**NAME: PRIYANSHU RAJ**

**REG. NO: 22BEC1492**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No.** | **Date** | | **Title** |
| **1.** | **09-01-24** | | **Generation of signals** |
| **2.** | **23-01-24** | | **Convolution** |
| **3.** | **30-01-24** | | **Discrete Fourier Transform** |
| **4.** | **06-02-23** | | **Circular Convolution using FFT** |
| **5.** | **20-02-24** | | **IIR Filters** |
| **6.** | **27-02-24** | | **FIR Filters** |
| **7.** | **05-03-24** | | **Bandpass Filters** |
| **8.** | **16-04-24** | | **Sampling Rate Conversion** |
| **9.** | | **26-03-24** | **Hardware** |

**EXPERIMENT NO: 1**

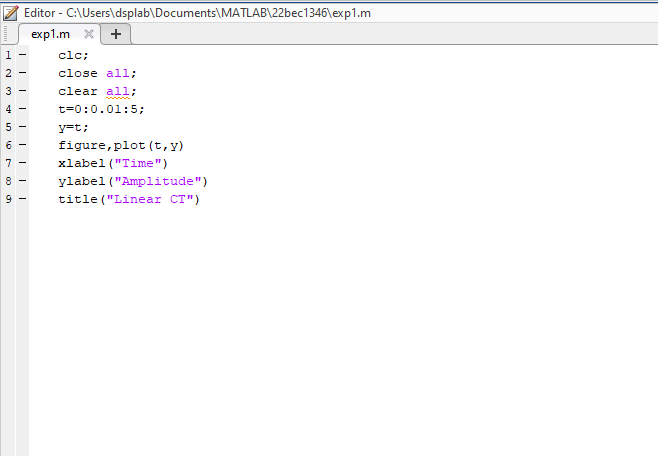
**DATE: 09-01-2024**

**GENERATION OF SIGNALS**

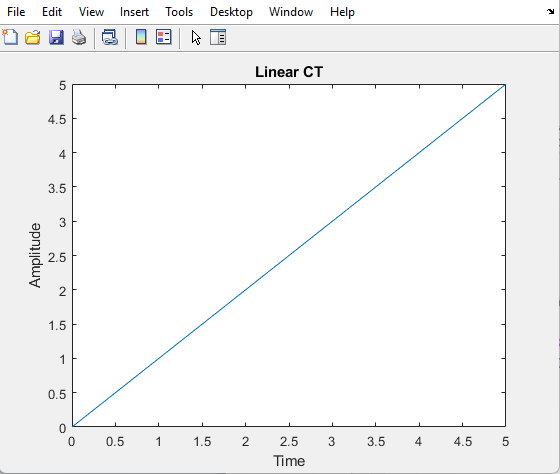
**AIM**

To generate the basic signals in MATLAB: linear, non-linear, exponential, sinusoidal, etc.

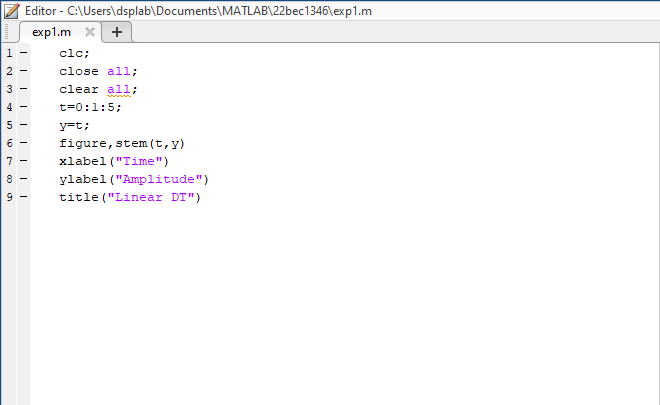
**MATLAB PROGRAM-1**:

