INDEX

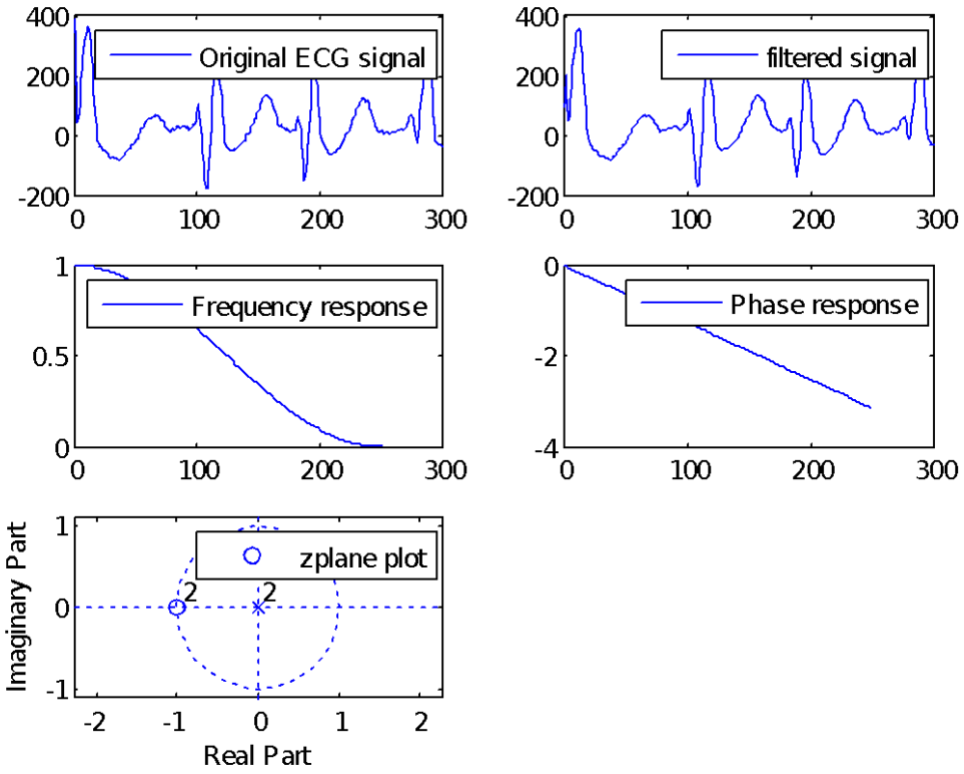
BMDSP MATLAB PROGRAMS

C-PROGRAMS

Prepared by Abhishek Appaji

1. **Implement a Hanning filter to smoothen the ECG data and plot pole zero plot using Matlab function**

```
%To display ECG signal
 load ECGS1.dat;
x=ECGS1(1:300);
subplot(3,2,1); plot(x); legend('Original ECG signal');
a=[1 2 1];
b=[4];
fs=250;
r=filter(a,b,x);
subplot(3,2,2); plot(r); legend('filtered signal');

 % Frequency response
w=0:0.01:pi;
subplot(3,2,3);
[h,m]=freqz(a,b,w);
disp(h); disp (m);
plot((m/pi)*fs,abs(h)); legend('Frequency response');

%Phase response
subplot(3,2,4);
plot((m/pi)*fs,angle(h)); legend('Phase response');
subplot(3,2,5); zplane(a,b); legend('zplane plot');
c=roots(a);  % c=1,-1
d=angle(c); %d= 3.1416, 3.1416
```



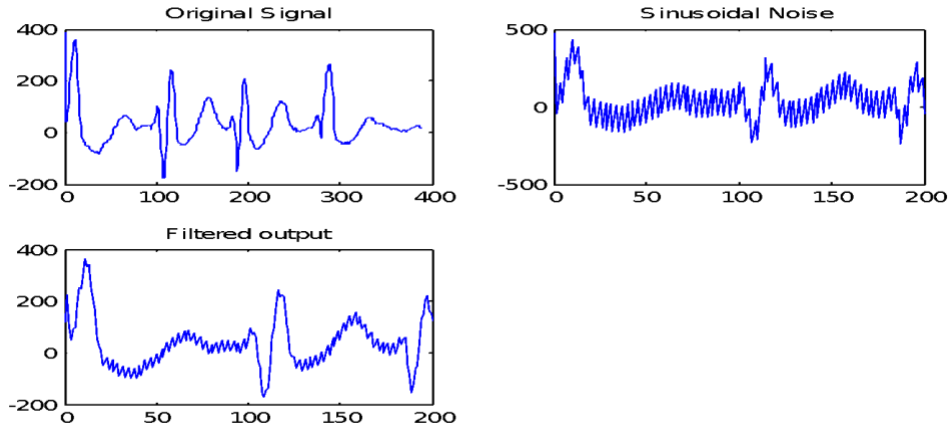
2. **Write a Matlab program to eliminate 60Hz noise. Plot pole zero plot.**

%60Hz Sinusoidal Noise

```
load ECGS1.dat;
n=1:200;
w=100*sin(2*pi*n*(60/180));
s=w(1:200)+ECGS1(1:200);
subplot(2,2,1); plot(ECGS1); title('Original Signal');
subplot(2,2,2); plot(s); title('Sinusoidal Noise');
```

%Filter 60Hz Noise

```
a=[1 2 1]; b=[4];
f=filter(a,b,s);
subplot(2,2,3); plot(f);title('Filtered output');
```



