MATLAB Programs

Program 1

Write a program to design an FIR filter using a rectangular window.

```
%FIR filter design using rectangular window
clear all;
fs=8000;t=0.0025;
Q=t*fs/2;
Q=10;
disp(Q);
for i=1:Q,
    x(i)=(sin(0.25*pi*i))/(i*pi);
end
a(Q+1)=0.25;
for i=1:Q,
    a(i)=x(Q-(i-1));
end
for i=2:Q+1
a(i+Q)=x(i-1);
end
w1=window(@rectwin,21)
for i=1:21,
    a1(i)=a(i)*w1(i);
end
stem(a1);xlabel('coefficient number'); ylabel('amplitude');
title('impluse response of the filter using rectangular
window');
for i=1:4000
sum(i)=0.0;
end
figure;
[h,w]=freqz(a1,1,256,8000);
plot(w/pi,20*log10(abs(h)));xlabel('normalized frequency');
ylabel('ampliudein dB');title('magnitude response plot-
rectangular window');
figure;
plot(w/pi,(atand(imag(h)/real(h))));xlabel('normalized
frequency');ylabel('angle in radians');title('Phase
response plot-kaise window');
freqz(a1,1,256,8000);title('magnitude and phase plot-
rectangular window');
```

Output

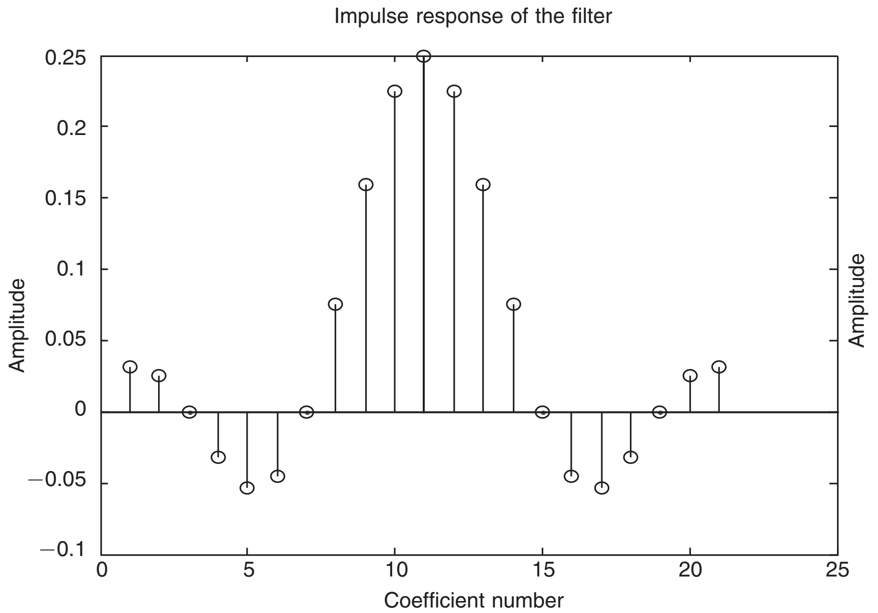

Figure 1 Impulse response of 21 coefficients FIR filter using rectangular window.

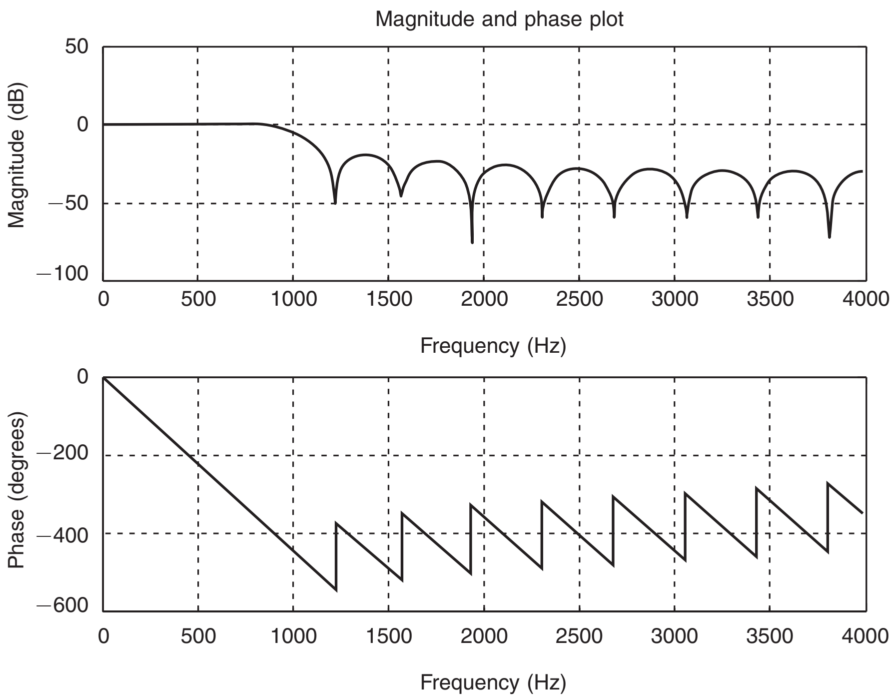

Figure 2 Magnitude and phase plot for FIR filter with 21 coefficients using rectangular window.

