

Computer Exercise 2

Transfer function models and Prediction

In this computer exercise, you will work with input-output relations, as well as prediction in time series models. Firstly, you will be acquainted with time series having an exogenous input, having to analyze the impulse respons of such a system and from it build a suitable model. Secondly, this computer exercise deals with prediction, perhaps the most important application of time series modeling. You will be expected to make predictions of all models introduced in this course.

1 Preparations before the lab

Review chapters 3, 4, and carefully read chapter 6 in the course textbook. Make sure to read section 4.5, as it deals with transfer function models, as well as this entire computer exercise guide. Answers to some of the computer exercise will be graded using the course's *Mozquizto* page. Ensure that you can access the system before the exercise and answer the preparatory questions as well as (at least) three of numbered exercise questions below *before the exercise*.

You can find the *Mozquizto* system at

<https://quizms.maths.lth.se>

It should be stressed that a thorough understanding of the material in this exercise is important to be able to complete the course project, and we encourage you to discuss any questions you might have on the exercises with the teaching staff. This will save you a lot of time when you start working with the project!

You are allowed to solve the exercise in groups of two, but not more. Please respect this.

2 Lab Tasks

The computer program Matlab and the functions that belong to its System Identification Toolbox (SIT) will be used. In addition, some extra functions will also be used in this exercise. Make sure to download these functions and the required data files from the course homepage. You are free to use other programs, such as R or Python, but then need to find the appropriate functions to use on your own.

2.1 Modeling of an exogenous input signal

In this and in the next section, you will work with modeling of input-output relations, both using the ARMAX model and the transfer function model frameworks. As modeling of a signal which has an exogenous input (an input which is known, i.e., deterministic) is generally more complex than the common time series models encountered so far in this course, one must take care and proceed with caution. Often very simple models of a low order will suffice, while complex ones will only add variance, detrimental to the precision of predictions.

We start by creating a typical time series with a deterministic input signal, using a slight generalization of the ARMAX model, i.e., the Box-Jenkins (BJ) model, having the form of

$$y_t = \frac{C_1(z)}{A_1(z)} e_t + \frac{B(z)z^{-d}}{A_2(z)} u_t$$

where y is the output signal, e is a white noise, u is the input signal, and d is the time delay between input and output. Note that if $A_1(z) = A_2(z)$, we have the standard ARMAX model.

Begin by generating some data following the Box-Jenkins model:

```
rng(0)
n = 500;
A1 = [1 -.65];
A2 = [1 .90 .78];
C = 1;
B = [0 0 0 0 .4];
e = sqrt(1.5) * randn(n + 100,1);
w = sqrt(2) * randn(n + 100,1);
A3 = [1 .5];
C3 = [1 -.3 .2];
u = filter(C3,A3,w);
y = filter(C,A1,e) + filter(B,A2,u);
u = u(101:end), y = y(101:end)
clear A1, A2, C, B, e, w, A3, C3
```

Remark: As discussed in the first computer exercise, we typically generate more data than needed when simulating a process to avoid initialisation effects. Why not use your function from the first exercise here instead?

As you can see, the known input u has here been generated as an ARMA(1,2) process. The `clear` command is to ensure that only y_t and u_t is known in the following, and thus not the polynomials and setting used to generate the realisation.

To model y as a time series of u and e , several steps must be taken beyond regular ARMA modeling. We must first select the appropriate model orders for the polynomials in the model, then proceeding to estimate the parameters of these polynomials.

1. As a first step, we wish to determine the orders of the $B(z)$ and $A_2(z)$ polynomials. Using the transfer function framework, we denote the transfer function from u to y by $H(z) = B(z)z^{-d}/A_2(z)$. In order to estimate the order of the $B(z)$ and $A_2(z)$ polynomials, as well as determining the delay d , we need to form an estimate of the (possibly infinite) impulse response, and from it identify the appropriate models for these polynomials.

