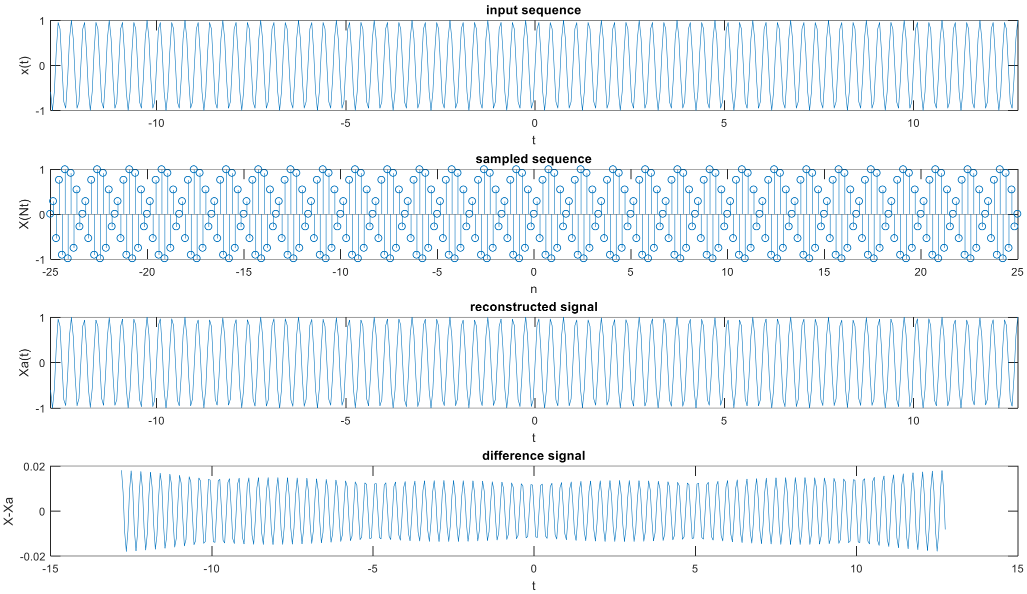
SAMPLING THEOREM

```
clc;
clear all;
close all;
N=512;
dt=0.05;
t=dt*(-N/2:N/2-1);
fm=input('enter the frequency of messege signal')
fs=input('enter the sampling frequency')
W=2*pi*fm;
X=sin(W*t);
Ts=1/fs;
T=Ts/1.1;
ts=-25:T:25;
Xs=sin(ts*W);
Xa=zeros(1,N);
for i=1:N
  Xa(i)=Xa(i)+sum(Xs.*sinc(1/T*(t(i)-ts))) ;
end
subplot(4,1,1);
plot(t,X);
xlabel('t');
ylabel('x(t)');
title('input sequence');
axis([min(t) max(t) -1 1]);
subplot(4,1,2);
stem(ts,Xs);
xlabel('n');
ylabel('X(Nt)');
title('sampled sequence');
subplot(4,1,3);
plot(t,Xa);
xlabel('t');
ylabel('Xa(t)');
title('reconstructed signal');
axis([min(t) max(t) -1 1]);
subplot(4,1,4);
plot(t,X-Xa);
xlabel('t');
ylabel('X-Xa');
title('difference signal');
```

OUTPUT

```
enter the frequency of messege signal 3

fm =

  3

enter the sampling frequency 6

fs =

    6
```



enter the frequency of messege signal 3

fm =

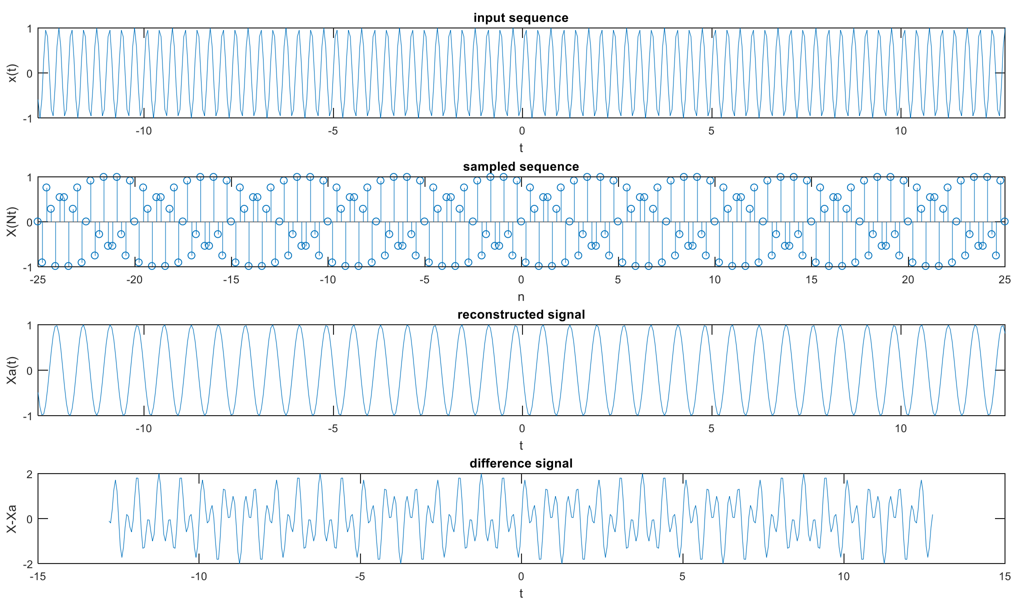
3

enter the sampling frequency 4

fs =

4

>>



enter the frequency of messege signal 3

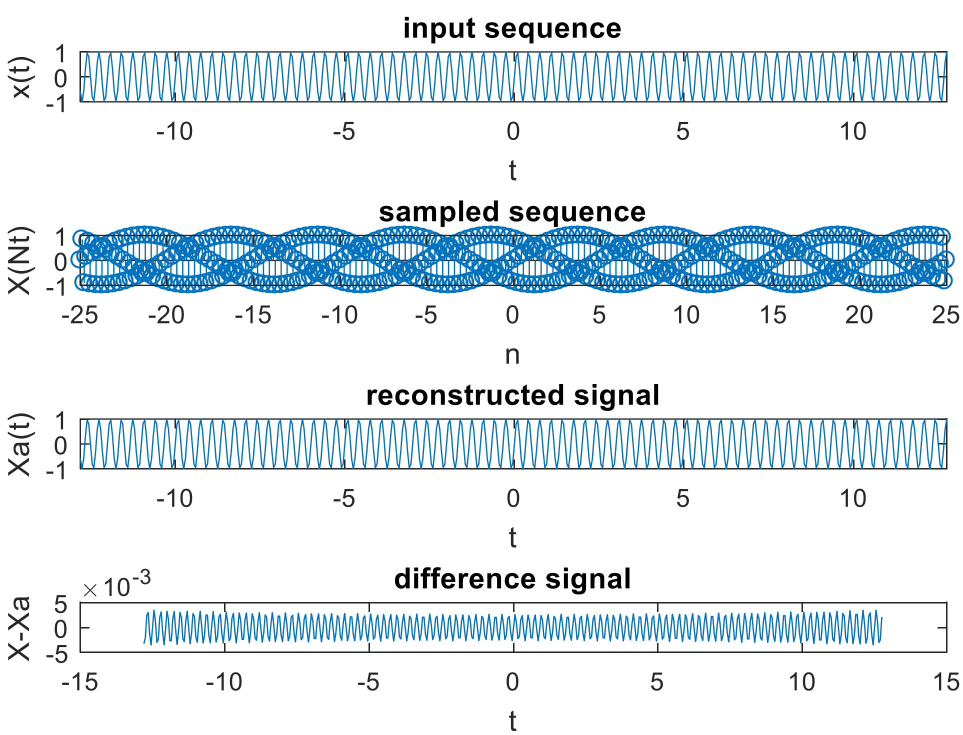
fm =

3

enter the sampling frequency 8

fs =

8



>>

LINEAR CONVOLUTION

```
x=input('enter the sequence x(n)= ')
a=0:length(x)-1;
subplot(2,2,1);
stem(a,x);
xlabel('n');
ylabel('x(n)');
title('SEQUENCE x(n)');
grid on;

h=input('Enter the sequence h(n)= ')
b=0:length(h)-1;
subplot(2,2,2);
stem(b,h);
xlabel('n');
ylabel('h(n)');
title('SEQUENCE h(n)');
grid on;

disp('linear  convolution of two given sequences ');
y=conv(x,h)
c=0:length(y)-1;
subplot(2,2,3);
stem(c,y);
xlabel('n');
ylabel('y(n)')
title('linear  convolution of two given sequences');
grid on;
```

OUTPUT

enter the sequence x(n)= [2,2,1]

x =

2 2 1

Enter the sequence h(n)= [2,2,2]

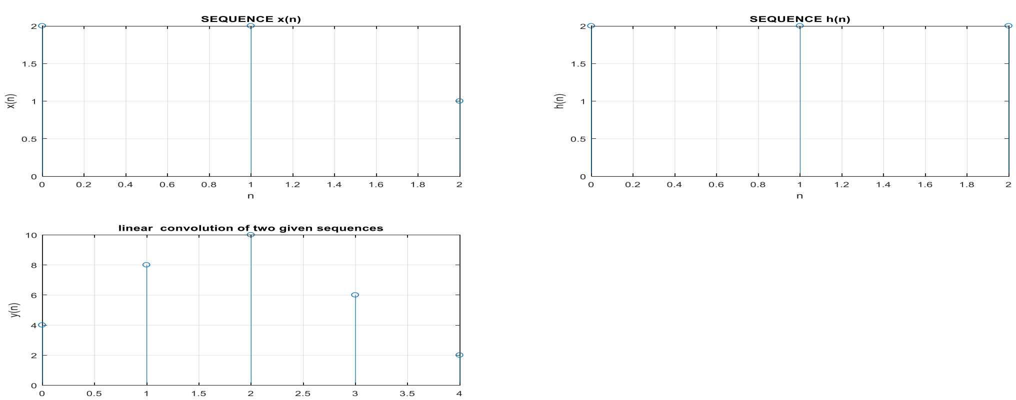
h =

2 2 2

linear convolution of two given sequences

y =

4 8 10 6 2



PROPERTIES

```
1.COMMUTATIVE PROP
clc
clear all
x=input('enter the first sequence');
h=input('enter the second sequence');
z=conv(x,h);
disp(z)
y=conv(h,x);
disp(y)
if (y==z)
disp('convolution is commutative')
else
    disp('convolution is npt commutative')
end
```

