LAB 2:
Dinesh Vaithyalingam Gangatharan
Harsha Nandeeshappa Krishnarajanagar

Preparation Exercises

1.

```
%% Signal definitions

n = -10:20;
h = [1 1 1 1 ];

x1 = sin((12/100*pi).*n);

u1=(n>=0);
u2=(n>=6);
x2 = u1 -u2;

x3 = ((9/10).^n).*u1;

delta1 = (n==1);
delta2 = (n==2);
delta3 = (n==3);
x4 = ((5/10).*delta1)+ delta2+((5/10).*delta3);

x5 = ((9/10).^n).*(cos(((2/10)*pi).*n));

%x6 = sin(((2/10)*pi).*n)./(((2/10)*pi).*n);
x6 = sinc(((2/10)*pi).*n);
```

2. Convolution Example

$x(n) = \delta(n) + \delta(n-1) + \delta(n-2) \rightarrow [1\ 1\ 1]$

$h(n) = \delta(n) - \delta(n-1) \rightarrow [0\ -1]$

By hand –

$y(n) = \delta(n) - \delta(n-4)$

Using convolution matrix –

x = [1 1 1]

$$h = \begin{matrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{matrix}$$

y = [1 0 0 -1]

3 and 4

3) $H(n)$ is stable if all the poles of $H(z)$ are inside the unit circle.

$$\left| \frac{-a_1 \pm \sqrt{a_1^2 - 4a_2}}{2} \right| < 1$$

$h(0) = b_0$

$h(1) = b_1 - a_1 b_0$

$h(2) = b_2 - b_0 a_2 - a_1 b_1 + a_1^2 b_0$

$h(n)$ will be finite if both a_1 & $a_2 = 0$

4) $y(n) - 0.25(y(n-2)) = x(n) - 0.25\,x(n-1)$

$Y(z) - 0.25\,Y(z)z^{-2} = X(z) - 0.25\,z^{-1}X(z)$

$$H(z) = \frac{1 - 0.25z^{-1}}{1 - 0.25z^{-2}} = \frac{z^2 - 0.25z}{z^2 - 0.25}$$

Poles $= z^2 - 0.25$ @ $z = +0.5$ & -0.5

zeroes $= z^2 - 0.25z$ @ $z = 0, 0.25$

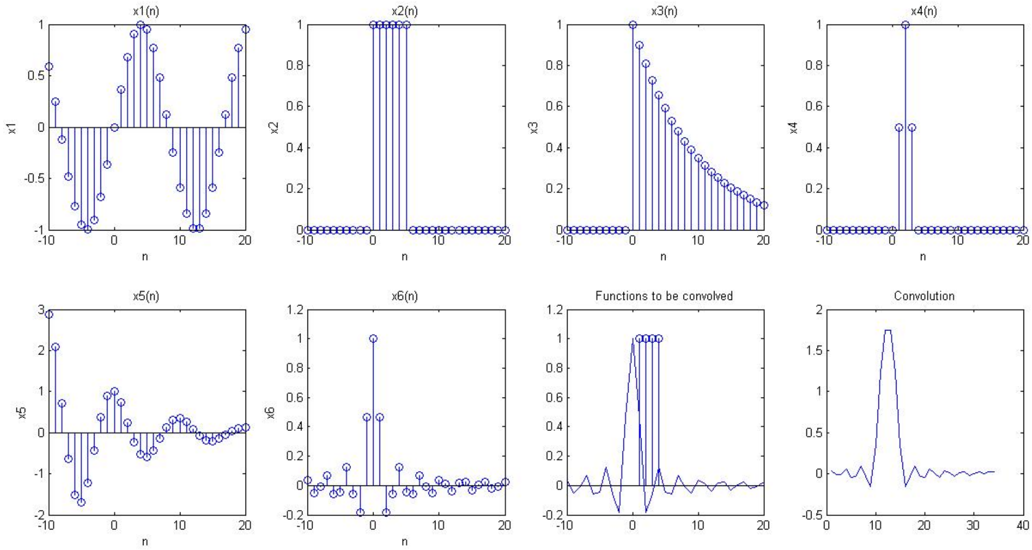
$$H(z) = 0.25\frac{z}{z-0.5} + 0.75\frac{z}{z+0.5}$$

