EE – Automatic Control
2014-03-17

EL2520

CONTROL THEORY AND PRACTICE ADVANCED COURSE

Lå 13/14

COMPUTER EXERCISES LABORATORY EXERCISE

Table of Contents

EE – Automatic Control

Control Theory and Practice
Advanced Course

Computer Exercise: CLASSICAL LOOP-SHAPING

Magnus Jansson och Elling W. Jacobsen, January 1998

v.2014

1 Introduction

Loop-shaping is a classical procedure for control design. In the basic course it was denoted lead lag design. Loop-shaping was introduced during world war II and it was used to construct single variable circuits, such as amplifiers in feedback (Bode). This knowledge has later been transferred to other areas of automatic control, and it has been extended to multi-variable systems, i.e., systems with multiple input and output signals.

The idea is to shape the *open-loop* gain with a controller in order to achieve intended properties of the *closed-loop* system under feedback. In the 70'ies and 80'ies advanced methods for loop-shaping based on optimization were developed. However, in this computer exercise we will focus on basic classical loop-shaping. Frequency domain descriptions are fundamental in control design!

We will here only consider SISO systems (single input single output), but the ideas are also applicable to MIMO systems (multiple input multiple output).

Preparations: Chapters 7.1-7.4 in the course book (Ljung, Glad, "Control theory"). It is also recommended to repeat Chapter 5.5 in the basic course book (Glad, Ljung, "Reglerteknik-Grundläggande teori").

Presentation: All problems in this exercise should be solved and be presented in a written report. The date when the report should be handed in is indicated on the course website. The report should be a full report, contain *abstract* and *conclusions*, relevant figures and tables with *captions* and *legends*, etc. Focus on explaining what you have done, and do not repeat theorems but cite them. Remember to write concise (a report is not a novel), be specific in your writing ("the overshoot is 20%" compared to "the overshoot is not so good"), and check your speling. The report will be graded based on both the content and presentation. The exercise should be performed in pairs of students.

2 Background

Consider the control system in Figure 1.

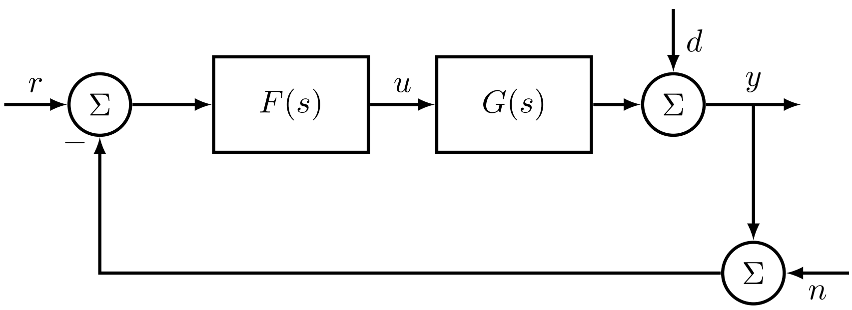


Figure 1: K–controller, G–system, r–reference signal, u–control signal, d–disturbance signal, y–output signal, n–measurement noise.

The loop gain is given by $L = GF$, the sensitivity function $S = (I + L)^{-1}$ and the complementary sensitivity function $T = (I+L)^{-1}L$. Remember that we have $S+T = I$. The control error depends on the input signals as

$$e = r - y = Sr - Sd + Tn.$$

Since we wish to have a small control error, we obtain the following conditions

$$e \approx 0 \Rightarrow \begin{cases} i) \quad S \approx 0 \ \Rightarrow \ T \approx I \ \Rightarrow \ L \text{ large} \\ ii) \quad T \approx 0 \ \Rightarrow \ S \approx I \ \Rightarrow \ L \text{ small} \end{cases}$$

We obviously have contradictive conditions! The case $i)$ corresponds to reference tracking and disturbance attenuation while case $ii)$ corresponds to noise attenuation (and sensitivity to model errors, robustness). For example, if we wish to track low frequency reference signals we have to design the loop gain to be large at low frequencies.

Apart from keeping the control error small, the control signal should not be too large or vary too much. Since

$$u = F(r - y - n)$$

this condition implies that the control gain must not be designed too large, F small $\Rightarrow \ L = GF$ small.

Stability is another important issue. The slope of the curve $|L(i\omega)|$ is coupled to the phase $\arg\{L(i\omega)\}$. For example, $L = a/s^n$ has slope $-n$ and phase $-n\pi/2$. In order to keep a reasonable stability margin, $|L|$ must not have too large slope around the cross-over frequency ω_c. Typically, $|L|$ is designed to have slope ≈ -1 at ω_c.

Also note that the phase margin is coupled to control performance. For example we have resonance peaks $M_S = \max_\omega |S|$ and $M_T = \max_\omega |T|$

$$M_T > \frac{1}{\phi_M} \ ; \quad M_S > \frac{1}{\phi_M}$$

where the phase margin ϕ_M is given in radians. For example, if we demand that the resonance peaks should be smaller than 2, then the phase margin has to be larger than 30°.

Such contradictive constraints give rise to different strategies to shaping the loop L so that performance demands are met. They also provide limits of achievable control performance.

3 Introduction to Control System Toolbox

In this computer exercise we will use MATLAB to shape the loop, just as we did in the basic course. Most of the functions are in Control System Toolbox. Let us start by defining some useful function. Recall that you get access to the MATLAB help be typing `help "function name"`.

A transfer function

$$G(s) = \frac{s+2}{s^2+2s+3}$$

is defined in MATLAB by typing

```
s=tf('s');     G=(s+2)/(s^2+2s+3)
```

The product of two transfer functions is obtained by

```
G12 = G1 * G2
```

For a system with 2 inputs and 2 outputs, the closed-loop transfer matrix is obtained with

```
S=feedback(eye(2),G*F);     T=feedback(G*F,eye(2))
```

For a SISO system this can be written

```
S=1/(1+G*F);     T=G*F/(1+G*F)
```

For numerical reasons it is **very important** to use the function `minreal`, for example `minreal(T)`. This creates an equivalent system where all canceling pole/zero pair or non minimal state dynamics are eliminated.

The bode diagram for `G` is plotted by typing

`bode(G)` or `bode(G,{wmin,wmax})`

Amplitude and phase at a given frequency are obtained by

```
[m,p]=bode(G,w)
```

Phase margin, amplitude margin and corresponding frequencies are obtained by

```
[Gm,Pm,wp,wc]=margin(G*F)
```

To simulate a step response in the control signal, use the function

`step(G)` or `step(G,tfinal)`

In the same way, to simulate a step response in the reference signal, we type

```
step(T)
```

4 Exercises

4.1 Basics

Consider a system which can be modeled by the transfer function

$$G(s) = \frac{3(-s+1)}{(5s+1)(10s+1)}.$$

Exercise 4.1.1. Use the procedure introduced in the basic course to construct a lead-lag controller which eliminates the static control error for a step response in the reference signal.

$$F(s) = K \underbrace{\frac{\tau_D s + 1}{\beta \tau_D s + 1}}_{\text{Lead}} \underbrace{\frac{\tau_I s + 1}{\tau_I s + \gamma}}_{\text{Lag}}$$

The phase margin should be 30° at the cross-over frequency $\omega_c = 0.4$ rad/s.

. . . .

. . . .

. . . .

Exercise 4.1.2. Determine the bandwidth of the closed-loop system and the resonance peak M_T. Also, determine the rise time and the overshoot for step changes in the reference when the controller designed in 4.1.1. is used.

. . . .

. . . .

. . . .

Exercise 4.1.3. Modify the controller in 4.1.1. such that the phase margin increases to 50° while the cross-over frequency is unchanged. For the corresponding closed-loop system, determine the bandwidth and resonance peak. Also, determine the rise time and the overshoot of the step response.

. . . .

. . . .

. . . .

. . . .

. . . .

4.2 Disturbance attenuation

Now we will construct a controller which both tracks the reference signal and attenuates disturbances. The block diagram of the control system is given in Figure 2. We assume that the signals have been scaled such that $|d| < 1$, $|u| < 1$ and $|e| < 1$ where $e = r - y$.

The exercise is about designing F_r and F_y in Figure 2 such that:

- The rise time for a step change in the reference signal less than 0.2 s and the overshoot is less than 10%.

