid on**

**y=fft(x);**

**y=fftshift(y);**

**subplot(2,2,2)**

**plot(abs(y))**

**grid on**

**x=rectpuls(t,0.02);**

**subplot(2,2,3)**

**plot(t,x)**

**grid on**

**y=fft(x);**

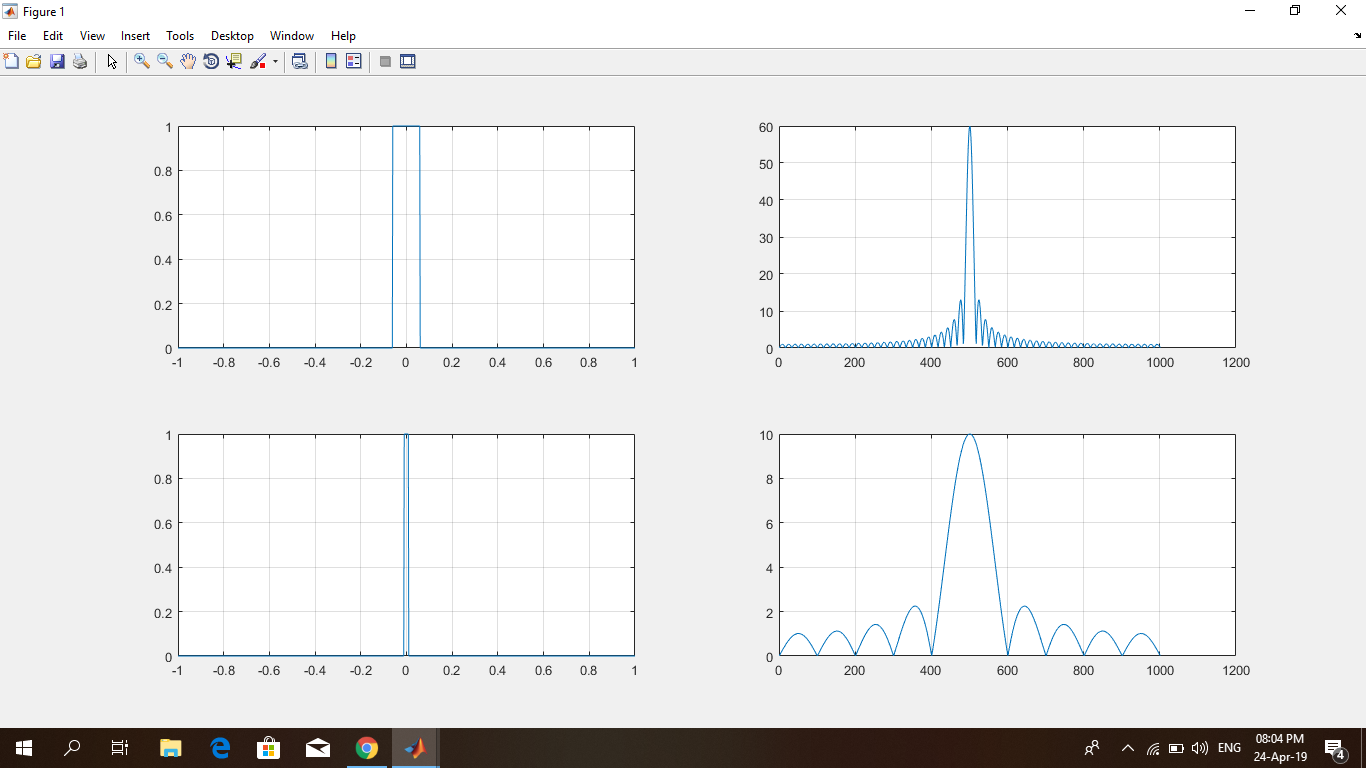
**y=fftshift(y);**

**subplot(2,2,4)**

**plot(abs(y))**

**grid on**

**Figure:**

****

**Experiment No:03**

**DTFT On a 2D Rectangular Pulse:**

**MATLAB Code:**

