Lund University
Centre for Mathematical Sciences
Mathematical Statistics

Stationary stochastic processes
HT 2022

Computer exercise 1 (Ladok-Lab 1).

In this computer exercise you will study estimation of expected value, covariance function and spectral density for some process realizations. You will also study and compare the frequency content from recordings of music instruments.

Please find all the additional files and data on the course webpage and download these into your computer. **Remember to save your MATLAB commands and code in scripts for easier access of the results at the presentation.**

Please work in groups of two students!

1 Introduction to estimation of expected value, covariance function and spectral density

In this section you will learn how to use the basic techniques for estimation of important functions of a stationary stochastic process.

1.1 Estimation of expected value

Load the file `data.mat` using the command `load data`. The file contains three realizations, each of 100 samples of Gaussian white noise. Plot the sequences,

```
>> plot([data1 data2 data3])
```

Q1. What conclusions can you make from this figure? Is it reasonable to assume that all realizations are from the same stationary stochastic process? Does this process have zero mean?

Estimate the three expected values using

```
>> m1=mean(data1)
```

and similar for the other two realizations. Derive 95 % confidence intervals of all your expected value estimates, using the knowledge that the variance is 0.25 in all cases.

Q2. Viewing the three confidence intervals, can you say that none, some or all of the underlying processes have zero-mean?

Use the knowledge (or assumption) that all the realizations actually come from the same process. Compute a more reliable estimate of the expected value of the process.

Q3. What is your final estimated expected value of the white noise process?

1.2 Estimation of covariance function

Load the file `data2` which contains a realisation of an unknown process as the variable `y`. Estimate and plot the covariance function using `xcov`,

```
>> [r,lags]=xcov(y,20,'biased');
>> plot(lags,r)
```

Use `help xcov` for more information on definition and input parameters. In a new figure, estimate the correlation function using

```
>> [rho,lags]=xcov(y,20,'coeff');
>> plot(lags,rho)
```

Verify the correspondence between the covariance and correlation functions by comparing the values in the two plotted curves.

Q4. What is the connection written as a formula?

In a new figure, plot all possible values of `y(t)` and `y(t-k)` against each other in a 'scatter plot' for different values of `k`, e.g.

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

Compare with the resulting figures if you change to `k=2` and `k=3` (in new figures).

Q5. Explain how the view in the scatter plots relate to the values in the correlation function.

1.3 Estimation of spectral density

The script `signalsim` simulates one realization of a Gaussian stationary harmonic process according to

$$x(n) = A_1 \cos(2\pi \frac{f_1}{fs} n + \phi_1) + A_2 \cos(2\pi \frac{f_2}{fs} n + \phi_2), \quad n = 0 \ldots N-1,$$

with frequencies $f_k = \{10, 20\}$Hz, number of samples $N = 500$ and sample frequency $fs = 256$ Hz. The independent phases are $\phi_k \in Rect(0, 2\pi)$ and the independent amplitudes are $A_k \in$ Rayleigh(σ_k) with parameters $\sigma_k = \{2, 2\}$.

Code and plot a simulated realization of the above process corresponding to a time scale in seconds or run the script `signalsim`.

Q6. What is the connection between the maximum time-value in seconds, number of samples N and the sampling frequency fs in Hz?

Estimate and view the covariance function for an appropriate `maxlag` and plot the resulting function. Let the x-scale be lag-values expressed in seconds!

Q7. Do the periodicities of the realization and covariance function match each other?

If your answer is no, you have made a mistake in your calculation of the lag-scale (or the time-scale)!

Estimate the spectral density with the periodogram, at `NFFT=2048` frequency values, as

```
>> [R,f]=periodogram(x,[],NFFT,fs);
```

Plot the resulting periodogram with f as the x-scale.

Q8. How does the frequency of the peaks relate to the periodicity of the realization (and covariance function)?

Study the peak heights of the two frequencies shown in the periodogram.

Q9. Why are the heights of the periodogram peaks different from each other?

Run your code or the `signalsim`-script again to view a new simulated realization. Compute and plot the periodogram. Repeat the procedure of simulting a new realization and plotting the periodogram a couple of times. Sometimes the resulting periodogram seems to contain just one frequency. Choose one such case and study the periodogram in a new figure by changing the y-scale to decibel (dB),

```
>> plot(f,10*log10(R))
```

Compare the two different ways of visualizing the periodogram.

Q10. What is the advantage of using the dB-scale?

2 Instruments in a symphony orchestra

The sound from an acoustic instrument consists of a fundamental frequency, often termed a keynote, and some overtones or harmonics. The phases of the overtones typically depend on the instrument and are partly correlated with the oscillation 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, the frequencies of the overtones will depend on the type of instrument. For string instruments, the overtones can be well represented as $f_k = kf_0$, $k = 1, 2, \ldots$ The examples used in this exercise are recorded by the Philharmonia musicians, and are just a few examples of many, found at their webpage[1].