- For a step in the disturbance, we have $|y(t)| \leq 1 \; \forall t$ and $|y(t)| \leq 0.1$ for $t > 0.5$ s.
- Since the signals are scaled the control signal obeys $|u(t)| \leq 1 \; \forall t$.

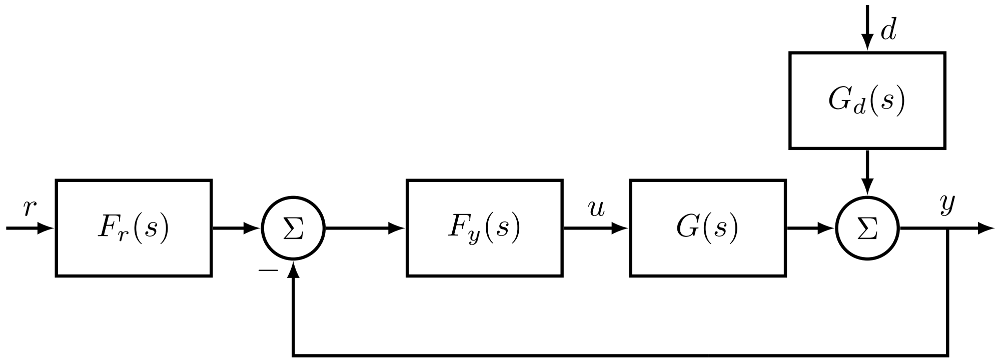


Figure 2: F_r–prefilter, F_y–feedback controller, G–system, G_d–disturbance dynamics, r–reference signal, u–control signal, d–disturbance signal, y–measurement signal.

The transfer functions have been estimated to

$$G(s) = \frac{20}{(s+1)((\frac{s}{20})^2 + \frac{s}{20} + 1)}$$

$$G_d(s) = \frac{10}{s+1}$$

Exercise 4.2.1. For which frequencies is control action needed? Control is needed at least at frequencies where $|G_d(j\omega)| > 1$ in order for disturbances to be attenuated. Therefore the cross-over frequency must be large enough. First, try to design F_y such that $L(s) \approx \omega_c/s$ and plot the closed-loop transfer function from d to y and the corresponding step response. (A simple way to find $L = \omega_c/s$ is to let $F_y = G^{-1}\omega_c/s$. However, this controller is not proper. A procedure to fix this is to "add" a number of poles in the controller such that it becomes proper. How should these poles be chosen?)

. . . .

. . . .

A loop gain of slope -1 at all frequencies gives in our case poor disturbance attenuation. To understand the reason for this, note that the output is given by

$$y = SG_d d = (1+L)^{-1} G_d d.$$

Provided the signals have been scaled we want $|(1+L)^{-1}G_d| < 1$ for all ω. For frequencies where $|G_d| > 1$ this approximately implies $|L| > |G_d|$ or $|F_y| > |G^{-1}G_d|$. Most often we also want integral action and as a starting point we can choose

$$F_y = \frac{s + \omega_I}{s} G^{-1} G_d, \tag{1}$$

where ω_I determines the frequency range of efficient integral action. We see that if $G_d \approx 1$, the controller should contain the inverse of the system. On the other hand if $G_d \neq 1$ the controller should be designed in some other fashion. Especially, we observe that if the disturbance is on the input side to the system we have $G_d = G$ and then F_y should be chosen as a PI controller according to (1).

Note that the controller (1) cannot be used if it is not proper, causal and stable. To ensure these properties, approximations of (1) may be necessary.

Exercise 4.2.2. Let us now reconstruct F_y according to the instructions above. We will start with the disturbance attenuation. In a second step, adjustments can be made on F_r to obtain the desired reference tracking properties. Start by choosing F_y according to (1). Try different approximations of the product $G^{-1}(s)G_d(s)$ and choose ω_I large enough so that step disturbances are attenuated according to the specifications.

....

....

....

Exercise 4.2.3. To fulfill the reference tracking specifications, we can combine lead lag control and prefiltering of the reference signal. First, try to add lead action to F_y to reduce the overshoot. Then it can be necessary to add prefilter action to fulfill all specifications. Note that F_r should be as simple as possible (why?). Also, remember to check the size of the control signal ($u = F_yF_rSr - F_yG_dSd$)! Typically a low pass filter is chosen, for example

$$F_r = \frac{1}{1+\tau s}.$$

....

....

Exercise 4.2.4. Finally, check that all specifications are fulfilled. Plot the sensitivity and complementary sensitivity functions.

....

....

EE – Automatic Control

Control Theory and Practice
Advanced Course

Computer Exercise: MULTIVARIABLE SYSTEMS

Anders Hansson, Per Bodin och Elling Jacobsen, December 1998

v.2014

1 Introduction

In this computer exercise we will investigate properties of multivariable systems. The application is a model of a four-tank-process. In particular we will consider pairings between different input and output signals and non-minimum phase dynamics. The pairings are analyzed using RGA and we will investigate their influence in decentralized PI control.

Preparations: Chapters 3.3, 3.5, 6.5, 7.3-7.5, 7.7 and 8.3 in Glad, T. and Ljung, L.: "Control theory—multivariable and nonlinear methods".

Presentation: All problems in this exercise should be solved, but only the tasks on the report form should be handed in. The report form and the date when it should be handed in can be found on the course website. The exercise should be performed in pairs of students.

2 Theoretical overview

This section contains a short overview of basic theory for multivariable systems. The content is based on the course book.

2.1 Poles and zeros

As for SISO systems we can define poles and zeros for a linear MIMO system with transfer matrix $G(s)$.

The poles of $G(s)$ are defined as the eigenvalues of the system matrix A in a minimal state space realization of the system and they are calculated as the roots of the *pole polynomial*, $\det(sI - A)$. The pole polynomial can also be obtained by calculating the least common denominator of all sub-determinants of $G(s)$.

It is more difficult to extend the definition of zeros from the SISO case to the MIMO case. The most common definition of a zero of the system is a value of s where the transfer matrix $G(s)$ looses rank. For the special case of square transfer matrices, the zeros are given by the roots of $\det G(s) = 0$.

2.2 Singular values, directions and H_∞ norms

The singular values σ_i of a matrix A are defined as $\sigma_i = \sqrt{\lambda_i}$, where λ_i are the eigenvalues of the matrix A^*A. The largest singular value of A is denoted $\overline{\sigma}(A)$, and the smallest as $\underline{\sigma}(A)$. If $y = Ax$, it follows from the singular value definition that

$$\underline{\sigma}(A) \leq \frac{|y|}{|x|} \leq \overline{\sigma}(A)$$

where the relation between the norm of y and the norm of x, $|y|/|x|$, can be interpreted as the gain of the matrix A. If x is parallel with the eigenvector corresponding to the largest eigenvalue of $A^T A$ then we have $|y| = \overline{\sigma}(A)|x|$, and if x is parallel to the eigenvector corresponding to the smallest then we obtain $|y| = \underline{\sigma}(A)|x|$. This way we can define directions corresponding to the largest and smallest singular value for A respectively.

For a linear stable multivariable system with transfer matrix $G(s)$ we have

$$Y(i\omega) = G(i\omega)U(i\omega)$$

where $Y(s)$ and $U(s)$ are the Laplace transforms of the system's output and input signals. According to the definition of singular values we therefore have

$$\underline{\sigma}(G(i\omega)) \leq \frac{|Y(i\omega)|}{|U(i\omega)|} \leq \overline{\sigma}(G(i\omega))$$

Introducing

$$|G(i\omega)| := \overline{\sigma}(G(i\omega))$$

it holds

$$|Y(i\omega)| \leq |G(i\omega)||U(i\omega)|.$$

This notation is analogous to the SISO case where the norm is interpreted as the absolute value of $G(i\omega)$, and the inequality above turns to an equality.

To understand the inequalities in the MIMO case we can choose the input parallel to the direction corresponding to the largest or smallest singular values of $G(i\omega)$. The directions decide the "mix" of the input signal components that results in the largest and smallest gain of the system respectively.

The largest gain of the multivariable system $G(i\omega)$ is denoted $\|G\|_\infty$, which is given by

$$\|G\|_\infty = \max_\omega |G(i\omega)|.$$

$\|G\|_\infty$ is called the H_∞ norm of G. For output signal $y(t)$ and input signal $u(t)$ it holds

$$\|y\| \leq \|G\|_\infty \|u\|.$$

Therefore the H_∞ norm can be interpreted as the time domain gain of the system. It holds

$$\sup_u \frac{\|y\|}{\|u\|} = \|G\|_\infty,$$

which can be seen as the definition of the H_∞ norm for G.

