Spectral analysis of Heart Rate Variability

The tools that you learn in the stationary stochastic processes course are fundamental and can be applied for extraction of information in many areas. In the following description, the contents of your data-files will be explained. For the two scheduled computer sessions, a number of investigations and questions are proposed. These are chosen to give you ideas for your further analysis. Some usual mistakes are also highlighted. It is recommended that some of the exercises are made before the computer session, e.g., easier tasks and implementation of m-files for computations and visualizations. Important to remember is that the project is in all parts open for your own ideas!

We will later examine these and more advanced methods of time series modeling in much further details in the course on time series analysis and other courses in mathematical statistics.

Project outline and data description

Spectral analysis of Heart Rate Variability (HRV) has become increasingly common for non-invasive assessment of autonomic cardiac regulation. The HRV signal is extracted from the ElectroCardioGraphic (ECG) signal from the heart and is normally sampled with 1000 Hz. The so-called RR-intervals, which are the distances between two following heart beats, are detected and each interval length is represented as a level corresponding to the interval length. The created HRV-signal is then a staircase signal which is downsampled to 4 Hz, generating a signal to be analyzed for the variation of the heart rate.

Of particular interest is the power of the high frequency component (HF-HRV) that is negatively related to respiration, i.e., respiratory sinus arrhythmia (RSA), which mirrors parasympathetic or vagal regulation of the heart. This means, in the ideal case, that the estimated power of the HF-HRV should have a correlation coefficient close to −1 corresponding to the estimated respiratory frequency. In view of the substantial amount of research relating e.g., psychosocial stressors in work life to cardiovascular disease, it is important to reliably estimate the power of the HF-HRV.

The aim of this project is to investigate different spectral techniques to estimate the power of the HF-HRV component as well as the respiratory frequency. The traditional measure of the HF-HRV power is the average power in the frequency band 0.12-0.4 Hz. A novel strategy is to use the respiratory frequency as center frequency and define a smaller frequency band around the center frequency for the average power, i.e., using a band defined by \( f_0(t) \pm \delta f_0 \), where \( f_0(t) \) might be varying with time and \( \delta f_0 = 0.1 \) Hz.
To be able to make an estimate of the correlation coefficient between the HRV power and the respiratory frequency the data samples in a set have to be divided into several short sequences for the analysis. These sequences could overlap. As example, if you choose the length 128 of a sequence, and e.g., time-shift of 32 samples (75% overlap), the different short sequences are related to a time-scale, and will be represented by samples 1-128, 33-160, 65-192, 97-224, 129-256,... Estimate the HRV spectrum and the respiratory spectrum for each pair of sequences. Find the maximum peak of the respiratory spectrum as the respiratory frequency \( f_0(t) \) at time \( t \) and compute the power of the HRV spectrum, \( P_{HRV}(t) \), at time \( t \). The power of the HRV spectrum can be computed using different spectrum estimation techniques, e.g., periodogram and AR-modeling. The average power should also be computed using the traditional frequency band 0.12-0.4 Hz as well as in the new band defined by \( f_0(t) \pm 0.1 \) Hz.

From the analysis you will have two sequences of estimated values. Compute the correlation coefficient estimate as

\[
\gamma = \frac{\sum_t (f_0(t) - m_{f_0})(P_{HRV}(t) - m_{P_{HRV}})}{\sqrt{\sum_t (f_0(t) - m_{f_0})^2 \sum_t (P_{HRV}(t) - m_{P_{HRV}})^2}},
\]

(1)

where \( m_{f_0} \) and \( m_{P_{HRV}} \) are the mean values of the sequences \( f_0(t) \) and \( P_{HRV}(t) \) respectively.

Some strategies and questions for the project are:

- The data sequence lengths for estimation of spectra could be longer or shorter, e.g., 64, 128 or 256. What is the advantages in each case? Compare a few different sequence lengths for the spectrum estimates and compare the resulting correlation coefficient estimates.
- A larger overlap of the sequences gives a larger number of observations for the correlation coefficient estimate. Does this improve the result?
- Compare the traditional method and the novel strategy for estimating the power of the HRV. Which of these strategies give the best result?
- Compare using the periodogram or an AR-model for the estimation of the HRV spectrum and the respiratory spectrum for each pair of sequences. Which method gives negative correlation closest to -1?

The data material is found in the file \texttt{hrvdata.mat}. All data are sampled with the sampling frequency 4 Hz. The segments consist of the following:

- A controlled experiment was made for the data in the variable HRVjump and corresponding respiratory data in RESPjump. The experiment was performed during 6 minutes where the participant was asked to breath with the frequency of a signal generator with changing frequency. The signal generator started with \( f = 0.5 \) Hz for 2 minutes, changed to \( f = 0.25 \) Hz for 2 minutes and finally to \( f = 0.125 \) Hz for 2 minutes. Figure 1 shows the resulting step-wise stationary data from the recording with the HRV-signal in the upper figure and the breathing data in the lower figure. This data can be divided in three different stationary sequences.
• A set of five different measurements of resting HRV-data and connected to each of these sequences are the measurement of respiratory data, as columns in the matrices HRV_{rest} and RESP_{rest} respectively. The sequences might be seen as stationary. Unfortunately, one of the measurements of RESP_{rest} has a large sinusoidal disturbance, which possibly is from a 50 Hz disturbance of the original measurement. Due to a mistake no antialiasing filter was used and the 50 Hz signal arises as a 2 Hz disturbance of data.

