EE354 HW 8 Computer Exercise

Dominique Hinds

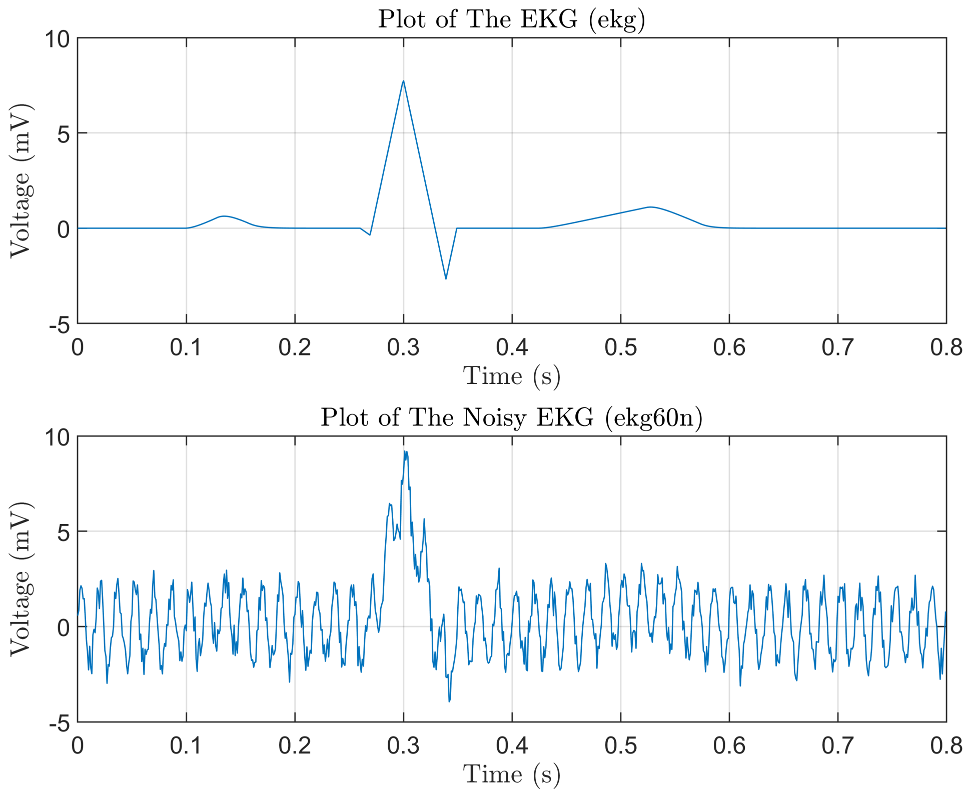
3A.

Plot ekg and ekg60n vs time (seconds).

```
% Define ekg data sets
fs = 1000;
ts = 1/fs;
t = (0 : ts : (0.8-ts));
ekg = [0.0000000e+000  0.0000000e+000  0.0000000e+000  0.0000000e+000  0.0000000e+000  0.000000
ekg60n = [5.6949404e-001  8.4219654e-001  1.9013700e+000  2.1381450e+000  1.9975515e+000  1.448

% Plot ekg data sets
subplot(2,1,1);
figure(1);
plot(t, ekg);
grid on; % xlim([]); % ylim([]);
title('Plot of The EKG (ekg)','Interpreter',"latex");
xlabel('Time (s)','Interpreter',"latex");
ylabel('Voltage (mV)','Interpreter',"latex");

subplot(2,1,2);
plot(t, ekg60n);
grid on; % xlim([]); % ylim([]);
title('Plot of The Noisy EKG (ekg60n)','Interpreter',"latex");
xlabel('Time (s)','Interpreter',"latex");
ylabel('Voltage (mV)','Interpreter',"latex");
```



3B.

Plot the magnitude of the Fourier transforms of ekg and ekg60n as a function of frequency (in Hz) in 0 to 100 Hz range, i.e. plot only the partial results. This will better display the signals of interest. Use a 1024 point FFT. Note that the matlab will automatically append zeros to your data points (to make it 1024 points long) if you gave the command fft(ekg, 1024).