**x=zeros(32);**

**x(12:17)=ones(6,1);**

**subplot(2,2,1)**

**title('2D Rectangular Pulse')**

**mesh(x)**

**x=fft(x);**

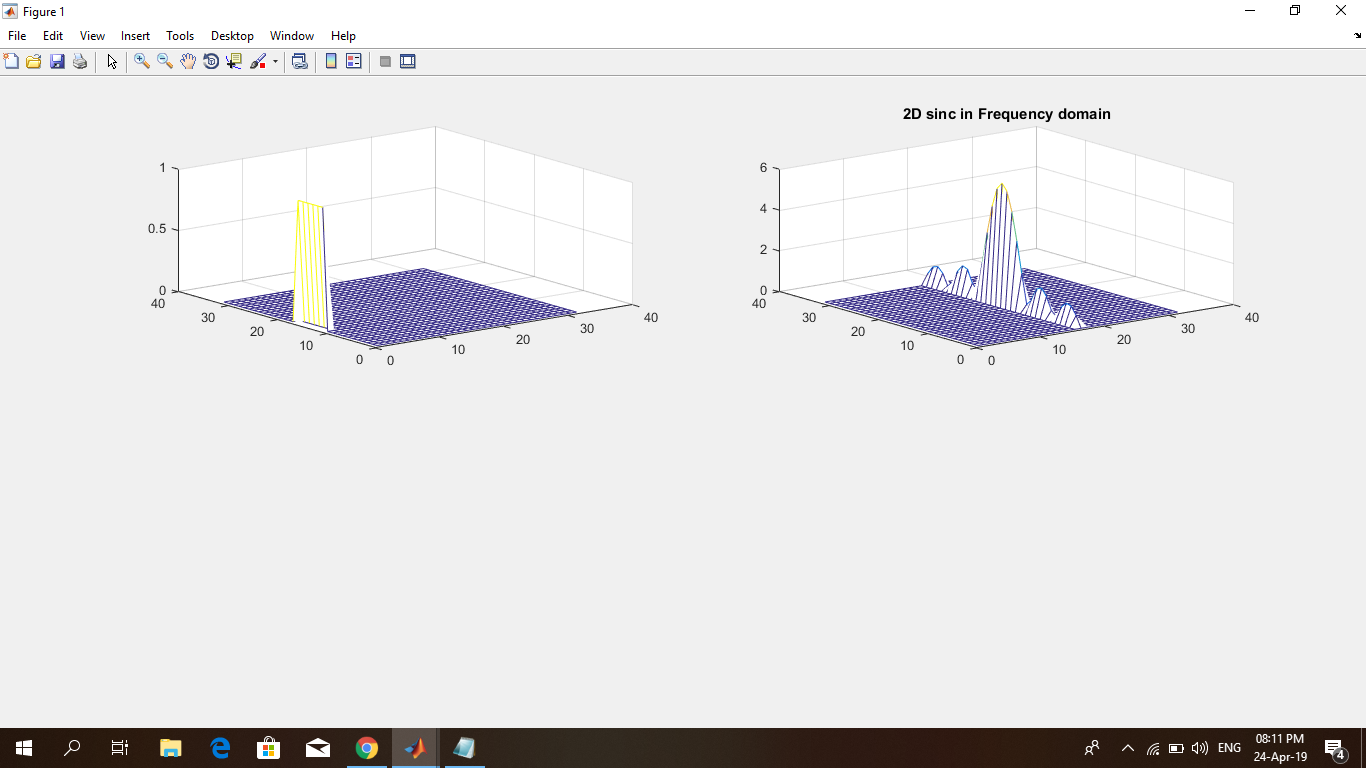
**x=fftshift(x);**

**subplot(2,2,2)**

**mesh(abs(x))**

**title('2D sinc in Frequency domain')**

**Figure:**

****

**Experiment No:04**

**DTFT On a 3D Rectangular Pulse:**

**MATLAB Code:**

**x=zeros(32);**