• Another experiment where the respiration frequency slowly increases, giving a slowly decrease of the power of the HF-HRV component. This data set is not stationary but might be modeled as such on a short-time basis.

![HRV measurement with corresponding respiratory data for step-wise changing respiratory frequency.](image)

**Computer session 1**

The purpose of these instructions and questions is to help you study the data in your project using the tools for stationary stochastic processes that you learn in chapter 1-4 of the compendium. **Especially read chapter 2.2.1, 2.2.3, 2.5.1, 2.5.3, 3.3.2, 3.4, 4 and preferably also chapter 8.3 before you begin.**

Load your data file into MATLAB, `load name.mat`, and use the command `whos` to see all included variables in the file. Go through the following question dictionary and examine your data sets. Make sure you try to find an answer to the question and that you really investigate your data properly.

1. How is the sample frequency variable `fs` related to the data? What is the actual distance between the samples? Plot your data with a time scale related to reality.
2. Can the data be represented by a stationary stochastic process? Check this by dividing one column of, e.g., HRVrest in shorter sequences, e.g., samples 1-200, 201-400, 401-600, ... of each sequence. Check if the mean value is around the same constant value of all short sequences? If you can assume this, the data has constant mean value. Plot and discuss the other data sequences as well.

3. To be defined as a stationary process, the covariance function has to depend only on the distance $t - s$. Implement an estimator of the covariance function in Theorem 2.5. Save your implementation as a function of an m-file. One example might look like this:

```matlab
function [cov,tau]=covfunc(data,taumax);
% Estimate of the covariance function up to taumax of a vector of data.
% cov: covariance function
% tau: values of tau
n=length(data);
m=mean(data);
for i=0:taumax
    cov(i+1)=((data(1:n-i)'-m)*(data(i+1:n)-m))/n;
end
tau=[0:taumax]';
```

Use the function to check the stationarity of your sequences by computing and comparing the different covariance functions of your short sequences, e.g., for the first column of 'HRVrest':

```matlab
[cov1,tau]=covfunc(HRVrest(1:200,1),100);
covfunc(HRVrest(201:400,1),100);
covfunc(HRVrest(401:600,1),100);
```

4. Visualize the dependence (correlation) of the different total sequences of data in a scatter plot

```matlab
>> plot(data(1:end-k),data(k+1:end),'.')
```

and find the values of $k$ for which the dependence in each sequence is strong (both negative and positive). Does the strong positive dependence match high positive covariance and a strong negative dependence a high negative covariance? Check that the result of your covariance function matches the scatter plots above. You can, e.g., plot the correlation as

```matlab
>> plot(tau,cov/cov(1))
```
where the covariance values are normalized with the variance giving the correlation function.

5. In Chapter 4, a number of different processes are described. Could any of these be applicable as models of your data set? Why or why not?

6. The stationary Gaussian process is one of the most utilized distribution models. Examine if the data is a stationary Gaussian process. Use e.g., the command normplot, if available (statistics toolbox) or the command hist. Your data will probably not be exactly a Gaussian process but can in most cases be approximated as such a process.

7. The respiratory frequency should be found from the spectrum estimate. Estimate the spectrum of e.g., 'RESPvar', using the periodogram, defined as

\[ R_x^\ast(f) = \frac{1}{n}|X(f)|^2, \]  

(2)

where

\[ X(f) = \sum_{t=0}^{n-1} x(t)e^{-i2\pi ft}, \]  

(3)

is the Fourier transform of the data vector. The Discrete Fourier Transform (DFT) can be computed in MATLAB, using the command fft

\[ \text{>> X=fft(data);} \]

The spectrum estimate is computed as

\[ \text{>> S=(abs(X).^2)/n;} \]

or

\[ \text{>> S=(X.*conj(X))/n;} \]

where n is the data length. Create a frequency vector of the same length as your spectral estimate,

\[ \text{>> ny=[0:n-1]/n;} \]

Plot your spectrum estimate

\[ \text{>> plot(ny,S)} \]

What does it show? How are the frequencies related to the true frequency scale? The true frequency scale is found as

\[ \text{>> f=ny*fs;} \]

Plot the spectrum estimate using this frequency scale,
8. An even better view is given using zero-padding, i.e.,

\[
>> x = \text{fft}(\text{data},N);
\]

where \( N \) is a number around 4-10 times as large as the length of \( \text{data} \) chosen as \( N = 2^I \) where \( I \) is an integer. (To learn more about this, read chapter 8.3). Compute and plot the resulting spectral estimate,

\[
>> S = (\text{abs}(X)^2)/n;
\]

\[
>> f = [0:N-1]*fs/N;
\]

\[
>> \text{plot}(f,S)
\]

Note that the normalizing factor of the spectral estimate should be the data length \( n \) and not the FFT-length \( N \). How does the use of zero-padding change your spectral estimate?

