DIGITAL SIGNAL PROCESSING LAB

(Practical file)

IT-353

BACHELOR OF TECHNOLOGY
INFORMATION TECHNOLOGY

University School of Information, Communication and Technology

GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY, DWARKA

Submitted to:
Dr. Mansi Jhamb

Submitted by:
Apoorv Aron
01316401520
B.Tech IT 5TH SEM

INDEX

EXPERIMENT - 1

AIM:

To plot the waveforms for: Sine, Cos, Exponential, Ramp, Unit Impulse, Unit Step signal in continuous time using MATLAB.

THEORY:

1. **Sine Signal:** Sine signal is in the form of $x(t) = A\sin(\omega_0 \pm \Phi)$.
2. **Cosine Signal :** Cosine signal is in the form of $x(t) = A\cos(\omega_0 \pm \Phi)$.
3. **Exponential Signal:** Exponential signal is in the form of $x(t) = e^{at}$
4. **Unit Step Signal:** Step signal is denoted by u(t) and It is defined as u(t)={0 if t<0 & 1 if t>0 }
5. **Ramp Signal:** Ramp signal is denoted by r(t) and It is defined as r(t)={t if t≥0 & 0 at t<0}
6. **Unit Impulse Signal:** Impulse signal is denoted by δ(t) and It is defined as δ(t)={1 at t=0 & 0 at t≠0}

CODE:

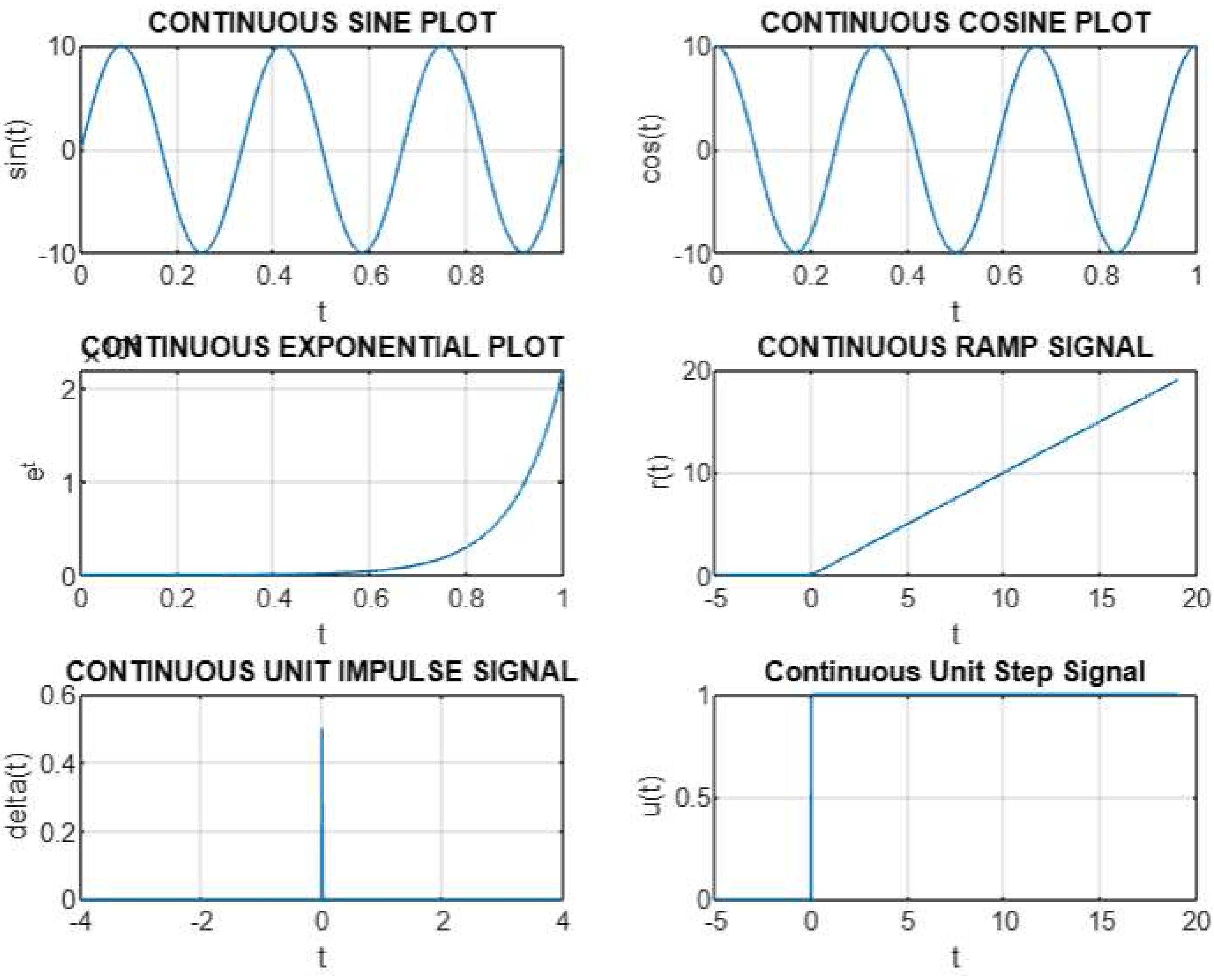
```
clc;
clear
all;
close
all;

% Sine
A=input('Amplitude for sine signal:');
f=input('Frequency for sine signal:');
t=0:0.01:1;
Y=A*sin(2*pi*f*t);
subplot(3,2,1);
plot(t,Y);
title('CONTINUOUS SINE PLOT');
ylabel('sin(t)');
xlabel('t');
grid on;
```

```
% Cos
A=input('Amplitude for cosine signal: ');
f=input('Frequency for cosine signal: ');
t=0:0.01:1;
Y=A*cos(2*pi*f*t);
subplot(3,2,2);
plot(t,Y);
title('CONTINUOUS COSINE PLOT');
ylabel('cos(t)');
xlabel('t');
grid on;

% Exponential signal
A=input('Enter Exponent Coefficient:');
t=0:0.01:1;
Y=exp(A*t);
subplot(3,2,3);
plot(t,Y);
title('CONTINUOUS EXPONENTIAL PLOT');
ylabel('e^t');
xlabel('t');
grid on;

% ramp
N=input('Enter input range for ramp and unit impulse:');
n=-5:0.01:N-1;
Y=n.*(sign(n)+1)/2;
subplot(3,2,4);
plot(n,Y);
ylabel('r(t)');
xlabel('t');
title('CONTINUOUS RAMP SIGNAL');
grid on;

% Unit impulse signal
N = -4:0.01:4;
Y=(sign(n)+1)/2-(sign(n-0.0001)+1)/2;
subplot(3,2,5);
plot(n,Y);
ylabel('delta(t)');
xlabel('t');
```

```
title('CONTINUOUS UNIT IMPULSE SIGNAL');
grid on;

% Unit step signal
N=input('Enter input range for unit step signal: ');
n=-5:0.01:N-1;
Y=(sign(n)+1)/2;
subplot(3,2,6);
plot(n,Y);
ylabel('u(t)');
xlabel('t');
title('CONTINUOUS UNIT STEP SIGNAL');
grid on;
```

Command Window:

```
Amplitude for sine signal:
10
Frequency for sine signal:
3
Amplitude for cosine signal:
10
Frequency for cosine signal:
3
Enter Exponent Coefficient:
10
Enter input range for ramp and unit impulse:
20
Enter input range for unit step signal:
20
>> |
```


CONTINUOUS SINE PLOT
sin(t)
t
CONTINUOUS COSINE PLOT
cos(t)
t
CONTINUOUS EXPONENTIAL PLOT
e^t
t
CONTINUOUS RAMP SIGNAL
r(t)
t
CONTINUOUS UNIT IMPULSE SIGNAL
delta(t)
t
Continuous Unit Step Signal
u(t)
t

EXPERIMENT - 2

AIM:

To plot the waveforms for: Sine, Cos, Exponential, Ramp, Unit Impulse, Unit Step signal in discrete time using MATLAB.

