Digital Signal Processing 2
Homework Assignment 2

Name: Amitesh Kumar Sah
NYU ID: N19714360

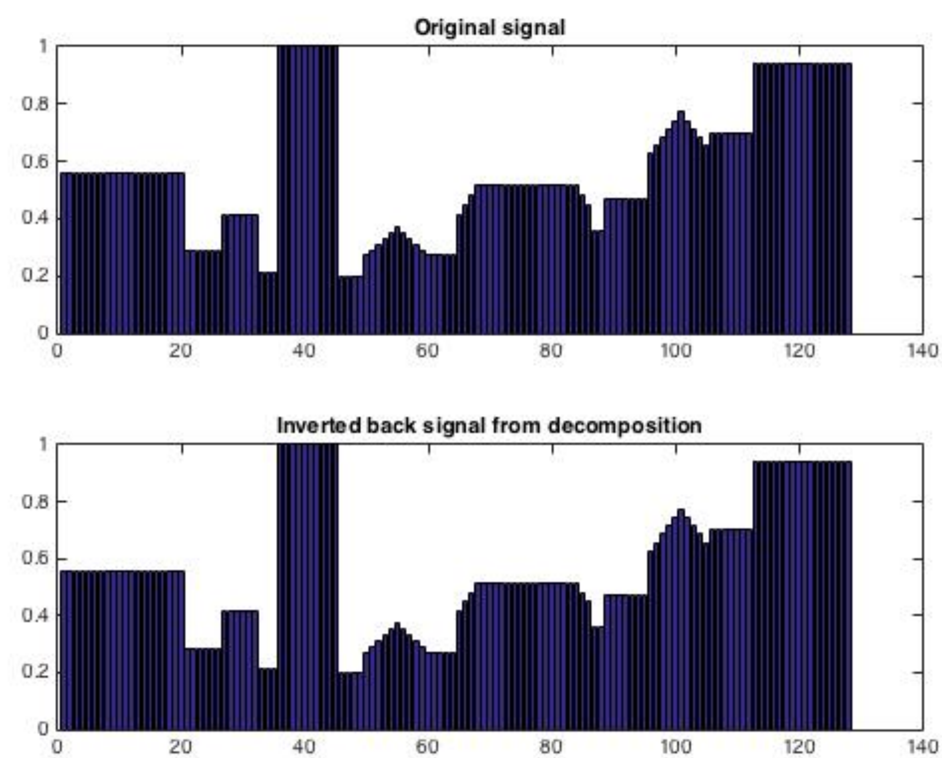
Question A

Parseval identity
-the sum of squares of signal values is equal to the sum of squares of wavelet coefficients.

For Haar Transform

```
% Reading a file
X = dlmread('skyline.txt');
l=length(X);    %Calculating the length of file
dup_X=X;
% Padding if odd length
if (mod(l,2)~=0)
       X(l+1,1)=0;
        l=l+1;    %New length is increased by 1
end

%
% To find the number of level-operation
k=l;
m=1;
while k~=2
   k=k/2;
   m=m+1;
end
%

% For M-Level Operation forward transform
p=l;
coef=1/2;
for j=1:m
% For a single block operation
for i=1:(p/2);
   c(i,j)= coef*X(2*i-1)+coef*X((2*i));
   d(i,j)= coef*X(2*i-1)-coef*X((2*i));
end
posd(j)=i;
clearvars X;
X=c(:,j);
p=p/2;
end
%

figure,
for j=1:m
```

```
   subplot(m+1,1,j+1),bar(1:posd(m-j+1),d(1:posd(m-j+1),m-j+1));
end
    subplot(m+1,1,1),bar(1,c(1:posd(m),m));

% For M-Level Operation inverse transform
c1=c(1:posd(m),m);
d1=d;
k=m;
while k>=1
for i=1:posd(k)
   y((2*i)-1,1)=c1(i)+d1(i,k);
   y((2*i),1)=c1(i)-d1(i,k);
end
c1=y;
k=k-1;
end

figure,
subplot(2,1,1),bar(dup_X), title('Original signal');
subplot(2,1,2),bar(y), title('Inverted back signal from decomposition');

% Proving Parseval Identity
x_sum=sum(dup_X.^2)
c_sum=sum(sum(c.^2));
d_sum=sum(sum(d.^2));
wavelet_sum=c_sum+d_sum
y_sum=sum(y.^2)
```



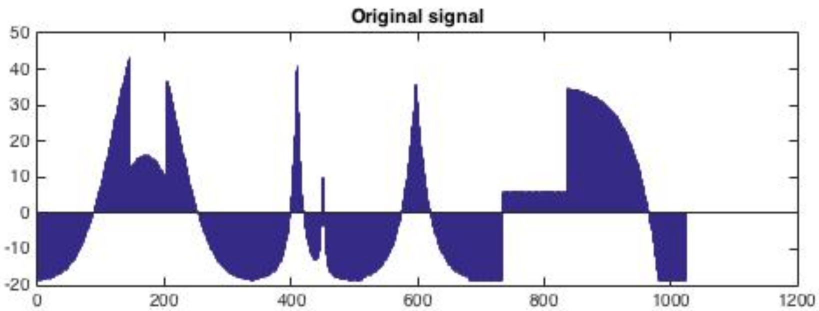

```
x_sum =
```

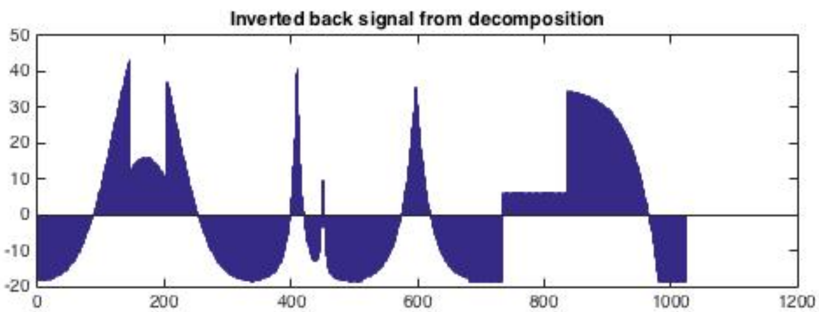
49.3502
wavelet_sum =
  48.3063
y_sum =
  49.3502