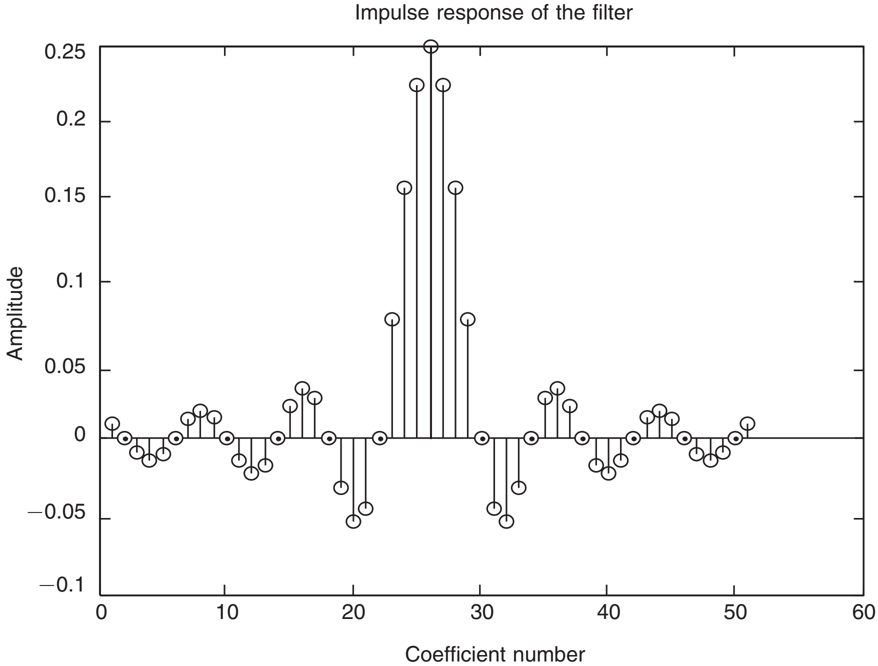

Figure 3 Impulse response of the FIR filter using 51 coefficients.

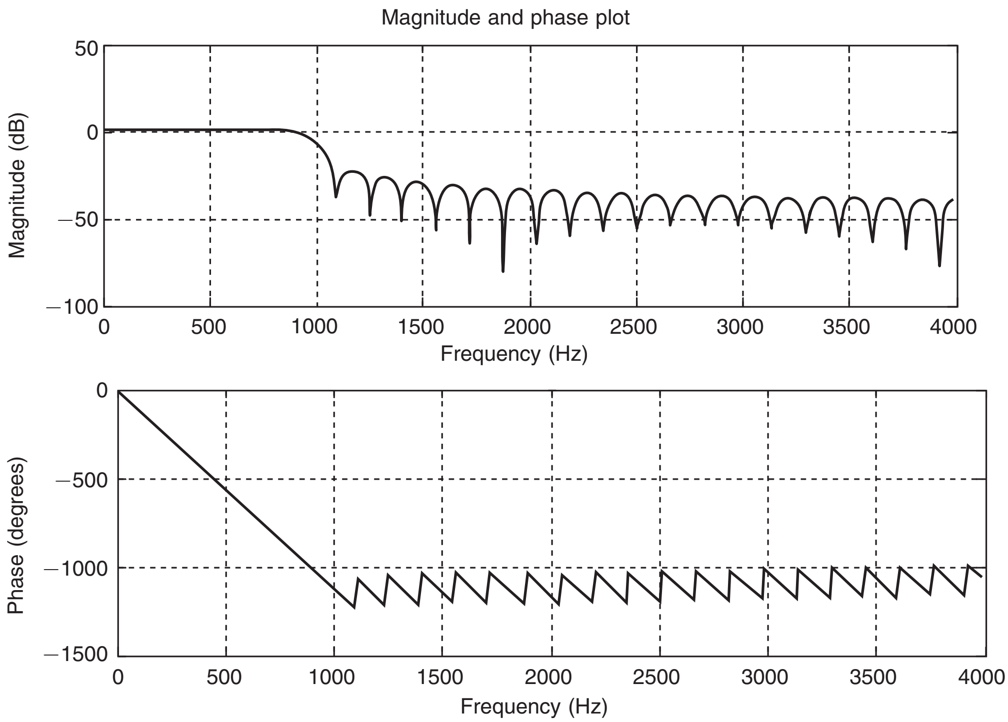

Figure 4 Magnitude and phase plot for 51 coefficients FIR filter.

Program 2

Write a program to design FIR filter using Hamming window.

```
%FIR filter design using Hamming window
clear all;
fs=8000;t=0.0025;
Q=t*fs/2;
disp(Q);
for i=1:Q,
    x(i)=(sin(0.25*pi*i))/(i*pi);
end
a(Q+1)=0.25;
for i=1:Q,
    a(i)=x(Q-(i-1));
end
disp(a);
for i=2:Q+1
a(i+Q)=x(i-1);
end
disp(a);
w1=window(@Hamming,21);
for i=1:21,
    a1(i)=a(i)*w1(i);
end
stem(a1);xlabel('coefficient number'); ylabel('amplitude');
title('impluse response of the filter using rectangular
window');
for i=1:4000
sum(i)=0.0;
end
figure;
[h,w1]=freqz(a1,1,256,8000);
plot(w1/pi,20*log10(abs(h)));xlabel('normalized frequency');
ylabel('ampliudein dB');title('magnitude response
plot-rectangular window');
figure;
plot(w1/pi,(atand(imag(h)/real(h))));xlabel('normalized
frequency');ylabel('angle in radians');title('Phase
response plot-kaise window');
freqz(a1,1,256,8000);title('magnitude and phase plot-
rectangular window');
```

Output

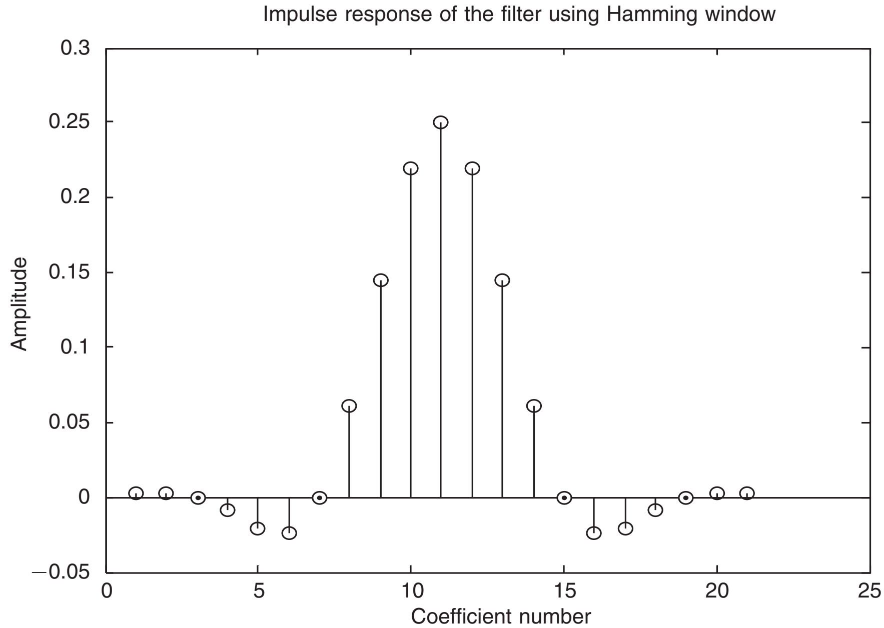

Figure 5 Plot of windowed coefficients using Hamming window.

