Multivariable Control

Laboratory Exercise 1

Loop Shaping[1]

Department of Automatic Control
Lund University

1. Introduction

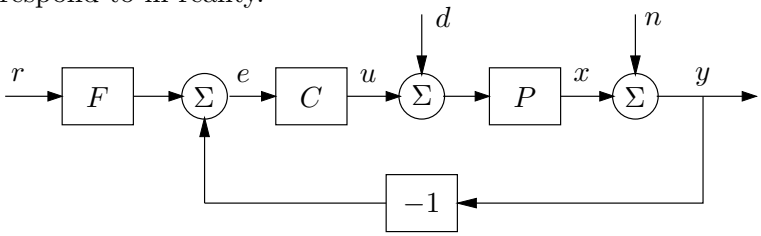
The purpose of this laboratory exercise is to design a controller for a flexible servo using frequency-domain methods. We will use the technique of *loop shaping*, where we shape the Bode diagram of the open-loop system using different controllers.

Preparations

Read about Bode diagrams and lead/lag compensation, in for instance the textbook from the Basic Course. You should have solved exercises 1–4 from exercise session 4 before the lab. At the beginning of the lab, you should also be able to discuss the points below:

Discussion points

1. Explain what parts of the block diagram below that are relevant to the flexible servo, and what the different signals correspond to in reality.



2. Make connections between the items below:

Increase feedback gain	Improve stability
Add a lead-filter	Reduce stationary error
Add a lag-filter	Get faster response
Introduce integral action	

3. Sketch a typical desired Bode diagram of an open-loop system $L(s) = C(s)P(s)$. Mark the amplitude and phase margins and the cut-off frequency.

4. The flexible servo is a "resonant system". How can this be seen in the Bode diagram of $P(s)$?

[1]Written by Tomas Olsson, latest updated 2011-09-20 by Alfred Theorin.

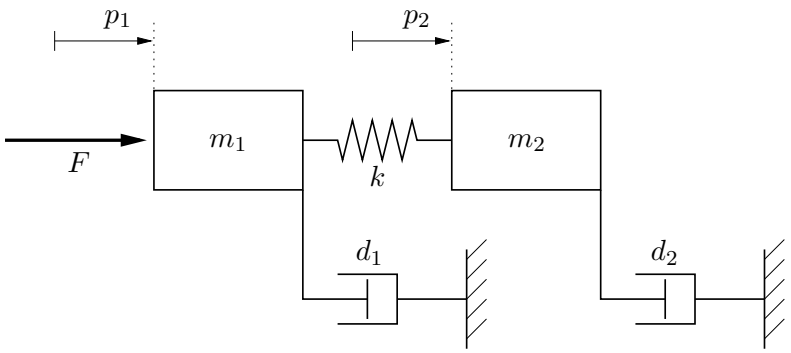


Figure 1 The flexible servo process.

The Process

The flexible servo process consists of two masses connected by a spring, see Fig. 1. You may remember the process from Laboratory 3 in the Basic Course. The mass on one side is driven by a DC-motor, which exerts a force F on the mass. Note that the only damper that is visible on the real process is d_2. However, other dampings in the system can be added and modeled according to Fig. 1.

The purpose is to control the position p_2 of the mass m_2. On the lab process the positions of both masses can be measured, but we will only use the position measurement p_2.

Linear model

We have the two masses m_1 and m_2. The spring between the masses has the spring constant k. The dampings are d_1 and d_2.

One of the masses is driven by a DC-motor. Here we neglect the internal dynamics of the motor. The force from the motor on the mass is proportional to the voltage u, that is

$$F = k_m \cdot u$$

A force balance gives us the following model:

$$m_1 \frac{d^2 p_1}{dt^2} = -d_1 \frac{dp_1}{dt} - k(p_1 - p_2) + F(t)$$

$$m_2 \frac{d^2 p_2}{dt^2} = -d_2 \frac{dp_2}{dt} + k(p_1 - p_2)$$

Introducing the state vector $x = [\, p_1 \quad \dot{p}_1 \quad p_2 \quad \dot{p}_2 \,]^T$ and the output $y = p_2$, the system can be written on state-space form,

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t)$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{m_1} & -\frac{d_1}{m_1} & \frac{k}{m_1} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k}{m_2} & 0 & -\frac{k}{m_2} & -\frac{d_2}{m_2} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{k_m}{m_1} \\ 0 \\ 0 \end{bmatrix}, \quad C = [\, 0 \quad 0 \quad k_y \quad 0 \,]$$

For a real lab process we have measured and estimated the following constants and coefficients

$$\begin{aligned} m_1 &= 2.29 \text{ kg} \\ m_2 &= 2.044 \text{ kg} \\ d_1 &= 3.12 \text{ N/m/s} \\ d_2 &= 3.73 \text{ N/m/s} \\ k &= 400 \text{ N/m} \\ k_m &= 2.96 \text{ N/V} \\ k_y &= 280 \text{ V/m} \end{aligned}$$

The transfer function from u to y is given by

$$P(s) = \frac{7.083 \cdot 10^4}{s^4 + 3.187s^3 + 372.9s^2 + 585.4s}$$

Design specifications

A general control system should satisfy a number of properties. For example, the system should be able to follow reference signals, reject load disturbances, attenuate measurement noise, reduce the sensitivity to modeling errors, and satisfy constraints on the control signal.

The desired behavior of the system can be specified in a number of different ways. In our case, we have specified the behavior of the closed-loop system in the time domain, see Fig. 2. We want a well-damped reference step response with a rise-time between 0.2 and 0.6 seconds and a settling time of at most 2 seconds. A constant load disturbance should be rejected in at most 2 seconds. Also, the control signal should not be too violent or noisy.

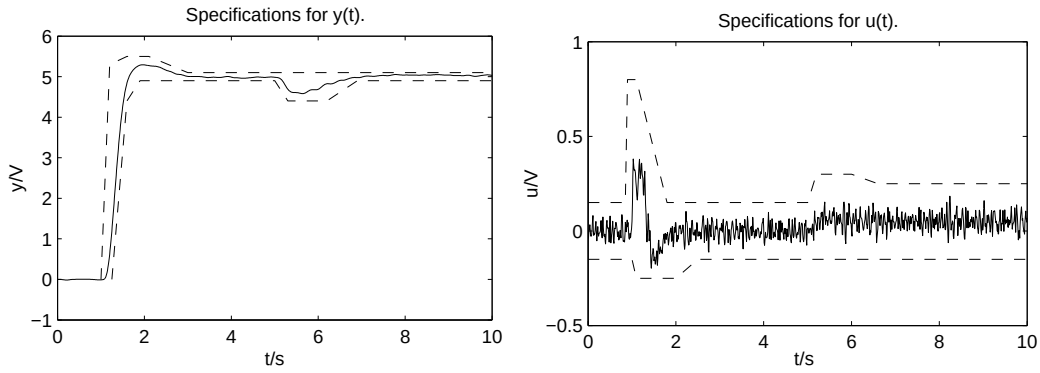


Figure 2 Specifications in the time domain: response to reference step at $t = 1$ and load disturbance at $t = 3$ (left) and the corresponding control signal (right).

Furthermore, the system should be robust to process variations. Therefore, we also require

- a phase margin of at least 35°
- a gain margin of at least 1.88 (equal to 5.5 dB)

2. The lab interface

The controllers are designed and evaluated in MATLAB/Simulink. There is one Simulink model for simulation and another one for experiments on the real process. In both cases, you define your controller by simply entering the transfer functions `C` (the feedback compensator) and `F` (the feedforward filter) in the MATLAB workspace. Example:

```
s = tf('s');
C = 1/s;    % A pure I-controller
F = 1;      % No feedforward filter before exercise 6
```

To save time, you can define your controller directly in *loop_shaping.m* and run this script before each simulation. The script creates a Bode plot with phase margin, amplitude margin and cutoff frequency. If needed, more plots can be added.

