LAB PRACTICAL FILE

Signal And Systems EC-205

DELHI TECHNOLOGICAL UNIVERSITY

Shahbad Daulatpur, Main Bawana road

Dellhi – 110042

Submitted To

Dr. Priyanka Jain

Submitted By

Harshal Chowdhary

2K19/EC/071

INDEX

Experiment – 1

Aim:

Generate basic signals (unit impulse, sine, cosine, unit step, unit ramp & exponential function) with MATLAB

Introduction:

The **unit impulse function** is usually taken to mean a rectangular pulse of **unit** area, and in the limit the width of the pulse tends to zero whilst its magnitude tends to infinity. The **Sine Function** is defined in the context of a right triangle: for the specified angle, it is the ratio of the length of the side that is opposite that angle to the length of the longest side of the triangle. The **Cosine Function** is the ratio of the adjacent side to that of the hypotenuse.

The **ramp functions** with unity slope i.e. having magnitude of one always, is called **unit ramp function**. **Exponential functions** have the form $f(x) = b^x$, where $b > 0$ and $b \neq 1$. Just as in any exponential expression, b is called the **base** and x is called the exponent.

Code:

```
clc;
clear all;
close all;

%Sine Function
a1=input('Enter the Amplitude for Sine Function:');
f1=input('Enter the Frequency for Sine function:');
t1_start=input('Enter the starting time for sine Function:');
t1_end=input('Enter the ending time for Sine Function:');
t1=t1_start:0.01:t1_end;
xs=a1*sin(2*pi*f1*t1);
figure('Name','2K19/EC/071 Harshal Chowdhary','NumberTitle','off');
subplot(3,2,1);
plot(t1,xs,'g-');
title('Sine Function');
xlabel('Time');
ylabel('Amplitude');

%Cosine Function
a2=input('Enter the Amplitude for cOSine Function:');
f2=input('Enter the Frequency for cosine function:');
t2_start=input('Enter the starting time for cosine Function:');
t2_end=input('Enter the ending time for cosine Function:');
t2=t2_start:0.01:t2_end;
```

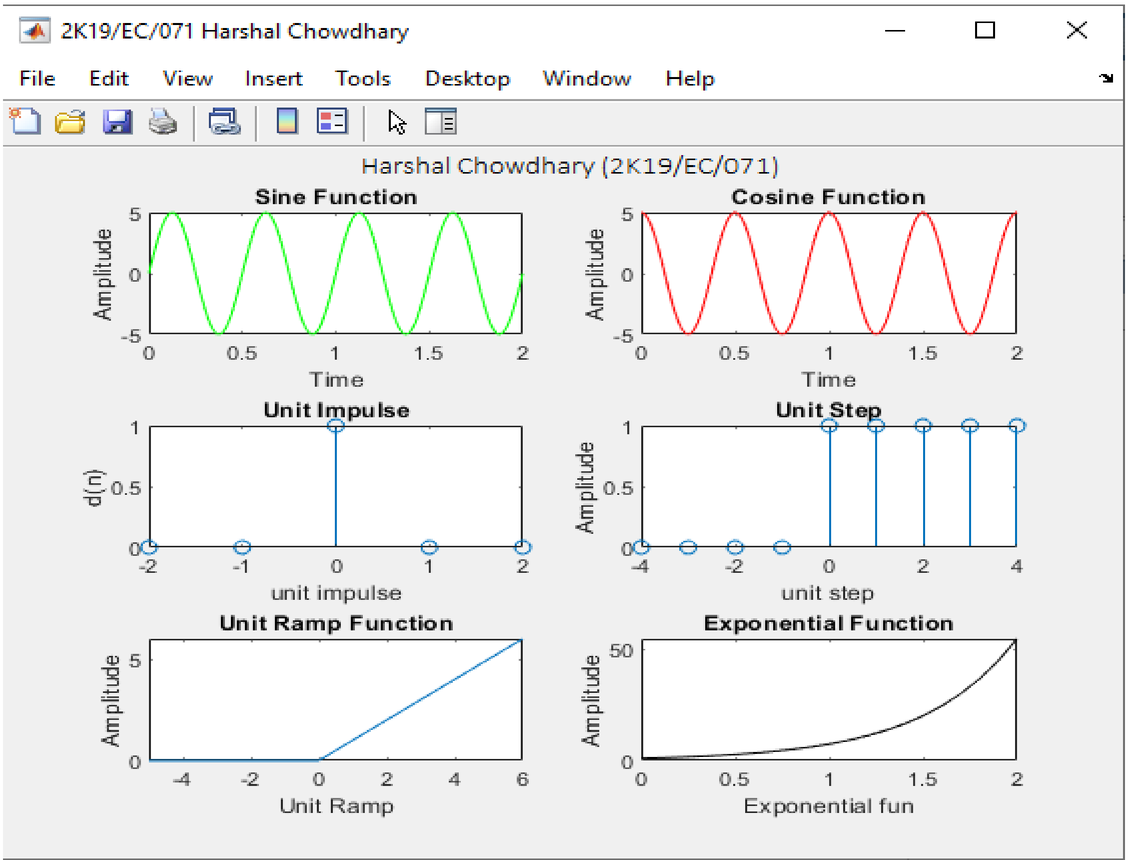
```
xc=a2*cos(2*pi*f2*t2);
subplot(3,2,2);
plot(t2,xc,'r-');
title('Cosine Function');
xlabel('Time');
ylabel('Amplitude');

%Unit Impulse function
t=-2:2;
y=[zeros(1,2), ones(1,1), zeros(1,2)];
subplot(3,2,3);
stem(t,y);
title('Unit Impulse');
ylabel('d(n)');
xlabel('unit impulse');

%Unit Step Function
n1=input('Pls enter the value of n:');
t7=-n1+1:n1-1;
y3=[zeros(1,n1-1),ones(1,n1)];
subplot(3,2,4);
stem(t7,y3);
title('Unit Step');
ylabel('Amplitude');
```

```
xlabel('unit step');

%Unit Ramp Function
n2=input('Till what value you want to see graph:');
y6=1:n2;
t5=-5:n2;
xr=[zeros(1,6) y6];
subplot(3,2,5);
plot(t5,xr);
title('Unit Ramp Function');
ylabel('Amplitude');
xlabel('Unit Ramp');

%Exponential Function
nf=input('Enter length of exponential seq:');
t6=0:0.1:nf-1;
p=input('Enter a value:');
y6=exp(p*t6);
subplot(3,2,6)
plot(t6,y6,'k-')
title('Exponential Function');
xlabel('Exponential fun');
ylabel('Amplitude');
```

Input:

```
Command Window
Enter the Amplitude for Sine Function:5
Enter the Frequency for Sine function:2
Enter the starting time for sine Function:0
Enter the ending time for Sine Function:2
Enter the Amplitude for cOSine Function:5
Enter the Frequency for cosine function:2
Enter the starting time for cosine Function:0
Enter the ending time for cosine Function:2
Pls enter the value of n:5
Till what value you want to see graph:6
Enter length of exponential seq:3
Enter a value:2
```

Output:



Learning Outcome:

Grasps the basic knowledge of MATLAB software and got a clear vision about plotting procedure of different signals on the MATLAB software. Learned about different types of signals.

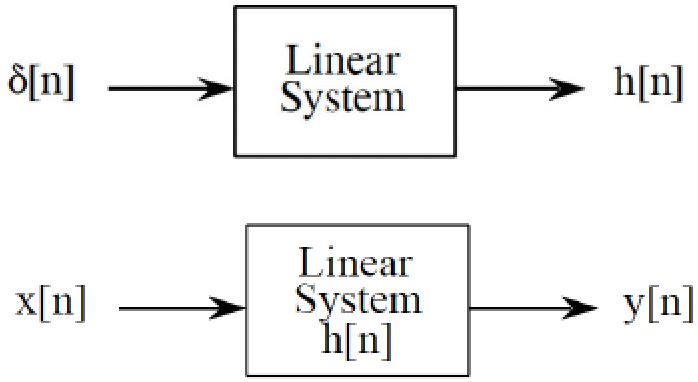
Experiment-2

Aim

Generate a MATLAB code to implement the Convolution between two functions

Theory

Convolution is a mathematical way of combining two **signals** to form a third **signal**. It is the single most important technique in Digital **Signal** Processing. ... **Convolution** is important because it relates the three **signals** of interest: the input **signal**, the output **signal**, and the impulse response.



$$x[n] * h[n] = y[n]$$

$$y[n] = x[n] ** h[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n-k]$$

Code (For Standard Sequences):

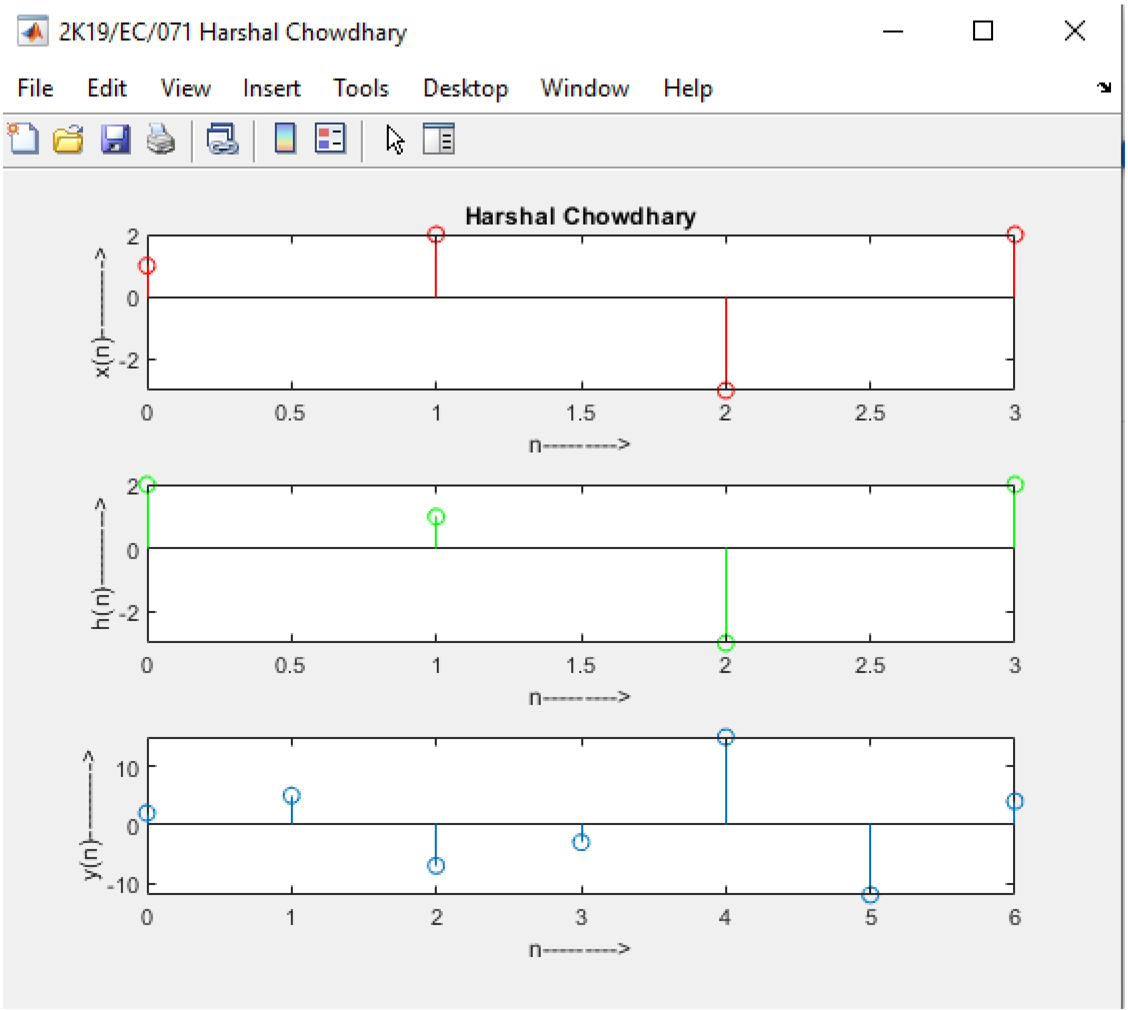
```
clc;
clear all;
close all;

%Taking the sequence from users.
x=input('Enter The first Sequence: ');
nx=0:1:length(x)-1;
h=input('Enter The second Sequence: ');
nh=0:1:length(h)-1;

%Using Conv Function
y=conv(x,h)
ny=0:length(y)-1;

%Plotting of Graphs
figure('Name','2K19/EC/071 Harshal Chowdhary','NumberTitle','off');

subplot(3,1,1);
stem(nx,x,'r-');
title('Harshal Chowdhary');
ylabel('x(n)---------->');
xlabel('n---------->');

subplot(3,1,2);
```

```
subplot(3,1,2);
stem(nh,h,'g-');
ylabel('h(n)---------->');
xlabel('n---------->');

subplot(3,1,3);
stem(ny,y);
ylabel('y(n)---------->');
xlabel('n---------->');
```

Command Window:

```
Enter The first Sequence: [1 2 -3 2]
Enter The second Sequence: [2 1 -3 2]

y =

     2     5    -7    -3    15   -12     4
```

Output:



Code (For General Sequences):

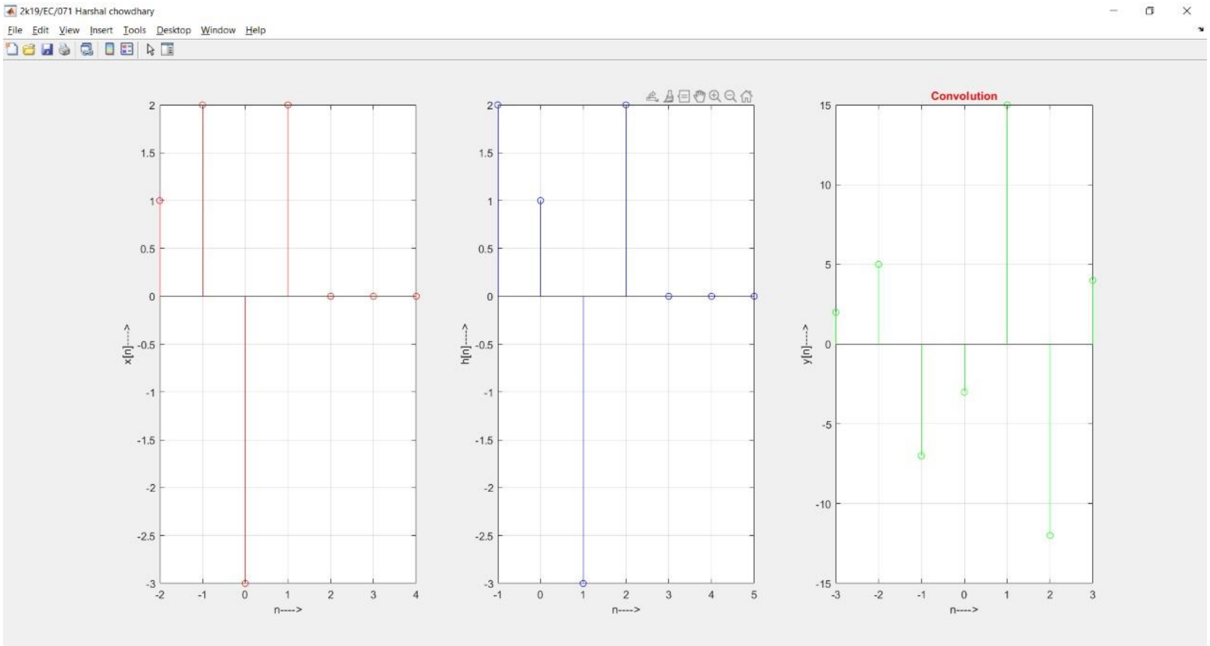
```
clc;
clear all;
close all;

%Input from user of sequence
x=input('Enter The first Sequence: ');
h=input('Enter The second Sequence: ');

%Input Regarding Starting Of Sequence
sx=input('starting point of x[n]: ');
sh=input('Starting point of h[n]: ');
sy=sx+sh;

N=length(x)+length(h)-1;
```

```
yl=[zeros(1,N)];
xl=[x,zeros(1,length(x)-1)]
hl=[h,zeros(1,length(h)-1)]

for i=1:N
    for j=1:i
        yl(i)= yl(i)+xl(j)*hl(i-j+1);
    end;
end;
yl

%Plotting Graphs
figure('Name','2K19/EC/071 Harshal Chowdhary','NumberTitle','off');

subplot(3,1,1);
stem(sx:1:sx+length(xl)-1,xl,'r-');
title('Harshal Chowdhary');
ylabel('Amplitude --------->');
xlabel('n--------->');

subplot(3,1,2);
stem(sh:1:sh+length(hl)-1,hl,'g-');
ylabel('Amplitude --------->');
xlabel('n--------->');

subplot(3,1,3);
stem(sy:1:sy+length(yl)-1,yl);
ylabel('Convolution --------->');
xlabel('n--------->');
```