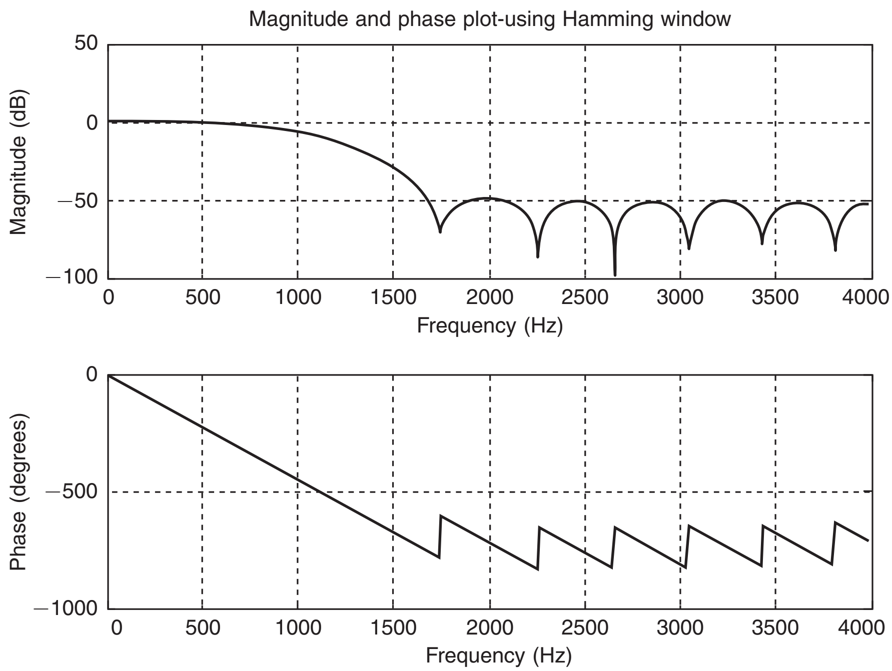

Figure 6 Magnitude and phase plot using Hamming window.

Program 3

Write a program to design a filter using Kaiser window.

```
%FIR filter design using kaiser window
clear all;
d=40;
fs=8000;
N=(d-8)/(2.285*0.2*pi);
Q=floor((N+1)/2);
Q=Q+1;
disp(Q);
b=0.5842*(d-21)^0.4+0.07886*(d-21);
disp(b);
disp(Q);
for i=1:Q,
    x(i)=(sin(0.4*pi*i))/(i*pi);
end
a(Q+1)=0.4;
for i=1:Q,
    a(i)=x(Q-(i-1));
end
for i=2:Q+1
a(i+Q)=x(i-1);
end
w5 = window(@kaiser,2*Q+1,b);
for i=1:25,
    z(i)=a(i)*w5(i);
end
subplot(3,1,1);
stem(a); xlabel('coefficient number');ylabel('amplitude');
title('impluse response of the filter using FS method');
subplot(3,1,2);
stem(w5);xlabel('coefficient number'); ylabel('amplitude');
title('Kaiser window coefficients');
subplot(3,1,3);
stem(z);xlabel('coefficient number'); ylabel('amplitude');
title('impluse response of the filter using Kaiser window');
```

```
for i=1:4000
sum(i)=0.0;
end
figure;
[h,w]=freqz(z,1,256,8000);
plot(w/pi,20*log10(abs(h)));xlabel('normalized frequency');
ylabel('ampliudein dB');title('magnitude response plot');
figure;
plot(w/pi,(atand(imag(h)/real(h))));xlabel('normalized
frequency');ylabel('angle  in  radians');title('Phase
response plot');
freqz(z,1,256,8000);title('magnitude and phase plot-using
Kaiser window');
```

Output

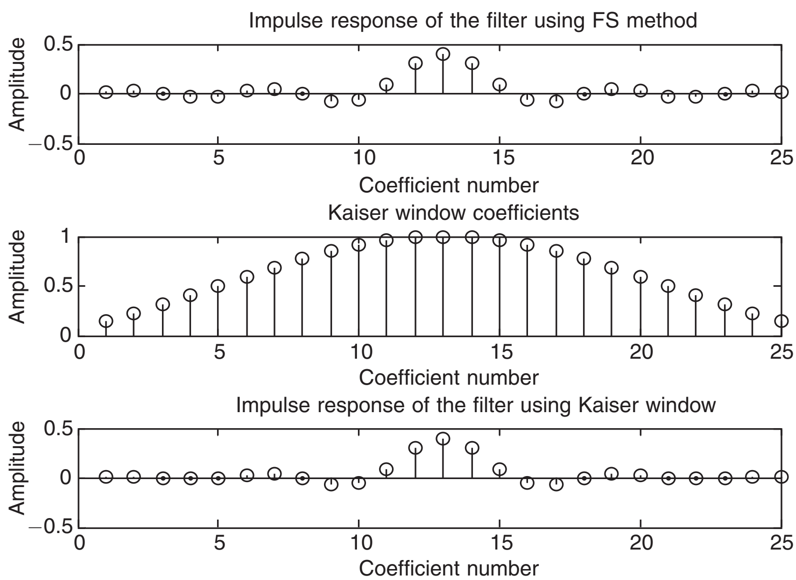

Figure 7 Plot of “*a*”, “*w*” and “*W*” coefficients.