2.3 Decentralized control

A fundamental problem in multivariable control is the pairings between the inputs and outputs. This means that if one input changes there is generally a change in all outputs. A measure of the strength of the pairings in a multivariable system, $G(s)$, is given by the *Relative Gain Array*, RGA of the transfer matrix G. It is defined by

$$\text{RGA}(G(i\omega)) := G(i\omega). * \left[G^{-1}(i\omega)\right]^T,$$

where ".$*$" denotes element wise multiplication. We can use RGA to determine which input and output that are suitable to pair in a decentralized controller. Two rules of thumb:

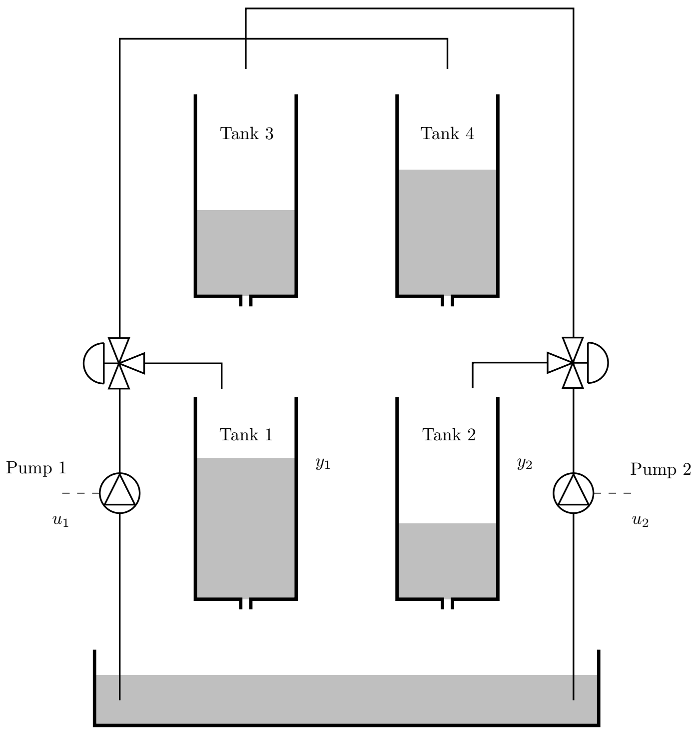


Figure 3: The four-tank process

1. Find a pairing such that diagonal elements of $\mathrm{RGA}(G(i\omega_c))$ are as close to 1 as possible, where ω_c is the intended cross-over frequency.

2. Avoid pairings which correspond to negative elements in $\mathrm{RGA}(G(0))$.

3 Exercises

In this computer exercise linear models of a four-tank process will be investigated. The system is shown schematically in Figure 3. The input signals are the voltages of the pumps, u_1 and u_2. The output signals that we want to control are the levels in the lower tanks, y_1 and y_2. Connected to each pump there is a valve that divides the water to the upper and lower tanks. A linear multivariable model with 2 inputs and 2 outputs is given by $Y(s) = G(s)U(s)$ where

$$Y(s) = \begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix}, \quad U(s) = \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} \quad \text{and} \quad G(s) = \begin{bmatrix} g_{11}(s) & g_{12}(s) \\ g_{21}(s) & g_{22}(s) \end{bmatrix}$$

Depending on the settings of the valves we obtain different $G(s)$. In this computer exercise two different valve settings will be investigated: in the first, most of the water will go directly to the lower tanks and $G(s)$ is minimum phase; in the second, most of the water go to the upper tanks and $G(s)$ will be non-minimum phase.

3.1 Poles, zeros and RGA

A linear state space model for the four-tank process is generated by the MATLAB functions `minphase` and `nonminphase`. To put the minimum phase model in the variable `sysmp` we write

```
sysmp = minphase
```

The following MATLAB functions can be used in this exercise. The poles of the system `sys` are obtained with

```
pole(sys)
```

and its zeros with

```
tzero(sys)
```

The system should be given as a (minimal) state space description when these functions are used. (Otherwise numerical problems can appear in MATLAB.)

The singular values for a system are calculated with `sigma`. To extract system matrices and transfer functions the functions `ssdata` and `tfdata` are used, respectively. The singular value decomposition are calculated using `svd`. To calculate step responses for linear systems the function `step` can be used. The Bode diagram of a system is plotted using `bode`. Note that dB scale is used. Use the MATLAB Help to learn more about the functions.

The problems below should be solved both for the minimum phase and non-minimum phase case. It is suitable to start with the former.

Exercise 3.1.1. Calculate the transfer matrix $G(s)$. Investigate each element of the matrix (Hint: `G(1,1)` extracts element (1,1)). Calculate the poles and zeros of the elements.

. . . .

. . . .

Exercise 3.1.2. Calculate the poles and zeros of the multivariable system. Do these imply any constraint on the achievable control performance?

. . . .

. . . .

Exercise 3.1.3. Investigate the largest and smallest singular values for the system at different frequencies. Calculate the H_∞ norm of the system.

. . . .

. . . .

Exercise 3.1.4. Investigate the RGA of the system at frequency 0. What conclusions can we draw about the possibility of using decentralized control?

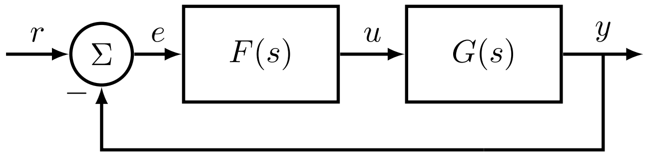


Figure 4: Control.

. . . .

. . . .

Exercise 3.1.5. Plot the step response for one input at the time. Investigate the outputs: is the system coupled? Is this in line with the properties of RGA?

. . . .

. . . .

Now solve the above problems above for the non-minimum phase case.

Exercise 3.1.6. Describe the most important differences between the two cases and discuss how it affects the control performance.

. . . .

. . . .

3.2 Decentralized control

We will now investigate control of the four-tank process as illustrated in Figure 4, where both the process $G(s)$ and the controller $F(s)$ are 2×2 matrices of transfer functions. The simplest way to control a system is to use decentralized control. This means that one input is paired with one output, which is controlled with a one-dimensional controller. An example is depicted in Figure 5, where output y_1 is controlled with the input u_1 through the controller $f_1(s)$. Similarly, the output y_2 is controlled with the input u_2 through $f_2(s)$. Here y_1 is paired with u_1 and y_2 is paired with u_2. This corresponds to Figure 4 with

$$F(s) = \begin{bmatrix} f_1(s) & 0 \\ 0 & f_2(s) \end{bmatrix}$$

The other way around, if y_1 is paired with u_2 and y_2 is paired with u_1, then $F(s)$ is given by

$$F(s) = \begin{bmatrix} 0 & f_1(s) \\ f_2(s) & 0 \end{bmatrix}$$

In the first case, the controllers $f_1(s)$ and $f_2(s)$ are designed using single-variable control design with the transfer functions between u_1 and y_1 and between u_2 and y_2. A

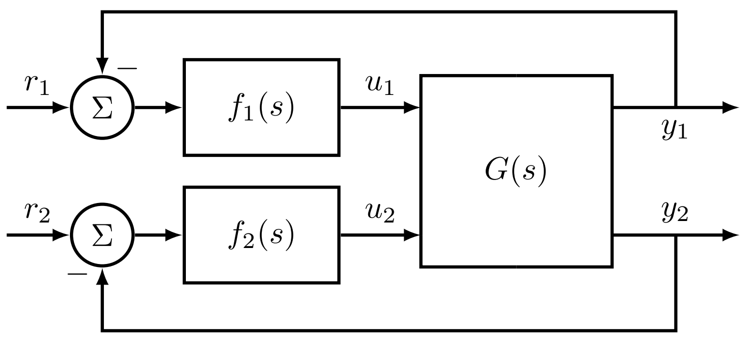


Figure 5: Decentralized control.

procedure for one-dimensional control design was investigated in the computer exercise on loop-shaping. Here we will design PI controllers

$$f_j(s) = K_j \left(1 + \frac{1}{sT_{i_j}}\right), \quad j = 1, 2$$

such that the intended phase margin φ_m and cross-over frequency ω_c are obtained. The loop gain is given by $L = GF$. Therefore we wish to shape l_{11} and l_{22} in such a way that given specifications are fulfilled.

