UPPSALA UNIVERSITET
AVDELNINGEN FÖR SYSTEMTEKNIK
BC 9901 (Adapted from SI-lab 4), Last rev January 24, 2008

Computer exercise 4 for the course:

- Empirical Modelling

Some practical aspects

Preparation exercises:
1. Read the lab instructions carefully.
2. Read Sections 12.4 and 14.3 (including the example 14.2)
3. Check the ideas behind Grey-box modelling ("skräddarsytt"), see Section 14.4 and Example 14.3

Name	Assistant's comments
Program Year of reg.	
Date	

Contents

1 Goals

In this computer lab you will investigate some various problems, namely:

- The effect on non zero means in the data. The understanding of this problem is very important in the project work!
- System identification as a way of model approximation when the model structure is not rich enough to describe the true dynamics. In particular, the use of *prefiltering of the data* in order to get a good model accuracy in a certain frequency region will be studied.
- Estimation of physical parameters.

2 Effect of non zero means

In this first task the aim is to illustrate the effect on non zero means both on the input and output signal.

The least squares method, LSM, is used for identifying a system given by

$$\begin{aligned}(1-0.8q^{-1})y_o(t) &= 1.0q^{-1}u(t)+e(t)\\ y(t) &= y_o(t)+m\end{aligned}$$

The input signal is white noise (plus an offset, see point 3 below) independent of the white noise $e(t)$, which has zero mean and a small variance $\lambda^2 = 0.01^2$. The parameter m is used to describe a non zero mean value (offset) of y which is not due to the input.

The system is simulated using 100 data points. An ARX model of order 2 is estimated using the least squares method. Consider the following cases

1. Both the input and output have zero means (i.e. $m = 0$)
2. The input has zero means but $m = 10$.
3. The input has mean value 10 and $m = 0$ (due to the input, the output will then also have a mean value different from zero).

Simulate the above cases. The file *dctest.m* printed below solves case 1. For case 2, you need to add a the term 10 to y and repeat the estimation. For case 3 you need to add 10 to u (but not to y) and simulate the system before making the estimation. You need to edit the file accordingly.

```
u=randn(100,1); %Generate input signal
lambda=0.01;  %Low noise level!
e=randn(100,1)*lambda; %Generate noise
B=[0 1]; A=[1 -0.8];  %System parameters
y=filter(B,A,u)+filter(1,A,e); % Simulate ARX system
th=arx([y u],[1 1 1]); %Estimate ARX model of correct order
present(th);
```

Summarize your findings below. Did any case (except 1) gives a correct model?

Answer: ______________________________

____

____

____

____

Give two approaches which can be used when you have data with non-zero mean values which may cause bias in the estimated model parameters.

Answer: ______________________________

____

____

____

____

Consider the following two modelling problems:

- Estimation of a dynamic model from a system where it is known that $u = 0$ leads to $y = 0$. This is the case when the output is only affected by the input and the sensor measuring the output is not "biased"
- Estimation of a dynamic model from a system where it is known that $u = 0$ not yields $y = 0$. This is the case when the output is affected by a constant disturbance (for example, in the project you will not obtain a zero nitrate level when the external carbon flow rate is zero) or when the sensor measuring the output is "biased".

Which one of the above cases must be treated with care in the identification in order not to give a biased estimate?

Answer: ______________________________

____

____

3 The use of prefiltering for weighting of accuracy in frequency domain

In this task we will examine how system identification can be viewed as a form of model approximation, when the system dynamics is too complex to belong the model structure considered. To simplify the study we consider a noise–free situation. Consider the following system, which has two distinct resonances.

$$G(q^{-1}) = \frac{1.0q^{-2} - 1.3q^{-3} + 0.8q^{-4}}{(1 - 1.5q^{-1} + 0.9q^{-2})(1 + 0.0q^{-1} + 0.9q^{-2})}$$

Simulate the system using the input $u(t)$ as white binary noise of zero mean and unit amplitude. Generate 100 data points. Use the function `gendata3` listed below.

```
%file gendata3.m
%Computer Laboratory 3
% Model approximation -- frequency domain effects
% Generating the data

num=[0 0 1 -1.3 0.8];
den=conv([1 -1.5 0.9],[1 0 0.9]);
[msys,psys,w]=dbode(num,den,1);
u=sign(randn(100,1));
y=filter(num,den,u);
z=[y u]; %This is for use in the graphical interface ident
```

Next we will study the estimation of second order ARMAX model using prefiltered data. The following items will be studied.