THEORY:

1. **Sine Signal:** Sine signal is in the form of $x(t) = A\sin(\omega_0 \pm \Phi)$.
2. **Cosine Signal:** Cosine signal is in the form of $x(t) = A\cos(\omega_0 \pm \Phi)$.
3. **Exponential Signal:** Exponential signal is in the form of $x(t) = e^{at}$
4. **Unit Step Signal:** Step signal is denoted by u(t) and It is defined as $u(t)=\{0$ if $t<0$ & 1 if $t>0$ }
5. **Ramp Signal:** Ramp signal is denoted by r(t) and It is defined as $r(t)=\{t$ if $t\geq 0$ & 0 at $t<0\}$
6. **Unit Impulse Signal:** Impulse signal is denoted by $\delta(t)$ and It is defined as $\delta(t)=\{1$ at $t=0$ & 0 at $t\neq 0\}$

CODE:

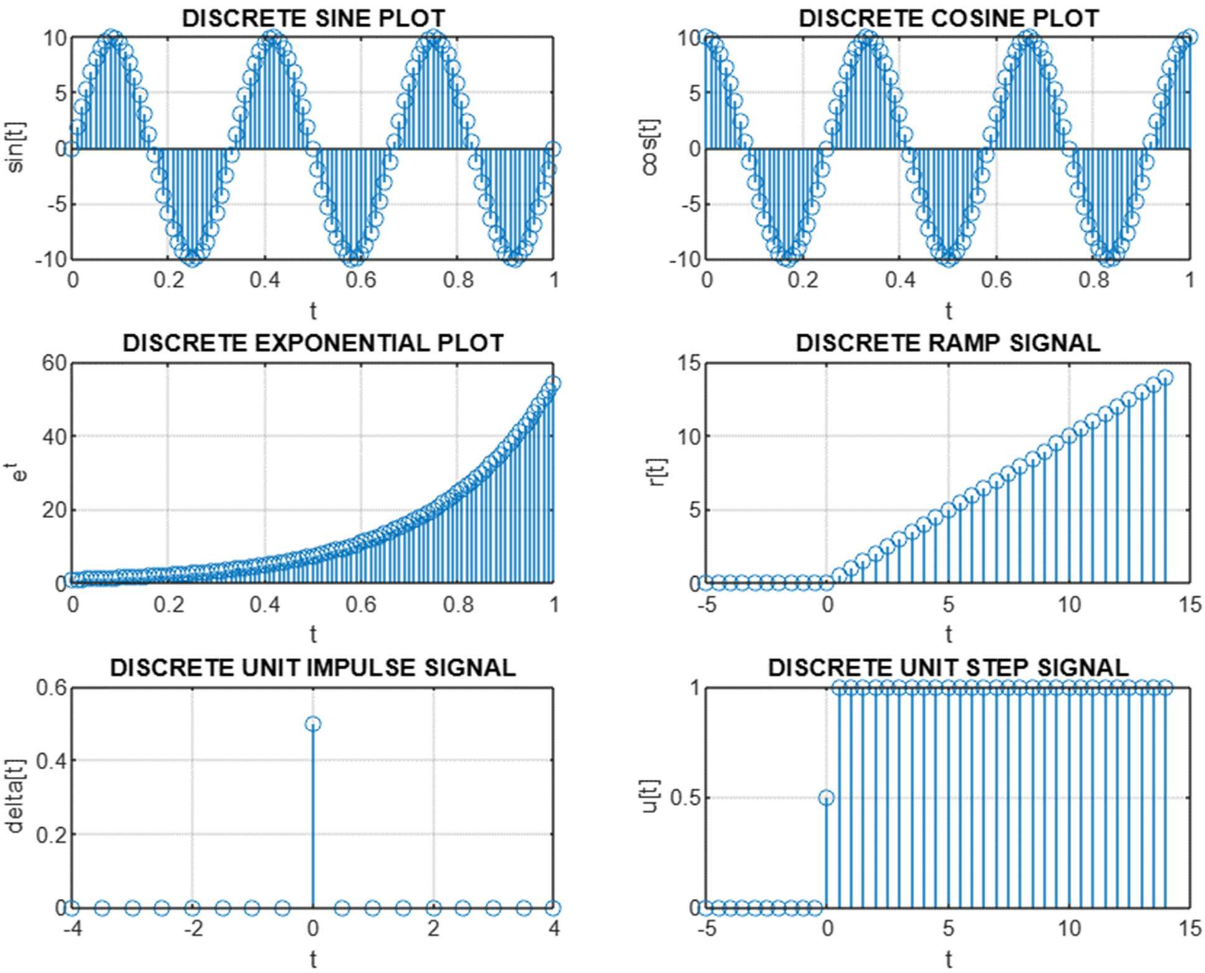
```
clc;
clear all;
close all;

% Sine
A=input('Amplitude for sine signal: ');
f=input('Frequency for sine signal: ');
t=0:0.01:1;
Y=A*sin(2*pi*f*t);
subplot(3,2,1);
stem(t,Y);
title('DISCRETE SINE PLOT');
ylabel('sin[t]');
xlabel('t');
grid on;
```

```
% Cos
A=input('Amplitude for cosine signal: ');
f=input('Frequency for cosine signal: ');
t=0:0.01:1;
Y=A*cos(2*pi*f*t);
subplot(3,2,2);
stem(t,Y);
title('DISCRETE COSINE PLOT');
ylabel('cos[t]');
xlabel('t');
grid on;

% Exponential signal
A=input('Enter Exponent Coefficient: ');
t=0:0.01:1;
Y=exp(A*t);
subplot(3,2,3);
stem(t,Y);
title('DISCRETE EXPONENTIAL PLOT');
ylabel('e^t');
xlabel('t');
grid on;

% ramp
N=input('Enter input range for ramp and unit impulse: ');
n=-5:0.5:N-1;
Y=n.*(sign(n)+1)/2;
subplot(3,2,4);
stem(n,Y);
ylabel('r[t]');
xlabel('t');
title('DISCRETE RAMP SIGNAL');
grid on;

% Unit impulse signal
n=-4:0.5:4;
Y=(sign(n)+1)/2-(sign(n-0.0001)+1)/2;
subplot(3,2,5);
stem(n,Y);
ylabel('delta[t]');
```

```
xlabel('t');
title('DISCRETE UNIT IMPULSE SIGNAL');
grid on;

% Unit step signal
N=input('Enter input range for unit step signal: ');
n=-5:0.5:N-1;
Y=(sign(n)+1)/2;
subplot(3,2,6);
stem(n,Y);
ylabel('u[t]');
xlabel('t');
title('DISCRETE UNIT STEP SIGNAL');
grid on;
```