****

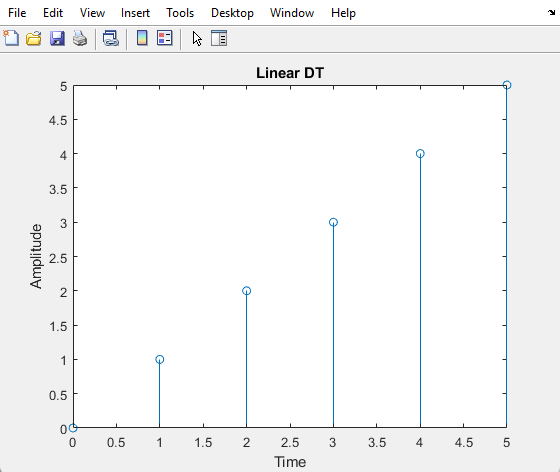
**OUTPUT-1:**

****

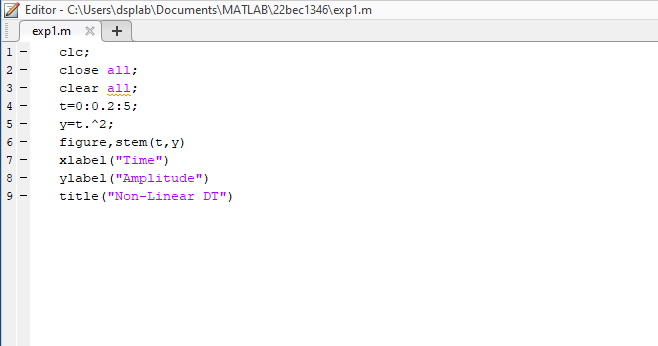
**MATLAB PROGRAM-2:**

****

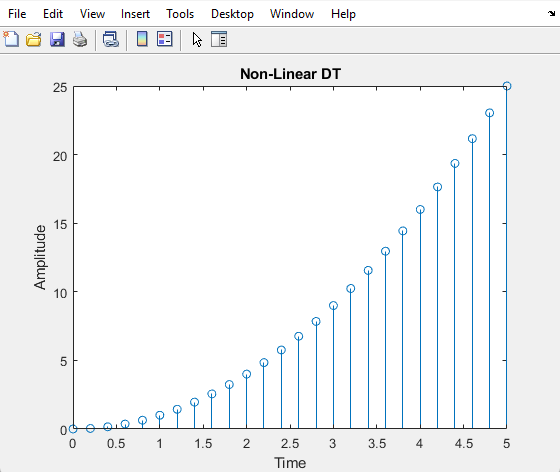
**OUTPUT-2**

****

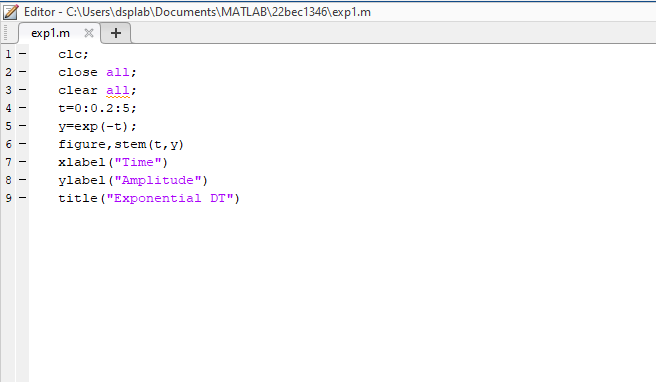
**MATLAB PROGRAM 3:**

****

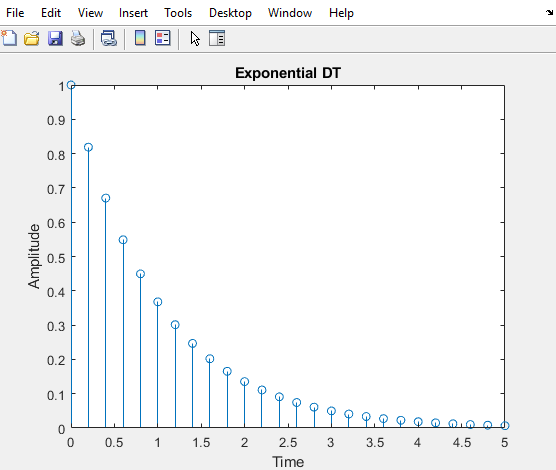
**OUTPUT-3:**

****

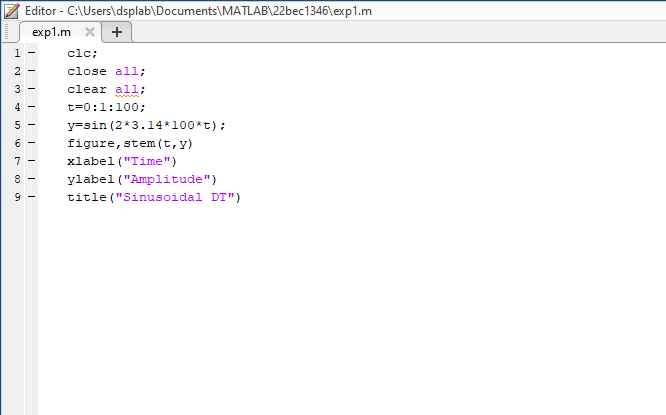
**MATLAB PROGRAM-4:**

****

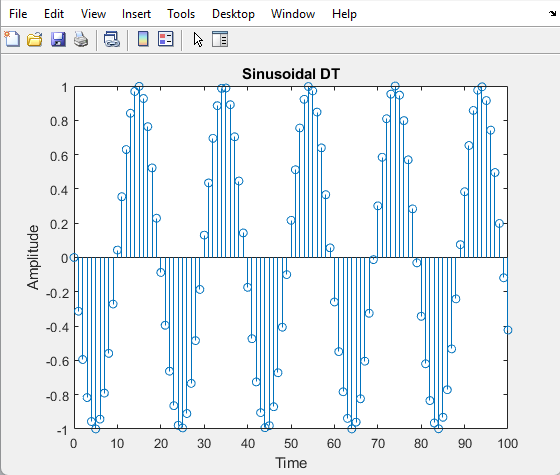
**OUTPUT-4:**

****

**MATLAB PROGRAM-5:**

****

**OUTPUT-5**

****

**CONCLUSION:** Hence all basic signals are generated.

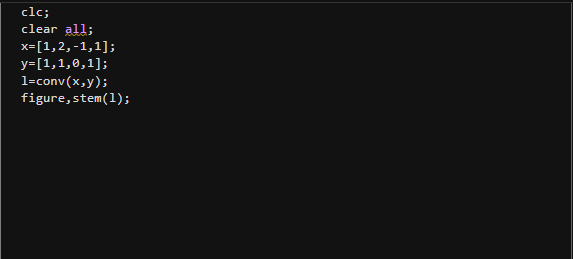
**EXPERIMENT NO: 2**

**DATE: 23-01-2024**

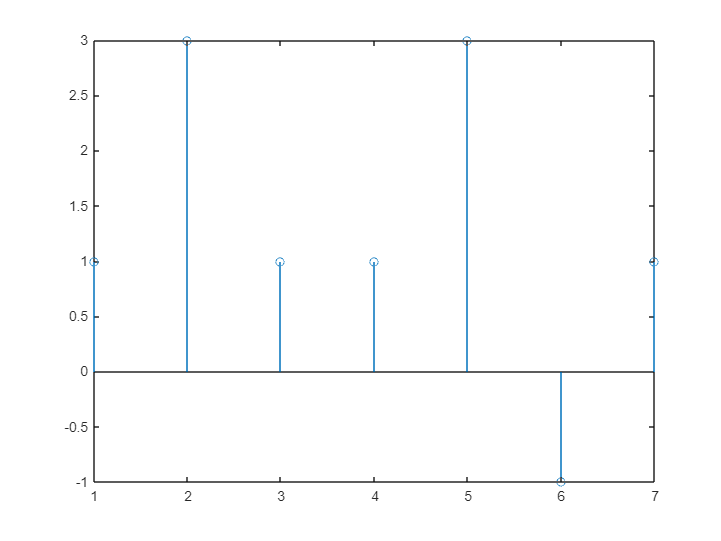
**LINEAR CONVOLUTION AND CIRCULAR CONVOLUTION**