1. The data will be filtered with

$$\begin{aligned} u^F(t) &= F(q^{-1})u(t) \\ y^F(t) &= F(q^{-1})y(t) \end{aligned}$$

The filtering has the effect that we give emphasis to certain frequency ranges, depending on the choice of the filter. Let F be a sharp bandpass filter around one of the resonance frequencies of the true system. Use a 5'th order Butterworth filter. Do `help butter` in order to find out the syntax[1] for this command.

A reasonable starting filter is obtained by the command

```
[nn,dd]=butter(5,[0.15 0.25]);
```

[1]Note in particular that the cut off frequency for the filter is normalized between 0 and 1, where 1 corresponds to the maximum frequency which in discret time is equal to π (for a sampled signal the physical correspondence is half the sampling rate). The following example illustrates the case when we want to low pass filter a sampled continuous time signal: Assume that the sampling frequency $fs = 1000$ Hz and that we want to remove frequencies above 300 Hz. That is, the band with of the filter should (in continuous time) be 300 Hz . The value $Wn = 1$ in the filter corresponds to $fs/2 = 500$ Hz. Hence we should chose $Wn = 300/500 = 0.6$ in the filter design.

The prediction error method using the *filtered* data is then applied. The result is evaluated in the frequency domain by drawing Bode plots of the model and the true dynamics.

2. Repeat the previous subtask but for a filter which emphasizes the other resonance frequency.

3. Repeat the previous subtask but let F be a low pass filter.

The estimation part of the task can be carried out by running the following Matlab code. It is available as the file `compl3e`.

Note that you need to run `butter` first in order to get filter parameters.
We strongly recommend (good for the project work!) that you also try to solve the task (or part of it) using the interface `ident`. Note that in `ident` you can also do prefiltering (but where the filter bandwidth is between 0 and π)
Summarize your findings below.

Answer: ____

____

____

____

____

```
% Model approximation -- frequency domain effects
clf

%Estimation based on filtered data,
%Before the routine can be used, filter
%parameters must be calculated.
% Use for example a 5th order Butterworth filter.
% See help butter (but be happy!)

clg

filtnum=input('Give filtnumerator with [] ')
filtden=input('Give filtdenominator with [] ')

yf=filter(filtnum,filtden,y);
uf=filter(filtnum,filtden,u);
that1=armax([yf uf],[2 2 2 1])
```

```
[a1,b1,c1]=polydata(that1)
[m1,p1]=dbode(b1,a1,1,w);
[mf,pf]=dbode(filtnum,filtden,1,w);

subplot(121)

loglog(w,m1,w,msys,'--',w,mf,':')
axis([0.1,pi,0.1,10]);
title('Frequency functions');
xlabel('Angular frequency'); ylabel('Amplitude');
txt='filtnum: ';
for j=1:length(filtnum);
 txt=[txt,num2str(filtnum(j)),' '];
end
text('Units', 'normalized');
text(0.20,0.25,txt,'sc')
txt='filtden: ';
for j=1:length(filtden);
 txt=[txt,num2str(filtden(j)),' '];
end
text(0.20,0.15,txt,'sc')

subplot(122);
semilogx(w,p1,w,psys,'--'); hold off;
title('Frequency functions');
ylabel('Phase'),xlabel('Angular frequency')
```

4 Estimation of physical parameters

As an example/illustration we will consider estimation the dynamics of a lake/reservoir which net flow can be adjusted by some flood gates[2]. The example is very simple but illustrate the general procedure on how to first use physical modelling and then apply system identification to a model with hopefully few unknown parameters.

Consider a reservoir with a water level of h (assumed spatially constant), and an area of A. By a simple mass balance we have that

$$A\frac{dh}{dt} = q$$

where q is the net flow through the reservoir. In the Laplace domain we have

$$H(s) = \frac{1}{As}Q(s)$$

Lets assume that the net flow can be adjusted by a flood gate with the following

[2] A remark: In the last years, we have had two MSc theses dealing with modelling and control of irrigation water channels. The work were conducted in Univ of Melbourne Australia..

dynamics

$$Q(s) = \frac{k_{gate}}{1+sT}U(s)$$

where $U(s)$ is the Laplace transformation of the control signal to the gate. The gate dynamic is hence a a fist order system with gain k_{gate} and time constant T.

The (continuous-time) transfer function from control signal to water level is given by

$$G(s) = \frac{K}{s(1+sT)}$$

where $K = k_{gate}/A$ and T are the parameters to be determined. In this example, it is trivial to see that the gate gain and reservoir area can not be determined independently. In the following we will study the estimation of K and T. If we know K and T (from the procedure described below) but wants the area A we must determine k_{gate}. Describe a simple experiment how to determine k_{gate}. It is assumed that during the experiment Q can be measured.