Command Window

```
Amplitude for sine signal:
10
Frequency for sine signal:
3
Amplitude for cosine signal:
10
Frequency for cosine signal:
3
Enter Exponent Coefficient:
4
Enter input range for ramp and unit impulse:
15
Enter input range for unit step signal:
15
```



DISCRETE SINE PLOT
sin[t]
t
DISCRETE COSINE PLOT
cos[t]
t
DISCRETE EXPONENTIAL PLOT
e^t
t
DISCRETE RAMP SIGNAL
r[t]
t
DISCRETE UNIT IMPULSE SIGNAL
delta[t]
t
DISCRETE UNIT STEP SIGNAL
u[t]
t

EXPERIMENT – 3

AIM:

To perform sampling. Consider an analog signal x(t) = 3 cos2000(pi)t+ 5sin6000(pi)t+10cos12000(pi)t. What is the Nyquist rate for this signal? What is a discrete time signal obtained after sampling at 5000 samples/s?

THEORY:

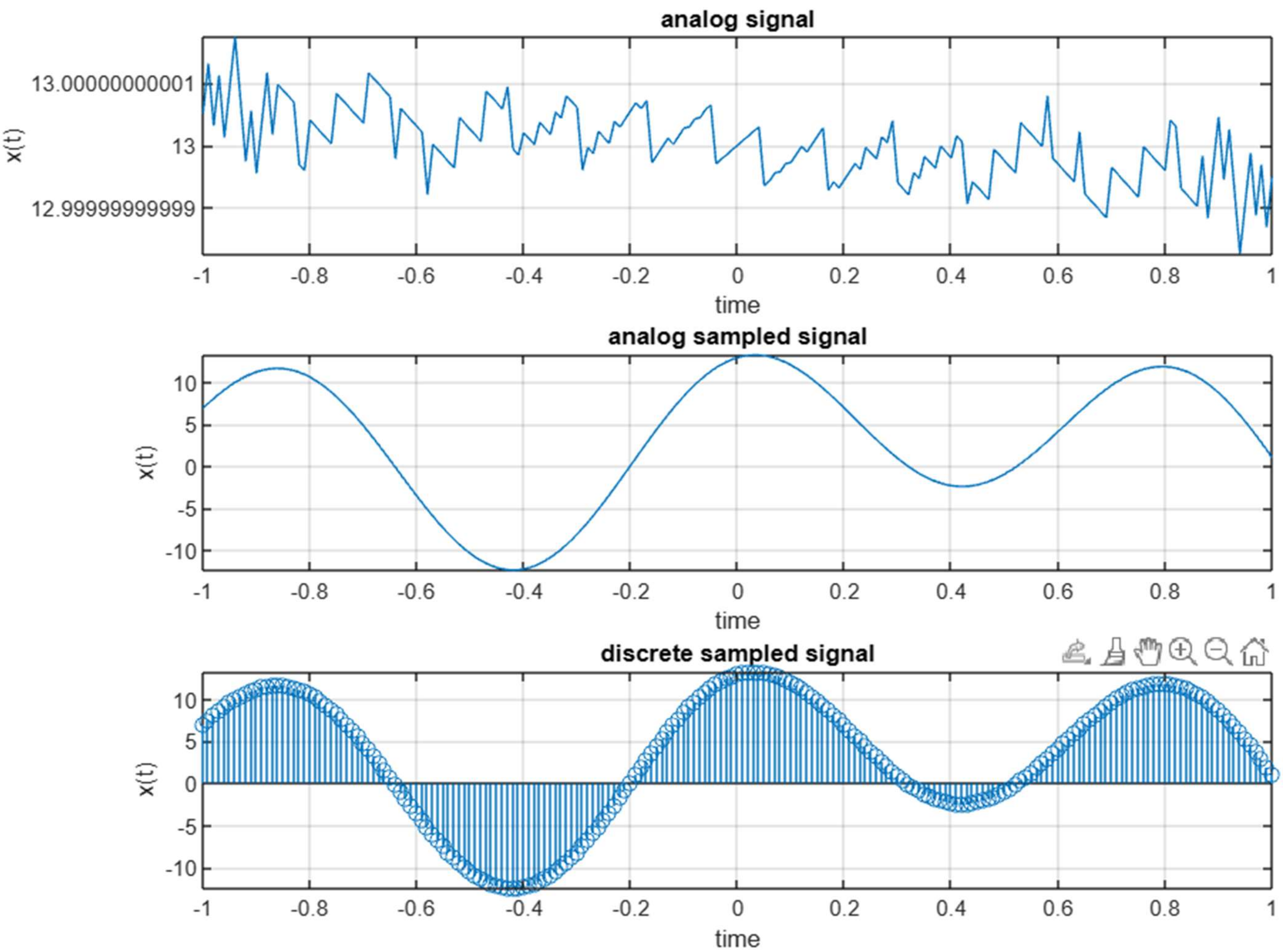
Nyquist rate : the minimum rate at which a signal can be sampled without introducing errors, which is twice the highest frequency present in the signal.

Signal : x(t) = 3 cos2000(pi)t+ 5sin6000(pi)t+10cos12000(pi)t

CODE:

```
clc;
clear all;
t = -1:0.01:1;
x = 3*cos(2000*pi*t) + 5*sin(6000*pi*t) + 10*cos(12000*pi*t);
subplot(3,1,1);
plot(t,x);
xlabel('time');
ylabel('x(t)');
title('analog signal');
grid;

%after sampling for 5000 sample/s
x = 3*cos(2/5*pi*t) + 5*sin(6/5*pi*t) + 10*cos(12/5*pi*t);
subplot(3,1,2);
plot(t,x);
grid;
xlabel('time');
ylabel('x(t)');
```

```
title('analog sampled signal');
subplot(3,1,3);
stem(t,x);
grid; xlabel('time');
ylabel('x(t)');
title('discrete sampled signal')
```



EXPERIMENT – 4

AIM:

To perform operations of folding and the delaying (or advancing) on signal.

THEORY:

Folding is a technique that involves reflecting the signal about a certain point in time.

