

Computer exercise 2 in Stationary stochastic processes, HT 19.

The purpose of this exercise is to study different techniques for spectral density estimation and some examples using the cross-spectrum and the coherence estimate.

1 Preparations

- Read through the entire computer exercise.
- Study chapters 6.1-6.2 and 9 in the course book and familiarize yourself with the MATLAB functions, `fft`, `periodogram`, `pwelch`, `cpsd`, and `mscohere`.
- **Answer the exercises for Computer exercise 2 in Mozquizto at latest Monday evening September 30 at 23.00.**

Mozquizto is found at <http://quizms.maths.lth.se/> where you login with your Stil-login. **You need to pass the test in order to attend the computer exercise!** Prepare well, you might be asked to discuss and explain your answers during the exercise.

2 The periodogram and zero-padding

Please find the additional files and data on the webpage and load these into your computer. Receiving a file of (possibly unknown) data, the first step before more advanced spectral analysis always should be to make a check of the spectral content and possible strong disturbances, e.g., 50 Hz from the electricity current. Some basic algorithm, such as the periodogram can be used for a first view. Load the file `unknowndata`. The unknown data is contained in `data`.

Plot the data and check if there seems to be a large mean that differ from zero. If this is the case, the periodogram analysis will not work properly and an estimated mean, should be subtracted before the analysis. Decide if this is the case and in that case adjust data to zero-mean.

Your own periodogram is easily implemented using the Discrete Fourier Transform (DFT) and can be computed in MATLAB, using the command `fft`

```
>> X=fft(x);
```

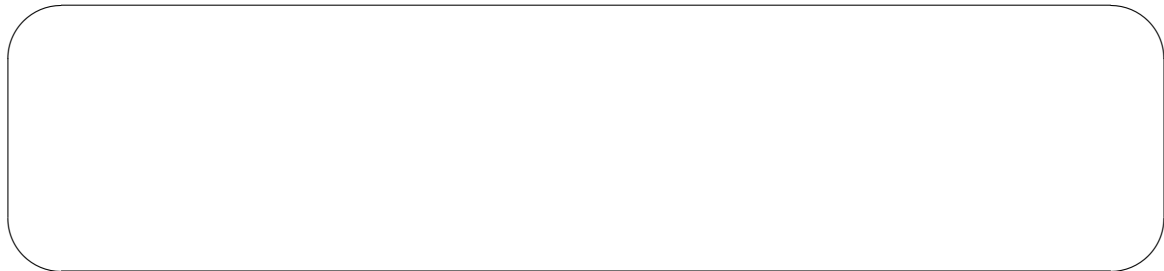
and

```
>> n=length(x);
>> Rhat=(X.*conj(X))/n;
>> f=[0:n-1]/n;
>> plot(f,Rhat)
```

If you prefer the view of frequencies $-0.5 \leq f < 0.5$, use the command `fftshift` according to

```
>> plot(f-0.5,fftshift(Rhat))
```

Q. What do you see? Can you say anything about the signal?



An much better resolution view is given using *zero-padding*, i.e.,

```
>> X=fft(x,N);
```

where N specifies the number of frequency values your spectrum estimate is computed for. For fast computation the number of frequency values should be chosen as $N = 2^I$ where I is an integer, (e.g., $N=512, 1024, 2048, 4096 \dots$). Compute and plot the resulting spectrum estimate,

```
>> Rhat=(X.*conj(X))/n;
>> f=[0:N-1]/N;
>> plot(f,Rhat)
```

Note that the normalizing factor of the spectral estimate should still be the data length and not the FFT-length N .

Q. How does the use of zero-padding change your spectral estimate? The unknown signal is a realization of a cosine function process, see preparation exercise 2, (initially with a non-zero-mean). Can you identify the frequency?



The inverse FFT can be used to compute the covariance function estimate, and we plot the values for small τ , $\tau = 0, \dots, 15$, as

```
>> rhat=ifft(Rhat);
>> plot([0:15],rhat(1:16))
```

Q. Can you explain the shape of the estimate from theory?



3 Investigation of spectral estimation techniques

Different non-parametric spectral estimation methods should be compared in their ability to estimate realizations of a stationary stochastic process model, an ARMA(4,2)-process. We will learn more of this model the coming weeks of the course. The true spectral density of the process is depicted in Figure 1, in logarithmic scale.