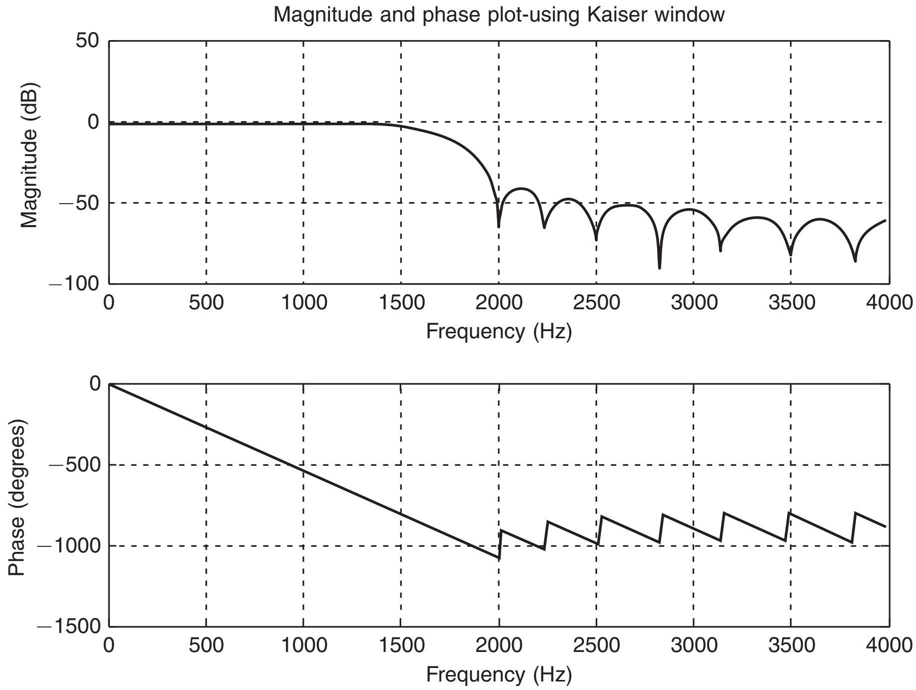

Figure 8 Magnitude and phase plot.

Program 4

Write a program to design HPF using FS expansion method.

```
%HPF design using FS expansion method
clear all;
fs=8000;
t=0.0025;
Q=fs/2*t;
disp(Q);
for i=1:10,
    x(i)=(sin(pi*i))/(i*pi)-(sin(0.25*pi*i))/(i*pi);
end
a(11)=0.75;
for i=1:10,
    a(i)=x(Q-(i-1));
end
for i=2:11
a(i+Q)=x(i-1);
end
```

```
disp(a);
stem(a);xlabel('coefficient number'); ylabel('amplitude');
title('impluse response of the HPF filter');
for i=1:4000
sum(i)=0.0;
end
figure;
[h,w]=freqz(a,1,256,8000);
plot(w/pi,20*log10(abs(h)));xlabel('normalized frequency');
ylabel('ampliudein dB');title('magnitude response plot');
figure;
plot(w/pi,(mod(angle(h), pi)*180/pi));xlabel('normalized
frequency');ylabel('angle in radians');title('Phase
response plot');
figure;
freqz(a,1,256,8000);title('magnitude and phase plot for
HPF');
```

Output

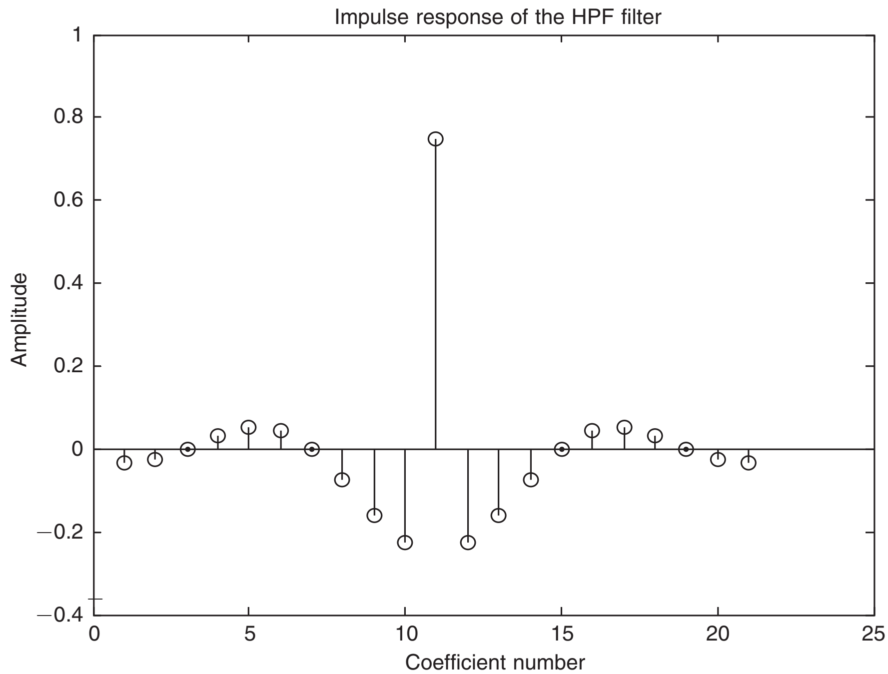

Figure 9 Impulse response of HPF using FS method.

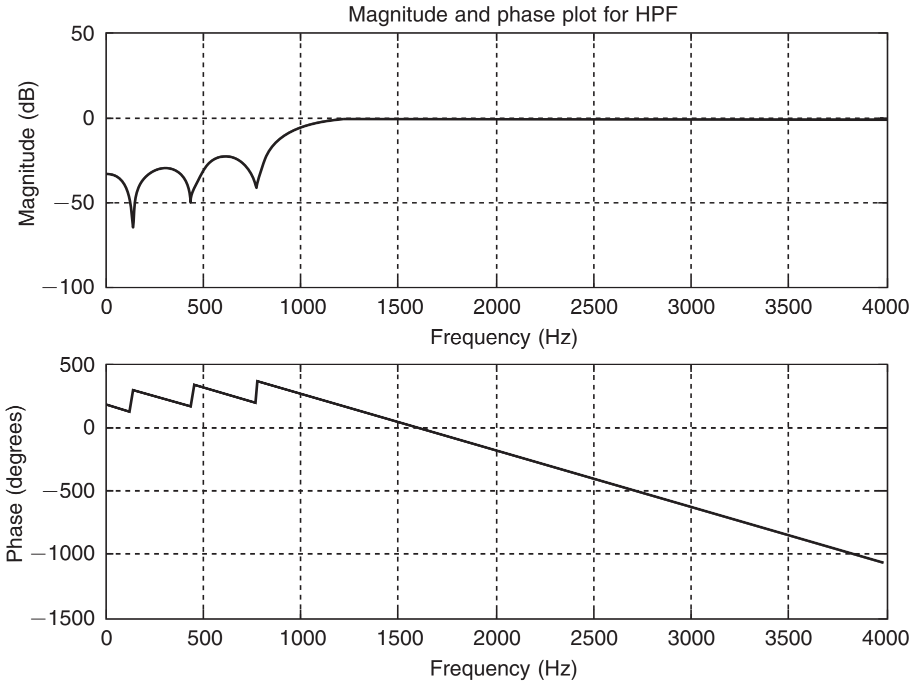

Figure 10 Magnitude and phase plot for HPF.