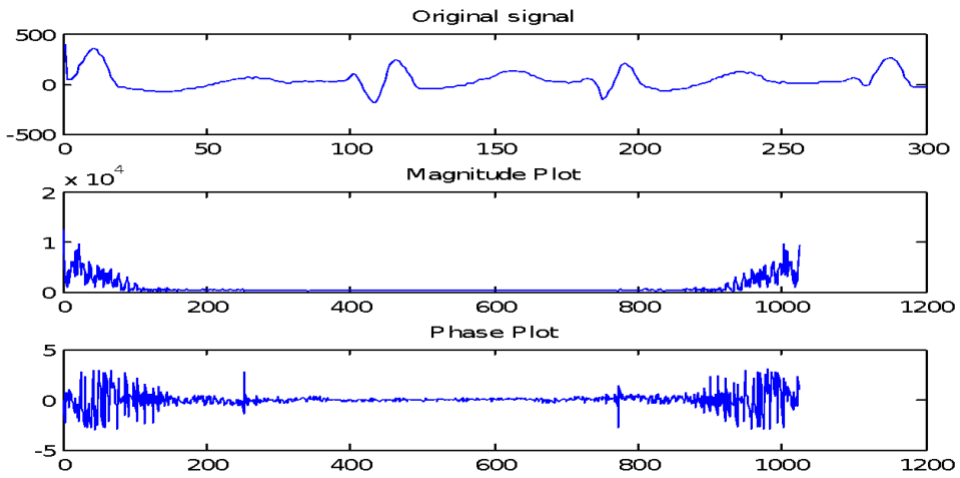
3. **Write a Matlab program to plot the magnitude and phase spectrum of the given ECG signal**

%Load ECG

```
load ECGS1.dat;
x=ECGS1(1:300);
subplot(3,1,1); plot(x); title('Original signal');
```

%Magnitude & Phase Plot

```
a=fft(ECGS1,1024);
subplot(3,1,2);plot(abs(a)); title('Magnitude Plot');
subplot(3,1,3);plot(angle(a)); title('Phase Plot');
```

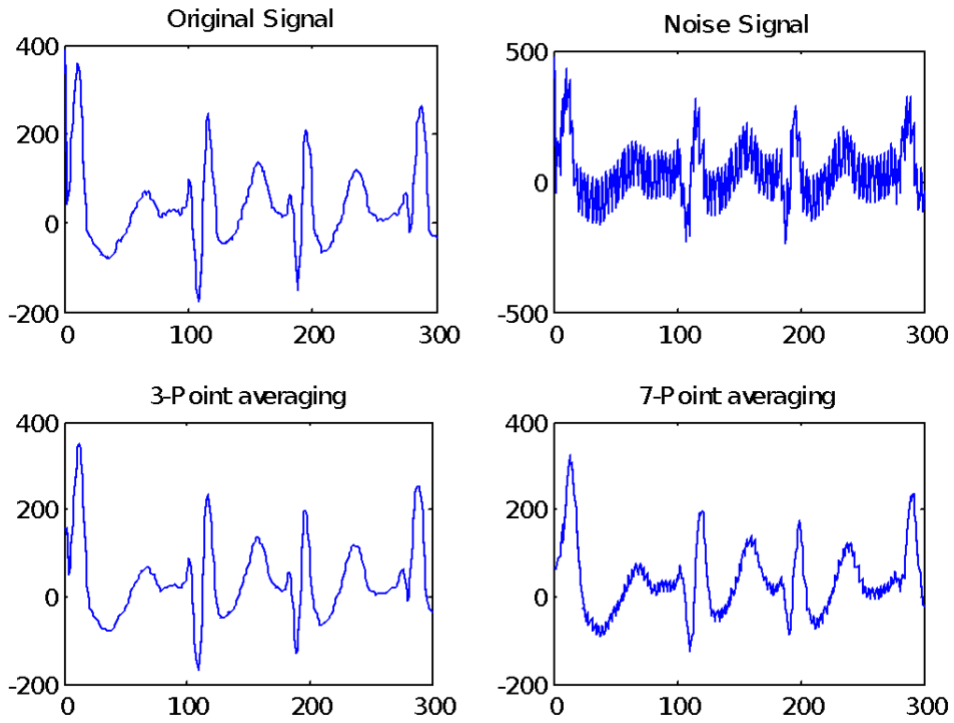


4. **Using Matlab function implement the following difference equation to average the given ECG data and conclude results.**

   y(n)=1/7[x(n) + x(n-1) +……..+ x(n-6)]