**AIM:** To perform linear and circular convolution.

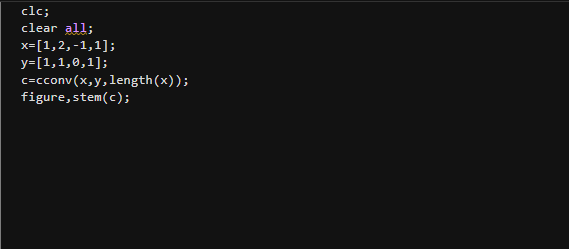
**CODE:**

****

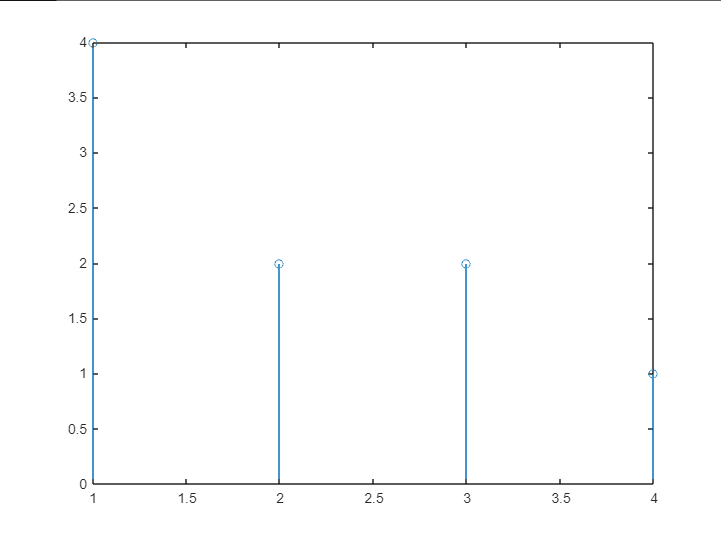
**OUTPUT:**

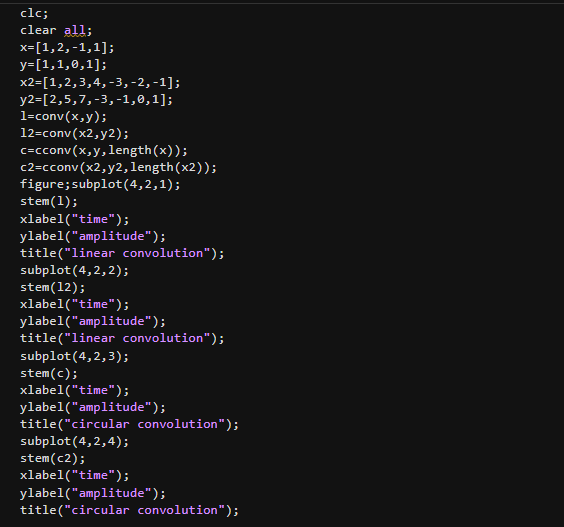
****

**CODE:**

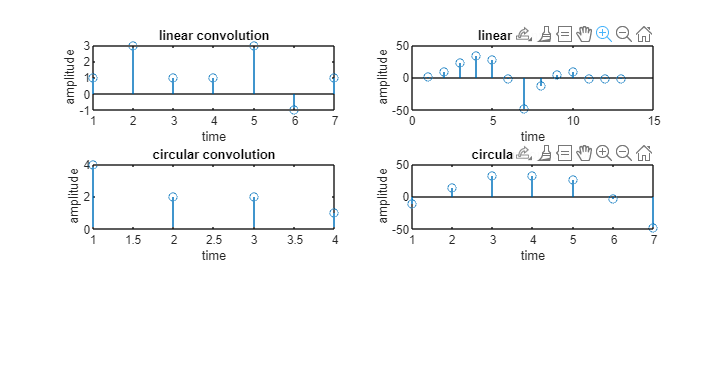
****

**OUTPUT:**

****

****

**OUTPUT:**

****

**CONCLUSION:** Hence linear convolution is performed.

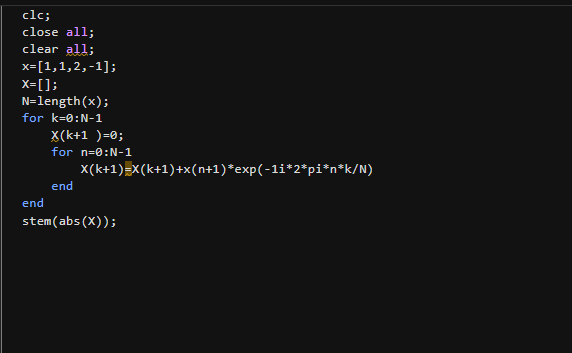
**EXPERIMENT NO: 3**

**DATE: 30-01-2024**

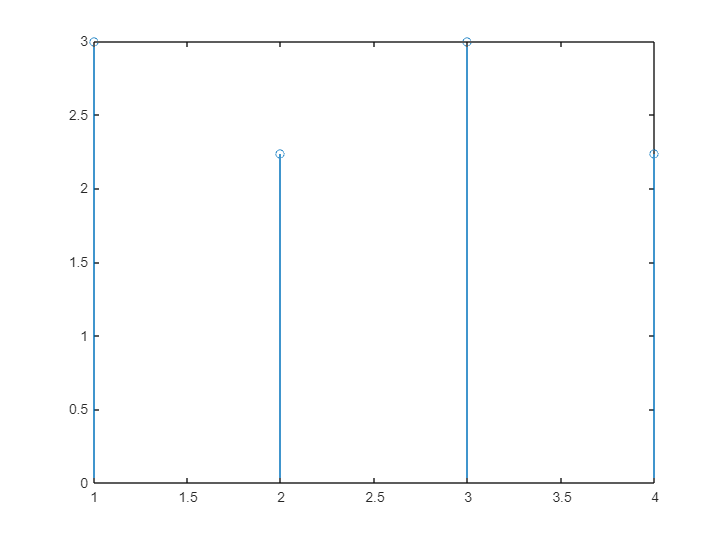
**DISCRETE FOURIER TRANSFORM**

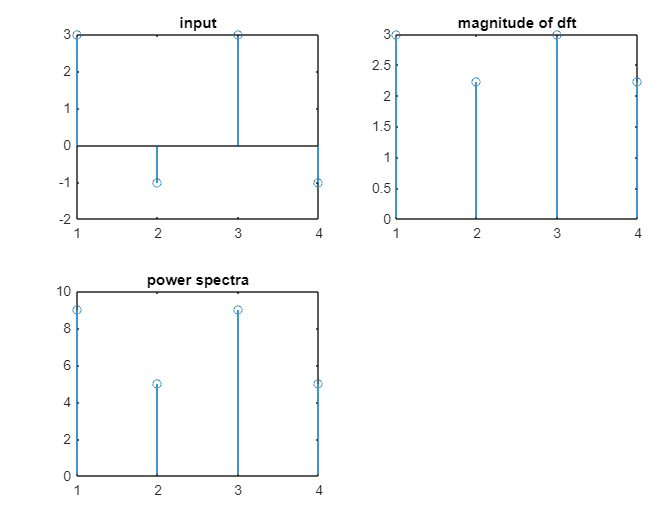
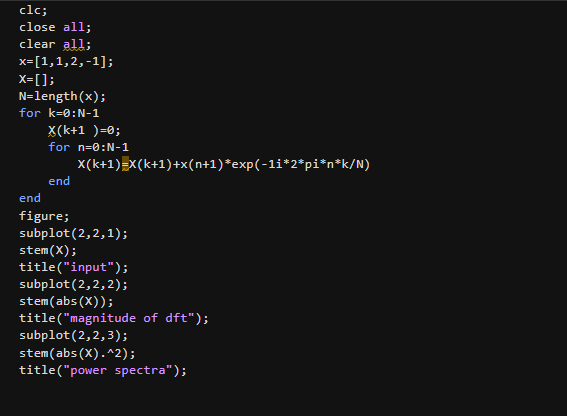
**AIM:** To perform discrete Fourier transform.