For Daubechies Wavelet transform

```
% Reading a file
X = dlmread('pwsmooth.txt');
% X = dlmread('skyline.txt');
l=length(X);    %Calculating the length of file
dup_X=X;
% Padding if odd length
if (mod(l,2)~=0)
      X(l+1,1)=0;
       l=l+1;    %New length is increased by 1
end

%  the number of level-operation
m=4;

% Definng the multipliers h0, h1, h2, h3
h0=(1+sqrt(3))/(4*sqrt(2));
h1=(3+sqrt(3))/(4*sqrt(2));
h2=(3-sqrt(3))/(4*sqrt(2));
h3=(1-sqrt(3))/(4*sqrt(2));

% For 4-Level Operation forward transform
p=l/2;
for j=1:m
% For a single block operation
for i=1:p;
   if i==p
      c(i,j)= h0*X(2*i-1)+h1*X(2*i)+h2*X(2*i)+h3*X(2*i);
      d(i,j)= h3*X(2*i-1)-h2*X(2*i)+h1*X(2*i)-h0*X(2*i);
   else
      c(i,j)= h0*X(2*i-1)+h1*X(2*i)+h2*X((2*i)+1)+h3*X((2*i+2));
      d(i,j)= h3*X(2*i-1)-h2*X(2*i)+h1*X((2*i)+1)-h0*X((2*i+2));
   end
end
posd(j)=i;
clearvars X;
X=c(:,j);
p=p/2;
end

figure,
for j=1:m
   subplot(m+1,1,j+1),bar(1:posd(m-j+1),d(1:posd(m-j+1),m-j+1));
end
```

```
    subplot(m+1,1,1),bar(1:posd(m),c(1:posd(m),m));

% For M-Level Operation inverse transform
c1=c(1:posd(m),m);
d1=d;
k=m;

while k>=1
for i=1:posd(k)
    j=1;
    if i==1
    y((2*i)-1,1)=h0*c1(i)+h2*c1(i)+h3*d1(i,k)+h1*d1(i,k);
    y((2*i),1)=h1*c1(i)+h3*c1(i)-h2*d1(i,k)-h0*d1(i,k);
    else
    y((2*i)-1,1)=h0*c1(i)+h2*c1(i-1)+h3*d1(i,k)+h1*d1(i-1,k);
    y((2*i),1)=h1*c1(i)+h3*c1(i-1)-h2*d1(i,k)-h0*d1(i-1,k);
    end
end
c1=y;
k=k-1;
end

figure,
subplot(2,1,1),bar(dup_X), title('Original signal');
subplot(2,1,2),bar(y), title('Inverted back signal from decomposition');

% Proving Parseval Identity
x_sum=sum(dup_X.^2)
c_sum=sum(sum(c.^2));
d_sum=sum(sum(d.^2));
wavelet_sum=c_sum+d_sum
y_sum=sum(y.^2)
```