OUTPUT

```
enter the first sequence[1,2,3]
enter the second sequence[3,4,5]
     3    10    22    22    15

     3    10    22    22    15

convolution is commutative
```

2.DISTRIBUTIVE PROPERTY

```
clc
clear all
x1=input('enter the first sequence');
x2=input('enter the second sequence');
h=input('enter the third sequence');
z=conv(h,(x1+x2))
disp(z)
y=(conv(h,x1)+conv(h,x2))
disp(y)
if (y==z)
disp('convolution is distributive')
else
disp('convolution is not distributive')
end
```

OUTPUT

```
enter the first sequence[2,3,4]
enter the second sequence[5,9,13]
enter the third sequence[6,7,1]
z =

    42   121   193   131    17

    42   121   193   131    17
y =
```

```
    42   121   193   131    17

    42   121   193   131    17

convolution is distributive
```

3.ASSOCIATIVE PROPERTY

```
clc
clear all
x1=input('enter the first sequence');
x2=input('enter the second sequence');
h=input('enter the third squnce');
z=conv(h,(conv(x1,x2)))
disp(z)
y=conv(conv(h,x1),x2)
disp(z)
if (y==z)
disp('convolution is associative')
else
    disp('convolution is not associative')
end
```

OUTPUT

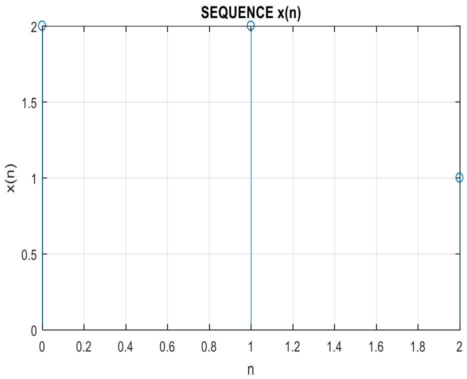
```
enter the first sequence[1,2,3]
enter the second sequence[2,3,4]
enter the third squnce[4,5,6]

z =

     8    38   111   190   229   162    72

     8    38   111   190   229   162    72


y =

     8    38   111   190   229   162    72

     8    38   111   190   229   162    72

convolution is associative
```

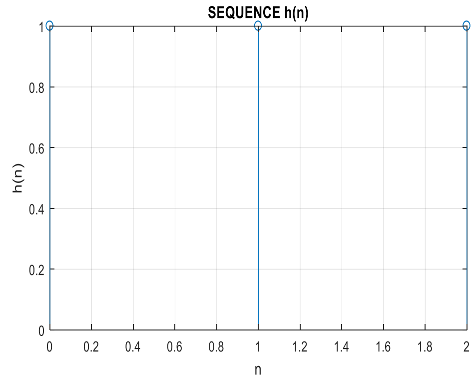
CIRCULAR CONVOLUTION

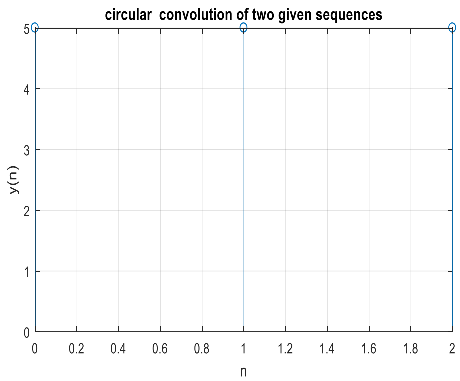
```
clc
clear all
x=input('enter the sequence x(n)= ')
a=0:length(x)-1;
subplot(2,2,1);
stem(a,x);
xlabel('n');
ylabel('x(n)');
title('SEQUENCE x(n)');
grid on;

h=input('Enter the sequence h(n)= ')
b=0:length(h)-1;
subplot(2,2,2);
stem(b,h);
xlabel('n');
ylabel('h(n)');
title('SEQUENCE h(n)');
grid on;

disp('circular  convolution of two given sequences ');
l=max(length(x),length(h));
y=cconv(x,h,l)
c=0:length(y)-1;
subplot(2,2,3);
stem(c,y);
xlabel('n');
ylabel('y(n)')
title('circular  convolution of two given sequences');
grid on;
```

OUTPUT

```
enter the sequence x(n)= [2,2,1]

x =

    2    2    1

Enter the sequence h(n)= [1,1,1]

h =

    1    1    1

circular  convolution of two given sequences

y =

    5    5    5
```


SEQUENCE x(n)
x(n)
n
0
0.5
1
1.5
2
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2


SEQUENCE h(n)
h(n)
n
0
0.2
0.4
0.6
0.8
1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2


circular convolution of two given sequences
y(n)
n
0
1
2
3
4
5
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

CROSS CORRELATION

```
clc
clear all
x=input('enter the sequence of x(n)');
a=0:length(x)-1;
subplot(2,2,1)
stem(a,x)
xlabel('n')
ylabel('x(n)')
title('sequence x(n)')
grid on
h=input('enter the sequence h(n)')
b=0:length(h)-1;
subplot(2,2,2)
stem(b,h)
xlabel('n')
ylabel('h(n)')
grid on
disp('cross correalation of the two given sequences')
y=xcorr(x,h)
c=-(floor(length(y)/2)):1:(floor(length(y)/2))
subplot(2,2,3)
stem(c,y)
xlabel('n')
ylabel('y(n)')
title('cross correlaton of the two given sequences')
grid on
```

OUTPUT

```
enter the sequence of x(n)[1,1,1]
enter the sequence h(n)[2,2,1]
```

h =

2 2 1

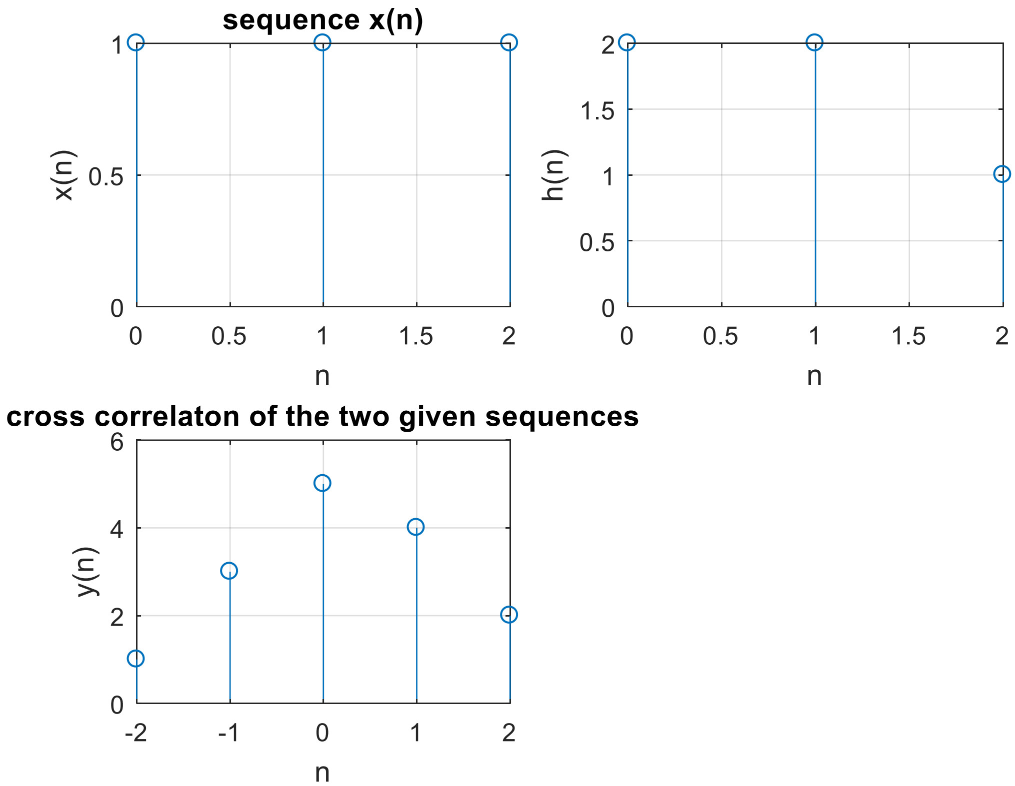
cross correalation of the two given sequences

y =

1.0000 3.0000 5.0000 4.0000 2.0000

c =

-2 -1 0 1 2


sequence x(n)
x(n)
n
h(n)
n
cross correlaton of the two given sequences
y(n)
n

AUTO CORRELATION

```
%1 get in sequence 1 & plot input sequence1
clc
clear all
x=input('enter the sequence x(n)=');
a=0:length(x)-1;
subplot(2,2,1);
stem(a,x);
xlabel('n');
ylabel('x(n)');
title('sequence x(n)');
grid on;
%2 find the autocorrelation of the given sequence and plot the o/p sequence
disp('auto correlation of given sequence');
Rxx=xcorr(x,x)
c=0:length(Rxx)-1;
subplot(2,2,2);
stem(c,Rxx);
xlabel('n');
ylabel('Rxx(n)');
title('autocorrelation of the given sequence');
grid on;
%verification of properties
%property 1 energy
energy=sum(x.^2)
centre_index=ceil(length(Rxx)/2)
Rxx_0=Rxx(centre_index)
if Rxx_0==energy
 disp('Rxx(0) gives energy_not proved');
else
   disp('Rxx(0) gives energy_not proved');
end
%property 2 even or odd
Rxx_right=Rxx(centre_index:1:length(Rxx))
Rxx_left=Rxx(centre_index:-1:1)
if Rxx_right==Rxx_left
   disp('Rxx is even');
else
   disp('Rxx is odd');
end
```