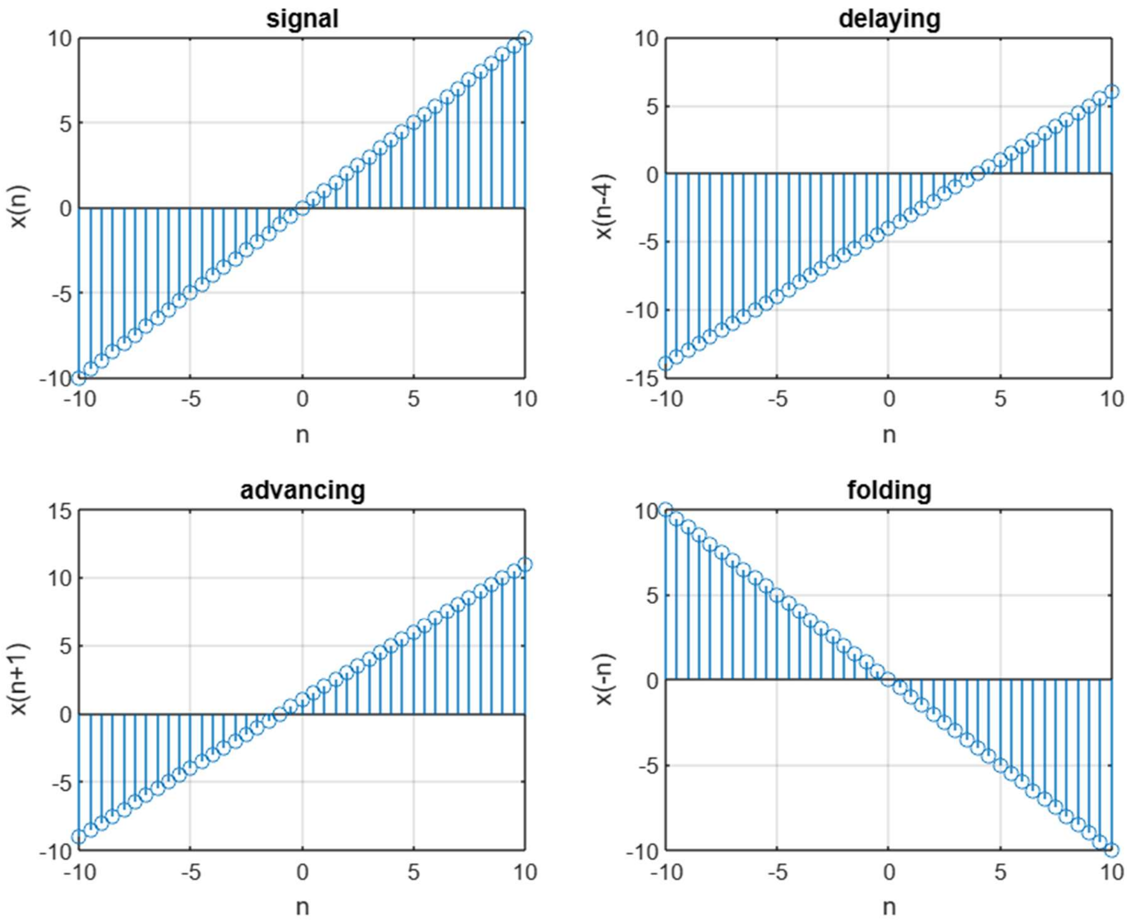
Delaying a signal involves shifting the signal in time by a certain amount.

Advancing a signal is like delaying a signal but involves shifting the signal in the opposite direction in time.

CODE:

```
clc;
clear all;
close all;

n= -10:0.5:10;
x= n; % x(n)
x1=n-4;% x(n-4)
x2=n+1;% x(n+1)
x3=-n; % x(-n)
subplot(2,2,1);
stem(n,x);
xlabel('n');
ylabel('x(n)');
title('signal');
grid;

subplot(2,2,2);
stem(n,x1);
xlabel('n');
ylabel('x(n-4)');
```

```
title('delaying');
grid;

subplot(2,2,3);
stem(n,x2);
xlabel('n');
ylabel('x(n+1)');
title('advancing');
grid;

subplot(2,2,4);
stem(n,x3);
xlabel('n');
ylabel('x(-n)');
title('folding');
grid;
```



EXPERIMENT – 5

AIM:

To perform Convolution.

THEORY:

Convolution: Convolution is the process by which one may compute the overlap of two graphs. In fact, convolution is also interpreted as the area shared by the two graphs over time. Metaphorically, it is a blend between the two functions as one passes over the other.

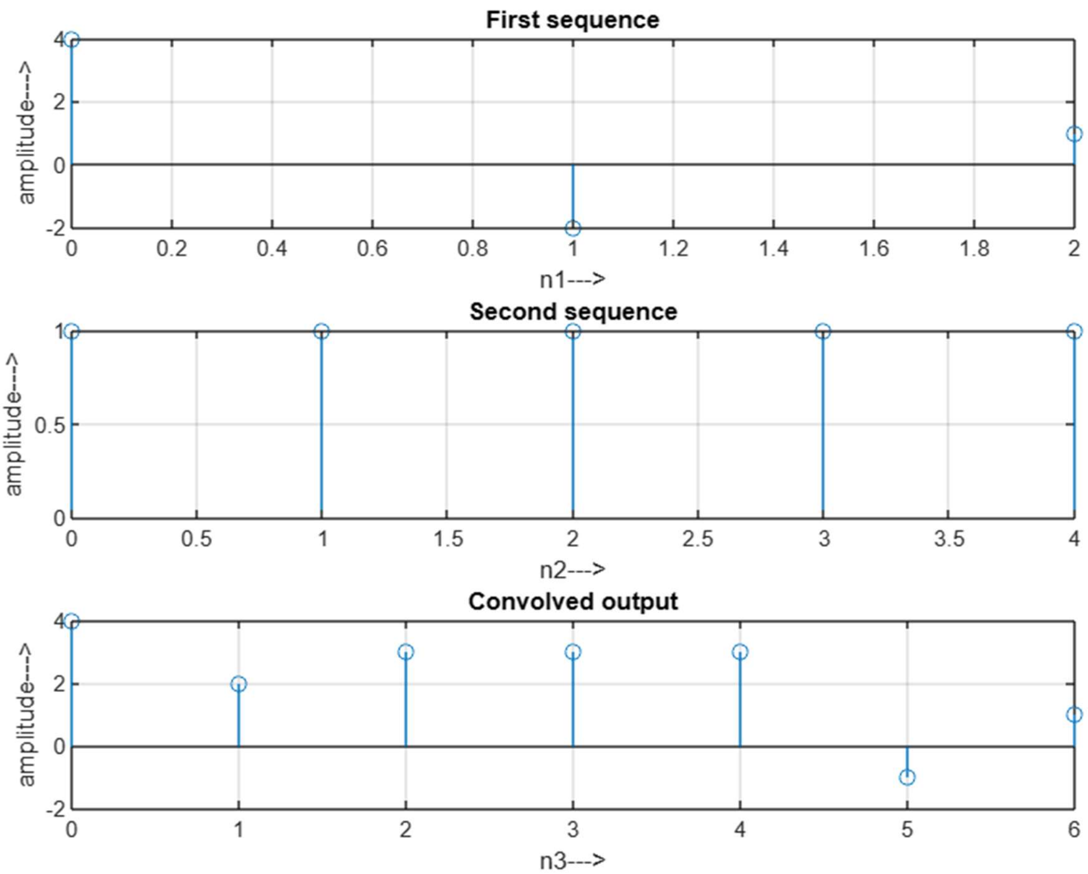
CODE:

```
clc;
close all;

x1=input('Enter the first sequence x1(n) = ');
x2=input('Enter the second sequence x2(n) = ');
L=length(x1);
M=length(x2);
N=L+M-1;
yn=conv(x1,x2);
disp('The values of y(n) are= ');
disp(yn);
n1=0:L-1;
subplot(311);
stem(n1,x1);
grid on;
xlabel('n1--->');
ylabel('amplitude--->');
title('First sequence');
n2=0:M-1;
subplot(312);
stem(n2,x2);
grid on;
```

```
xlabel('n2--->');
ylabel('amplitude--->');
title('Second sequence');
n3=0:N-1;
subplot(313);
stem(n3,yn);
grid on;
xlabel('n3--->');
ylabel('amplitude--->');
title('Convolved output');
```