Simulation model

The control system can be simulated in the Simulink model *servo_simulated* (type `servo_simulated` to open it). The model is shown in Fig. 3. It is possible to change the process parameters by double-clicking on the button *Change process parameters*. The button *Check specifications* will plot the results from the simulation together with the specifications. Using the switches we can enable/disable the measurement noise and change the load disturbances.

Experiments on the real process

The Simulink model *servo_real* is similar to *servo_simulated*, but here we use the controller on the real process instead. Before each experiment you need to reset the current position to zero by pressing the two buttons on the servo marked "POS RESET". If you get a process overload, you will also need to press the "RESET" button.

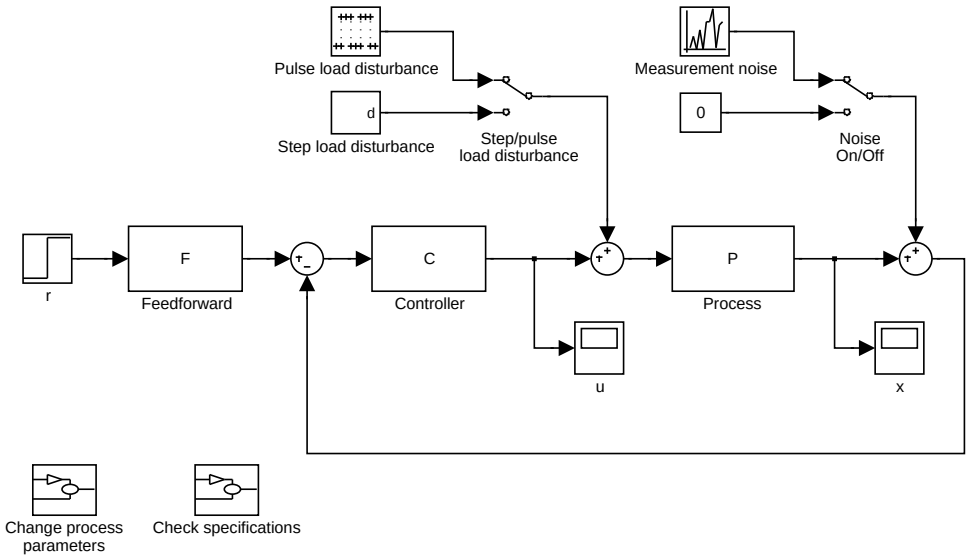


Figure 3 Simulink model `servo_simulated`.

3. Introductory experiments

Exercise 1 Read the process model into Matlab using the command *servo_model*. Type *help servo_model* in Matlab to see how this command works. Plot the Bode diagram (see Fig. 4). Where are the poles and zeros located (use `pzmap`)? Simulate a short push on mass 1 with the command `impulse`. Try the same thing on the real process. Will the process be difficult to control?

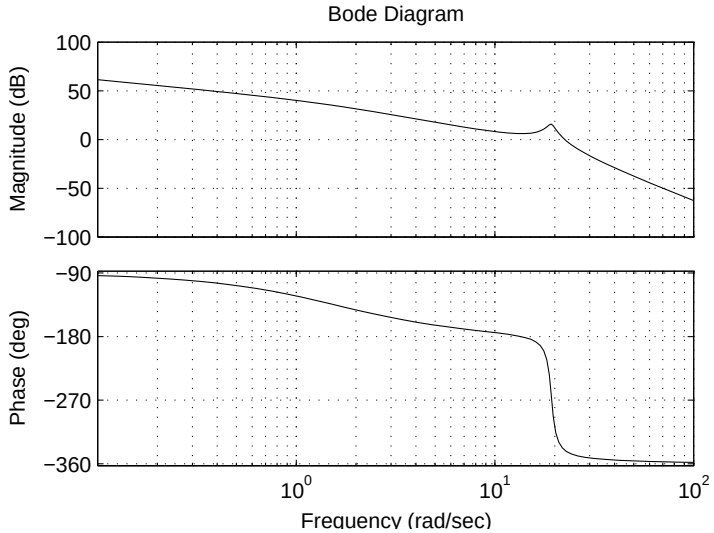


Figure 4 Bode diagram for the process.

