FRT 041 System Identification
Laboratory Exercise 1

Michael Lundh
Jonas Eborn
Rolf Johansson
Mikael Petersson
Anders Robertsson
Magnus Gäfvert
Revised: Johan Bengtsson

Department of Automatic Control
Lund Institute of Technology
Lund University
January 2002

Revised: March 2005

1. Introduction

The purpose of this laboratory experiment is to determine the frequency response of a linear process. The frequency response is determined to get insight in the process dynamics and is to be used for controller design. The process will finally be controlled.

1.1 Equipment

The process is a beam mounted on a motor shaft, see Figure 1.1. The process input is a reference signal to an internal velocity controller and the output is the beam angle. There are some mechanical resonances in the beam. A PC running Matlab/Simulink will be used both for the initial Frequency Response Analysis and for the latter design and control. The real-time control is carried out in Simulink using locally developed I/O-blocks.

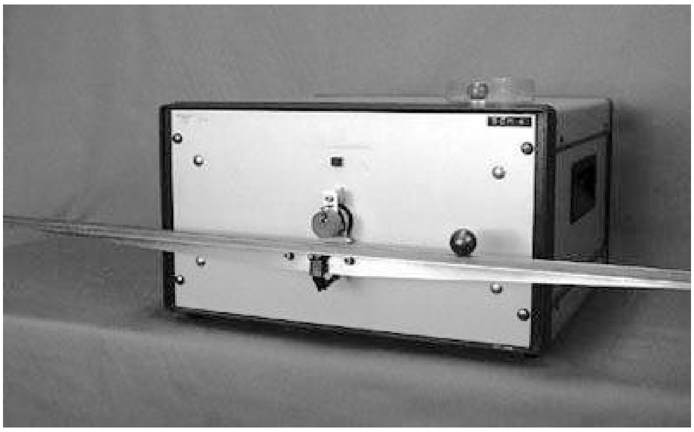

Figure 1 The beam dynamics should be identified in the laboratory exercise.

1.2 Preparations

Read Chapter 2 in [Johansson, 1993]. Solve exercises marked with *Preparation* in this guide. Study section 6.6 on compensation in [Åström, 1968] or Sections 5.4-5.5 in [Glad and Ljung, 1989]. Solutions should be presented before the experiments start, and some questions concerning the theory should be answered.

2. The Method

The system is forced by a sinusoidal input signal $u(t) = u_0 \sin(\omega t)$ where the u_0 and ω are to be chosen. When transients have decayed, the output signal is described by $y(t) = |G(i\omega)|u_0 \sin(\omega t + \arg G(i\omega))$. The experiment is repeated for a number of frequencies ω so that a Bode-diagram may be drawn.

Direct, accurate measurement of the amplitudes of u and y and their phase lag is difficult. A better method is to integrate for a certain time T

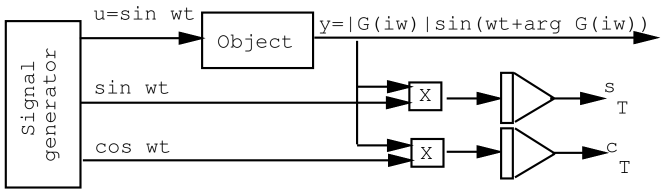

Figure 2 Schema of the Frequency Response Analysis method.

to get

$$Y_s(T) = \int_0^T y(t)\sin(\omega t)dt$$
$$Y_c(T) = \int_0^T y(t)\cos(\omega t)dt$$

for each frequency ω. If the experiment time is chosen to be a multiple of the period of u, i.e. $T = n \cdot 2\pi/\omega$ we will have

$$Y_s(T) = \frac{T}{2}u_0 \mathrm{Re}\ G(i\omega)$$
$$Y_c(T) = \frac{T}{2}u_0 \mathrm{Im}\ G(i\omega)$$

From these quantities the magnitude and the phase shift of $G(i\omega)$ is computed.

$$|G(i\omega)| = \frac{2}{Tu_0}\sqrt{Y_s^2(T) + Y_c^2(T)}$$
$$\arg G(i\omega) = \arctan \frac{Y_c(T)}{Y_s(T)}$$

The correlation method can be viewed as filtering $y(t)$ through a band-pass filter with center frequency ω and bandwidth proportional to $1/T$.

