Lund University
Centre for Mathematical Sciences
Mathematical Statistics

Stationary stochastic processes
HT 2020

Computer exercise 1 in Stationary stochastic processes, HT 20.

The purpose of this computer exercise is to study the estimation of the expected value, covariance function and spectral density for some process realizations, both simulated and real measurements.

Please work in groups of 2, whenever possible, during the computer exercise.

1 Passing the computer exercise

In order to pass the computer exercise you have to:

- Attend your scheduled session. Be on time and make sure that you are at your scheduled session.
- Finish and present the computer exercise within your scheduled 2 hour session. Prepare well.
- Present your work to a teaching assistant before the end of the session. Call the attention of a teaching assistant, who will ask you questions relating to the exercises. Passing requires understanding of the main concepts of the exercise.

Please note that you are always welcome to ask questions during your computer exercise session, just make sure you have prepared well (see next section).

A valid reason for missing a scheduled session, e.g. sickness, needs to be reported to the course responsible (lecturer) BEFORE the start of your session.

2 Preparations

- Carefully read through the entire computer exercise.
- Study chapters 2 and 4 in the course book.
- **Answer the exercises for Computer exercise 1 in Mozquizto at latest Monday September 14 at 12.00 noon.**

  Mozquizto is found at http://quizms.maths.lth.se/ where you create a personal account using 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.
- The computer exercises require basic knowledge of MATLAB. If you have not used MATLAB, prepare by looking up the commands needed in this exercise.

3 Estimation of the expected value, covariance function and spectral density

3.1 Estimation of expected value

Please find the additional files and data on the course webpage and upload these into your computer. Load the file `data1.mat` using the command `load data1`. The file contains a realization of 100 measurements of white noise with the unknown expected value m. **Remember to save your MATLAB commands in a script.**

Plot the sequence,

```
>> plot(data1.x)
```

Q. Does this process have a zero mean?

Estimate the mean with

```
>> mean(data1.x)
```

The measurement values are independent as the process is white noise.

Q. Derive the 95 % confidence interval of the estimate of the expected value, m^*. Can you say that this a zero-mean process?

(Hint: use the MATLAB function `std`).

3.2 Estimation of the covariance function

Load the file `covProc` which contains a realisation of an unknown process. Plot y_t against y_{t-k} for different values of k, e.g. start with the commands,

```
>> k=1
>> plot(covProc(1:end-k),covProc(k+1:end),'.')
```

and change to `k=2` and `k=3` and examine the differences between the plots.

Q. Sketch the view. What do these "scatter plots" represent?

Estimate the covariance function with `xcov`,

```
>> [ycov,lags]=xcov(covProc,20,'biased');
```

You could also use `[ycorr,lags]=xcov(covProc,20,'coeff');` to normalize to the correlation function.

Q. What values did you get? Explain how these values relate to the plots above?

3.3 Spectrum estimate of a sum of harmonics

Amongst the exercise files, there is a function `spekgui` that you can use to estimate the covariance functions and spectral densities for some processes. You can find the help text for `spekgui` on the last page of this exercise. Start the function with the command

```
>> spekgui
```

The function `simuleraenkelsumma` simulates a stationary Gaussian process with a discrete spectral density with the frequencies $f_k = \{5, 10\}$ and the variances $\sigma_k^2 = \{2, 2\}$ and where $\phi_k \in Rect(0, 2\pi)$ and $A_k \in \text{Rayleigh}(\sigma_k^2)$ are independent stochastic variables (If you want to know why the amplitude is assumed to be Rayleigh distributed, read chapter 5.2.2 in the book). A new realization is simulated each time you call the function. The current realization is saved in the MATLAB-variable `data`. Import this to `spekgui` and analyze it using the periodogram.

Q. Do the peaks have equal heights?

To explain this effect, study the variance of the periodogram estimates by simulating new realizations using `simuleraenkelsumma` and import into `spekgui`. Call the function 2-3 times and import `data` into spekgui and analyze for each realization. Investigate how the spectral estimates change.

Q. Draw rough sketches of the spectral density estimates obtained using the periodogram. How do you explain the differences?

4 Student in a symphony orchestra

4.1 Keynotes and overtones

The sound from most acoustic instruments consist of a fundamental frequency, often termed a keynote, and some overtones. The phases of the overtones typically depend on the instrument and are partly correlated with the swinging of the keynote. This, together with the relation between the power of the overtones, produces the perceived sound of the instrument.

If the keynote has frequency f_0, what are the frequencies of the overtones? This will depend on the type of instrument, but for string instruments, the overtones can be well represented[1] as

$$f_k = kf_0,$$

with $k = 1, 2, \ldots$. Load the files `cello.mat` and `trombone.mat`. These files contain

[1] It is worth noting that the stiffness of the string will actually produce some frequency offsets such that the overtones will not be exact multiples of the fundamental frequency. A more precise model of the overtones taking the string stiffness into account can be found as

$$f_k = kf_0\sqrt{1 + Bk^2}$$

where B is a positive stiffness parameter.

the signals of a tone played by a cellist and a trombonist at The Academic Orchestra in Lund[2]. You can listen to the tones by using the command `soundsc(cello.x)`.

Import the data (`cello` or `trombone`) into `spekgui` and estimate the spectral densities using some appropriate method (e.g., using Welch's method with 2-3 overlapping windowed sequences; this is given in `spekgui` as a parameter). Examine the result using both a linear and logarithmic scale.

Q. What are the frequencies of the cello and trombone keynotes ?

Q. Do the overtones appear at integer multiples of the keynotes and how many overtones can you see for the cello and the trombone sounds? (hint: use the logarithmic scale).

These sounds were recorded using a really bad tape-recorder, and thus contain a lot of noise.

Q. Can you see a strong noise peak at a particular frequency? (hint: the tape recorder was not battery charged.)

4.2 Aliasing

Start with studying the spectrum of the cello using `spekgui`. Then, create a down-sampled realisation by extracting every second sample from the original signal (save your MATLAB code)

```
>> n=2;
>> cello2.x=cello.x(1:n:end);
>> cello2.dt=cello.dt*n;
```

Import `cello2` to `spekgui` and study the spectrum.

[2]Founded 1745, the orchestra still today plays at doctoral promotions, professor installations, hälsningsgillen and symphony concerts.

Q. Has the spectrum changed? How has the spectrum range changed? At what frequencies do aliased peaks (if any) appear?

Examine the `trombone` process in the same way. Try some different values of `n` for different down-sampling.

Q. How much slower must this signal be sampled to give an aliasing in the spectrum?

A correct down-sampling, without aliasing, is obtained if the signal is low-pass filtered before the down-sampling. This can be done using the MATLAB-function `decimate`

```
>> cello2.x=decimate(cello.x,2);
>> cello2.dt=cello.dt*2;
```

Q. Are there still any aliased peaks in the spectrum?

5 MATLAB–functions

spekgui

```
function spekgui(action,varargin)
% SPEKGUI
%
% spekgui  opens a window for spectral estimation.
%
% Import data by putting them into a "structure", write the name in the "Import"-box
% and push the button.
% Example:
%
% >> litedata.t=linspace(0,50,1001);
% >> litedata.x=sin(2*pi*litedata.t)+randn(1,1001)*0.5;
%
% The different spectral estimation methods are :
%
%   Periodogram
%   Bartlett: averaging over m periodograms.
%   Welch: averaging with 50% overlap and Hanning windows.
%
% The covariance function is estimated from data or from
% the spectral density estimate.
%
% The estimates of the covariance function and spectral density
% is exported to Matlab with the "Export"-button.
%
```