**CODE:**

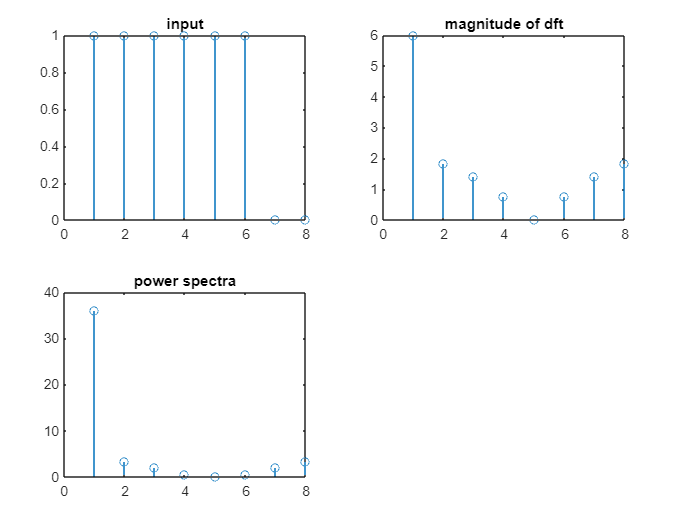
****

**OUTPUT:**

****







**CONCLUSION:** Hence discrete fourier transform is performed.

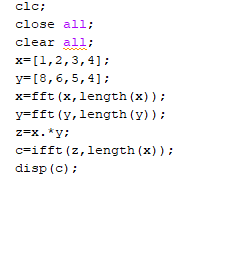
**EXPERIMENT NO: 4**

**DATE: 06-02-2024**

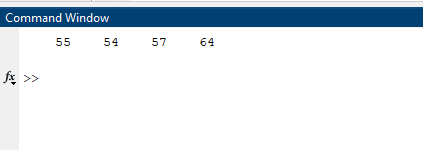
**CIRCULAR CONVOLUTION USING FFT**

**AIM:** To perform circular convolution using fast fourier transform.

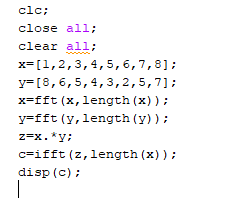
**CODE:**

****

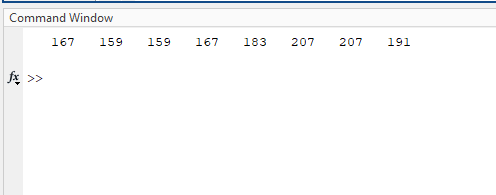
**Output:**

****

**CODE:**

****

**Output:**

****

**CONCLUSION:** Hence circular convolution is performed.

**EXPERIMENT NO: 5**

**DATE: 20-02-2024**

**IIR FILTERS**

**AIM**: To design the following filters:

**BUTTERWORTH LPF**

**CODE:**

clc;

clear all;

close all;

f1=100; f2=300; fs=1000;

t=0:1/fs:4;

x=5\*cos (2\*pi\*f1\*t) + 3\*sin (2\*pi\*f2\*t);

figure;

stem (t, x);

N=5;

fc=150;

[b,a]=butter( N,fc/(fs/2));

figure;

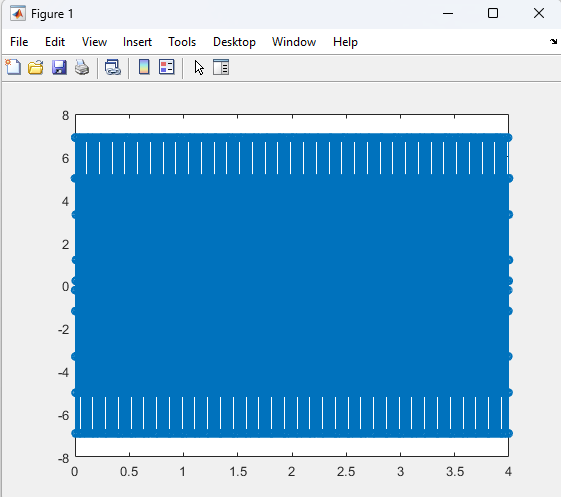
freqz (b,a,[],fs);

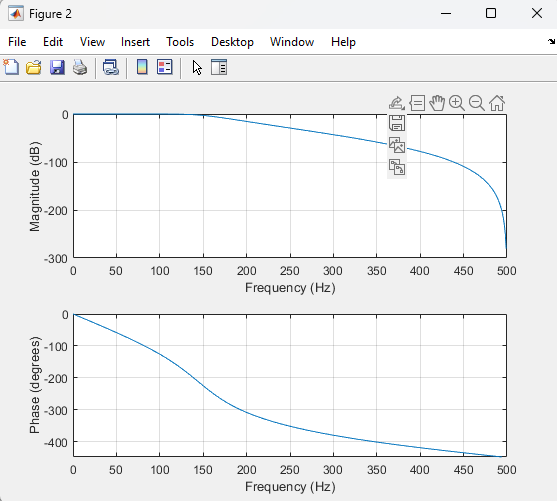
y=filter(b,a,x);

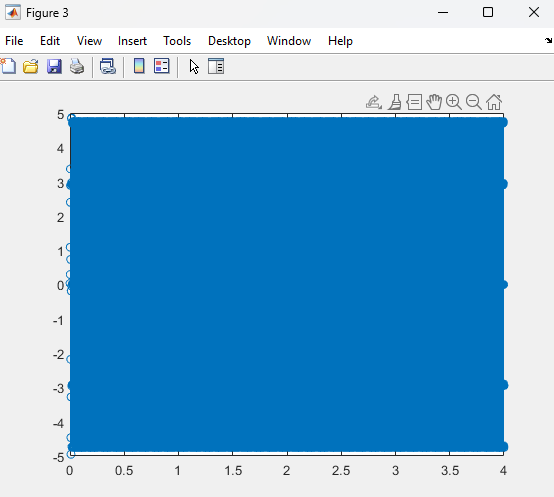
figure;

stem(t,y);