As noted in the course textbook, if u is a white noise, the (scaled) impulse response can be directly estimated using the cross correlation function (CCF) from u to y . However, if u is not white, we need to perform pre-whitening, i.e., we need to form a model for the input, such that it may be viewed as being driven by a white noise, and then inverse filter both input and output with this model. In order to do so, we form an ARMA model of the input

$$A_3(z)u_t = C_3(z)u_t^{pw}$$

and then replace u with u^{pw} , i.e.,

$$y_t = \frac{C_1(z)}{A_1(z)}e_t + \frac{B(z)z^{-d}}{A_2(z)} \frac{C_3(z)}{A_3(z)}u_t^{pw}$$

The pre-whitening step, i.e., multiplying with $A_3(z)/C_3(z)$, yields

$$\underbrace{\frac{A_3(z)}{C_3(z)}y_t}_{y_t^{pw}} = \underbrace{\frac{A_3(z)}{C_3(z)} \frac{C_1(z)}{A_1(z)}e_t}_{v_t} + \underbrace{\frac{B(z)z^{-d}}{A_2(z)}}_{H(z)} u_t^{pw}$$

and the preferred transfer function model may thus be expressed as

$$y_t^{pw} = v_t + H(z)u_t^{pw}$$

Note that the pre-whitened y_t^{pw} is now the output of the transfer function model, having the preferred uncorrelated signal as its input, allowing $H(z)$ to be estimated using the CCF from u^{pw} to y^{pw} .

Task: Use the basic analysis (`acf`, `pacf`, and `normplot`) to create an ARMA model for the input signal u as a function of white noise, u^{pw} . Which model did you find most suitable for u ? Is it reasonably close to the one you used to generate the input?

QUESTION 1 In *Mozquizto*, answer question 1.

We then pre-whiten y_t , i.e., creating y_t^{pw} . Next, we compute the CCF from u_t^{pw} to y_t^{pw} by typing

```
M = 40; stem(-M:M, crosscorr(upw,ypw,M));
title('Cross-correlation function'), xlabel('Lag')
hold on
plot(-M:M, 2/sqrt(length(w))*ones(1,2*M+1), '—')
plot(-M:M, -2/sqrt(length(w))*ones(1,2*M+1), '—')
hold off
```

As the estimated CCF now yields an estimate of the impulse response, $H(z)$, we can proceed to use this to determine suitable model orders for the delay, and the $B(z)$ and $A_2(z)$ polynomials using Table 4.7 in the textbook. Use `pem` to estimate your model, using

```
A2 = ... ;
B = ... ;
Mi = idpoly([1],[B],[ ],[ ],[A2]);
zpw = iddata(ypw,upw);
Mba2 = pem(zpw,Mi); present(Mba2)
vhat = resid(Mba2,zpw);
```

where the delay may be added to `B` by adding d zeros in the beginning of the vector. If the model orders are suitable, the CCF from u^{pw} to v should be uncorrelated.

Task: Analyze the CCF u^{pw} to y^{pw} to find the model orders of the transfer function. Calculate the residual v_t and verify that it is uncorrelated with u^{pw} . Also, analyze the residual using the basic analysis. Can you conclude that v_t is white noise? Should it be?

QUESTION 2 *In Mozquizto, answer question 2.*

2. We have now modeled y as a function of the input u , but have not yet formed a model of the ARMA-process in the BJ model, i.e., modeled the polynomials $C_1(z)$ and $A_1(z)$. Therefore, defining the ARMA-part as

$$x_t = \frac{C_1(z)}{A_1(z)} e_t$$

we use the estimated polynomials $B(z)$ and $A_2(z)$ and calculate

$$x_t = y_t - \frac{\hat{B}(z)z^{-\hat{d}}}{\hat{A}_2(z)} u_t$$

By filtering out the input-dependent part of the process y_t , we may then estimate determining suitable orders for the polynomials $C_1(z)$ and $A_1(z)$ using the standard ARMA-modeling procedure.

Task: Use the estimates of the polynomials $B(z)$ and $A_2(z)$ obtained for the pre-whitened data and form x_t . Determine suitable model orders for $A_1(z)$ and $C_1(z)$. Was all dependence from u_t removed in x_t ?