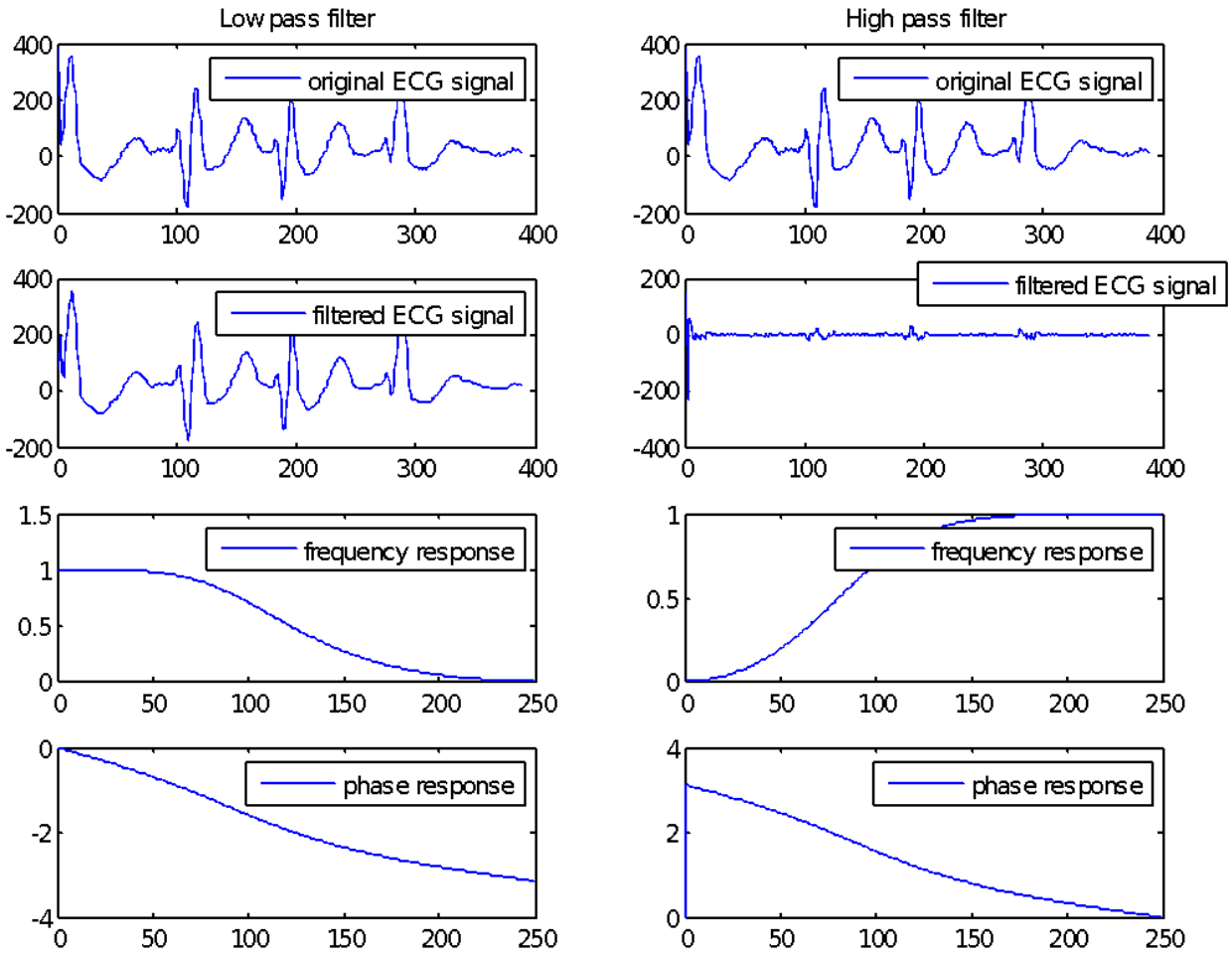
   y(n)=1/3[x(n) + x(n-1) +x(n-2)]

```
% Load ECG
load('ECGS1.dat');
x=ECGS1(1:300);
subplot(2,2,1); plot(x); title('Original Signal');

%Noise Component
n=1:300;
r=100*sin(2*pi*n*(60/180));
s=x+r;
subplot(2,2,2);plot(s);title('Noise Signal');

%3-Point Averaging
b=[1 1 1];
a=3;
y=filter(b,a,s);
subplot(2,2,3); plot(y); title('3-Point averaging');

%7-Point Averaging
d=[1 1 1 1 1 1 1];
c=7;
z=filter(d,c,s);
subplot(2,2,4); plot(z); title('7-Point averaging');
```



5. **Write a Matlab program to retain the signal frequencies below 50Hz in a given ECG data. Plot Magnitude and phase response of the filter**
6. **Write a Matlab program to retain the signal frequencies above 50Hz in a given ECG data. Plot Magnitude and phase response of the filter**

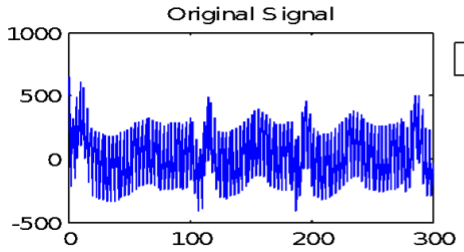
```
load ECGS1.dat;
x=ECGS1(1:388);
subplot(4,1,1); plot(x); legend('original ECG signal');
fc=input('Enter the cut off frequency:');
n=2; fs=250;
wn=fc/fs*2;
[a,b]=butter(n,wn,'low');  % 'high' for high pass
h=filter(a,b,x);
subplot(4,1,2); plot(h); legend('filtered ECG signal');
w=0:0.01:pi;
[h,m]=freqz(a,b,w);
subplot(4,1,3); plot((m/pi)*fs,abs(h),''); legend('frequency response');
subplot(4,1,4); plot((m/pi)*fs,angle(h),''); legend('phase response');
```

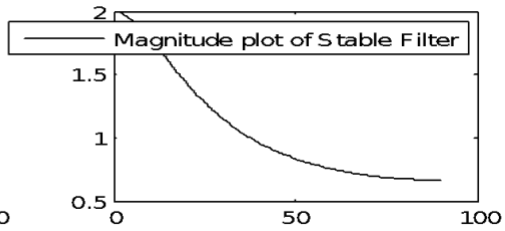


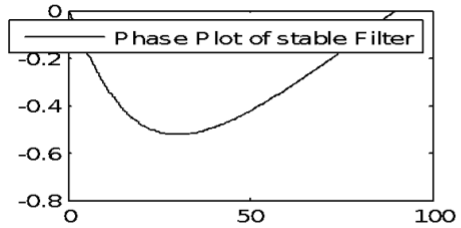
7. Write Matlab programs to implement the given transfer function. H(Z)= 1/1- βZ $^{-1}$ for β = 1/2,1,2. Plot the magnitude and phase responses, pole zero plot

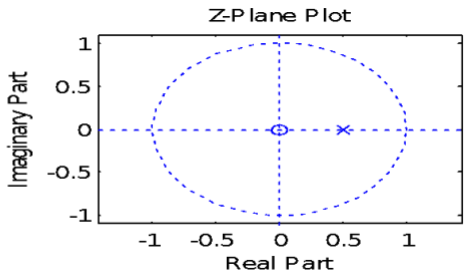
```
load ECGS1.dat;
e=ECGS1(1:300);
n=1:300;
x=300*sin(2*pi*n*(60/180));
x=x+e;
subplot(2,2,1); plot(x); title('Original Signal');

b=1;
a=[1 -1/2];  % Stable Filter Coeff
a1=[1 -2];   % Unstable filter Coeff
a2=[1 -1];   % Marginally Stable filter coeff
fs=180/2;
h=filter(b,a,x); h1=filter(b,a1,x); h2=filter(b,a2,x);
w=0:0.01:pi;
[h,m]=freqz(b,a,w); [h1,m1]=freqz(b,a1,w); [h2,m2]=freqz(b,a2,w);
```