Program 5

Write a program to design BPF using FS expansion method.

```
%BPF design using FS expansion method
clear all;
fs=8000;
t=0.0025;
Q=fs/2*t;
disp(Q);
for i=1:10,
    x(i)=(sin(0.5*pi*i))/(i*pi)-(sin(0.25*pi*i))/(i*pi);
end
a(11)=0.25;
for i=1:10,
    a(i)=x(Q-(i-1));
end
for i=2:11
a(i+Q)=x(i-1);
end
```

```
disp(a);
stem(a);xlabel('coefficient number'); ylabel('amplitude');
title('impluse response of the BPF filter');
for i=1:4000
sum(i)=0.0;
end
figure;
[h,w]=freqz(a,1,256,8000);
plot(w/pi,20*log10(abs(h)));xlabel('normalized frequency');
ylabel('ampliudein dB');title('magnitude response plot');
figure;
plot(w/pi,(mod(angle(h), pi)*180/pi));xlabel('normalized
frequency');ylabel('angle in radians');title('Phase
response plot');
figure;
freqz(a,1,256,8000);title('magnitude and phase plot for
BPF');
```

Output

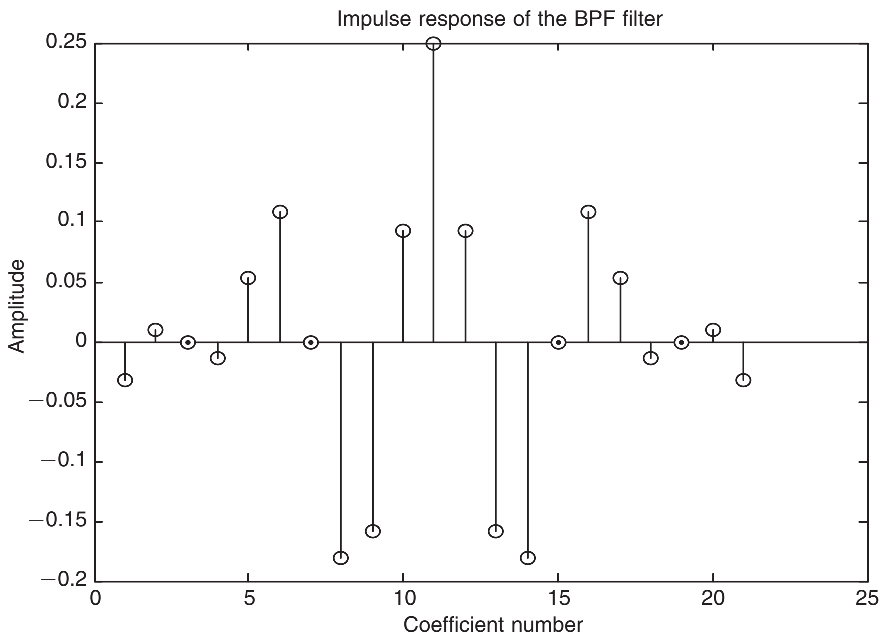

Figure 11 Impulse response of BPF using FS method.

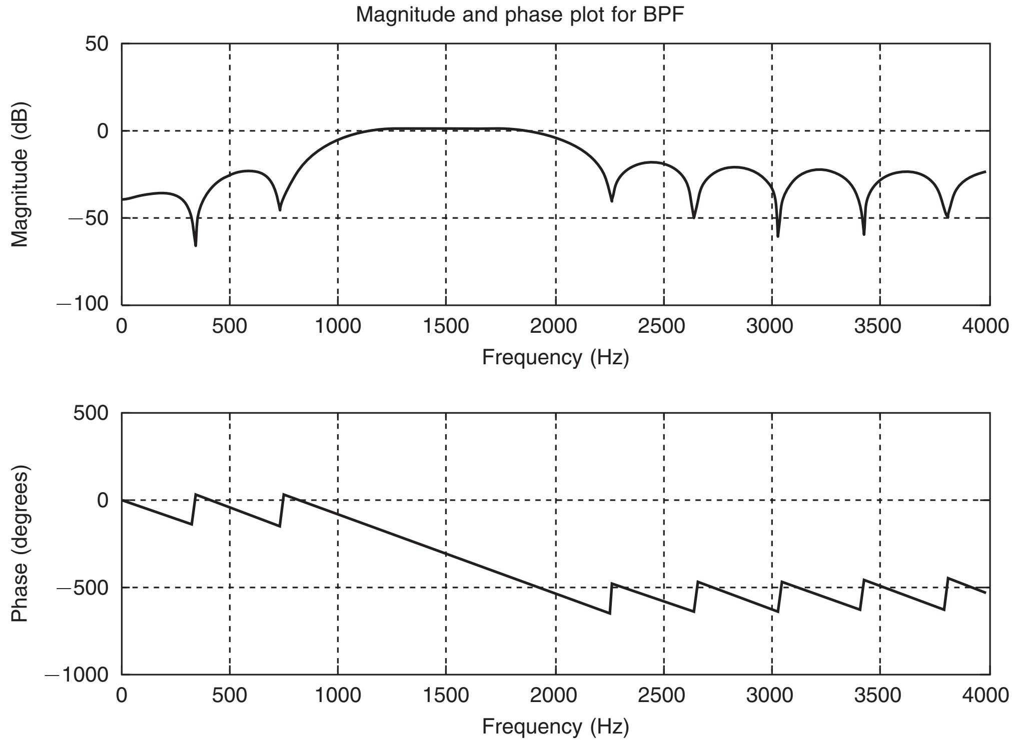

Figure 12 Magnitude and phase plot of BPF using FS method.