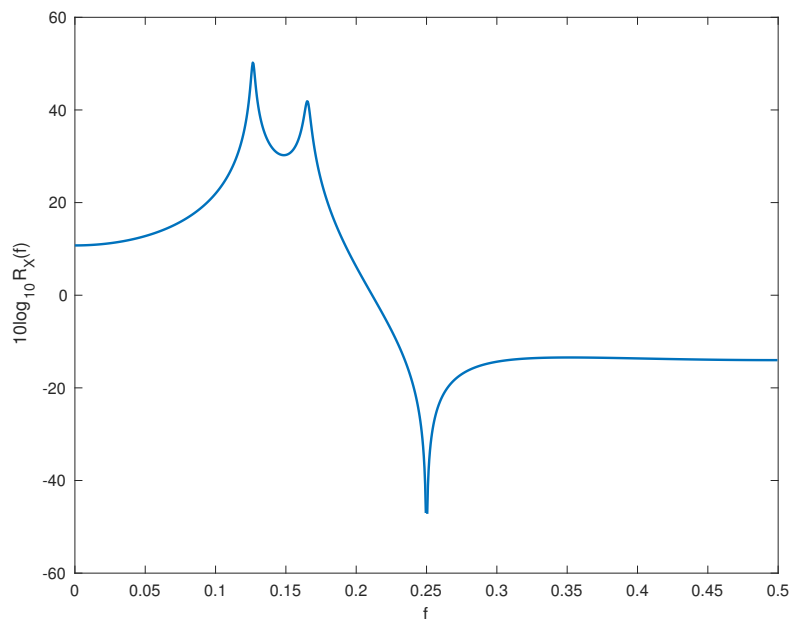


Figure 1: The spectral density of the ARMA(4,2)-process.

Simulate a 500 sample realization yourself of the ARMA(4,2)-process using

```
>> e = randn(500,1);
>> modell.A=[1 -2.39 3.35 -2.34 0.96];
>> modell.C=[1 0 1];
>> x = filter(modell.C, modell.A, e);
>> plot(x)
```

One may compute the spectral density $R_X(f)$ for the process as

```
>> [H,w]=freqz(modell.C,modell.A,2048);  
>> R=abs(H).^2;  
>> plot(w/2/pi,10*log10(R))
```

where `freqz` is a MATLAB function that calculates the frequency function corresponding to the polynomials `C` and `A` in a number of equally spaced frequency values (default 512) between $0 \leq \omega < \pi$, with $\omega = 2\pi f$.

Estimate the spectral density using the MATLAB periodogram function, with zero-padding up to fft-length 4096,

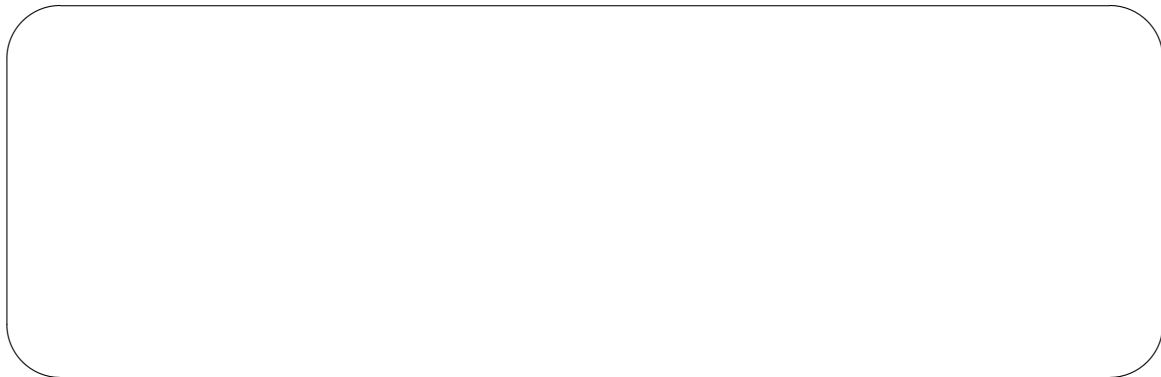
```
>> periodogram(x,[],4096);
```

which depicts in dB-scale. Note the frequency-scale that is normalized between 0 and 1, which is MATLAB normalization which actually should be interpreted as 0 to 0.5! Compare with the true spectral density and verify that the frequency peaks are at correct locations. Run the commands above again to simulate a new realization and plot the periodogram. Note the large variation in the spectrum estimate from one realization to another!