$h(n) = (0.25 \times 0.5)^n + (0.75 \times 0.5)^n u(n)$

	0	1	2	3	4
$H = [$	1	-0.25	0.25	-0.0625	0.0625 $]$

Experiments

6.1 Discrete Time Signals




```
%% Plots

figure

subplot(2,4,1);
stem(n,x1);
xlabel('n');
ylabel('x1');
title('x1(n)');

subplot(2,4,2);
stem(n,x2);
title('x2(n)');
xlabel('n');
ylabel('x2');

subplot(2,4,3);
stem(n,x3);
title('x3(n)');
xlabel('n');
ylabel('x3');

subplot(2,4,4);
stem(n,x4);
title('x4(n)');
xlabel('n');
ylabel('x4');

subplot(2,4,5);
stem(n,x5);
title('x5(n)');
xlabel('n');
ylabel('x5');

subplot(2,4,6);
stem(n,x6);
title('x6(n)');
xlabel('n');
ylabel('x6');
```

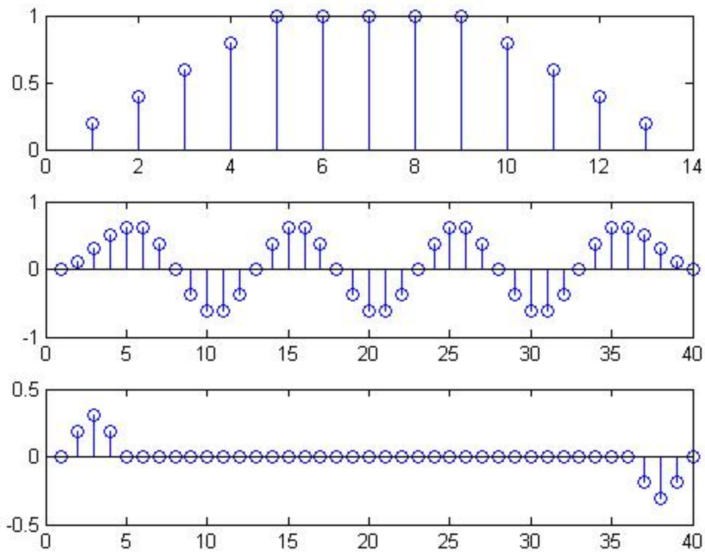
```
%% Convolution

y = convmat(x6,h);
subplot(2,4,7);
plot(n,x6);
hold on;
stem(h);
title('Functions to be convolved');
subplot(2,4,8);
plot(y);
title('Convolution');
```

Convolution

```
function [ y ] = convmat( x,h )
%CONVMAT Summary of this function goes here
%   Detailed explanation goes here
%x,h,y are column vectors
lenx=length(x);
lenh=length(h);
N=lenx+lenh-1;
H=[];
for (count=1:lenx)
    tmp=[zeros(count-1,1);h;zeros(N-count-lenh+1,1)];
    H=[H tmp];

end
y=H*x;

end
```

Averaging Filter

```
clear all;
clc;

u = ones(1,5);
h = u.*(2/10);

x1 = ones(1,9);
n = 0:35;
x2 = sin(((2/10)*pi).*n);
x3 = sin(((4/10)*pi).*n);
y1 = convmat(x1,h);
y2 = convmat(x2,h);
y3 = convmat(x3,h);

figure (1);
subplot(3,1,1);
stem(y1);
subplot(3,1,2);
stem(y2);
subplot(3,1,3);
stem(y3);
```



