Chapter 1
PRV estimation function

This document explains the PRV_est Matlab function that I put together to estimate instantaneous PRV features. The function is called as follows:

[u_k, du_k, AR_Model, PRV_features] = PRV_est(PPG, sf, varargin)

The inputs are:

- **PPG:** the column vector containing the PPG signal.
- **sf:** sampling frequency of the PPG signal.

The options in varargin are:

- **decimation_ratio:** decimation ratio used while searching the peaks using the first method (derivative method). *The default value is decimation_ratio=10.*
- **maxhr:** maximum heart rate that can be realistically accepted by the algorithm. This is used to clean the series of peak locations to avoid getting unreal hr estimates. *The default value is maxhr=180.*
- **shift:** Once the initial estimate of the peaks are located in a filtered version of the original signal, the algorithm uses them as a reference to search for the peaks in the original signal and avoid delays introduced by filtering. Therefore, the algorithm shifts

the location of the initial peaks by a percentage of the minimum peak interval [0 – 1], and proceeds to search new peaks in the intervals defined by the new location vector. *The default value is shift=1/3.*

- **bw_ppg:** The second method filters the PPG signal before using the Hilbert transform. The filter has a bandwidth bw_ppg. *The default value is bw_ppg=[0.5; 5] Hz.*
- **pkfinder_mode:** Sets the method used to find the initial peaks. Mode 1 uses the Hilbert transform to find local minima. Mode 2 uses the second derivative to find the maximum acceleration in the PPG signal. *The default value is pkfinder_mode=1.*
- **ensemble_filter:** whether or not the algorithm uses an ensemble filter to correct the identified peaks that do not correspond to an actual PPG wave. *The default value is ensemble_filter='yes'.*
- **wl_ens:** length of the window used to obtain the static correlation in the ensemble filter. The length is given as a ratio of the median interval. *The default value is wl_ens=0.25.*
- **th_corr:** threshold of the correlation coefficient that defines a PPG wave as not correlated with the mean wave. *The default value is th_corr=0.45;*
- **median_filter:** whether or not the algorithm uses a median filter to correct the identified peaks that define intervals too far from the median. *The default value is median_filter='yes'.*
- **th_med:** threshold of the factor that defines if an interval is too far from the median. *The default value is th_corr=5;*
- **missing_beat_correction:** whether or not the algorithm fills in the blanks left by the peaks that are missing. *The default value is missing_beat_correction='yes'.*

- **ar_p:** order of the autoregressive model. *The default value is ar_p=5.*
- **ar_w:** window length in seconds for the AR model. *The default value is ar_w=90.*
- **ar_weight:** exponential weight of the previous values on the distribution of PP intervals. *The default value is ar_weight=0.98.*

The outputs are:

- **u_k:** The location of the peaks in the original PPG signal.
- **du_k:** The interval length between the peaks.
- **AR_Model:** Structure with the parameters of the AR model.
- **AR_Model.Theta:** Parameters of the AR model for lags [1 – ar_p].
- **AR_Model.Theta0:** Static parameter of the AR model lag=0.
- **AR_Model.Kappa:** Shape parameter of the Inverse Gaussian distribution.
- **AR_Model.LogLikel:** Log likelihood value obtained with the model.
- **AR_Model.L:** Likelihood of the model.
- **PRV_features:** Instantaneous features of the PPG signal.
- **PRV_features.PP:** Instantaneous mean of the IG distribution.
- **PRV_features.Var:** Instantaneous variance of the IG distribution.
- **PRV_features.Mean_PP:** Weighted mean of the past and current PP values (instantaneous mean).
- **PRV_features.LF:** Instantaneous Low Frequency Power of the PP sequence.
- **PRV_features.HF:** Instantaneous High Frequency Power of the PP sequence.
- **PRV_features.Coherence:** Instantaneous coherence feature derived from the PP sequence spectrum.

- **PRV_features.LFHF_ratio:** Instantaneous Low Frequency to High Frequency Power ratio.

1.1 System requirements

To use this function it is necessary to have Matlab. Also, the Nonlinear Identification Toolbox (nlid_tools) should be added to the path. The toolbox can be found in: https://github.com/reklab/reklab_public. Finally, make sure that Matlab has a C compiler set up.

1.2 Description of the algorithm

The algorithm consists, so far, in three main stages – i.e. peak search, PP sequence correction, and AR modelling and feature extraction.