Program 6

Write a program to design an FIR filter using frequency sampling method.

```
%FIR filter design using frequency sampling method
clear all;
a=[1];
b=[0.06942 -0.05403 -0.10945 0.04733 0.31938 0.45455 0.31938
0.04733 -0.10945 -0.05403 0.06942];
freqz(b,a,256,1000);title('magnitude and phase response of
FIR LPF using frequency sampling method');
```

Output

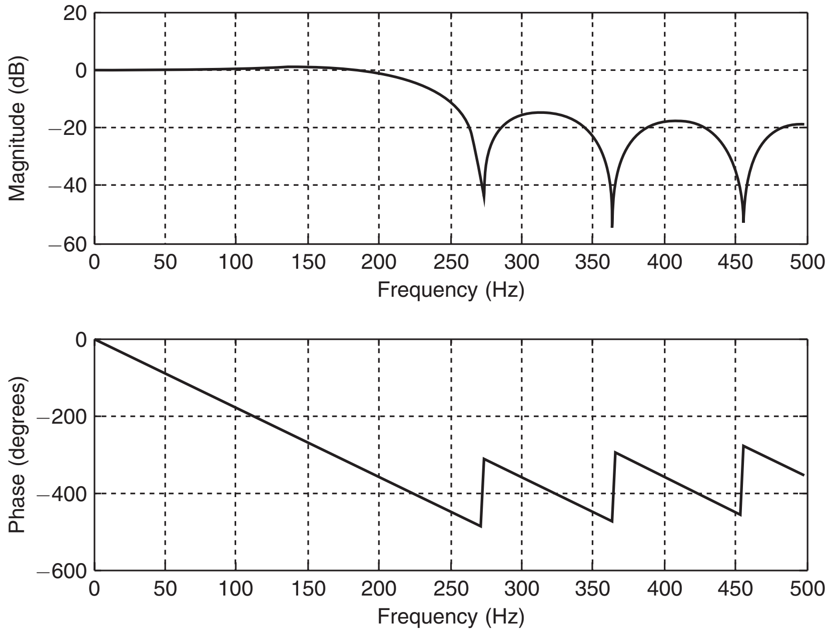

Figure 13 Magnitude and phase response for FIR filter using frequency sampling method.

Program 7

Write a program to design an FIR LPF design using frequency sampling method.

```
%FIR LPF design using frequency sampling method
clear all;
c=[0 exp(-j*6*pi/7) exp(-j*6*pi/7) 0 0 exp(j*6*pi/7)
exp(j*6*pi/7)];
d=ifft(c);
plot(d);title('Impulse response coefficients');xlabel
('coefficient number');ylabel('coefficient value');
disp(d);
figure;
a=[1];
freqz(d,a,256,1000);title('magnitude and phase response of
FIR LPF using frequency sampling method')
```

Output

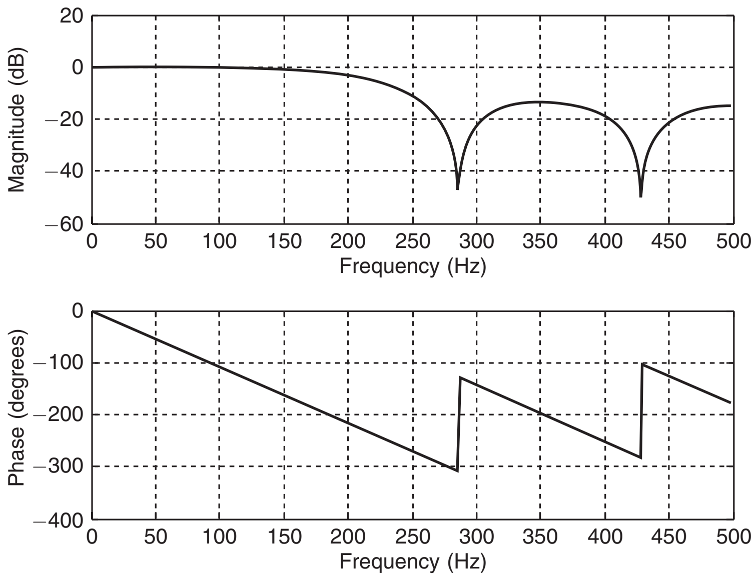

Figure 14 Magnitude and phase response for FIR LPF using frequency sampling method.

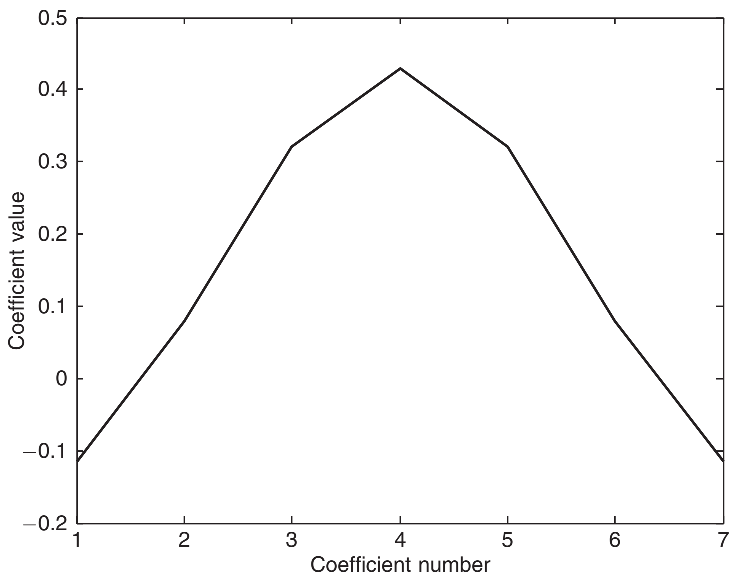

Figure 15 Impulse response coefficients.

