UPPSALA UNIVERSITET
AVDELNINGEN FÖR SYSTEMTEKNIK
BC 9901. Last revision September 18, 2014

Computer exercise 2 for the course: Empirical Modelling

Using the graphical interface Ident and comparing some methods

Preparation exercises:
1. Read Ch 12 and 14 in the textbook

Name	(Assistant's comments)
Program Year of reg.	
Date	

Contents

1 Goals

The aim of this computer laboratory is to get you familiar with the the System Identification toolbox. Of particular interest is the use of the graphical interface (ident) developed for the identification toolbox commands.

A further study of the toolbox and a validations of ARX models and ARMAX models are also given as an optional task.

2 Preparations

The necessary background for this computer laboratory can be found in Chapters 12 and 14 of *Modellbygge and simulering*.

3 Ident- The graphical user interface (main task)

In this first task the aim is to learn how to use the System Identification Toolbox (SIT). In particular the graphical user intercase ident will be used.

1. Start the demo by typing `iddemo`. Select demo 1 and follow the guided tour on how to use the graphical user interface. In task 36 in the demo change
   "Use the popup menu in the Parametric Models dialog to select State Space"
   to
   "Use the popup menu in the Parametric Models dialog to select ARX. Then click on order Selection followed by Estimate. All model orders from 1 to 10 is then estimated and evaluated"
   After the demos is finished, you may inspect more carefully different alternatives in the interface.

2. Generate some data by yourselves and use the interface to identify a parametric model. You may proceed as follows[1].

```
%Define the true system (the example below corresponds to an ARX model with
%nk=1, nb=2, na=2). DO help idpoly for more information
B = [0 1 0.5];
A = [1 -1.5 0.7];
Mtrue=idpoly(A,B,1)

%Generate a binary input signal (details:  help idinput)
N=500; %Number of data points
u = idinput(N);

%Generate white Gaussian measurement noise
lambda=0.5;
e = idinput(N,'rgs')*lambda;
% Remark: the "blips" around rgs may in the text above look like
% superscripted comma signs but it should
```

[1] An alternative would be to use 'filter' to generate the data as in Computer lab 1. We chose here to use commands in the SIT since we then very easy could load the true system into ident

```
% be an ordinary apostrophes.
%See help idinput for how the blips should look like.

%Simulate the system. DO help idmodel/sim
y=sim(Mtrue,[u e]);
%
% Open ident and load the data (y and u). Also import the true system
% Mtrue (Go to Models/Import in Ident). Then, start estimating
%(ARX) models and compare the estimates, for example
% in the frequency domain, with the true system Mtrue
% Of course, in practice Mtrue is unknown. But for the purpose of illustratin
% concept it may be good to know Mtrue!
```

4 Some other demos - Optional

You may not have time during the lab excercise to go through the two examples below, but it may be instructive to do this exercises at you own at a later stage!

1. Select the demonstrations *Compare different identification methods* and notice how different identification methods can be used. Neglect the part with the "pem" and "iv"methods (not included in the course). After the pem, ARX modelling is covered which should be studied. Also learn how the models could be evaluated in various ways (poles and zeros, bode diagram, comparison with spectral analyses, and Nyquist plots)
2. Select demo *Build simple models from real laboratory process data* where data from a hair dryer is used. Check how different methods can be used and compared, including simple methods like correlation analyses (neglect the part with pem, which is not covered in the course)

5 Further aspects on model validation - Optional

This exercise gives initial training on how to validate a model. Note that the used tests also can be found on the Ident GUI.

1. In this task (it can be solved by running the m-file `compl2a`, see below) we will use the least squares method, LSM, for identifying a system given by

   $$(1 - 1.5q^{-1} + 0.7q^{-2})y(t) = (1.0q^{-1} + 0.5q^{-2})u(t) + (1 - 1.0q^{-1} + 0.2q^{-2})e(t)$$

   The input signal is binary white noise ($u(t) = \pm 1$), independent of the white noise $e(t)$, which has zero mean and variance $\lambda^2 = 1$.

   The system is simulated using 250 data points. An ARX model of order n is estimated using the least squares method. Different values of n ($n = 1, \ldots, 6$) are tried. Some various methods The following methods for model validation is presented in order to decide what model (order) to accept:

   (a) Plot of the residuals $\varepsilon(t)$. Do they look as an uncorrelated sequence?

