

---

# Preprocessing

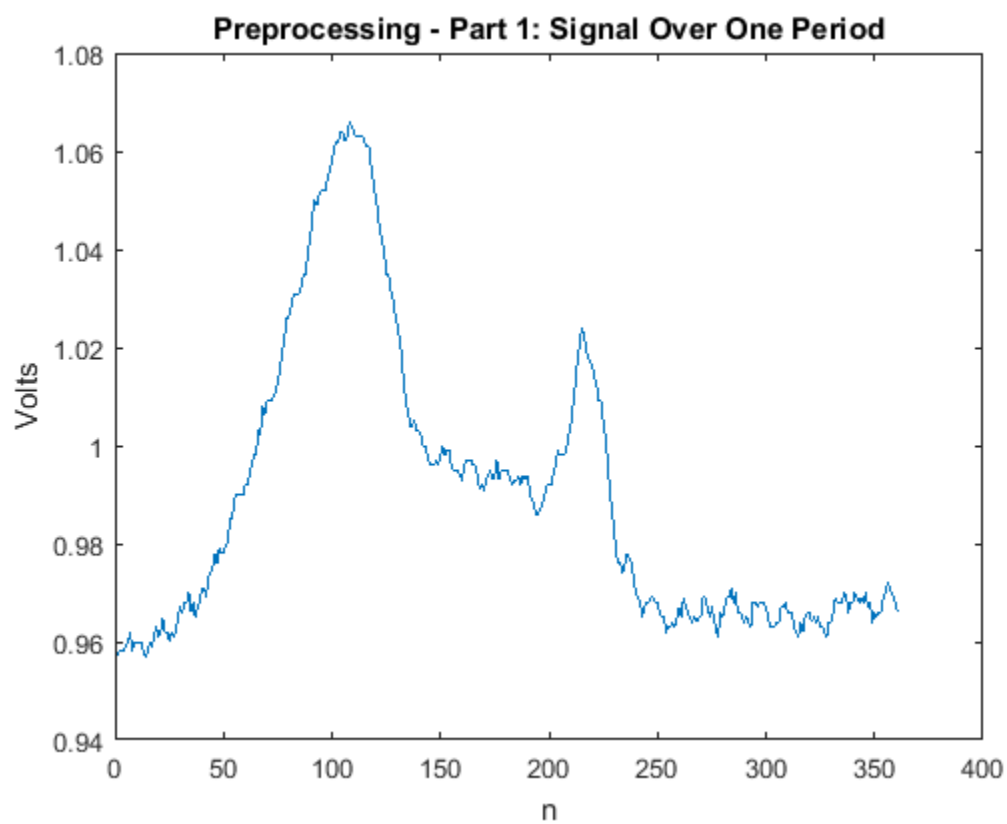
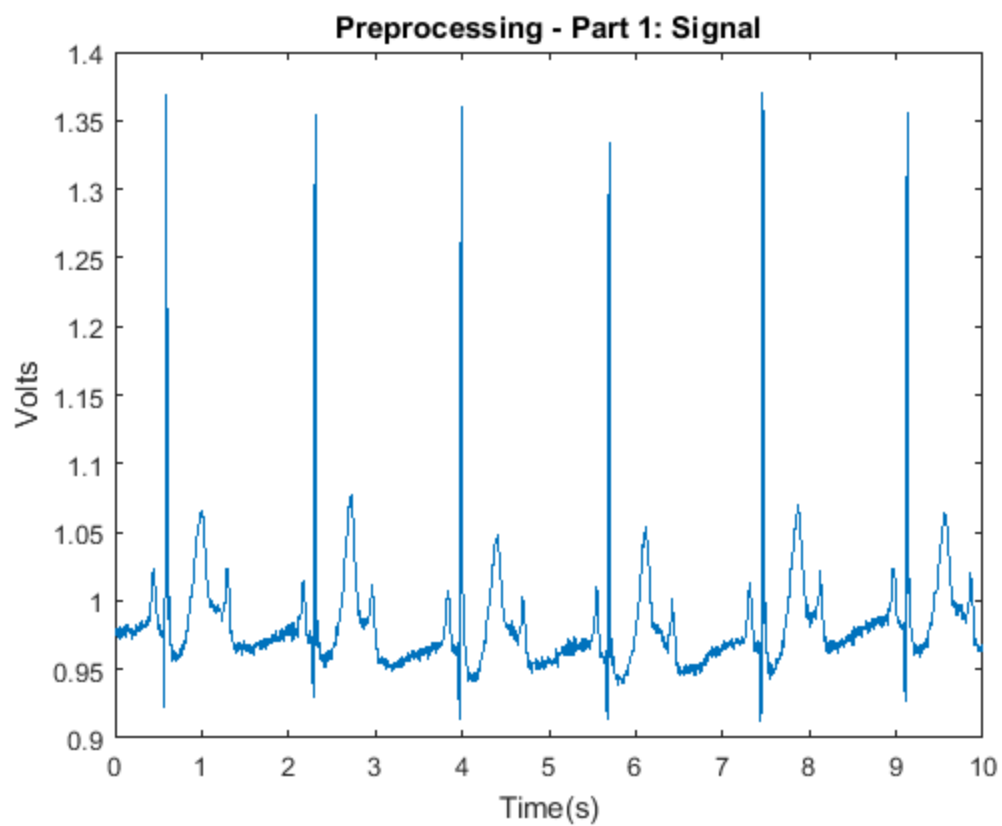
## Table of Contents