Program 8

Write a program to design an FIR BPF using frequency sampling method.

```
%FIR BPF design using frequency sampling method
clear all;
c=[0 exp(-j*6*pi/7) exp(-j*12*pi/7) 0 0 exp(j*12*pi/7)
exp(j*6*pi/7)];
d=ifft(c);
plot(d);title('Impulse response coefficients');xlabel
('coefficient number');ylabel('coefficient value');
disp(d);
figure;
a=[1];
freqz(d,a,256,8000);title('magnitude and phase response of
FIR BPF using frequency sampling method');
```

Output

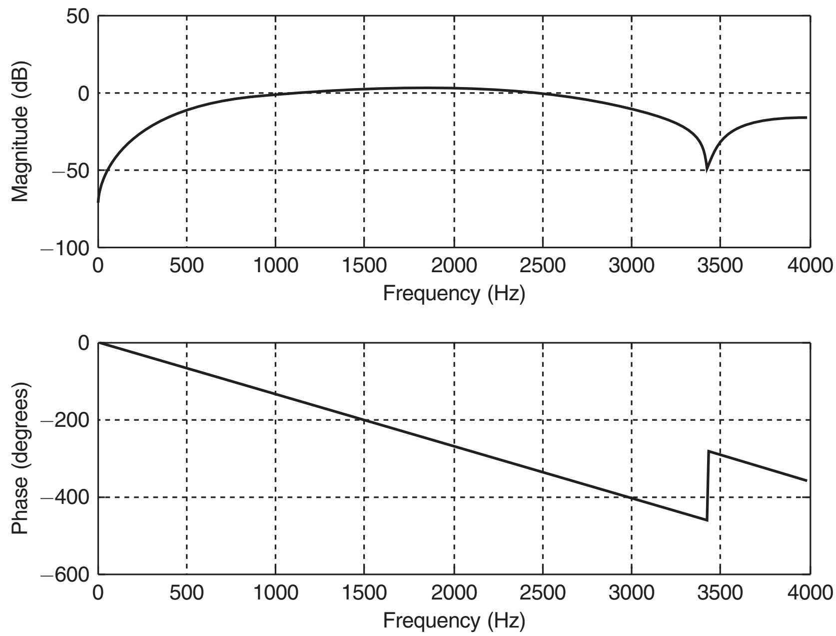

Figure 16 Magnitude and phase response for FIR BPF using frequency sampling method.

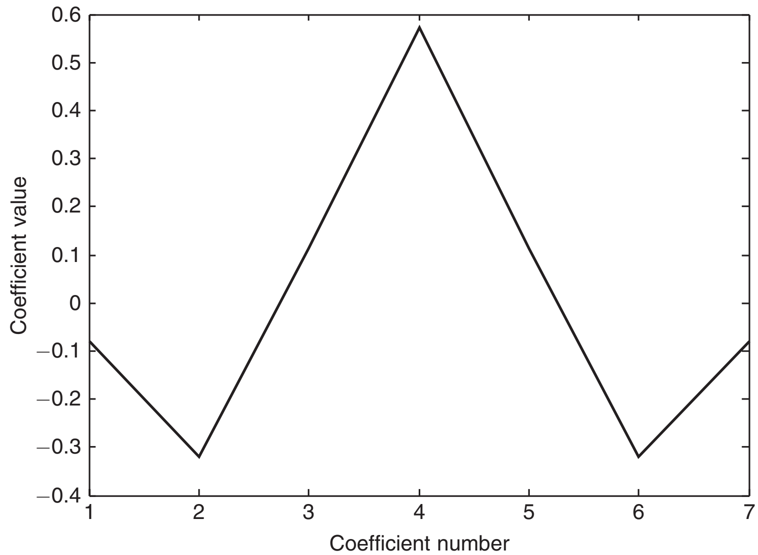

Figure 17 Impulse response coefficients.

Program 9

Write a program using FIR differentiator design.

```
%FIR differentiator design
clear all;
N=7;
for i=1:(N-1)/2,
    h(i)=cos(i*pi)/i;
end
a((N+1)/2)=0;
for i=1:3,
    a(i)=-h((N+1)/2-i);
end
for i=(N+3)/2:N;
    a(i)=h(i-(N+1)/2);
end
disp(a);
[h,w]=freqz(a,1,256,1000);
plot(w,abs(h));xlabel('normalized frequency');ylabel
('ampliude');title('magnitude response plot for a
differentiator with Rectangular window');

%FIR differentiator using Hamming window
clear all;
```

```
N=7;
for i=1:(N-1)/2,
    h(i)=cos(i*pi)/i;
end
a((N+1)/2)=0;
for i=1:3,
    a(i)=-h((N+1)/2-i);
end
for i=(N+3)/2:N;
    a(i)=h(i-(N+1)/2);
end
disp(a);
w1=window(@Hamming,7);
for i=1:N,
c(i)=a(i)*w1(i);
end
[h,w]=freqz(c,1,256,1000);
plot(w,abs(h));xlabel('normalized  frequency');ylabel
('ampliude');title('magnitude  response  plot  for  a
differentiator with Hamming window');
```

Output

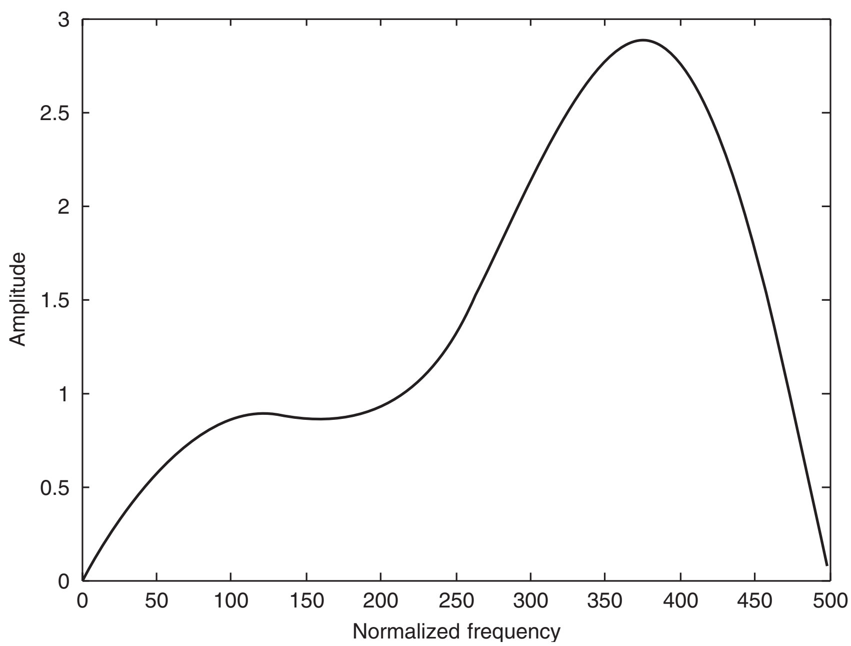

Figure 18 Magnitude response plot for differentiator using rectangular window.