   (b) Plot of the estimated auto- and cross-correlation function $\hat{r}_\varepsilon$ and $\hat{r}_{\varepsilon u}$.

(c) Plot the loss function $V_N(\hat{\theta})$, the AIC and FPE. All as a function of the model order.

(d) Plot of poles and zeros of the models. If a pole–zero cancellation (almost) occurs, it is a sign that the model order has been chosen unnecessarily high.

The task can be carried out by running the following Matlab code. It is available as the file `compl2a`.

Discuss what model order the different criteria gave.

Answer: ____

____

____

____

____

Remark: It could be of interest to also study the Bode diagram of the different estimate.

```
% file compl2a.m ******************
%Computer Laboratory 2
%System Identification
%TS 920315, 950405
%rev EKL 040525
clear
clf
format compact

% preparation exercise
np=250;
a0=[1 -1.5 0.7]; b0=[0 1 .5]; c0=[1 -1.0 0.2];
s2=1; l2=1;
u=sign(randn(2*np,1))*sqrt(s2); e=randn(2*np,1);

th=idpoly(a0,b0,c0,1,1,l2);
y=idsim([u e],th);
uv=u(np+1:end);ev=e(np+1:end);yv=y(np+1:end);
u=u(1:np);e=e(1:np);y=y(1:np);
ze=iddata(y,u);zv=iddata(yv,uv);
%Use of LSM

nlag=10;resl=[]; plsm=[]; zlsm=[];
```

```
np=length(y);
nmax=6;
for n=1:nmax,
  that1=arx(ze,[n n 1]);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Change the above command to use a prediction error method %
%                                                           %
% that1=armax(ze,[n n n 1]);                                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 eval(['TH' num2str(n) '=that1;']); % to save that1 for n=1:nmax
 %present(that1);
 [comp1(n),fit1(n)]=compare(ze,that1);
 [comp2(n),fit2(n)]=compare(zv,that1);
 V(n)=that1.EstimationInfo.LossFcn;
 FPE(n)=that1.EstimationInfo.FPE;
 AIC(n)=log(V(n))*np+2*2*n;
 % Modify the AIC for ARMAX because we have n more free variables!
 % i.e. AIC(n)=log(V(n))*np+3*2*n;
 format short
end;


echo on
%b) Testing based on the covariance functions of the residuals
echo off
for n=1:nmax
 eval(['resid(TH' num2str(n) ',ze);']);
 subplot(211)
 txt=['autocorrelation of residual e, n=',num2str(n)];
 title(txt)
 subplot(212)
 txt=['crosscorrelation between e and u, n=',num2str(n)];
 title(txt)
 pause
end;

echo on
%c)  measured output and model output
%    for the estimation data set.
echo off
mm=min([100 np]);
for n=1:nmax
  subplot(320+n);
  plot(1:mm,[comp1{n}.y(1:mm),y(1:mm)]);
  title(['n = ',num2str(n),' Fit :',num2str(fit1(n))]);
  legend('y_m','y',0)
end;
pause
echo on
```

```
%   measured output and model output
%   for the validation data set.
echo off
for n=1:nmax
  subplot(320+n);
  plot(1:mm,[comp2{n}.y(1:mm) ,yv(1:mm)]);
  title(['n = ',num2str(n),' Fit :',num2str(fit2(n))]);
  legend('y_m','y',0)
end;
pause
echo on

%d) loss function, AIC and FPE
echo off
clf
subplot(311)
plot(1:nmax,V), title('loss function'), xlabel('n')
subplot(312)
plot(1:nmax,AIC), title('AIC'), xlabel('n')
subplot(313)
plot(1:nmax,FPE), title('FPE'), xlabel('n')
pause

clf
echo on
%e) pole-zero plot with confidens regions
%   corresponding to 3 standard deviations.
echo off
for n=1:nmax
  eval(['zpplot(TH' num2str(n) ',3);']);
  title(['Poles and zeros ARX, n = ',num2str(n)])
   pause
end
```

2. Repeat the previous task but use now the prediction error method applied with an ARMAX model of order n. Use the same data set as before. Try some values of n, say $n = 1, \dots, 4$. Apply the same validation procedures as before.

   It may be convenient to edit the file `compl2a` to solve the task. Note that an ARMAX model (with one delay between u and y) can be estimated with `theta=armax([u y],[n n n 1])`.

   What are your findings?

Answer: ____

____

____

____

____