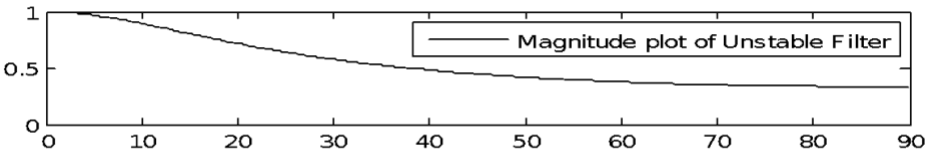
```
subplot(2,2,2); plot((m/pi)*fs, abs(h),'k');
legend('Magnitude plot of Stable Filter');
subplot(2,2,3); plot((m/pi)*fs,angle(h),'k');
legend('Phase Plot of stable Filter');
subplot(2,2,4); zplane(b,a); title('Z-Plane Plot');

figure,
subplot(3,1,1); plot((m/pi)*fs, abs(h1),'k');
legend('Magnitude plot of Unstable Filter');
subplot(3,1,2); plot((m/pi)*fs,angle(h1),'k');
legend('Phase Plot of Unstable Filter');
subplot(3,1,3); zplane(b,a1); title('Z-Plane Plot');

figure,
subplot(3,1,1); plot((m/pi)*fs, abs(h2),'k');
legend('Magnitude plot of Marginally Stable Filter');
subplot(3,1,2); plot((m/pi)*fs,angle(h2),'k');
legend('Phase Plot of Marginally stable Filter');
subplot(3,1,3); zplane(b,a2); title('Z-Plane Plot');
```

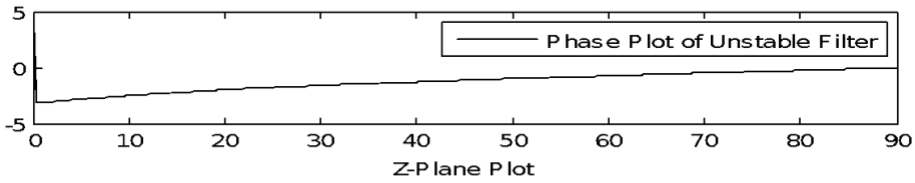

Original Signal
1000
500
0
-500
0
100
200
300

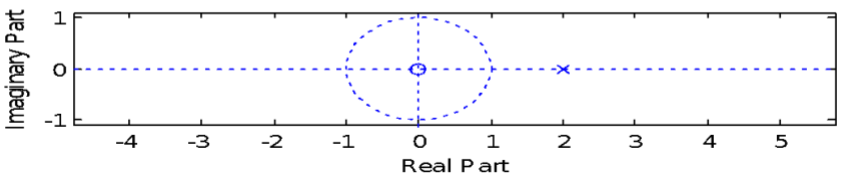

Magnitude plot of Stable Filter
2
1.5
1
0.5
0
50
100

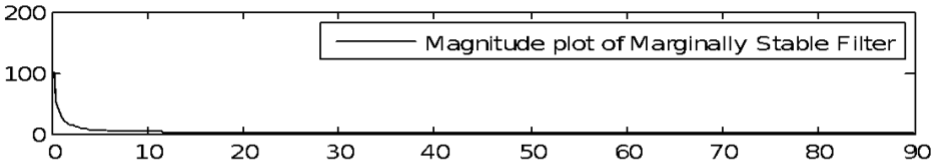

Phase Plot of stable Filter
0
-0.2
-0.4
-0.6
-0.8
0
50
100

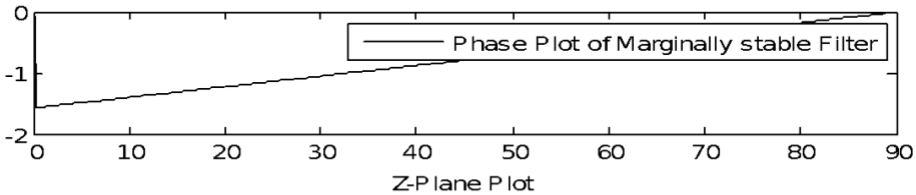

Z-Plane Plot
Imaginary Part
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
Real Part

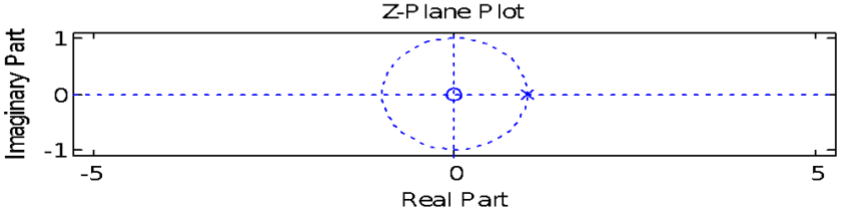

Magnitude plot of Unstable Filter
1
0.5
0
0
10
20
30
40
50
60
70
80
90


Phase Plot of Unstable Filter
5
0
-5
0
10
20
30
40
50
60
70
80
90


Z-Plane Plot
Imaginary Part
1
0
-1
-4
-3
-2
-1
0
1
2
3
4
5
Real Part


Magnitude plot of Marginally Stable Filter
200
100
0
0
10
20
30
40
50
60
70
80
90