Let us now investigate the case where y_1 is to be controlled with u_1. Denote the intended phase margin and cross-over frequency by φ_m and ω_c respectively. From l_{11} we see that K_1 and T_{i_1} are given by the following equations:

$$|g_{11}(i\omega_c) f_1(i\omega_c)| = 1 \tag{2}$$

$$\arg g_{11}(i\omega_c) + \arg f_1(i\omega_c) - \varphi_m = -\pi \tag{3}$$

Note that Equation (3) is equivalent to

$$\arg g_{11}(i\omega_c) + \arctan(\omega_c T_{i_1}) - \pi/2 - \varphi_m = -\pi$$

where $\arg g_{11}(i\omega_c)$ can be obtained from the Bode diagram of $g_{11}(s)$. This gives T_{i_1}. Then we can draw the Bode diagram for the loop gain

$$l_{11}(s) = g_{11}(s) \left(1 + \frac{1}{sT_{i_1}}\right)$$

Equation (2) now gives

$$K_1 = \frac{1}{|l_{11}(i\omega_c)|}$$

where $|l_{11}(i\omega_c)|$ is obtained from the Bode diagram of $l_{11}(s)$. Control design for other input/output pairings is performed similarly. Apart from MATLAB functions already mentioned, the following ones can help you: `tf, zoom, append, inv` and `feedback`.

The problems below should be solved both for the minimum phase and non-minimum phase case. It is suitable to start with the former.

Exercise 3.2.1. Design a decentralized controller by pairing inputs and outputs according to the RGA analysis. The intended phase margin is $\varphi_m = \pi/3$ and the cross-over frequency ω_c is 0.1 rad/s for the minimum phase case and 0.02 rad/s for the non-minimum phase case. (To make sure that the problem is correctly solved, investigate the Bode diagram of L.)

. . . .

. . . .

Exercise 3.2.2. Calculate the singular values of the sensitivity function

$$S(s) = (I + G(s)F(s))^{-1}$$

and the complementary sensitivity function

$$T(s) = (I + G(s)F(s))^{-1}G(s)F(s)$$

Is the design good with respect to sensitivity and robustness?

. . . .

. . . .

Exercise 3.2.3. Simulate the closed-loop system in Simulink by typing `closedloop`. A Simulink window will appear where the block diagram is shown. Make sure that the variables `F` and `G` in the MATLAB work-space contain the controller and the process respectively. Go to the Simulation menu and click Start. On the screen the unit step responses from the references to the outputs $y_1(t)$ (at $t = 100$) and $y_2(t)$ (at $t = 500$) are plotted together with the inputs. The time instant of the steps can be modified by clicking on the step blocks and changing the Step time. The total simulation time can be modified by changing the Stop time in the menu Simulation/Parameters. Simulation data is saved in the variable `simout` in the MATLAB workspace. Is the control good? Are the outputs coupled?

. . . .

. . . .

Now solve the above problems for the non-minimum phase case.

Exercise 3.2.4. Describe the most important differences between the two cases.

. . . .

. . . .

EE – Automatic Control

Control Theory and Practice
Advanced Course

Computer Exercise:
H_∞ CONTROL DESIGN

Marcus Berner and Peng Zhou, 2008

v.2014

1 Introduction

In H_∞ control design the sensitivity and complementary sensitivity function are shaped to meet certain desired specifications. The main goal of this lab is to get a feeling for how H_∞ control design can be used to obtain desired specifications on sensitivity and robustness.

Preparations: Recollect the following topics from the basic control course:

- Sensitivity and complementary sensitivity functions.
- Robustness and model errors.
- Poles and zeros effect on the system dynamics and the Bode diagram.

Read this document and answer the preparation tasks, 2.1.1, 2.2.1 and 2.3.1 before the exercise.

Presentation: All problems in this exercise should be solved, but only the tasks on the report form should be handed in. The report form and the date when it should be handed in can be found on the course website. The exercise should be performed in pairs of students.

2 Background

This section will begin with a brief summary of some important basic theory followed by more specific theory for the H_∞ design method.

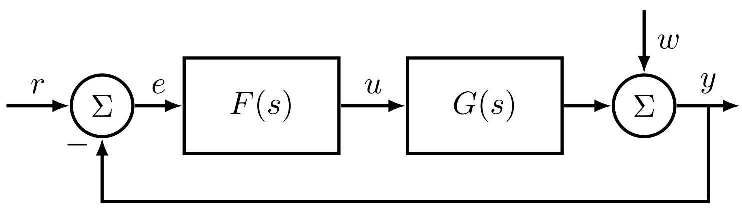


Figure 6: Block diagram of the feedback system used in this lab.

The system used in the entire lab has the structure depicted in fig. 6. The signals indicated in the figure are:

r: Reference value
e: Control error
u: Control signal
w: Disturbance on the output
y: Output

2.1 Sensitivity function and reduction of disturbances

The sensitivity function, denoted S, is the transfer function from the disturbance to the output, see eq. (4). Note that the equation only describes the relation between the disturbance and the output. The reference is therefore assumed to be zero.

$$y = Sw, \quad S = (1 + GF)^{-1} \tag{4}$$

By making the amplification of the sensitivity function small, the effects of disturbances on the output can be reduced. As this lab will show it is not possible to make it arbitrary small for all frequencies. This can easily be realized by looking at the amplification of the controller required to make the sensitivity very small.

2.1.1 Preparation task 1

Use the block diagram in fig. 6 to show that the transfer function from the disturbance (w) to the output (y) satisfies eq. (4).

. . . .

. . . .

. . . .

2.2 Complementary sensitivity and robustness

The complementary sensitivity function T, given by

$$T = (1 + GF)^{-1}GF, \tag{5}$$

can be used to prove robustness to model errors. For that a new system with model uncertainty Δ_G is introduced. The new system is depicted in fig. 7.

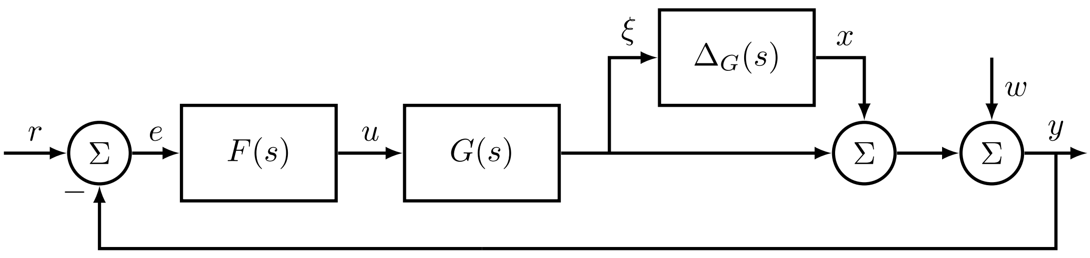


Figure 7: Block diagram of the feedback system with model error

The system in fig. 7 can be rewritten as the system depicted in fig. 8. With the small gain theorem, see the course book [1] or [2], closed loop stability can be guaranteed if Δ_G and T both are stable and condition (6) is satisfied.

$$|T(i\omega)| < \frac{1}{|\Delta_G(i\omega)|}, \quad \forall\omega \tag{6}$$

One important remark about the above result is that the small gain theorem is conservative. The condition on T is therefore sufficient but not necessary for stability.

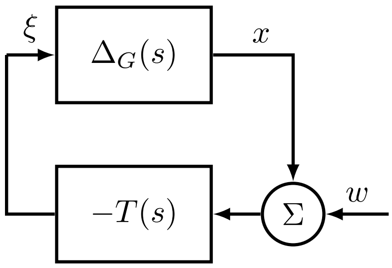


Figure 8: Block diagram of the feedback system with model error on the form used in the small gain theorem

2.2.1 Preparation task 2

Show that the systems in figs. 7 and 8 are equivalent if the reference signal is zero.

. . . .

. . . .

. . . .

. . . .

. . . .

2.3 H_∞ control design

In H_∞ control design the three functions S, T and G_{wu} are shaped to meet desired performance. The first two have already been discussed. Here G_{wu} denotes the transfer function from the disturbance to the control signal, see eq. (7) (where it is assumed $r = 0$).

$$u = G_u w, \quad G_{wu} = -(1 + FG)^{-1}F = -FS \tag{7}$$

It is desirable to make the magnitude of S, T and G_{wu} small. That is unfortunately not possible because they are related to each other. To deal with that, the weights (transfer functions) W_S , W_T and W_U are introduced. They decide how much emphasis to put on minimizing each closed loop transfer function.