Part 1 - Visual Signal Validation .....	1
Part 2 - FFT of the Signal .....	3
Part 3 - Motion Artifact Removal .....	4
Part 4 - Line Noise Removal .....	6
Part 5 - SNR Increasing .....	9

## Part 1 - Visual Signal Validation

SAME AS BEFORE

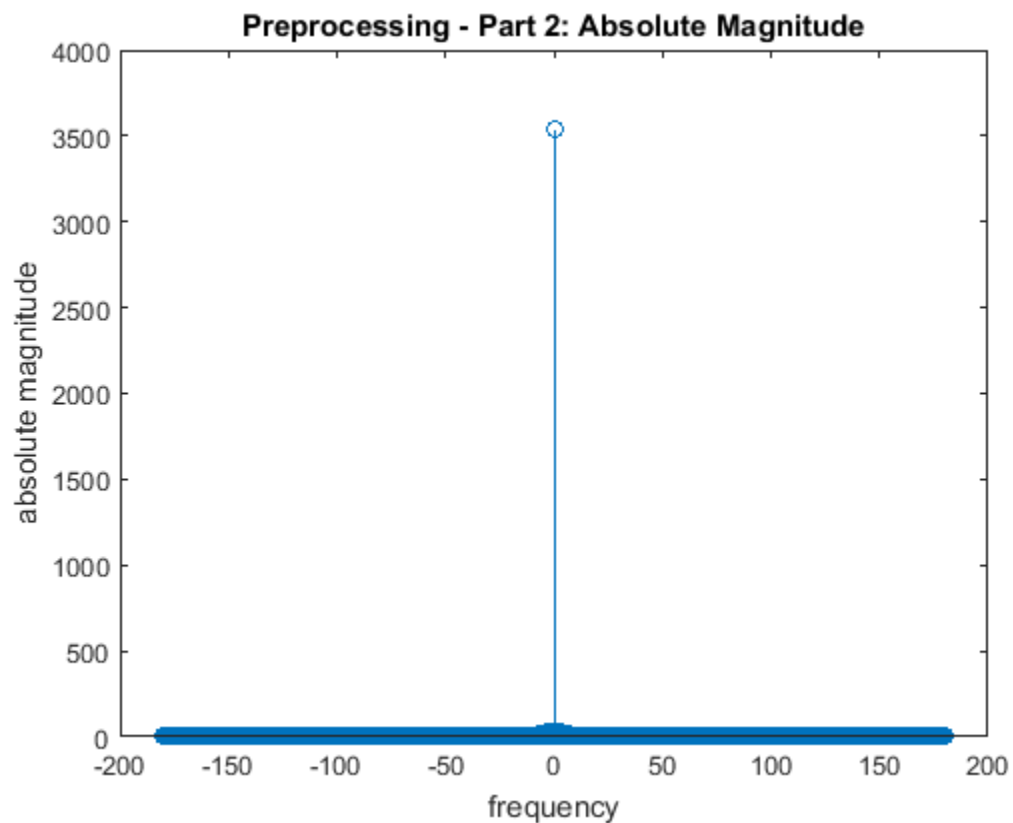
```
ECG = load('patient1.mat');
d = ECG.val/1000; % convert mili Volt to Volt
fs = 360; % constant sampling frequency
unit = 'Volts';
x = (250:+1:250+fs);
t = (1:+1:length(d))./fs;
figure(1);
plot(t,d);
xlabel('Time(s)');
ylabel(unit);
title('Preprocessing - Part 1: Signal');
p = d(x);
figure(2);
plot(p);
ylabel(unit);
xlabel('n');
title('Preprocessing - Part 1: Signal Over One Period');
```