1.2.1 Peak search

To obtain an estimate of PRV from PPG signals, it is necessary to estimate correctly the location of the start of every cardiac cycle. For this particular algorithm, I decided to use the diastolic minima, shown in Figure 1-1, of every cycle (the lowest point in the cycle) because it showed to be the easiest point to recognize. So far, I have worked with data from three sources:

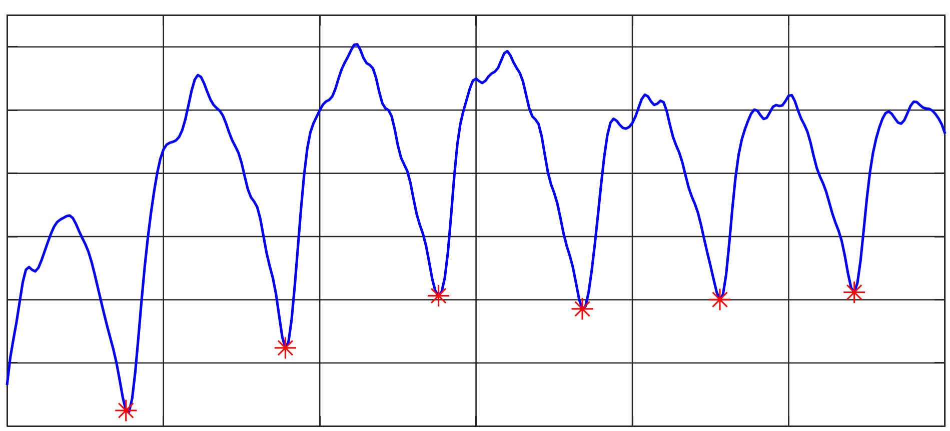

Figure 1-1: Location of diastolic minima in PPG signal

Thought Technology's TPS sensor, the Empatica wristband, and Ironova's Ankkoro wristband. To deal with data from these sources, I have developed two peak search methods: the first one is based on the derivatives of the signal, and the second one uses the Hilbert transform to search for the local minima of each cycle. The first method has proven fruitful with every source of data, while the second one works better only with Empatica data.

Method 1: Derivatives analysis

In this method, the PPG signal is first derived and decimated following a chose decimation ratio. I would recommend that the decimated sampling frequency remains over 20 Hz. Thus, neither Ankkoro or Empatica data should be decimated with a larger ratio than 2. TPS data can be decimated with a ratio of up to 10. Figure 1-2.A shows the first derivative (V) of the PPG signal. The signal V shows different peaks, which sometimes can be hard to discriminate from the peaks that define each cycle. As we are only interested in the positive peaks of V, the algorithm rectifies

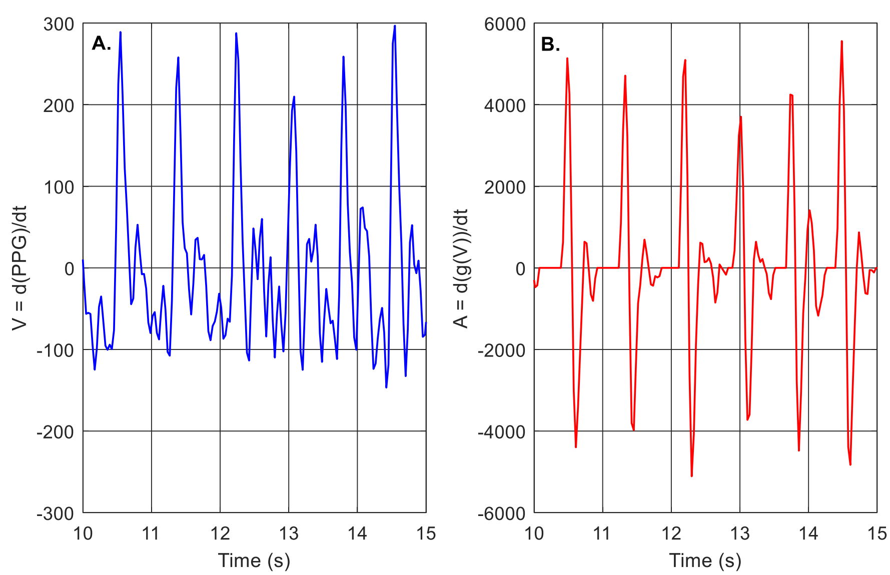

Figure 1-2: (A) Velocity of the PPG signal. (B) Derivative of the half-wave rectified velocity of the PPG signal.