Answer: ____

____

____

By choosing the level and net flow as state variables, we can represent the reservoir dynamics in state space form as

$$\begin{aligned} \dot{x} &= \begin{pmatrix} 0 & 1 \\ 0 & -1/T \end{pmatrix} x + \begin{pmatrix} 0 \\ K/T \end{pmatrix} u \\ y &= (1 \;\; 0)x \end{aligned}$$

We assume that a white noise disturbance is affecting the measurements of the reservoir level.
A discrete time model for the reservoir is given by

$$\begin{aligned} x(t+h) &= A_d x(t) + B_d u(t) \\ y(t) &= Cx(t) + e(t) \end{aligned}$$

In order to *generate* the data for the identification experiment we will use the following parameter values:

$$K = 4, \quad T = 0.5, \quad h = 0.1 \text{timeunits}$$

The input signal $u(t)$ is a square wave. The amplitude of the square wave is 0.1, its period is 8.0, and the total duration of the simulation is 80. The measurement noise has standard deviation $\lambda = 0.1$. Generate the data by running the file `fysmod`.

```
%file fysmod.m
%Computer Laboratory W4, Lab3
```

```
%TS last rev 950405, Adapted for W4 by BC 990211

clear,clf
format compact

%Generate data for lake modelling
%Set parameters

K=4
T=0.5
ramp=1;
lambda=0.1;
a=0.5
h=0.1
tper=8
ttot=80

%Define model
A=[0 1;0 -1/T],B=[0;K/T],C=[1 0], D=0,
[Ad,Bd]=c2d(A,B,h)

%Generate input signal
time=0:h:ttot;
np=ttot/h;
rr=kron(ones(ttot/tper/2,1),[1;-1]);
u=kron(rr,ones(tper/h,1));
u=[1;u(:)]*.1;
% Simulate system
[y0,x]=dlsim(Ad,Bd,C,D,u);
e=randn(np+1,1)*lambda;
y=y0+e;
plot(time,[y,u]),title('output and input')
```

We shall now see how system identification can be used to estimate physical parameters. Our interest is now to estimate the 'physical parameters' T (the time constant) and K from the data.

One approach would be to first identify a discrete-time (black box) transfer function of the process. A typical model order choice is of second order, and has 4 parameters. Hence, there is no unique way to derive K and T from the estimated model. It should also be clear that even if the true system has a pure integrator, the estimated model will have a pole that is only close to integrating (it will lie close to $z = 1$ in discrete time).

We will instead consider estimation of the parameters K and T directly from data by means of identification using the prediction error method. The model will not be of black box type, but have some internal structure, and a few 'physical parameters'. These parameters will be estimated by using the prediction error method, applied to the parameterized model. The general PEM approach still holds, but the implementation of how the loss function and its gradient are computed becomes more messy. Fortunately for the user, the System Identification Toolbox provides appropriate facilities for handling such problems.

We will here consider the case when there is no coupling or constraint between the model parameters. In this case we will consider the parameterization

$$\begin{aligned} \dot{x} &= \begin{pmatrix} 0 & 1 \\ 0 & \theta_1 \end{pmatrix} x + \begin{pmatrix} 0 \\ \theta_2 \end{pmatrix} u \\ y(t) &= \begin{pmatrix} 1 & 0 \end{pmatrix} x(t) + e(t) \end{aligned}$$

After we have identified the parameter vector θ we can easily find the physical parameters as

$$T = -1/\theta_1 \quad K = -\theta_2/\theta_1$$

Some Matlab code to run the estimation part is provided in the file `compl3d`. See also iddemo nr 8 and help for the commands `modstruc` and `ms2th`. First some statements for defining how the parameters enter the state space matrices are included. The vector `thguess` contains the initial guesses of the parameter values.

```
%file compl3d.m
%Computer Laboratory 3
%System Identification
%TS last rev 950405, rev 990211 by BC, 040606 by EKL

%Estimation of physical parameters

%Define model structure
% Initial guess
Ai=[0 1;0 -1],Bi=[0; 1],Ci=[1 0],Di=0,
Ki=[0;0],xi=[0;0]
% Free parameters as NaN
Aest=[0 1;0 NaN],Best=[0; NaN],Cest=[1 0],Dest=0,
Kest=[0;0],xest=[0;0]

ms2=idss(Ai,Bi,Ci,Di,Ki,xi,'Ts',0)
setstruc(ms2,Aest,Best,Cest,Dest,Kest,xest);

%Estimation
z = iddata(y,u,0.1);
thest=pem(z,ms2)
Test=-1/thest.A(2,2)
Kest=-thest.B(2)/thest.A(2,2)
```

Run the file above and summarise your findings below.

Answer: ____

____

____