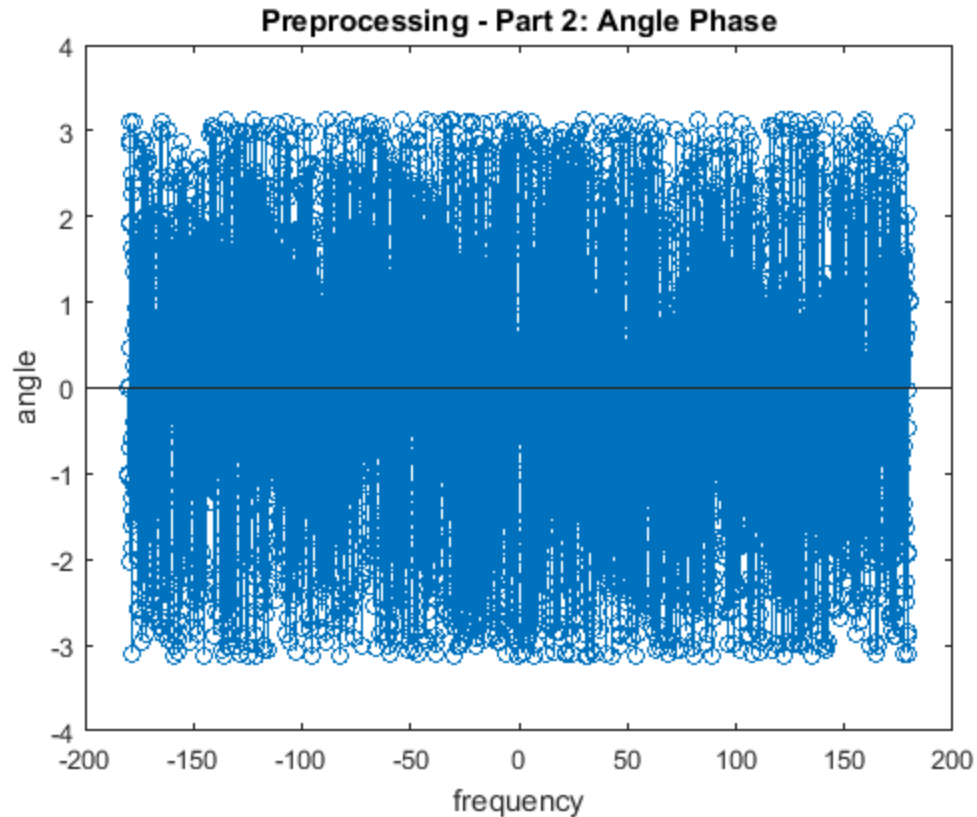


## Part 2 - FFT of the Signal

SAME AS BEFORE

```
y = fft(d);  
d_f = fftshift(y);  
ny = length(y);  
fd_f = (-ny/2:ny/2-1)*(fs/ny);  
figure(3);  
stem(fd_f, abs(d_f));  
xlabel('frequency');  
ylabel('absolute magnitude');  
title('Preprocessing - Part 2: Absolute Magnitude');  
figure(4);  
stem(fd_f, angle(d_f));  
xlabel('frequency');  
ylabel('angle');  
title('Preprocessing - Part 2: Angle Phase');
```