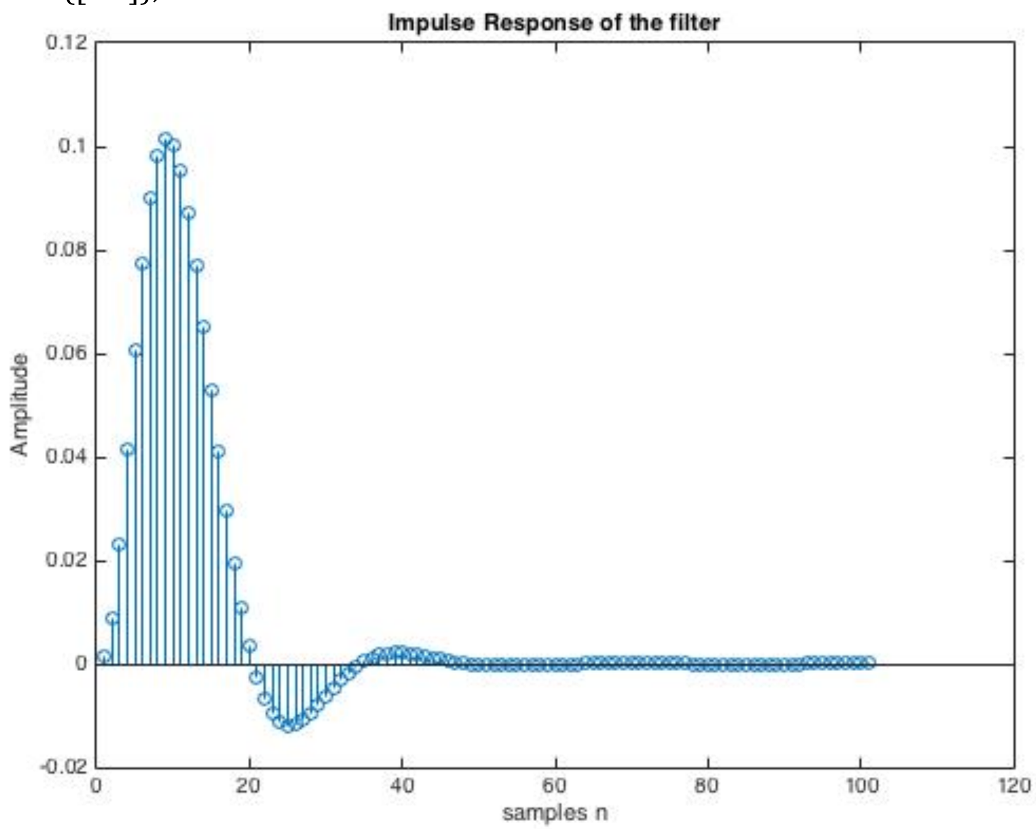


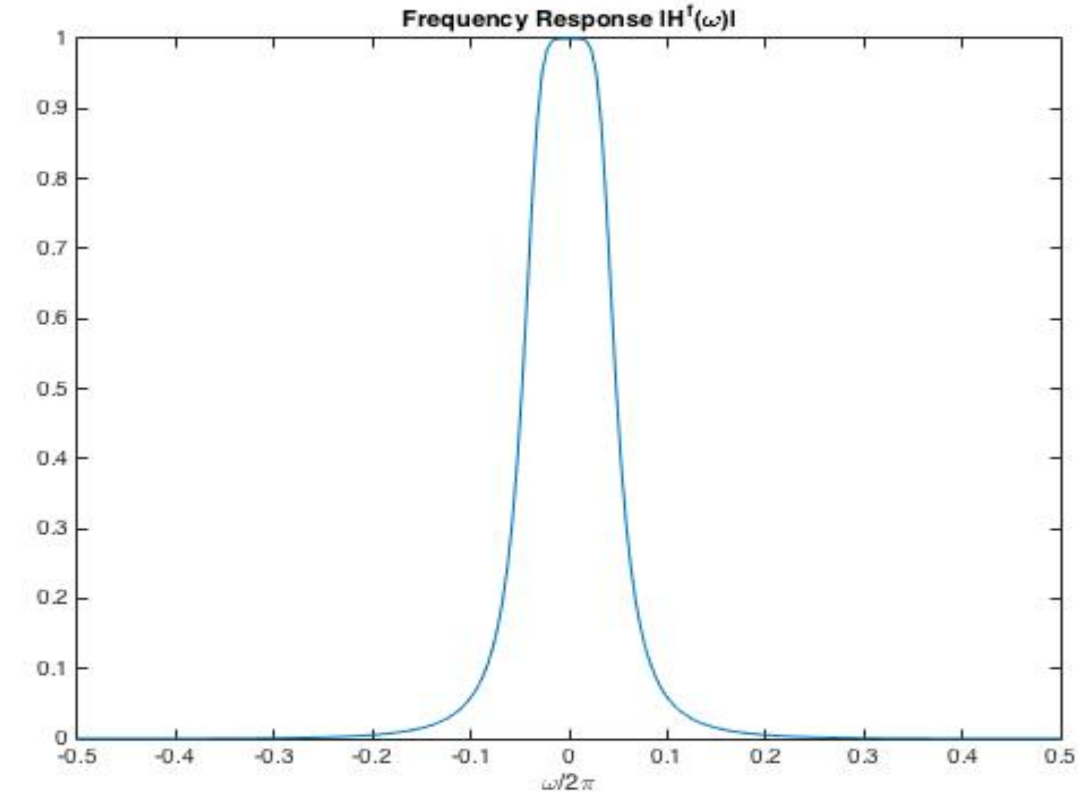
```
x_sum =
  3.2926e+05
wavelet_sum =
  1.3099e+06
y_sum =
  3.2924e+05
```

Question B

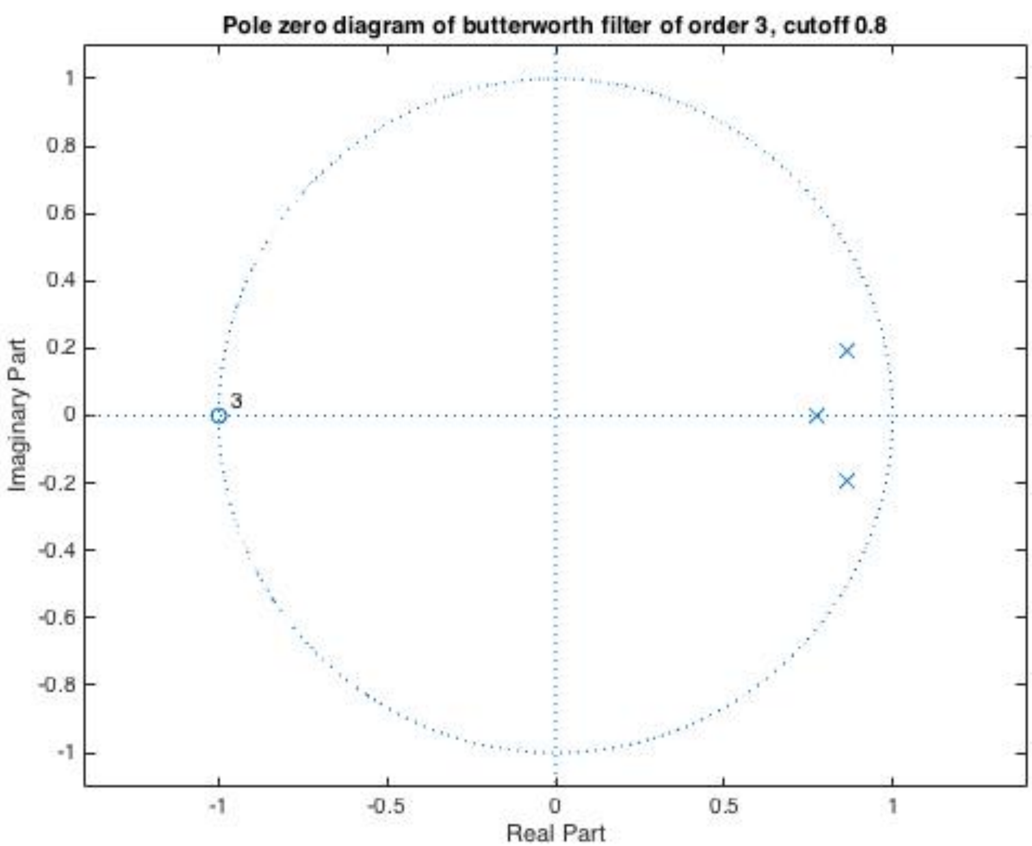
```
clc;clear all;close all;

% To get the filter coefficient of butterworth filter H(z)=B(z)/A(z)
[b,a]=butter(3,0.08); %Third order filter with 0.3pi radian/sec as cutoff freq

% Impulse Response
imp=[1 zeros(1,100)];
h=filter(b,a,imp);
figure,stem(h)
xlabel('samples n');
ylabel('Amplitude');
title('Impulse Response of the filter');

% Frequency Response
j=sqrt(-1);
om=linspace(-pi,pi,200);
Hf=polyval(b,exp(j*om))./polyval(a,exp(j*om));
figure,plot(om/(2*pi),abs(Hf))
title('Frequency Response |H^f(\omega)|');
xlabel('\omega/2\pi');

figure,
% Pole zero diagram
zplane(b,a)
title('Pole zero diagram of butterworth filter of order 3, cutoff 0.8')

% load ECG signal
Fs=256;
x=ecgsyn(Fs,10);
n=0:length(x)-1;
t=n/Fs;
figure,plot(t,x);
title('ECG signal');
xlabel('Time in sec');
xlim([4 7]);

% Noise signal
N=length(x);
sigma=0.2;              % sigma : noise standard deviation
noise=sigma*randn(N, 1);    % noise : white Gaussian noise
g1=x+noise;             % g1 : noisy ECG (white noise)
```

```
g2=g1+cos(0.2*pi*n');       % g2 : ECG with white noise and tonal noise
figure,plot(t,g2);
title('Noisy ECG signal');
xlabel('Time in sec');
xlim([4 7]);

% Fourier Transform of noise signal

f=fftshift(fft(g2,N));%Fourier Transform of noise signal
t1=(-N/2:N/2-1)/N;
f2=fftshift(fft(x,N))% Fourier transform of original signal
figure,plot(t1,abs(f),t1,abs(f2));
legend('noisy ecg','true ecg');
xlabel('Normalized Frequency');
ylabel('Magnitude');
title('Double Sided FFT - with FFTShift of noise signal');

% Apply low pass filter to noisy ECG signal
y=filter(b,a,g2);
figure,plot(t,y,t,x)
legend('filtered output','true ecg')
title('Filter output signal')
xlabel('Time (seconds)')
xlim([4 6]);
```