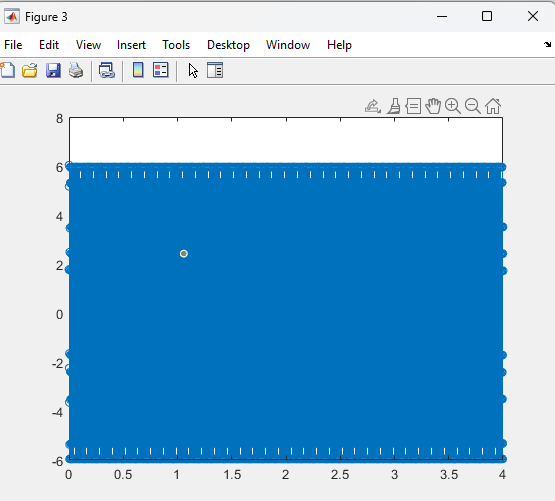
**OUTPUT:**



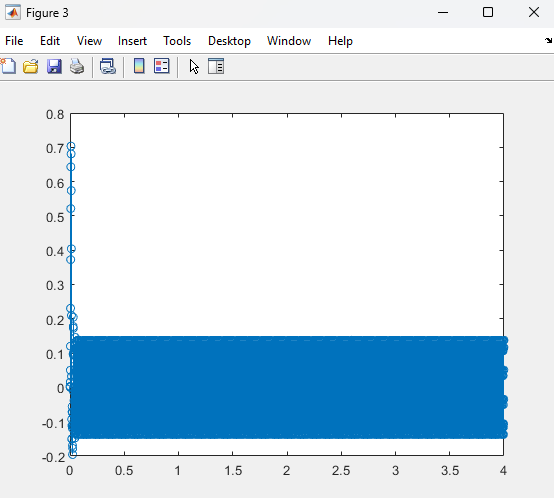




WHEN Fc>f1, f2(say 400 in this case)

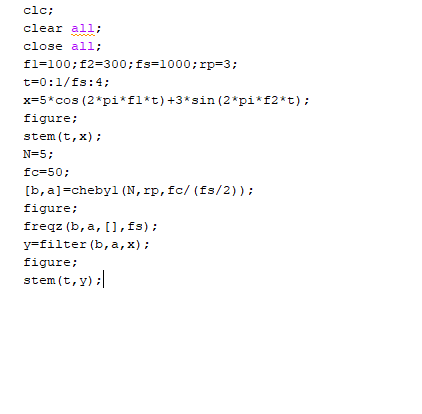


When Fc<f1, f2(say 50 in this case)

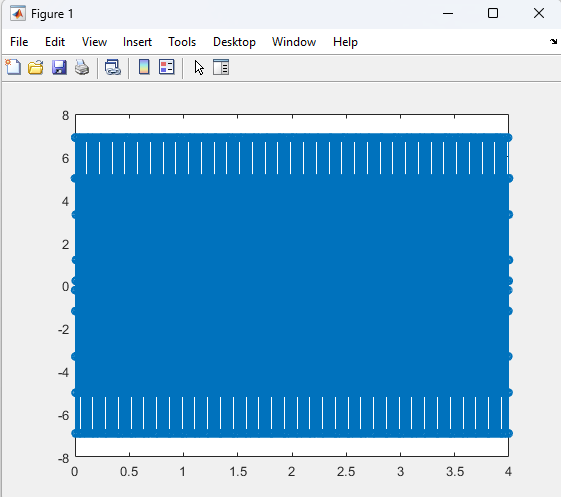
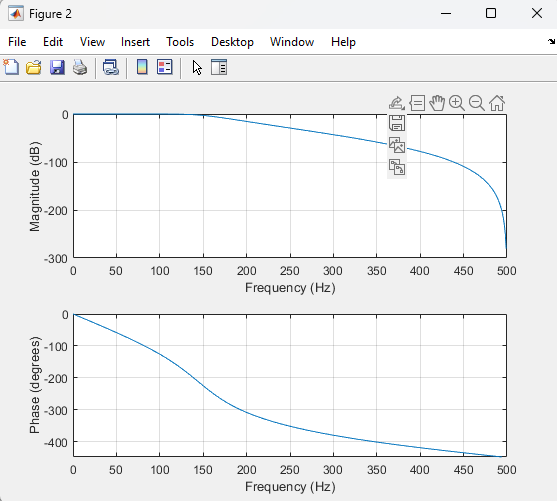


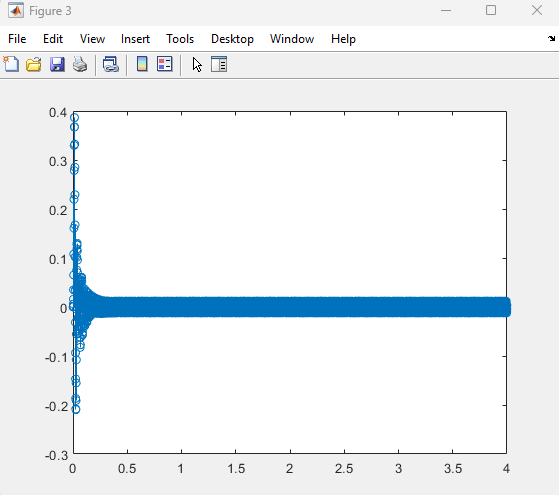
**CHEBYSHEV LPF**

CODE:



OUTPUT:



**CONCLUSION:**

Hence both the filters are designed.

**EXPERIMENT NO: 6**

**DATE: 27-02-2024**

**FIR Filters**

**AIM:** To showcase the application of a Finite Impulse Response (FIR) filter to a given signal.

**CODE:**

clc;

clear all;

close all;

f1 = 100;

f2 = 300;

f3 = 200;

fs = 500;

t = 0:1/fs:0.5;

x = cos(2\*pi\*f1\*t) + cos(2\*pi\*f2\*t) + cos(2\*pi\*f3\*t);

figure;

plot(t, x);

xlabel('Time (s)');

ylabel('Amplitude');

title('Original Signal');

n = 5;

fc = 100;

b = fir1(n, fc/(fs/2), 'low', boxcar(n+1));

figure;

freqz(b, 1, 512, fs);

title('Frequency Response of the Filter');

y = filter(b, 1, x);