3. Experiment Setup

The frequency response analyzer described above is implemented in Matlab/Simulink and performs the calculations automatically. But first some experiment conditions must be determined. The most important ones are discussed here.

3.1 Frequency Range

Determine the frequency interval where it is essential to know the process. To do this some a priori knowledge is required, maybe from a previous experiment. A reasonable range for the beam is the interval $[0.5, 300]$ rad/s.

3.2 Amplitude and Bias

Determine the amplitude u_0 of the input signal. Select the amplitude so that the system remains linear, i.e. signals must in general be sufficiently small. The frequency response analysis may be repeated for different amplitudes to investigate linearity. For the beam let $u_0 \in [0.5, 1.0]$ V.

Since the process is not asymptotically stable (a constant input, u, gives a constant angular velocity and increasing angle) you may need to add a bias term to the input signal, $u(t) = u_b + u_0 \sin(\omega t)$. This is necessary for the beam to remain in approximately the same position during long experiments.

3.3 Measurement Time

The noise level determines the measurement time required to obtain desired accuracy. A long measurement time implies low noise sensitivity. In these experiments we chose the measurement time to be multiple of periods of the forcing signal.

Preparation: How is accuracy related to measurement time if the measurement of y is corrupted by almost white noise?

3.4 Delay

Determine how long to wait before starting measuring. It is the time from forcing with a certain frequency until integration begins. This time depends on how fast the transients decay in the system. Transients of the beam decay in a few seconds.

4. Experiment

In a real situation little is initially known about the process. The identification procedure will therefore be iterative. A new experiment is planned using experience from previous ones. You were given some hints in the last section.

You should do identification experiments until you have such knowledge of $G(i\omega)$ that you are able to plot a Bode-diagram without strange discontinuities. This requires high frequency resolution in some intervals.

Investigate also the linearity of the process, by comparing two identical experiments, except for different signal amplitudes.

The experiments are performed in open loop despite the fact that the process is not asymptotically stable. It is however stable. The frequency response analysis will work since the drift is small.

4.1 Connections

The switch Man/Auto on the backside of the process should be in **Auto** and the following connections should be made:

PC	Beam process
Analog Out 0	IN
Analog In 0	angle
ground	ground

4.2 Files

Log on to your account at the *efd-domain* and create a directory for the lab exercise. Download the Matlab-files from the course webpage to this directory.

4.3 Matlab Session

Frequency Response Analysis Start Matlab and and type `initfra` at the command prompt. This sets up Matlab and opens the Simulink model `solatron`, see Figure 3. The matlab command `fra` performs an experiment for each frequency specified in the `omega`-vector.

```
function EstVec=fra(omega,Amplitude,Bias,NrOfPeriods,SettlingTime);
% EstVec=fra(omega,Amplitude,Bias,NrOfPeriods,SettlingTime);
%
%  Returns an estimate of the transfer function at the frequencies of
%  'omega':  EstVec=[ omega ; G_hat(iw) ];
%
%   omega        -- frequency vector
%   Amplitude    -- the amplitude [V] for the  sinus wave sent to
%                   the process.
%   Bias         -- bias term to compensate for drift of the process
%   NrOfPeriods  -- Nr of periods for the measurements
%   SettlingTime -- Settling time for transients before measuring
```