QUESTION 3 *In Mozquizto, answer question 3.*

3. Finally, now having determined all the polynomial orders in our model, we estimate all polynomials all together using `pem`.

```
A1 = ... ;
A2 = ... ;
B   = ... ;
C   = ... ;
Mi = idpoly(1,B,C,A1,A2);
z = iddata(y,u);
MboxJ = pem(z,Mi);
present(MboxJ)
ehat = resid(MboxJ,z);
```

Task: Analyze the model residual, verifying that the CCF from u to e , as well as the basic analysis, shows it to be white. Are the parameter estimates significantly different from zero? Can you conclude that the residual is white noise, uncorrelated with the input signal? If not, can you twiddle with the model slightly to improve the residual?

Be prepared to answer these questions when discussing with the examiner at the computer exercise!

2.2 Hairdryer data

In this section, we will try to construct a model for a set of measured data. In the file `tork.dat`, you will find 1000 observations from an input-output experiment. These measurements have been obtained from a laboratory process, which essentially is a hair dryer with measuring equipment, i.e., air is propelled by a fan through a pipe. The air is heated at the entrance of the pipe and its temperature is measured at the outlet. The input signal that is applied, stored in the second column of the data set, is the voltage over the heating coil and the output signal. The first column is the temperature of the airflow at the outlet. This physical systems can be reasonably well modeled using a simple linear model of the process. The sampling distance is 0.08 s.

Start by accessing the data material and subtract the mean values, create an `iddata` object, now having both an input and an output, and plot the first 300 points of the object using

```
load('tork.dat')
tork = tork - repmat(mean(tork),length(tork),1);
y = tork(:,1); u = tork(:,2);
z = iddata(y,u);
plot(z(1:300))
```

Task: Model this input-output relation using the Box-Jenkins model introduced above. Repeating the steps in section 2.1, use the basic analysis and the CCF to find suitable model orders. Finally, estimate the model in its

entirety and plot the `acf`, `pacf`, `normplot`, and CCF from u to e .

How long is the delay from u to y in seconds? Can you conclude that the residual is white noise, uncorrelated with the input signal? Are the parameter estimates significantly different from zero? If not, can you twiddle with the model slightly to improve the residual?

Be prepared to answer these questions when discussing with the examiner at the computer exercise!

2.3 Prediction of ARMA-processes

In this section, we examine how to predict future values of a process, using temperature measurements from the Swedish city Svedala. The temperature data is sampled every hour during a period in April and May 1994, with its (estimated) mean value subtracted (11.35° C). Load the measurements with the command `load svedala`. Suitable model parameters for the data set are

$$\begin{aligned} A &= [\ 1 \ -1.79 \ 0.84 \]; \\ C &= [\ 1 \ -0.18 \ -0.11 \]; \end{aligned}$$

To make a k -step prediction, $\hat{y}_{t+k|t}$, one needs to solve the equation

$$C(z)\hat{y}_{t+k|t} = G_k(z)y_t$$

This can be done using the filter command¹

```
yhat_k = filter( Gk, C, y );
```

where G_k is obtained from the Diophantine equation

$$C(z) = A(z)F_k(z) + z^{-k}G_k(z).$$

To solve the Diophantine equation, you can use the provided function `polydiv`

```
[Fk, Gk] = polydiv( C, A, k );
```

The prediction error is formed as

$$y_{t+k} - \hat{y}_{t+k|t} = F_k(z)e_{t+k},$$

Note in particular that the prediction error will (for a perfect model) have the form of an $MA(k-1)$ process with the generating polynomial

$$F_k(z) = 1 + f_1 z^{-1} + \dots + f_{k-1} z^{-(k-1)}.$$

Note also that if $k = 1$, then $F_1(z) = 1$, suggesting that the prediction error should be a white noise, and that, for this case, the prediction error thus allows for an estimate of the noise variance.

QUESTION 4 *In Mozquizto, answer question 4.*

¹Remember that one has to remove the initial samples after using the command `filter`. Of course, you do not need to worry about forgetting this, as you are using your own filtering function from the first computer exercise, but this will be worth recalling when your friends ask you about why they have problems...