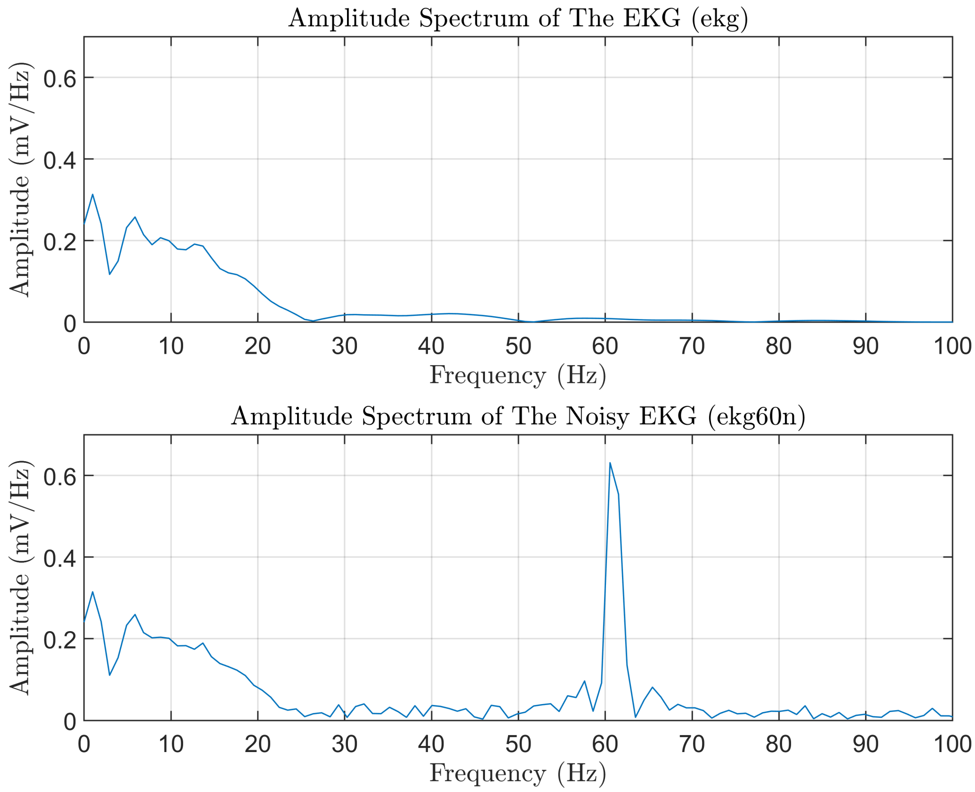
```
% Generate frequency axis
N = 2^10;
df = fs/N;
f = -(fs/2 - df) : df : fs/2;

% Define the amplitude spectrums of both ekg data sets
EKG = ts*fftshift(fft(ekg, N));
EKG60n = ts*fftshift(fft(ekg60n, N));

% Plot amplitude spectrums of both ekg data sets
figure(2);
subplot(2,1,1);
plot(f, abs(EKG));
grid on; xlim([0, 100]); ylim([0, 0.7]);
title('Amplitude Spectrum of The EKG (ekg)','Interpreter',"latex");
xlabel('Frequency (Hz)','Interpreter',"latex");
ylabel('Amplitude (mV/Hz)','Interpreter',"latex");

subplot(2,1,2);
plot(f, abs(EKG60n));
grid on; xlim([0, 100]); ylim([0, 0.7]);
```

```
title('Amplitude Spectrum of The Noisy EKG (ekg60n)','Interpreter',"latex");
xlabel('Frequency (Hz)','Interpreter',"latex");
ylabel('Amplitude (mV/Hz)','Interpreter',"latex");
```



3C.1.