After choosing the weights the following problem is solved:

Find F such that

$$\left\| \begin{bmatrix} W_S S \\ W_T T \\ W_U F S \end{bmatrix} \right\|_\infty \leq \gamma$$

with the smallest value of γ possible.

This is an approximate, but computationally convenient, way of expressing the desire that all the individual transfer functions $W_S S$, $W_T T$ and $W_U F S$ should have H_∞-norm less than γ.

The requirement that the individual transfer functions should have H_∞-norm less than γ can be rewritten as

$$\begin{cases} |S(i\omega)| \leq \gamma |W_S^{-1}(i\omega)|, & \forall\omega \\ |T(i\omega)| \leq \gamma |W_T^{-1}(i\omega)|, & \forall\omega \\ |FS(i\omega)| \leq \gamma |W_U^{-1}(i\omega)|, & \forall\omega \end{cases} \tag{8}$$

In this way, we can see the weight functions as a means to define "forbidden regions" for the frequency responses of the individual transfer functions.

To compute the controller that gives the smallest value of γ is far from trivial, especially for higher order systems and weights. It is therefore not done by hand in this lab. The computations are instead done numerically in the design tool.

2.3.1 Preparation task 3

Use the block diagram in fig. 6 to show that the transfer function from the disturbance (w) to the control signal (u) satisfies eq. (7).

. . . .

. . . .

. . . .

3 Software tool

This lab is run in MATLAB. The files needed can be found on the course homepage. A graphical design tool will be used to design the weights and compute the resulting controller. There is also a Simulink model used for simulations. How to use the design tool and do the simulations will be described in this section.

3.1 H_∞ graphical design tool

Before the design tool can be opened a transfer function for the system must be defined. The model has to be strictly stable and proper, which means that is has at least as many poles as zeros and all poles are in the left half plane. An example sequence of how the tool is started can be seen below.

```
s=tf('s');
G=1e4*(s+2)/(s+3)/(s+100)^2;
Hinf(G);
```

The tool should now open and look like in fig. 9.

There are three Bode plots, (1a), (1b), (1c). They show S, T , G_{wu} and the inverse of their respective weights.

To the right of each Bode there are lists of poles and zeros and some buttons, (4) in fig. 9. Notice that the poles corresponds to zeros in the inverse of the weight that is plotted and the opposite for the zeros.

The weights can be changed in two ways. One is to add and remove poles or zeros with the corresponding buttons. The other is to open a graphical editor for the poles and zero. It is done by clicking the 'Edit Pole-Zero diagram" button. A new window, seen in fig. 10, will open. In the new window different tools can be chosen from the toolbar, (1) in fig. 10.

The tools from left to right are:

Add real pole: Add real pole by clicking in the diagram.
Add complex pole pair: Add complex pole pair by clicking in the diagram.
Add real zero: Add real zero by clicking in the diagram.
Add complex zero pair: Add complex zero pair by clicking in the diagram.
Remove: Removes pole or zero by clicking on it.
Move: Move pole or zero by clicking and dragging.

The weights are on the form (9).

$$W(s) = k\frac{(s-z(1))(s-z(2))\dots(s-z(m))}{(s-p(1))(s-p(2))\dots(s-p(n))} \tag{9}$$

To edit the constant k, just type the new value in the gain input field (3) in fig. 9 and press enter.

The weights on T and G_{wu} can be disabled. Just press the disable button (5) in fig. 9. It can be enabled again by pressing the enable button that replaces the disable button.

The magnitude scale is automatically fitted to the weight but the frequency scale has to be set by the user. There are two ways of doing it. One is to enter a maximum and minimum value in the fields (2a) and (2b) in fig. 9 and press enter. The other option is to use the "auto-set frequency" option in the plots menu (6) in fig. 9

To compute the controller from the weights simply use the "compute controller" option in the controller menu, (6) in fig. 9. The controller will then be computed and the plots updated. Some information about the controller can be displayed in MATLAB's command window. The controller can then finally be exported to the workspace with the "export controller" option in the controller menu (6) in fig. 9. It will be saved as F in workspace. If there already exists a variable with that name it will be overwritten.

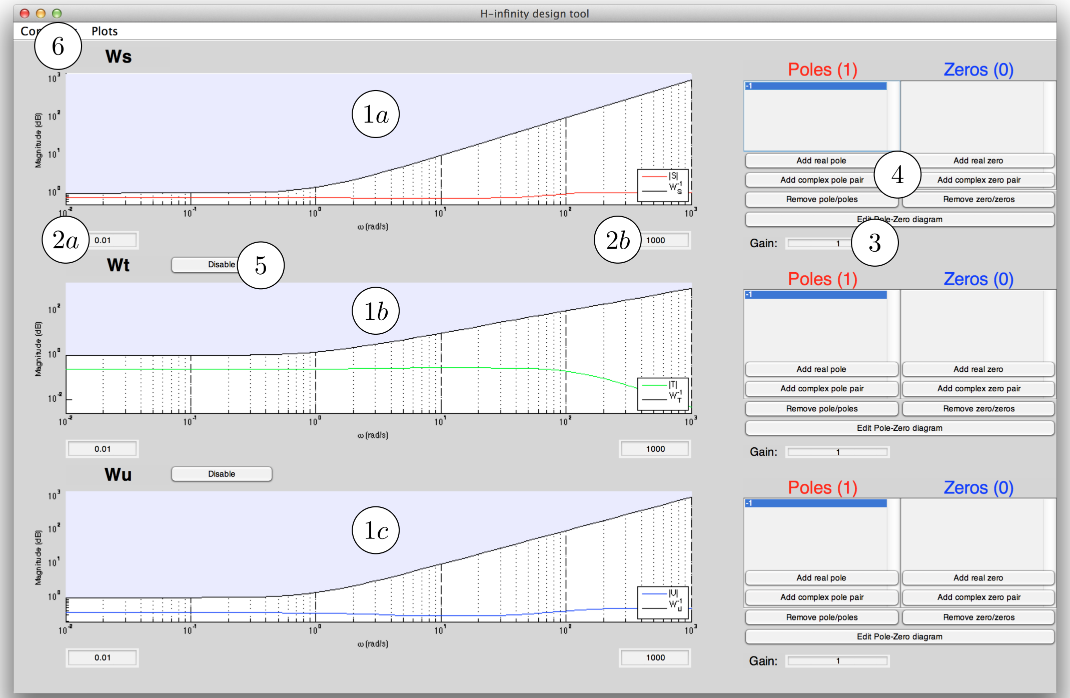


Figure 9: The H_∞ graphical design tool

3.2 Simulations

The Simulink model named `servo1.mdl` is used to simulate the system in fig. 6. A step is used as reference and disturbances can be added as band-limited white noise and a sinusoid. To run the simulation the short `macro` command, found on the homepage, can be used. In the beginning of the file `macro.m` there are some parameters that can be changed to customize the simulation. The parameters are described in the m-file.

To run the macro, the controller and system must first be saved as transfer functions named `Fsim` and `Gsim` in workspace. The macro runs the simulation and plots the results in a new figure.

Below is an example of a sequence that simulates the system. The system and controller are assumed to be defined as transfer functions in workspace with the names `F` and `G`.

```
Fsim=F; Gsim=G ;
% Edit parameters in macro.m
macro
```

Now the simulation should start and plot the results when ready. If the simulation takes very long time it can be stopped by pressing `Ctrl-C` in the MATLAB command window.