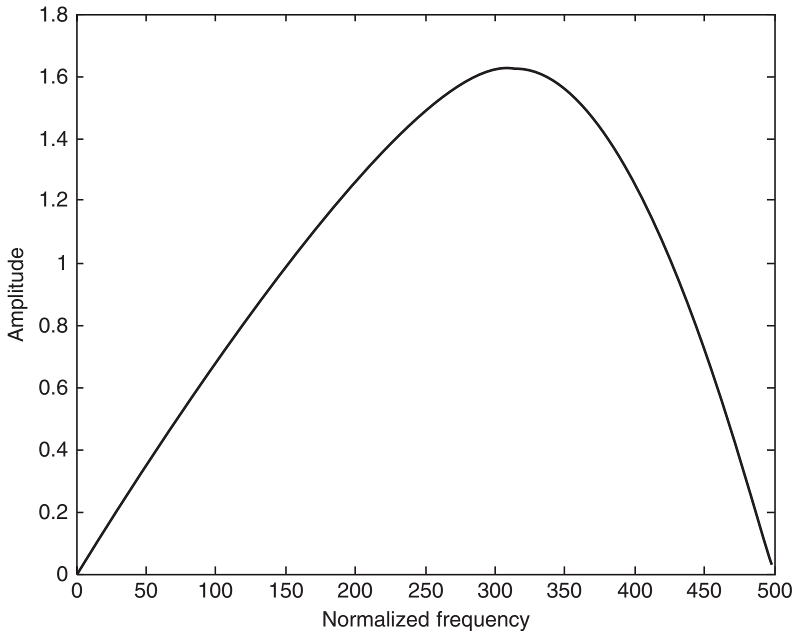

Figure 19 Magnitude response plot for differentiator with Hamming window.

Program 10

Write a program to design Hilbert transformer.

```
%Design of Hilbert transformer
clear all;
N=7;
for i=1:(N-1)/2,
    h(i)=(1-cos(i*pi))/i*pi;
end
a((N+1)/2)=0;
for i=1:3,
    a(i)=-h((N+1)/2-i);
end
for i=(N+3)/2:N;
    a(i)=h(i-(N+1)/2);
end
disp(a);
[h,w]=freqz(a,1,256,1000);
plot(w,abs(h));xlabel('normalized  frequency');ylabel
('ampliude');title('magnitude response plot for Hilbert
transformer with Rectangular window');
```

Program 11

Write a program to design Hilbert transformer using Hamming window.

```
%Hilbert transformer using Hamming window
clear all;
N=7;
for i=1:(N-1)/2,
    h(i)=(1-cos(i*pi))/i*pi;
end
a((N+1)/2)=0;
for i=1:3,
    a(i)=-h((N+1)/2-i);
end
for i=(N+3)/2:N;
    a(i)=h(i-(N+1)/2);
end
disp(a);
w1=window(@Hamming,7);
for i=1:7,
    c(i)=a(i)*w1(i);
end
[h,w]=freqz(c,1,256,1000);
plot(w,abs(h));xlabel('normalized  frequency');ylabel
('ampliude');title('magnitude response plot for Hilbert
transformer with Hamming window');
```

Output

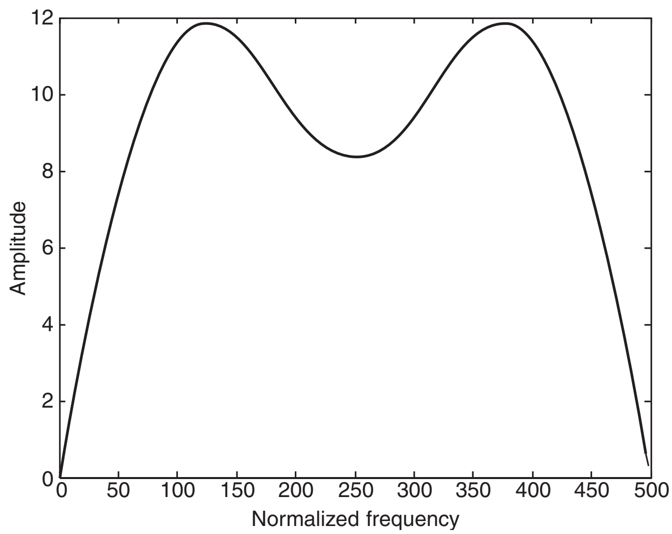

Figure 20 Magnitude response plot for Hilbert transformer using rectangular window.

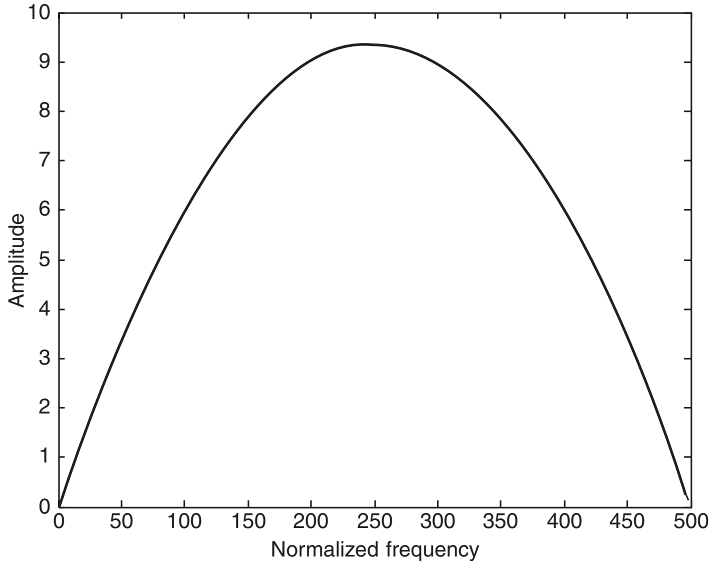

Figure 21 Magnitude response plot for Hilbert transformer using Hamming window.