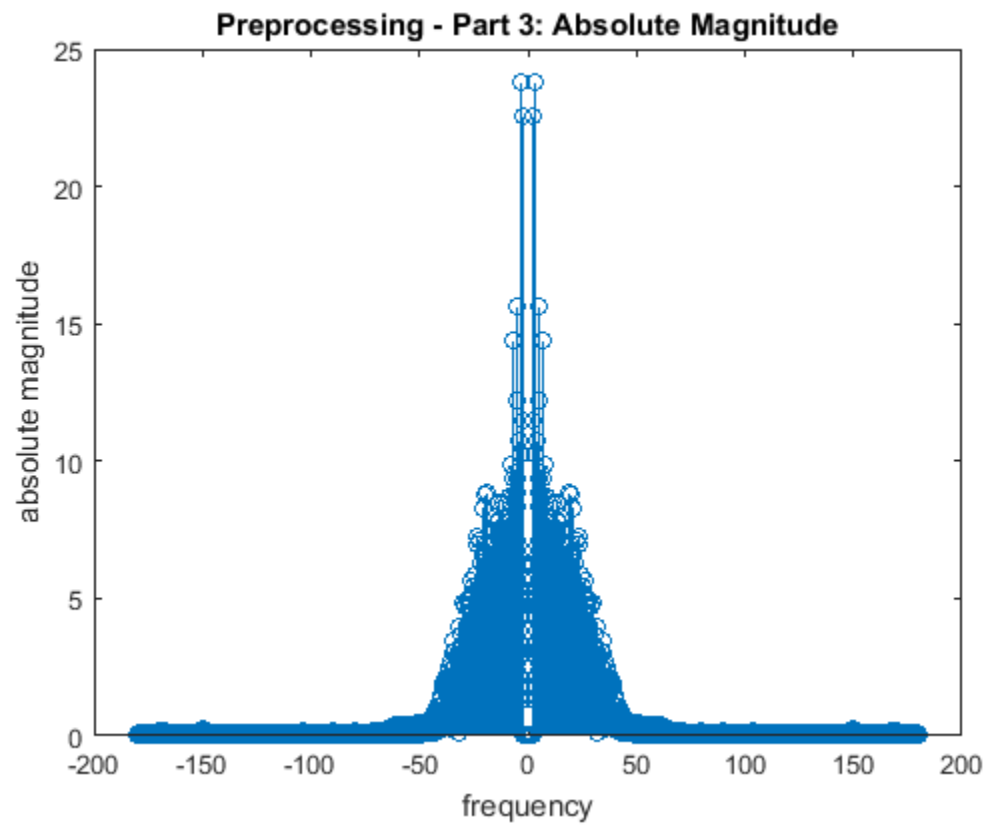
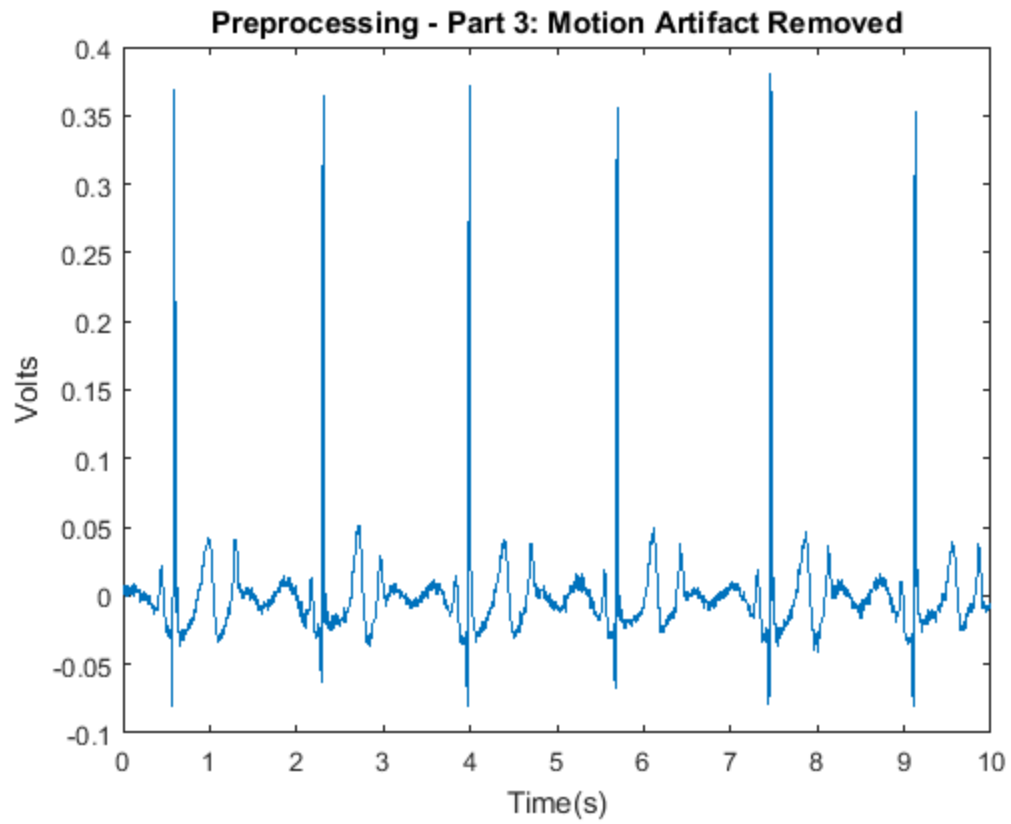
## Part 3 - Motion Artifact Removal

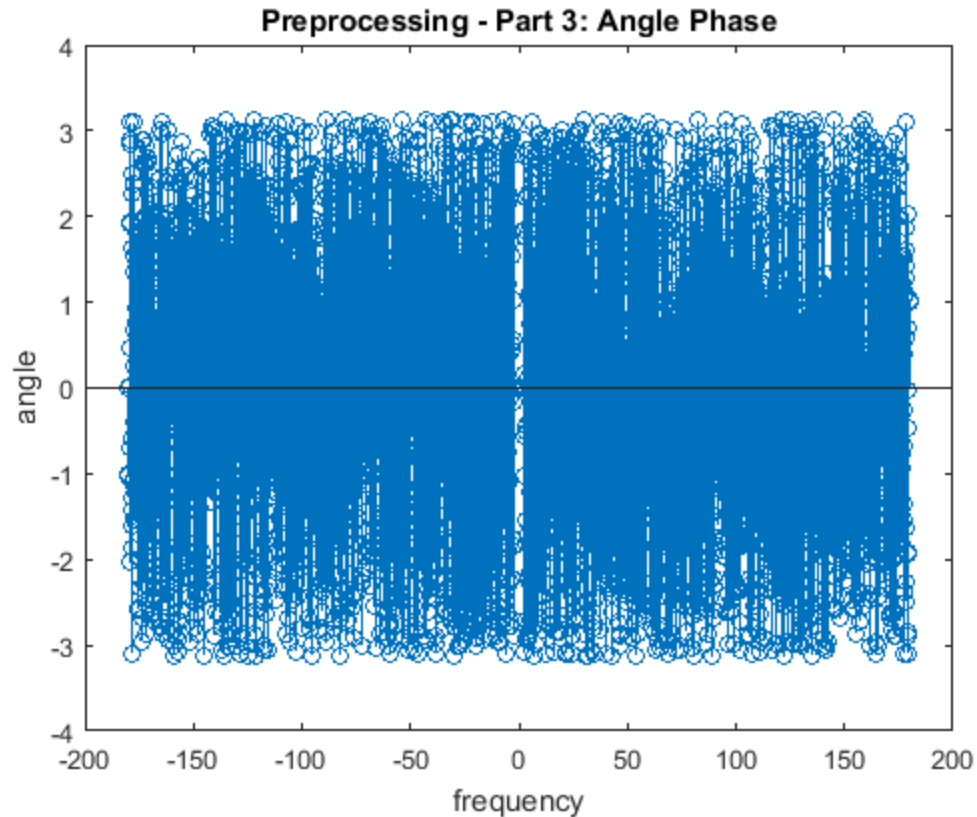
SAME AS BEFORE

```

cof = 2; % cut off frequency
b = cof.*ny./fs; % stop band
hb = floor(b); % high band
lb = ceil(-1.*b); % lower band
mar_f = d_f; % motion artifact removed
mar_f([lb+ny/2+1:hb+ny/2+1])=0; % ideal high pass filter
mar = ifft(ifftshift(mar_f));
figure(5);
plot(t,mar);
ylabel(unit);
xlabel('Time(s)');
title('Preprocessing - Part 3: Motion Artifact Removed');
figure(6);
stem(fd_f, abs(mar_f));
xlabel('frequency');
ylabel('absolute magnitude');
title('Preprocessing - Part 3: Absolute Magnitude');
figure(7);
stem(fd_f, angle(mar_f));
xlabel('frequency');
ylabel('angle');
title('Preprocessing - Part 3: Angle Phase');