Command Window:

```
Enter The first Sequence: [1 2 -3 2]
Enter The second Sequence: [2 1 -3 2]
starting point of x[n]: -2
Starting point of h[n]: -1

xl =

     1     2    -3     2     0     0     0


hl =

     2     1    -3     2     0     0     0


yl =

     2     5    -7    -3    15   -12     4
```

Output



2k19/EC/071 Harshal chowdhary
File Edit View Insert Tools Desktop Window Help
Convolution
x[n]---->
h[n]---->
y[n]---->
n---->
n---->
n---->

Manual Solution

8

$x[n] = [1 \ 2 \ -3 \ 2]$

$h[n] = [2 \ 1 \ -3 \ 2]$

Graphical Method

Convolution $y[n] = x[n] * h[n]$

Starting point of $x[n]$ $S_x = 0$

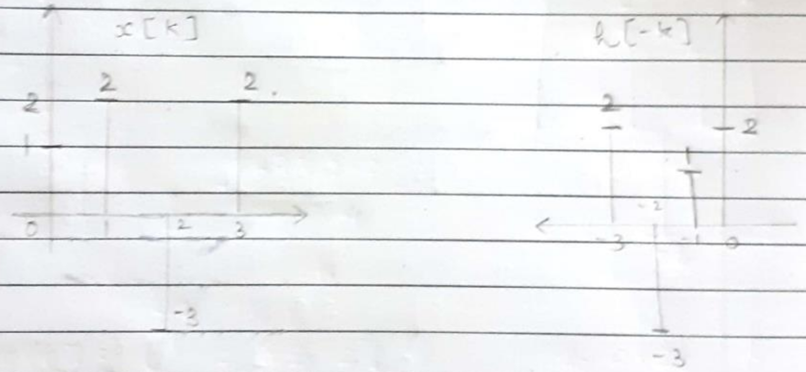
Starting point of $h[n]$ $S_h = 0$

No. of samples in $y[n] = 4 + 4 - 1 = 7$

$$y[n] = \sum_{k=0}^{3} x[k]\, h[n-k]$$

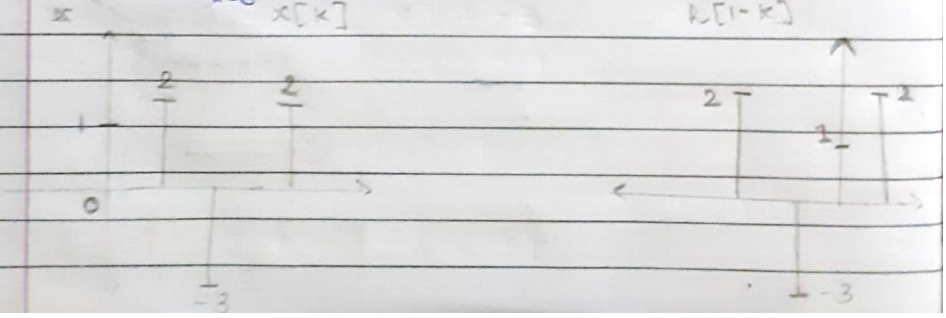
$$y[0] = \sum_{k=0}^{3} x[k]\, h[n-k]$$

∴



$y[0] = 1 \times 2 = 2$

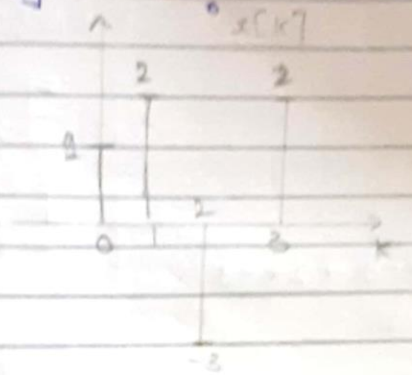
∴ $y[1] = \sum_{k=0}^{3} x[k]\, h[1-k]$

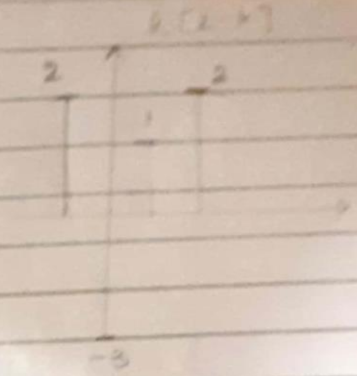


$y[1] = 1\times1 + 2\times2 = 5$

$\boxed{y[1] = 5}$

* $y[2] = \sum_{0}^{3} x[k]\, h[2-k]$

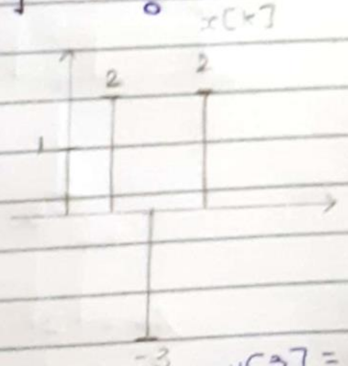


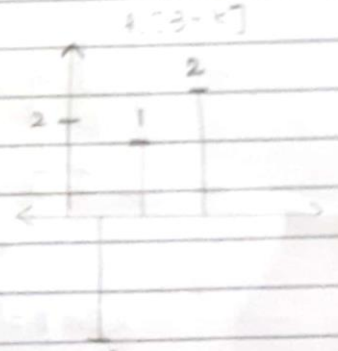


$y[2] = 1\times(3) + 2\times1 - 3\times2$

$\boxed{y[2] = -7}$

* $y[3] = \sum_{0}^{3} x[k]\, h[3-k]$

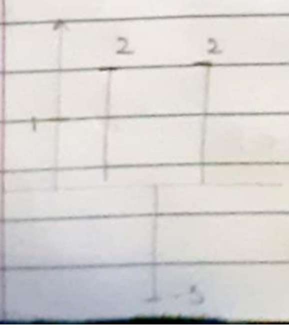


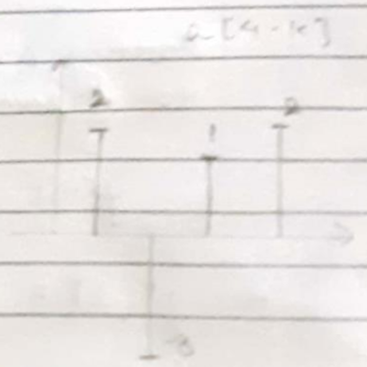


$y[3] = 1\times2 + 2\times-3 + -3\times1 + 2\times2$

$\boxed{y[3] = -3}$

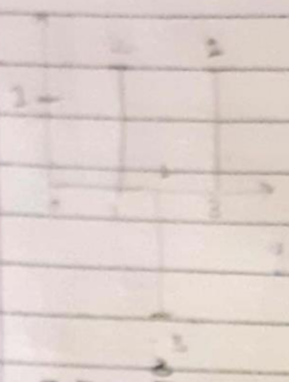
* $y[4] = \sum_{0}^{3} x[k]\, h[4-k]$

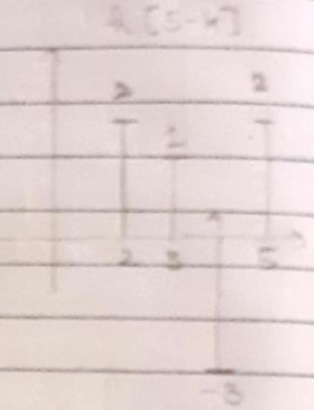




$y[4] = 2\times2 + -3\times-3 + 2\times1$
$= -15$

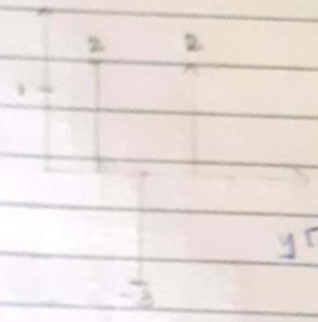
* $y[5] = \sum_{k=0}^{3} x[k]\,h[5-k]$

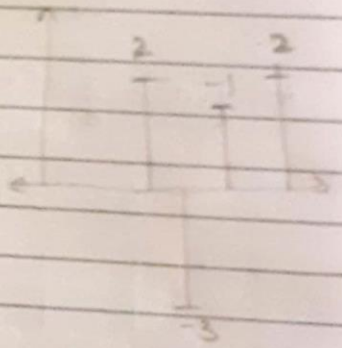




$y[5] = 2\times-3 + 2\times-3$
$= -12$

* $y[6] = \sum_{k=0}^{3} x[k]\,h[6-k]$





$y[6] = 2\times2$
$= 4$

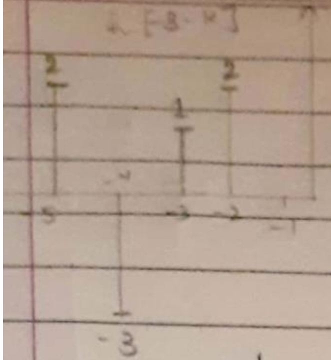
$$\begin{array}{l} \quad\quad -2 \quad\ \ x=0\downarrow \\ \longrightarrow \quad x[n] = [\,1 \quad 2 \quad -3 \quad 2\,] \\ \quad\quad\ \ h[n] = [\,2 \quad 1 \quad -3 \quad 2\,] \\ \quad\quad -1 \quad \uparrow x=0 \end{array}$$

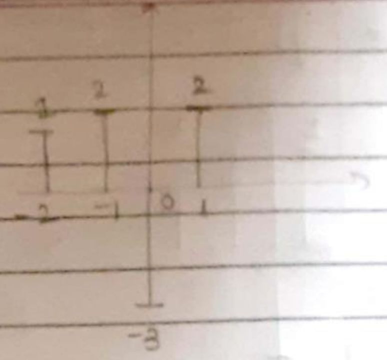
$l_x = -2$
$l_y = -1$

Starting point of $y[x] = -3$
No. of samples $= 4+4-1 = 7$

$y[n] = \sum_{k=-2}^{1} x[k]\,h[n-k]$

* $y[-3] = \sum_{-2}^{1} x[k]\,h[-3-k]$

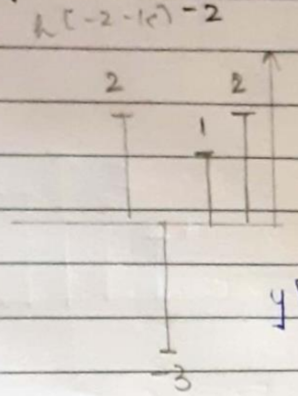


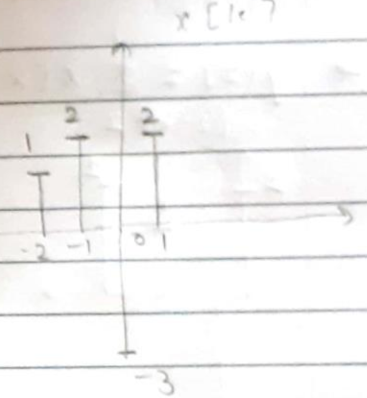


$y[-3] = 2 \times 1$

$= 2$

* $y[-2] = \sum_{-2}^{1} x[k]\, h[-2-k]$

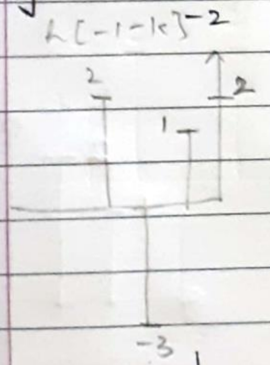




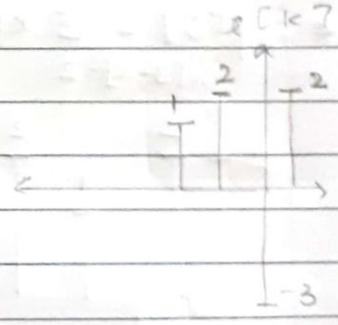
$y[-2] = 1 \times 1 + 2 \times 2$

$= 5$

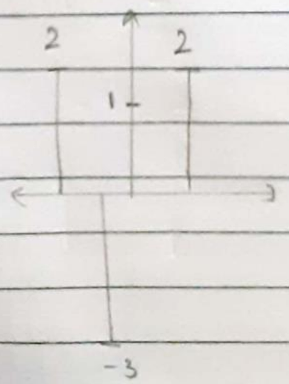
* $y[-1] = \sum_{-2}^{1} x[k]\, h[-1-k]$