Command Window

```
Enter the first sequence x1(n) =
[4,-2,1]
Enter the second sequence x2(n) =
[1,1,1,1,1]
The values of y(n) are=
     4     2     3     3     3    -1     1
```



EXPERIMENT – 6

AIM:

To perform Cross-Correlation and Auto Correlation.

THEORY:

Cross-correlation is a measure of the similarity between two signals x[n] and y[n] as a function of the time lag between them.

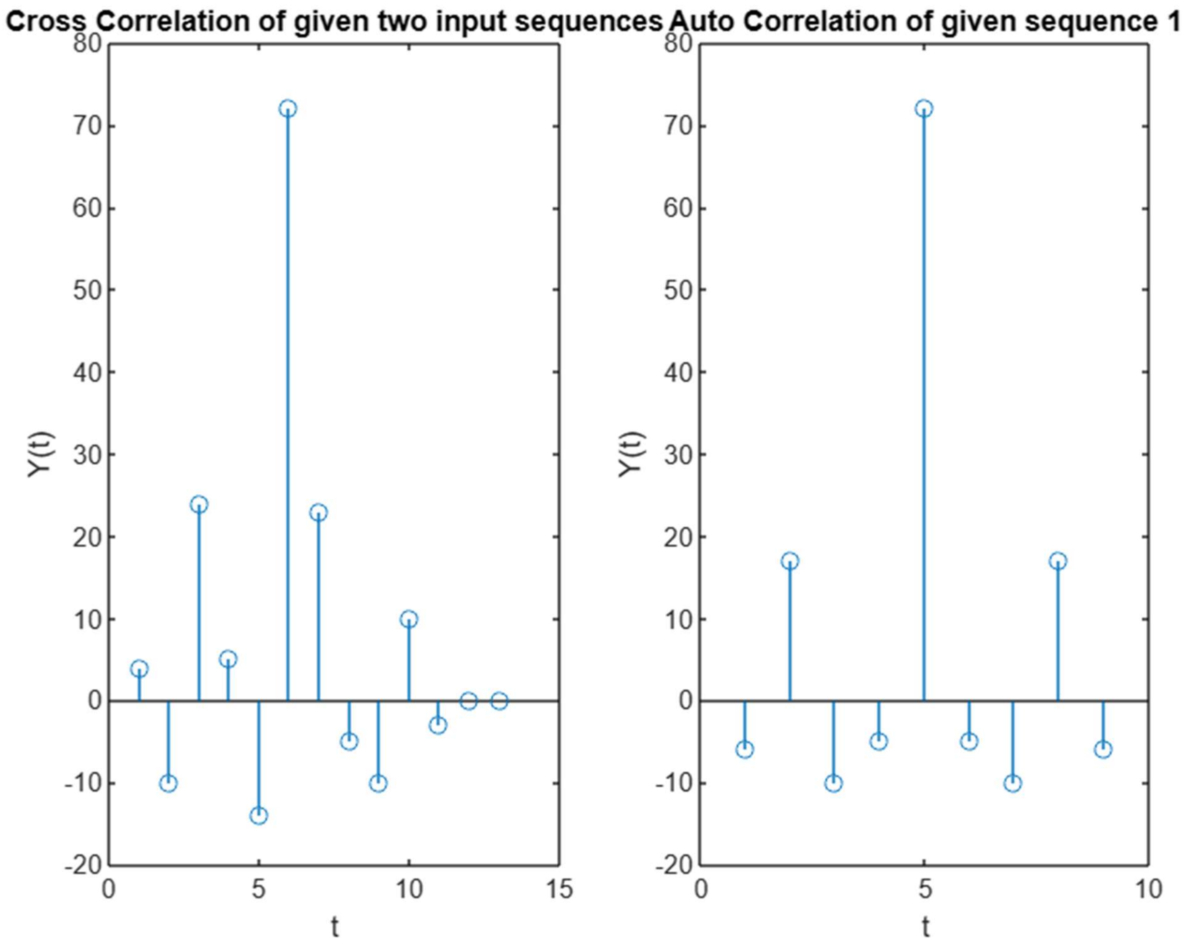
Autocorrelation is like cross-correlation but involves comparing a signal to a copy of itself rather than to another signal.

CODE:

```
clc;
close all;
a = input('Enter 1st sequence x1(n) = ');
b = input('Enter 2nd sequence x2(n) = ');
disp('Cross Correlation of given two input sequences');
y = xcorr(a,b);
disp(y);
subplot(1,2,1);
stem(y);
xlabel('t');
ylabel('Y(t)');
title('Cross Correlation of given two input sequences');
disp('Auto Correlation of given sequence 1');
y = xcorr(a,a);
disp(y);
subplot(1,2,2);
stem(y);
xlabel('t');
ylabel('Y(t)');
title('Auto Correlation of given sequence 1');
```

Command Window

```
Enter 1st sequence x1(n) =
[2,-1,3,7, -3]
Enter 2nd sequence x2(n) =
[1,-1,2,5,7,-4,2]
Cross Correlation of given two input sequences
    4.0000  -10.0000   24.0000    5.0000  -14.0000   72.0000   23.0000   -5.0000  -10.0000   10.0000   -3.0000    0.0000    0.0000

Auto Correlation of given sequence 1
   -6.0000   17.0000  -10.0000   -5.0000   72.0000   -5.0000  -10.0000   17.0000   -6.0000
```



EXPERIMENT – 7

AIM:

To compute DFT.