Task: In the following questions, examine the k -step prediction using $k = 3$ and 26. Answer the following questions:

1. What is the estimated mean and the expectation of the prediction error for each of these cases?
2. Assuming that the estimated noise variance is the true one, what is the theoretical variance of the prediction error? Using the same noise variance, what is the estimated variance of the prediction error? Comment on the differences in these variances.
3. For each of the cases, determine the theoretical 95% confidence interval of the prediction errors?
4. How large percentage of the prediction errors are outside the 95% confidence interval? A useful trick might be to use `sum(res>c)` to compute how many elements in `res` that are greater than `c`.
5. Plot the process and the predictions in the same plot, and in a separate figure, plot the residuals. Check if the sequence of residuals behaves as an $MA(k-1)$ process by, e.g., estimating its covariance function using `covf`. If it does not, try to explain why.

Be prepared to answer these questions when discussing with the examiner at the computer exercise!

2.4 Prediction of ARMAX-processes

When predicting ARMAX-processes, one needs to consider also the external input. We will now make use of an additional temperature measurement done at the airport Sturup. The Swedish Meteorological and Hydrological Institute (SMHI) has made a 3-step predictions of the temperature for Sturup, which may be used as an external signal to our temperature measurements in Svedala. Load the SMHI predictions into Matlab, with `load sturup`, and set the model parameters to be

$$\begin{aligned} A &= [1 \quad -1.49 \quad 0.57]; \\ B &= [0 \quad 0 \quad 0 \quad 0.28 \quad -0.26]; \\ C &= [1]; \end{aligned}$$

How large is the delay in this temperature model? How do you know? Form the k -step predictor of the temperature at Svedala using $k = 3$ and 26 as

$$C(z)\hat{y}_{t+k|t} = B(z)F_k^y(z)u_t + G_k^y(z)y_t,$$

where the F_k^y and G_k^y polynomials are computed as above. Then, solve

$$C(z)\hat{u}_{t+k|t} = G_k^u(z)u_t$$

where $G_k^u(z)$ is found from the equation

$$B(z)F_k^y(z) = C(z)F_k^u(z) + z^{-k}G_k^u(z).$$

QUESTION 5 *In Mozquizto, answer question 5.*

After computing the polynomials, solve $C(z)\hat{y}_{t+k|t}^0 = G_k^y(z)y_t$, and then form the prediction

$$\hat{y}_{t+k|t} = \hat{u}_{t+k|t} + \hat{y}_{t+k|t}^0.$$

Task: Using $k = 3$, what is the variance of the prediction errors? Plot the process, the prediction and the prediction errors.

A very common error when making predictions of ARMAX and BJ processes is to forget to add $\hat{u}_{t+k|t}$. Plot this erroneous prediction and the corresponding prediction errors. Can you see how this error appears in your prediction?

Be prepared to answer these questions when discussing with the examiner at the computer exercise!

2.5 Prediction of SARIMA-processes

The temperature measurements from Svedala are very periodical and can therefore likely be well modeled as a SARIMA-process. Load the data and find the suitable period. What is the period for this data?

In Matlab, it is easier to write the SARIMA-processes as an ARMA process by including the differentiation in the $A(z)$ polynomial. Let

$$A^S(z)y_t = (1 - z^{-S})y_t$$

which, in Matlab, is the same as the polynomial

$$AS = [1 \quad \text{zeros}(1, S-1) \quad -1];$$

Forming predictions for the SARIMA-model

$$A(z)A^S(z)y_{t+k|t} = C(z)e_{t+k},$$

may be done seeing it as a non-stable ARMA-model (recall that that polynomial multiplication may be computed using `conv`), and performing predictions for such a model.

Task: After removing the season, form an appropriate model for the svedala data. Which model did you find? Compute the estimated prediction error variance for $k = 3$ and 26, and compare them with the variance obtained from the ARMA model. Are they any better?

Be prepared to answer these questions when discussing with the examiner at the computer exercise!