Our objective is to filter the data to remove the large peak at 60 Hz and to minimize the random noise. We are going to use the low pass RC filter of Examples 3.1-1 and 3.1-2 of Carlson; the first example gives the impulse response (equation 8b) and the second the frequency response (equations 18a, b). Note that a single parameter - the product RC or B ($B = 1/2\pi RC$) - determines the filter properties, in this case the frequency of the 3 dB point. We shall design two filters with the following bandwidths:

(i) B = 30 Hz

(ii) B = 50 Hz

Write down the expressions for the impulse responses h30(t) and h50(t) for the filters. Discretize each filter to obtain digital filters represented by h30(n) and h50(n). Choose sampling frequency to be fs = 1000 Hz (sampling period Ts = 0.001, i.e. discretize using $t = 0.001 * n$, n = 0, 1, 2, 3, ... Choose the length N of the discretized filters such that $(N - 1)T_s = 4RC$, i.e. truncate the analog impulse response at four time constants (4RC). Note that the lengths of the two filters will be different.

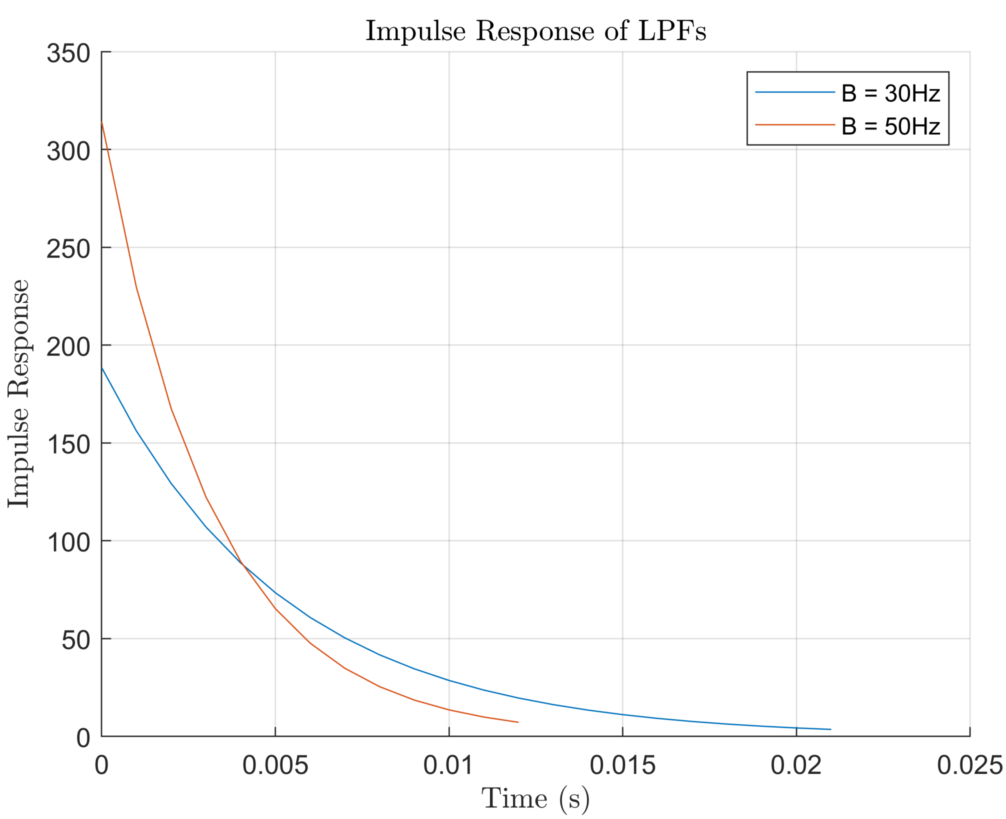
```
% Define the bandwidths and time axes of both LPFs
B30 = 30;
t30 = 0 : ts : 2/(pi*B30);
B50 = 50;
t50 = 0 : ts : 2/(pi*B50);
```

```
% Define the impulse response of both LPFs
h30 = (2*pi*B30)*exp(-(2*pi*B30)*t30);
h50 = (2*pi*B50)*exp(-(2*pi*B50)*t50);
```

3C.2.

