Uppsala University
Information Technology
Dept. of Systems and Control
Torsten Söderström 1992
Revised JS
Revised EKL 0101

System Identification

Computer exercise 2

Introduction to System Identification

Preparation exercises:

1. Study Chapters 2 and 3 of System Identification.
   Also get familiar with the MATLAB code given in the lab.
2. Solve preparation exercises 1-3.

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

1 Goals

The aim of this computer laboratory is to introduce the System Identification Toolbox, and to illustrate non-parametric identification as well as the least squares method on some data sets.

2 Tasks

We will in what follows consider two different systems, namely

$$\mathcal{S}_1: \quad A(q^{-1})y(t) = B(q^{-1})u(t) + e(t)$$

$$\mathcal{S}_2: \quad \begin{aligned} A(q^{-1})x(t) &= B(q^{-1})u(t) \\ y(t) &= x(t) + e(t) \end{aligned}$$

In both cases $e(t)$ is assumed to be white Gaussian noise, independent of the input $u(t)$, and of zero mean and variance λ^2. Note that the noise enters differently in the two systems. In $\mathcal{S}_1$ it enters as process noise, while in $\mathcal{S}_2$ it appears as pure white measurement noise.

In what follows we will let the system parameters be given by

$$A(q^{-1}) = 1 - 0.8q^{-1} \qquad B(q^{-1}) = 1.0q^{-1} \qquad \lambda^2 = 1$$

for both systems. Note that in the simulations below, 100 experiments are conducted. This is to illustrate the biases (systematic error) and variances of different methods. From one single realization it may be hard to know if the results are due to bad, or good "luck".

2.1 Non-parametric Identification

Non-parametric identification schemes are characterized by the property that the resulting models are curves of functions, not necessarily parametrized by a finite-dimensional parameter vector. In this section we will study three different non-parametric methods, namely, transient analysis, correlation analysis and spectral analysis.

Preparation exercise 1. Under what experimental conditions will the correlation analysis method work well (e.g. easy to apply)?

Answer:

Preparation exercise 2. What is the expected effect of the window factor M of the spectral analysis method (Task 4)?

Answer:

1. We begin by illustrating transient analysis. The system $\mathcal{S}_1$ with the input $u(t)$ being a step ($u(t) = 1$ for $0 \leq t \leq 100$) is simulated and the response is plotted. The task can be done using the following MATLAB code, that is available as file `lab2a`.

```
%file lab2a.m
%Computer Laboratory 1 System Identification
%TS 920315. rev TS 950405. last rev KB 990201

% Transient analysis
clear
a=-0.8;b=1;A=[1 a];B=[0 b];lambda=1;t=10;N=100;

e=randn(N,100)*lambda; u=[zeros(9,1);ones(91,1)];
y1=filter(B,A,u); y=y1*ones(1,100)+filter(1,A,e);
ym=mean(y')'; yvar=std(y')';

subplot(211)
plot([1:100],[y(:,1),y1])
axis([0 100 -2 8])
title('Step responses')
legend('output y','noise free output',4)

subplot(212)
plot([1:100],[ym,y1],[1:100],[ym+yvar,ym-yvar],'r--')
axis([0 100 -2 8])
title('Step responses, averaged over 100 simulations')
legend('mean output y','noise free output','standard deviation',4)
```

Discuss briefly benefits and drawbacks with the step response analysis. Is it easy to determine the system dynamics (*e.g.*, time constants, static gain and resonance frequencies) from the step response?

Answer:

2. Next we consider correlation analysis. Let the input $u(t)$ be white binary noise of length $N = 100$, taking the values ± 1. Then use correlation analysis to estimate the weighting function of the system for lags $k = 1, \ldots 20$. Compare with the true values. The task can be done using the following MATLAB code, that is available as file `lab2b`.

```
%file lab2b.m
%Computer Laboratory 1 in System Identification
%TS 920315. rev TS 950405. last rev KB 990201

% Correlation analysis
clf, clear h hest
m=20, u=sign(randn(N,100));
y=filter(B,A,u)+filter(1,A,e);
for k=1:m, hest(k,1:100)=diag((y(k+1:N,:))'*u(1:N-k,:))'/N;end
for k=1:m, h(k,1)=b*(-a)^(k-1);end
h_mean=mean(hest')';
h_var=std(hest')';

subplot(211)
plot([1:20],[hest(:,1),h])
title('Correlation analysis: weighting functions')
xlabel('time lag')
legend('Estimated','True')

subplot(212)
plot([1:20],[h_mean(:,1),h],[1:20],[h_mean+h_var,h_mean-h_var],'r--')
title('Averaged over 100 simulations')
xlabel('time lag')
legend('Mean of estimates','True','Standard deviation')
```

Discuss briefly the results from the correlation analysis.