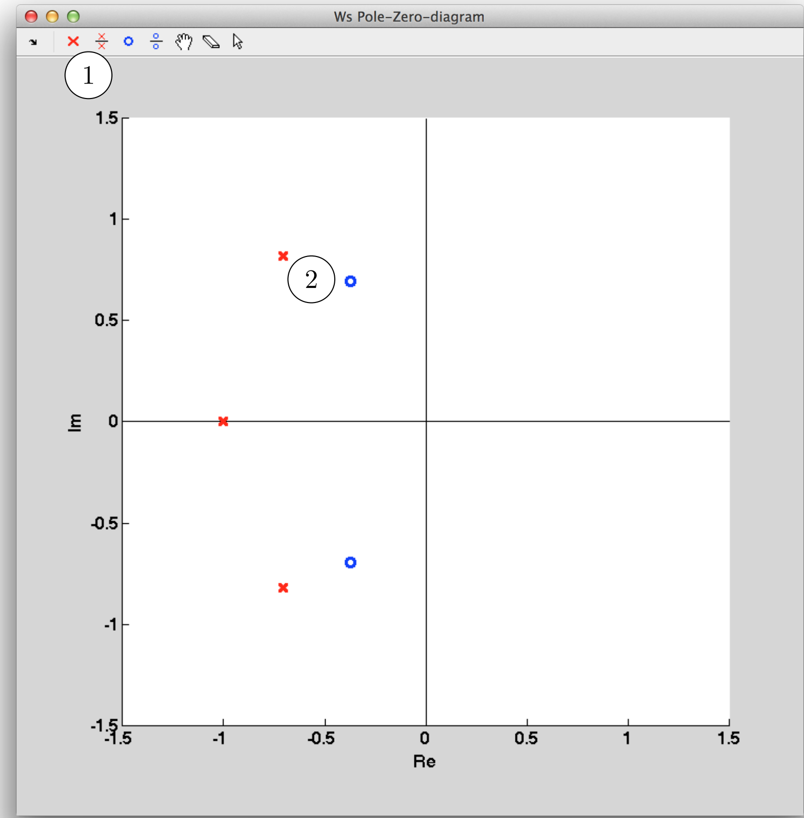


Figure 10: The pole-zero editor

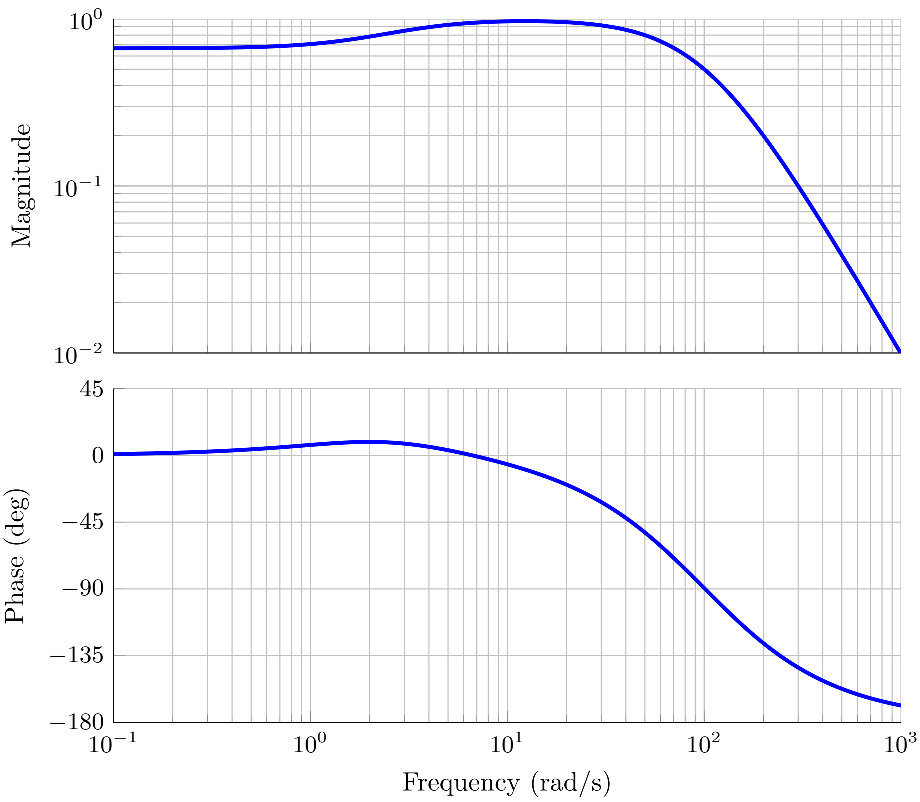


Figure 11: Bode diagram of the system in eq. (10)

4 Exercises

The system to control in this lab is an electrical device powered by the 50 Hz (**NB:** 100π rad/s) power grid. A proposed model of the system is given by the transfer function:

$$G(s) = 10^4 \frac{(s+2)}{(s+3)(s+100)^2} \tag{10}$$

The closed loop system is considered to be fast enough, when it comes to following the reference, without a controller, but not the suppression of disturbances. A controller will therefore be designed focusing on the disturbances.

4.1 Suppression of disturbances

i. The aim is to damp the 50 Hz disturbances. Propose a suitable weight, W_S , by drawing it in the empty Bode diagram in fig. 12. Also draw the expected resulting sensitivity function. Keep in mind that the sensitivity function can not be small for all frequencies.

ii. Now try to design the weight in the software. In this part only the sensitivity is considered. The weights on T and G_{wu} can therefore be disabled. When the

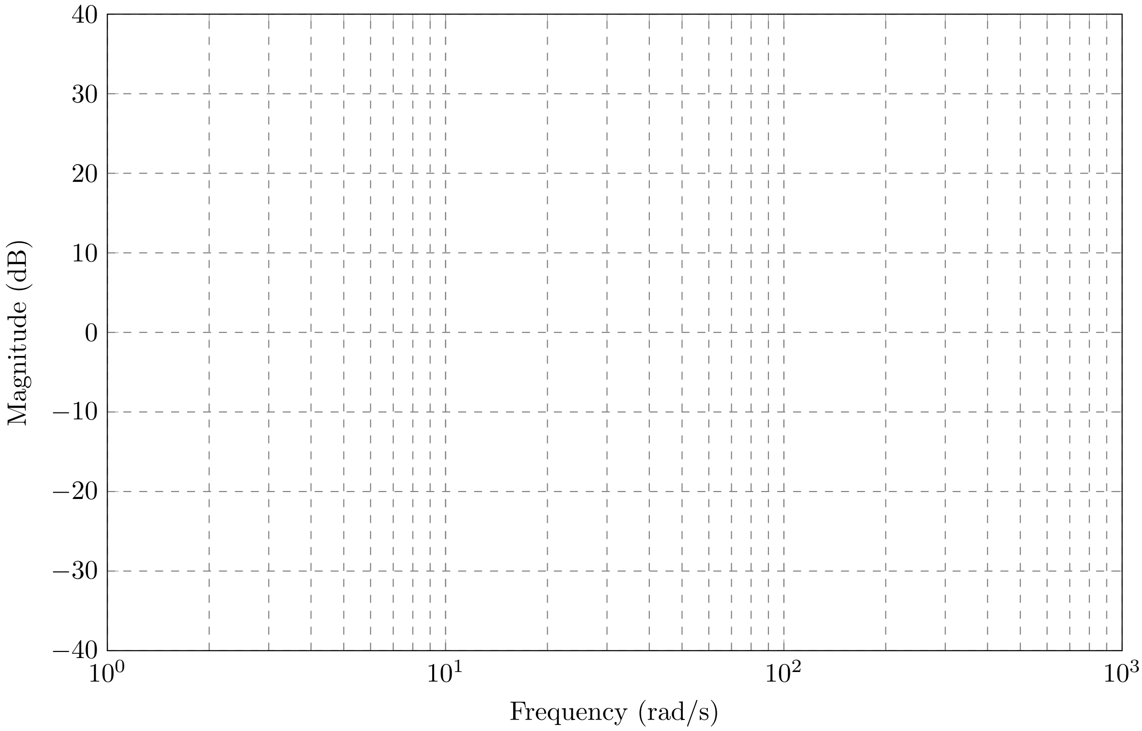


Figure 12: Draw W_S and S here.

sensitivity function is satisfactory, export the controller to workspace and run the simulation with a 50 Hz sinusoidal disturbance as described in section 3.2. Use the parameters in table 1. Fill out table 2 with the results from the simulation.

Hint: Placing poles in $s = -\epsilon \pm i\sqrt{\omega^2 - \epsilon^2}$, where ϵ is small, gives a peak at ω rad/s

Parameter	**Value**
u_max	∞
sin_dist_freq	100π
sin_dist_amp	1
white_noise_amp	0
step_size	0
sim_time	10

Table 1: Parameters to use in `macro.m` for simulation

Signal	**Amplitude**
disturbance (w)	
output (y)	
control signal (u)	

Table 2: Results from simulation in exercise 4.1 ii.

How much is the disturbance damped on the output? ....

