Uppsala University
Information Technology
Dept. of Systems and Control
Joakim Sorelius 1999
Revised by Erik K Larsson May 25, 2004

System Identification

Computer exercise 5

Time series modeling and prediction

Preparation exercises:
1. Solve the tasks in Section 2.

Name		Assistant's comments
Program	Year of reg.	
Date		
Passed prep. ex.	Sign	
Passed comp. ex.	Sign	

1 Goals

In this computer laboratory we will investigate how to analyze time series. The necessary background to the lab is given in the book *System Identification* (SI). The tasks contain

1. Detrending (polynomial fitting)
2. Estimation of periodical components
3. AR modeling
4. Prediction

The methods for time series modeling and prediction taught in the system identification course require the time series under study to be *stationary*. This means that its mean and variance are independent of time. To obtain such a time series, we usually *detrend* the data. This can for example be done by fitting a polynomial to the data and then removing this polynomial trend.
The presence of deterministic *periodical components* in the data may also hamper the analysis. Such periodical components should thus be removed before modeling.
There are many possible mathematical models for stationary time series: AR, MA, ARMA etc (see Chapter 6 in SI). In this lab we will focus on describing the data as an *AR process* and determine a suitable model order using *Akaike's Information Criterion*, AIC (Chapter 11).
Once a model has been fitted to the data, this model can be used for *prediction* of the future behavior of the time series (Chapter 7). In the case of an AR model, the optimal predictor is particularly easy to determine.

2 Preparations

As a preparation exercise you should solve the following tasks in advance, and study the provided MATLAB code. Note that the provided MATLAB code can give you some hints, how to solve some of the tasks given below.

1. Show how to fit a pth order polynomial to a set of N data points using a least-squares (LS) approach.

Answer:

2. The frequency ω of a periodical component

$$x(t) = A\sin(\omega t + \phi) \tag{2.1}$$

in the data $z(t)$ can be determined from the peak of the periodogram

$$\Phi_z(\omega) = \frac{1}{N}\left|\sum_{t=1}^{N} z(t)e^{-i\omega t}\right|^2 \tag{2.2}$$

Once ω is available, postulate the model $z(t) = x(t) + \varepsilon(t)$, and show how the LS method can be used to determine the parameters necessary to remove $x(t)$ from the time series, using N data points.
Hint: In order to deal with the phase ϕ in a linear manner it can be fruitful to rewrite $x(t)$ using the trigonometric identity $A\sin(\omega t+\phi) = \alpha\sin(\omega t) + \beta\cos(\omega t)$.

Answer:

3. Assume that the detrended time-series (with any periodical components removed) can be described by the following AR model:

$$y(t) = \frac{1}{A(q^{-1})}e(t) \qquad t = 1, 2, \dots, N \tag{2.3}$$

where $e(t)$ is a zero-mean white noise sequence with variance σ^2 and where

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}. \tag{2.4}$$

By using the Yule-Walker method, show how to estimate the AR parameters $\{a_k\}_{k=1}^{n}$ and the driving noise variance σ^2, given an estimate of $r(k) = E\{y(t)y^*(t-k)\}$. Also show how to obtain an estimate of the spectrum $\Phi_y(\omega)$.

Answer:

4. The optimal multistep predictor (for $k \le n$) is given by

$$\begin{aligned}\hat{y}(t+k|t) &= -a_1\hat{y}(t+k-1|t) - a_2\hat{y}(t+k-2|t) - \ldots - a_k y(t) \\ &\quad -a_{k+1}y(t-1) - \ldots - a_n y(t-n+k)\end{aligned} \tag{2.5}$$

where $\hat{y}(t+k|t)$ denotes the prediction of $y(t+k)$ given data up to and including time t.

(a) Give the optimal *one step* predictor for the AR process above. In view of equation (2.5), give the optimal predictor for the case $k > n$ and explain why the output of the AR-predictor tends to zero as k increases.

(b) Using the optimal predictor above for the random component of the time series, explain how to predict the total time series including the trend and the periodical component.

Answer:

3 Prediction of a simulated time series

In this section, we will work on a "fictitious" data series having the following form:

$$y(t) = \frac{1}{A(q^{-1})} e(t) \tag{3.1}$$

$$z(t) = y(t) + p(t) + \sum_{k=1}^{K} A_k \sin(\omega_k t + \phi_k) \tag{3.2}$$

where $p(t)$ is a polynomial of order P, K is the number of sinusoids in the "periodical component" and $A(z)$ in (3.1) is given by (2.4). The data is generated with the MATLAB command **lab5gen** (try several realizations of the data).
The objective of this exercise is to predict the future behavior of $z(t)$ given $N = 300$ data points. In order to solve this task we first have to make the time series stationary (by removing the polynomial trend, see preparation exercise 1). Then the periodical component is removed (preparation exercise 2) and the AR model is estimated (preparation exercise 3). The predictor determined in preparation exercise 4 is then applied to the data series. These tasks are solved with the interactive MATLAB program **lab5**.

1. Generate the data with **lab5gen**. The raw data $z(t)$ is displayed as a function of time. Also the periodogram of $z(t)$ is shown. Is the data stationary?

Answer:

2. Use the program **lab5** to analyze the data. What order of the polynomial trend did you choose? What would happen if you choose a polynomial of a too high degree?

Answer:

3. How many sinusoids did you find in the data? What are their frequencies? Does the estimated periodical component fit the data well?

Answer:

4. What model order did you choose for the AR part of the data? Is the spectrum given by the estimated model close to the spectrum obtained from the data? Are there any discrepancies? If yes, how can they be explained?

Answer:

5. Predict the future behavior of the data. Compare the predicted data to the real time series. Did you do a good job predicting the future?

Answer:

5

6. As can be seen in the top figure, the AR predictor tends to zero as the prediction horizon becomes large (see preparation exercise 4). This is of course the reason why we cannot predict arbitrarily far ahead with good accuracy! How many steps ahead do you manage to predict with reasonable results?

Answer:

4 Prediction of real data

Situate yourself in the happy 80's: the year is 1987, money is abundant and the real-estate market is flourishing. The politicians of a small coastal town in the far north of Sweden have plans to build a large outdoor water activity center: with tax-payer's money, of course. The first reactions from the public were negative: the weather is too cold and the season is far too short for this kind of project!
But the Mayor is not worried; he has read about global warming and is convinced that the average temperature in his town will increase rapidly, especially since he himself has bought a very large American car that pollutes just below the authorized limit.
To support his ideas, you have been given the task to look at temperature data and see if you can find evidence of warmer weather coming the Mayor's way. To your help you have the monthly temperatures measured since 1860. The data is prepared for processing by invoking the MATLAB command **lab5real**.
Analyze the data using **lab5**. Note below the parameters you choose. According to your predictions, what is your advice to the optimistic Mayor? Do you find any signs of warming in this part of Sweden? Will the project be a success?

Answer:

5 Stability monitoring in a nuclear power plant

When a nuclear power plant is running with low core circulation flow and high power, there is an increased risk that the reactor become unstable. This gives an oscillating neutron flux which may lead to a shut down of the reactor and/or fuel damages.

In order to surveillance the reactor stability, many nuclear power plants are using stability detectors. The idea is as follows. A time series model is recursively estimated from the data. Typically an AR model is used:

$$A(q^{-1})y(t) = e(t)$$

The stability margin can be determined by using a measure how close the model is to instability. The standard approach is to use the *decay ratio* (DR), which is defined as the damping between two peaks in the impulse response. DR=1 then corresponds to instability. Here we will study a simplified method, namely to inspect the radius of the pole closest to the unit circle.

In this exercise we will study (real!) data from a Swedish nuclear power plant. Load the data by the command `load zdec`. Plot the data (`plot(zdec)`)! What you see is the reactor power (prefiltered!), sampled at a sampling rate of 3 Hz. Notice that it is difficult to comment about the stability of the process by just inspecting the plotted data.

In the first exercise we will recursively estimate a 2'nd order AR model and plot the largest pole radius as a function of time. Run the file `stabtest` and study the result. Use $\lambda = 0.999$ as default value. As you see, the pole radius is well within the unit circle. However look at the spectra of the signal:

```
g=spa(zdec);
bode(g);
```

Is a second order model feasible?

Answer:

Try different model orders in `stabtest` and inspect the results. You may also want to try different forgetting factors. Is it possible to detect any instability using a higher order model?