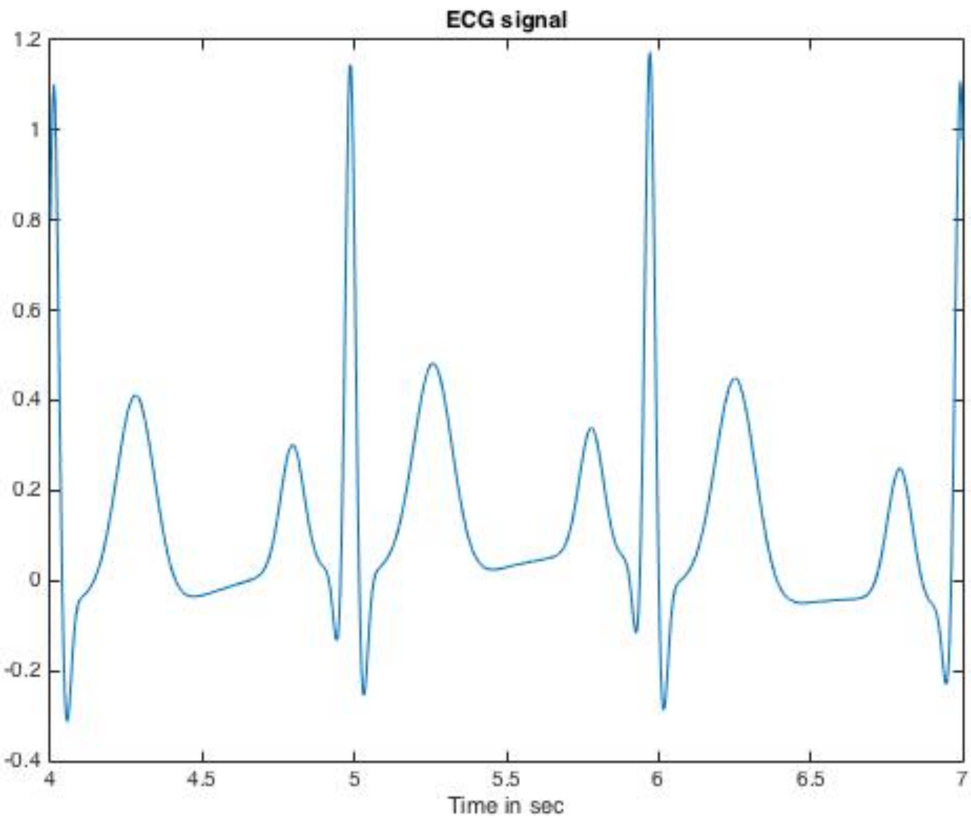


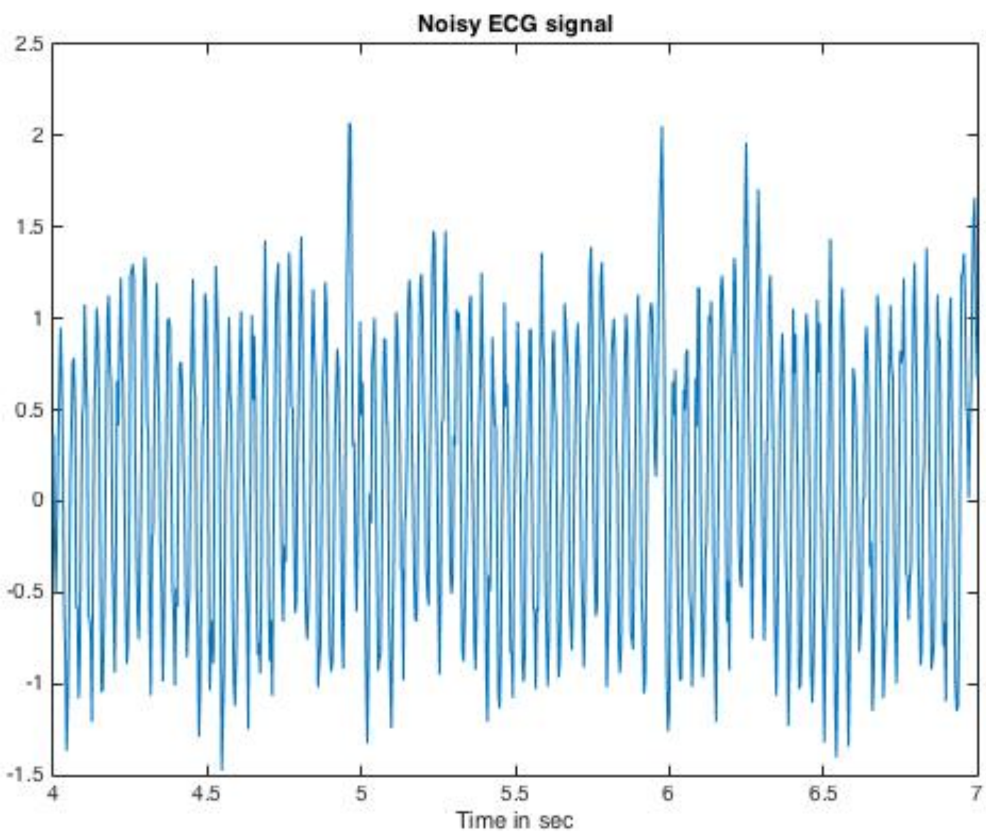


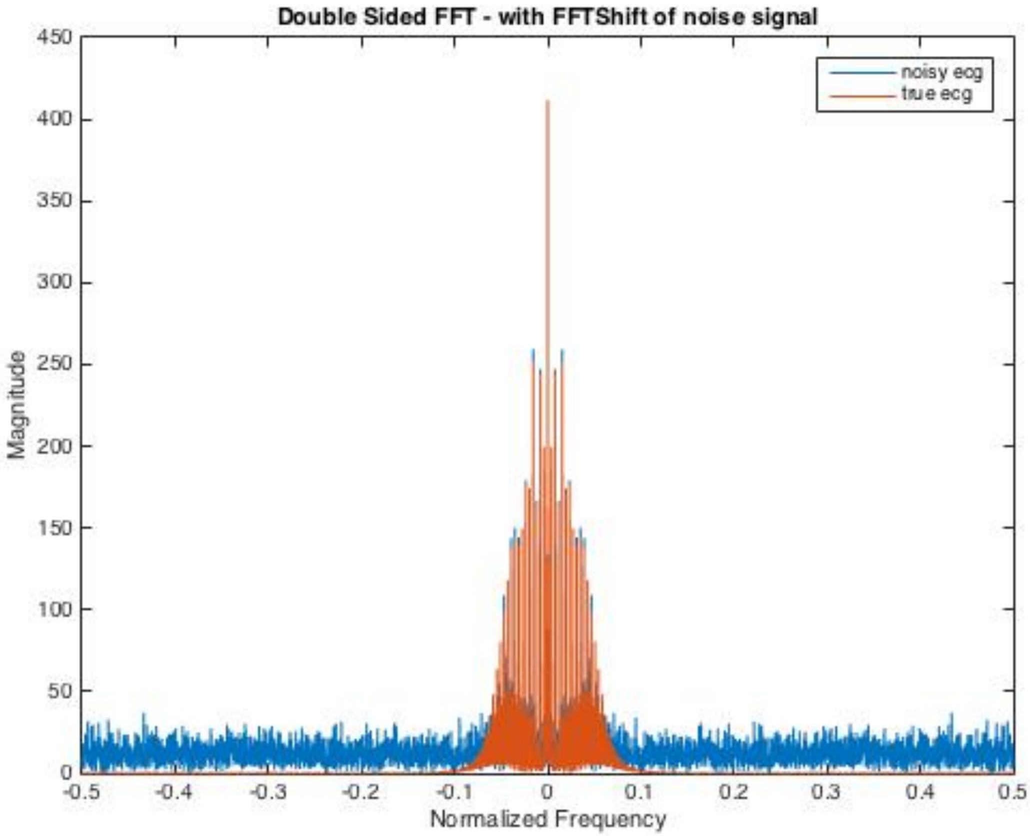
Frequency Response $|H^{\dagger}(\omega)|$
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5
$\omega/2\pi$