(The rate between the disturbance amplitude and the output oscillations)

Approximately what amplification is required for a P-controller to get the same rate and what are the advantages/disadvantages of such a controller?

....

....

....

Hint: If $|FG| \gg 1$, $|S| \approx |FG| - 1$.

4.2 Robustness

Unfortunately the model (10) is not accurate. It was obtained by sending sinusoids, with different frequency, into the system and measure the output amplitude. By ignoring the phase-shift some important dynamics was not detected. The true system is given by eq. (11).

$$G_o(s) = G(s)(1 + \Delta_G(s)) = 10^4 \frac{(s+2)}{(s+3)(s+100)^2} \cdot \frac{(s-1)}{(s+2)} \tag{11}$$

i. What influences will this error have on the system behavior, and will they be a limiting factor on achievable control performance?

....

....

....

ii. Simulate the system with the controller designed in the previous exercise and the system (11). Run the simulation with the same parameters as before.

(The simulation time might need to be increased to see the results)

Comment on the results from the simulations:

....

....

....

....

....

iii. What is the condition on T required to guarantee stability for the new system, using the small gain theorem?

. . . .

. . . .

. . . .

. . . .

. . . .

Is the condition fulfilled? *(Look at the Bode diagram in the graphical design tool.)*

. . . .

iv. Use the software to design a new controller that suppresses 50 Hz disturbances but is stable with G_o. Keep the W_S used earlier but enable the weight on T. Then try to find a weight on T that makes the closed loop system stable and still has good suppression of the disturbance. Export the controller to workspace and run the simulation.

Remark: The controller should be designed for G in eq. (10) *but simulated with G_o in eq.* (11).

Compare the results to table 2.

. . . .

. . . .

. . . .

. . . .

. . . .

4.3 The use of control signal

Enable the weight on G_{wu} and try to reduce the control signal. Try to reduce the amplitude to half of the one used in exercise 4.2 iv. How is the amplitude of the output affected?

. . . .

. . . .

. . . .

Hint: Remember that the tool is solving optimization problem (8).

5 References

[1] Torkel Glad & Lennart Ljung: *Control theory - Multivariable and Nonlinear Methods*, Taylor & Francis (2000)

[2] Torkel Glad & Lennart Ljung: *Reglerteori - Flervariabla och olinjära metoder*, Studentlitteratur (2007)

EE – Automatic Control

Control Theory and Practice
Advanced Course

Computer Exercise: DECOUPLING & GLOVER-MCFARLANE ROBUST LOOP-SHAPING

Anders Hansson, Alf Isaksson och Magnus Jansson, January 1999

v.2014

1 Introduction

The purpose of this computer exercise is to investigate different procedures for multivarible control design. The process is the same as in the computer exercise on multivariable systems, *i.e.* the four-tank process. First, we will investigate static and dynamic decoupling. The control performance will be evaluated. Then, the design will be robustified using a method proposed by Glover and McFarlane.

Preparations: Chapters 8.3 and 10.5 in Glad, T. and Ljung, L.: "Control theory—multivariable and nonlinear methods".

Presentation: All problems in this exercise should be solved, but only the tasks on the report form should be handed in. The report form and the date when it should be handed in can be found on the course website. The exercise should be performed in pairs of students.

2 Theoretical overview

This section provides the theory that you will need to solve the problems. It is based on the course book.

2.1 Decentralized control and decoupling

A fundamental problem in multivariable control is that the input and output signals are coupled. This means that if one input is changed then, in general, all outputs are affected. A measure of the strength of the coupling in a multivariable system ($G(s)$) is given by the *Relative Gain Array*, RGA of the transfer matrix G, defined as:

$$\mathrm{RGA}(G(i\omega)) := G(i\omega). * \left[G^{-1}(i\omega)\right]^T,$$

where ".∗" denotes element wise multiplication. In decentralized control the RGA can help us to determine which inputs and outputs that are suitable to pair. Two rules of thumb:

1. Find a pairing such that the diagonal elements in $\mathrm{RGA}(G(i\omega_c))$ are as close to 1 as possible, where ω_c is the intended cross-over frequency.

2. Avoid pairings that correspond to negative elements in $\mathrm{RGA}(G(0))$.

If it is not possible to find a suitable pairing of inputs and outputs, one can try to make a better system using *decoupling*. Consider the following example: one output is a function of the difference of two inputs, while another output depends on the sum of these two inputs. In this case, it is suitable to introduce two new inputs which denote the sum and the difference respectively of the two original inputs. This is the main idea in decoupling. Generally, decoupling is performed in the following way. Introduce the new variables $\tilde{y} = W_2 y$ and $\tilde{u} = W_1^{-1} u$, so that the transfer function from $\tilde{u}$ to $\tilde{y}$ becomes

$$\tilde{G}(s) := W_2(s)G(s)W_1(s),$$

where we try to design $\tilde{G}$ as diagonal as possible. Typically, we let $W_2 = I$. The idea is to find a $\tilde{G}$ which is more suitable for decentralized control than the original system G. In general, a completely diagonal $\tilde{G}$ is not realizable. However, we can try to design $\tilde{G}$ to be as decoupled as possible in a certain frequency range with the dynamical transfer matrices $W_1(s)$ and $W_2(s)$. Alternatively, we can decouple the system for one frequency, *e.g.* $\omega = 0$ or $\omega = \omega_c$, with constant matrices W_1 and W_2.

2.2 Glover-McFarlane robust loop-shaping

The decentralized control can be robustified using the method proposed by Glover and McFarlane. It is described in Chapter 10.5 in the course book. A summary of the design procedure, step by step, is given below.

1. Start by pairing the input and output signals in such a way that the system becomes as diagonal as possible. A useful mathematical tool is RGA (relative gain array).

2. Design an initial controller using pre-compensation W_1 and post-compensation W_2. To start with, we can typically choose $W_2 = I$ and $W_1 = W_{dc}F_{diag}$ where W_{dc} decouples the system at a suitable frequency (*i.e.* $\omega = 0$ or the intended ω_c) and $F_{diag}(s)$ is a diagonal lead-lag controller designed using classical loop-shaping (*c.f.* computer exercises 1 and 2).

3. Robust stabilization. Design the Glover-McFarlane controller F_r for the system $G_s = W_2 G W_1$. If $\gamma > 4$, redesign W_1.

4. The final controller is $F(s) = W_1 F_r W_2$.

3 Exercises

In this computer exercise the four-tank process will be investigated. Please recall the MATLAB functions introduced in the exercise on multivariable systems.

NB: numerical problems in MATLAB can occur if you work with systems as transfer functions (`tf` objects in MATLAB). It is therefore important that you instead work with state space representations (`ss`). When performing multiplication and division of systems, it is highly recommended to use the function `minreal`, which creates an equivalent system where all canceling pole-zero pairs or non minimal state dynamics are eliminated. Numerical properties can depend on the MATLAB version that you use.

3.1 Static decoupling

Static decoupling is obtained by choosing $W_2(s) = I$ and $W_1(s) = G^{-1}(0)$. This implies that $\tilde{G}(s) = G(s)G^{-1}(0)$ is decoupled at $s = 0$.

The problems below should be solved both for the minimum phase and non-minimum phase case. It is suitable to start with the former.

Exercise 3.1.1. Calculate the static decoupling for the system and plot the Bode diagrams of $\tilde{G}(s)$ for verification.

. . . .

. . . .

Exercise 3.1.2. Design a diagonal controller $\tilde{F}(s)$ for $\tilde{G}(s)$. Design the controllers $\tilde{f}_i(s)$ as PI controllers. The intended phase margin is $\varphi_m = \pi/3$. The intended cross-over frequency ω_c is 0.1 rad/s for the minimum phase case and 0.02 rad/s for the non-minimum phase case.

. . . .

. . . .

The controller is now given by

$$F(s) = G^{-1}(0)\tilde{F}(s)$$

Exercise 3.1.3. Calculate the singular values of the sensitivity function

$$S(s) = (I + G(s)F(s))^{-1}$$

and the complementary sensitivity function

$$T(s) = (I + G(s)F(s))^{-1}G(s)F(s)$$

Is the design good with respect to sensitivity and robustness?

. . . .

. . . .