OUTPUT

enter the sequence x(n)=[1,2,3,4]

auto correlation of given sequence

Rxx =

4.0000 11.0000 20.0000 30.0000 20.0000 11.0000 4.0000

energy =

30

centre_index =

4

Rxx_0 =

30

Rxx(0) gives energy_not proved

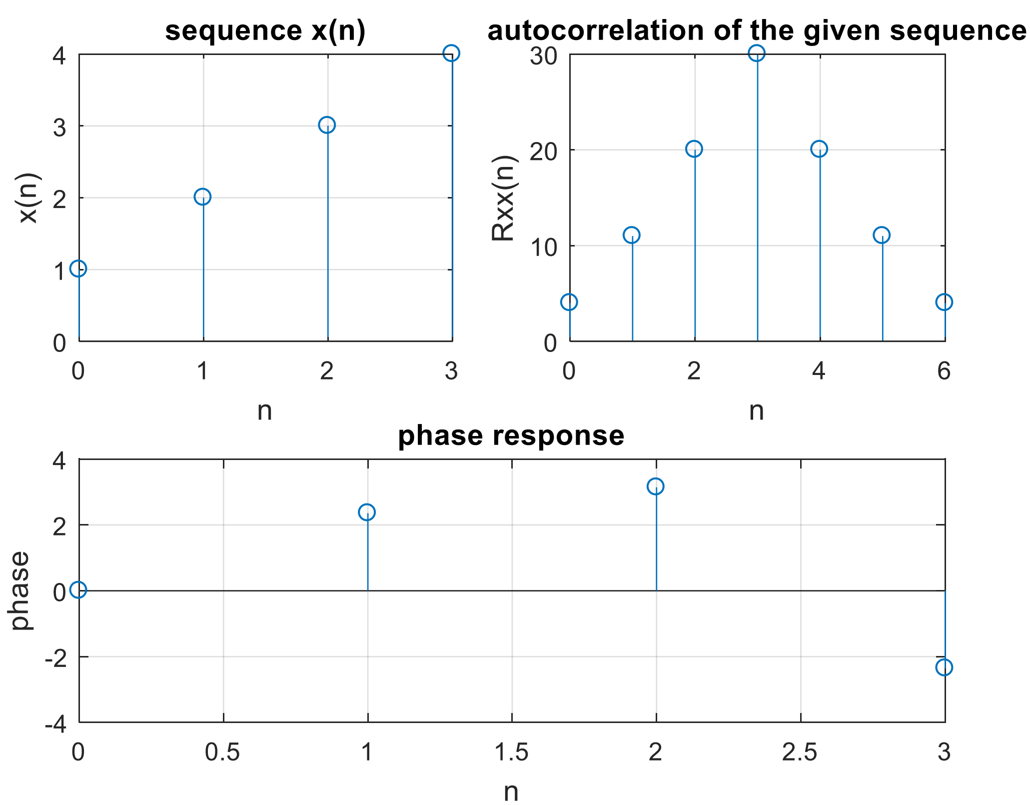
Rxx_right =

30.0000 20.0000 11.0000 4.0000

Rxx_left =

30.0000 20.0000 11.0000 4.0000

Rxx is even



DFT OF A SEQUENCE

```
clc
clear all
x=input('enter the sequnece');
L=length(x)
xk=fft(x,L)
y=ifft(xk,L)
magnitude=abs(xk)
subplot(2,1,1)
a=0:length(magnitude)-1;
stem(a,magnitude)
xlabel('n')
ylabel('magnitude')
title('magnitude response')
grid on
phase=angle(xk)
subplot(2,1,2)
b=0:length(phase)-1;
stem(a,phase)
xlabel('n')
ylabel('phase')
title('phase response')
grid on
```

OUTPUT

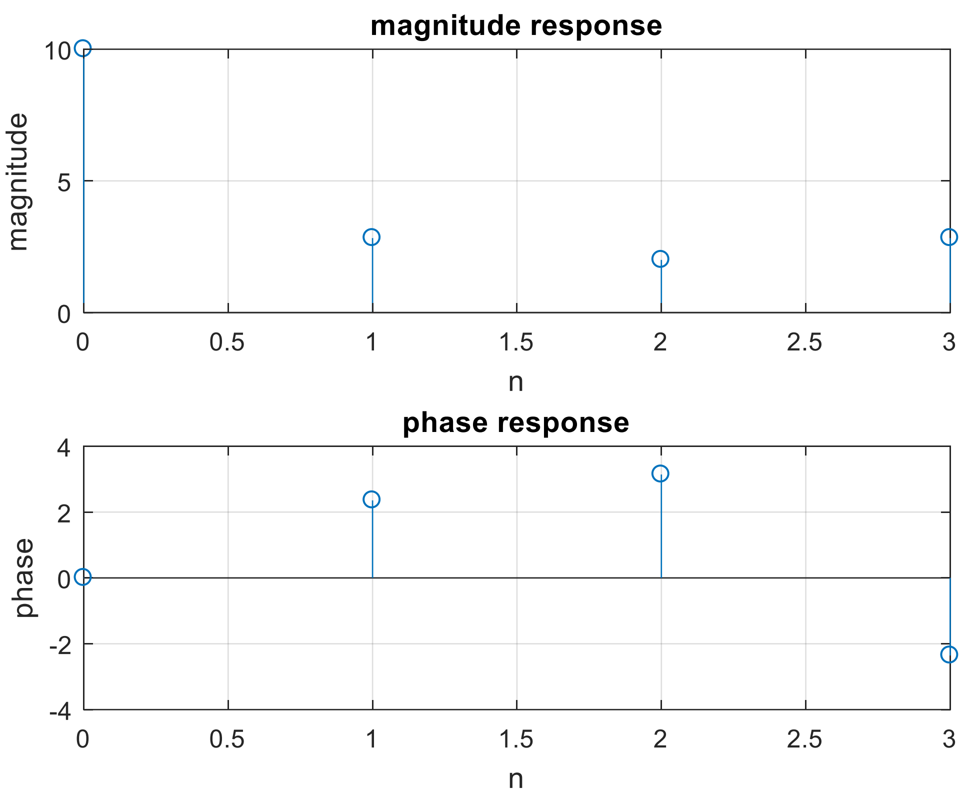
```
enter the sequnece[1 2 3 4 ]

L =

     4

xk =

  10.0000 + 0.0000i  -2.0000 + 2.0000i  -2.0000 + 0.0000i  -2.0000 - 2.0000i

y =

     1     2     3     4
```

magnitude =

10.0000 2.8284 2.0000 2.8284

phase =

0 2.3562 3.1416 -2.3562



DFT Without using INBUILT FUNCTIONS

```
clc
clear all
x=input('enter the sequence');
l=length(x);
n=input('enter how many point dft');
disp(n);
k1=0:1:n-1;
s=n-l;
xn=[x,zeros(1,s)];
xk=[zeros(1,n)];
for k=1:n
   for n=1:n
      xk(k)=xk(k)+xn(n)*exp(-(i*(2*3.142*(k-1)*(n-1)))/n);
   end
end
magnitude=abs(xk);
disp('magnitude of the sequence');
disp(magnitude);
subplot(2,1,1)
stem(k1,magnitude)
xlabel('k');
ylabel('x(k)');
title('magnitude plot');
phase=angle(xk);
disp('phase of the sequence');
disp(phase);
subplot(2,1,2)
stem(k1,phase)
xlabel('x(k)');
ylabel('angle in radians');
title('phase plot');
```

OUTPUT

Enter x(n) : [1,2,3,4]

x =

1 2 3 4

Enter how many point dft 4

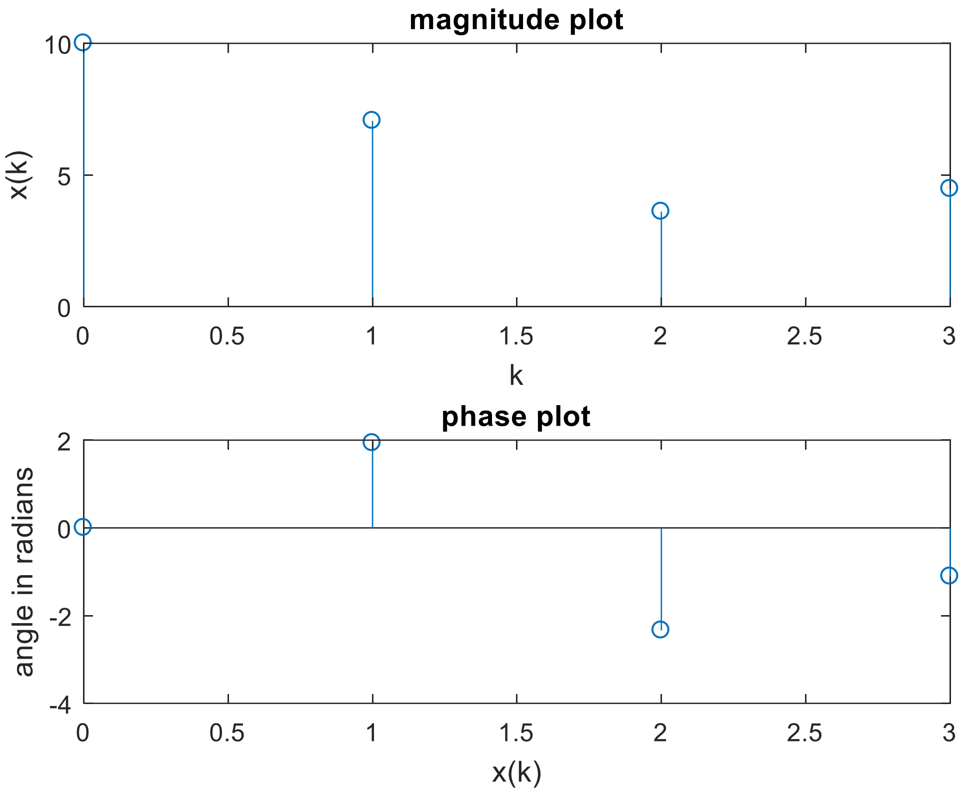
4

mag of the sequence

10.0000 2.8279 2.0000 2.8302

phase of the seq

0 2.3554 3.1400 -2.3586


magnitude plot
x(k)
k
10
5
0
0
0.5
1
1.5
2
2.5
3
phase plot
angle in radians
x(k)
2
0
-2
-4

