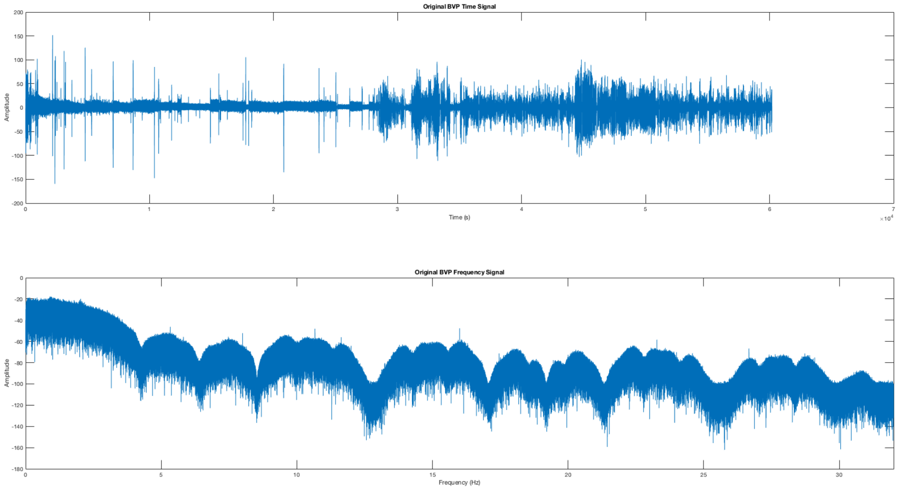
Table of Contents

Basic Importing

```
clc;
filename = 'Example3/BVP.csv';
bvp = csvread(filename);
bvp = bvp(2:length(bvp));
Fs = bvp(1);
bvp = bvp(5:length(bvp));

N = size(bvp);
N = N(1);

t = 0:(1/Fs):(N-1)/Fs;
f = Fs*(0:(N/2))/N;
```

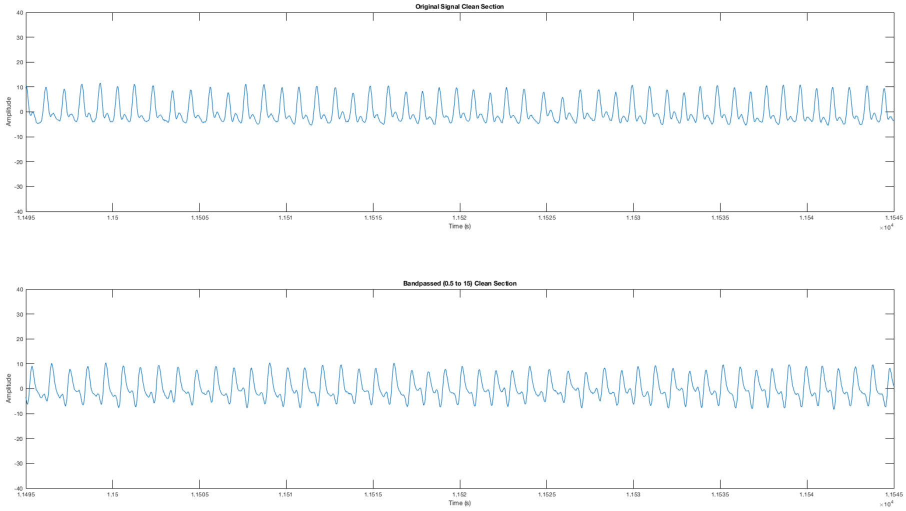
Fourier Transformation

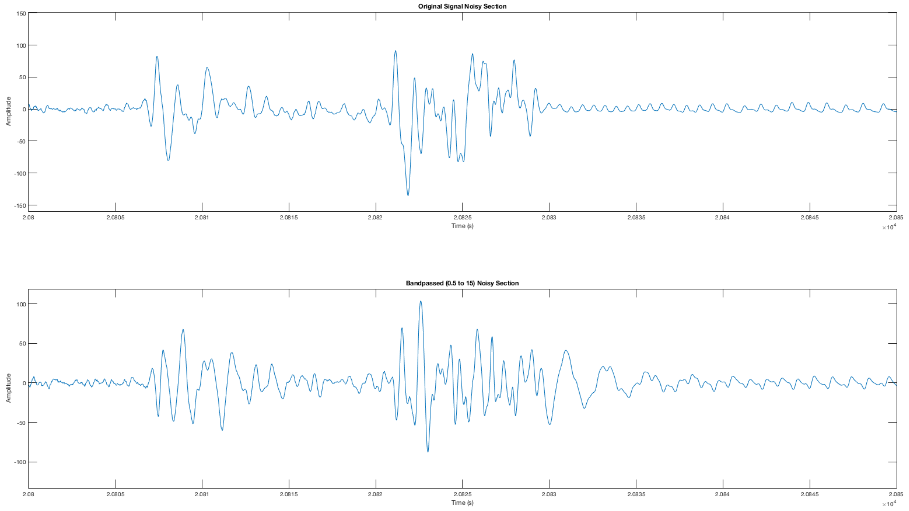
```
bvp_fft = fft(bvp);
original_with_fourier_plot(t, bvp, bvp_fft, f, Fs, N);
```



Bandpass Filtering the BVP %%

```
bandpassed_bvp = filter_signal(bvp);
plot_against_original(t, bvp, bandpassed_bvp, 40, {'Original Signal
 Clean Section','Bandpassed (0.5 to 15) Clean Section'}, 1.1495e4,
 50);
plot_against_original(t, bvp, bandpassed_bvp, inf, {'Original Signal
 Noisy Section','Bandpassed (0.5 to 15) Noisy Section'}, 20800, 50);
```

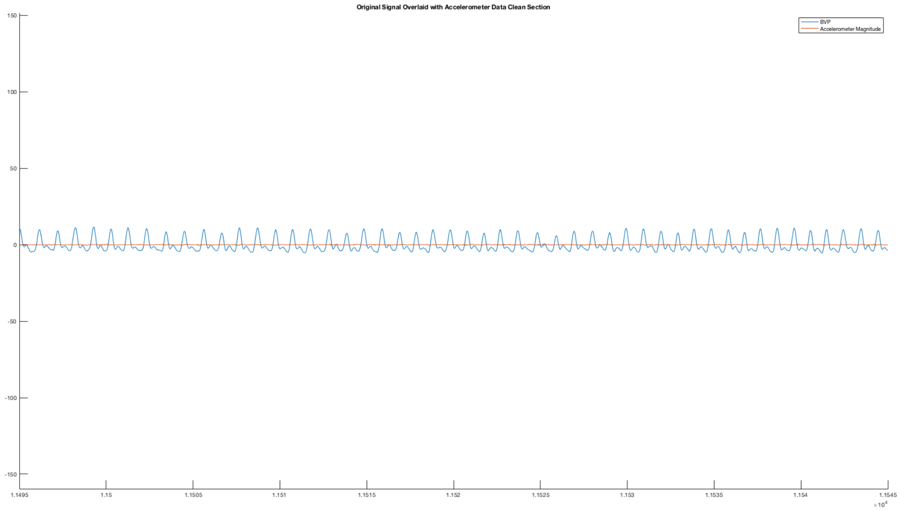


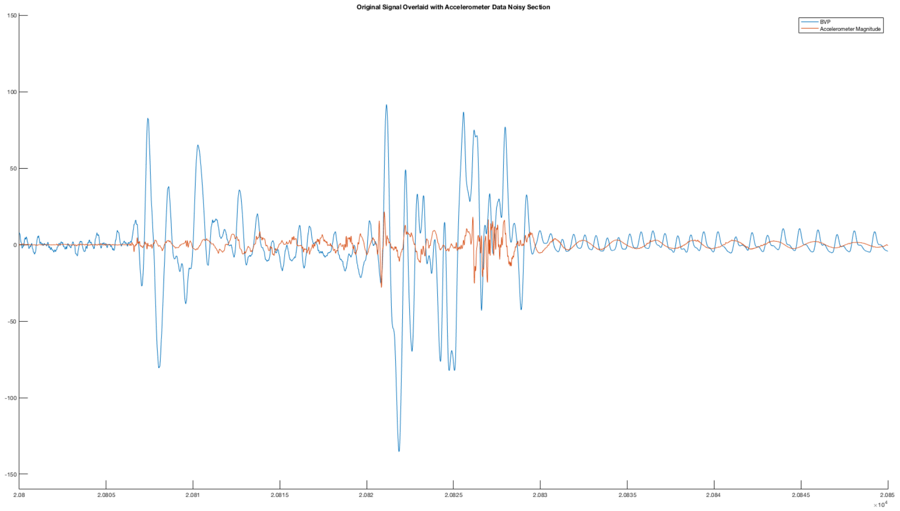


Accelerometer Importing, Fourier Transform, and Plotting

```
[mag_acc, Fs_acc] = get_accelerometer();
```

```
mag_acc = resample(mag_acc, Fs, Fs_acc);
mag_acc = mag_acc - mean(mag_acc);
mag_acc = filter_signal(mag_acc);
plot_on_original(t, bvp, mag_acc, {'BVP', 'Accelerometer
 Magnitude'}, 'Original Signal Overlaid with Accelerometer Data Clean
 Section', 1.1495e4, 50, inf);
plot_on_original(t, bvp, mag_acc, {'BVP', 'Accelerometer
 Magnitude'}, 'Original Signal Overlaid with Accelerometer Data Noisy
 Section', 20800, 50, inf);
```





RLS Filtering Based on Accelerometer Data

```
[rls_filtered, extracted_noise, error] = rls_filter(bandpassed_bvp,
 mag_acc);
```

```
plot_rls_with_original(t, bvp, rls_filtered, extracted_noise, error,
 inf, {'Original Signal Clean Section', 'Noise Extracted from BVP
 Clean Section', 'Error Clean Section', 'RLS Filter Subtracted Signal
 Clean Section'}, 1.1495e4, 50);
plot_rls_with_original(t, bvp, rls_filtered, extracted_noise, error,
 inf, {'Original Signal Noisy Section', 'Noise Extracted from BVP
 Noisy Section', 'Error Noisy Section', 'RLS Filter Subtracted Signal
 Noisy Section'}, 20800, 50);
```

Accelerometer Data Import

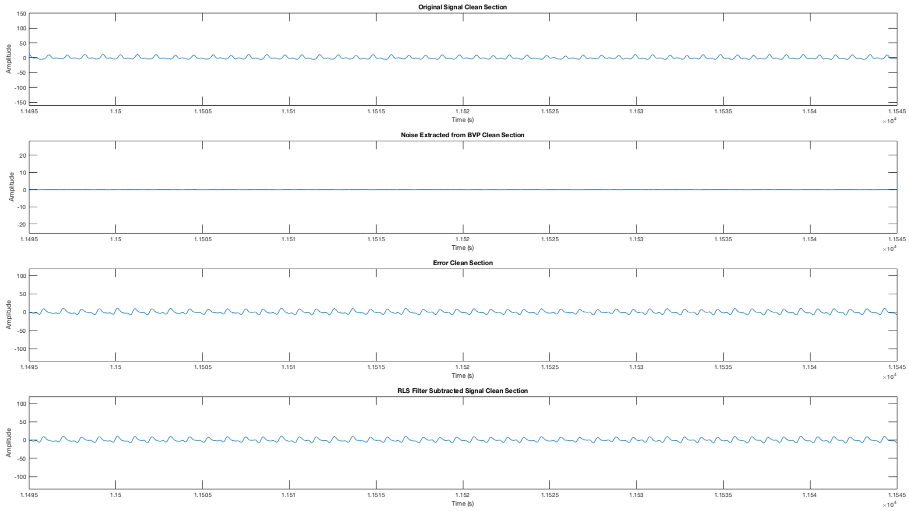
```
function [mag_acc, Fs_acc] = get_accelerometer()
    raw_acc = csvread('Example3/ACC.csv');
    Fs_acc = raw_acc(2,1);
    acc_x = raw_acc(3:end,1);
    acc_y = raw_acc(3:end,2);
    acc_z = raw_acc(3:end,3);

    % Center them about 0 %
    acc_x = acc_x - mean(acc_x);
    acc_y = acc_y - mean(acc_y);
    acc_z = acc_z - mean(acc_z);

    % Bandpass Filter them %
    acc_x = filter_signal(acc_x);
    acc_y = filter_signal(acc_y);
    acc_z = filter_signal(acc_z);
    mag_acc = sqrt((acc_x).^2 + (acc_y).^2 + (acc_z).^2);
end
```

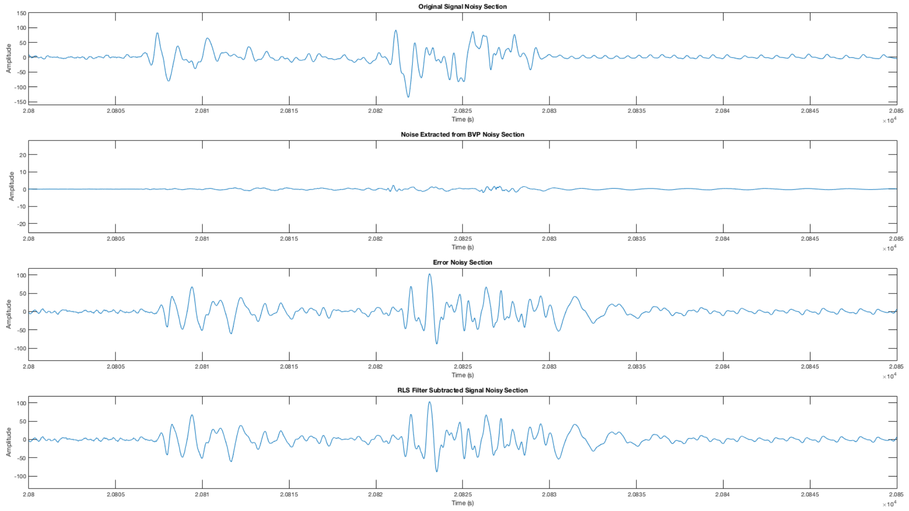
Filtering Functions

```
function [rls_filtered, extracted_noise, error] = rls_filter(signal,
 noise_source)
    filterLength = 32;

    signalLength = length(signal);
    delay = zeros(filterLength,1);
    signal = [delay;signal(1:signalLength-filterLength)];

    rlsFilt = dsp.RLSFilter(filterLength);
    [extracted_noise, error] = rlsFilt(noise_source, signal);
    rls_filtered = signal - extracted_noise;
end

function bandpassed = filter_signal(bvp)
    bandpassed = bandpass_filter(bvp);
end

function bandpassed = bandpass_filter(bvp)
    Fs = 64;  % Sampling Frequency

    N   = 30;    % Order
    Fc1 = 0.4;  % First Cutoff Frequency
    Fc2 = 30;   % Second Cutoff Frequency

    % Construct an FDESIGN object and call its BUTTER method.
    h  = fdesign.bandpass('N,F3dB1,F3dB2', N, Fc1, Fc2, Fs);
    Hd = design(h, 'butter');

    bandpassed = filter(Hd, bvp);
end
```

Plotting Functions

```
function plot_rls_with_original(t, bvp, rls_filtered, extracted_noise,
 error, max_amp, titles, start, window_length)
    figure();
    subplot(4,1,1);
    plot(t, bvp);
    xlabel('Time (s)');
    ylabel('Amplitude');
    title(titles{1});
    axis([start start+window_length -max_amp max_amp]);

    subplot(4,1,2);
    plot(t, extracted_noise);
    xlabel('Time (s)');
    ylabel('Amplitude');
    title(titles{2});
    axis([start start+window_length -max_amp max_amp]);

    subplot(4,1,3);
    plot(t, rls_filtered);
    xlabel('Time (s)');
    ylabel('Amplitude');
    title(titles{3});
    axis([start start+window_length -max_amp max_amp]);

    subplot(4,1,4);
    plot(t, error);
    xlabel('Time (s)');
    ylabel('Amplitude');
    title(titles{4});
    axis([start start+window_length -max_amp max_amp]);

    x0=10;
    y0=50;
    width=2000;
    height=1000;
    set(gcf,'units','points','position',[x0,y0,width,height])
end

function original_with_fourier_plot(t, bvp, bvp_fft, f, Fs, N)
    P2 = abs(bvp_fft/N);
    P1 = P2(1:N/2+1);
    P1(2:end-1) = 2*P1(2:end-1);

    subplot(2,1,1);
    plot(t,bvp);
    title('Original BVP Time Signal');
    xlabel('Time (s)');
    ylabel('Amplitude');

    subplot(2,1,2);
    plot(f,20*log10(P1));
    title('Original BVP Frequency Signal');
    xlabel('Frequency (Hz)');
    ylabel('Amplitude');
    xlim([0 (Fs/2)]);

    x0=10;
    y0=50;
    width=2000;
    height=1000;
    set(gcf,'units','points','position',[x0,y0,width,height])
end

function plot_against_original(t, original, new, max_amp, titles,
 start, window_length)
    figure();
    subplot(2,1,1);
    plot(t, original);
    xlabel('Time (s)');
    ylabel('Amplitude');
    title(titles{1});
    axis([start start+window_length -max_amp max_amp]);

    subplot(2,1,2);
    plot(t, new);
    xlabel('Time (s)');
    ylabel('Amplitude');
    title(titles{2});
    axis([start start+window_length -max_amp max_amp]);

    x0=10;
    y0=50;
    width=2000;
    height=1000;
    set(gcf,'units','points','position',[x0,y0,width,height])
end

function plot_on_original(t, original, new, legend_values, name,
 start, window_length, max_amp)
    figure()
    hold on;
    plot(t,original);
    plot(t,new);
    title(name);
    legend(legend_values{1}, legend_values{2})
    axis([start start+window_length -max_amp max_amp]);

    x0=10;
    y0=50;
    width=2000;
    height=1000;
    set(gcf,'units','points','position',[x0,y0,width,height])
    hold off;
end
```





Published with MATLAB® R2018b