THEORY:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}$$

$$where \quad W_N^{nk} = e^{-j\frac{2\pi nk}{N}} \text{ [TWIDDLE FACTOR]}$$

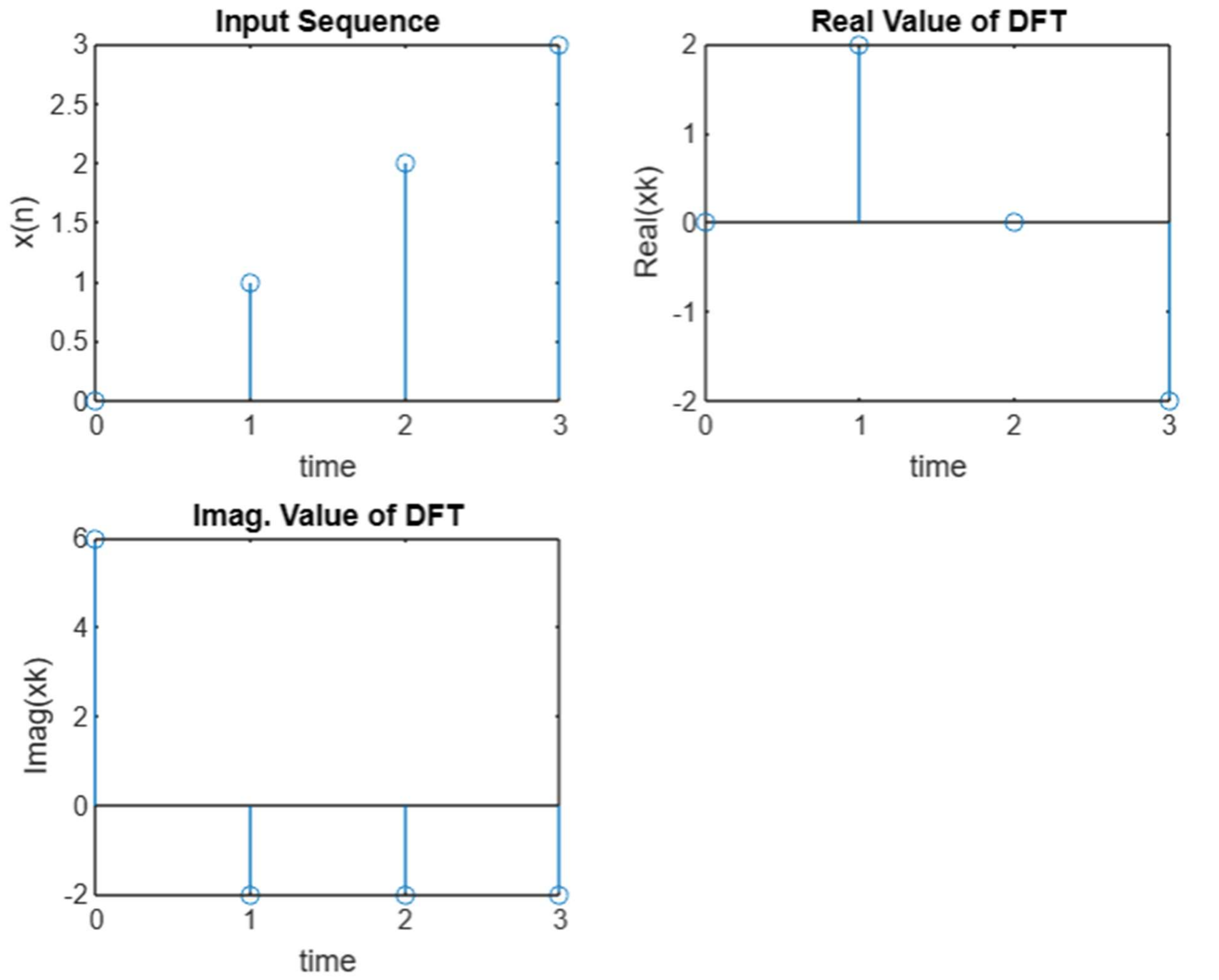
CODE:

```
clc;
clear all;
close all;

xn = input('Enter the input sequence: ');
N = length(xn);
n=0:1:N-1;
subplot(2,2,1);
stem(n,xn);
xlabel('time');
ylabel('x(n)');
title('Input Sequence');
xk = fft(xn,N);
disp(xk);
k=0:1:N-1;
subplot(2,2,2);
stem(k,imag(xk));
xlabel('time');
ylabel('Real(xk)');
title('Real Value of DFT');
subplot(2,2,3);
stem(k,real(xk));
```

```
xlabel('time');
ylabel('Imag(xk)');
title('Imag. Value of DFT')
```

Command Window

```
Enter the input sequence:
[0,1,2,3]
   6.0000 + 0.0000i  -2.0000 + 2.0000i  -2.0000 + 0.0000i  -2.0000 - 2.0000i
```



EXPERIMENT - 8

To perform Circular Convolution.

THEORY:

Circular convolution is a technique used to perform convolution of two periodic signals. It is defined as:

$$y[n] = \sum x[k]h[n-k]$$

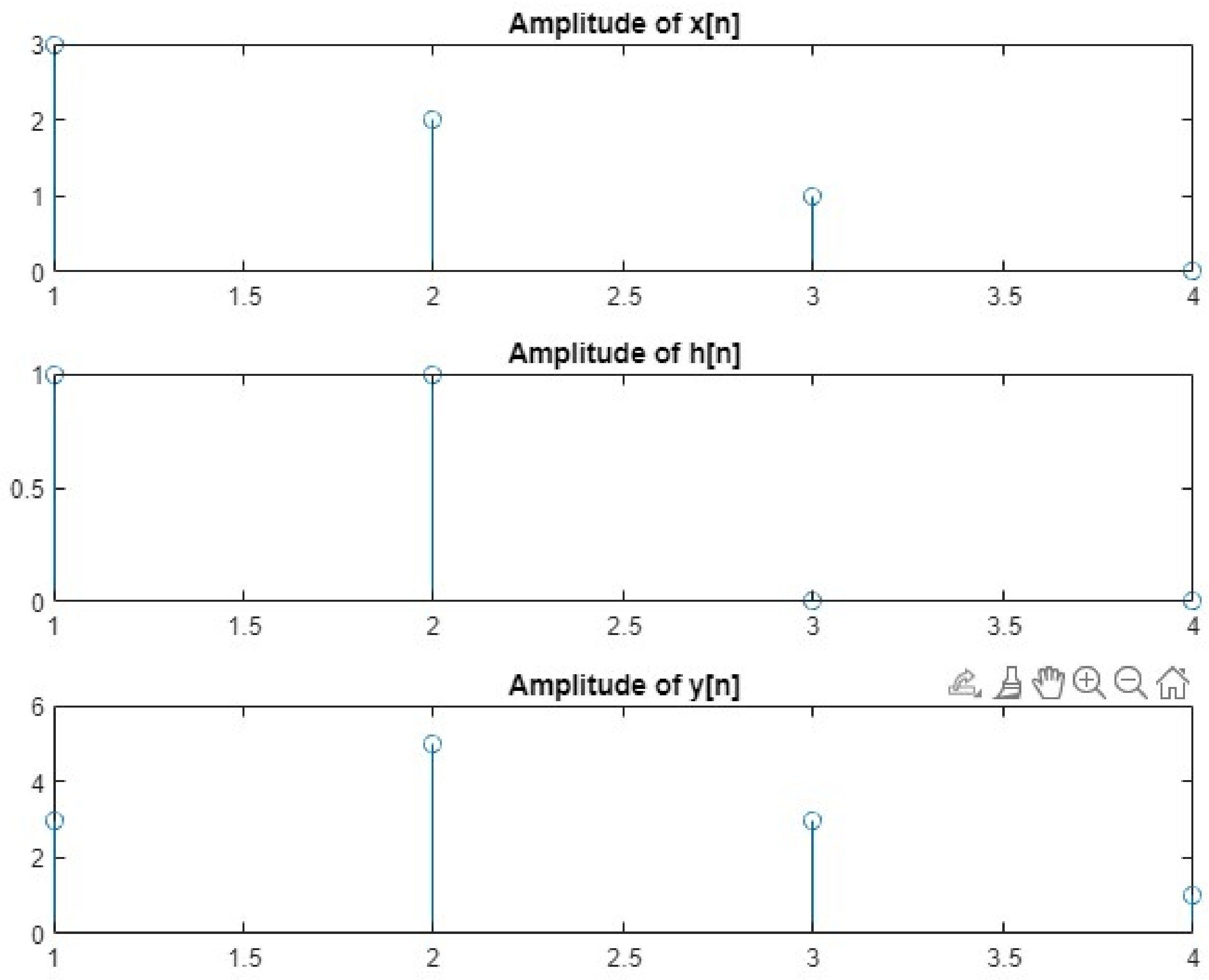
where the sum is taken over all time indices k such that k is equivalent to n modulo the period of the signals.