Phase Plot of Marginally stable Filter
0
-1
-2
0
10
20
30
40
50
60
70
80
90


Z-Plane Plot
Imaginary Part
1
0
-1
-5
0
5
Real Part

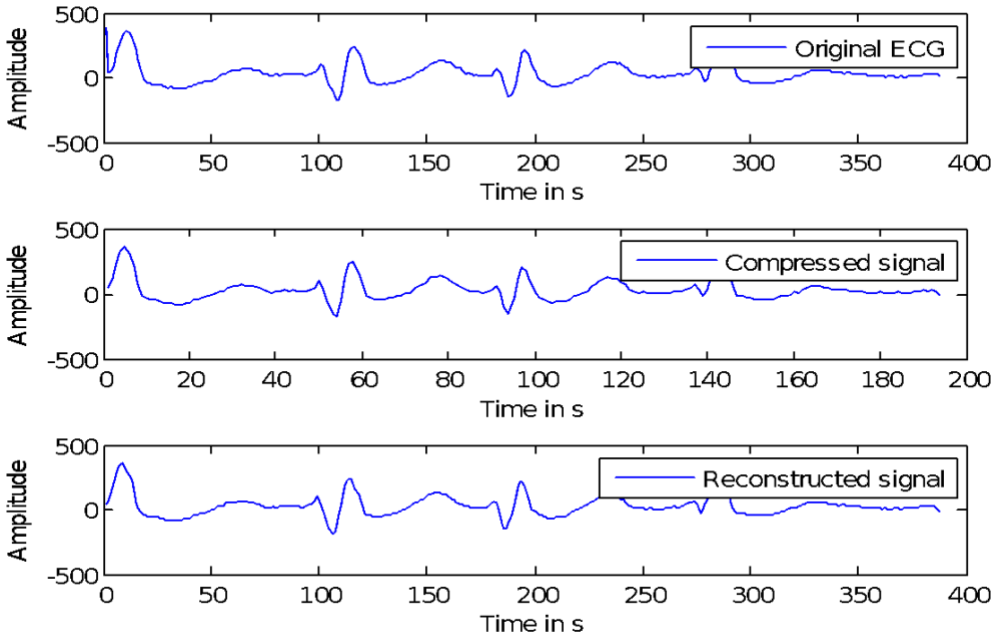
8. Write a Matlab program to compress the given ECG data using Turning Point Algorithm

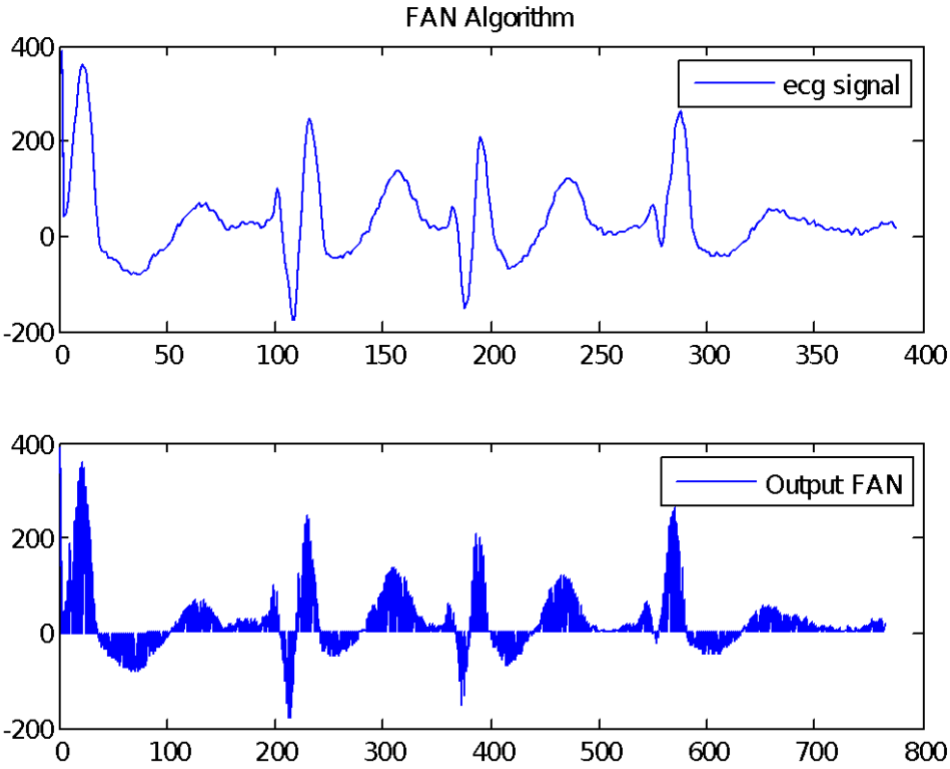
```
ecg=load('ECGS1.dat');
n=length(ecg);
compecg=zeros(1,n/2);          %compressed ecg vector
compecg(1)=ecg(1);             %save first sample
p=0;                           %index for compressed ecg
for k=1:2:n-2
   s1=sign(ecg(k+1)-ecg(k));
   s2=sign(ecg(k+2)-ecg(k+1));
   if(s2-s1)>0
      compecg(p+1)=ecg(k+1);
   else
      compecg(p+1)=ecg(k+2);
   end
   p=p+1;
end
compecg=compecg';      %convert to col vector
rececg=interp(compecg,2);      %reconstruct ecg signal
subplot(3,1,1); plot(ecg); legend('Original ECG');
xlabel('Time in s'); ylabel('Amplitude');
subplot(3,1,2); plot(compecg); legend('Compressed signal');
xlabel('Time in s'); ylabel('Amplitude');
subplot(3,1,3); plot(rececg); legend('Reconstructed signal');
xlabel('Time in s'); ylabel('Amplitude');

cr=length(compecg)/length(ecg);
n1=length(compecg);
n2=length(rececg);
disp('length of ECG is:'); disp(n);                          % 388
disp('length of compressed ECG is:'); disp(n1);     % 194
disp('length of reconstructed ECG is:'); disp(n2);   %388
```