DIFFERENTIAL EQUATION

```
clc
 clear all
 y=input('enter the y(n) coefficient');
 x=input('enter the x(n) coefficient');
 yic=[1/4];
 n=[0:5];
 disp('values of x(n)');
 xn=(1/2).^n
 disp('filter initial condition');
 xic=filtic(x,y,yic)
 disp('the solution for difference eqvation');
 h=filter(x,y,xn,xic)
 stem(n,h);
 xlabel('n');
 ylabel('magnitude');
 title('the difference eqvation for the given sequence with initial condition');
 grid on;
```

OUTPUT

enter the y(n) coefficient[1 -.5]

enter the x(n) coefficient1

values of x(n)

xn =

1.0000 0.5000 0.2500 0.1250 0.0625 0.0313

filter initial condition

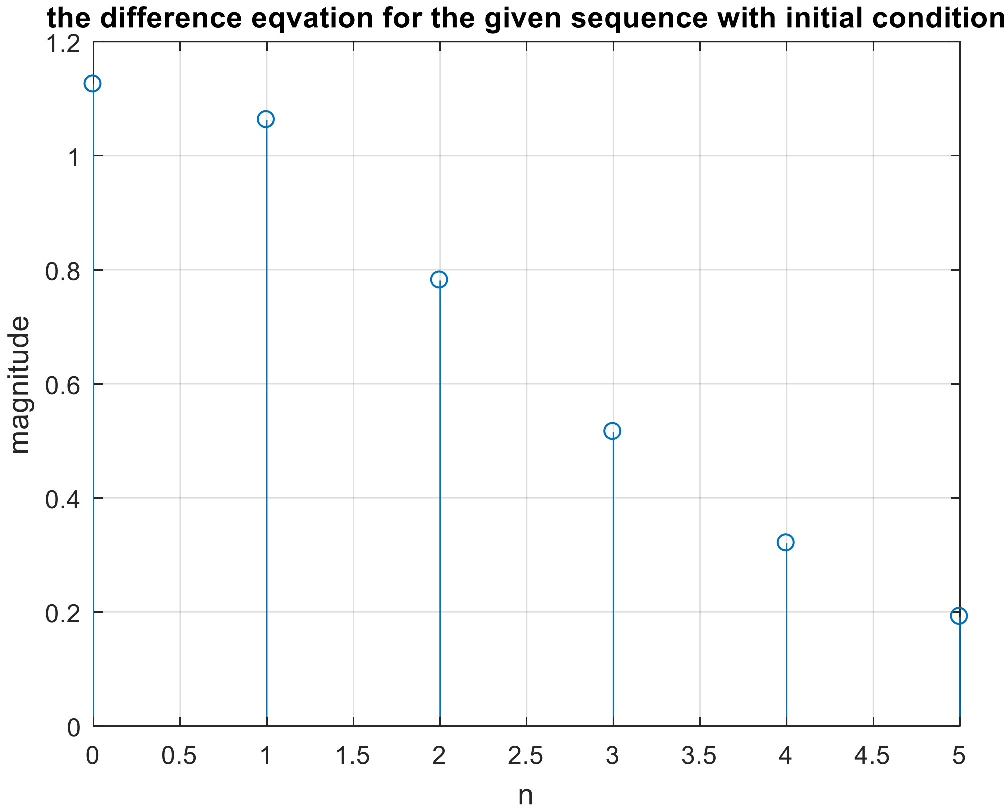
xic =

0.1250

the solution for difference eqvation

h =

1.1250 1.0625 0.7813 0.5156 0.3203 0.1914


the difference eqvation for the given sequence with initial condition
1.2
1
0.8
0.6
0.4
0.2
0
magnitude
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
n

Properties of DFT

Linear property

```
clc
clear all
a=input('Enter the variable a');
b=input('Enter the variable b');
x1n=input(' Enter the first sequnece');
x2n=input('Enter the second sequence');
N1=length(x1n);
N2=length(x2n);
x1k=fft(x1n);
x2k=fft(x2n);
y=a*x1n+b*x2n;
h=fft(y);
z=a*x1k+b*x2k;
disp('dft[a(x1n)+b(x2n)]=');
disp(h);
disp('a[(x1k)+b(x2k)]=');
disp(z);
if(h==z)
   disp('dft is linear');
else
   disp('dft is non linear');
end
```

OUTPUT

Enter the variable a3

Enter the variable b3

Enter the first sequnece[1,2,3]

Enter the second sequence[1,1,1]

dft[a(x1n)+b(x2n)]=

27.0000 + 0.0000i -4.5000 + 2.5981i -4.5000 - 2.5981i

a[(x1k)+b(x2k)]=

27.0000 + 0.0000i -4.5000 + 2.5981i -4.5000 - 2.5981i

dft is linear

PARSEVALS THEOREM

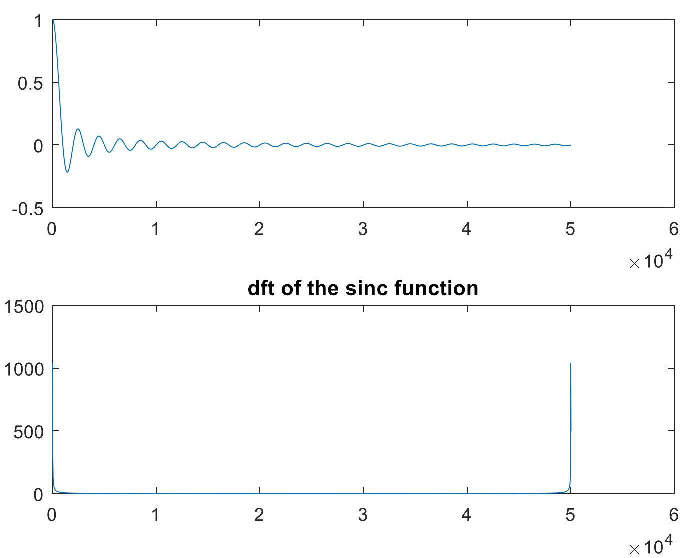
```
clc
clear all
x=input('enter the sequnece');
N=length(x);
Xc=conj(x);
Xk=fft(x,N);
Xkc=conj(Xk);
a=sum(x.*Xc);
b=sum((Xk.*Xkc)/N);
   if a==b
disp('parsevals theorem is proved');
   else
disp('parsevals theorem is not proved');
   end
```

OUTPUT

enter the sequence[1 2 3 4]
parsevals theorem is proved

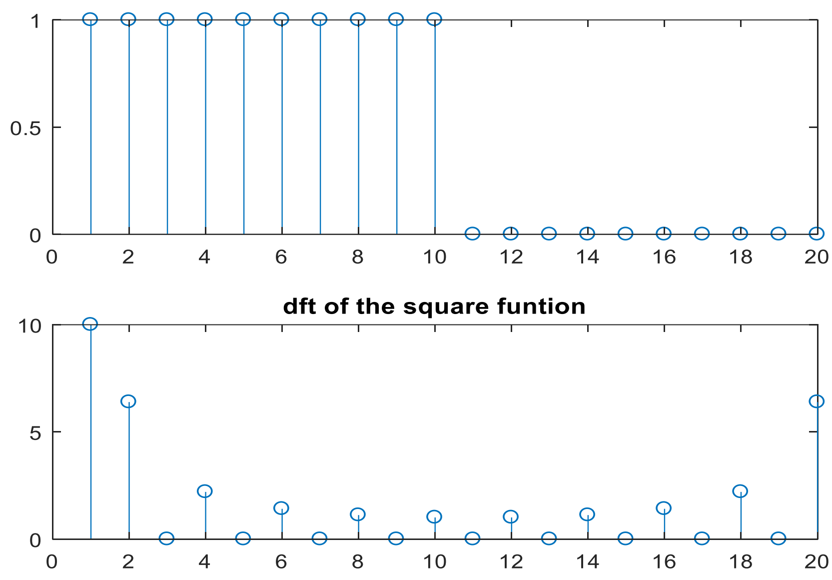
DFT of sinc function

```
clc
clear all
t=[0:.001:50];
y=sinc(t);
subplot(2,1,1);
plot(y);
y=fft(y);
subplot(2,1,2);
plot(abs(y));
title('dft of the sinc function');
```



DFT of square function

```
clc
clear all
y=[ones(1,10),zeros(1,10)];
subplot(2,1,1);
stem(y);
y=fft(y);
subplot(2,1,2);
stem(abs(y));
title('dft of the square funtion');
```



DESIGN AND IMPLEMENTATION OF FIR FILTER TO MEET GIVEN SPECIFICATIONS (LOW PASS FILTER USING HAMMING WINDOW)

```
clc
clear all
close all
Rp=input('enter the passband attenuation in dB')
Rs=input('enter the stopband attenuation in dB')
Wp=input('enter the passband attenuation in pi radians')
Ws=input('enter the stopband attenuation in pi radians')

%rectangular
if Rs<=21
    N=ceil(4/(Ws-Wp))
    y=boxcar(N);
    figure,
    subplot(2,1,1)
    stem(y)
    title('rectangular window')
end

%bartlett
if Rs>21 & Rs<=25
    N=ceil(8/(Ws-Wp))
    y=bartlett(N);
    figure,
    subplot(2,1,1)
    stem(y)
    title('batlett window')
end
```

```
%hanning
if Rs>25 & Rs<=44
   N=ceil(8/(Ws-Wp))
   y=hanning(N);
   figure,
   subplot(2,1,1)
   stem(y)
   title('hanning window')
end

%hamming
if Rs>44 & Rs<=53
   N=ceil(8/(Ws-Wp))
   y=hamming(N);
   figure,
   subplot(2,1,1)
   stem(y)
   title('hamming window')
end

%blackman
if Rs>53 & Rs<=74
   N=ceil(12/(Ws-Wp))
   y=blackman(N);
   figure,
   subplot(2,1,1)
   stem(y)
   title('blackman window')
end
```

```
B=fir1(N-1,Wp,y)
subplot(2,1,2)
stem(B)
title('impulse response of low pass FIR')
figure,
freqz(B,1)
title('freq response of FIR low pass filter')
```