the signal and keeps only the positive values. In the V signal, the point of zero-crossing just before the larger peak correspond to the minimum point in the PPG signal, as the signal stop decreasing to start increasing with a large velocity. This zero-crossing point is now a corner in the rectified signal, preceding the largest peak of each cycle. Thus, the peaks in the derivative of the rectified signal (signal A) will correspond to the approximate position of the diastolic minima in each PPG cycle. Figure 1-2.B shows this new signal where the maximum peaks are very clear and cannot be confused as in the first derivative of PPG. This method is particularly advantageous in signals with large drifts or non-stationarities that are not easily removed with bandpass filters – i.e. TPS and Ankkoro data.

The local maxima of signal A is easily found using Matlab's function 'findpeaks'. This function can be constrained to allow better recognition of the correct peaks. Thus, the algorithm sets that the minimum distance between peaks corresponds to the maximum heart rate allowed (maxhr), and the minimum size of the peak prominences has to be equal to the std of the total

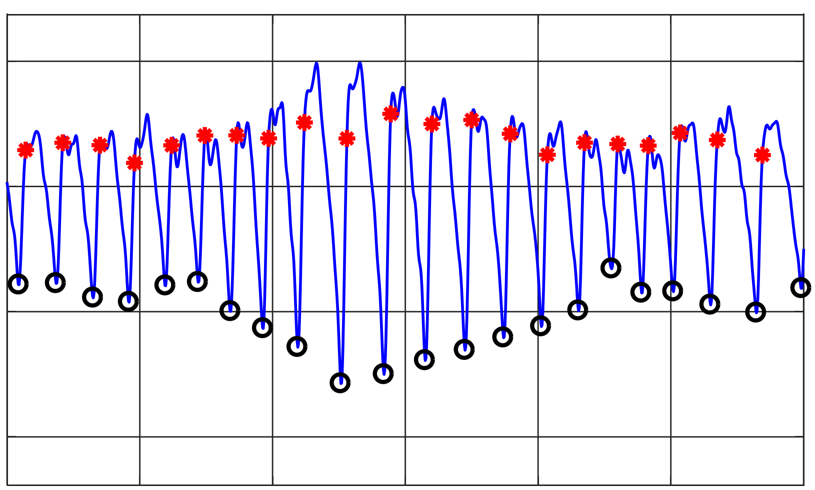

Figure 1-3: Reference position of each cycle (red stars) and the position of each diastolic minima (black circles).

signal A. Once the peaks of signal A are found, they can be used as reference to search for the real peaks in the PPG signal. The location of the peaks of signal A could be used to mark the beginning of each cardiac cycle, but the decimation and derivation processes could have introduced delays that we would prefer to not have. Therefore, the algorithm uses the locations of these peaks, shifts them by a predefine ratio and searches for peaks in sections of the PPG signal delimited by each pair of peak locations of the signal A. Figure 1-3 illustrates this principle, where the red start mark the shifted location of the peaks from signal A, as a reference of the location of each cycle, and the black circles show the identified diastolic minima from each segment delimited by each pair of red stars. The function pkfinder1 returns the location of all the black circles as the output.

Method 2: Hilbert transform analysis

In this method we use the properties of the Hilbert transform to find the local minima of a time series. This method works better with Empatica data, as their preprocessing algorithm produces a signal without large non-stationarities. I have tried to use this method with TPS or Ankkoro data but the no-stationarities make it impossible to obtain a good discrimination of local minima even after filtering.

First, the algorithm filters the signal with a bandpass filter to reduce high frequency noise that could be produced by motion artifacts. Then the phase of the Hilbert transform is estimated with Matlab's 'hilbert' and 'angle' functions. The instantaneous phase, obtained with the Hilbert transform, slips every time it encounters a local minima. Figure 1-4.A shows a plot of the phase as a function of time, where we can see how the phase jumps from $+\pi$ to $-\pi$ regularly. These jumps represent the encounter of local minima. To amplify this feature, the algorithm takes the