CODE:

```
clc;
clear all;
close all;
x = input('Enter the first sequence: ');
h = input('Enter the second sequence: ');
n = length(x);
y = cconv(x, h, n);
amplitude_x = abs(x);
amplitude_h = abs(h);
amplitude_y = abs(y);
figure;
subplot(3,1,1);
stem(amplitude_x);
title('Amplitude of x[n]');
subplot(3,1,2);
stem(amplitude_h);
title('Amplitude of h[n]');
subplot(3,1,3);
stem(amplitude_y);
title('Amplitude of y[n]');
disp('Circular Convolution');
disp(amplitude_y);
```

Command Window

```
Enter the first sequence:
[3, 2, 1, 0]
Enter the second sequence:
[1, 1, 0, 0]
Circular Convolution
     3     5     3     1

>>
```



EXPERIMENT - 9

To perform z transform.

THEORY:

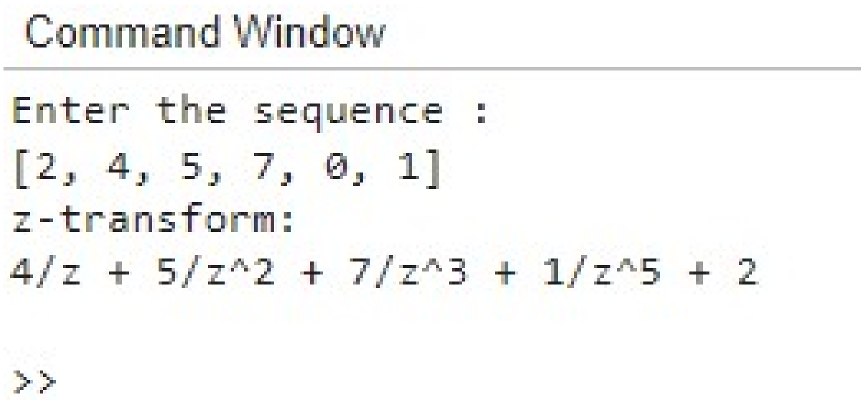
The z-transform is a mathematical tool used to represent discrete-time signals and systems in the frequency domain. It is defined as:

$$X(z) = \sum x[n]z^{-n}$$

where x[n] is the discrete-time signal, z is a complex variable, and the sum is taken over all time indices n.

CODE:

```
clc;
close all;
z=sym('z');
x = input('Enter the sequence : ');
l=length(x);
X=0;
for i=0:l-1
X=X+x(i+1)*z^(-i);
end
disp('z-transform: ') ;
disp(X);
```



EXPERIMENT - 10

AIM:

To perform linear convolution using DFT.

THEORY:

Linear convolution can be computed using the discrete Fourier transform (DFT) by taking the DFTs of the two signals, multiplying them element-wise, and then computing the inverse DFT (IDFT) of the product. This method is known as the fast convolution algorithm.

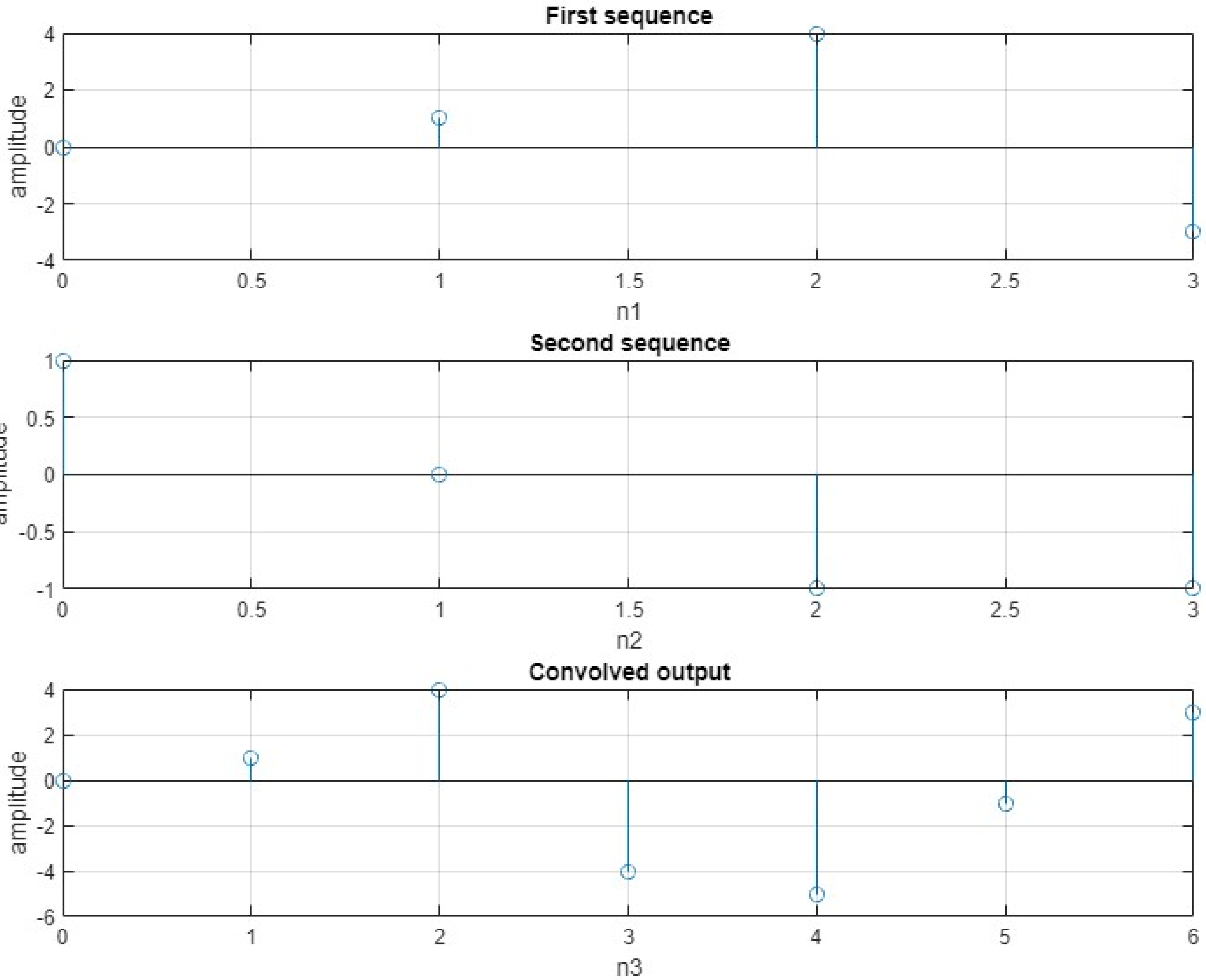
CODE:

```
clc;
clear all;
close all;
x1 = input('Enter the first input sequence xl[n]:');
x2 = input('Enter the second input sequence x2[n]:');
Lx1=length(x1);
Lx2=length(x2);
N=Lx1+Lx2-1;
X1=fft (x1,N);
X2=fft (x2,N);
Y=X1.*X2;
y=ifft (Y, N);
disp ('Linear Convolution of x1 [n] & x2[n] through dft is ')
disp (y)

% displaying First sequence (x1)
n1=0:Lx1-1;
subplot(3,1,1);
stem(n1,x1);
grid;
xlabel('n1');
ylabel('amplitude');
title('First sequence');
```

```
% displaying Second sequence (x2)
n2=0:Lx2-1;
subplot(3,1,2);
stem(n2,x2);
grid;
xlabel('n2');
ylabel('amplitude');
title('Second sequence');

% displaying Convolved output (yn)
n3=0:N-1;
subplot(3,1,3);
stem(n3,y);
grid;
xlabel('n3');
ylabel('amplitude');
title('Convolved output');

%Verification
z=conv (x1, x2);
disp ('Linear Convolution of x1 [n] and x2 [n] using Built-in function is z [n]= ');
disp (z);
```