**x(12:17,12:17)=ones(6);**

**subplot(2,2,3)**

**title('2D Rectangular Pulse')**

**mesh(x)**

**x=fft(x);**

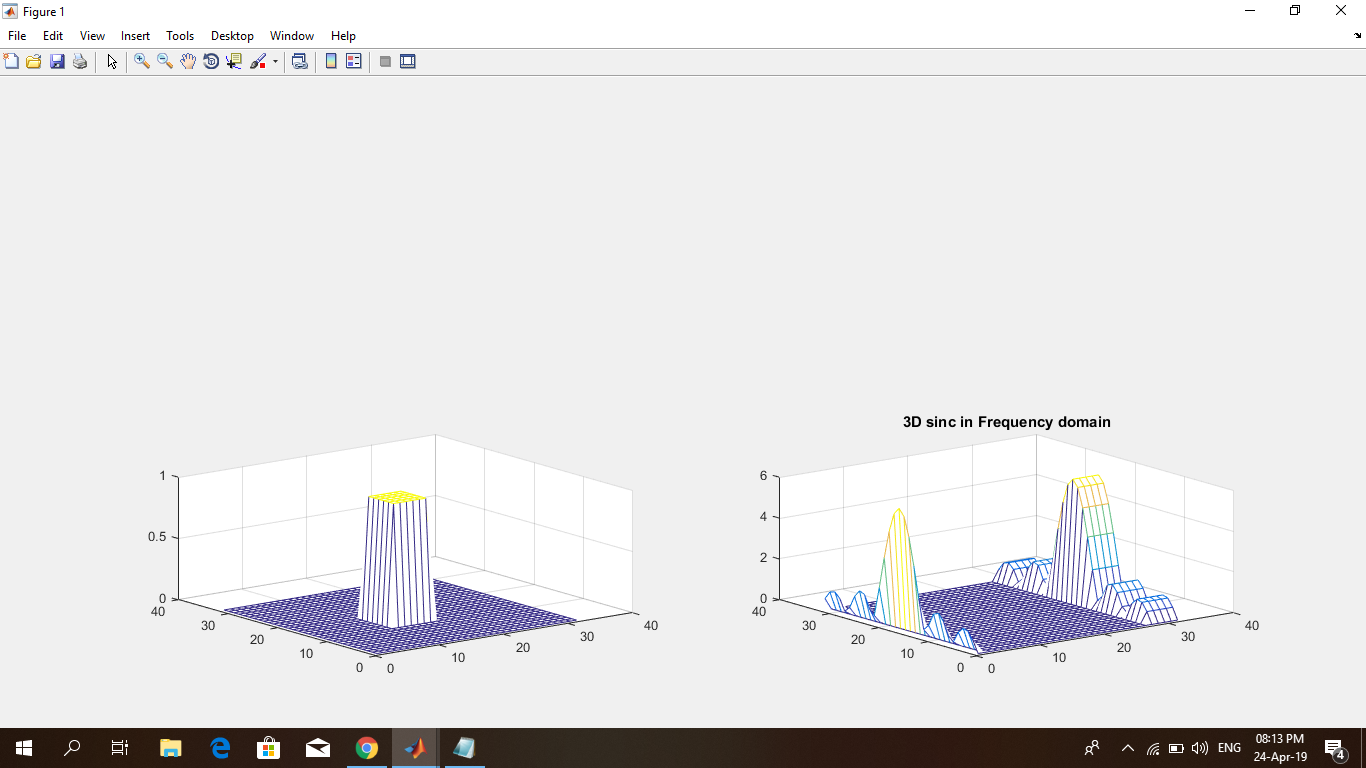
**x=fftshift(x);**

**subplot(2,2,4)**

**mesh(abs(x))**

**title('3D sinc in Frequency domain')**

**Figure:**

****

**Discussion:**

As we can see everything was completed successfully we can say that the result of the experiment is not bad.