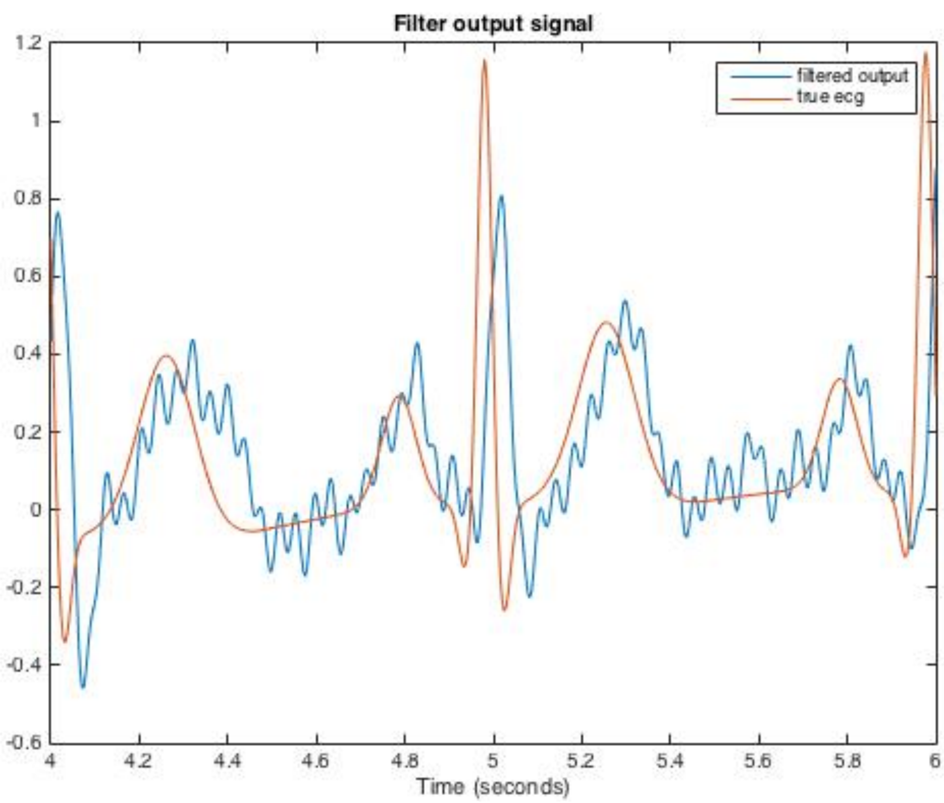


Pole zero diagram of butterworth filter of order 3, cutoff 0.8
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
3
Imaginary Part
-1
-0.5
0
0.5
1
Real Part


ECG signal
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
4
4.5
5
5.5
6
6.5
7
Time in sec


Noisy ECG signal
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
4
4.5
5
5.5
6
6.5
7
Time in sec





Observation:

After removing the noise signal using butterworth filter 3rd order and at 0.08 cutoff frequency, we still could not get the proper result. The waveform is slightly shifted to the right and the magnitude too has decreased. There is a sinusoidal distortion in the output waveform. I am not satisfied with the result.

Question D

Modified Butterworth filter

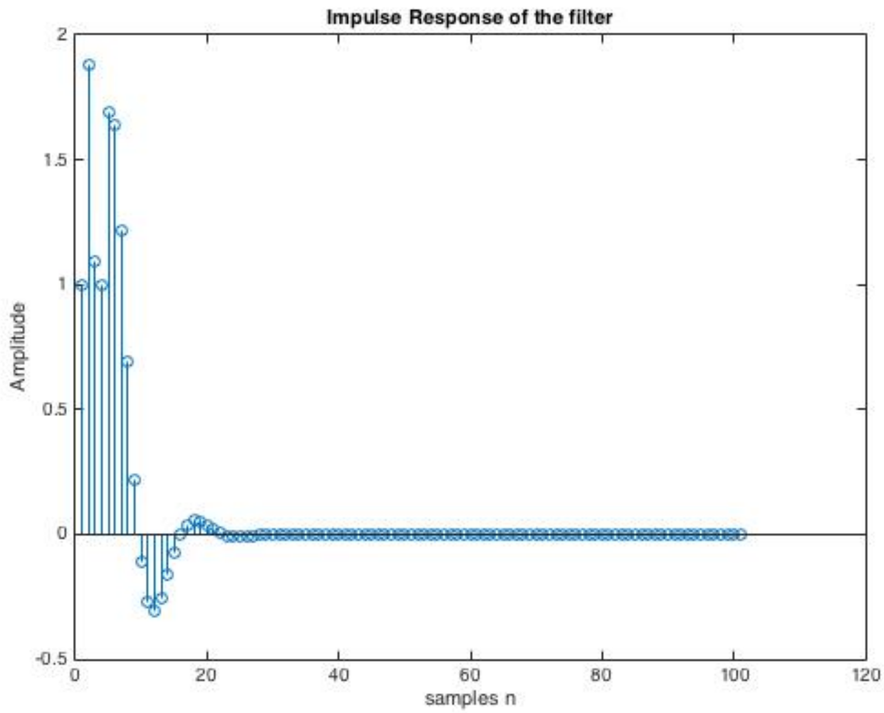
Program

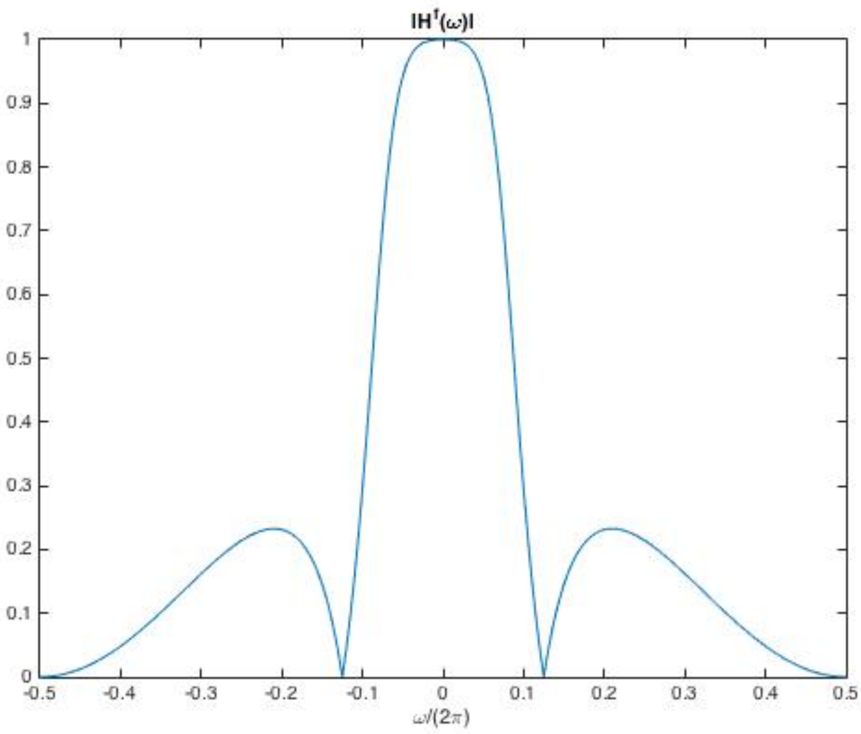
```
clear all;close all;
x=input('Enter the order of the filter');
y=input('Enter the w (0 to 1) for null freq')

for i=1:x
   x1(i)=-1;
   x2(i)=1;
end
w=pi*y;
j=sqrt(-1);
e1=exp(j*w);
e1conj=exp(-j*w);
om = linspace(-pi, pi, 201);
b=poly([x1 e1 e1conj]);

f1=poly([x1 x1 e1 e1conj])
f2=poly([x2 x2 0 0]);

k=100;
p=f1+k*f2;
% zplane(p);
r2=roots(p)
r=r2( abs(r2) < 1 )   % select roots inside unit circle for stability
a=poly(r)

% Impulse Response
imp=[1 zeros(1,100)];
h=filter(b,a,imp);
figure,stem(h)
xlabel('samples n');
ylabel('Amplitude');
title('Impulse Response of the filter');

%Frequency Response
m=(polyval(b,1))/(polyval(a,1));
Hf=(polyval(b,exp(j*om))./polyval(a,exp(j*om)))/m;
figure,plot(om./(2*pi),abs(Hf));
title('|H^f(\omega)|')
xlabel('\omega/(2\pi)')


% Pole zero diagram
figure,zplane(b,a)
title('Pole zero diagram of butterworth filter of order n and null freq')
```

