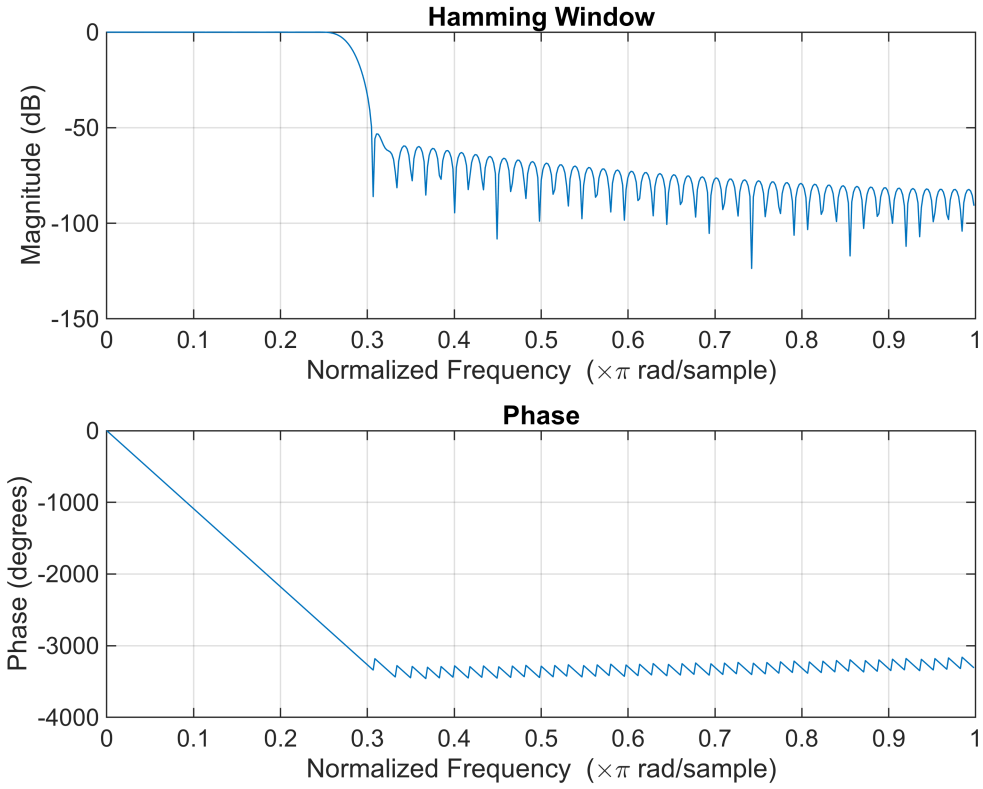
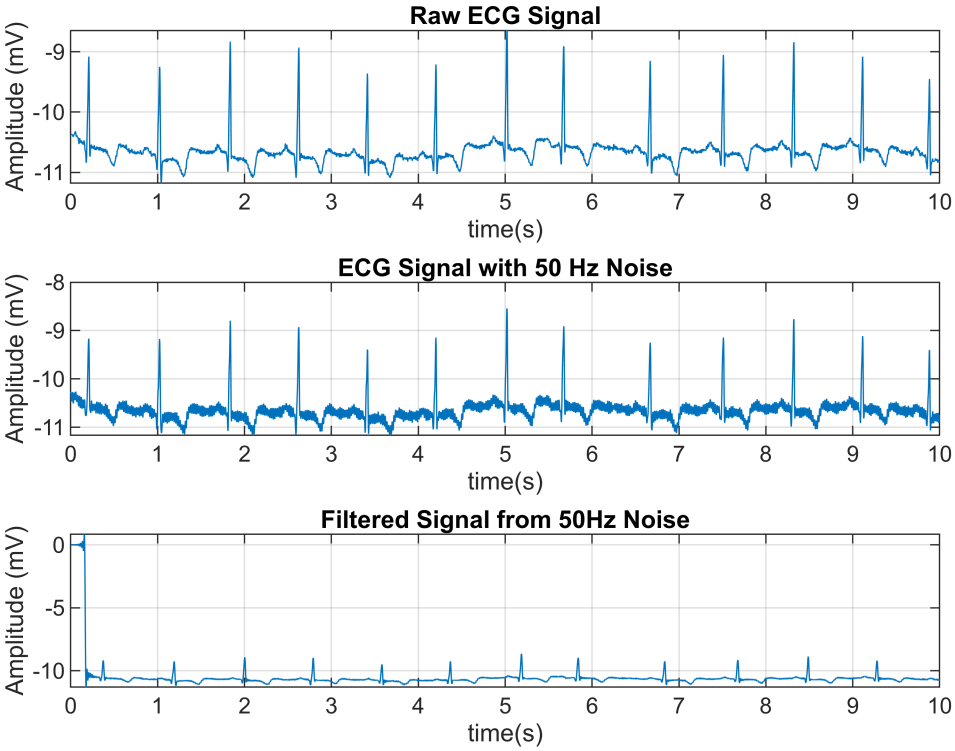
```
% FIR Filter design using fir1() with Hamming Window
clc;
clear all;
close all;

load('100m.mat')

ECGsignal_original = (val -1024)/100;
Fs = 360;

%Frequency of PowerLine Interference
f1 = 50;

t = (0:length(ECGsignal_original)-1)/Fs;

%Generating Noise for f1 = 50 Hz
N= 0.1*cos (2*pi*f1*t);

%Adding Noise to ECG signal
ECG_Noise = ECGsignal_original + N;

%Specifications
wc = 0.2778*pi;
R_p = 1;
R_s = 40;
ws = 0.305*pi;
wp = 0.25*pi;
dw=ws-wp;

%Order of Filter
M_hamming=6.64*pi/dw;

Y=fir1(ceil(M_hamming),wc/pi);

%Transfer Function for FIR Filter
tf = tf(Y,1,1/Fs)
```

```
tf =

  0.0002397 z^121 + 0.0004291 z^120 + 0.0003183 z^119 - 3.935e-05 z^118 - 0.0004133 z^117 - 0.0005286 z^116 - 0.000

Sample time: 0.0027778 seconds
Discrete-time transfer function.
Model Properties
```

```
freqz(Y,1);
title("Hamming Window")
```




```
%Filtering the Noisy Signal
ECG_Filtered = filter(Y,1,ECG_Noise);

%Plotting
subplot(3,1,1)
plot(t,ECGsignal_original)
xlabel('time(s)')
ylabel('Amplitude (mV)')
title('Raw ECG Signal')
grid on
subplot(3,1,2)
plot(t,ECG_Noise)
xlabel('time(s)')
ylabel('Amplitude (mV)')
title('ECG Signal with 50 Hz Noise')
grid on
subplot(3,1,3)
plot(t,ECG_Filtered)
xlabel('time(s)')
ylabel('Amplitude (mV)')
title('Filtered Signal from 50Hz Noise')
grid on
```


Raw ECG Signal
Amplitude (mV)
-9
-10
-11
0
1
2
3
4
5
6
7
8
9
10
time(s)
ECG Signal with 50 Hz Noise
Amplitude (mV)
-8
-9
-10
-11
0
1
2
3
4
5
6
7
8
9
10
time(s)
Filtered Signal from 50Hz Noise
Amplitude (mV)
0
-5
-10
0
1
2
3
4
5
6
7
8
9
10
time(s)