Command Window

```
Enter the first input sequence x1[n]:
[0, 1, 4, -3]
Enter the second input sequence x2[n]:
[1, 0, -1, -1]
Linear Convolution of x1 [n] & x2[n] through dft is
    0.0000    1.0000    4.0000   -4.0000   -5.0000   -1.0000    3.0000

Linear Convolution of x1 [n] and x2 [n] using Built-in function is z [n]=
     0     1     4    -4    -5    -1     3

>>
```



First sequence
amplitude
n1
Second sequence
amplitude
n2
Convolved output
amplitude
n3

EXPERIMENT - 11

AIM:

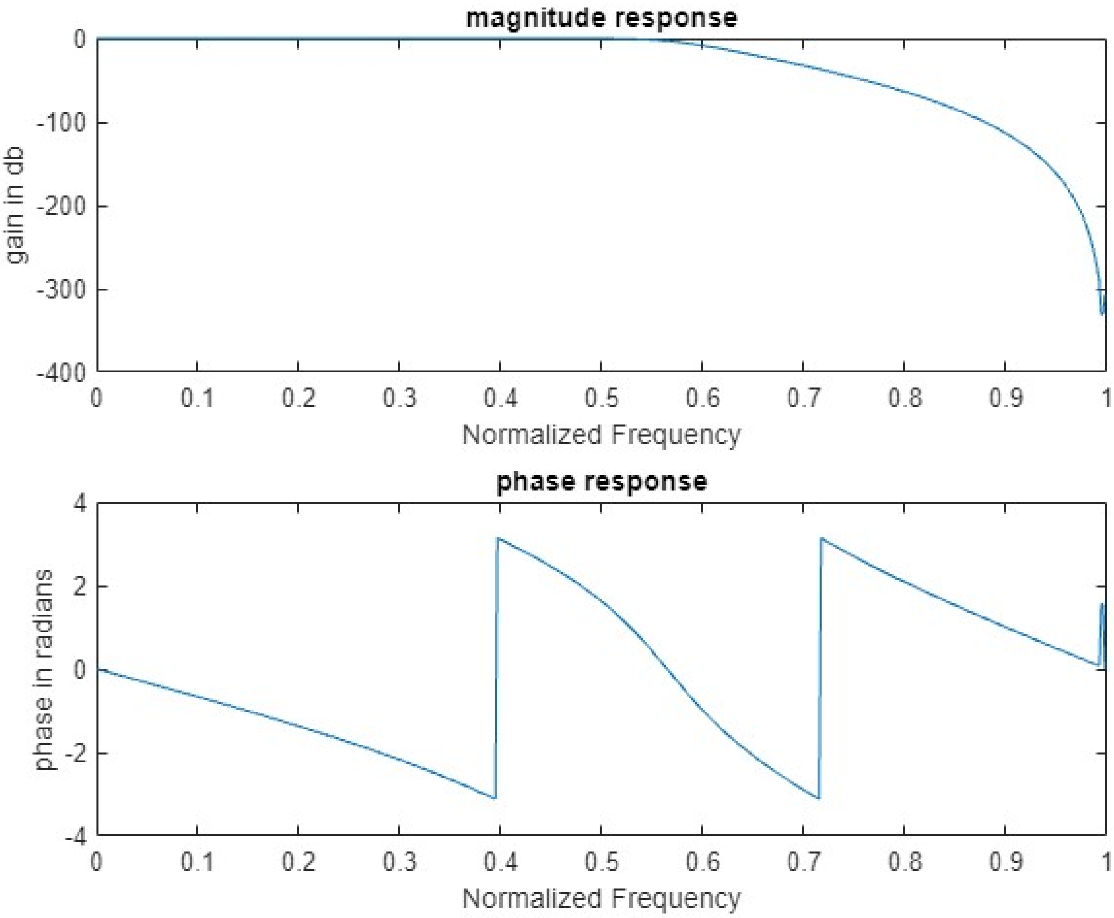
Design ButterWorth low pass Filter.

THEORY:

A Butterworth lowpass filter is a type of linear filter that is designed to pass low-frequency signals and reject high-frequency signals. It is characterized by a flat frequency response in the passband and a monotonic roll-off in the stopband.

CODE:

```
clc;
clear all;
close all;
alphas = 50;
alphap = 1;
fpass = 1050;
fstop = 600;
fsam = 3500;
wp = 2*fpass/fsam;
ws = 2*fstop/fsam;
[n,wn] = buttord(wp,ws,alphap,alphas);
[b,a] = butter(n,wn);
[h,w] = freqz(b,a);
subplot(2,1,1);
plot(w/pi,20*log10(abs(h)));
xlabel('Normalized Frequency');
ylabel('gain in db');
title('magnitude response');
subplot(2,1,2);
plot(w/pi,angle(h));
xlabel('Normalized Frequency');
ylabel('phase in radians');
title('phase response');
```


magnitude response
gain in db
0
-100
-200
-300
-400
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Normalized Frequency
phase response
phase in radians
4
2
0
-2
-4
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Normalized Frequency