The spectrum estimate in Matlab is given between zero and one instead of \(-0.5 \) and \( 0.5 \) which is the usual definition. The upper part is the mirrored spectral estimate of the lower part so it is natural to just present the interval representing half the sampling frequency,

\[
>> \text{plot}(f(1:N/2),S(1:N/2))
\]

Does the strong frequencies of the spectrum estimate match visible periodicity in data? What is the frequency estimate of 'RESPvar'? Does it match the visible frequency in data?

9. Estimate the spectrum from e.g., one of the columns of 'RESPrest' instead. How does the spectrum change? Why? If you check your data you will see that the measurements in 'RESPrest' are not located around zero mean, which gives a high peak at zero frequency. For a better view and easier analysis use zero mean data in your further analysis!

**Computer session 2**

The purpose of these instructions and questions is to help you study the data in your project using some of the tools for stationary stochastic processes that you learn in Chapter 5-8. Especially read chapter 5.2, 6.1, and 6.2 before you begin.

1. Remember that one of your recorded sequences were disturbed by a strong sinusoidal signal. If you have not found it by visual inspection, it is \( \text{RESPrest}(\cdot,5) \). Your first question should be, do I have to sort these data sequences out from my analysis or can I do something to save the important information and suppress the disturbances? Can you use some kind of linear filter to clean your data? Apply low-pass filters as well as high-pass filters and explain the resulting output data. The following commands might be useful: \( \text{fir1} \) and \( \text{filter} \). Make e.g., a low-pass filter with cut-off frequency \( f = 0.4 \cdot fs/2 \) using,
Note that the normalized frequency in Matlab here is given between 0 and 1 where 1 is relating to \( \frac{fs}{2} \). Usually, in literature, the normalized frequency is given between 0 and 0.5 where 0.5 is relating to \( \frac{fs}{2} \). What is the cut-off frequency of your real world data set with the given sampling frequency? Is this sufficient for your purpose?

Plot the resulting output and explain the difference in phase between the output data. Read e.g., example 5.1 in the compendium. Compare with the original data. Is the phase-shift important for your further analysis? If so you can try to use the command \texttt{filtfilt} which corrects the phase by filtering forward and backward.

2. What is happening if you choose a higher or a lower frequency of the filter? Also check the amplitude function of the filter to make sure that it looks like you expect,

\[
\begin{align*}
\texttt{>> [H,F]=freqz(h,1,N);} \\
\texttt{>> plot(F,abs(H).^2);} \\
\end{align*}
\]

or

\[
\begin{align*}
\texttt{>> [H,F]=freqz(h,1,N,Fs);} \\
\texttt{>> plot(F,abs(H).^2);} \\
\end{align*}
\]

to relate to your data sampling frequency. \textit{Before you go on to further analysis, make sure that the data you use is zero mean level corrected as well as free from sinusoidal disturbances!}

3. Autoregressive models (AR) are the most usual parametric models and is used for spectral estimation, parameter estimation, prediction and many other purposes. Implement the algorithm for estimation of AR-parameters in Chapter 6.2, (an example of the AR(2)-process is also given in Computer exercise 2 of the Stationary Stochastic Process course). Implement a m-file with a data vector and a specified model order as input and the AR-parameters and the estimated variance of the input white noise as output. One suggestion follows here but preferably you implement your own algorithm:

\[
\begin{align*}
\texttt{function [A,fpe]=ar(x,p);} \\
\texttt{[n,m]=size(x);} \\
\texttt{if n==1} \\
\texttt{\quad x=x';} \\
\texttt{\quad n=m;} \\
\texttt{\end{if}} \\
\texttt{X=x(p+1:n);} \\
\end{align*}
\]
U=[];
for i=1:p
    U(:,i)=-x(p+1-i:n-i);
end

Apar=inv(U'*U)*U'*X;

Q=(X-U*Apar)'*(X-U*Apar);
A=[1 Apar'];
fpe=Q/(n-p);

Make sure you understand the estimation procedure and verify your algorithm using simulated AR-processes! Simulation of AR-processes are made using e.g.,

\[ x = \text{filter}(1,A,e); \]

where \( e=\text{randn}(n,1) \) is a white Gaussian noise sequence. Note that the estimated parameters as well as the estimated noise variance vary for different realizations of noise.

4. Use some of your data sequences and try to find appropriate model orders for these. If you increase the order of your model, you will find that the estimated variance \( fpe \) will decrease. To be able to choose an appropriate order, a compensation for the over-parameterizing is needed. One of the most known criterion for model order estimation is the Akaike information criterion (AIC),

\[
\text{AIC}(p) = \ln e(p) + \frac{2p}{n},
\]

where \( n \) is the data sequence length, \( e(p) \) is the variance error, \( fpe \), from the AR-modeling for order \( p \). The model order \( p \) that gives the minimum value of \( \text{AIC}(p) \) is chosen. You can implement an AIC-function to estimate appropriate model orders of your data sequences.