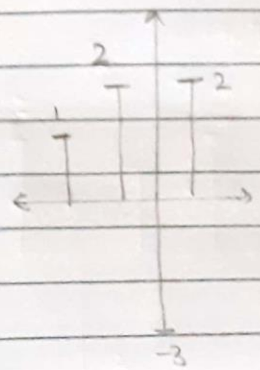
$y[-1] = -7$

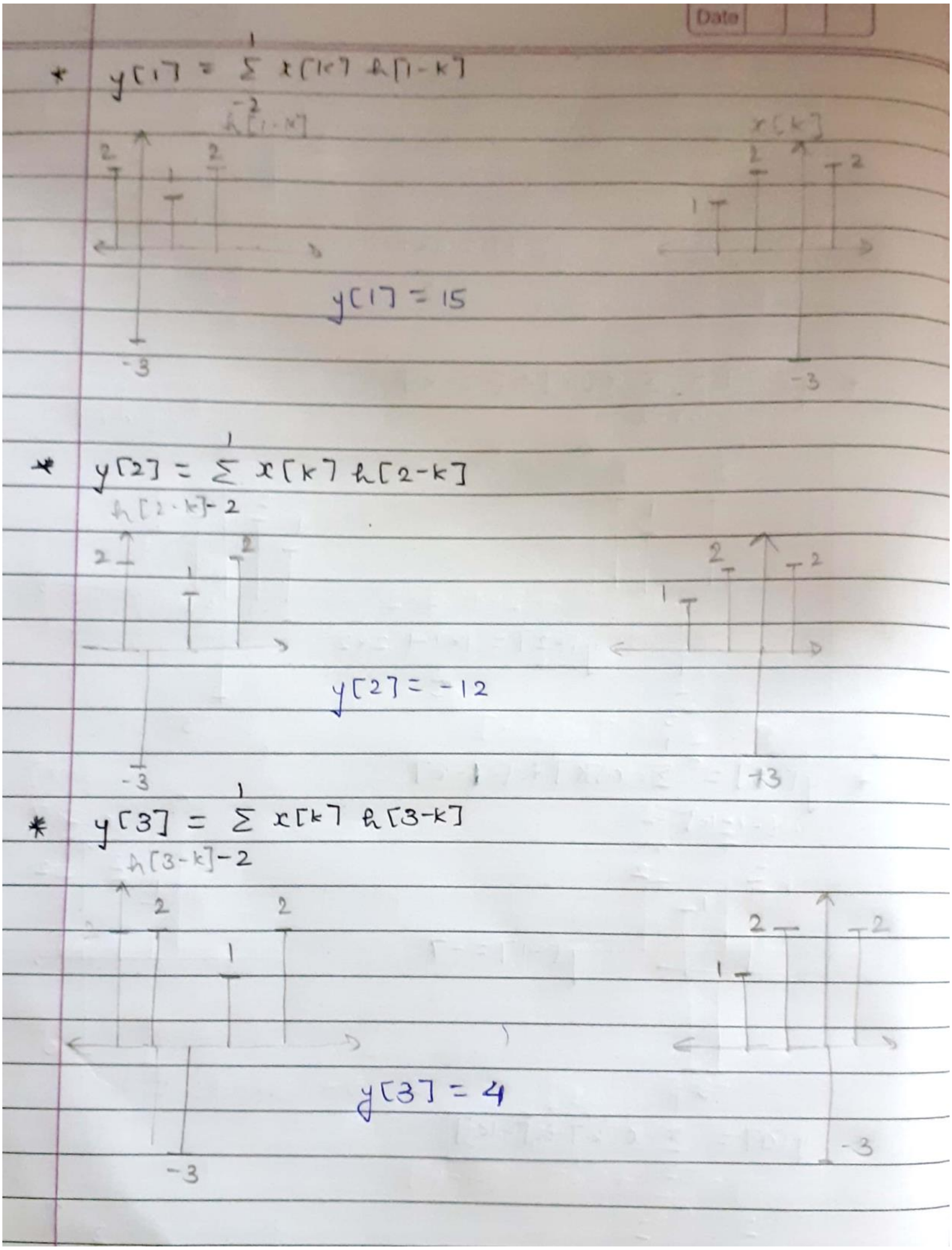


* $y[0] = \sum_{-2}^{1} x[k]\, h[-k]$



$y[0] = -3$





Learning Output:

Grasps the basic knowledge of Convolution of Signals and made code for both general and standard sequences. Also verified the solution by manually performing convolution.

Experiment-3

Aim

Generate a MATLAB code to implement the Correlation.

Theory

In general, **correlation** describes the mutual relationship which exists between two or more things. The same definition holds good even in the case of **signals**. That is, **correlation** between **signals** indicates the measure up to which the given **signal** resembles another **signal**.

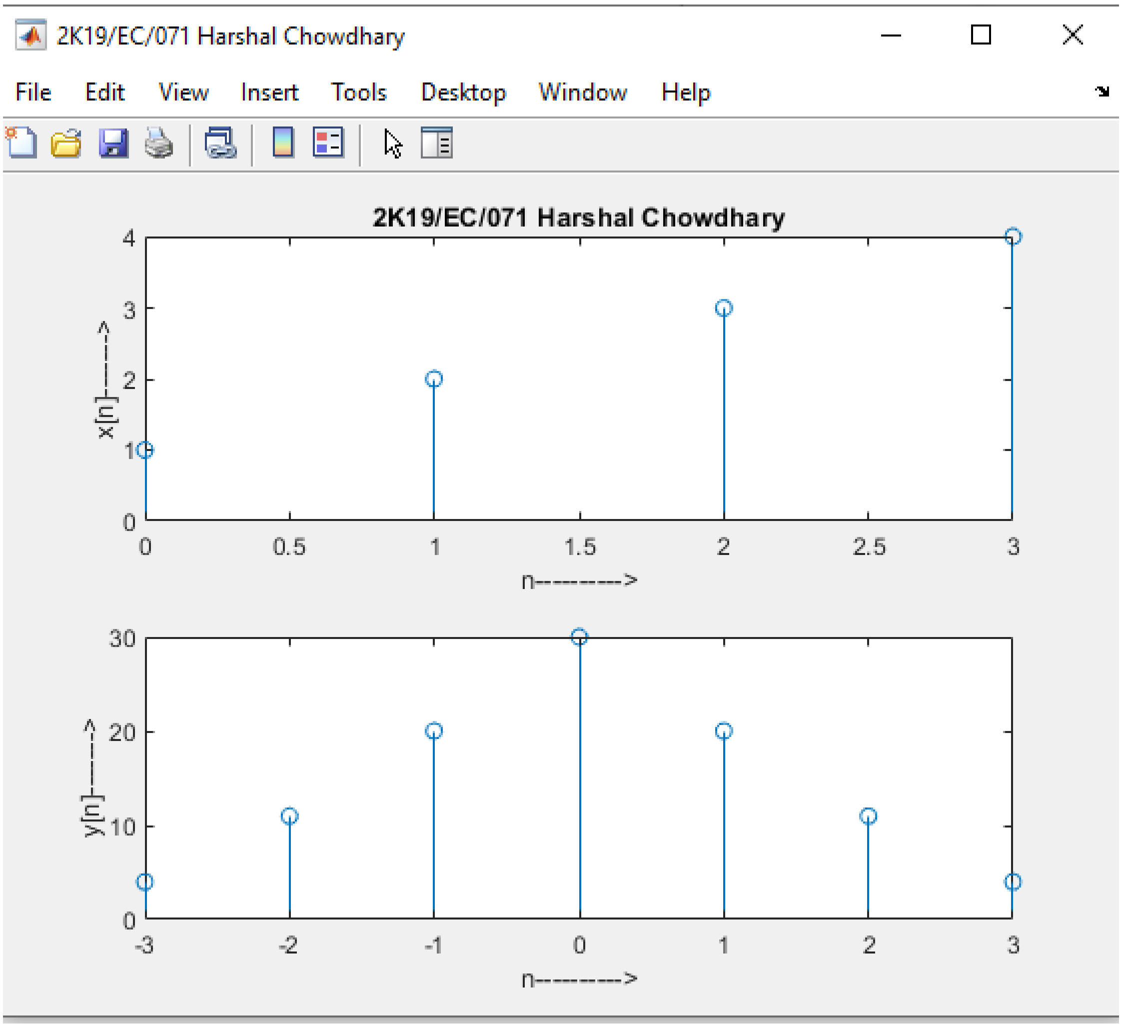
Code (AutoCorrelation):

```
clc;
clear all;
close all;

x=input('Please Enter The sequence');
n=1-length(x):length(x)-1;

y=xcorr(x,x)

figure('Name','2K19/EC/071 Harshal Chowdhary','NumberTitle','off');

subplot(2,1,1);
stem(x);
title('2K19/EC/071 Harshal Chowdhary');
xlabel('n---------->');
ylabel('x[n]------->');

subplot(2,1,2);
stem(n,y);
xlabel('n---------->');
ylabel('y[n]------->');
```

Input:

```
Please Enter The sequence[1 2 3 4]

y =

    4.0000   11.0000   20.0000   30.0000   20.0000   11.0000    4.0000
```

Output:



2K19/EC/071 Harshal Chowdhary
File Edit View Insert Tools Desktop Window Help
2K19/EC/071 Harshal Chowdhary
x[n]--------->
n--------->
y[n]--------->
n--------->

Manual Solution:

$\longrightarrow$ $x[n] = \{\overset{\downarrow}{1}, 2, 3, 4\}$

Range would be -3 to 3

$$y(-3) = \sum_{k=0}^{3} x(k)\,x(k+3)$$

$= x(0)\,x(3) + x(1)\,x(4)$

$= 1 \times 4 = 4$

$$y(-2) = \sum_{k=0}^{3} x(k)\,x(k+2)$$

$= x(0)\,x(2) + x(1)\,x(3)$

$= 3 + 8 = 11$

$$y(-1) = \sum_{k=0}^{3} x(k)\,x(k+1)$$

$= x(0)\,x(1) + x(1)\,x(2) + x(2)\,x(3)$

$= 1\times2 + 2\times3 + 3\times4$

$= 20$

$$y(0) = \sum_{k=0}^{3} x(k)\,x(k)$$

$= x(0)\,x(0) + x(1)\,x(1) + x(2)\,x(2) +$
$x(3)\,x(3)$

$= 1\times1 + 2\times2 + 3\times3 + 4\times4$

$= 30$

$$y(1) = \sum_{k=0}^{3} x(k)\,x(k-1)$$

$= x(0)\,x(-1) + x(1)\,x(0) + x(2)\,x(1)$

$+ x(3)\,x(2)$ ~~+x~~

$= 1\times2 + 2\times3 + 3\times4$

$= 20$

Page No.
Date

$$y(2) = \sum_{k=0}^{3} x(k)\,x(k-2)$$

$$= x(0)\,x(-2) + x(1)\,x(-1) + x(2)\,x(0) + x(3)\,x(1) \cancel{+x}$$

$$= 1\times 3 + 4\times 2 = 11$$

$$y(3) = \sum_{k=0}^{3} x(k)\,x(k-3)$$

$$= x(0)\,x(-3) + x(1)\,x(-2) + x(2)\,x(-1) + x(3)\,x(0)$$

$$= 4\times 1 = 4$$

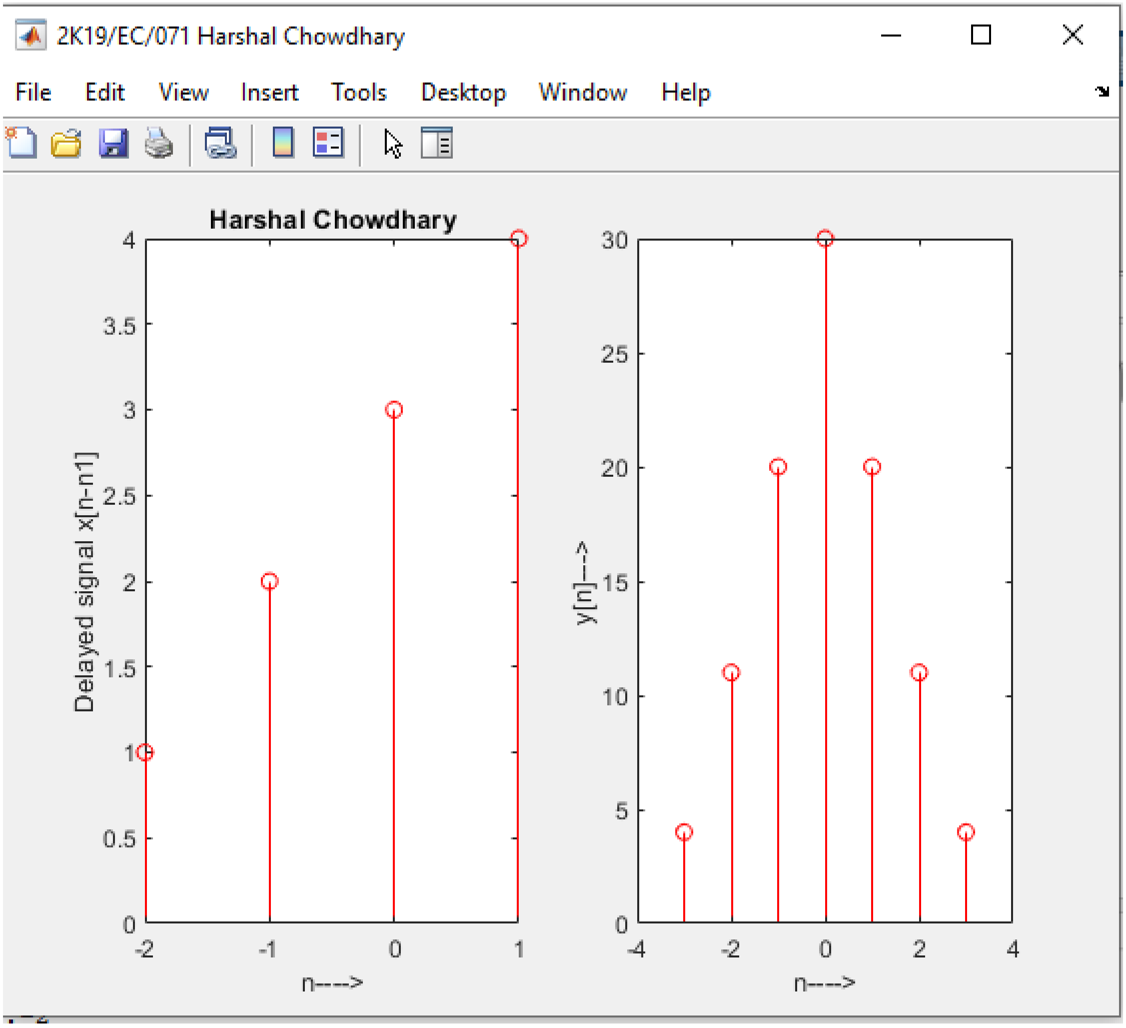
Code (Auto-Correlation for Non-Casual Signal):

```
clc;
clear all;
close all;

%input
x=input('Please enter the sequence:');
n=1-length(x):length(x)-1;

%starting point
n1=input('Enter the starting point:');

%correlation
y=xcorr(x,x)

figure('Name','2K19/EC/071 Harshal Chowdhary','NumberTitle','off');
subplot(1,2,1);
stem(n1:1:n1+length(x)-1,x,'r');
xlabel('n---->');
ylabel('Delayed signal x[n-n1]');
title('Harshal Chowdhary');

subplot(1,2,2);
stem(n,y,'r');
xlabel('n---->');
ylabel('y[n]--->');
```

Input:

```
Please enter the sequence:[1 2 3 4]
Enter the starting point:-2

y =

    4.0000   11.0000   20.0000   30.0000   20.0000   11.0000    4.0000
```

Output:



2K19/EC/071 Harshal Chowdhary
File Edit View Insert Tools Desktop Window Help
Harshal Chowdhary
Delayed signal x[n-n1]
n---->
y[n]--->
n---->
4
3.5
3
2.5
2
1.5
1
0.5
0
-2
-1
0
1
30
25
20
15
10
5
0
-4
-2
0
2
4

Manual Solution:

$\longrightarrow$ $x[n] = [\underset{-2}{1} \quad 2 \quad \underset{\uparrow}{3} \quad 4]$