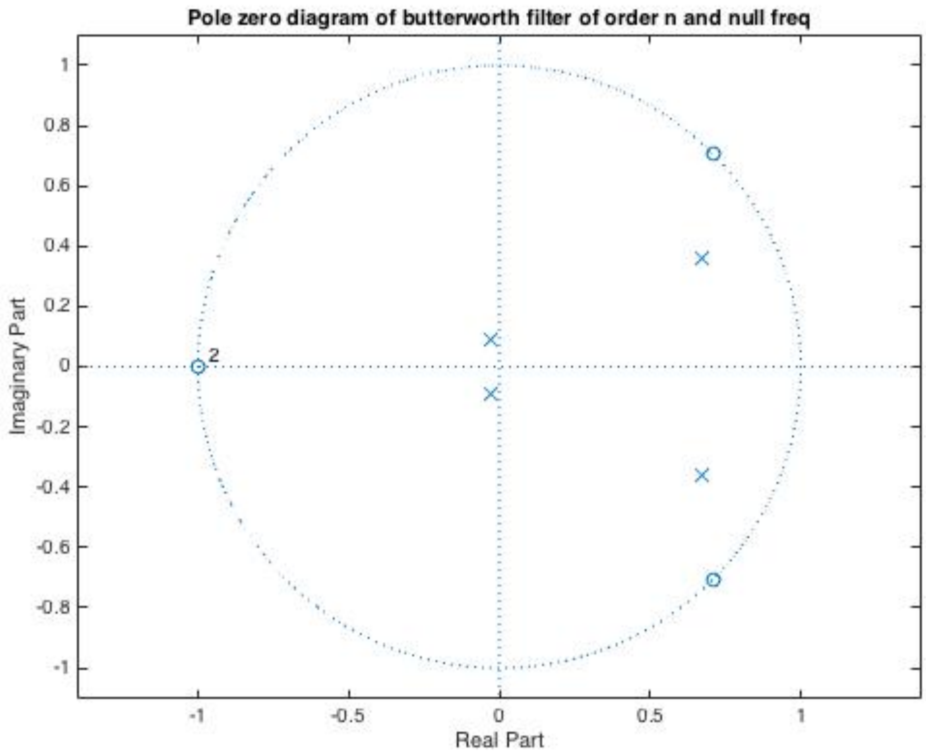
Result

```
>> Modified_butter
Enter the order of the filter2
Enter the w (0 to 1) for null freq0.25

y =

    0.2500


f1 =

    1.0000    2.5858    1.3431   -0.4853    1.3431    2.5858    1.0000


r2 =

   1.3188 + 0.5210i
   1.3188 - 0.5210i
   0.6747 + 0.3587i
   0.6747 - 0.3587i
  -0.0262 + 0.0880i
  -0.0262 - 0.0880i


r =

   0.6747 + 0.3587i
   0.6747 - 0.3587i
  -0.0262 + 0.0880i
  -0.0262 - 0.0880i


a =

    1.0000   -1.2971    0.5217    0.0192    0.0049
```

Impulse Response


Impulse Response of the filter
Amplitude
samples n
2
1.5
1
0.5
0
-0.5
0
20
40
60
80
100
120