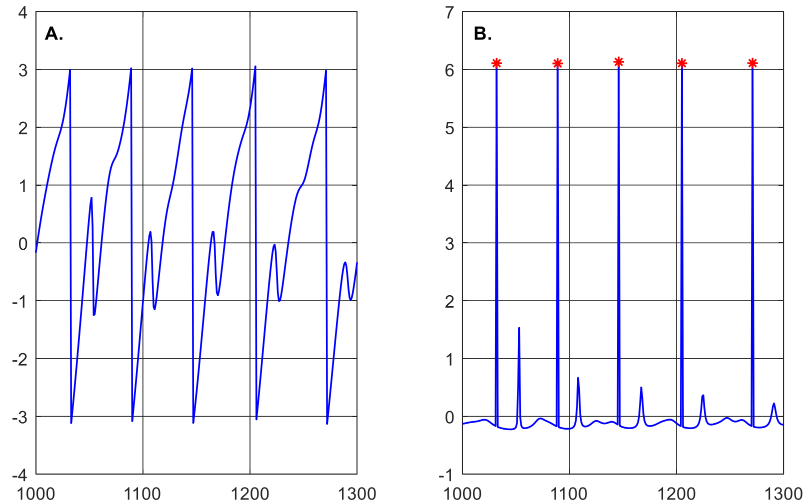

Figure 1-4: (A) Instantaneous phase provided by the Hilbert transform. (B) Peak search in the first difference of the phase.

negative of first difference of the phase, such that every jump will produce peaks approximately 2π in amplitude. Figure 1-4.B shows the location of these peaks. Finally, the algorithm uses these peaks as reference to look for the peaks in the original signal, just like it does in the derivative method.

1.2.2 Correction of the PP sequence

The described algorithm provides a first estimate of the location of the diastolic minima for each cardiac cycle in the PPG signal. However, some peaks might have been missed or been assigned erroneously. For this reason, the algorithm uses three methods of sequence correction: ensemble filter, non-destructive median filter and missing beats detection.

Ensemble filter:

The idea behind this filter is simple, and the flow of the algorithm can be understood from the comments in the code. Basically, the algorithm searches for the average wave around all the identified peaks. Then, the algorithm compares each wave with the average wave in terms of the static cross-correlation coefficient. If the coefficient is lower than the threshold, the peak is deemed erroneous and a vector is returned with all the marked erroneous beats.

Median filter:

This filter is also simple. Instead of looking into the location of the beats, the algorithm looks at the difference of these locations, the inter-beat interval lengths. If the absolute difference of the IBI lengths with its median is too large compared to the median of this difference, then the respective interval is considered to be an outlier in length, and it is marked erroneous. Finally all the erroneous beats obtained with the ensemble and median filters are removed.

Missed beat detection:

This algorithm also looks at the IBI lengths, and for each interval it compares if they are too long with respect to their neighbors. If that is the case, it is assume that one or more beats were missed inside that interval and the algorithm proceeds to fill in the blanks. To do so, the algorithm draws new IBI lengths from a rectangular distribution in the range defined by the mean +/- the std of the last 10 beats. Then, this IBI lengths and converted into new beat positions and added to the final vector of beats positions.

1.2.3 Autoregressive model

The third part of the algorithm consists in modelling the PP sequence using an autoregressive model and use it to extract relevant features. This is done using the method proposed by Barbieri et al. (2005). So far, the algorithm returns only temporal and spectral features as defined in the first section of this document.

1.3 Using the algorithm

Refer to the script PRV_tst.m to learn how to use the algorithm. As an example, load the data from the file 'PPG_emp.m'. This data corresponds to an experiment during for the first 5 minutes, the subject did a cardiac coherence exercise to increase its HRV. The next 5 minutes, the subject underwent a mental arithmetic test to increase their level of stress. Finally, the subject finished with 5 minutes of the cardiac coherence exercise again.

The outputs of the algorithm carry the features we desired to obtain from the PPG signal. Figure 1-5 shows the location of diastolic minima in the PPG signal as defined by the algorithm output 'u_k'. The output 'du_k' corresponds to the IBI length, and it is used to identify the AR system that underlies the PP interval sequence.