Plot the impulse response (discretized version) for each filter as a function of time (seconds).

```
% Plot impulse response of both LPFs
figure(3);
hold on;
plot(t30, h30);
plot(t50, h50);
hold off;
grid on; xlim([0, 0.025]); ylim([0, 350]);
title('Impulse Response of LPFs','Interpreter',"latex");
xlabel('Time (s)','Interpreter',"latex");
ylabel('Impulse Response','Interpreter',"latex");
legend('B = 30Hz','B = 50Hz');
```



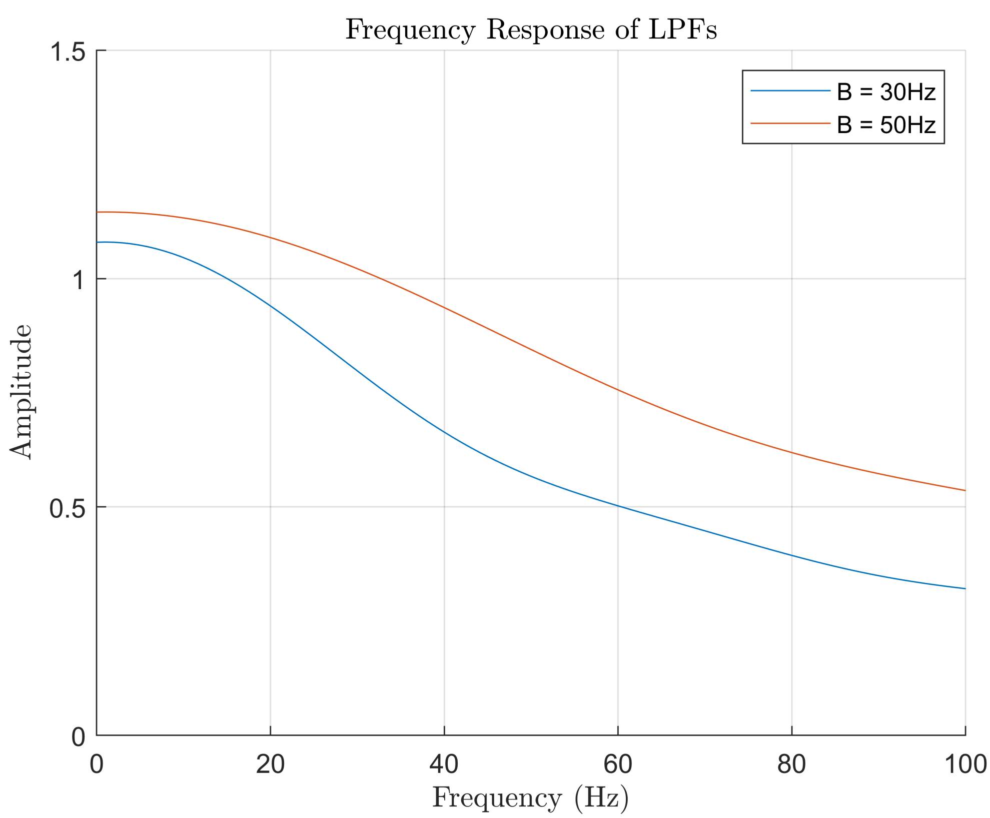
3C.3.