Now compute a spectrum estimate using a Hanning windowed, (modified) periodogram,

```
>> periodogram(x,hanning(500),4096);
```

Q. How does the periodogram and the Hanning windowed periodogram differ? Sketch the estimates and explain.



To visualize in linear/logarithmic scale and to receive the computed spectrum estimates as output from the periodogram function use

```
>> [Rhat,f]=periodogram(x,[],4096,1);  
>> plot(f,Rhat) % Linear scale  
>> semilogy(f,Rhat) % Logarithmic scale
```

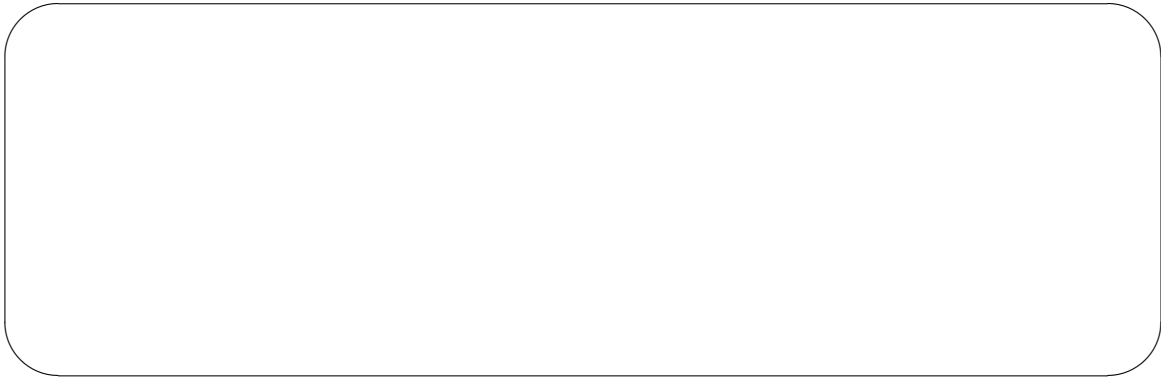
where the last 1 after the fft-length 4096 is the sampling frequency. Specifying this will give the output frequency axis `f` between 0 and 0.5, which is convenient. See the MATLAB documentation for further details.

Now use the Welch method with $K=10$ time-shifted Hanning windows and 50% overlap. The length L of the time-shifted window needs to be specified. Use your result from preparation exercise 5.

```
>> pwelch(x,hanning(L),[],4096);
```

Simulate another realization (or a few other) of your ARMA-process, and apply the Welch method for the spectrum estimate.

Q. What are the differences using the Welch estimator compared to the periodogram? Sketch and explain.



A simple performance analysis of methods can be made using just a white noise sequence, i.e., e of your ARMA-process. As the spectrum estimate of the white noise sequence should be constant we can compare the variance of the spectrum estimates of a single realization, e.g., from the periodogram and the Welch method,

```
>> Rhate=periodogram(e,[],4096);  
>> Rhatew=pwelch(e,hanning(L),[],4096);
```

where we compute the variances of R_{hate} and R_{hatew} and the ratio of them, i.e.,

```
>> var(Rhate)/var(Rhatew)
```

Repeat this a few times for a new simulation of the white noise sequence.

Q. What is the relation between the two variances? Is this in concordance with theory?



If you are interested, also use the Thomson multitaper estimator with 10 multitapers and compare the resulting spectrum estimate to the result of the Welch estimator.

```
>> pmtm(x,(10-1)/2,[],4096);
```

4 Spectral analysis of an EEG signal

The measurements of the electrical signals from the brain are called ElectroEncephalogram (EEG) signals and are often clinically used for detection of certain diseases or brain damages. A flickering light as stimulus is used as trigger for a possible seizure for diagnosis of photosensitive epilepsy. However, also for a person without epilepsy the response of the such a stimulus is quite strong, shown as a periodic variation with the same frequency as the light flickers. In general, the EEG signal is a non-stationary process, but for shorter periods (1-2 seconds), the EEG activity can be assumed to be stationary. In this exercise, you will study a number of EEG sequences using `spekgui` and judge which sequence that is recorded from a measurement with a flickering light. Start by loading the file `eegdata12` which contains three different measurements of EEG (5 s) from the same person. One of the sequences are measured when a flickering light of 12 Hz was introduced.

Study the variables `data1`, `data2` and `data3` by loading them one by one into `spekgui` and examine the realization, the covariance estimate and the spectrum estimate. Compare the use of the periodogram and the Welch method.

Q. Which sequence is most likely to come from the 12 Hz flickering light?

Load the file `eegdatax` and study the different variables `data1`, `data2` and `data3` in `spekgui`. Now the frequency of the flickering light is unknown. (Hint: it is between 5 and 20 Hz).