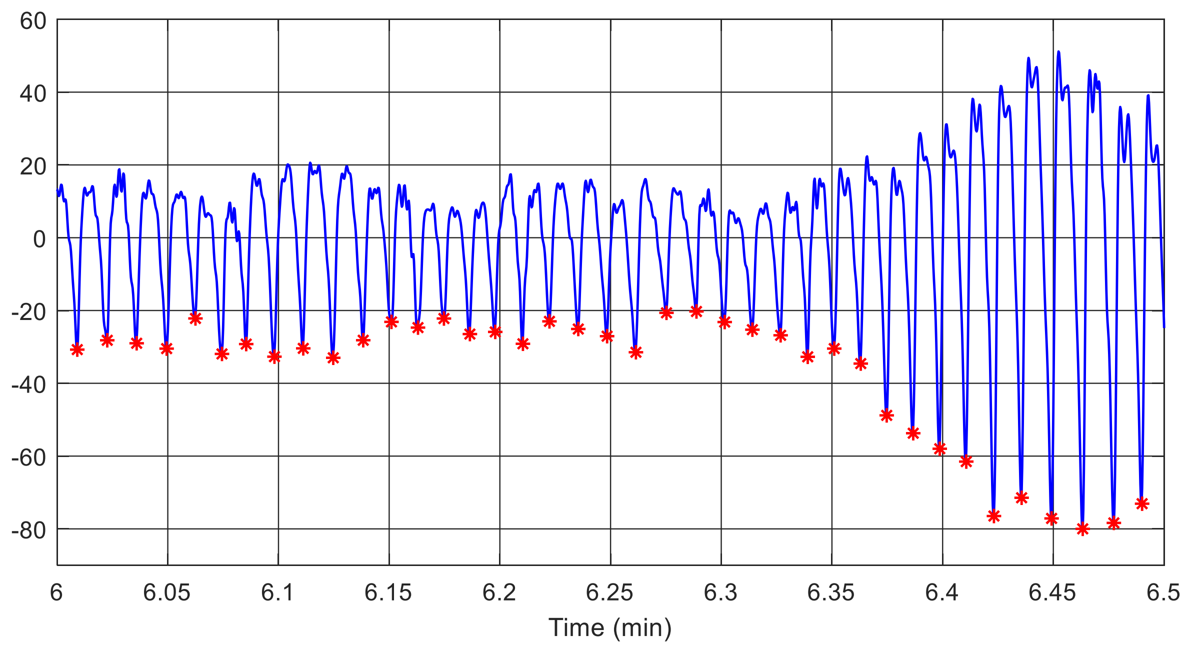

Figure 1-5: Location of diastolic minima identified by algorithm.

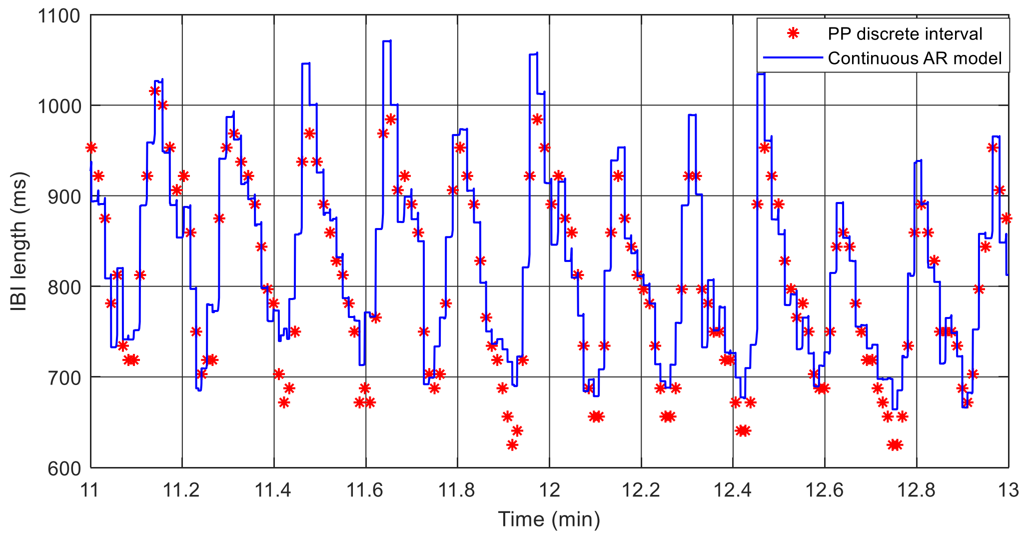

Figure 1-6: IBI interval lengths as defined by the observed diastolic minima sequence (red stars) and the AR model (blue line).

Figure 1-6 shows the sequence of IBI lengths positioned in time, and compares it with the continuous length obtained from the AR model. The figure shows that the AR model captures well the time-varying behavior of the PP intervals.