Page No.
Date

$x[n] = [\underset{-2}{1} \quad 2 \quad \underset{\uparrow}{3} \quad 4]$

starting point of signal $= -2 + (-2+4-1)$
$= -3$

ending point of signal $= -3 + (4+4-2)$
$= 3$

Range of the signal would be -3 to 3

$$y[-3] = \sum_{k=-2}^{1} x[k]\,x[k+3]$$
$$= 1\times 4 = \underline{4}$$

$$y[-2] = \sum_{k=-2}^{1} x[k]\,x[k+2]$$
$$= 1\times 3 + 2\times 4 = \underline{11}$$

$$y[-1] = \sum_{k=-2}^{1} x[k]\,x[k+1]$$
$$= 1\times 2 + 2\times 3 + 3\times 4 = \underline{20}$$

$$y[0] = \sum_{k=-2}^{1} x[k]\,x[k]$$
$$= 1\times 1 + 2\times 2 + 3\times 3 + 4\times 4$$
$$= \underline{30}$$

$$y[1] = \sum_{k=-2}^{1} x[k]\,x[k-1]$$
$$= 2\times 1 + 3\times 2 + 4\times 3 = \underline{20}$$

$$y[2] = \sum_{k=-2}^{1} x[k]\,x[k-2]$$
$$= 3\times 1 + 4\times 2 = \underline{11}$$

$$y[3] = \sum_{k=-2}^{1} x[k]\,x[k-3]$$
$$= 4\times 1 = \underline{4}$$

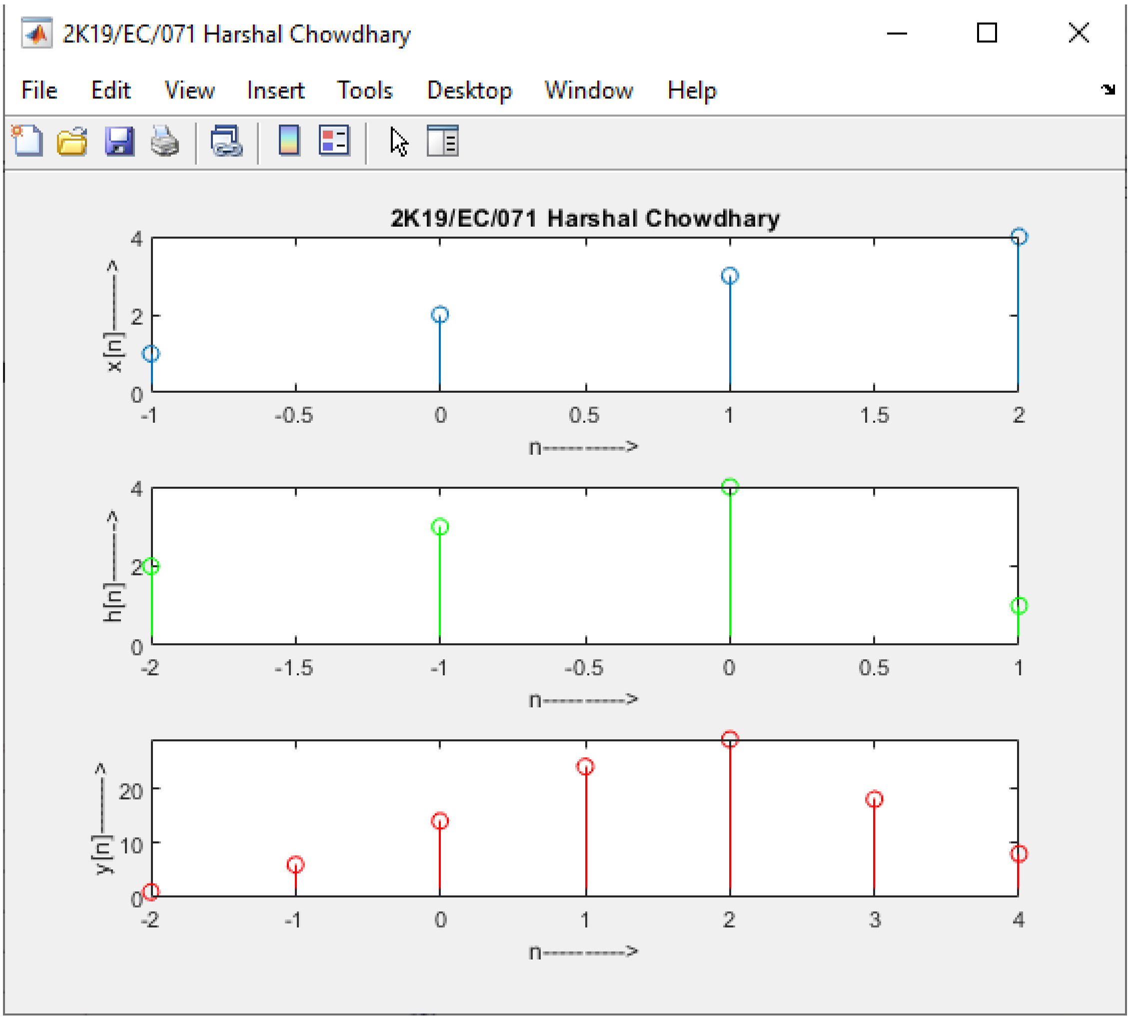
Code (Cross-Correlation):

```
clc;
clear all;
close all;

x=input('Please Enter The First sequence: ');
sx=input('Enter The starting point: ');
h=input('Please Enter The Second sequence: ');
sh=input('Enter The starting point: ');
sn=sx-(sh+length(h)-1);
se=sh+(length(x)+length(h)-2);

y=xcorr(x,h)

figure('Name','2K19/EC/071 Harshal Chowdhary','NumberTitle','off');

subplot(3,1,1);
stem(sx:sx+length(x)-1,x);
title('2K19/EC/071 Harshal Chowdhary');
xlabel('n---------->');
ylabel('x[n]------->');

subplot(3,1,2);
stem(sh:sh+length(h)-1,h,'g-');
xlabel('n---------->');
ylabel('h[n]------->');

subplot(3,1,3);
stem(sh:se,y,'r-');
xlabel('n---------->');
ylabel('y[n]------->');
```

Input:

```
Please Enter The First sequence: [1 2 3 4]
Enter The starting point: -1
Please Enter The Second sequence: [2 3 4 1]
Enter The starting point: -2

y =

    1.0000    6.0000   14.0000   24.0000   29.0000   18.0000    8.0000
```

Output:


2K19/EC/071 Harshal Chowdhary
File Edit View Insert Tools Desktop Window Help
2K19/EC/071 Harshal Chowdhary
x[n]------->
n------------>
h[n]------->
n------------>
y[n]------->
n------------>

Manual Solution:

(3) Cross Correlation

$x[n] = [1;\ \overset{\downarrow}{2}\ \ 3\ \ 4]$, starting at -1

$h[n] = [2\ \ 3\ \ \underset{\uparrow}{4}\ \ 1]$, starting at -2

Starting point of final signal $= (-1) - (-2+4-1)$
$= -2$

ending point of final signal $= -2 + (4+4-2)$
$= 4$

Range would be $= -2$ to 4

$$y[n] = \sum_{-\infty}^{\infty} x[k]\, h[k-n]$$

$$y[-2] = \sum_{k=-2}^{1} x[k]\, h[k+2] = 1\times 1 = 1$$

$$y[-1] = \sum_{k=-2}^{1} x[k]\, h[k+1] = 1\times 4 + 2\times 1 = 6$$

$$y[0] = \sum_{k=-2}^{1} x[k]\, h[k] = 1\times 3 + 2\times 4 + 3\times 1 = 14$$

$$y[1] = \sum_{k=-2}^{1} x[k]\, h[k-1] = 1\times 2 + 2\times 3 + 3\times 4 + 4\times 1 = 24$$

$$y[2] = \sum_{k=-2}^{1} x[k]\, h[k-2] = 2\times 2 + 3\times 3 + 4\times 4 = 29$$

$$y[3] = \sum_{k=-2}^{1} x[k]\, h[k-3] = 3\times 2 + 4\times 3 = 18$$

$$y[4] = \sum_{k=-2}^{1} x[k]\, h[k-4] = 4\times 2 = 8$$

Learning Outcome:

Grasps the basic knowledge of Correlation of Signals and made code for both casual and non-casual signals. Also verified the solution by manually performing correlation.

Experiment-4

Aim

To study the frequency response of a first order system

Theory

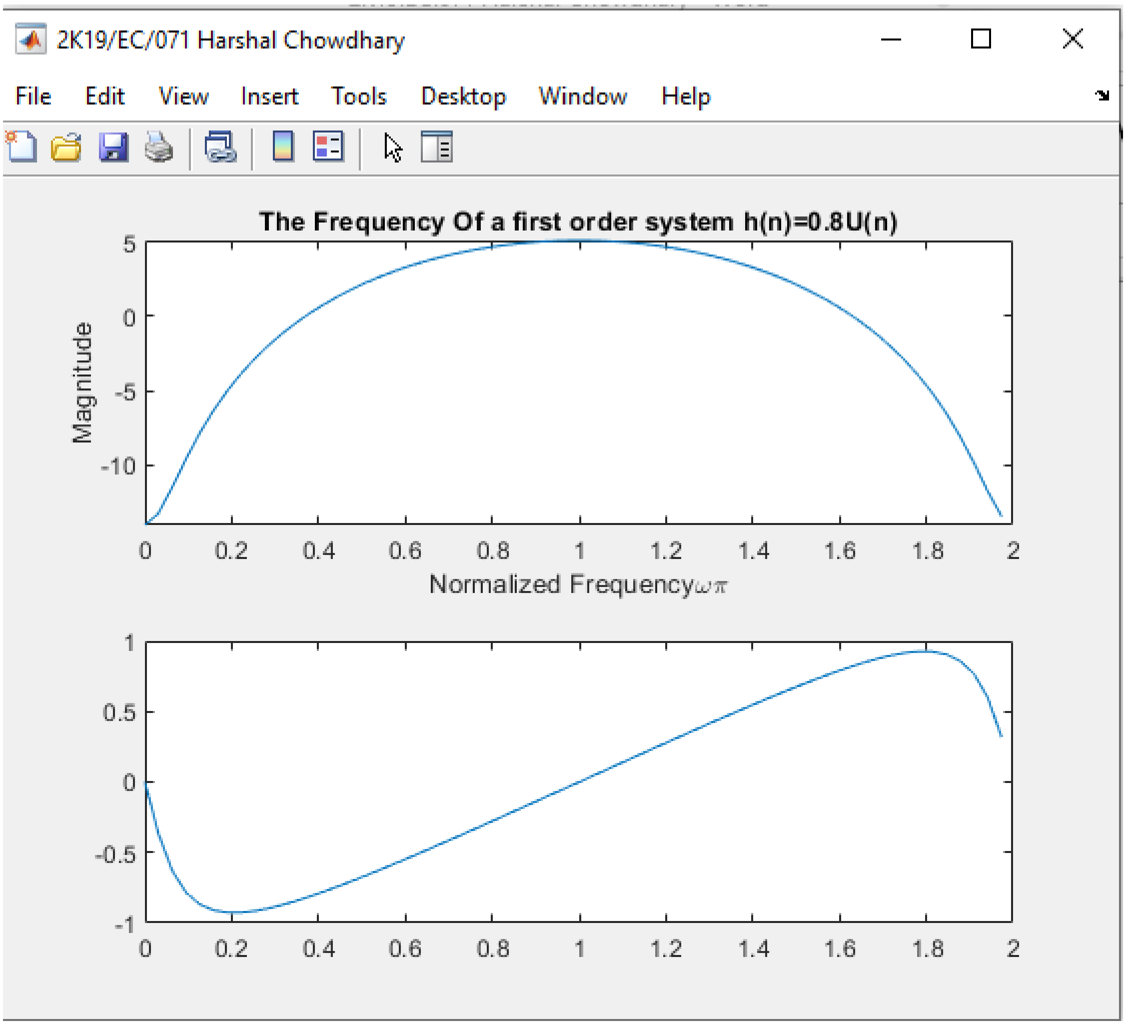
A first order system or network is one that contains but a single energy storage element such as an inductor or capacitor. Each of these elements, either singly or in combination and in association with resistors, may be arranged in series, parallel, series-parallel or parallel-series.

Whatever the arrangement, the differential equation that governs the behaviour of the network is of first order.

Code:

```
clc;
clear all;
close all;

b=[1];
a=[1 -0.8];
w=0:0.1:2*pi;
h=freqz(b,a,w);

figure('Name','2K19/EC/071 Harshal Chowdhary','NumberTitle','off');

subplot(2,1,1);
plot(w/pi,-20*log10(abs(h)));
xlabel('Normalized Frequency\omega\pi');
ylabel('Magnitude');
title('The Frequency Of a first order system h(n)=0.8U(n)');

subplot(2,1,2);
plot(w/pi,angle(h));
```

Output:


2K19/EC/071 Harshal Chowdhary
File Edit View Insert Tools Desktop Window Help
The Frequency Of a first order system h(n)=0.8U(n)
Magnitude
5
0
-5
-10
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2
Normalized Frequencyωπ
1
0.5
0
-0.5
-1
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2

Experiment – 5

Aim:

Generate a MATLAB code to perform Basic Signal Operation: time scaling, time reversing, even & odd part of a signal, time shifting, amplitude scaling

Theory:

The basic signal operations are categorized into two types depending on whether they operated on dependent or independent variable(s) representing the signals.

Time Shifting - Suppose that we have a signal $x(t)$ and we define a new signal by adding/subtracting a finite time value to/from it. We now have a new signal, $y(t)$. The mathematical expression for this would be $x(t \pm t_0)$. **Time Reversal** - t can be negative. In fact, one can make it negative just by multiplying it by -1. This causes the original signal to flip along its y-axis. That is, it results in the reflection of the signal along its vertical axis of reference. As a result, the operation is aptly known as the time reversal or time reflection of the signal. **Time Scaling** means that, if we multiply the time variable by a factor of 2, then we will get our output signal contracted by a factor of 2 along the time axis. Thus, it can be concluded that the multiplication of the signal by a factor of n leads to the compression of the signal by an equivalent factor.