Answer:

```
% stabtest
if(~exist('zdec')), load zdec, end
norder=input('Give order (RETURN = 2): ' );
if(isempty(norder)),norder=2;end
lam=input('Give forgetting factor (RETURN = 0.999): ' );
if(isempty(lam)),lam=0.999;end

% An AR model of order norder is to be recursively estimated
th=rarx(zdec,norder,'ff',lam);

% Find the largest root of the estimated model
for j=1:length(th);rr(j,:)=max(abs(roots([1 th(j,:)])))';end

% Plot the pole radius (skip first 100 samples)
plot([100:length(th)]/3,rr(100:length(th)))
title('Largest pole radius as a function of time')
```

6 MATLAB Code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                               %
% MATLAB FILE LAB5  data analysis                               %
%                                                               %
% Computer laboratory number 5, "Time Series Modeling and       %
% Prediction" in the System Identification Course.              %
%                                                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

zoom off
clf;figure(1);zoom xon
clear; load data;

z=y;

% Plot the raw data

subplot(211)
plot(time(1:M)',z(1:M));
title('Raw data')
ylabel('z(t)')
```

```
xlabel('t')

subplot(212)
P=periodogram(z(1:M),zpad);
P=P(zpad/2+1:zpad);
f=linspace(0,1,zpad/2)';
plot(f,P)
title('Periodogram raw data')
xlabel('Normalized frequency')
ylabel('Periodogram')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Estimate polynomial trend
%
disp(' ')
n=input('Give order of polynomial trend: ');

phat=polyfit(t(1:M)',z(1:M),n);

subplot(211)
hold on
plot(time(1:M)',polyval(phat,t(1:M)),'-r');
hold off
legend('Raw data','Polynomial trend',0)

disp('Hit any key to remove polynomial trend...')
pause

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Remove polynomial trend
%

z=z-polyval(phat,t');

subplot(211)
plot(time(1:M)',z(1:M));
title('Data with polynomial trend removed')
ylabel('z(t)')
xlabel('t')

subplot(212)
P=periodogram(z(1:M),zpad);

P=P(zpad/2+1:zpad);
f=linspace(0,1,zpad/2)';

plot(f,P)
%plot(P)
title('Periodogram polynomial trend removed')
xlabel('Normalized frequency')
```

```
ylabel('Periodogram')

disp('Estimation of sinusoidal frequencies...')
disp(' ')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Estimate peaks
%
n=input('Give the number of sinusoids: ');

fhat=f(findpeaks(-P,n)');

disp(['fhat=',num2str(sort(fhat)')]);

subplot(212)
ax=axis;
ax=ax(3:4)';
line([1;1]*fhat',ax*ones(1,n))
title('Periodogram, estimated peak frequencies')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Estimate data
%

Ph=[cos(pi*fhat*t) ; sin(pi*fhat*t)];
th=inv(Ph*Ph')*Ph*z;

zsin=Ph'*th;

subplot(211)
hold on
plot(time(1:M)',zsin(1:M),'-r')
hold off
legend('Data','Estimated sinusoidal data')

disp('Hit any key to remove sinusoidal data...')
pause

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Remove sinusoidal data
%

z=z-zsin;

subplot(211)
plot(time(1:M)',z(1:M));
title('Data with sinusoid(s) removed')
ylabel('z(t)')
xlabel('t')
```

```
subplot(212)
P=periodogram(z(1:M),zpad);
P=P(zpad/2+1:zpad);
f=linspace(0,1,zpad/2)';
plot(f,P)
title('Periodogram with sinusoid(s) removed')
xlabel('Normalized frequency')
ylabel('Periodogram')

disp('Hit any key to select the order of the model...')
pause

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Choose a model order
%

V=[];
Ahat=[];
si2=0;

nmax=50;
Sys=zeros(nmax,nmax+1);
for n=1:nmax,

  r=xcov(z(1:M),z(1:M),n,'biased');
  R=toeplitz(r(n+1:2*n+1));

  ah=-inv(R(2:n+1,2:n+1))*R(2:n+1,1);

  Ahat=[1 ah'];
  Sys(n,1:n+1)=Ahat;
  si2(n)= R(1,1:n+1)*Ahat';

  e=filter(Ahat,1,z(1:M));


  V(n)=e'*e/M;

end


subplot(212)
plot(V(1:20))
title('AIC loss function V_N (\theta)')
xlabel('Number of parameters')
ylabel('Loss function')


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
```

```
% Estimate AR system of order n
%
disp(' ');
norder=input('Give the order of the AR process: ');