Q. Can you judge which frequency it has and which sequence it is introduced into?

5 Identification of stationary Gaussian processes

5.1 Spectral estimates

Load the data of the file, `threeprocessdata`. The sequences `y1`, `y2`, and `y3` are realizations of length `N` from three different zero-mean processes for which the spectral densities are seen in the figure below. The task is now to combine the correct spectral density with the corresponding sequence, `y1`, `y2`, or `y3`, (one from each process), using all tools you have learned. As additional information you also have the white Gaussian noise sequences, `x1`, `x2`, and `x3`, that have been used as input sequences of the filters,

from which the outputs are the sequences y_1 , y_2 , and y_3 , respectively.

Plot the sequences, y_1 , y_2 , and y_3 , and calculate the covariance functions using `covf`. When the frequency contents of realizations are very similar, visualization of data and covariance estimates are not the best tools. Estimate the spectral densities using the periodogram (with some zero-padding), for the three process realizations. Also use the Welch method, e.g., with $K = 10$ time-shifted Hanning windows and study the resulting spectrum. If you have time, change the window length (and thereby also the number of windows) and study the results.

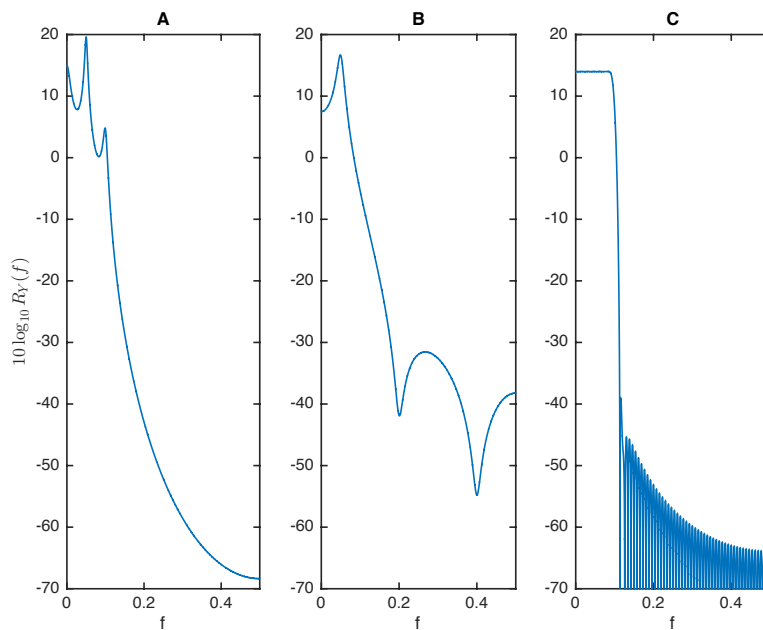


Figure 2: The spectral densities of the three processes.

Q. Compare your periodograms and Welch spectra and the true spectral densities of Figure 2. Identify which sequence that belongs to each of the spectral densities, A, B and C. Sketch and explain.

5.2 Coherence spectrum

You have possibly a good guess of what process each of the realizations comes from, although maybe you are not definitely sure. However, as we also have access to the input sequences, $\mathbf{x1}$, $\mathbf{x2}$, and $\mathbf{x3}$, (white zero-mean Gaussian noise), we could use the cross-spectrum and coherence spectrum to further investigate the realizations. If you want to be sure that no mistakes have been made in the data description and labeling, verify that the input sequences actually are white noise sequences using the Welch method.

We choose the sequences the $\mathbf{y1}$ and $\mathbf{y3}$, and would also like to use information from the input sequences $\mathbf{x1}$ and $\mathbf{x3}$. Talking with the colleague that sent you the data file, you get indications that the input sequences $\mathbf{x1}$ and $\mathbf{x3}$ might have been mixed up and that actually $\mathbf{x1}$ is $\mathbf{x3}$ and $\mathbf{x3}$ is $\mathbf{x1}$. You can easily check this using the coherence spectra between the input sequences (and the shifted input sequences) and output sequences of respective process using the Welch estimator, with 10 windows and overlap 50 %, i.e.

```
>> mscohere(x1,y1,hanning(L), [],4096);
```

```
>> mscohere(x3,y1,hanning(L), [],4096);
```

Q. What is the coherence spectral density according to theory, (see preparation exercise 6)? Relating to this knowledge, which of your coherence estimates seems to be correct? Were the names of the input sequences shifted?