Finally, it is also possible to obtain instantaneous spectral features from the parameters estimated with the AR model. Here we show 4 features: LF power, HF power, LF/HF ratio and cardiac coherence. LF power refers to the integral of the power spectrum of the PP sequence in the band [0.04 – 0.15] Hz. HF power is the power in the band [0.15 – 0.45] Hz. The LF/HF ratio is the LF power divided by the HF power. Cardiac coherence searches for a LF peak (in the band [0.04 – 0.26] Hz and divides the power of this peak +/- 0.015 Hz and the remaining power of the spectrum. The time course of these features are shown in Figure 7. The use of these features seems promising, as they show big differences during different emotional states.

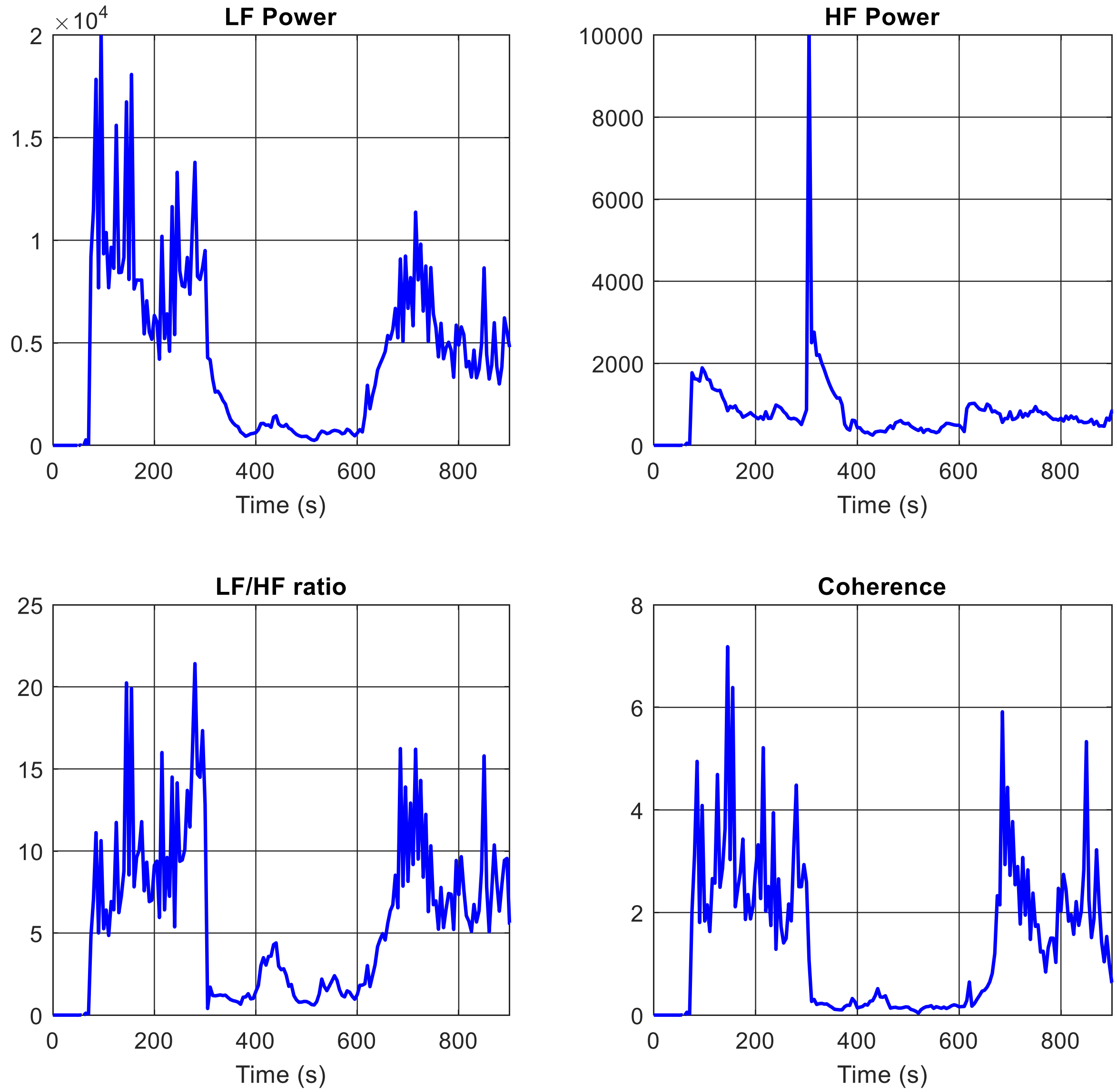

Figure 1-7: Plots of the features obtained with the PRV_est algorithm: LF power, HF power, LF/HF ratio and cardiac coherence.