Use a 1024 point FFT to determine the frequency response of each filter. Plot the amplitude |H(f)| and phase for each filter as a function of frequency in the range 0 to 100 Hz. What is the magnitude (in dB) of |H(f = 60Hz)|/|H(f = 0Hz)| for each filter?

```
% Generate frequency axis
N = 2^10;
df = fs/N;
f = -(fs/2 - df) : df : fs/2;

% Define transfer function of both LPFs
H30 = ts*fftshift(fft(h30, N));
H50 = ts*fftshift(fft(h50, N));

% Plot frequency response of both LPFs
figure(4);
hold on;
plot(f, abs(H30));
plot(f, abs(H50));
hold off;
grid on; xlim([0, 100]); ylim([0 , 1.5]);
title('Frequency Response of LPFs','Interpreter',"latex");
xlabel('Frequency (Hz)','Interpreter',"latex");
ylabel('Amplitude','Interpreter',"latex");
legend('B = 30Hz','B = 50Hz');
```



For the LPF with B = 30Hz, |H(f = 60Hz)|/|H(f = 0Hz)| =~ -6.65 dB.

For the LPF with B = 30Hz, |H(f = 60Hz)|/|H(f = 0Hz)| =~ -3.7 dB.

These numbers were calcaluated by plugging the following equations into MATLAB:

20*log(abs(H30(574))/abs(H30(512)))/log(10)

20*log(abs(H50(574))/abs(H50(512)))/log(10)

H30(574) is the value of H30(f) at f = ~60 Hz and H30(512) is the value of H30(f) at f = 0 Hz.

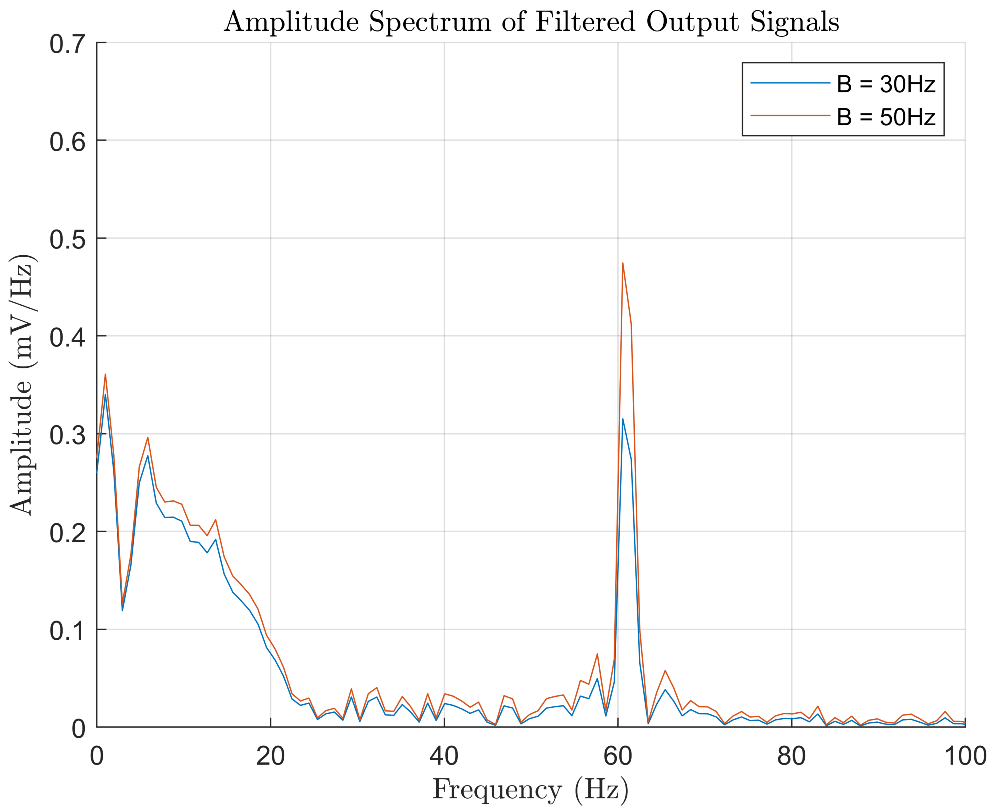
3D.1.