OUTPUT

1) For rectangle window

```
enter the passband attenuation in dB : .25
Rp =
        0.2500
enter the stopband attenuation in dB : 18
Rs =
        18
enter the passband attenuation in pi radians : .2
Wp =
        0.2000
enter the stopband attenuation in pi radians : .3
Ws =
        0.3000
N =
        41
B =
 Columns 1 through 9
-0.0000   -0.0103   -0.0177   -0.0187   -0.0123    0.0000    0.0140    0.0245    0.0265

 Columns 10 through 18
  0.0179   -0.0000   -0.0218   -0.0398   -0.0454   -0.0328    0.0000    0.0492    0.1060

 Columns 19 through 27
```

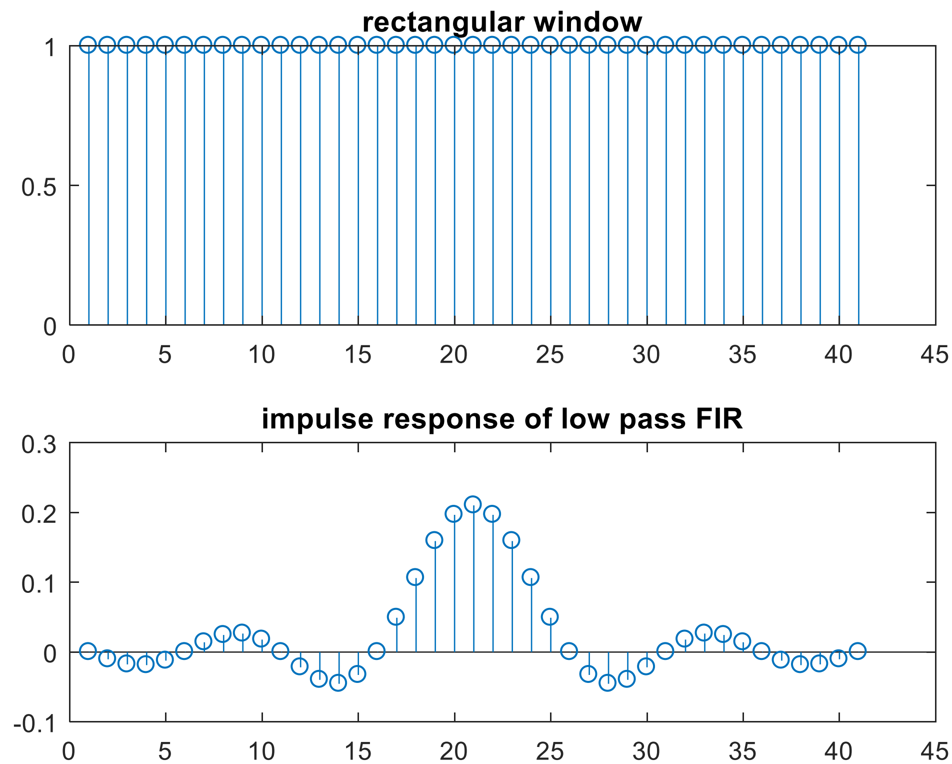
0.1591   0.1966   0.2102   0.1966   0.1591   0.1060   0.0492   0.0000   -0.0328

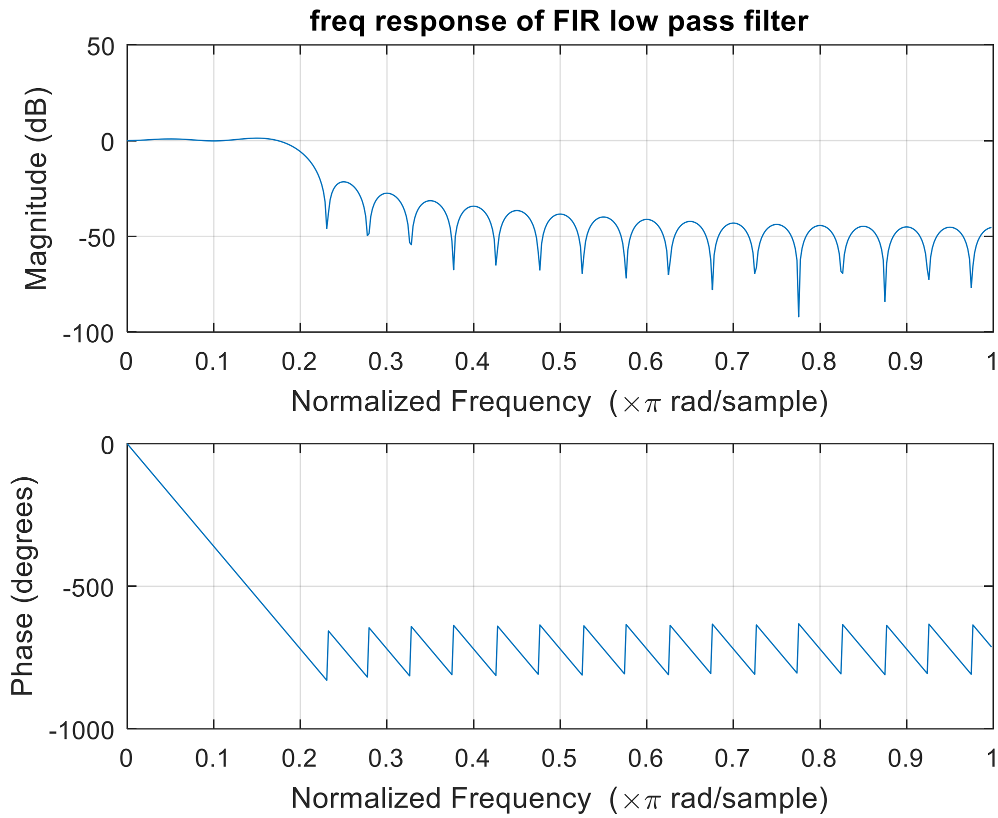
Columns 28 through 36

-0.0454   -0.0398   -0.0218   -0.0000   0.0179   0.0265   0.0245   0.0140   0.0000

Columns 37 through 41

-0.0123   -0.0187   -0.0177   -0.0103   -0.0000





2) For Bartlett window

enter the passband attenuation in dB : .25

Rp =

0.2500

enter the stopband attenuation in dB : 23

Rs =

23

enter the passband attenuation in pi radians : .2

Wp =

0.2000

enter the stopband attenuation in pi radians : .3

Ws =

0.3000

N =

81

B =

Columns 1 through 9

0 -0.0001 -0.0004 -0.0006 -0.0005 0.0000 0.0008 0.0016 0.0019

Columns 10 through 18

0.0014 -0.0000 -0.0018 -0.0033 -0.0037 -0.0026 0.0000 0.0032 0.0057

Columns 19 through 27

0.0063 0.0043 -0.0000 -0.0053 -0.0095 -0.0105 -0.0072 0.0000 0.0089

Columns 28 through 36

0.0161 0.0181 0.0126 -0.0000 -0.0165 -0.0310 -0.0366 -0.0272 0.0000

Columns 37 through 45

0.0432 0.0957 0.1474 0.1870 0.2050 0.1870 0.1474 0.0957 0.0432

Columns 46 through 54

0.0000 -0.0272 -0.0366 -0.0310 -0.0165 -0.0000 0.0126 0.0181 0.0161

Columns 55 through 63

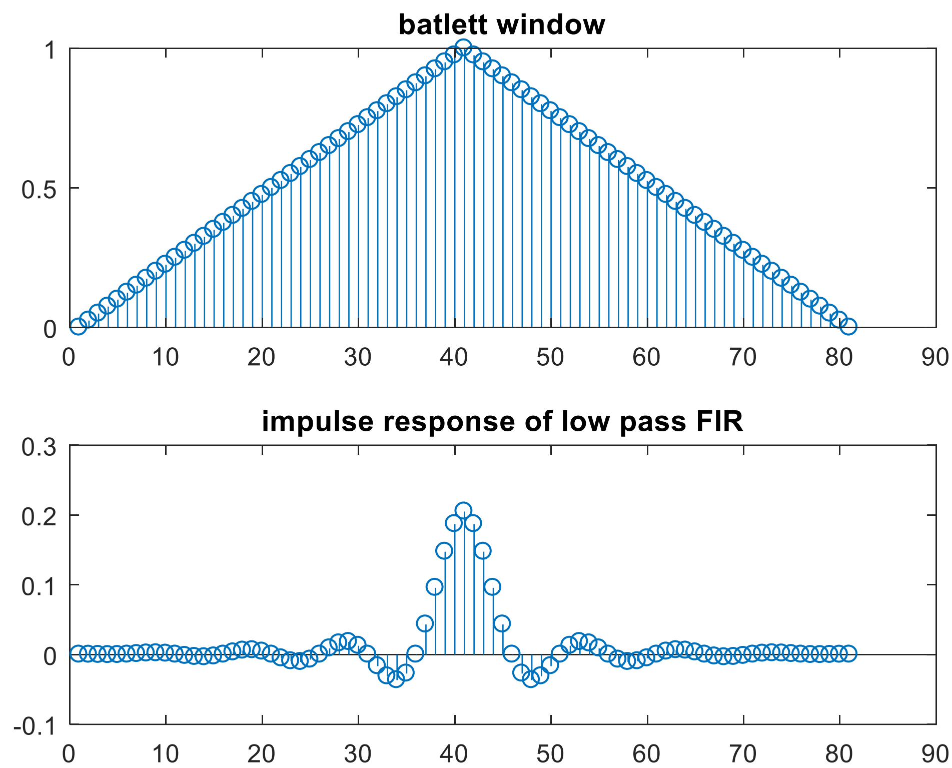
0.0089 0.0000 -0.0072 -0.0105 -0.0095 -0.0053 -0.0000 0.0043 0.0063