Answer:

3. Now repeat the previous task but for another input with a more low-frequency character. Use the input signal

$$u_2(t) = u_0 \frac{1}{1-0.8q^{-1}} u(t), \qquad u_0 \triangleq \sqrt{1-0.8^2}$$

where the normalization u_0 will assure that $u_2(t)$ have the same variance as $u(t)$. Do not take any particular action in the identification algorithm though. The task can be done using the MATLAB macro `lab2c`. This file is the same as `lab2b`, except that we now use the input $u_2(t)$

```
u2=filter(1,[1 -.8],u)*sqrt(1-.8^2);
```

It is clear from the figure that the estimate $\hat{h}(k)$ in this case is severely biased. Why is this so? (Hint: What is implicitly assumed when $\hat{h}(k)$ is estimated in the macro `lab2b`? Compare with pages 42-43 in system identification.)

Answer:

Remark: The bias problem above can be somewhat cured. One alternative is to pre-whiten the input before applying the estimation scheme. This is implemented in the MATLAB function **cra**. A second alternative is to solve the *Wiener-Hopf equation* by assuming a truncated weighting function, see system identification page 43. This method is implemented in the MATLAB macro `lab2c_2`. Run the macro to see that an "unbiased" estimate is obtained.

4. Next we will use data from the previous exercise and apply spectral analysis. The resulting model in form of Bode plots are compared with the frequency functions of the true system. The spectral analysis command `spa` is implemented with a Hamming lag window of length M. Try various values of M.

   The task can be done using the following MATLAB code, that is available as file `lab2d`.

```
%file lab2d.m
%Computer Laboratory 1 in System Identification
%TS 920315. rev MC 950109. last rev KB 000201

% Spectral analysis
M=input('Give M '), clf

w=logspace(-2,pi);
for k=1:100,
  z=[y(:,k),u2(:,k)];
  gest=spa(z,M,w);
  [tmp1 tmp2]=bode(gest);
  mest(1:50,k)=squeeze(tmp1);pest(1:50,k)=squeeze(tmp2);
end;
m_mean=mean(mest')'; p_mean=mean(pest')';
m_var=std(mest')'; p_var=std(pest')';
[mtrue,ptrue]=dbode(B,A,1,w);

figure,clf
subplot(211)
loglog(w,[mest(:,1),mtrue,m_mean]), hold on
loglog(w,[m_mean+m_var,m_mean-m_var],'r--'), hold off
xlabel('Angular frequency'),ylabel('Amplitude')
txt=['Spectral analysis, M = ', num2str(M)];
title(txt)
legend('Estimate (one example)','True','Mean and std of 100 simulations',3)
subplot(212)
semilogx(w,[pest(:,1),ptrue,p_mean]), hold on
semilogx(w,[p_mean+p_var,p_mean-p_var],'r--'), hold off
```

```
xlabel('Angular frequency'),ylabel('Phase')
title(txt);
legend('Estimate (one example)','True','Mean and std of 100
simulations',3)
```

Discuss how M affects the estimate, what is a reasonable value of M? Moreover, how is the estimate affected by the fact that we use a low frequent input $u_2(t)$? Would there be any difference if we used a white noise input $u(t)$ instead ?

Answer:

2.2 Parametric Identification using the Least Squares Method

We now turn to the least squares method, which in contrast to the previous methods gives a parametric model. The LS method is applicable to models of the form

$$A(q^{-1})y(t) = B(q^{-1})u(t) + \varepsilon(t) \tag{2.1}$$

which equivalently can be expressed as the linear regression model

$$y(t) = \varphi^T(t)\theta + \varepsilon(t) \tag{2.2}$$

where

$$\varphi^T(t) = [-y(t-1) \cdots - y(t-n_a) \quad u(t-1) \cdots u(t-n_b)] \tag{2.3}$$

$$\theta = [a_1 \cdots a_{n_a} \quad b_1 \cdots b_{n_b}] \tag{2.4}$$

Under some weak assumptions the estimate is obtained as

$$\hat{\theta} = [\frac{1}{N}\sum_{t=1}^{N} \varphi(t)\varphi^T(t)]^{-1} \frac{1}{N}\sum_{t=1}^{N} \varphi(t)y(t) \tag{2.5}$$

Preparation exercise 3. Find the theoretical limits (as $N \to \infty$) of the least-squares parameter estimates for the systems $\mathcal{S}_1$ and $\mathcal{S}_2$ assuming the input $u(t)$ to be white with zero mean and unit variance.