|H^f(ω)|
ω/(2π)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5



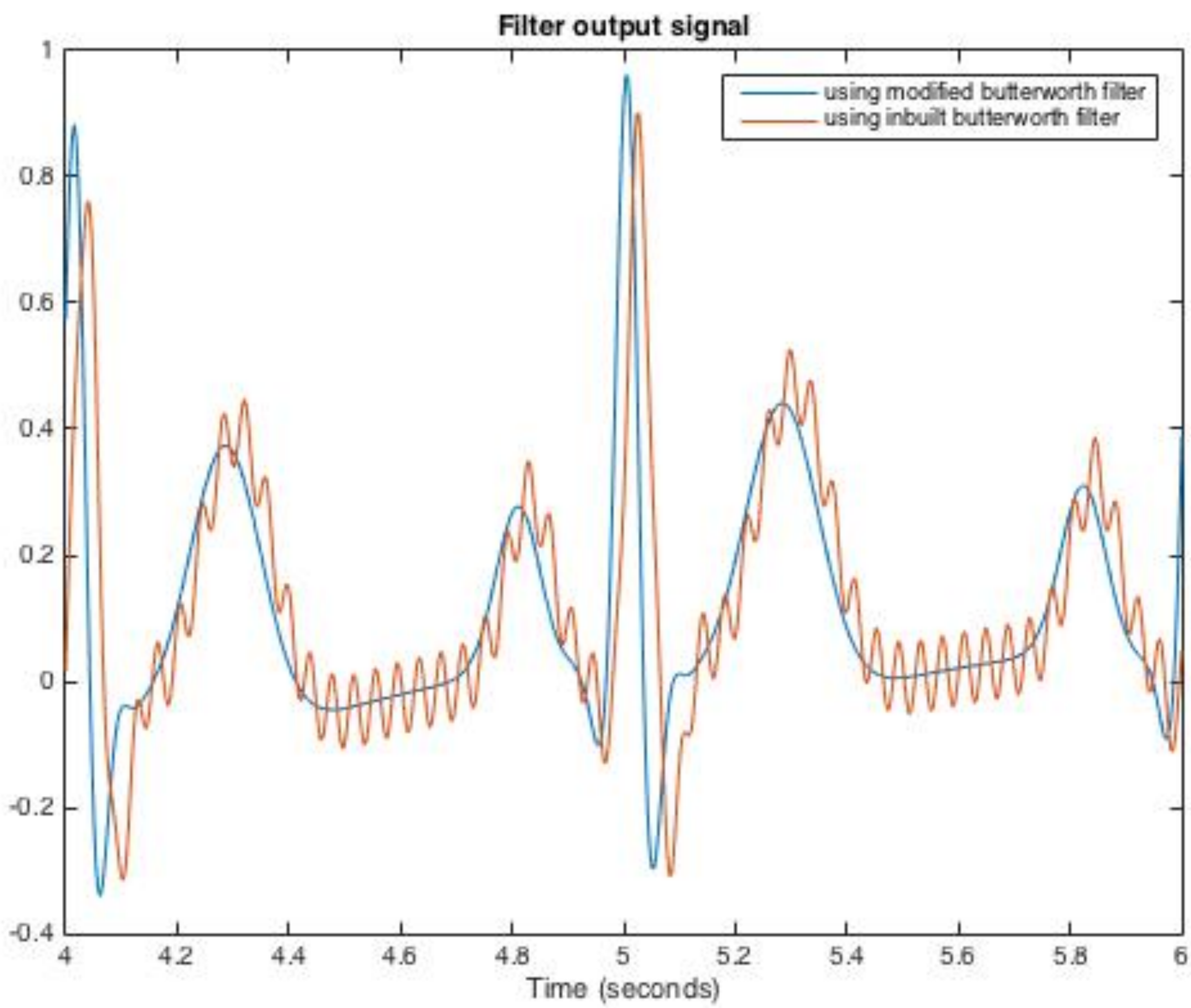
Question E

```
clear all;close all;

% load ECG signal
Fs=256;
x=ecgsyn(Fs,10);
n=0:length(x)-1;
t=n/Fs;

% Noise signal
N=length(x);
sigma=0.2;                  % sigma : noise standard deviation
noise=sigma*randn(N, 1);    % noise : white Gaussian noise
g1=x+noise;                 % g1 : noisy ECG (white noise)
g2=x+cos(0.2*pi*n');        % g2 : ECG with white noise and tonal noise
g3=x+cos(0.2*pi*n');        % g3 : ECG with tonal noise
figure,plot(t,g2);
title('Noisy ECG signal')
xlabel('Time in sec');
xlim([4 7]);

% Fourier Transform of noise signal

ff=fftshift(fft(g2,N));%Fourier Transform of noise signal
t1=(-N/2:N/2-1)/N;
f2=fftshift(fft(x,N))% Fourier transform of original signal
```

```
figure,plot(t1,abs(ff),t1,abs(f2));
legend('noisy ecg','true ecg');
xlabel('Normalized Frequency');
ylabel('Magnitude');
title('Double Sided FFT - with FFTShift of noise signal');

Order=input('Enter the order of the filter');
fre=input('Enter the w (0 to 1) for null freq')

for i=1:Order
   x1(i)=-1;
   x2(i)=1;
end
w=pi*fre;
j=sqrt(-1);
e1=exp(j*w);
e1conj=exp(-j*w);
om = linspace(-pi, pi, 201);
b=poly([x1 e1 e1conj]);

f1=poly([x1 x1 e1 e1conj])
f2=poly([x2 x2 0 0]);

k=100;
p=f1+k*f2;
% zplane(p);
r2=roots(p)
r=r2( abs(r2) < 1 )   % select roots inside unit circle for stability
a=poly(r)

% Impulse Response
imp=[1 zeros(1,100)];
h=filter(b,a,imp);
figure,stem(h)
xlabel('samples n');
ylabel('Amplitude');
title('Impulse Response of the filter');

%Frequency Response
m=(polyval(b,1))/(polyval(a,1));
Hf=(polyval(b,exp(j*om))./polyval(a,exp(j*om)))/m;
figure,plot(om./(2*pi),abs(Hf));
title('|H^f(\omega)|')
xlabel('\omega/(2\pi)')


% Pole zero diagram
figure,zplane(b,a)
title('Pole zero diagram of butterworth filter of order n and null freq')
```

```
% Filter using inbult butterworth filter
[b1,a1]=butter(3,0.08); %Third order filter with 0.3pi radian/sec as cutoff freq
y1=filter(b1,a1,g3);

% Apply low pass filter to noisy ECG signal
y=filter(b,a,g3);
figure,plot(t,y/8,t,y1)
legend('using modified butterworth filter','using inbuilt butterworth filter')
title('Filter output signal')
xlabel('Time (seconds)')
xlim([4 6]);
```

Results



Here, we can see that by modifying the butter worth filter , we were able to remove the tonal noise of frequency 0.2pi in this case. I used 2nd order modified butter worth filter and 3rd order Matlab inbuilt butterworth filter. In modified butterworth filter there is no oscialltion after filtering the ecg signal with tonal noise but in inbuilt function there is still oscillation in filtered output ,which is undesirable and not satisfactory.