Two identification results can be concatenated to one estimated model by the command `sortfqs`.

```
%----- Example ----------
>> w10_110 = 10:20:110
>> Amplitude=1.0;    %
>> Bias=0;           % may need to change!
>> G1=fra(w10_110,Amplitude,Bias);  %  EstVec = [ omega ; G_hat(iw) ];

>> w1_200 = logspace(log10(1),log10(200),10);
>> G2=fra(w1_200,Amplitude,Bias)

% concatenate (overlapping) estimates to be used  for bode plots etc.
>> G12=sortfqs(G1, G2);
```

The frequency response estimates are stored in matrices (`G12` in the example above) which has two columns, one with frequencies ω_k (in radians/s) and the other with complex numbers $G(i\omega_k)$. In Matlab you can make nice Bode and Nyquist plots of the process. Frequency response plotting is done using

```
bopl(Giw1,Giw2,Giw3,Giw4)
nypl(Giw1,Giw2,Giw3,Giw4)
```

Arguments `Giw2` to `Giw4` are optional. Use `bogrid` to get a grid in the plot.

4.4 Discussion

Try to determine the characteristics of the process using the Bode-diagram. What can you say about

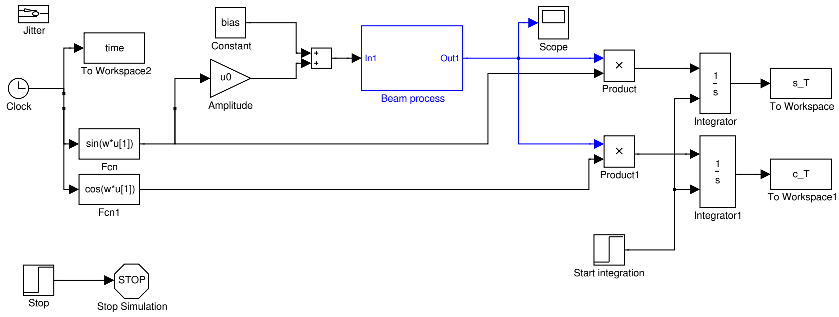

Figure 3 Simulink model for frequency response experiments (used with the matlab function `fra.m`.)

- the order of the system?
- poles and zeros?
- static gain?
- linearity?

5. Design and Control

Here we assume that the main purpose for the identification is to achieve a frequency response $G(i\omega)$, that could be used to design a controller giving desired closed-loop system properties.

MATLAB is used to design a controller on the form

$$u(s) = \frac{S(s)}{R(s)} \left(\frac{B_{ff}}{A_{ff}} r(s) - y(s) \right)$$

This is a two-degree of freedom controller. First the feedback compensator is designed to give the desired loop-transfer function. Then a feedforward filter may be used to shape the response for command signals. The closed loop system will then be

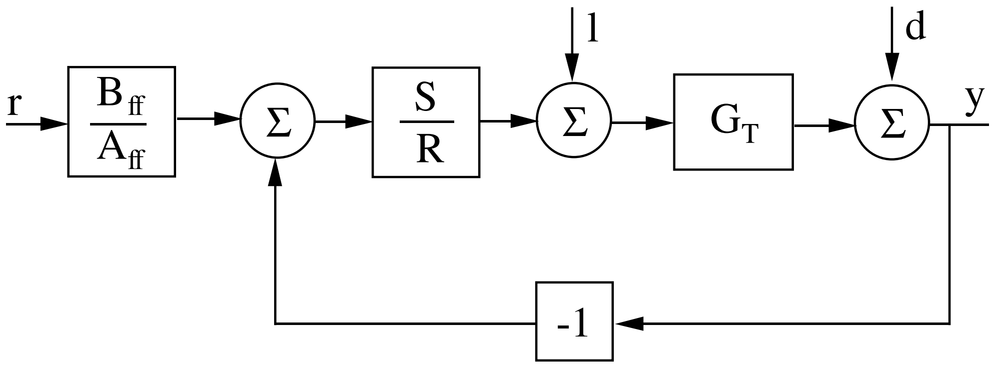

Figure 4 Closed-loop system.