Answer:

In the first experiment we will use the same data (system $\mathcal{S}_1$ and input $u_2(t)$) as before, and the model structure

$$(1 + aq^{-1})y(t) = bu(t-1) + \varepsilon(t)$$

$$\theta = (a \quad b)^T$$

Compute the estimates and compare with the true parameter values. The task can be solved using the file `lab2e`, see the print-out below.

Answer:

Also illustrate the obtained model by computing and plotting the frequency function. What are your findings?

Answer:

```
%file lab2e.m
%Computer Laboratory 2 in System Identification
%TS 920315, rev TS 950405, rev KB 990201
%rev EKL 040525
% Least Squares Method

clear
a=-0.8;,b=1;,A=[1 a];,B=[0 b];,lambda=1;N=100;

e=randn(N,100)*lambda;
u=sign(randn(N,100));
u2=filter(1,[1 a],u)*sqrt(1-a^2);
y=filter(B,A,u2)+filter(1,A,e);
w=logspace(-2,pi);
for k=1:100,
  z=[y(:,k),u2(:,k)];
  th=arx(z,[1 1 1]);
  [aest,best]=polydata(th);
  Aest(k,1:2)=aest; Best(k,1:2)=best;
  [mest(1:50,k),pest(1:50,k)]=dbode(best,aest,1,w);
end;
m_mean=mean(mest')'; p_mean=mean(pest')';
m_var=std(mest')'; p_var=std(pest')';
A_mean=mean(Aest), B_mean=mean(Best),
```

```
A_var=std(Aest), B_var=std(Best),
[mtrue,ptrue]=dbode(B,A,1,w);
subplot(211)
loglog(w,mtrue,w,m_mean,'r'), hold on
loglog(w,[m_mean+m_var,m_mean-m_var],'r--'), hold off
txt=['LS method  Sys 1   Bode plots   N=',num2str(N)];
title(txt),xlabel('Angular frequency'),ylabel('Amplitude')
legend('True','Mean and std of 100 simulations',3)
subplot(212)
semilogx(w,ptrue,w,p_mean,'r']), hold on
semilogx(w,[p_mean+p_var,p_mean-p_var],'r--'), hold off
title(txt),xlabel('Angular frequency'),ylabel('Phase')
legend('True','Mean and std of 100 simulations',3)
```

2.2.1 Low Freq. input /White Noise input for the System $\mathcal{S}_1$

Next we will see how the input signal can influence the estimate. We will feed the system $\mathcal{S}_1$ with two different input signals:

- A white noise sequence. The input $u_1(t)$ is in this case white binary noise of length $N = 100$, taking the values ± 1.
- A low-frequent character sequence. The input $u_2(t)$ is a low pass filtered version of $u_1(t)$ according to

$$u_2(t) = u_0 \frac{1}{1 - 0.8q^{-1}} u_1(t), \qquad u_0 \triangleq \sqrt{1 - 0.8^2}$$

  where the normalization u_0 will assure that $u_2(t)$ have the same variance as $u_1(t)$.

Note that these two inputs are the same as have been used before. We will use the same model structure as in the previous task, and compute the estimate by using the least squares method. The results will be presented as frequency functions, by means of bode plots. The exercise can be done using the MATLAB macro `lab2f`. This file has the same basic structure as `lab2e`, so a print-out is therefore omitted.

Are the estimates consistent for both the inputs? More generally, in order to get consistent estimates the input has to fulfill a certain condition, namely? (Hint: See Remark 2 on page 121 of system identification)

Answer:

By looking in the frequency domain, compare the estimates for the two different inputs. In what frequency range are the estimates most accurate? Is there any difference for the two different inputs? If so, why?

Answer:

2.2.2 Low Frequent input / White Noise input for the System $\mathcal{S}_2$

As a final exercise we will repeat the previous task but with the system $\mathcal{S}_2$ instead of the system $\mathcal{S}_1$. The model structure is, however, still the same. Compute the parameter estimates, and compare with the true values. Illustrate the results by means of frequency functions. This task can be done using the MATLAB macro `lab2g`. This file has the same basic structure as `lab2e`, hence a print-out is omitted.

By comparing with the previous task, what are your findings?

Answer: