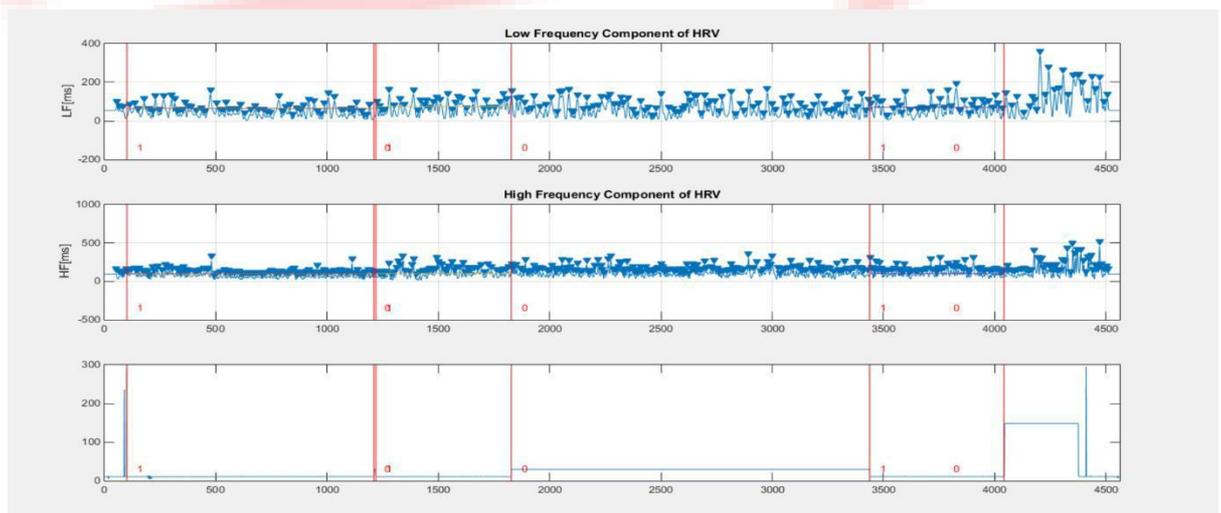


## Automatic Detection Algorithm of events (signals) in HRV Data using MATLAB

There are two frequency which are Low Frequency(LF) and High Frequency(HF) that been used in this project. The LFHF.m and HRV data are attached together as a reference.

- 1) By using acceleration data (XAcceleration, YAcceleration,ZAcceleration) , determine the 'resting' time to detect the events in LF and HF signals. The 'resting' time should be equal to resting time where there is no physical activity occurred at that time. The acceleration must be divided into 2 regions which are the 'active' time and the 'resting' time. This region must be identified by using filtering (e.g: low/high pass filter) to make the detection signal looks more accurately. Use FFT first to find the frequency for the filtering process.
- 2) After the 'active' and 'resting' time is identified, draw a subplot of 3 signals (LF, HF, Acceleration done with filtering process).
- 3) During length of 'resting' time regions, put a horizontal line as a threshold value (the threshold line can obtain from the value of mean along one period of 'resting' time) for every LF and HF. Note: the threshold line must be draw for every 'resting' time condition of the acceleration signal. The value of threshold line is depending on the number of peaks of the signal within one 'resting' time signal. It may different from each other depending on the signal.
- 4) For HF and LF signal, calculate the width between one peak to other peak that are above the threshold value. Note: Put an indication marks of every width of each of the signal.
- 5) The value of width, mean, standard deviation and area of the peaks for Low Frequency(LF) and high frequency(HF) in 'resting' time signal in acceleration should be appear in notepad. Note: Please put a numbering for 'resting' time signal for all the data in notepad. For example, it may contain more than one 'resting' time in the signal. For the first 'resting' time signal, it is necessary to put "Rest Time Signal 1" and so on.



Example of the signal

Script function for calling HFLF.m file:

```
[LF, HF, LFHF, filterorder, f_low, f_high, W_low, W_high]  
= Get_LFHF (HRVDatams) ;
```

Function Get\_LFHF:

```
function [LF, HF, LFHF, filterorder, f_low, f_high, W_low, W_high]
= get_LFHF(HRVDatams) %#codegen

filterorder = 60;
Fs =1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
y = HRVDatams;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%resample
t= 1:length(HRVDatams)
t(1) = HRVDatams(1)/1000;
for i = 2:length(HRVDatams);

t(i) = t(i-1) + HRVDatams(i);

end

t = t/1000;

[t,yt] = resample(HRVDatams,1,Fs,1,'spline');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

t2 = 1:length(y);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% define center Frequencies and shift data
f_low = 0.095;
f_coherence = 0.1;
f_high = 0.275;

s_low = exp(-1j*2*pi*f_low*t2);
s_coherence = exp(-1j*2*pi*f_coherence*t2);
s_high = exp(-1j*2*pi*f_high*t2);

x_shifted_low = y' .* s_low;
x_shifted_coherence = y' .* s_coherence;
x_shifted_high = y' .*s_high;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% define low Pass Frequencies and filter
data W_low = 0.055/0.5/Fs;
b_low = fir1(filterorder,W_low,chebwin(filterorder+1,30));

W_coherence = 0.01/0.5/Fs;
b_coherence = fir1(filterorder,W_coherence,chebwin(filterorder+1,30));

W_high = 0.125/0.5/Fs;
b_high = fir1(filterorder,W_high);

a = 1;

x_filtered_low = filtfilt(b_low,a,x_shifted_low);
x_filtered_coherence = filtfilt(b_coherence,a,x_shifted_coherence);
x_filtered_high = filtfilt(b_high,a,x_shifted_high);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% calculate absolute values and delete Start and End Values
x_abs_low =2 * sqrt(real(x_filtered_low).^2 + imag(x_filtered_low).^2);
%x_abs_low = x_abs_low(100:end-100);
```

```

x_abs_coherence = 2 * sqrt(real(x_filtered_coherence).^2 +
imag(x_filtered_coherence).^2);

x_abs_high = 2* sqrt(real(x_filtered_high).^2 + imag(x_filtered_high).^2);
%x_abs_high = x_abs_high(100:end-100);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i = 1:50
x_abs_high(i) = mean(x_abs_high(100:end-100));
x_abs_high(end-i:end) = mean(x_abs_high(100:end-100));
x_abs_low(i) = mean(x_abs_low(100:end-100));
x_abs_low(end-i:end) = mean(x_abs_low(100:end-100));
x_abs_coherence(i) = mean(x_abs_coherence(100:end-100));
x_abs_coherence(end-i:end) = mean(x_abs_coherence(100:end-100));

end

LF = x_abs_low;
HF = x_abs_high;
COH = x_abs_coherence;

LF = smooth(LF);
HF = smooth(HF);
%
% figure;
% plot(t2,LF,t2,HF);
% LF = sqrt(sum(LF(50:end-50).^2));
% HF = sqrt(sum(HF(50:end-50).^2));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Calculate LFHF
%
%
LFHF = LF/HF;
%
% LFHF_mean = mean(LFHF(50:end-50));
% LFHF_std = std(LFHF(50:end-50));
% LFHF_norm = LFHF ./ (max(LFHF)-min(LFHF));

end

```