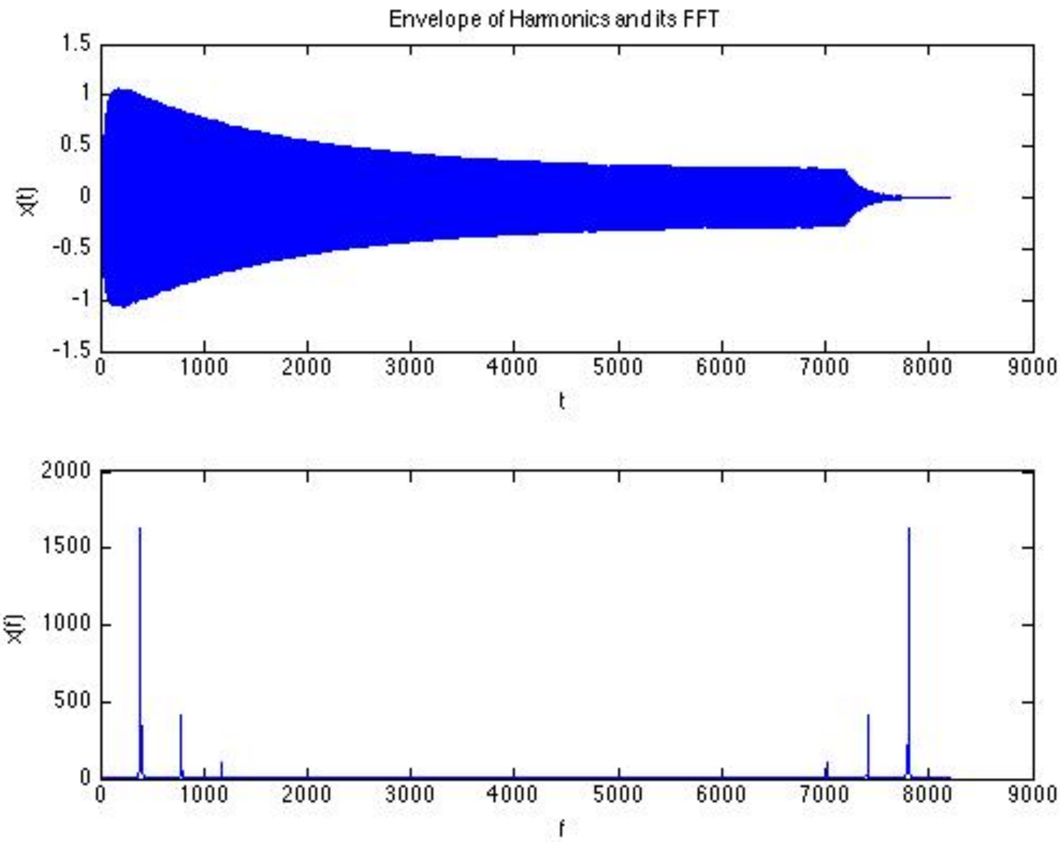
6.2 Musical Tone Synthesis

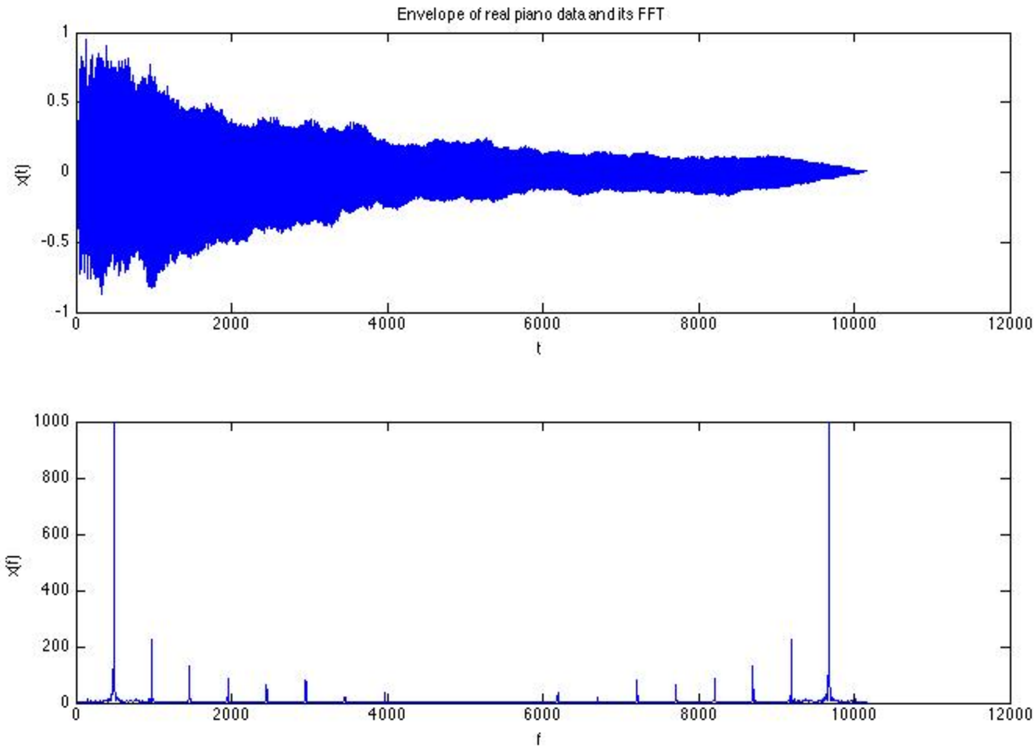
Code –

```
clear all
%Musical Tone Synthesis 6.2 - 1
f1 = 392; %in Hz
fs = 8192; %in Hz
t1 = 1/f1;
T = 1/fs;
starttime = 0;
endtime = 1 - T;
t = starttime:T:endtime;
s1s = sin(2 * pi * f1 * t);
% stsf = fft(s1s);
% figure(1)
% subplot(2,1,1) , stem(s1s);
% % stem(s1s);
% subplot(2,1,2) , plot(1:numel(stsf),stsf);
soundsc(s1s, fs);
disp('Press any key to continue..');
pause
% Harmonics - 2

snew = s1s;
for k = 2:8
    fk = f1 * k;
    pk(k) = 0.25.^(k-1);
    sh = pk(k) * (sin(2 * pi * fk * t)); %harmonic
    snew = snew + sh; %adding harmonic signal
end
stsf = fft(snew);
% figure(2)
% subplot(2,1,1) , stem(snew);
% subplot(2,1,2) , plot(1:numel(stsf),stsf);
soundsc(snew, fs);

disp('Press any key to continue..');