Multiply the Fourier transform of ekg60n found in 3A. to those of the filters found in 3C.2. to obtain 1024 point output Fourier transform Y (f) for each of the filters (Note: Y (f) = H(f)X(f)). Plot Y (f) in each case as a function of frequency (Hz) in 0 to 100 Hz range.

```
% Generate frequency axis
N = 2^10;
df = fs/N;
f = -(fs/2 - df) : df : fs/2;

% Define the spectrum of the filtered outputs
Y30 = EKG60n.*H30;
Y50 = EKG60n.*H50;

% Plot the amplitude spectrums of the filtered outputs
figure(5);
hold on;
plot(f, abs(Y30));
plot(f, abs(Y50));
hold off;
grid on; xlim([0, 100]); ylim([0, 0.7]);
title('Amplitude Spectrum of Filtered Output Signals','Interpreter',"latex");
xlabel('Frequency (Hz)','Interpreter',"latex");
ylabel('Amplitude (mV/Hz)','Interpreter',"latex");
legend('B = 30Hz','B = 50Hz');
```



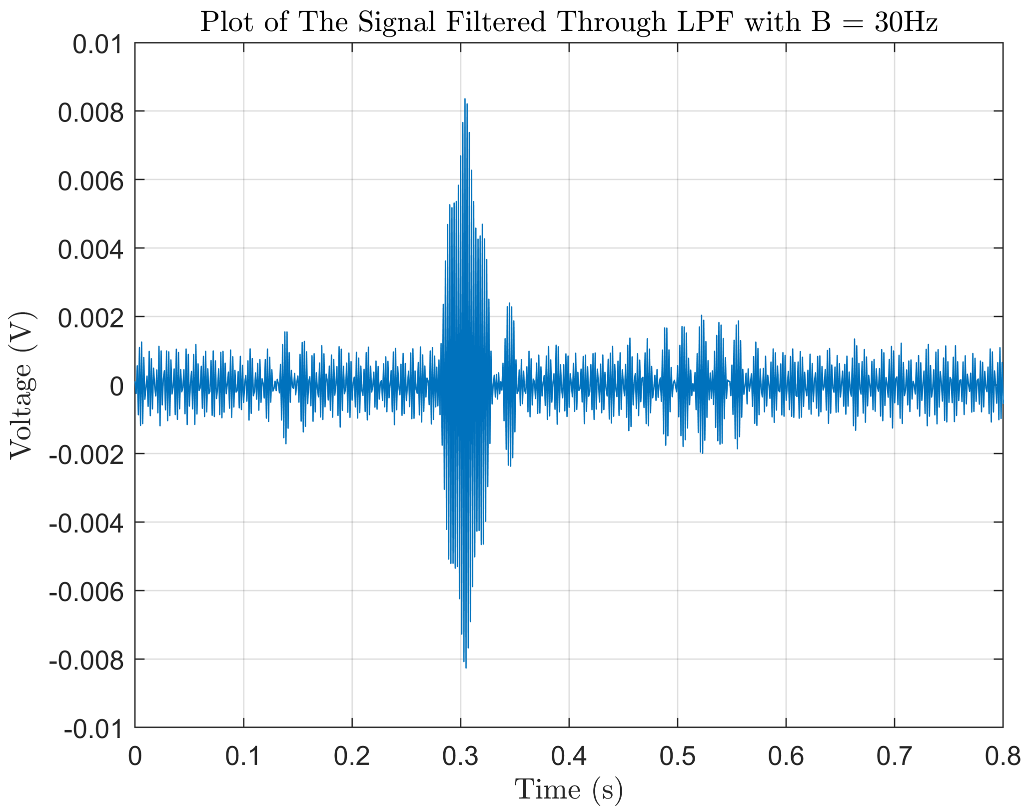
3D.2.

Use a 1024 point inverse FFT (IFFT) to find y(t) in each case. Plot y(t) in each case as a function of time (s). Compare filtered ekg60n, i.e. y(t) in each case with ekg. Which of the two filters works better? Why?