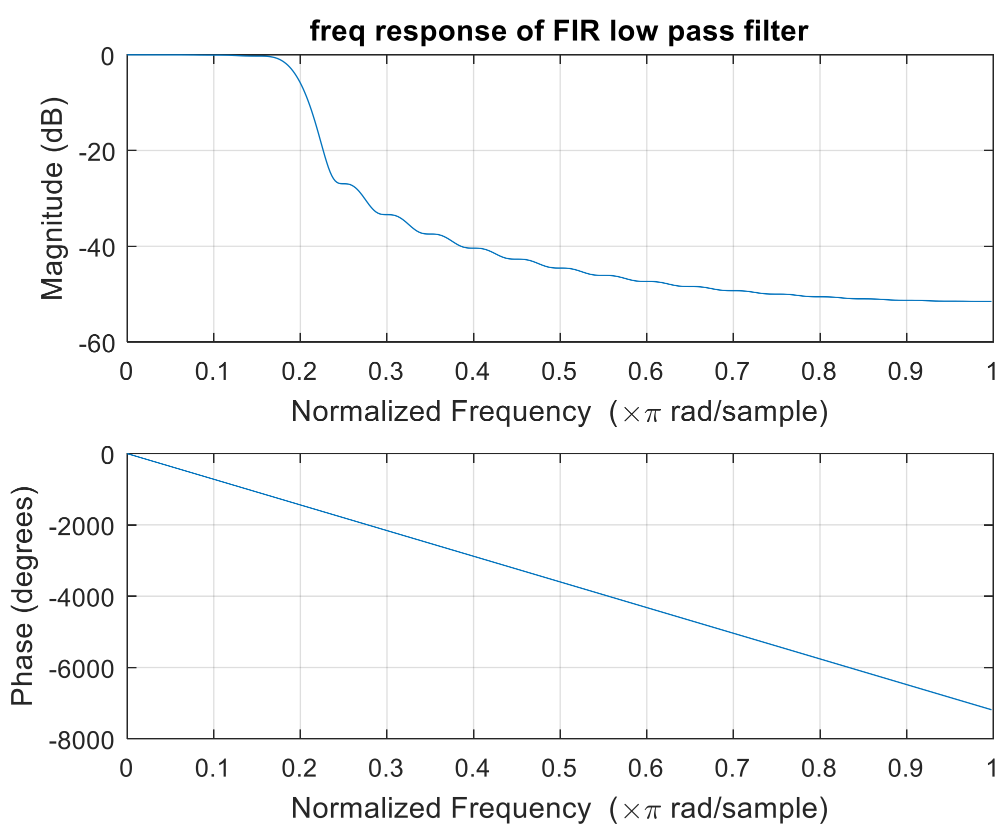
Columns 64 through 72

0.0057 0.0032 0.0000 -0.0026 -0.0037 -0.0033 -0.0018 -0.0000 0.0014

Columns 73 through 81

0.0019 0.0016 0.0008 0.0000 -0.0005 -0.0006 -0.0004 -0.0001 0


batlett window
impulse response of low pass FIR


freq response of FIR low pass filter
Magnitude (dB)
Normalized Frequency (×π rad/sample)
Phase (degrees)
Normalized Frequency (×π rad/sample)

3) For hanning window

enter the passband attenuation in dB : .25

Rp =

0.2500

enter the stopband attenuation in dB : 38

Rs =

38

enter the passband attenuation in pi radians : .2

Wp =

0.2000

enter the stopband attenuation in pi radians : .3

Ws =

0.3000

N =

81

B =

Columns 1 through 9

-0.0000 -0.0000 -0.0001 -0.0002 -0.0002 0.0000 0.0004 0.0008 0.0011

Columns 10 through 18

0.0008 -0.0000 -0.0013 -0.0025 -0.0029 -0.0021 0.0000 0.0029 0.0053

Columns 19 through 27

0.0061 0.0043 -0.0000 -0.0055 -0.0100 -0.0113 -0.0078 0.0000 0.0099

Columns 28 through 36

0.0180 0.0203 0.0142 -0.0000 -0.0184 -0.0344 -0.0402 -0.0296 0.0000

Columns 37 through 45

0.0457 0.0996 0.1505 0.1868 0.2000 0.1868 0.1505 0.0996 0.0457

Columns 46 through 54

0.0000 -0.0296 -0.0402 -0.0344 -0.0184 -0.0000 0.0142 0.0203 0.0180

Columns 55 through 63

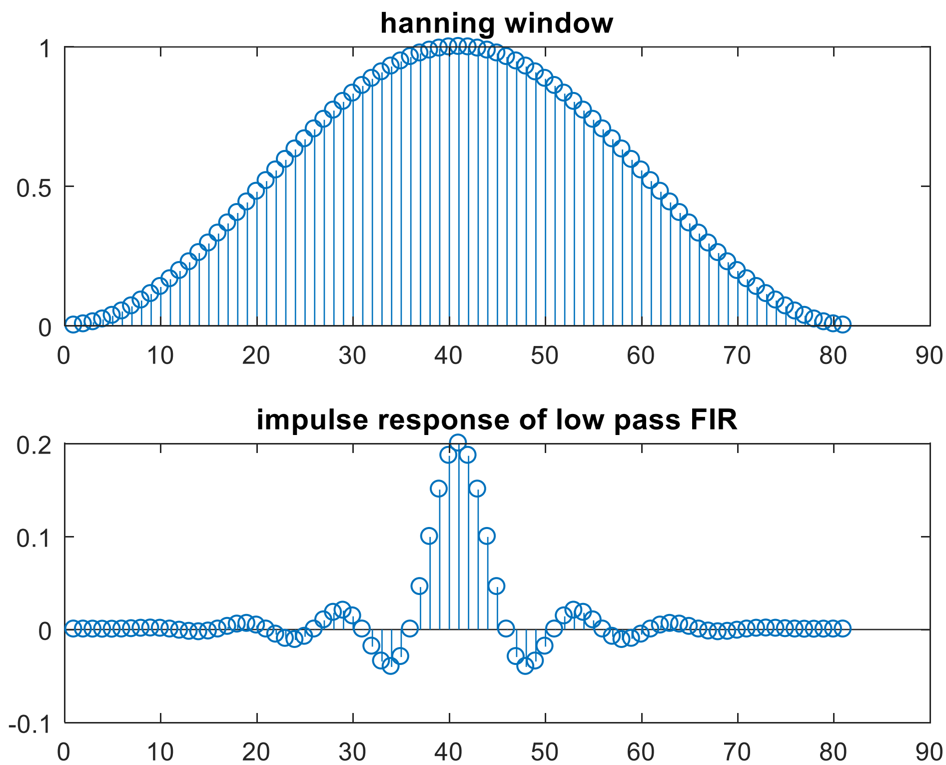
0.0099 0.0000 -0.0078 -0.0113 -0.0100 -0.0055 -0.0000 0.0043 0.0061