Ahat=Sys(norder,1:norder+1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Compare AR spectrum
%

clf

ind=exp(i*linspace(-pi,pi,M));
Phat=1/M*e'*e./(polyval(Ahat,ind).*polyval(Ahat,conj(ind)))';
Phat=Phat(round(M/2)+1:M);

f=linspace(0,1,zpad/2)';
fh=linspace(0,1,length(Phat))';
plot(f,P,fh,Phat)
title('True and estimated periodograms')
xlabel('Normalized frequency')
ylabel('Periodogram')
legend('Periodogram data','Estimated periodogram')
grid

disp('Hit any key to predict data ...')
pause

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Optimal predictor
%
tlag=t(M:M+maxlag);

zpred=[z(M-norder:M) ; zeros(size(tlag'))];

for k = norder+2 : maxlag+norder+1,
  zpred(k)=-Ahat(2:norder+1)*zpred(k-1:-1:k-norder);
end;

zpred=zpred(norder+1:maxlag+norder+1);

% Predicted AR data

subplot(211)
plot(time(M-maxlag:M+maxlag)',z(M-maxlag:M+maxlag),'-b',time(M:M+maxlag)',zpred,'-

title(['Crossvalidation AR data, Prediction starts at Jan 1987'])
ax=axis;
line([time(M) time(M)],[ax(3) ax(4)])
```

```
legend('Measured data','Predicted data',0)
ylabel('y(t)')
xlabel('t')

% total data

Ph=[cos(pi*fhat*tlag) ; sin(pi*fhat*tlag)];
zsin=Ph'*th;
zpoly=polyval(phat,tlag');
zptot=zpred+zsin+zpoly;

subplot(212)
plot(time(M-maxlag:M+maxlag)',y(M-maxlag:M+maxlag),'-b',time(M:M+maxlag)',zptot,'-

title(['Crossvalidation total data, Prediction starts at Jan 1987'])
ax=axis;
line([time(M) time(M)],[ax(3) ax(4)])
legend('Measured data','Predicted data',0)
ylabel('z(t)')
xlabel('t')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                               %
% MATLAB FILE LAB5GEN  Simulated data generation                %
%                                                               %
% Computer laboratory number 5, "Time Series Modeling and       %
% Prediction" in the System Identification Course.              %
%                                                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Generate the raw data

K=3;       % Number of sinusoids
N=1000;    % Number of data points
M=300;     %       - " -          used in estimation
tN=1:N;
maxlag=10;
t=tN(1:M+maxlag);
time=tN(1:M+maxlag);

e_w=randn(N,1);
x=randn(N,1);

om=[1.24 1.51 1.6]';
fi=[3 3 6]';
a=ones(K,1);

A=[1 0.64 0.7];
%A=[1 -2.76 3.809 -2.654 0.924];
B=[1];
```

```
x=filter(B,A,e_w);

p=[5e-2 1e-3];

y=x+(a'*sin(om*tN+fi*ones(size(tN))))' + polyval(p,tN)';
y=y(1:M+maxlag);

save data

zoom off
clf;figure(1);zoom xon
clear; load data;

z=y;

% Plot the raw data

subplot(211)
plot(t(1:M)',z(1:M));
title('Raw data')
ylabel('z(t)')
xlabel('t')

subplot(212)
P=periodogram(z(1:M));
P=P(M/2+1:M);
f=linspace(0,1,M/2)';
semilogy(f,P)
title('Periodogram raw data')
xlabel('Normalized frequency')
ylabel('Periodogram')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                               %
% MATLAB FILE LAB5REAL  Prepare real data                       %
%                                                               %
% Computer laboratory number 5, "Time Series Modeling and       %
% Prediction" in the System Identification Course.              %
%                                                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Prepare the real data

load temperature;

X=stensele;
clear stensele;

year=flipud(X(:,1))';
X=flipud(X(:,2:13));
[N,m]=size(X);
```

```
years=127+1/12;
M=round(12*years);
maxlag=24;
y=reshape(X',m*N,1);
y=y(1:M+maxlag);
time=linspace(year(1),year(1)+years+maxlag/12,M+maxlag+1);
time=time(2:M+maxlag+1);
t=1:M+maxlag;
zpad=2^15;

save data;

zoom off
clf;figure(1);zoom xon
clear; load data;

z=y;

% Plot the raw data

subplot(211)
plot(time(1:M)',z(1:M));

title('Raw data')
ylabel('z(t)')
xlabel('t')

subplot(212)
P=periodogram(z(1:M),zpad);
P=P(zpad/2+1:zpad);
f=linspace(0,1,zpad/2)';
plot(f,P)
title('Periodogram raw data')
xlabel('Normalized frequency')
ylabel('Periodogram')
```