Code (For Time Reversal, Even or Odd Function):

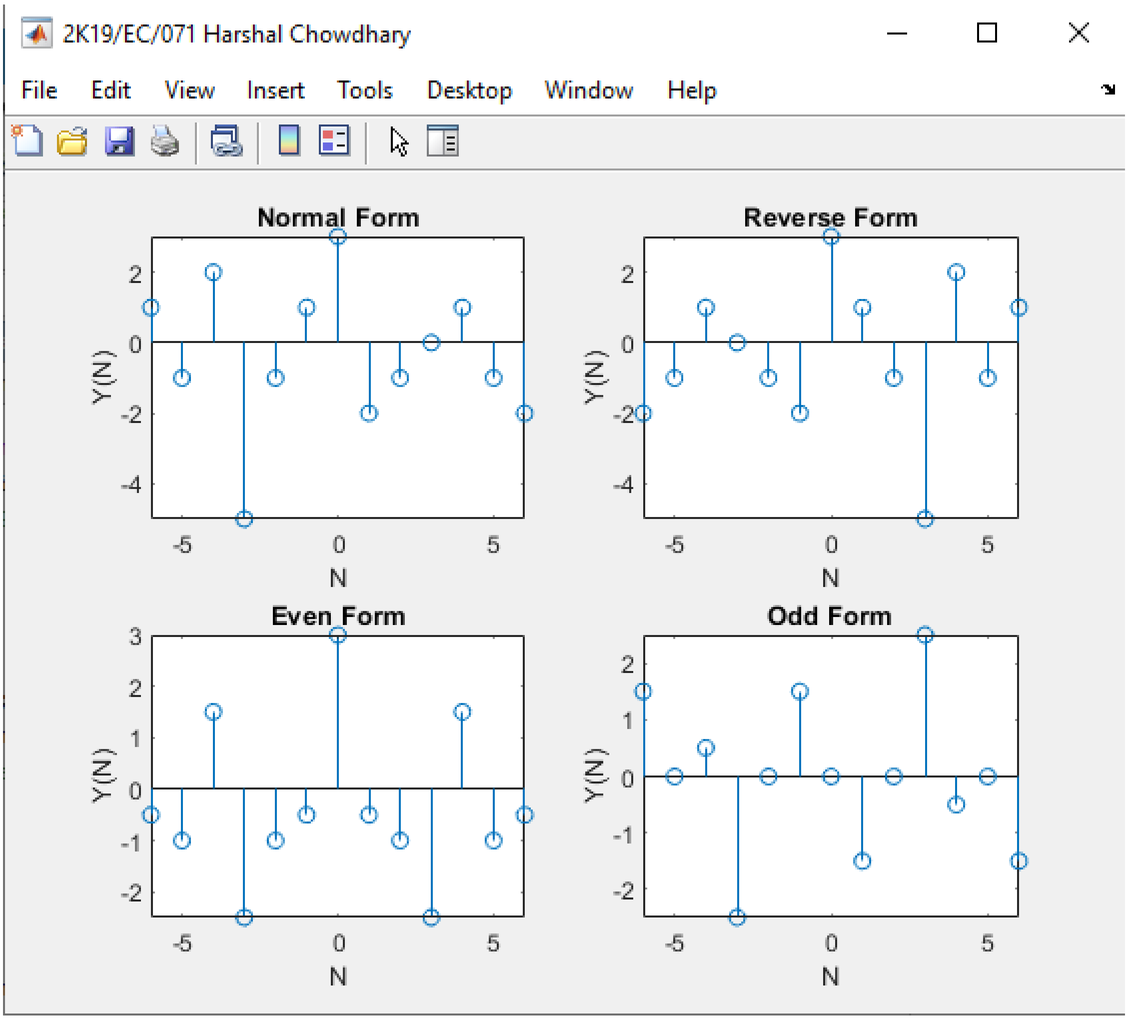
```
clc;
close all;
clear all;
%Setting X-axis Limit
Nmin=-6;
Nmax=6;
N=Nmin:1:Nmax;
%Calling the function
Yo=Yd(N)
%Reversing Of Function
Yr=Yo(end:-1:1)
%Even Of a Function
Ye=(Yo+Yr)/2
%Odd of a function
YO=(Yo-Yr)/2
Ymax=max([max(Yo),max(Yr),max(YO),max(Ye)]);
Ymin=min([min(Yo),min(Yr),min(YO),min(Ye)]);
figure('Name','2K19/EC/071 Harshal Chowdhary','NumberTitle','off');
```

```
%Plotting Of Normal FORM
subplot(2,2,1);
stem(N,Yo);
xlabel("N");
ylabel("Y(N)");
title('Normal Form');
axis([Nmin, Nmax, Ymin, Ymax]);
%pLOTTING oF REVERSE OF SEQUENCE
subplot(2,2,2);
stem(N,Yr);
xlabel("N");
ylabel("Y(N)");
title('Reverse Form');
axis([Nmin, Nmax, Ymin, Ymax]);
%Plotting Of Even Part of function
subplot(2,2,3);
stem(N,Ye);
xlabel("N");
ylabel("Y(N)");
title('Even Form');
axis([Nmin, Nmax, min(Ye),max(Ye)]);
%Plotting Of Odd Part Of Function
subplot(2,2,4);
stem(N,YO);
xlabel("N");
ylabel("Y(N)");
title('Odd Form');
axis([Nmin, Nmax, min(YO), max(YO)]);

function X=Yd(N)
X=[1,-1,2,-5,-1,1,3,-2,-1,0,1,-1,-2]
end
```

Command Window:

```
X =

     1    -1     2    -5    -1     1     3    -2    -1     0     1    -1    -2


Yo =

     1    -1     2    -5    -1     1     3    -2    -1     0     1    -1    -2


Yr =

    -2    -1     1     0    -1    -2     3     1    -1    -5     2    -1     1


Ye =

   -0.5000   -1.0000    1.5000   -2.5000   -1.0000   -0.5000    3.0000   -0.5000   -1.0000   -2.5000    1.5000   -1.0000   -0.5000


YO =

    1.5000         0    0.5000   -2.5000         0    1.5000         0   -1.5000         0    2.5000   -0.5000         0   -1.5000
```

Output:



Code (For time Scaling, Amplitude Scaling ,Time Shifting):

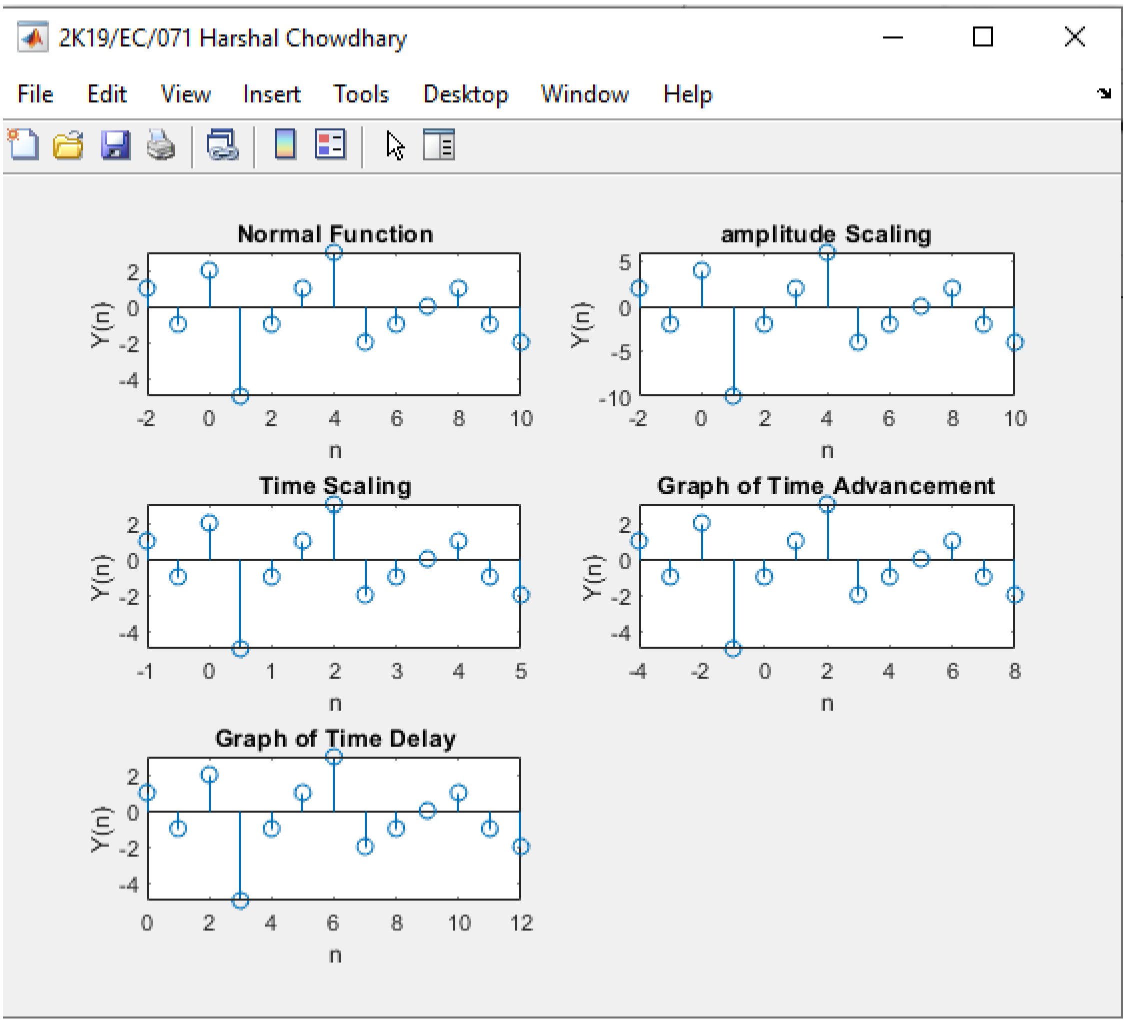
```
clc;
close all;
clear all;

Nmin=input('Enter The Starting Of the sequence: ');
Nmax=Nmin+12;
a=input('By what Factor Do you what to change function: ');
N=Nmin:Nmax;
Yo=Yd(N)
Ya=a*Yd(N)
Yomax=max(Yo);
Yomin=min(Yo);
Yamax=max(Ya);
Yamin=min(Ya);
figure('Name','2K19/EC/071 Harshal Chowdhary','NumberTitle','off');
subplot(3,2,1);
stem(N,Yo);
xlabel("n");
ylabel("Y(n)");
title('Normal Function');
axis([Nmin, Nmax, Yomin, Yomax]);
```

```
subplot(3,2,2);
stem(N,Ya);
xlabel("n");
ylabel("Y(n)");
title('amplitude Scaling');
axis([Nmin, Nmax, Yamin, Yamax]);

subplot(3,2,3);
stem((Nmin/a):1/a:(Nmax/a),Yo);
xlabel("n");
ylabel("Y(n)");
title('Time Scaling');
axis([(Nmin/a), (Nmax/a), Yomin, Yomax]);

subplot(3,2,4);
stem(Nmin-a:1:Nmax-a,Yo);
xlabel("n");
ylabel("Y(n)");
title('Graph of Time Advancement');
axis([(Nmin-a), (Nmax-a), Yomin, Yomax]);

subplot(3,2,5);
stem(Nmin+a:1:Nmax+a,Yo);
xlabel("n");
ylabel("Y(n)");
title('Graph of Time Delay');
axis([(Nmin+a), (Nmax+a), Yomin, Yomax]);

function X=Yd(N)
X=[1,-1,2,-5,-1,1,3,-2,-1,0,1,-1,-2]
end
```

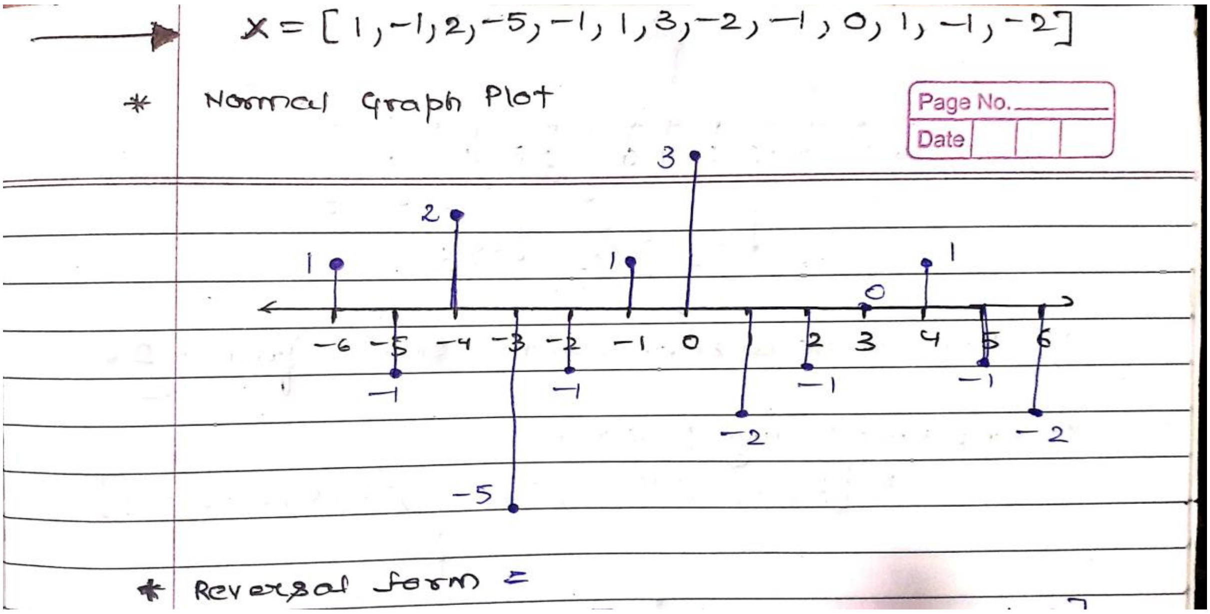
Command Window:

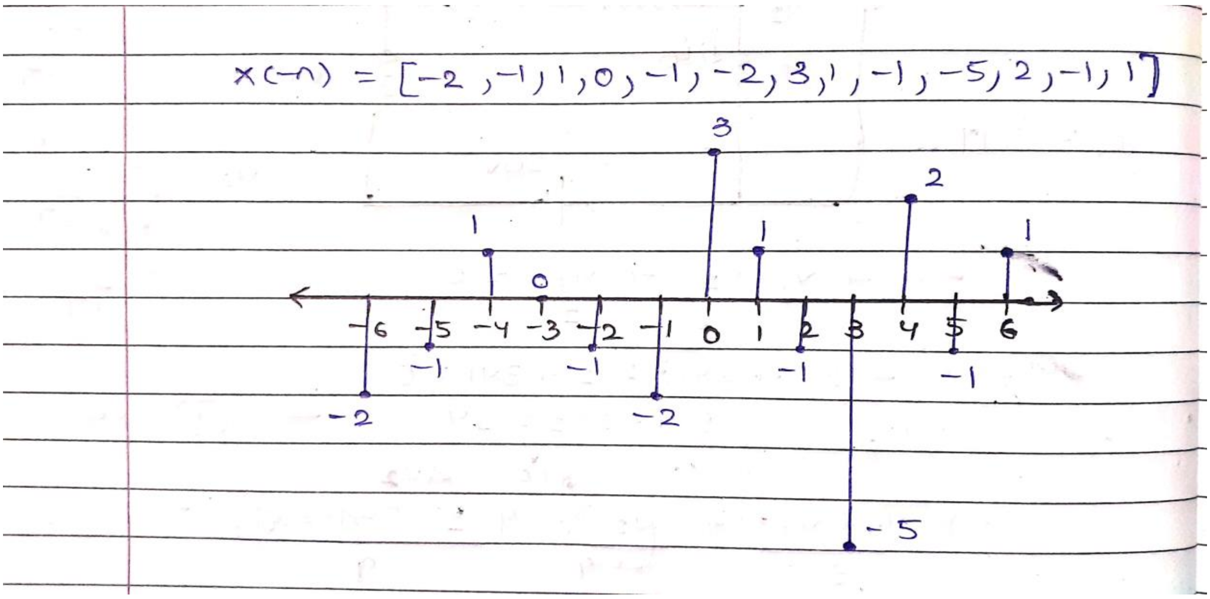
```
Enter The Starting Of the sequence: -2
By what Factor Do you what to change function: 2

X =

     1    -1     2    -5    -1     1     3    -2    -1     0     1    -1    -2


Yo =

     1    -1     2    -5    -1     1     3    -2    -1     0     1    -1    -2


Ya =

     2    -2     4   -10    -2     2     6    -4    -2     0     2    -2    -4
```