```



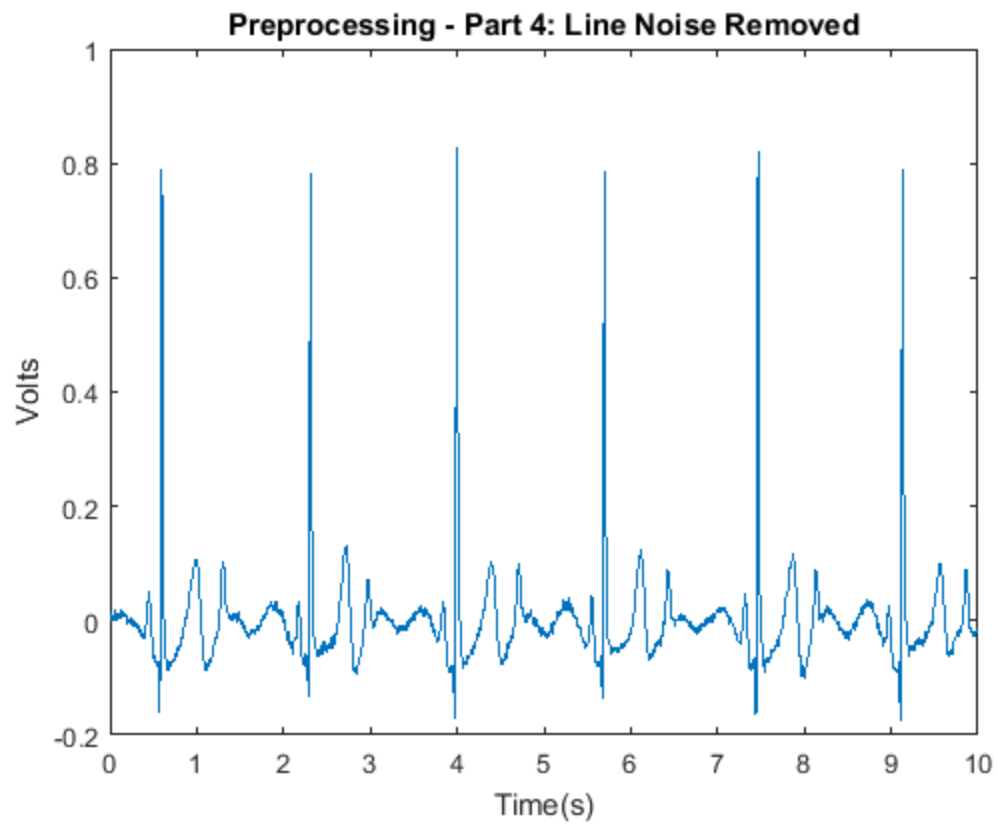


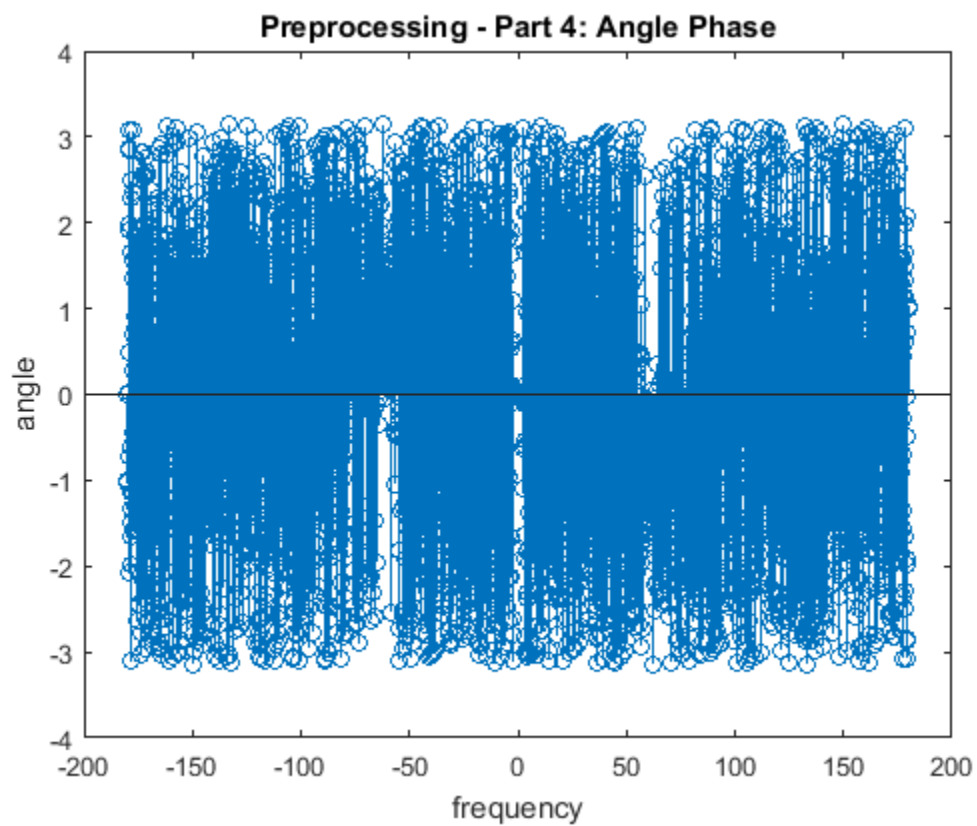
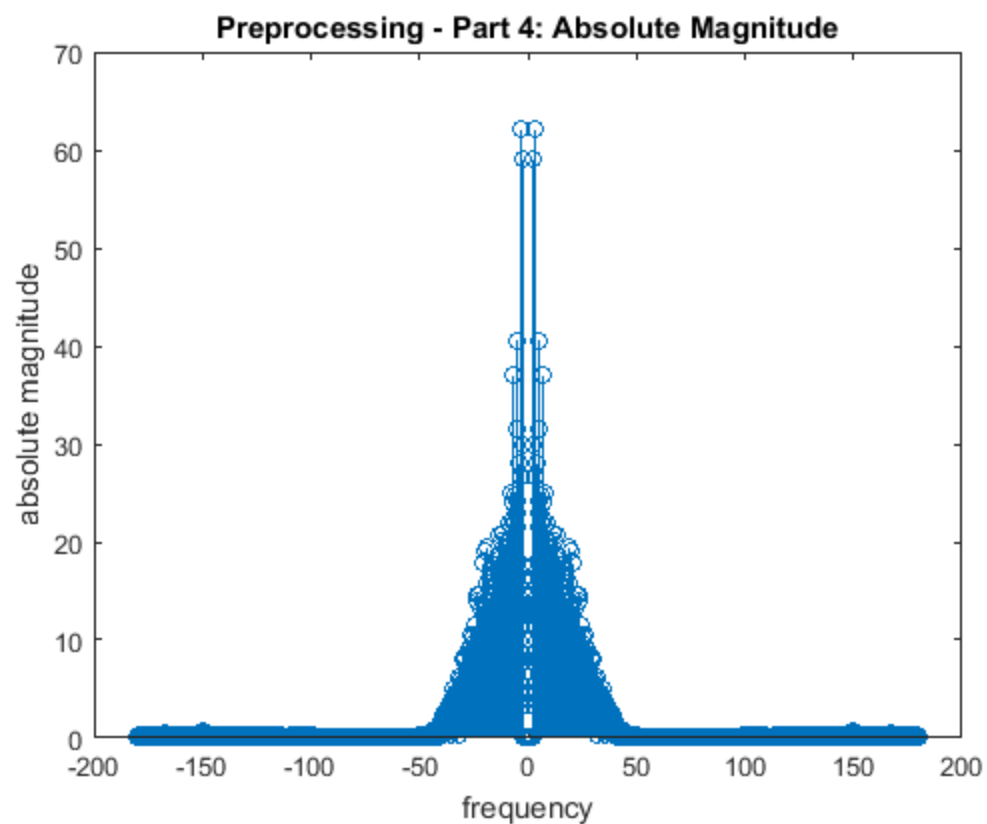
## Part 4 - Line Noise Removal