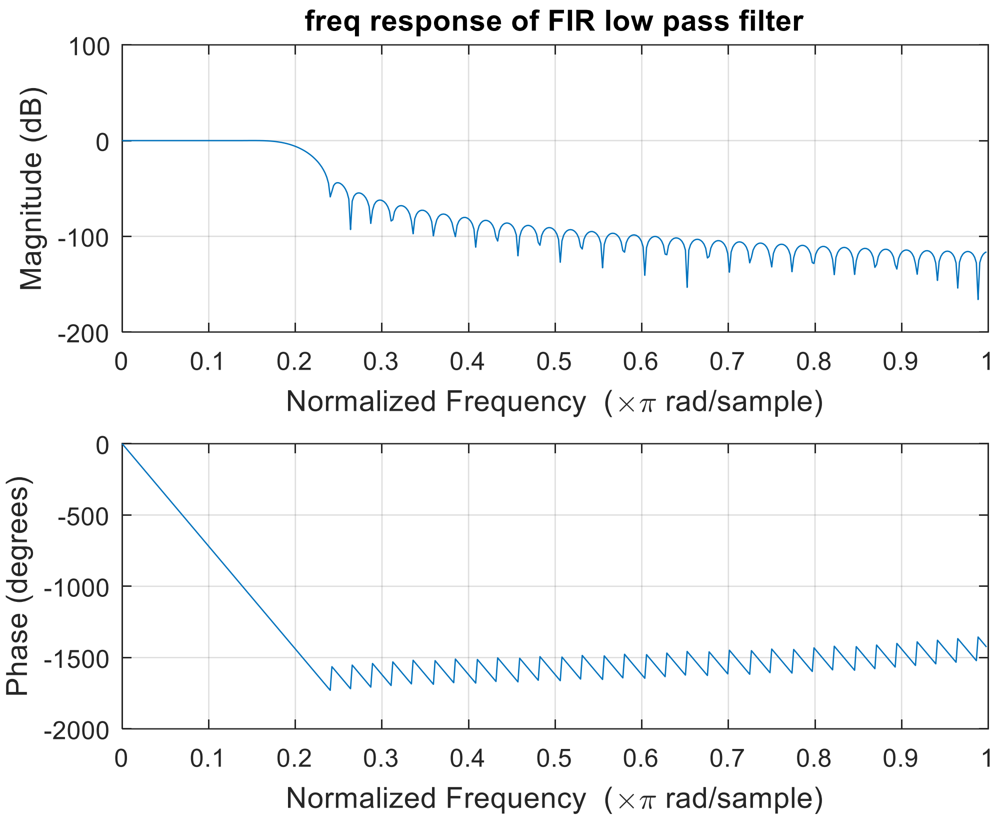
Columns 64 through 72

0.0053 0.0029 0.0000 -0.0021 -0.0029 -0.0025 -0.0013 -0.0000 0.0008

Columns 73 through 81

0.0011 0.0008 0.0004 0.0000 -0.0002 -0.0002 -0.0001 -0.0000 -0.0000





4) For hamming window

enter the passband attenuation in dB : .25

Rp =

0.2500

enter the stopband attenuation in dB : 49

Rs =

49

enter the passband attenuation in pi radians : .2

Wp =

0.2000

enter the stopband attenuation in pi radians : .3

Ws =

0.3000

N =

81

B =

Columns 1 through 9

-0.0000	-0.0004	-0.0007	-0.0008	-0.0005	0.0000	0.0007	0.0014	0.0016

Columns 10 through 18

0.0012	-0.0000	-0.0016	-0.0029	-0.0034	-0.0024	0.0000	0.0031	0.0057

Columns 19 through 27

0.0065	0.0045	-0.0000	-0.0057	-0.0103	-0.0115	-0.0080	0.0000	0.0100

Columns 28 through 36

0.0182	0.0205	0.0143	-0.0000	-0.0185	-0.0346	-0.0404	-0.0297	0.0000

Columns 37 through 45

0.0458	0.0998	0.1508	0.1872	0.2004	0.1872	0.1508	0.0998	0.0458

Columns 46 through 54

0.0000	-0.0297	-0.0404	-0.0346	-0.0185	-0.0000	0.0143	0.0205	0.0182

Columns 55 through 63

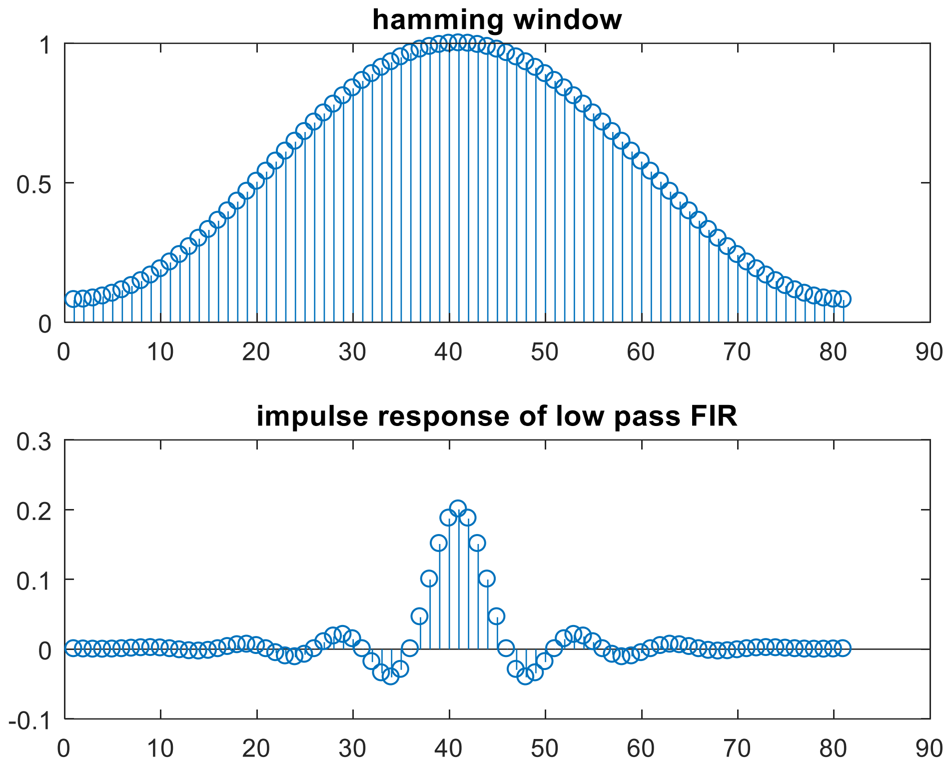
0.0100	0.0000	-0.0080	-0.0115	-0.0103	-0.0057	-0.0000	0.0045	0.0065

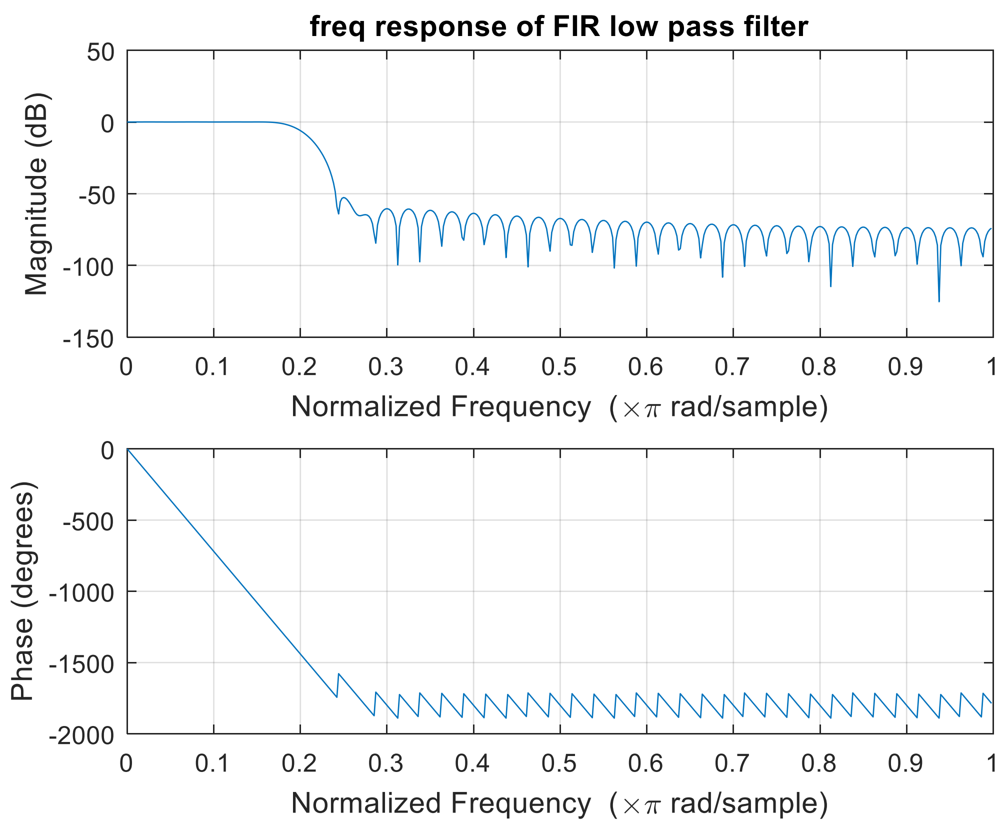
Columns 64 through 72

0.0057	0.0031	0.0000	-0.0024	-0.0034	-0.0029	-0.0016	-0.0000	0.0012

Columns 73 through 81

0.0016	0.0014	0.0007	0.0000	-0.0005	-0.0008	-0.0007	-0.0004	-0.0000


hamming window
1
0.5
0
0
10
20
30
40
50
60
70
80
90
impulse response of low pass FIR
0.3
0.2
0.1
0
-0.1
0
10
20
30
40
50
60
70
80
90


freq response of FIR low pass filter
Magnitude (dB)
50
0
-50
-100
-150
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Normalized Frequency (×π rad/sample)
Phase (degrees)
0
-500
-1000
-1500
-2000
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Normalized Frequency (×π rad/sample)

5) For blackman window

enter the passband attenuation in dB : .25

Rp =

0.2500

enter the stopband attenuation in dB : 60

Rs =

60

enter the passband attenuation in pi radians : .2

Wp =

0.2000

enter the stopband attenuation in pi radians : .3

Ws =

0.3000

N =

121

B =

Columns 1 through 9

0 -0.0000 -0.0000 -0.0000 -0.0000 0.0000 0.0000 0.0001 0.0001

Columns 10 through 18

0.0001 -0.0000 -0.0001 -0.0003 -0.0003 -0.0002 0.0000 0.0003 0.0006

Columns 19 through 27

0.0007 0.0005 -0.0000 -0.0007 -0.0013 -0.0015 -0.0010 0.0000 0.0013

Columns 28 through 36

0.0024 0.0027 0.0019 -0.0000 -0.0024 -0.0043 -0.0047 -0.0032 0.0000

Columns 37 through 45

0.0040 0.0071 0.0078 0.0053 -0.0000 -0.0065 -0.0116 -0.0128 -0.0087

Columns 46 through 54

0.0000 0.0107 0.0192 0.0214 0.0148 -0.0000 -0.0190 -0.0352 -0.0409

Columns 55 through 63

-0.0299 0.0000 0.0459 0.0999 0.1507 0.1869 0.2000 0.1869 0.1507

Columns 64 through 72

0.0999 0.0459 0.0000 -0.0299 -0.0409 -0.0352 -0.0190 -0.0000 0.0148

Columns 73 through 81

0.0214 0.0192 0.0107 0.0000 -0.0087 -0.0128 -0.0116 -0.0065 -0.0000

Columns 82 through 90

0.0053 0.0078 0.0071 0.0040 0.0000 -0.0032 -0.0047 -0.0043 -0.0024

Columns 91 through 99

-0.0000 0.0019 0.0027 0.0024 0.0013 0.0000 -0.0010 -0.0015 -0.0013

Columns 100 through 108

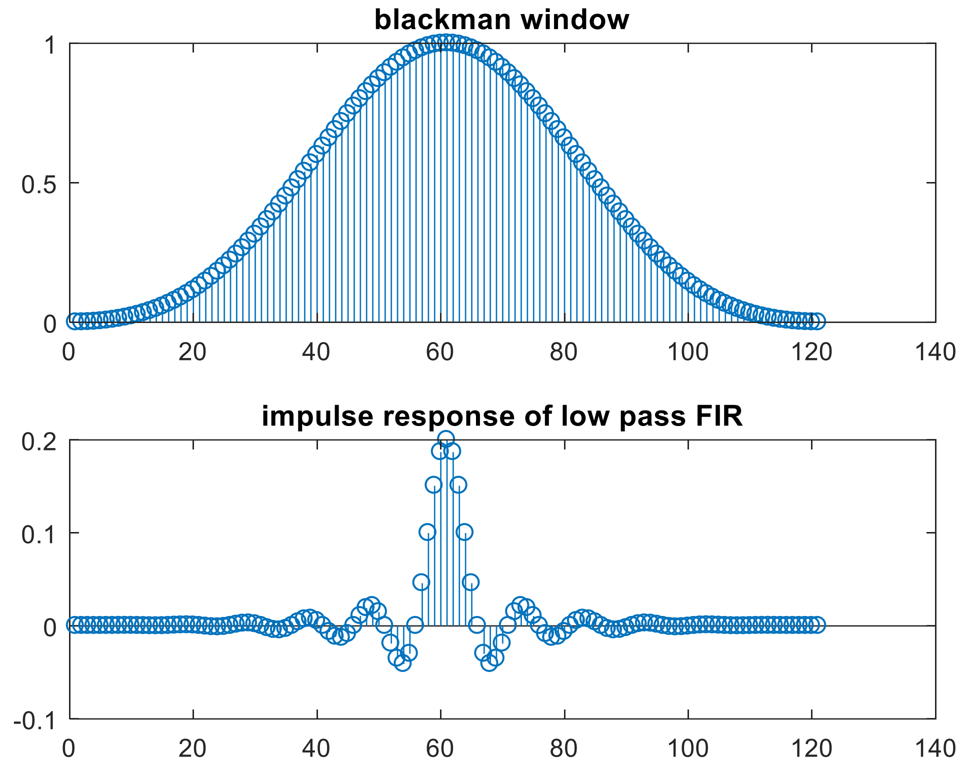
-0.0007 -0.0000 0.0005 0.0007 0.0006 0.0003 0.0000 -0.0002 -0.0003