pause


%envelope - 3
A = envelope(numel(snew), 0.25, [240 7200]);
```

```
esnew = A .* snew;
ESNEW = fft(esnew);
figure(3)
subplot(2,1,1),
plot(esnew);
xlabel('t'); ylabel('x(t)');
title('Envelope of Harmonics and its FFT')
subplot(2,1,2), plot(1:numel(ESNEW),abs(ESNEW));
xlabel('f'); ylabel('x(f)');


% piano - 4
piano = load('pianoG3.mat');
fftpiano = fft(piano.g);
soundsc(piano.g, piano.fs);
figure(4)
subplot(2,1,1),
plot(piano.g);
xlabel('t'); ylabel('x(t)');
title('Envelope of real piano data and its FFT');
subplot(2,1,2),
plot(1:numel(fftpiano),abs(fftpiano));
xlabel('f'); ylabel('x(f)');
```

Results -





Yes, difference is noticable.

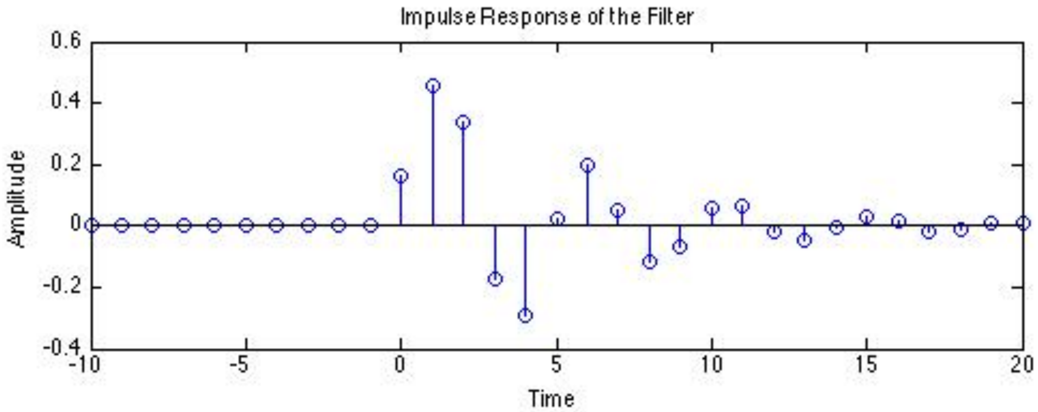
The realness of of the synthesized sound can be improved by adding more harmonics to match the real one.