9. Write a Matlab program to compress the given ECG data using Fan Algorithm

```
ecg=load('ECGS1.dat');
p=1;          %index for output fan
k=1;          % index for input ecg
x0=ecg(k);        %original point
fan(p)=x0;     %save first permanent point
k=k+1;
x1=ecg(k);     %save next point
t=1;          %initialize line length
e=0.08;       %set error limit %exp eth different value
xu1=x1+e;  xl1=x1-e;    %upper & lower bound of x1
while (k<length(ecg))
   k=k+1;
   x2=ecg(k);
   xu2=(((xu1-x0)/t)+xu1);
   xl2=(((xu1-x0)/t)+xl1);
   if(x2<=xu2)&(x2>=xl2)
      if(xu2<(xu2+e))
         xu2=xu2;
      else
         xu2=xu2+e;
      end
      if(xl2>(xl2-e))
         xl2=xl2;
      else
         xl2=xl2-e;
      end
      t=t+1;
      x1=x2;
   else
      p=p+1;fan(p)=t;
      p=p+1;
      fan(p)=x1;        %save final length
      x0=x1; x1=x2;     %reset all variables
      xu1=x1+e;       xl1=x1-e;
   end
end
n=length(ecg);
n1=length(fan);
disp('n') ; disp(n); disp('n1');  disp(n1); %n=388 , n1=765
figure;
subplot(2,1,1);plot(ecg); title('FAN Algorithm');
legend('ecg signal');
subplot(2,1,2); plot(fan); legend('Output FAN');
```