2.1 Keynotes and overtones

Download the file `cellodiffA`, where you find three variables, `celloA2`, `celloA3` and `celloA4`, each representing a recording of the note A. The sampling frequency is `fs=44100` Hz. Listen to the tones by using the command `sound`. Note: if the sampling frequency is not given as an input parameter to `sound`, the assumed sampling frequency is 8192 Hz, which will result in a strange sound!

Estimate the corresponding spectral densities using the periodogram with dB-scale and compare the keynote and overtones frequencies for the three cases).

Q11. What are the keynotes and overtones in the three cases and how do the frequencies of the three different A relate to what you hear? Does the frequency structure match the simple overtone-model described above?

[1] https://philharmonia.co.uk/resources/sound-samples/

2.2 Cello and flute

In the previous exercise, the extraction of frequencies from the different sounds could be a winning strategy to differ or classify these sounds. If we instead listen to two different instruments playing exactly the same note A, it is much more difficult to differ them. Download the file `celloandflute`, where you find two examples of the cello-tone A and two examples of the flute-tone A. Listen to the sounds!

Study the periodograms of all four cases.

Q12. What differences do you see in the periodograms between the two instruments? Suggest some relevant features to be extracted from periodogram structure that could be used to differ the two instruments.

2.3 The spectrogram

Download the file `cellomelody`, where you find two examples of sequences played at the cello, `melody1` and `melody2`. Listen to both sequences. Can you hear the difference between them so you are able to identify them?

As the sequences are now time-varying we use the spectrogram to visualize a number of stacked periodograms. Use a window length of `window=2048` for the data of each periodogram, an overlap of `1024` samples for the calculation of a new periodogram (window/2 is usually enough but can be increased up to window-1 for a more detailed time scale), `NFFT=8192` (must be larger or equal to the window length which represents the actual data length in each periodogram).

```
>> spectrogram(melody1,window,noverlap,NFFT,fs,'yaxis');
```

The x-axis represents time in seconds and the y-axis is the frequencies in Hz. The colours of the image show the power according to the colour bar. Visualize both sequences and compare the images. Try to identify which spectrogram that belongs to `melody1` and `melody2` respectively just by listening and viewing an image. Try to zoom in for a reduced view of the frequency scale below 1500 Hz, and see if it becomes easier.

Q13. Can you differ the two sequences by listening and also identify the differences in the spectrograms?

2.4 Decimation and aliasing

Similar to the exercise above, where you reduced the view to zoom in on the keynote for a proper identification, a down-sampling or decimation is often useful in analysis and modelling of oscillating data. Decimate `melody2` a factor 8 with

```
>> melody2dec=decimate(melody2,8);
```

Listen to the decimated sequence. Can you hear any differences from the original data? (The differences should be small and possible differences will depend on the quality of the loudspeaker.)

Make sure you keep track of the inut sampling frequencies to `sound`, so they match the actual sampling rate of the data!

Use `spectrogram` to visualize the original data and the decimated data in different figures. A better view of the decimated sequence will be given with use of a shorter window as there now are fewer samples in each tone. E.g. use a window length of `512` and an overlap of `256` samples. Compare the spectrogram of the original and decimated data.

Q14. Are the keynote frequencies the same in the two images?.

If the answer of this question is no, you should check the used sampling frequencies!

The `decimate`-command starts with a low-pass filtering of the data followed by the decimation by picking samples from the filtered data with an interval corresponding to the decimation factor. Use the following command

```
>> melody2alias=melody2(1:8:end);
```

to perform a decimation without the low-pass filter.

Listen to this sequence and visualize with the spectrogram.

Q15. What are the differences between `melody2dec` ***and*** `melody2alias` ***when you listen to them? What are the main differences between the spectrograms? Can you explain the differences?***

This comparison aims to show how important it is to be careful in the filtering procedure before decimation of data.

2.5 The mel spectrogram

The mel scale, named after the word melody, is a perceptual scale of pitch, where fundamental frequencies are judged by listeners to be equal in distance from one another. The mel scale was invented almost 100 years ago and is subjective to human perception. And yet, this is the overall most applied algorithm to construct time-frequency images of audio data, to be used as features for classification. A mel spectrogram, with standard settings, can be calculated using

```
>> melSpectrogram(melody2,fs);
```

Q16. Do you think the mel spectrogram always represents an optimal feature extraction for audio data?

Optional: If you are interested you can visualize the filterbank used in the mel spectrogram with

```
>> FI=[0:NFFT/2]/NFFT*fs;
>> [fb,cf]=designAuditoryFilterBank(fs,"FFTLength",NFFT,"NumBands",32);
>> plot(FI,fb)
```

The figure shows the structure of the triangular mel filters as a function of frequency from zero to half the sampling frequency (22050 Hz). For lower frequencies, the filters are keeping a better frequency resolution as the bandwidths of these filters are more narrow. For higher frequencies, the bandwidths of the filters are wider and thereby several adjacent frequencies are averaged. The frequency resolution therefore becomes worse for higher frequencies.