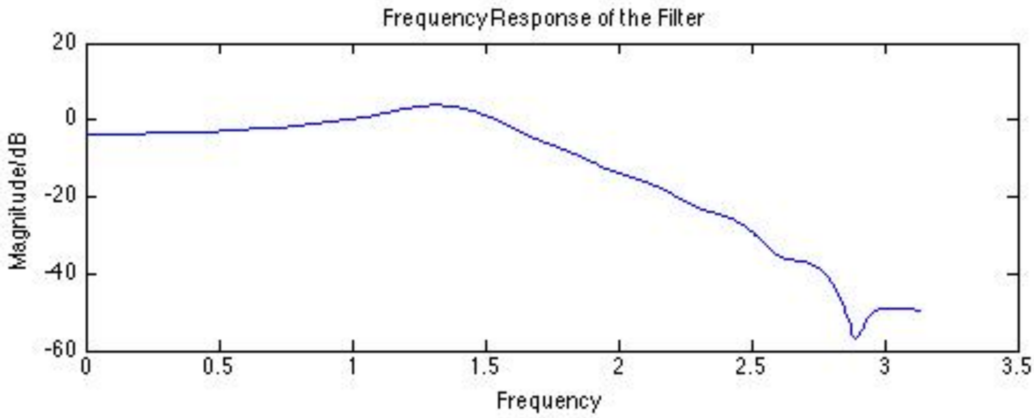
6.3 - Discrete Time Systems

```
%% 1-
b = [0.16 0.48 0.48 0.16];
a = [1 0.13 0.52 0.3];
n = -10:20;
delta = (n==0);
h = filter(b,a,delta);
[H,w] = freqz(h);
figure(1);
subplot(2,1,1);
stem(n,h);
title('Impulse Response of the Filter');
xlabel('Time'); ylabel('Amplitude');
subplot(2,1,2);
plot(w, 20*log10(abs(H)));
title('Frequency Response of the Filter');
xlabel('Frequency'); ylabel('Magnitude/dB');

%% 2-
n = 0:255;
K = 100;
k = 0:K;
wk = pi * k / K;
x = cos(n' * wk);
h = filter(b,a,x);
h = h(31:end,:);
H_new = max(abs(h));
```

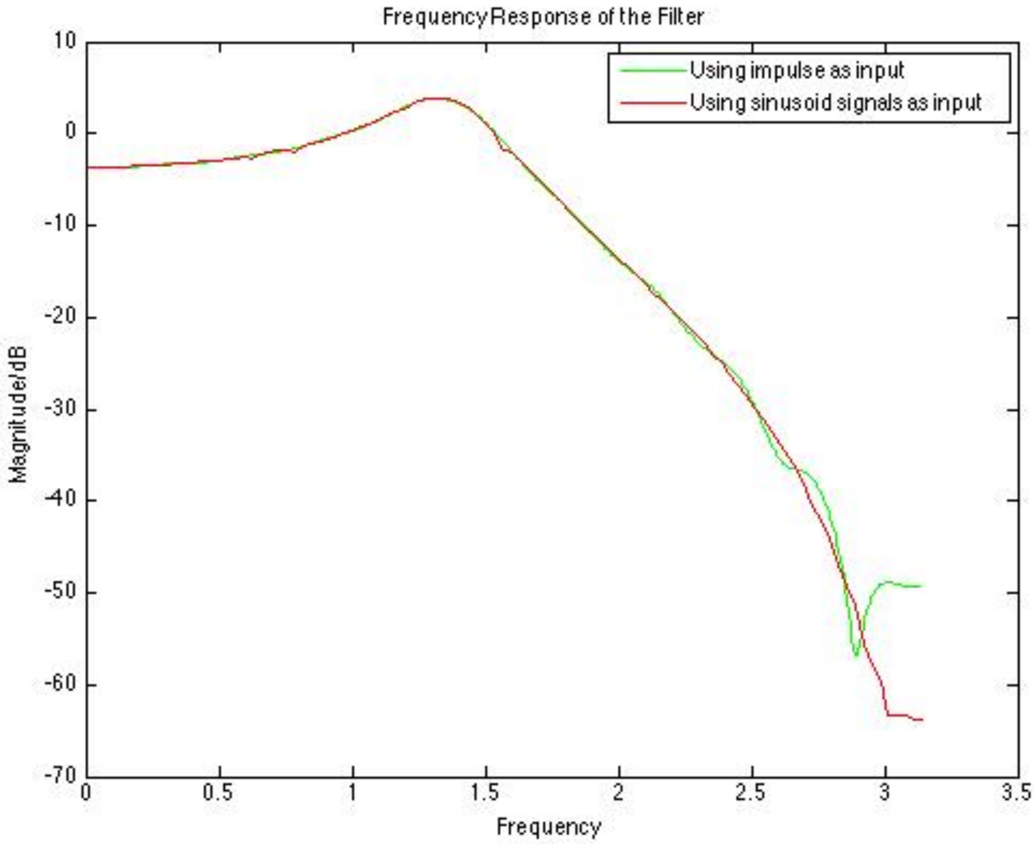
```
figure(2)
plot(w, 20*log10(abs(H)),'g',wk, 20*log10(H_new),'r');
title('Frequency Response of the Filter');
xlabel('Frequency'); ylabel('Magnitude/dB');
legend('Using impulse as input','Using sinusoid signals as input');
```