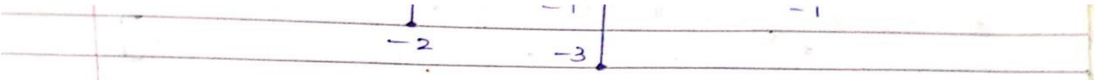
Output:


2K19/EC/071 Harshal Chowdhary
File Edit View Insert Tools Desktop Window Help
Normal Function
amplitude Scaling
Time Scaling
Graph of Time Advancement
Graph of Time Delay
Y(n)
n

Manual Solution:

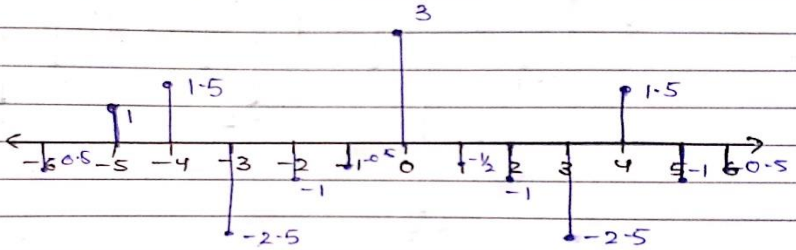






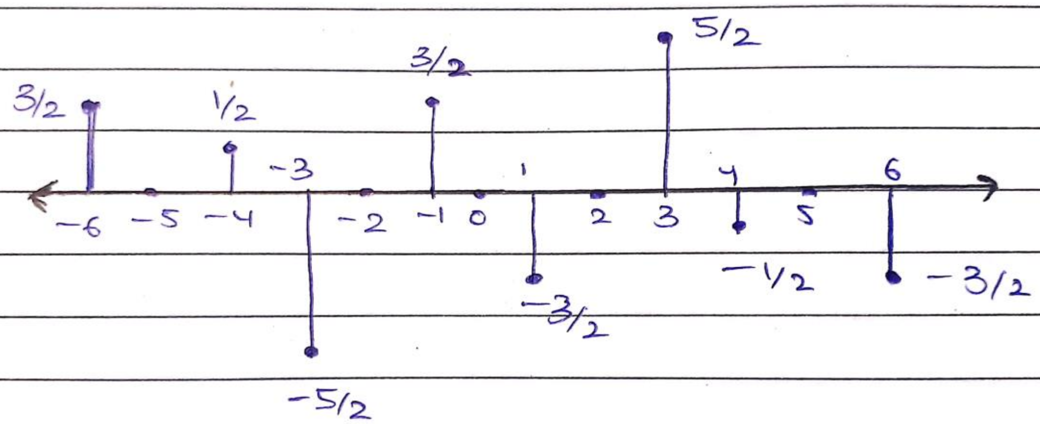
* Even form

$x_e = [-0.5, -1, 1.5, -2.5, -1, -0.5, 3, -0.5, -1, -2.5, 1.5, -1, -0.5]$



* Odd form

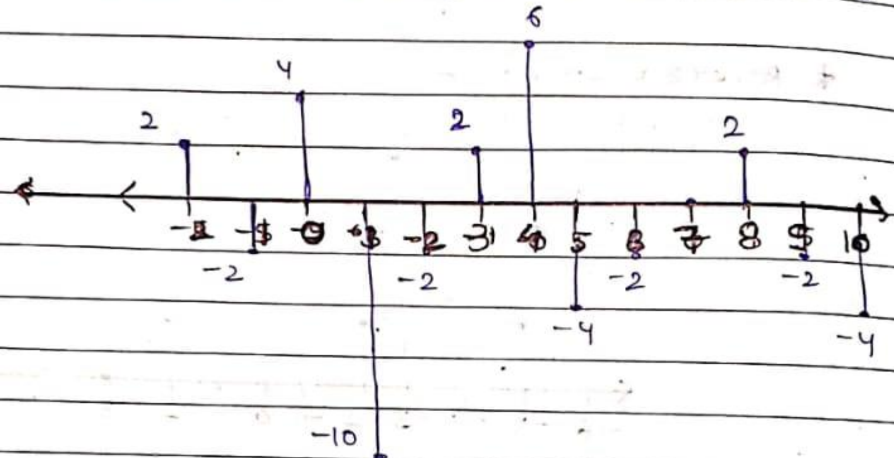
$x_o = [3/2, 0, 1/2, -5/2, 0, 3/2, 0, -3/2, 0, \cancel{3}\,5/2, -1/2, 0, -3/2]$



→ Now ~~the~~ we consider function starts from -2.

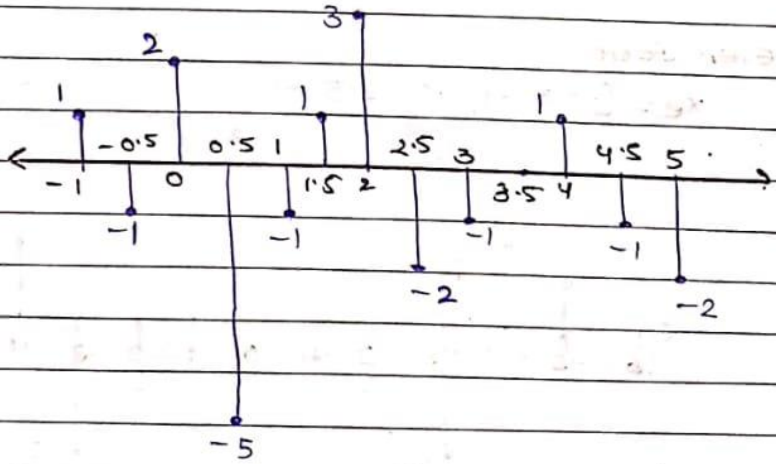
∵ Amplitude Scaling : $2x(t)$

$= [2, -2, 4, -10, -2, 2, 6, -4, -2, 0, 2, -2, -4]$



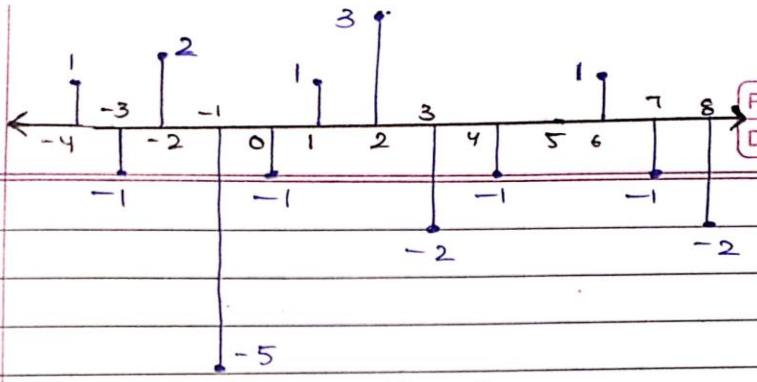
∵ Time scaling : $x(2t)$ Time change from 0.5 to 5

$= [1, -1, 2, -5, -1, 1, 3, -2, -1, 0, 1, -1, -2]$



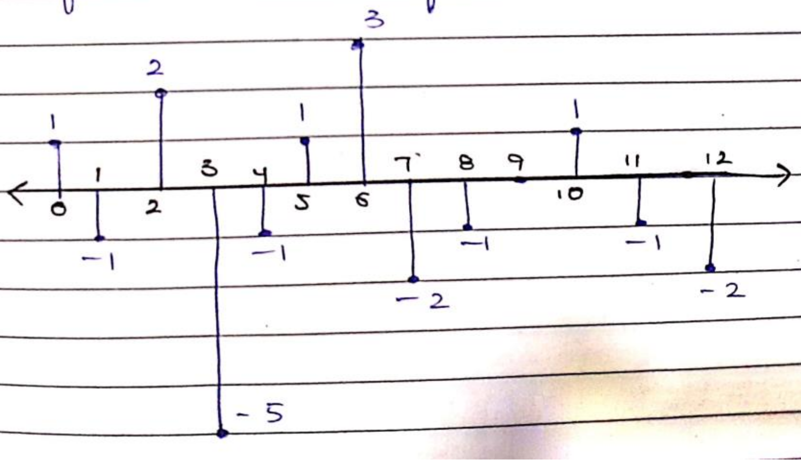
∵ Time shifting $x(t+2)$

Now the function would shift from -2 to -4



Time shifting : $x(t-2)$

The function would shift to 0 to 12



Experiment-6

Aim

Plot the Square Wave and hence verify Gibb's phenomenon using first 10 terms of Fourier Series

Theory

Fourier Series is representation of periodic signals. Fourier series can be represented into three ways:

1) **Trigonometric form**
2) **Cosine form**
3) **Exponential form**

The function x(t) must be a single valued function. The function x(t) has only a finite number of maxima and minima. The function x(t) has finite number of discontinuities.

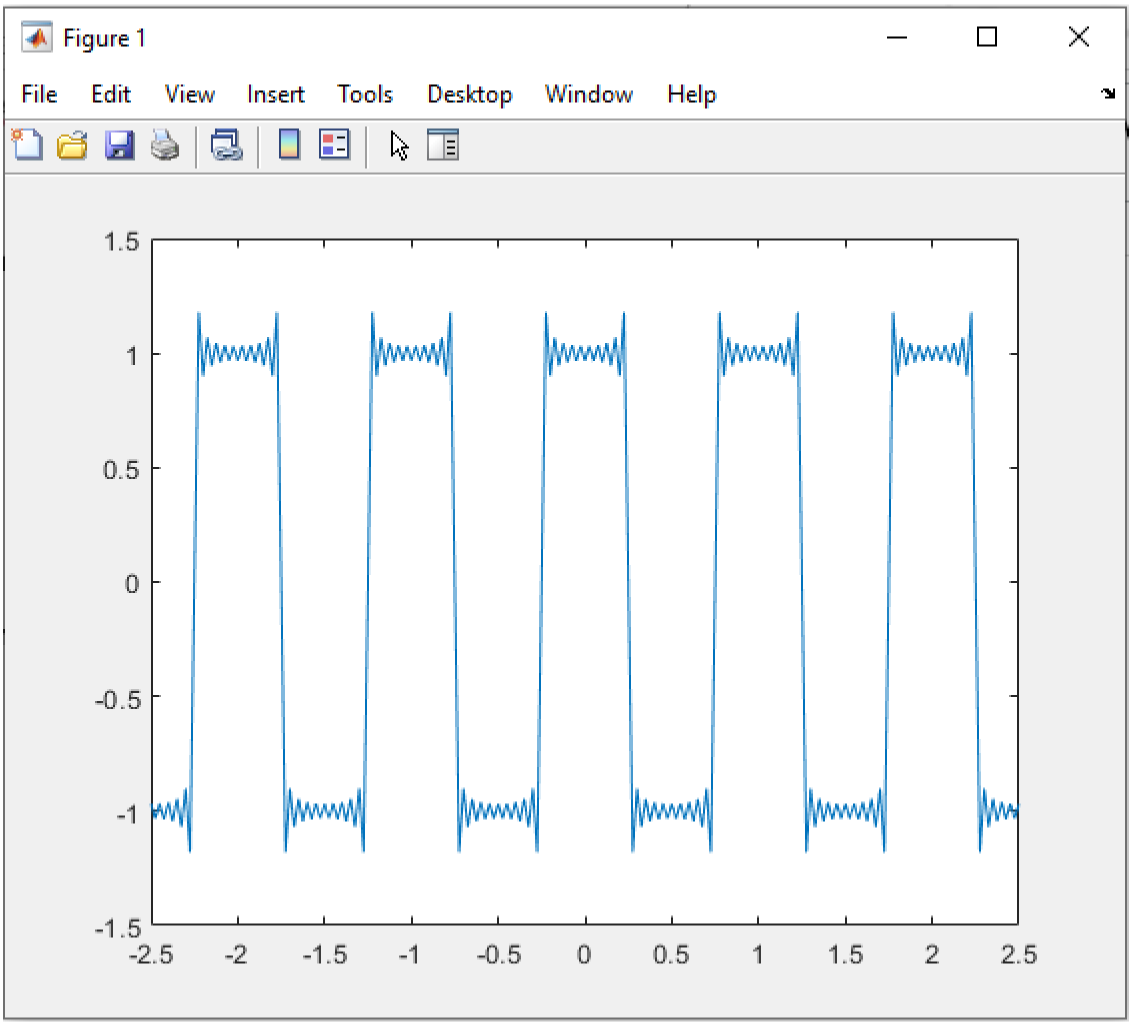
Code (For Even Function):

```
clc;
clear all;
close all;

T=input('Enter the period of Square Wave: ');
n1=input('Enter the No. of cycles to be plotted: ');
n=input('Enter the harmonics to be present: ');

wo=((2*pi)/T);
N=1:1:2*n-1;
syms t;
ao=(1/T)*(int(1,t,0,T/4)+int(-1,t,T/4,(3*T)/4)+int(1,t,(3*T)/4,T))

an=(2/T)*(int(1*cos(N*wo*t),t,0,T/4)+int(-1*cos(N*wo*t),t,T/4,(3*T)/4)+int(1*cos(N*wo*t),t,(3*T)/4,T))

bn=(2/T)*(int(1*sin(N*wo*t),t,0,T/4)+int(-1*sin(N*wo*t),t,T/4,(3*T)/4)+int(1*sin(N*wo*t),t,(3*T)/4,T))
t=-n1/2:n1/200:n1/2;
fn=0;
for i=1:2:2*n-1
    fn=fn+(an(i)*cos(i*wo*t))+(bn(i)*sin(i*wo*t));
    final=fn+ao;
    plot(t,final)
end
```

