02427 Advanced Time Series Analysis
Computer Exercise 2

The main topic of this exercise is parameter estimation.

Part 1

Estimate the parameters in at least one of the systems generating the time series that you simulated in Part 1 of Computer Exercise 1. Use the prediction error method (also referred to as the conditional least squares method) for the estimation.

Hints

See Section 5.5 on page 115. Implement and minimize the loss function

$$Q_n(\theta)$$

to estimate the parameters.

To make Part 2 feasible it is advisable to estimate two parameters and keep any remaining parameters fixed.

R: Use for example the function `optim()` to do the minimization, see `?optim`. You have to give the function that is to be optimized as an argument. So make a function which takes the parameters and returns the value of the loss function, e.g.

```
RSSSetar <- function(theta) {
 p1 <- theta[1]
 p2 <- theta[2]

 ## Calculate the objective function value
}
```

Then give the initial values of `theta` and the function to `optim()`.

Matlab: You may use `fminsearch`.

Part 2

Use the model chosen in Part 1 and plot contour curves for the loss function for different values of the number of observations, N. Discuss your findings in the report.

Hints

Choose two parameters p_1 and p_2 and calculate $Q_N(p_1, p_2)$ on a regular grid. Plot the contours of $Q_N(p_1, p_2)$ and repeat for subsets of different sizes. For example, if 3000 steps were simulated, plot contours based on

```
1:3000
1:300
1:30
1001:1300
1001:1030
```

R: Use for example `filled.contour()`, which makes a contour plot with a color key next to the figure. To change the colors, use `color.palette` as an argument, e.g.

```
color.palette=colorRampPalette(
c("#00007F","blue","#007FFF","cyan",
"#7FFF7F","yellow","#FF7F00",
"red","#7F0000"))
```

Another tip is to set the plot labels with `title()` and `expression()`, see `?plotmath`, e.g.

```
title(xlab=expression(a[1]),
ylab=expression(a[2]),
line=1.5)
```

Matlab: Use `imagesc()` for contour plots.

Part 3

Choose a model for a doubly stochastic system. Write the model on state-space form and simulate this model. Then use the Kalman filter to estimate the state based on the simulated measurements.

Hints

Write the model on state-space form with mathematics in the report. This is done by reparameterizing the model the same way as in the book on page 149. Implement and simulate, and comment on the results.

Part 4

The last part of this exercise is concerned with the Extended Kalman Filter (EKF).

Introduction

First, consider the following simple state-space model:

$$x_{t+1} = ax_t + u_t + v_t, \tag{1a}$$

$$y_t = x_t + e_t, \tag{1b}$$

where a is an unknown parameter, u_t, is an exogenous input that can excite the system, and v_t and e_t are mutually uncorrelated white-noise processes with variances σ_v^2 and σ_e^2, respectively.

Your task is to use an EKF to estimate a by including it in the state vector, and study the implications of (erroneous) specifications of the variance σ_v^2 and initial values in the filter.

Part 4a

Do a number of simulations (for example 20) of the model having

$$a = 0.4$$

and the noise variances

$$\sigma_v^2 = \sigma_e^2 = 1.$$

Rewrite the model by including the parameter in the state vector, such that it can be used for parameter estimation with the EKF, as described on page 152 (Section 7.4) in the book.

Part 4b

Do the following to draw some conclusions about the convergence properties for the EKF used. Estimate the parameter a in all simulations,

- with the initial value of a set to 0.5,
- and afterwards to -0.5.

Plot the estimates and do this for the four combinations:

- In the filter set $\sigma_v^2 = 10$ and initial value of the variance for the parameter a to 1.
- In the filter set $\sigma_v^2 = 1$ and initial value of the variance for the parameter a to 1.
- In the filter set $\sigma_v^2 = 10$ and initial value of the variance for the parameter a to 10.
- In the filter set $\sigma_v^2 = 1$ and initial value of the variance for the parameter a to 10.

Hints

If you experience challenges with the estimation, you may use the exogenous input signal, u_t, to *excite* the system in order to 'reveal' its dynamics. You can use a square wave (high values followed by low values followed by high values and so on), a sine wave, a sum of sines, or a pseudorandom binary signal (Matlab has an implementation: Link).

Alternatively, you may use a different variant of the EKF. As mentioned in the lecture, the EKF can be designed in different ways depending on the choice of linearization points. You can find the equations for a different version of the Kalman filter in (90)–(93) and (97)–(98) in the slides here. This variant might be slightly more precise than the variant from the lecture.

R: The EKF is implemented in `EKF-example.R`.

Matlab: The EKF is implemented in `ekf_example.m`.