In this example the notch frequency is 60 Hz.

```
fn = 60; % notch frequency
bw = 120; % bandwidth
apass = 1; % bandwidth attenuation
[b,a]=iirnotch(fn/(fs/2),bw/(fs/2),apass);
hd=dfilt.df2(b,a);
r = mar./max(mar);
lnr=filter(hd,r); % line noise removed
figure(8);
plot(t,lnr);
ylabel(unit);
xlabel('Time(s)');
title('Preprocessing - Part 4: Line Noise Removed');
y2 = fft(lnr);
lnr_f = fftshift(y2);
ny2 = length(y2);
flnr_f = (-ny2/2:ny2/2-1)*(fs/ny2);
figure(9);
stem(flnr_f, abs(lnr_f));
xlabel('frequency');
ylabel('absolute magnitude');
title('Preprocessing - Part 4: Absolute Magnitude');
figure(10);
```

```
stem(flnr_f, angle(lnr_f));  
xlabel('frequency');  
ylabel('angle');  
title('Preprocessing - Part 4: Angle Phase');
```



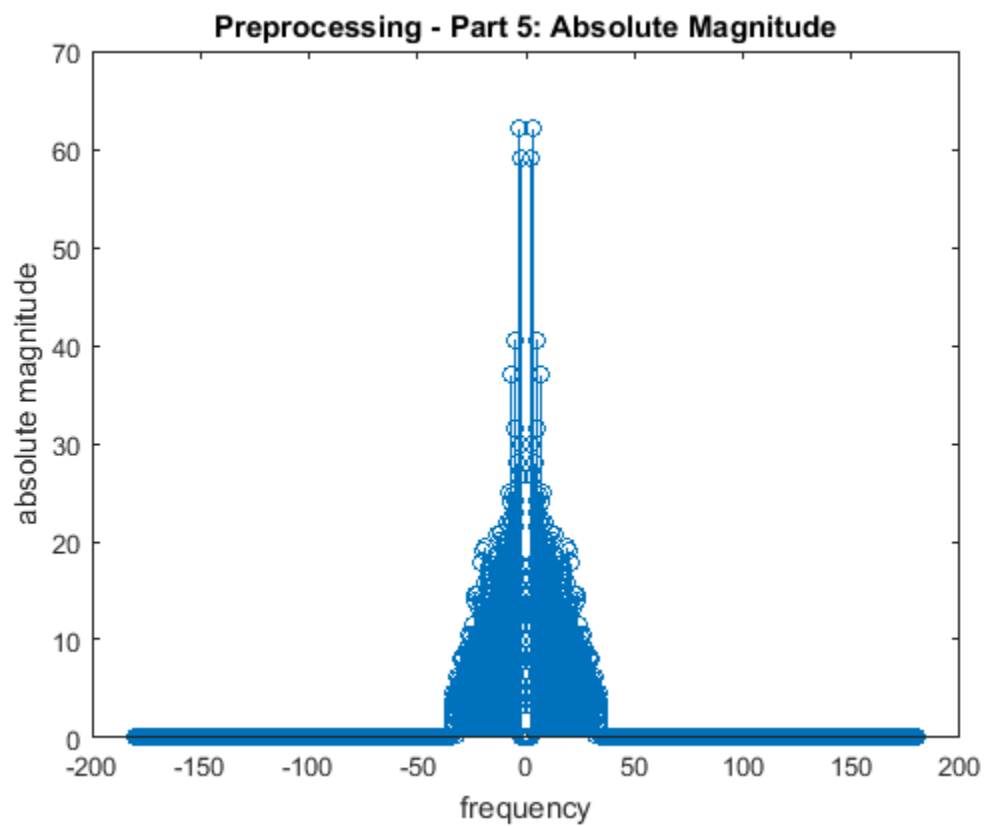
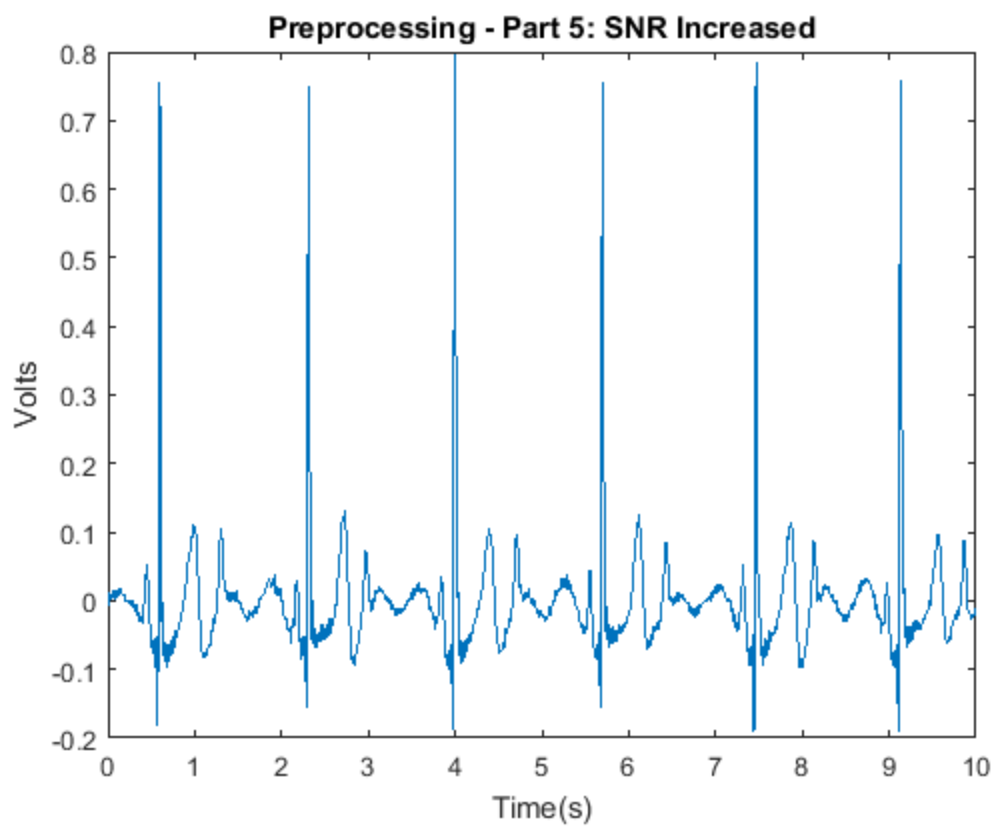