Design goal: *Make the closed-loop system as fast as possible and fulfill that the rise time be less than 0.2 s and the maximum overshot less than 10%.*

5.1 Feedback Design

A feedback compensator is designed using classical frequency compensation. The feedback is used mainly to get a reasonably fast and well damped closed-loop system by changing the cross-over frequency and phase margin. This is readily done with

```
[S,R,Liw,Tiw,Ciw]=fbdesign(Giw);
```

where S and R are feedback compensator polynomials, Ciw=S($i\omega$)/R($i\omega$), Liw=Giw·Ciw and Tiw=Liw/(1+Liw). This function is interactive and the user may specify gain, poles and zeros of the compensator. Write *hlp* to see the available commands. Plots of the uncompensated and the compensated system may be drawn.

Preparation: Find the transfer functions in the block diagram that have the frequency responses Ciw, Liw and Tiw.

5.2 Feed-forward Design

The feedback design specifies Tiw. A feedforward filter may then be added to reduce the high frequency content in the reference signal. This is done with

```
[Bff,Aff,YRiw,FFiw]=ffdesign(Tiw);
```

where the feedforward filter Bff/Aff has frequency response FFiw and the frequency response from r to y is YRiw=Tiw·FFiw. Gain, poles and zeros are chosen interactively as in `fbdesign`. Plots of Tiw and YRiw may be drawn.

If you have time, go over the feedback and feedforward design several times to create different control designs. Then you can compare a faster design with a slower one, etc. Save your workspace in Matlab before you start another design iteration.

5.3 Real Time Control

The controller design is carried out in continuous time, i.e. the polynomials are given in Laplace s, but a discrete time controller is needed since the controller is implemented in a computer. If the sampling rate is sufficiently high the discrete controller will behave as the continuous one. Bilinear approximation (Tustin) of Bff/Aff and S/R for the sampling interval h is made by typing `makec2d` at the command prompt. Before execution this command the sampling interval must be defined in the workspace. A reasonable choice is 0.01 s.

The commands carried out in `makec2d` are:

```
% Continuous controller
Gc = tf(S,R)
%
% Discrete counterpart
Gcd = c2d(Gc,h,'tustin')
%
```

```
% Extract polynomials
[Sd,Rd,hd,dd]=tfdata(Gcd,'v')
%
% RST representation
Td = Sd

% Continuous FF-filter
Gff = tf(Bff,Aff)
%
% Discrete FF-filter
Gffd = c2d(Gff,h,'tustin')
%
% Extract polynomials
[Bffd,Affd,hd,dd]=tfdata(Gffd,'v')
%
% Rd,Sd,Td,Bffd, and Affd are used in lab1.mdl
```

This command also rewrites the controller into standard RST-form. The parameters of the discrete time controller is written to the workspace for later use in the Simulink model `lab1`.

5.4 Control experiments

Save your workspace in Matlab to have a backup of your design. Type `lab1` to bring up the Simulink model which will be used for control. The model consists of a reference generator, a controller on RST-form, the I/O connection to the process, and some scopes for displaying the signals of the system.

The reference generator has an output sequence governed by the workspace parameter `refseq`. If you have time you can alter this sequence or replace the complete block, for example if you wish to have a sinusoidal reference.

Do the controllers and the system behave as expected?

6. Conclusions

The identification has been used for two purposes. Firstly an accurate frequency response was determined to gain insight into the dynamics of the beam. Secondly, the frequency response was used for design of a controller, which was tested on the real process to validate the identified model.

7. References

Åström, K. J. (1968): *Reglerteori (Control Theory)*. (In Swedish). Almqvist & Wiksell, Uppsala, Sweden.

Glad, T. and L. Ljung (1989): *Reglerteknik—Grundläggande teori*, 2nd edition. Studentlitteratur, Lund.

Johansson, R. (1993): *System Modeling and Identification*. Prentice Hall, Englewood Cliffs, New Jersey.