Results –





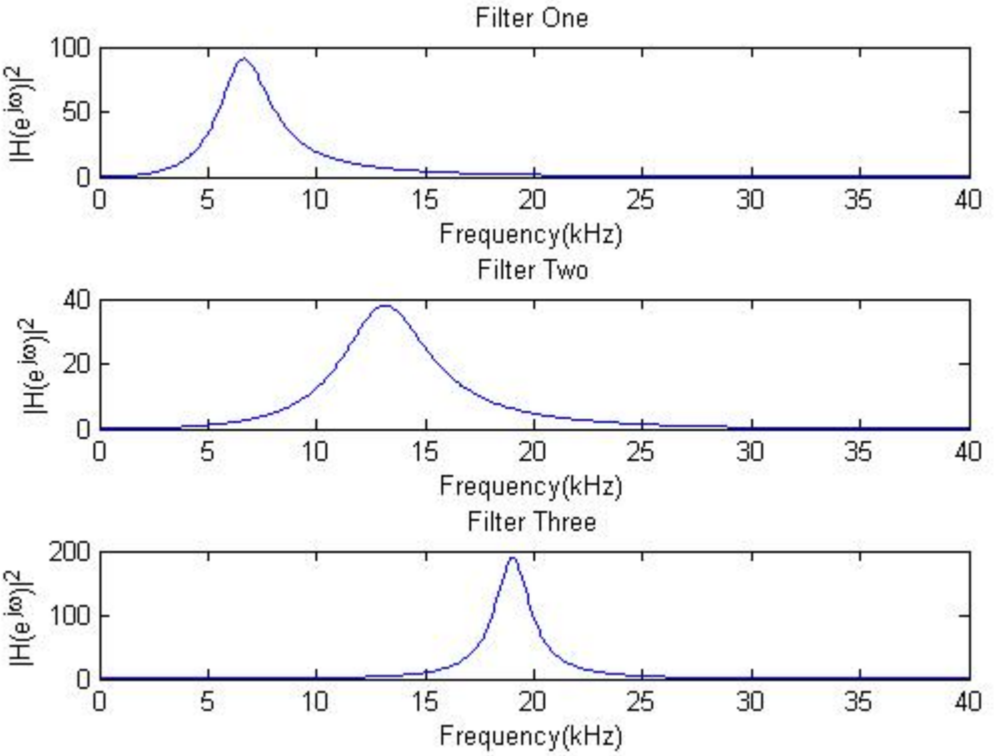
The fourier transform of impulse is 1, meaning it contains all the frequencies. Therefore an impulse response of the system gives info about the response to all the frequencies.
On the other hand, the test signal contains a subset of frequencies, since in practice this is the case(i.e not all the frequencies are used) this is a reasonable procedure.