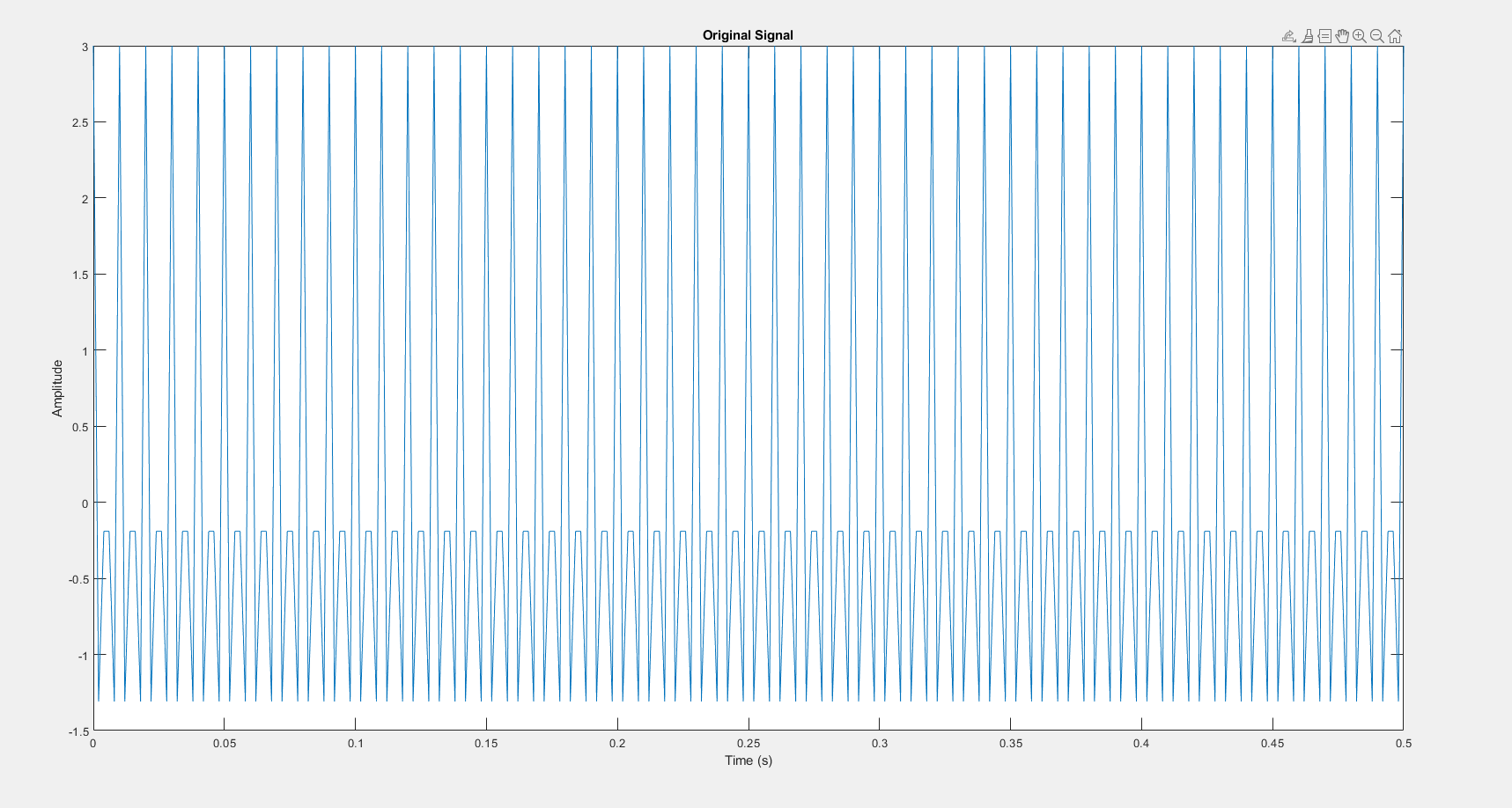
figure;

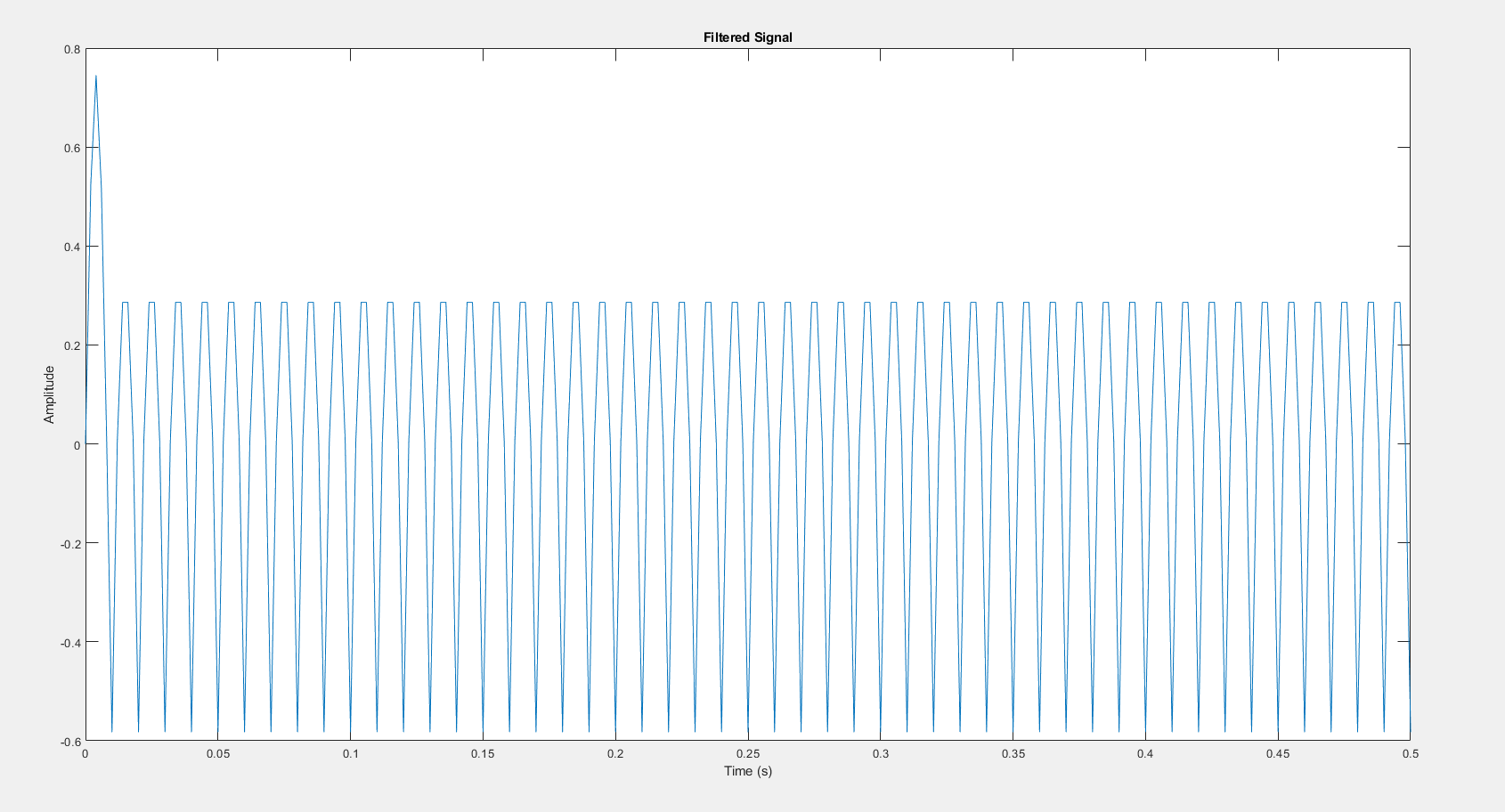
plot(t, y);

xlabel('Time (s)');