Exercise 2 What is the crossover frequency and the phase margin of the system, when it is controlled by a proportional controller with $K = 1$? Is the closed-loop system stable? Can you find a P-controller that gives a good step response in simulation? Can a P-controller handle constant load disturbances? Do you think that a PI-controller,

$$C(s) = K\left(1 + \frac{1}{sT_i}\right) = K\left(\frac{sT_i + 1}{sT_i}\right)$$

would work (look at the Bode diagram phase plot!)? A PID-controller?

4. Loop-shaping design

In this section, we will design different controllers $C(s)$ to shape the frequency response of the open-loop system $L(s) = C(s)P(s)$. We will ignore the feed-forward filter $F(s)$ until Exercise 6.

In the following exercises, you should work with the Bode diagram of $L(s)$ until you get the desired shape. The command `margin` plots a Bode diagram and also shows the amplitude and phase margins. Also simulate the system to check your design against the time-domain specifications.

Exercise 3 Use loop shaping to find a controller $C(s)$ that fulfills the design specifications as well as possible. You don't have to fulfill the specification on the overshoot in the reference step, since reasonable overshoot can be removed later in Exercise 6 by feedforward filter.

Start with a PI-controller and add poles and zeros to shape the Bode diagram. A strategy can be to start by trying to reduce the effects of the resonant peak to be able to get a cross-over frequency that is high enough, and then try to improve the phase margin. Aim for a cross-over frequency of around $\omega_c = 5$ rad/s. A single lead or lag filter will not be enough to fulfil the specifications.

Try to fulfil both the time domain specifications and the specifications on the phase and amplitude margin stated on page 3. The time domain specifications are easily checked using "Check specifications" in the Simulink model.

When you find a controller that works well in simulations, try to control the real process. Does it work? If not, try to explain why! Do you see any significant differences when comparing the step response from the real process with the simulated response?

Note: The motor that drives the masses is very strong. Therefore it is very important that you handle the process carefully!

Exercise 4 Try to change the mass m_2 in the simulation model, and try to control the modified process using the controller from the previous exercise. Use "Change process parameters" in the Simulink model. How is the robustness with respect to process variations? Try to change other process parameters and see how the performance of the controller changes.

Exercise 5 Try to cancel the resonant poles of the process by adding the corresponding zeros to the controller. Add more poles if that is neccesary to make the system proper. Try to make the control faster. Are there any problems with this method? Look at the Bode diagram of $G_{yd}(s)$ and the response to (impulse) load disturbances. Explain!

Feedforward filter

Sometimes rejection of disturbances and fast response to changes in the reference value are almost mutually exclusive properties of the control system. As you have seen in the previous exercise, it is difficult to satisfy all the specifications using a feedback controller. This is because it handles changes in the reference signal and changes in the process output in the same way, according to

$$U(s) = C(s)\left(R(s) - Y(s)\right).$$

Often it is better to first design a feedback controller to achieve good rejection of load disturbances, robustness to process variations, and small amplification of measurement noise. Then we can design the feedforward filter $F(s)$ to better handle changes in the reference signal.

The new controller can be written

$$U(s) = C(s)\left(F(s)R(s) - Y(s)\right).$$

The feedforward can be seen as a filter on the reference signal, and it does not affect the disturbance rejection properties of the system.

Exercise 6 Add a feedforward filter $F(s)$ to the design you obtained in Exercise 3. If you have an overshoot in the step response, you can for instance use a first-order filter

$$F(s) = \frac{1}{sT_f + 1}.$$

Test your controller on the real process. What is different from the simulations?

Exercise 7 (Extra) Change the process model so that we instead measure and control the position of mass 1. How does this change the process poles, zeros and Bode diagram? Try to design a controller for this process in the same way as for mass 2. Is it easier or more difficult than for mass 2? Test your controller on the real process.