TURNING POINT Algorithm

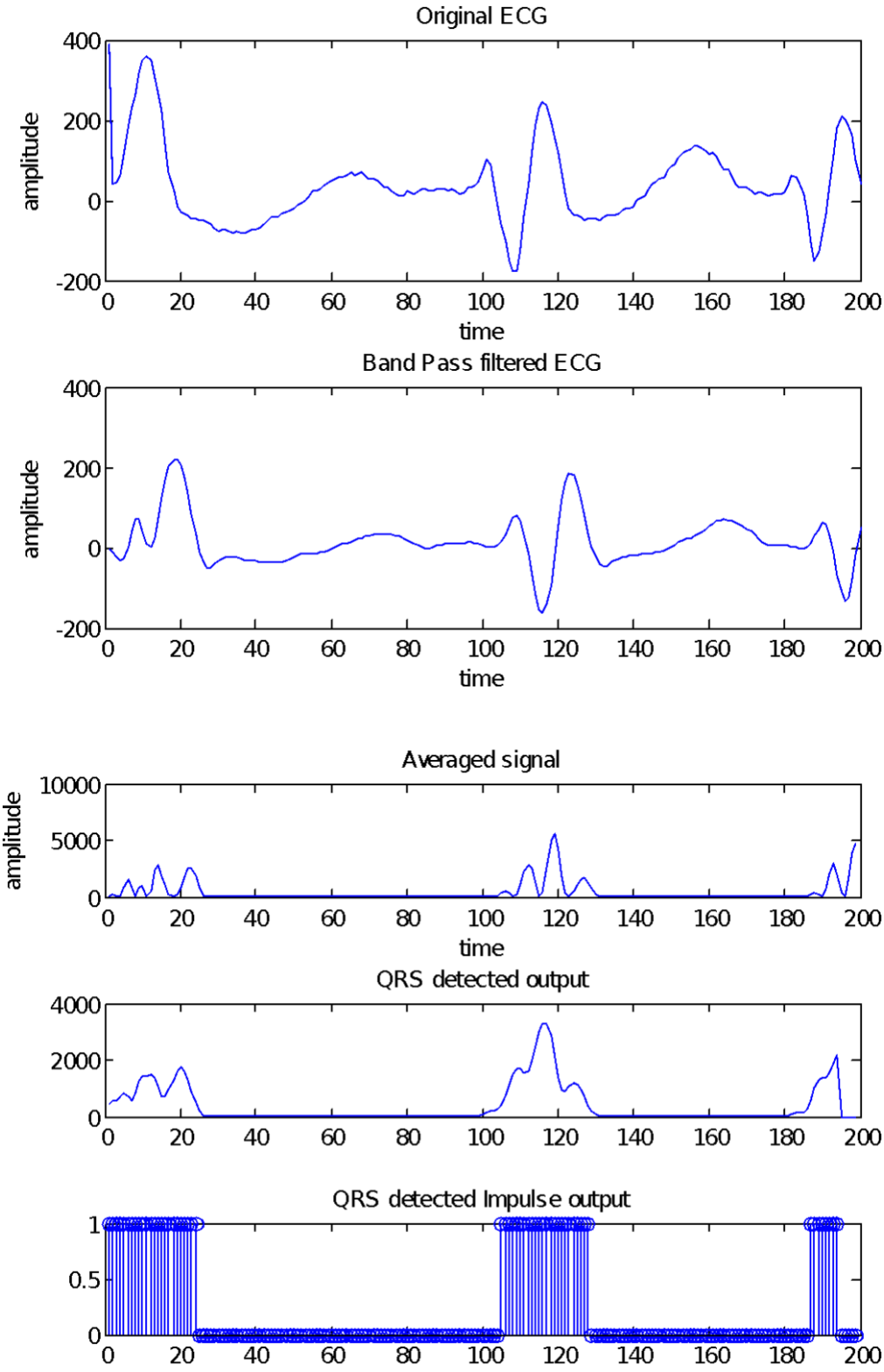




10. Write a Matlab program to detect the QRS component from the given ECG data using differentiation technique

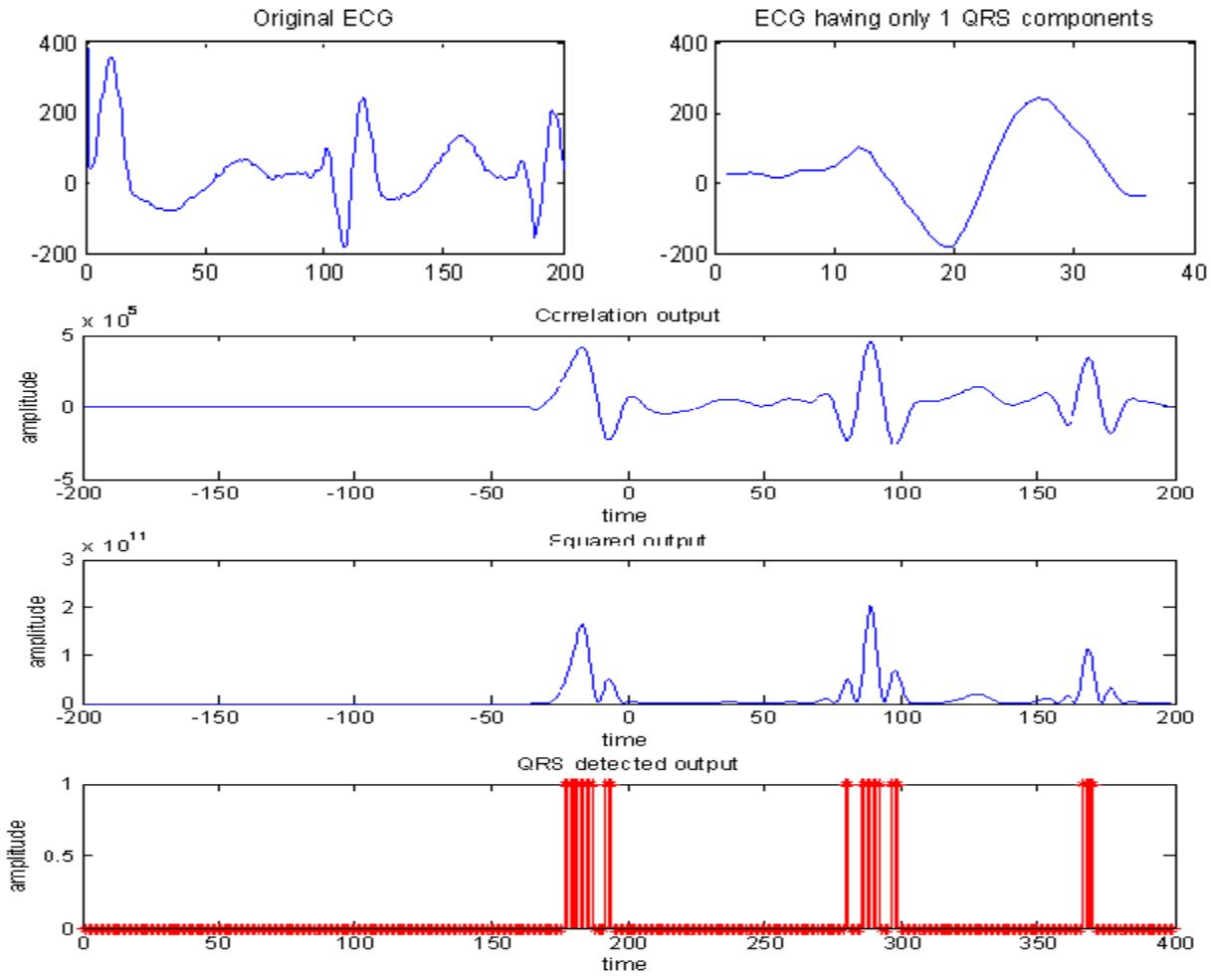
11. .Using Matlab functions, Write a program to find the Heart rate for the given ECG data

```
ecg=load('ECGS1.dat');
ecg1=ecg(1:200);
fs=200;ts=1/fs; t=1:200;

% N=15; % FIR Bandpass filter (14 21 Hz) % QRS energy is centered at 17Hz in ECG spectrum.
fl=14; fH=21; % lower & Upper cutoff frequency
wl=fl/(fs/2); wu=fH/(fs/2); %normalised discrete frequency
w=[wl wu];
b=fir1(N,w,'bandpass'); %fir bandpass filter
y=filter(b,1,ecg1); %bandpass filtered ECG
subplot(2,1,1),plot(t,ecg1);
title('Original ECG'); xlabel('time');ylabel('amplitude');
subplot(2,1,2),plot(t,y);
title('Band Pass filtered ECG'); xlabel('time');ylabel('amplitude');

y1=diff(y); %differentiate y,length of y1 will be equall to (length ofy-1)
y2=y1.^2;
%moving average filter
M=5; % window size
y3=[zeros(1,M) y2]; % append M zeros at the beginning
y4=zeros(size(y2)); %initialize output vector to same size of y2
for k=M+1:length(y2)
   y4(k-M)=mean(y2((k-M):k));
end
figure(2);
title('Differentiated and Square signal'); xlabel('time');ylabel('amplitude');
subplot(3,1,1),plot(t(1:end-1),y2);
title('Averaged signal'); xlabel('time');ylabel('amplitude');
subplot(3,1,2), plot(t(1:end-1),y4); title('QRS detected output');
th=max(y4)*.10; % set threshold
y5=(y4>th);
subplot(3,1,3),stem(y5); title('QRS detected Impulse output');
y6=diff(y5); % produces impulses corresponding to QRS pulses
y7=(y6>0); % neglect negetive impulses
y8=find(y7); % locate indices where +ve impulses occur
%t_qrs=t(y8);  %t_hrt=diff(t_qrs);

% heart rate  in beats/min
ecgp=(max(t)*ts);
heartbeat=(length(y8)*60)/ecgp;
 disp('Heartbeat=');disp(heartbeat);       % y8=2, Heartbeat=120
```


Original ECG
amplitude
time
Band Pass filtered ECG
amplitude
time
Averaged signal
amplitude
time
QRS detected output
QRS detected Impulse output