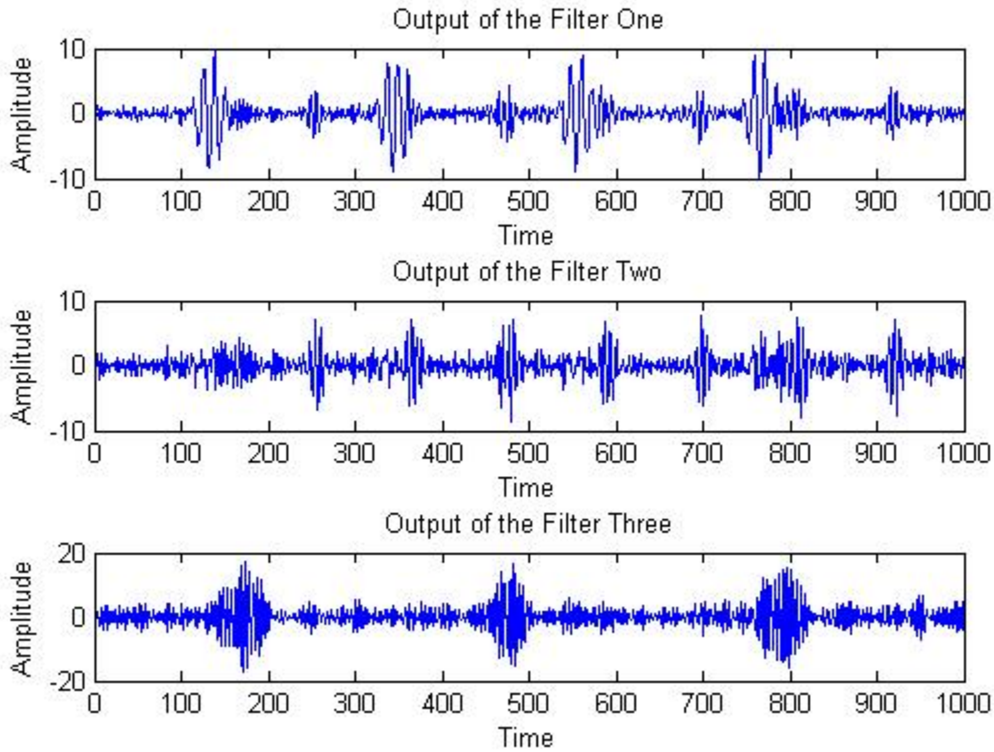


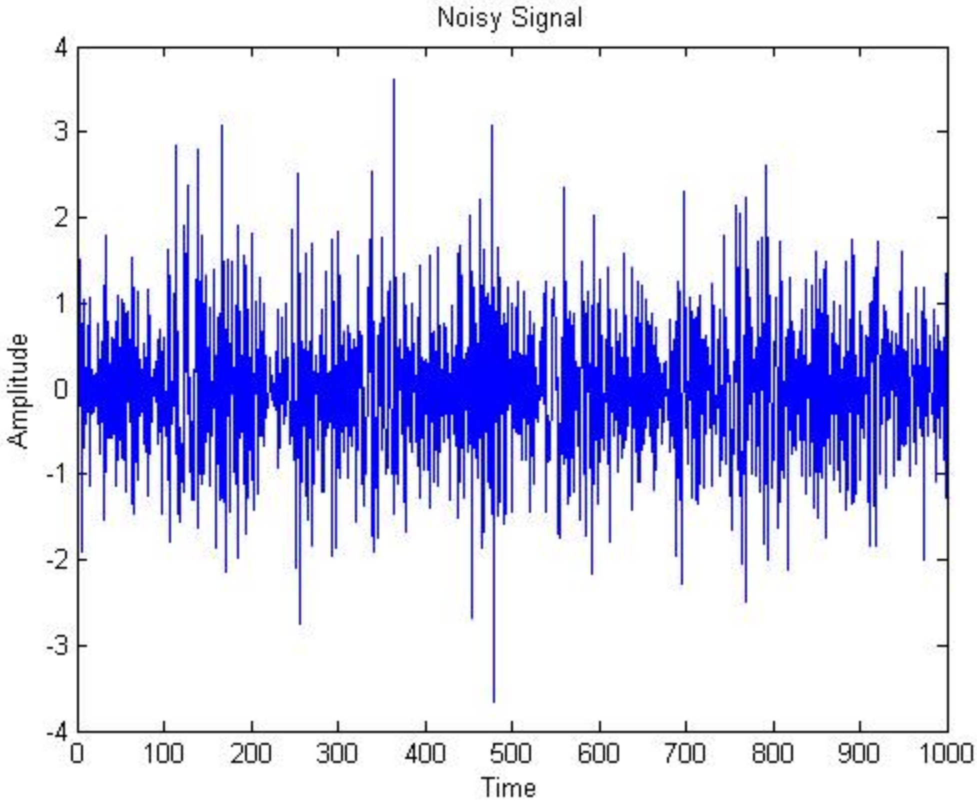
6.4 Bandpass Filtering

```
clear all;
clc;
%Sample with hidden pulses loaded
 load('..\Lab_2\files_lab2\b3pulses.mat');
%Sampling frenquency (in Hz)
fs = 80000;
%Stopband and Passband Frequency ranges (in Hz)
fp1 = 5000;     fs1 = 8000;
fp2 = 10500;    fs2 = 15500;
fp3 = 18000;    fs3 = 20000;
f01 = (fp1+fs1)/2; f02 = (fp2+fs2)/2;  f03 = (fp3+fs3)/2;
%Digital frequency bands
w01 = f01/fs*2*pi;
w02 = f02/fs*2*pi;
w03 = f03/fs*2*pi;
%Range (Digital domain)
delW = [(fs1-fp1)/fs*2*pi; (fs2-fp2)/fs*2*pi; (fs3-fp3)/fs*2*pi;];

r=[];
for i=1:3
    tmp=roots([4 -(8+delW(i)^2) 4]);
    r=[r tmp(2)];
end

b=[1 0 -1];
a1=[1 -2*r(1)*cos(w01) r(1)^2];
a2=[1 -2*r(2)*cos(w02) r(2)^2];
```

```
a3=[1 -2*r(3)*cos(w03) r(3)^2];

figure(1);
plot(x);
title('Noisy Signal');
xlabel('Time');
ylabel('Amplitude');

figure(2);
subplot(3,1,1);
[H1,w1]=freqz(b,a1);
plot(w1/2/pi*fs/1000,abs(H1).^2);
title('Filter One');
xlabel('Frequency(kHz)');ylabel('|H(e^{ j\omega})|^2');