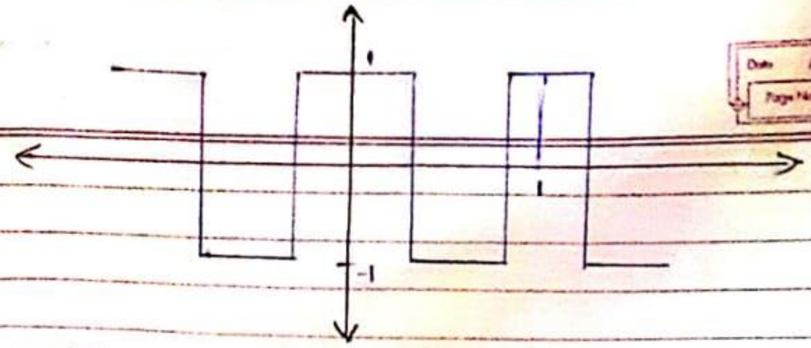
Command Window:

```
Enter the period of Square Wave: 1
Enter the No. of cycles to be plotted: 5
Enter the harmonics to be present: 10

ao =

0


an =

[ 4/pi, 0, -4/(3*pi), 0, 4/(5*pi), 0, -4/(7*pi), 0, 4/(9*pi), 0, -4/(11*pi), 0, 4/(13*pi), 0, -4/(15*pi), 0, 4/(17*pi), 0, -4/(19*pi)]


bn =

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Output:



Manual Solution:



$\omega_0 = 2\pi \qquad T = 1$

$$x(t) = \begin{cases} 1 & t \in [0, 1/4] \cup [3/4, 1] \\ -1 & t \in [1/4, 3/4] \end{cases}$$

This is an even symmetry so $b_n = 0$

$$a_0 = \frac{1}{T}\int_0^T x(t)\,dt$$

$$= \frac{1}{1}\left[\int_0^{1/4} dt + \int_{1/4}^{3/4} -1\,dt + \int_{3/4}^{1} dt\right]$$

$$= [1/4 - 0] + (1/4 - 3/4] + [1 - 3/4]$$

$$a_0 = 0$$

$$a_n = \frac{2}{T}\left[\int_0^T x(t)\cos n\omega t\,dt\right]$$

$$= \frac{2}{1}\left[\int_0^{1/4} \cos n2\pi t\,dt + \int_{1/4}^{3/4} -\cos n2\pi t\,dt + \int_{3/4}^{1} \cos 2\pi nt\,dt\right]$$

$$= 2\left[\frac{\sin 2\pi nt}{2\pi n}\Big|_0^{1/4} - \frac{\sin 2\pi nt}{2\pi n}\Big|_{1/4}^{3/4} + \frac{\sin 2\pi nt}{2\pi n}\Big|_{3/4}^{1}\right]$$

$$= \frac{2}{\pi n}\left[\sin\left(\frac{\pi n}{2}\right) - \sin\left(\frac{3\pi n}{2}\right)\right]$$

for $n = 1$; $a_1 = \frac{4}{\pi}$

$a_2 = a_4 = a_6 = a_8 = 0$

For $n=3$ $\quad a_3 = \frac{-4}{3\pi}$

For $n=5$ $\quad a_5 = \frac{4}{5\pi}$

Fouriere series can be represented as :-

$$= \frac{4}{\pi}(\cos 2\pi t) - \frac{4}{3\pi}\cos(6\pi t) + \frac{4}{5\pi}\cos(10\pi t)$$

$$= \sum_{n=\text{odd}} \frac{4}{n\pi}\left[\cos\left(2\pi n t - \pi\left[\frac{n}{2}\right]\right)\right]$$

$$a_3 = \frac{4}{3\pi}\left[\cos(2\pi n t - \pi)\right]$$

$$a_5 = \frac{4}{5\pi}\left[\cos(2\pi n t - 2\pi)\right]$$

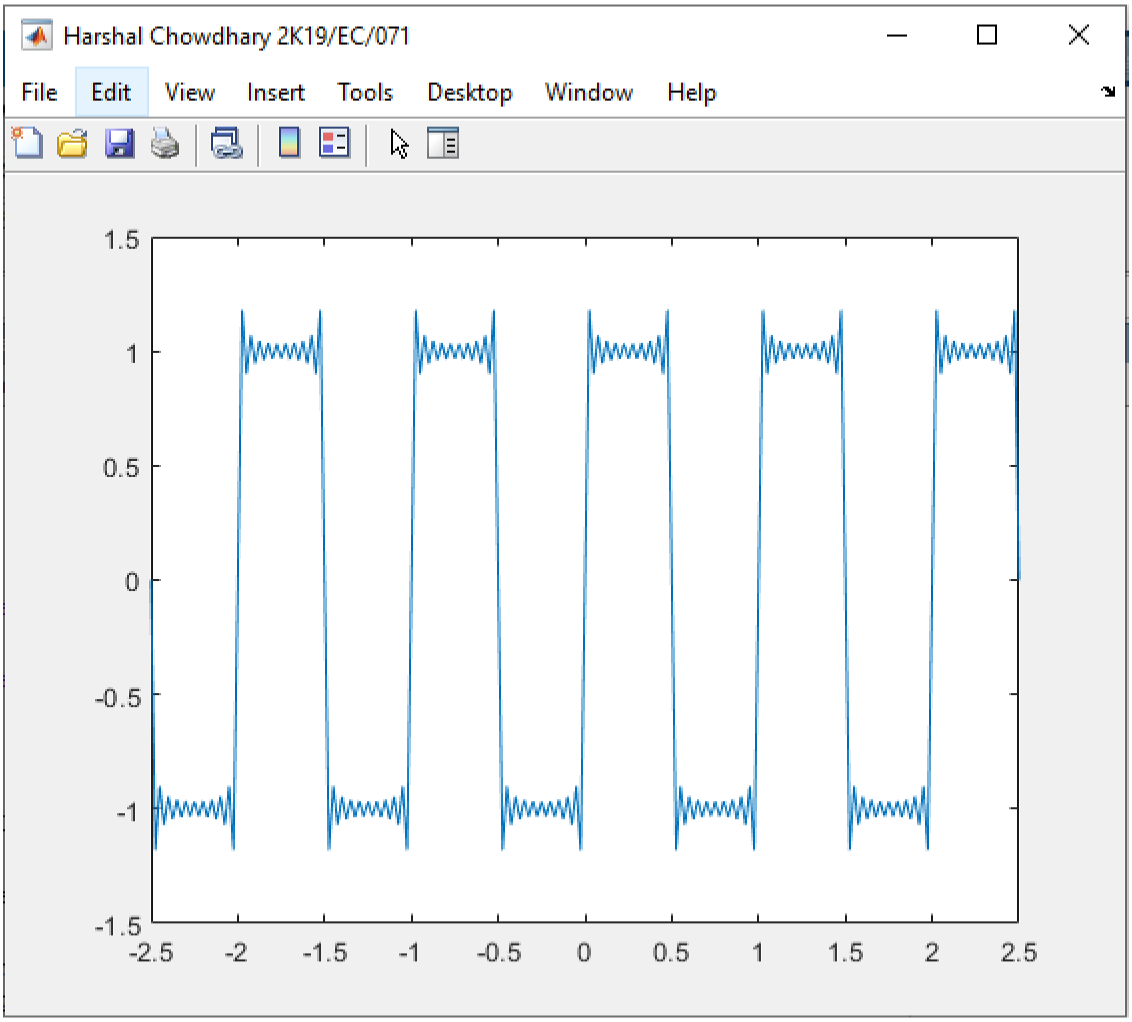
Code (For Odd Function):

```
clc;
clear all;
close all;

T=input('Enter the period of Square Wave: ');
n1=input('Enter the No. of cycles to be plotted: ');
n=input('Enter the harmonics to be present: ');
figure('Name','Harshal Chowdhary 2K19/EC/071','NumberTitle','off');
wo=((2*pi)/T);
N=1:1:2*n-1;
syms t;
ao=(1/T)*(int(1,t,0,T/2)+int(-1,t,T/2,T))

an=(2/T)*(int(1*cos(N*wo*t),t,0,T/2)+int(-1*cos(N*wo*t),t,T/2,T))

bn=(2/T)*(int(1*sin(N*wo*t),t,0,T/2)+int(-1*sin(N*wo*t),t,T/2,T))
t=-n1/2:n1/200:n1/2;
fn=0;
for i=1:2:2*n-1
    fn=fn+(an(i)*cos(i*wo*t))+(bn(i)*sin(i*wo*t));
    final=fn+ao;
    plot(t,final)
end
```

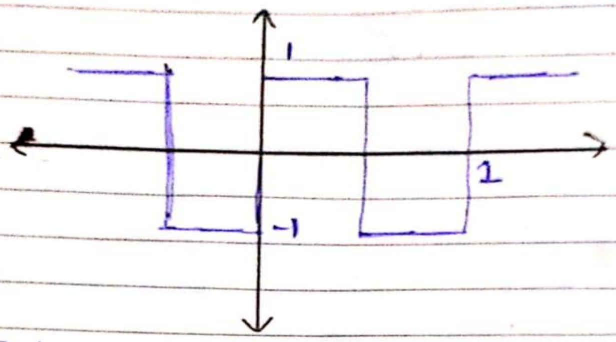
Command Window:

```
Enter the period of Square Wave: 1
Enter the No. of cycles to be plotted: 5
Enter the harmonics to be present: 10

ao =

0


an =

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]


bn =

[ 4/pi, 0, 4/(3*pi), 0, 4/(5*pi), 0, 4/(7*pi), 0, 4/(9*pi), 0, 4/(11*pi), 0, 4/(13*pi), 0, 4/(15*pi), 0, 4/(17*pi), 0, 4/(19*pi)]
```

Output:



Manual Solution:



$T = 1 \qquad \omega = 2\pi$

$$x(t) = \begin{cases} 1 & t \in [0, 1/2] \\ -1 & t \in [1/2, 1] \end{cases}$$

This is an odd symmetry $\Rightarrow a_n = 0$

$$a_0 = \frac{1}{T}\int_0^T x(t)\,dt = \frac{1}{1}\int_0^1 x(t)\,dt$$

$$= \int_0^{1/2} 1\,dt + \int_{1/2}^1 -1\,dt = 0$$

$$b_n = \frac{2}{T}\int_0^T x(t)\sin n\omega t\,dt$$

$$= 2\int_0^1 x(t)\sin 2\pi n t\,dt$$

$$= 2\int_0^{1/2} \sin 2\pi n t\,dt + \int_{1/2}^1 -\sin 2\pi n t\,dt$$

$$= 2\left[-\frac{\cos 2\pi n t}{2\pi n}\Big|_0^{1/2} + \frac{\cos 2\pi n t}{2\pi n}\Big|_{1/2}^1 \right]$$

$$= \frac{2}{\pi n}\left[\cos 2\pi n - \cos \pi n\right]$$

$b_1 = \frac{4}{\pi} \qquad b_2 = b_4 = b_6 = 0$

$b_3 = \frac{4}{3\pi} \qquad b_5 = \frac{4}{5\pi}$

The fourier series is represented as follows:-

$$\sum_{n=odd} \frac{4}{n\pi}\left[\sin(2\pi n t)\right]$$

Experiment-7

Aim

Generate a MATLAB code to find the partial fractional and also find the impulse and step response

Theory

The impulse response is an especially important property of any LTI system. We can use it to describe an LTI system and predict its output for any input. It has many important applications in sampling. The unit impulse signal is simply a signal that produces a signal of 1 at time = 0. It is zero everywhere else. With that in mind, an LTI system's impulse function is defined as follows:

The impulse response for an LTI system is the output, $y(t)$, when the input is the unit impulse signal, $\sigma(t)$. In other words,

$$x(t) = \sigma(t), \quad h(t) = y(t).$$

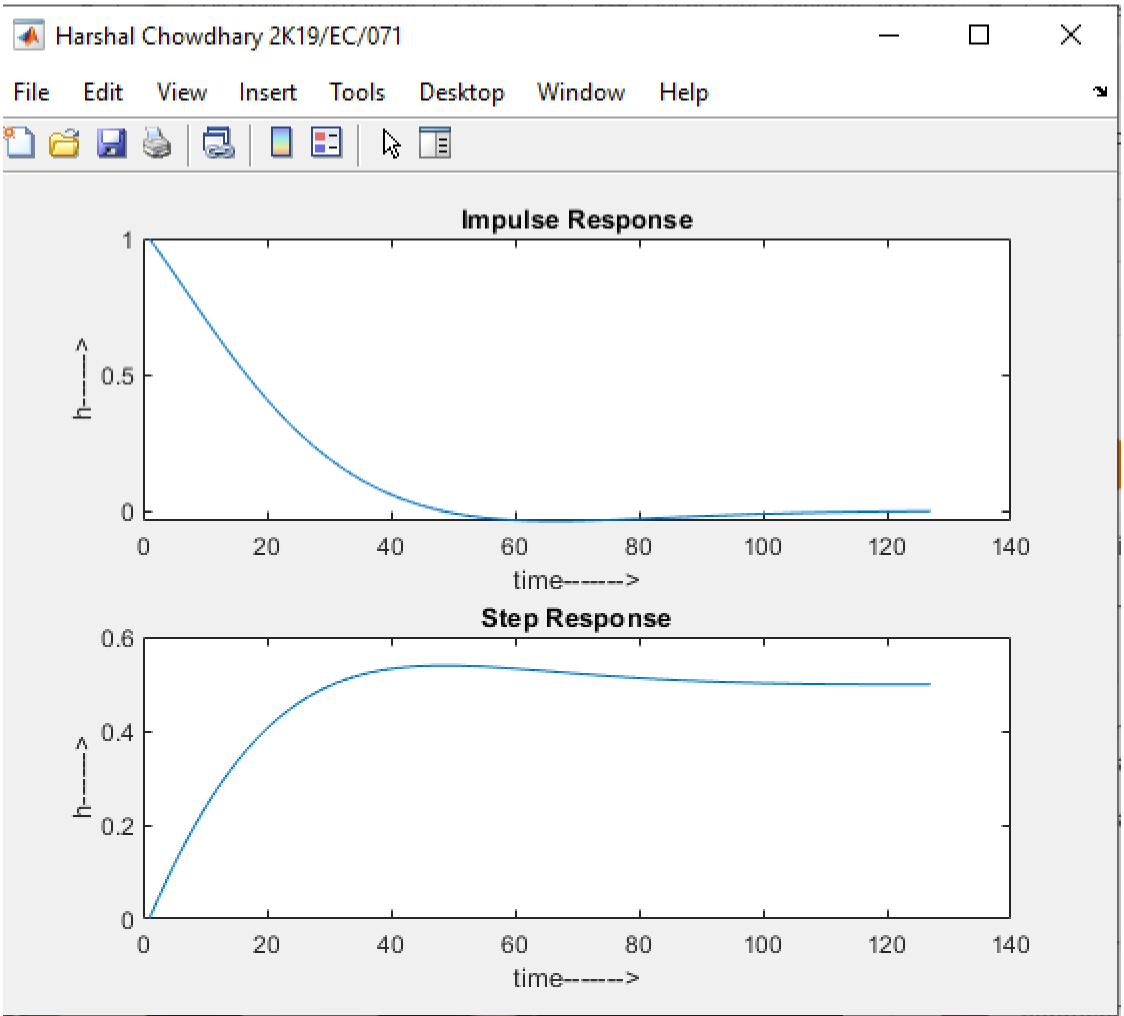
Code

```
clc;
clear all;
close all;

b=input('Enter the num coeff: ');
a=input('Enter the den coeff: ');
[r,p,k] = residue(b,a)
[r,p,k] = residuez(b,a)

figure('Name','Harshal Chowdhary 2K19/EC/071','NumberTitle','off');
[H,T]=impulse(b,a);
subplot(2,1,1);
xlabel('time------->');
ylabel('h------>');
title('Impulse Response');
plot (H);
[E]=step(b,a);
subplot(2,1,2);
xlabel('time------->');
ylabel('h------>');
title('Step Response');
plot (E);
```

Command Window:

```
Enter the numerator coeff:[1 2]

b =

     1     2

Enter the denominator coeff:[1 3 4]

a =

     1     3     4
```

Output:



Experiment-8

Aim

Generate a MATLAB code to find sinusoidal/exponential response of a system.

Theory

The transfer function of an LTI system is given by the Laplace transform of the impulse response of the system and it gives valuable information of the system's behavior and can greatly simplify the computation of the output response.

If the impulse response of a system $y(t)$ is given by $h(t)$ then the transfer function of that system is given by

$$H(S)=L(h(t))$$

We know that the output of an LTI system will be given by the convolution of the signal with the impulse response. Since the convolution in the time domain is equivalent to a multiplication in the Laplace domain, the output $Y(S)$ of a system with the transfer function $H(S)$ to the input $X(S)$ will be given by:

$$Y(S)=H(S)X(S)$$

One can easily calculate the output in the time domain by $y(t)=L^{-1}(Y(S))$

CODE

```
clc;
clear all;
close all;

s=tf('s');
g=(s+1)/(s^2+4*s+4);
t=0:0.1:15;
u1= sin(t);
u2= cos(t);
subplot(2,2,1);
lsim(g,u1,t);
title('Response to Sine input');
subplot(2,2,2);
lsim(g,u2,t);
title('Response to Cosine input');
h=(s+5)/(s^2+5*s+6);
V=exp(-3*t);
subplot(2,2,3);
lsim(h,V,t);
title('Response to Exponential input');
```

Command window:

```
u1 =

  Columns 1 through 9

        0    0.0998    0.1987    0.2955    0.3894    0.4794    0.5646    0.6442    0.7174

  Columns 10 through 18

    0.7833    0.8415    0.8912    0.9320    0.9636    0.9854    0.9975    0.9996    0.9917

  Columns 19 through 27

    0.9738    0.9463    0.9093    0.8632    0.8085    0.7457    0.6755    0.5985    0.5155

  Columns 28 through 36

    0.4274    0.3350    0.2392    0.1411    0.0416   -0.0584   -0.1577   -0.2555   -0.3508

  Columns 37 through 45

   -0.4425   -0.5298   -0.6119   -0.6878   -0.7568   -0.8183   -0.8716   -0.9162   -0.9516

  Columns 46 through 54

   -0.9775   -0.9937   -0.9999   -0.9962   -0.9825   -0.9589   -0.9258   -0.8835   -0.8323

  Columns 55 through 63

   -0.7728   -0.7055   -0.6313   -0.5507   -0.4646   -0.3739   -0.2794   -0.1822   -0.0831


  Columns 64 through 72

    0.0168    0.1165    0.2151    0.3115    0.4048    0.4941    0.5784    0.6570    0.7290

  Columns 73 through 81

    0.7937    0.8504    0.8987    0.9380    0.9679    0.9882    0.9985    0.9989    0.9894

  Columns 82 through 90

    0.9699    0.9407    0.9022    0.8546    0.7985    0.7344    0.6630    0.5849    0.5010

  Columns 91 through 99

    0.4121    0.3191    0.2229    0.1245    0.0248   -0.0752   -0.1743   -0.2718   -0.3665

  Columns 100 through 108

   -0.4575   -0.5440   -0.6251   -0.6999   -0.7677   -0.8278   -0.8797   -0.9228   -0.9566

  Columns 109 through 117

   -0.9809   -0.9954   -1.0000   -0.9946   -0.9792   -0.9540   -0.9193   -0.8755   -0.8228

  Columns 118 through 126

   -0.7620   -0.6935   -0.6181   -0.5366   -0.4496   -0.3582   -0.2632   -0.1656   -0.0663

  Columns 127 through 135

    0.0336    0.1332    0.2315    0.3275    0.4202    0.5087    0.5921    0.6696    0.7404

  Columns 136 through 144

    0.8038    0.8592    0.9060    0.9437    0.9720    0.9906    0.9993    0.9980    0.9868

  Columns 145 through 151

    0.9657    0.9349    0.8948    0.8457    0.7883    0.7229    0.6503

u2 =

  Columns 1 through 9

    1.0000    0.9950    0.9801    0.9553    0.9211    0.8776    0.8253    0.7648    0.6967

  Columns 10 through 18

    0.6216    0.5403    0.4536    0.3624    0.2675    0.1700    0.0707   -0.0292   -0.1288

  Columns 19 through 27

   -0.2272   -0.3233   -0.4161   -0.5048   -0.5885   -0.6663   -0.7374   -0.8011   -0.8569

  Columns 28 through 36

   -0.9041   -0.9422   -0.9710   -0.9900   -0.9991   -0.9983   -0.9875   -0.9668   -0.9365
```

```
  -0.9041   -0.9422   -0.9710   -0.9900   -0.9991   -0.9983   -0.9875   -0.9668   -0.9365

 Columns 37 through 45

  -0.8968   -0.8481   -0.7910   -0.7259   -0.6536   -0.5748   -0.4903   -0.4008   -0.3073

 Columns 46 through 54

  -0.2108   -0.1122   -0.0124    0.0875    0.1865    0.2837    0.3780    0.4685    0.5544

 Columns 55 through 63

   0.6347    0.7087    0.7756    0.8347    0.8855    0.9275    0.9602    0.9833    0.9965

 Columns 64 through 72

   0.9999    0.9932    0.9766    0.9502    0.9144    0.8694    0.8157    0.7539    0.6845

 Columns 73 through 81

   0.6084    0.5261    0.4385    0.3466    0.2513    0.1534    0.0540   -0.0460   -0.1455

 Columns 82 through 90

  -0.2435   -0.3392   -0.4314   -0.5193   -0.6020   -0.6787   -0.7486   -0.8111   -0.8654

 Columns 91 through 99

  -0.9111   -0.9477   -0.9748   -0.9922   -0.9997   -0.9972   -0.9847   -0.9624   -0.9304

 Columns 145 through 151

  -0.2598   -0.3549   -0.4465   -0.5336   -0.6154   -0.6910   -0.7597


v =

 Columns 1 through 9

   1.0000    0.7408    0.5488    0.4066    0.3012    0.2231    0.1653    0.1225    0.0907

 Columns 10 through 18

   0.0672    0.0498    0.0369    0.0273    0.0202    0.0150    0.0111    0.0082    0.0061

 Columns 19 through 27

   0.0045    0.0033    0.0025    0.0018    0.0014    0.0010    0.0007    0.0006    0.0004

 Columns 28 through 36

   0.0003    0.0002    0.0002    0.0001    0.0001    0.0001    0.0001    0.0000    0.0000

 Columns 37 through 45

   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000

 Columns 46 through 54

   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000

 Columns 91 through 99

   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000

 Columns 100 through 108

   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000

 Columns 109 through 117

   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000

 Columns 118 through 126

   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000

 Columns 127 through 135

   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000

 Columns 136 through 144

   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000

 Columns 145 through 151

   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
```

OUTPUT:

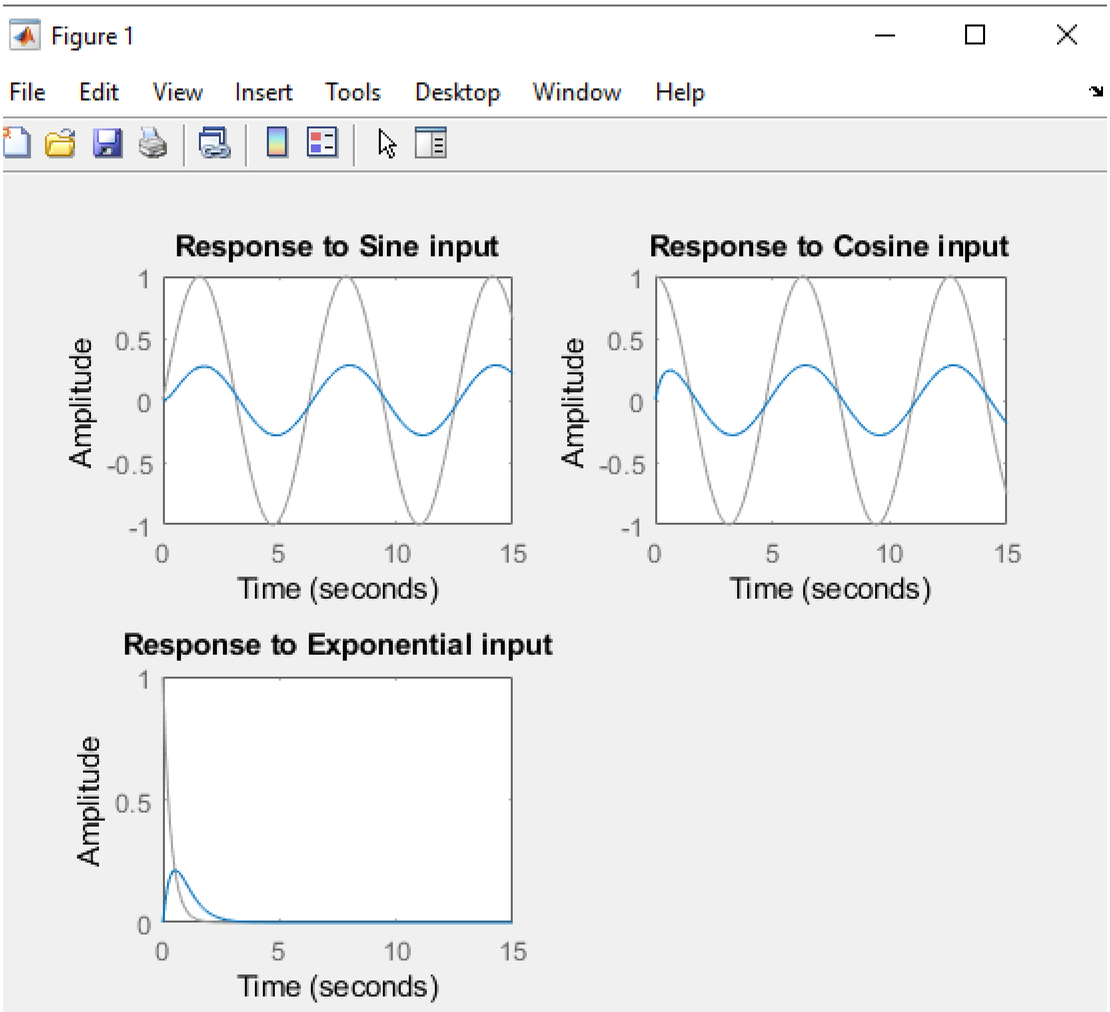


Figure 1
File Edit View Insert Tools Desktop Window Help
Response to Sine input
Amplitude
Time (seconds)
Response to Cosine input
Amplitude
Time (seconds)
Response to Exponential input
Amplitude
Time (seconds)

Experiment-9

Aim

Generate a MATLAB code to plot poles and zeros of a given z-transform of a signal.

Theory

Once the Z-transform of a system has been determined, one can use the information contained in function's polynomials to graphically represent the function and easily observe many defining characteristics. The Z-transform will have the below structure, based on Rational Functions:

$$X(z)=P(z)/Q(z)$$

The two polynomials, $P(z)$ and $Q(z)$, allow us to find the poles and zeros of the Z-Transform.

Zeros

The value(s) for z where $P(z) = 0$.
The complex frequencies that make the overall gain of the filter transfer function zero.

Poles

The value(s) for z where $Q(z) = 0$.
The complex frequencies that make the overall gain of the filter transfer function infinite.

Code:

```
clc;
clear all;
close all;

figure('Name','Harshal Chowdhary 2K19/EC/071','Numbertitle','off');
b=input('Enter the numerator coefficient');
a=input('Enter the denominator coefficent');
zplane(b,a)
title('Z Plane')
```

Input:

```
Command Window
Enter the numerator coefficient[1/4]
Enter the denominator coefficent[1 -3/4 1/8]
```

Output:

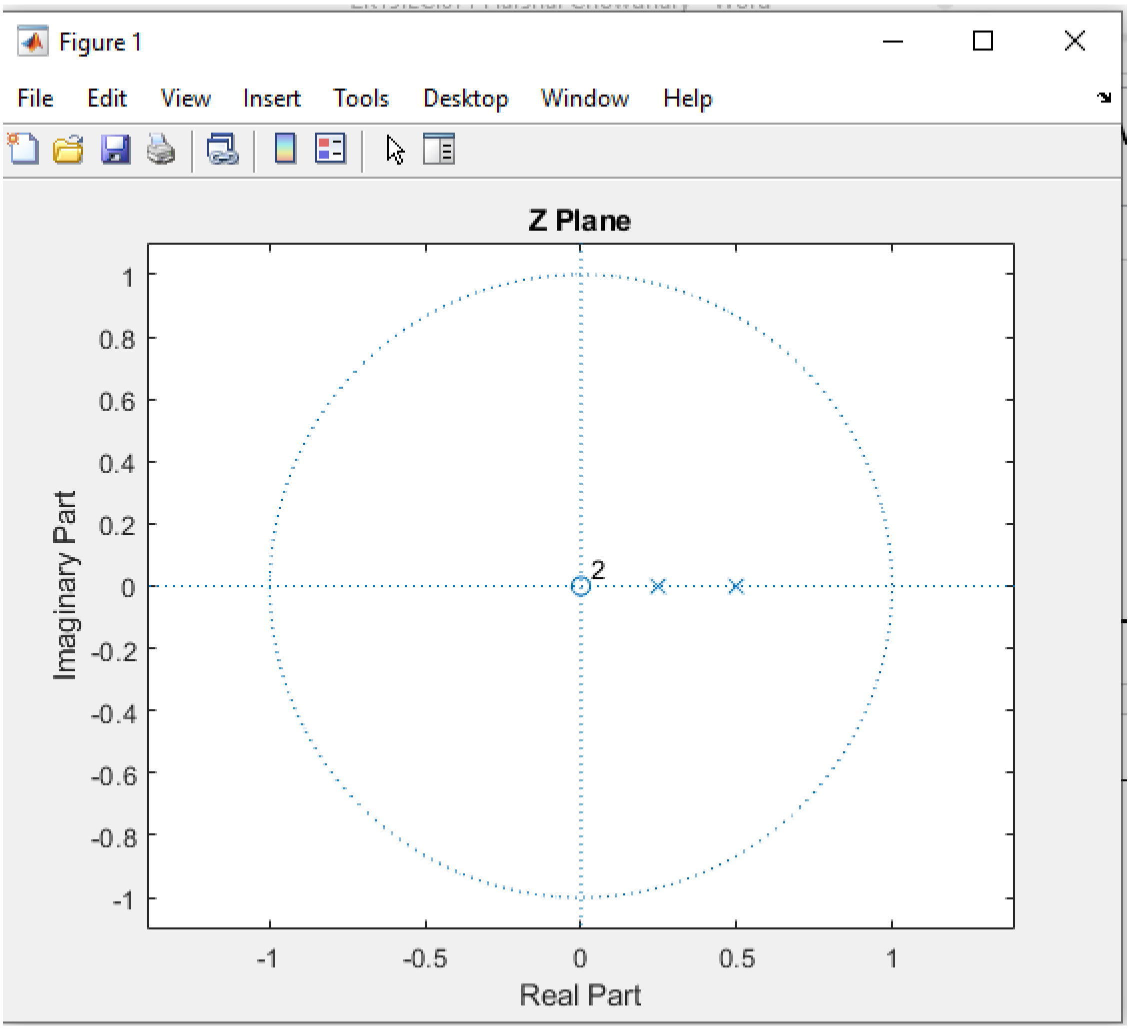


Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
Z Plane
Imaginary Part
Real Part
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1
2