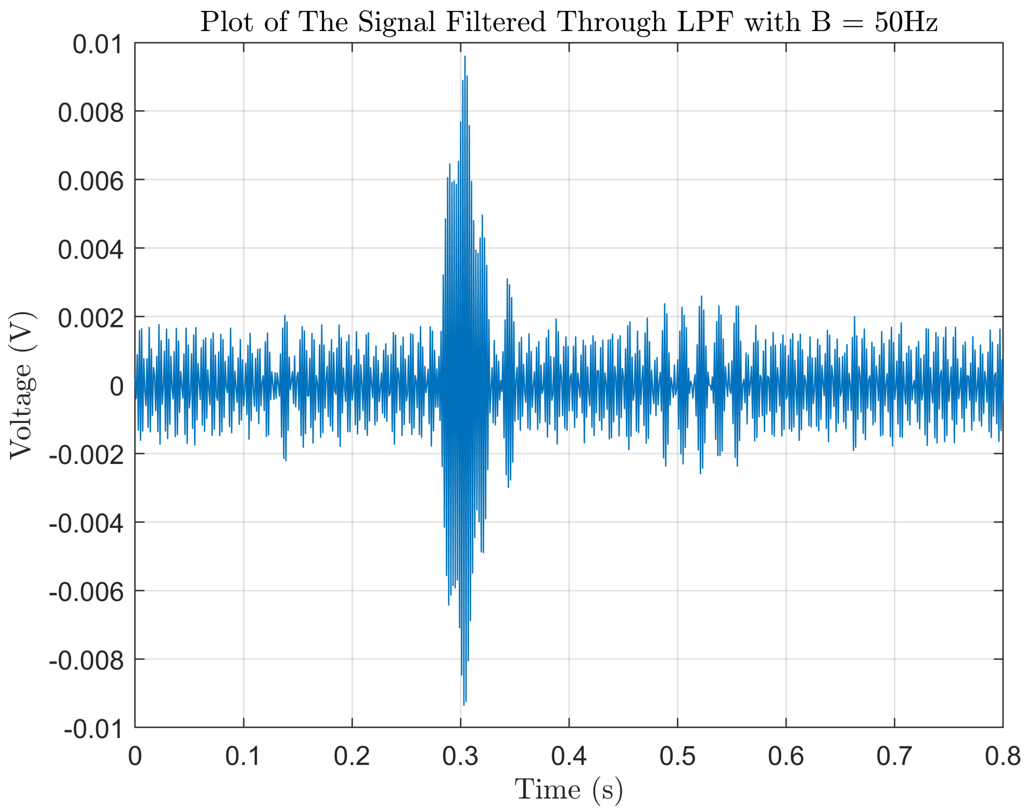
```
% Generate time axis
dt = N/fs;
t = (0 : ts : dt-ts);

% Define filtered outputs
y30 = ifft(Y30);
y50 = ifft(Y50);

% Plot the output filtered through the LPF with B = 30 Hz
figure(6);
plot(t, y30);
grid on; xlim([0, 0.8]); % ylim([]);
title('Plot of The Signal Filtered Through LPF with B = 30Hz','Interpreter',"latex");
xlabel('Time (s)','Interpreter',"latex");
ylabel('Voltage (V)','Interpreter',"latex");
```




```
% Plot the output filtered through the LPF with B = 50 Hz
figure(7);
plot(t, y50);
grid on; xlim([0, 0.8]); % ylim([]);
title('Plot of The Signal Filtered Through LPF with B = 50Hz','Interpreter',"latex");
xlabel('Time (s)','Interpreter',"latex");
ylabel('Voltage (V)','Interpreter',"latex");
```



Looking back at the amplitude spectrum of both of the filtered output signals, the signal that passed through the filter with the 30 Hz bandwidth filter had a smaller peak at 60 Hz compared to the 50 Hz bandwitdth output. Also, the two figures above show that the amplitude of the 30 Hz output is slightly lower than the 50 Hz output. Thus the filter with the 30 Hz bandwidth is better at fulfilling our objective of removing the large peak at 60 Hz and to minimize random noise.