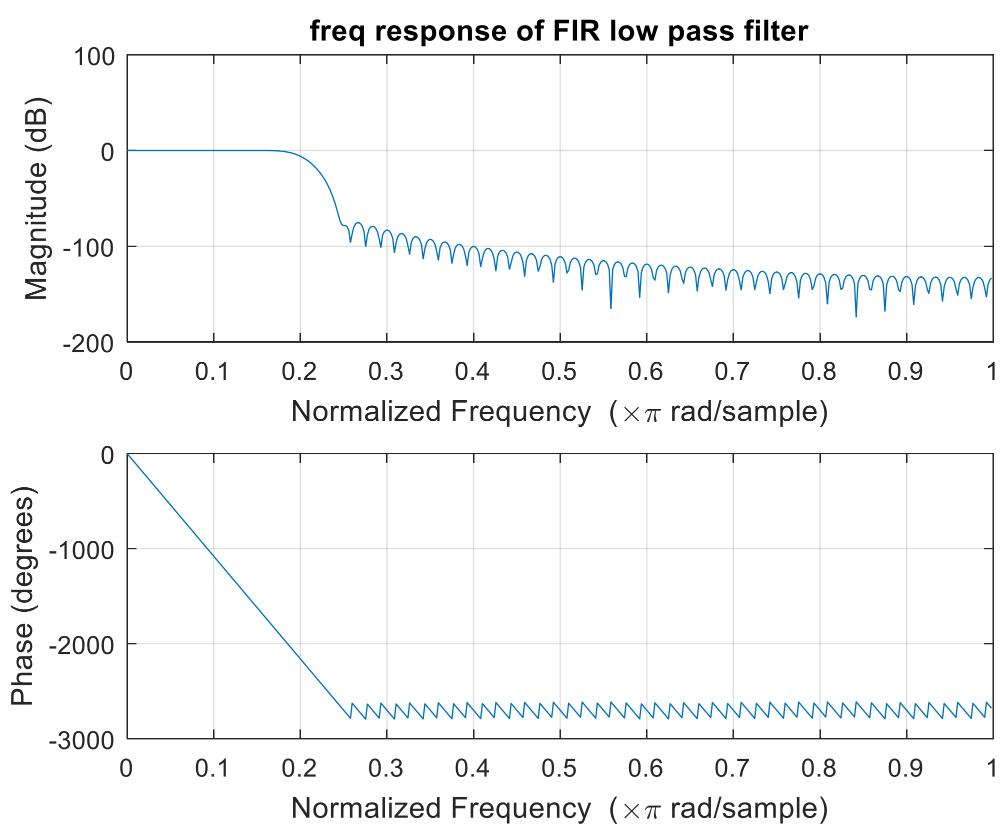
Columns 109 through 117

-0.0003 -0.0001 -0.0000 0.0001 0.0001 0.0001 0.0000 0.0000 -0.0000

Columns 118 through 121

-0.0000 -0.0000 -0.0000 0


blackman window
1
0.5
0
0 20 40 60 80 100 120 140
impulse response of low pass FIR
0.2
0.1
0
-0.1
0 20 40 60 80 100 120 140


freq response of FIR low pass filter
Magnitude (dB)
100
0
-100
-200
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Normalized Frequency (×π rad/sample)
Phase (degrees)
0
-1000
-2000
-3000
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Normalized Frequency (×π rad/sample)

DESIGN AND IMPLEMENTATION OF IIR FILTER TO MEET GIVEN SPECIFICATIONS

```
clc;
clear all;
close all;

Ap= input('Enter the passband edge ripple in dB : ');
As= input('Enter the stopband attenuation in dB : ');
Wp= input('Enter the passband edge frequency in rad : ');
Ws= input('Enter the stopband edge frequency in rad : ');

Fs= input('Enter the sampling frequency in Hz : ');

Wppre = 2*tan(Wp/2)
Wspre = 2*tan(Ws/2)

[N,Wc]=buttord(Wppre,Wspre,Ap,As,'s')

[num,den]=butter(N,Wc,'low','s')
HS = tf(num,den)

[B,A]=bilinear(num,den,1)
HZ=tf(B,A,1)
```

OUTPUT

```
Enter the passband edge ripple in dB : 3.01

Enter the stopband attenuation in dB : 15

Enter the passband edge frequency in rad : .5*pi

Enter the stopband edge frequency in rad : .75*pi

Enter the sampling frequency in Hz : 2000


Wppre =


    2.0000



Wspre =


    4.8284
```

```
N =

     2


Wc =

    2.0526


num =

         0         0    4.2130


den =

    1.0000    2.9027    4.2130


HS =

         4.213
  ---------------------
  s^2 + 2.903 s + 4.213

Continuous-time transfer function.


B =

    0.3005    0.6011    0.3005
```

```
A =

    1.0000    0.0304    0.1717


HZ =

  0.3005 z^2 + 0.6011 z + 0.3005
  ------------------------------
     z^2 + 0.03039 z + 0.1717

Sample time: 1 seconds
Discrete-time transfer function.
```

PART B

Experiment No: 1 **LINEAR CONVOLUTION OF THE GIVEN TWO SEQUENCE**

```
#include<stdio.h>
#include<math.h>
int y[20];

main()
{
    int m=6;                          /*Lenght of i/p samples sequence*/
    int n=6;                          /*Lenght of impulse response Coefficients*/
    int i=0,j;
    int x[15]={1,2,3,4,5,6};          /*Input Signal Samples*/
    int h[15]={1,2,3,4,5,6};          /*Impulse Response Co-efficients*/

    for(i=0;i<m+n-1;i++)
    {
    y[i]=0;
    for(j=0;j<=i;j++)
    y[i]+=x[j]*h[i-j];
    }

    printf("Linear Convolution\n");
    for(i=0;i<m+n-1;i++)
    printf("%d\n",y[i]);
}
```

Experiment No: 2 **CIRCULAR CONVOLUTION OF THE GIVEN TWO SEQEUNCE**

```
#include<stdio.h>
int m=4,n=4,x[30],h[30],y[30],i,j,k;
void main()
{
	int x[4]={1,2,3,4};
	int h[4]={1,1,1,1};
	for(k=0;k<n;k++)
	{
		y[k]=0;
		i=k;
		for(j=0;j<n;j++)
		{
			y[k]+=x[i]*h[j];
			i--;
			if(i==-1)
			i=n-1;
		}
		printf("%d\n",y[k]);
	}
}
```

Experiment No: 3 IMPULSE RESPONSE OF A GIVEN SYSTEM

```
#include <stdio.h>

#define Order 1

#define Len     6


float y[Len]={0,0,0},sum;


void main()

{

    int j,k;

    float a[Order+1]={2};

    float b[Order+1]={1,-0.5};

    for(j=0;j<Len;j++)

    {

        sum=0;

        for(k=1;k<=Order;k++)

        {

                if((j-k)>=0)

                sum=sum+(b[k]*y[j-k]);

        }

        if(j<=Order)

        {

                y[j]=a[j]-sum;

        }

        else

        {

                y[j]=-sum;

        }

       printf(" %f\n", y[j]);


    }

}
```

Experiment No: 4 **DFT OF THE GIVEN SEQUENCE**

```
#include<stdio.h>
#include<math.h>
float pi=3.1416,out_real[4]={0.0}, out_imag[4]={0.0};

void main(void)
{
	int N=4,k,n;
	int x[4]={1,2,3,4};
	float sumre=0, sumim=0,
	for(k=0;k<N;k++)
	{
		sumre=0;
		sumim=0;
		for(n=0;n<N;n++)
		{
			sumre=sumre+x[n]* cos(2*pi*k*n/N);
			sumim=sumim-x[n]* sin(2*pi*k*n/N);
		}
		out_real[k]=sumre;
		out_imag[k]=sumim;
		printf("X([%d])  %7.3f  %7.3f\n",k,out_real[k],out_imag[k]);
	}
}
```

Structured Enquiry

Program for realization in Cascade and Parellel.

```
//parallel realization
clc ;
clear ;
close ;
z=poly(0,'z');
s=poly(0,'s') ; //system function
Hp=(3*z*5*z-2)/(z+1/2)(3*z-1) ;
H=pfss(Hp/z) disp('//parallel realization factors//')
disp(H) //cascade realization
z=poly(0,'z') //system function
Hc=(z^3+2*z^2+4*z+5)/(z^4+3*z^3+2*z+1)
disp('//Cascade realization num and den factors//')
n=factors(Hc('num')) disp(n,'/Numerator factors/')
nr=roots(Hc('num')) disp(n,'/Denominator factors/')
d=factors(Hc('den'))
nd=roots(Hc('den'))
```