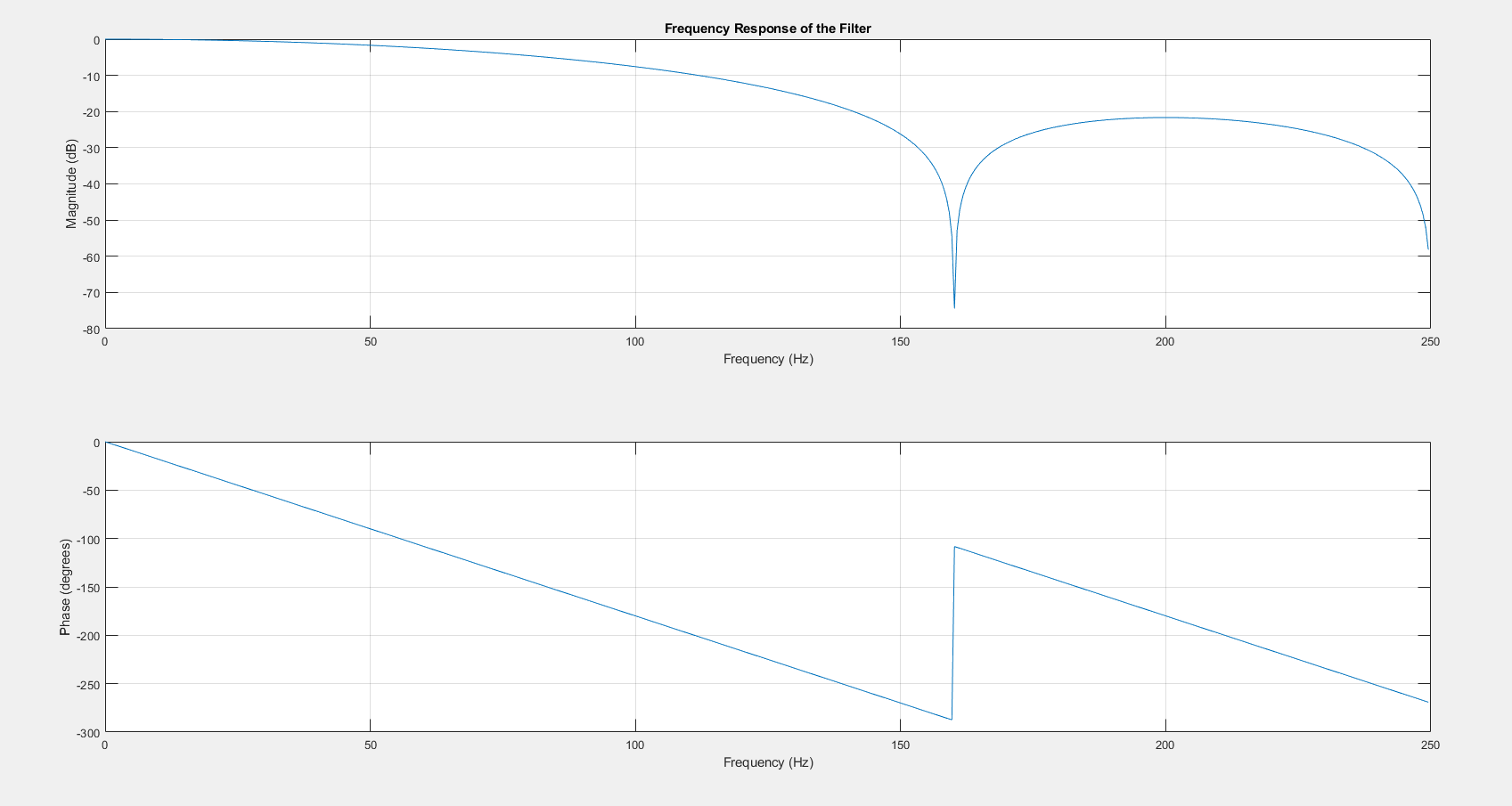
ylabel('Amplitude');

title('Filtered Signal');





**CONCLUSION:** Thus application of FIR filter is performed.



**EXPERIMENT NO: 7**

**DATE: 05-03-2024**

**Band-Pass Filters**

**AIM:** To showcase the application of a Band Pass filter to a given signal.

**CODE:**

clc

clear all;

close all;

f1=100, f2=300, f3=400, fs=1000;

t=0:1/fs:0.15;

x=cos(2\*pi\*f1\*t)+sin(2\*pi\*f2\*t);

figure;

plot(t,x);

N=5;

F=[0 0.1 0.3 0.5 0.7 0.8 0.9 1];

A=[0 0 1 1 1 1 0 0];

b=firpm(N,F,A);

figure;

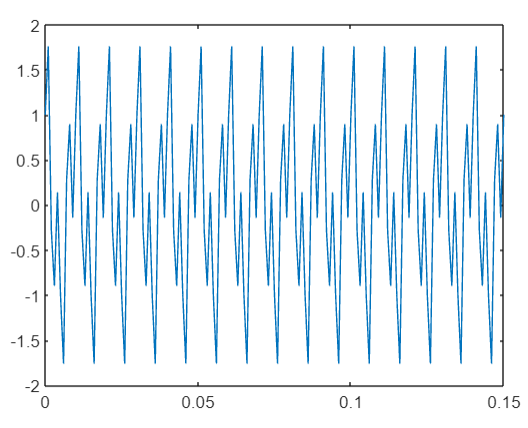
freqz(b,1,512);

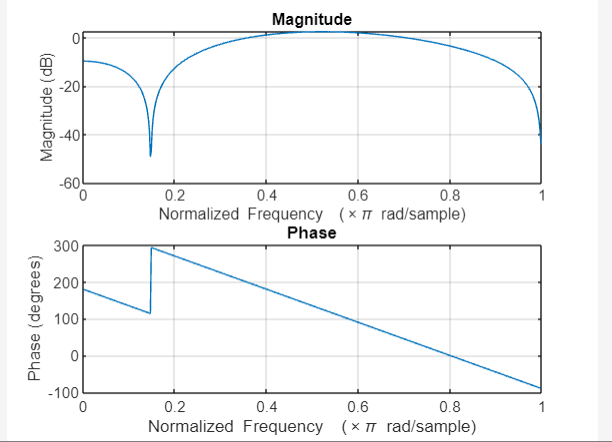
y=filter(b,1,x);