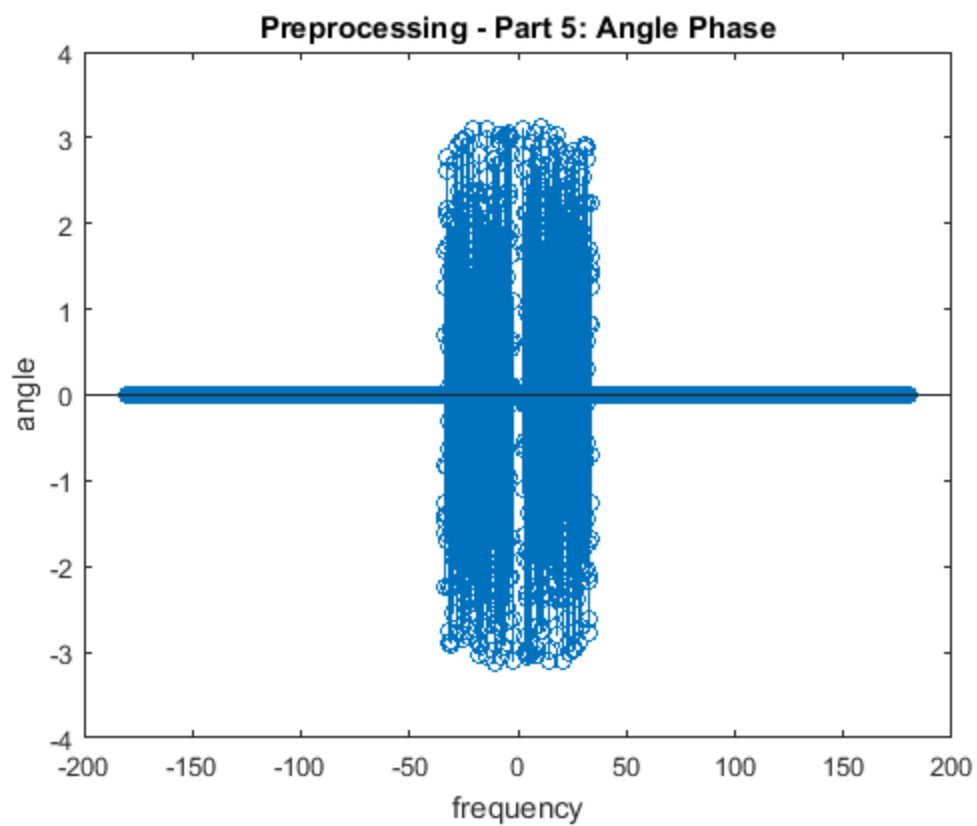


## Part 5 - SNR Increasing

SAME AS BEFORE

```
[wd,lo,hi,power]=obw(lnr,fs);
b = hi.*ny./fs; % stop band
hb = floor(b); % high band
lb = ceil(-1.*b); % lower band
snr_inc_f = zeros(ny,1); % SNR increasing
for i=1:ny
    if (i>=lb+ny/2+1 && i<=hb+ny/2+1)
        snr_inc_f(i)=lnr_f(i); % ideal low pass filter
    end
end
snr_inc = ifft(ifftshift(snr_inc_f));
figure(11);
plot(t,snr_inc);
ylabel(unit);
xlabel('Time(s)');
title('Preprocessing - Part 5: SNR Increased');
figure(12);
stem(fd_f, abs(snr_inc_f));
xlabel('frequency');
ylabel('absolute magnitude');
title('Preprocessing - Part 5: Absolute Magnitude');
figure(13);
stem(fd_f, angle(snr_inc_f));
xlabel('frequency');
ylabel('angle');
title('Preprocessing - Part 5: Angle Phase');
```





*Published with MATLAB® R2015a*