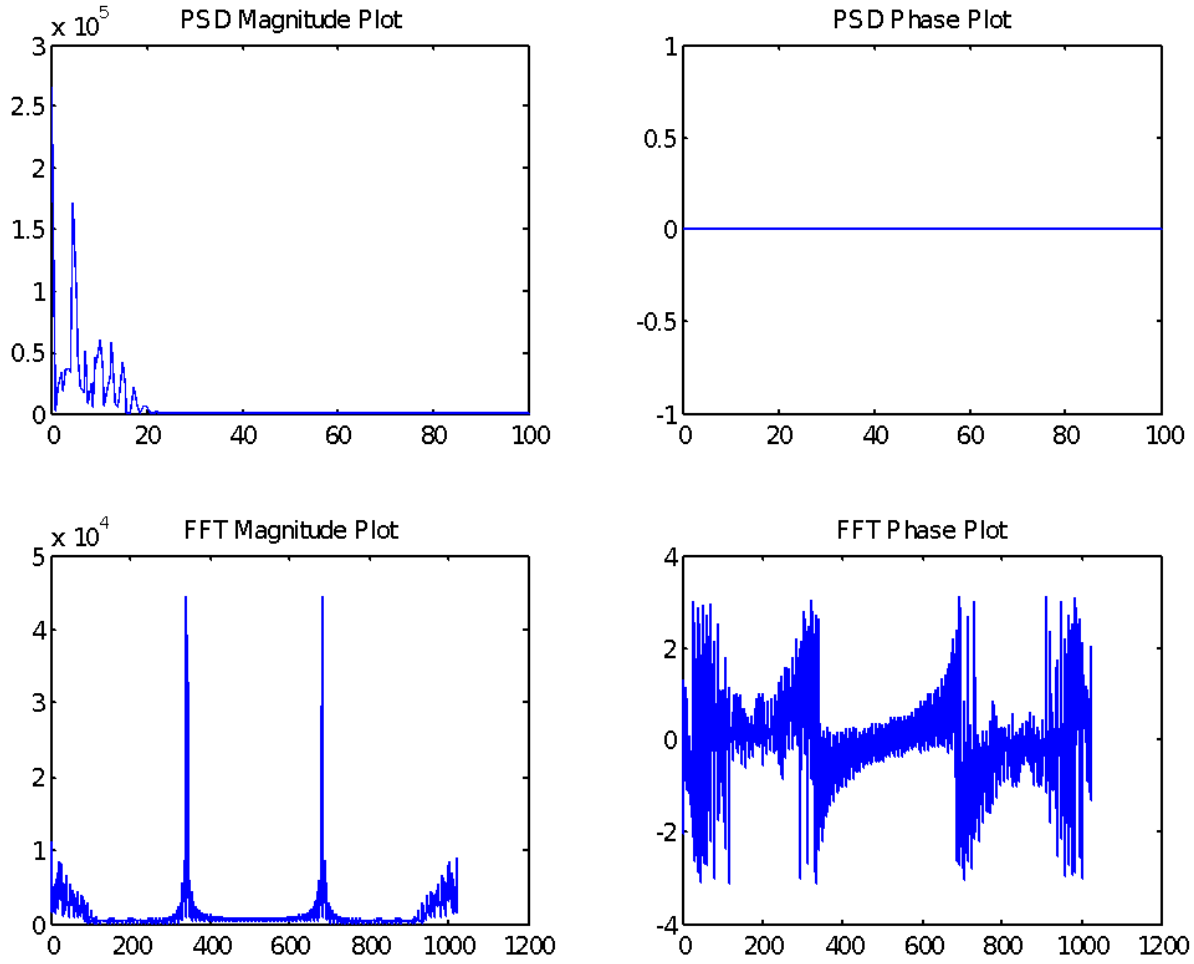
12. Write a Matlab program to detect the QRS component from the given ECG data using Template matching technique

```
load('ECGS1.dat');
ecg=ECGS1(1:200);
qrsecg=ecg(90:125);
t=1:200;
t1=1:length(qrsecg);
subplot(2,1,1),plot(t,ecg);title('Original ECG');
subplot(2,1,2),plot(t1,qrsecg);title('ECG having only 1 QRS components');

[y,lags]=xcorr(ecg,qrsecg);
figure,
subplot(3,1,1); plot(lags(t:end),y(t:end));
title('Correlation output'); xlabel('time');ylabel('amplitude');
%display correlation of positive lag only
y1=(y.^2); %squaring operation
subplot(3,1,2),plot(lags(t:end),y1(t:end));
title('Squared output'); xlabel('time');ylabel('amplitude');
th=0.2*max(y1); %set threshold
y2=(y1>th); %logical array
subplot(3,1,3),stem(y2,'r*');
title('QRS detected output'); xlabel('time');ylabel('amplitude');
```



13. Write a Matlab Program for PSD Estimation for ECG

```
load ECGS1.dat;
n=1:300;
x=sin(2*pi*n*60/180);
y=x*300+ECGS1(1:300);
[P,F]=psd(y,1024,180);
subplot(2,2,1); plot(f,p); title('PSD Magnitude Plot');
subplot(2,2,2); plot(f,angle(p)); title('PSD Phase Plot');
a=fft(y,1024);
subplot(2,2,3); plot(abs(a)); title('FFT Magnitude Plot');
subplot(2,2,4); plot(angle(a)); title('FFT Phase Plot');
```



14. Write a Matlab Program to Plot Power Spectrum of ECG signal

```
load ECGS1.dat;
N=300;
ecg=fft(ECGS1(1:N));
s=1/(N*2);
P(1)=abs(ecg(1))*s;
for k=2:N/2;
   P(k)=abs(ecg(k))+abs(ecg(N+1-k)*s);
end
P(N/2+1)=abs(ecg(N/2+1))*s;
n=0:1/150:1;
plot(n(1:150), P(1:150)); title('Power Spectrum');
```