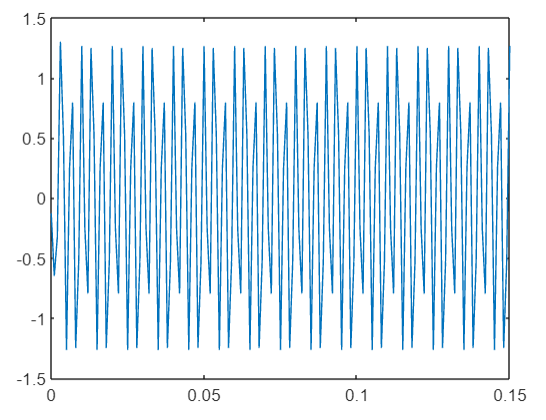
figure;

plot(t,y);

**OUTPUT:**







**CONCLUSION:** Hence application of bandpass filter is performed.

**EXPERIMENT NO: 8**

**DATE: 16-04-2024**

**SAMPLING RATE CONVERSION**

**AIM:** To perform sampling rate conversion on the given function.

**CODE:**

clc

clear all;

close all;

x=[1,3,-1,4,6,8,9,-1,3];

xu=upsample(x,2);

xd=downsample(x,3);

xud=downsample(xu,3);

figure;stem(x);

figure;stem(xu);

figure;stem(xd);

figure;stem(xud);

**OUTPUT:**

A screen shot of a graph

Description automatically generated

A screen shot of a graph

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a graph

Description automatically generated

**CONCLUSION:** Hence sampling rate conversion is performed.

**EXPERIMENT NO: 9**

**DATE:26/03/2024**

**HARDWARE**

**LCDK-C6748 HARDWARE APPLICATIONS AND FEATURES.**

**AIM**

To present the LCDK-C6748 HARDWARE APLICATIONS AND FEATURES.

**Introduction to LCDK C6748 Hardware**

The LCDK C6748 is a powerful and versatile development board designed for embedded systems and digital signal processing applications. This compact board packs a punch, featuring a high-performance ARM Cortex-A8 processor and a rich set of peripherals to enable a wide range of project possibilities.

# Processor Architecture

#### Powerful ARM Cortex-A8 CPU

The LCDK C6748 is powered by a high-performance ARM Cortex-A8 processor, clocked at up to 600MHz, delivering ample computational power for demanding applications.

#### Extensive Peripherals

The board includes a diverse array of on-chip peripherals, such as timers, serial interfaces, and analog-to-digital converters, enabling comprehensive system integration.

#### Efficient Power Management

The processor architecture is designed for optimal power efficiency, making the LCDK C6748 well-suited for battery-powered and energy-conscious projects.

****

# Connectivity Options

#### Ethernet

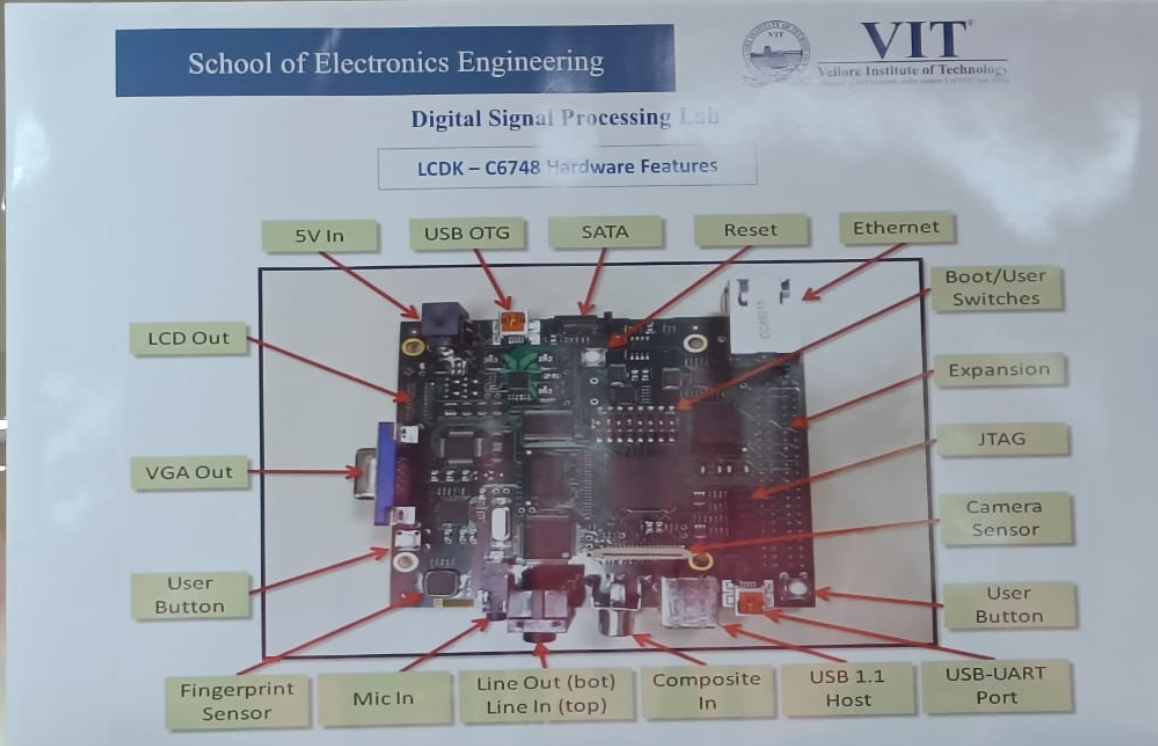
The LCDK C6748 features a built-in Ethernet interface, enabling high-speed, reliable network connectivity for networked applications and remote access.

#### Wi-Fi

Optional Wi-Fi connectivity allows the LCDK C6748 to integrate seamlessly into wireless networks and enable remote monitoring and control capabilities.

#### Bluetooth

Bluetooth support on the LCDK C6748 enables wireless communication with other Bluetooth-enabled devices, expanding the board's connectivity and application possibilities.



**Conclusion and Key Specifications**

|  |  |
| --- | --- |
| Processor | ARM Cortex-A8 (up to 600MHz) |
| Memory | 512MB DDR3 SDRAM, 64MB NAND Flash |
| Connectivity | Ethernet, USB, Wi-Fi, Bluetooth |
| Peripherals | Timers, Serial Interfaces, ADC |
| Power | USB or External Power Supply |

The LCDK C6748 is a powerful and versatile development board that offers a rich set of features and capabilities, making it an excellent choice for a wide range of embedded systems and digital signal processing applications.