subplot(3,1,2);
[H2,w2]=freqz(b,a2);
plot(w2/2/pi*fs/1000,abs(H2).^2);
title('Filter Two');
xlabel('Frequency(kHz)');ylabel('|H(e^{ j\omega})|^2');


subplot(3,1,3);
[H3,w3]=freqz(b,a3);
plot(w3/2/pi*fs/1000,abs(H3).^2);
title('Filter Three');
xlabel('Frequency(kHz)');ylabel('|H(e^{ j\omega})|^2');

y1=filter(b,a1,x);
y2=filter(b,a2,x);
y3=filter(b,a3,x);

figure(3);
subplot(3,1,1);
plot(y1);
title('Output of the Filter One');
xlabel('Time');ylabel('Amplitude');
subplot(3,1,2);
plot(y2);
title('Output of the Filter Two');
xlabel('Time');ylabel('Amplitude');
subplot(3,1,3);
plot(y3);
title('Output of the Filter Three');
xlabel('Time');ylabel('Amplitude');
```


Filter One
|H(e jω)|2
Frequency(kHz)
Filter Two
|H(e jω)|2
Frequency(kHz)
Filter Three
|H(e jω)|2
Frequency(kHz)


Output of the Filter One
Amplitude
Time
Output of the Filter Two
Amplitude
Time
Output of the Filter Three
Amplitude
Time


Noisy Signal
Amplitude
Time
4
3
2
1
0
-1
-2
-3
-4
0
100
200
300
400
500
600
700
800
900
1000