Exercise 3.1.4. Simulate the closed-loop system in Simulink by typing `closedloop`. A Simulink window will appear where the block diagram is shown. Make sure that the variables `F` and `G` in the MATLAB work-space contain the controller and the process respectively. Go to the Simulation meny and click Start. On the screen the unit step responses from the references to the outputs $y_1(t)$ (at $t = 100$) and $y_2(t)$ (at $t = 500$) are plotted together with the inputs. The time instant of the steps can be modified by clicking on the step blocks and changing the Step time. The total simulation time can be modified by changing the Stop time in the menu Simulation/Parameters. Simulation data is saved in the variable `simout` in the MATLAB workspace. Is the control good? Are the outputs coupled?

. . . .

. . . .

Now perform the exercises above for the non-minimum phase case.

Exercise 3.1.5. Describe the most important differences between the two cases.

. . . .

. . . .

3.2 Dynamical decoupling

Dynamical decoupling can be obtained for example by choosing $W_2(s) = I$ and $W_1(s)$ in such a way that $\tilde{G}(s) = G(s)W_1(s)$ is a diagonal matrix. The conditions for $\tilde{G}(s)$ to be diagonal are the following:

$$g_{11}(s)w_{12}(s) + g_{12}(s)w_{22}(s) = 0$$
$$g_{21}(s)w_{11}(s) + g_{22}(s)w_{21}(s) = 0$$

where $w_{ij}(s)$ denote the elements of the matrix $W_1(s)$. Since there are four unknowns and and two equations we have additional degrees of freedom. A suitable procedure is to let the diagonal elements of $W(s)$ be equal to one if the RGA of $G(s)$ indicates the pairings (u_1, y_1) and (u_2, y_2). Accordingly, for other pairings it is suitable to set the other two elements equal to one. For the case $w_{11}(s) = w_{22}(s) = 1$, we have

$$w_{12}(s) = -g_{12}(s)/g_{11}(s)$$
$$w_{21}(s) = -g_{21}(s)/g_{22}(s)$$

Divisions with `ss` object is not possible in MATLAB. After the divisions have been performed, we can return to state space descriptions using the function `ss`. (Notice that analytically, $\tilde{G}(s)$ is diagonal. However, numerically we can have off diagonal elements of size 10^{-16}, which can cause problems if we work with `tf` objects in MATLAB).

If, by some reason, the static gain of $\tilde{G}(s)$ happens to be negative, this can be modified by changing signs of $W_1(s)$. If $W_1(s)$ becomes non-proper (for example if it contains derivations), we can still realize the dynamical decoupling for frequencies up to approximately 10 times the intended cross-over frequency ω_c using the following modification

$$W_1(s) \leftrightarrow W_1(s)\frac{10\omega_c}{s + 10\omega_c}$$

The problems below should be solved both for the minimum phase and non-minimum phase case. It is suitable to start with the former.

Exercise 3.2.1. Calculate a dynamical decoupling $W(s)$ for the system and plot the Bode diagrams of $\tilde{G}(s)$ for verification.

. . . .

. . . .

Exercise 3.2.2. Design a controller $\tilde{F}(s)$ for $\tilde{G}(s)$. Design the controllers $\tilde{f}_i(s)$ as PI controllers so that we have phase margin $\varphi_m = \pi/3$. The intended cross-over frequency ω_c is 0.1 rad/s for the minimum phase case and 0.02 rad/s for the non-minimum phase case.

. . . .

. . . .

The controller is now given by

$$F(s) = W_1(s)\tilde{F}(s)$$

Exercise 3.2.3. Calculate the singular values of the sensitivity function

$$S(s) = (I + G(s)F(s))^{-1}$$

and the complementary sensitivity function

$$T(s) = (I + G(s)F(s))^{-1}G(s)F(s)$$

Is the design good with respect to sensitivity and robustness?

. . . .

. . . .

Exercise 3.2.4. Simulate the closed-loop system in Simulink. Is the control good? Are the outputs coupled?

. . . .

. . . .

Now solve the above problems for the non-minimum phase case.

Exercise 3.2.5. Describe the most important differences between the two cases.

. . . .

. . . .

Exercise 3.2.6. Which type of decoupling is the best for the minimum phase system and the non-minimum phase system respectively? What are the advantages and disadvantages of the static and dynamical decoupled designs?

. . . .

. . . .

3.3 Glover-McFarlane robust loop-shaping

In the above exercises we combined static and dynamical decoupling with decentralized PI control. In this exercise we will continue with the design that showed to be best for each of the two cases. Alternatively, we could start all over as described above in 2.2, but we will not do that. The reason is that the Glover-McFarlane method works well for reasonably well-tuned nominal controllers.

Therefore assume that we have a nominal loop gain

$$L_0(s) = G(s)W_1(s)\tilde{F}(s)$$

obtained in the exercises above. The Glover-McFarlane method adds a link $F_r(s)$ to this loop gain so that the final controller becomes

$$F(s) = W_1(s)\tilde{F}(s)F_r(s)$$

In MATLAB this link is calculated with the function

```
[Fr,gam] = rloop(L0,alpha)
```

A suitable choice for `alpha` is 1.1.

The problems below should be solved both for the minimum phase and non-minimum phase case. It is suitable to start with the former.

Exercise 3.3.1. Calculate L_0 corresponding to the best previous design procedure and plot the Bode diagrams to verify that L_0 has the intended cross-over frequency ω_c and that it is reasonably decoupled at ω_c. For the minimum phase case, ω_c is 0.1 rad/s and for the non-minimum phase case it is 0.02 rad/s.

. . . .

. . . .

Exercise 3.3.2. Calculate the Glover-McFarlane controller for L_0. Are you satisfied with the γ that you have obtained?

. . . .

. . . .

The controller is now given by

$$F(s) = W_1(s)\tilde{F}(s)F_r(s)$$

Exercise 3.3.3. Calculate the singular values of the sensitivity function

$$S(s) = (I + G(s)F(s))^{-1}$$

and the complementary sensitivity function

$$T(s) = (I + G(s)F(s))^{-1}G(s)F(s)$$

Describe the differences between the robust design and the nominal design in terms of the sensitivity functions. Is the robust design better with respect to sensitivity and robustness?

. . . .

. . . .

Exercise 3.3.4. Simulate the closed-loop system in Simulink. Compare with the result that you obtained simulating the nominal design. What are the differences and similarities?

. . . .

. . . .

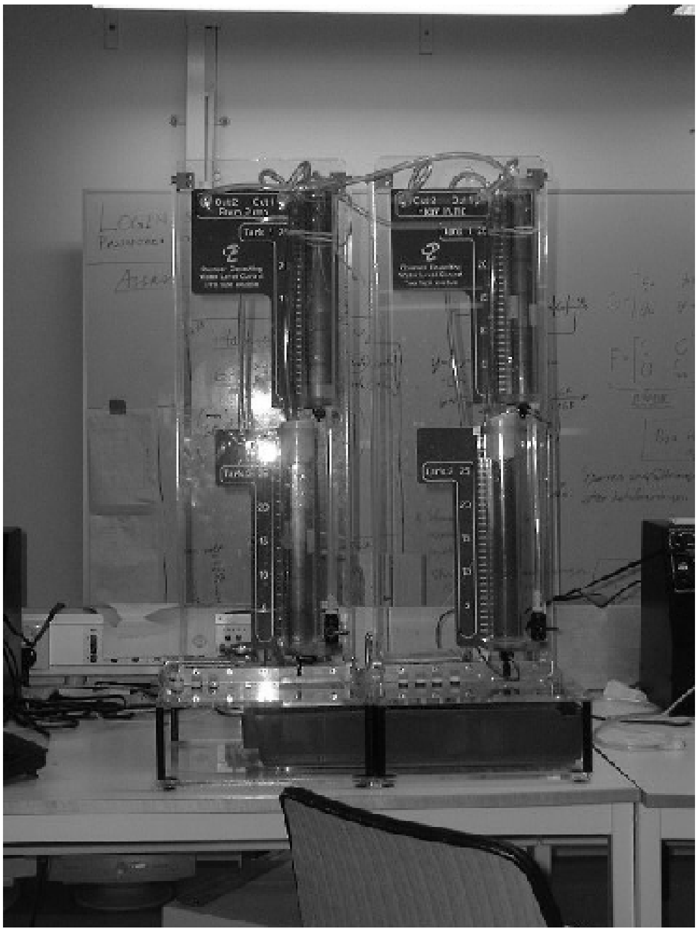
Now solve the above problems for the non-minimum phase case.

Exercise 3.3.5. Describe the most important differences between the two cases.

. . . .

. . . .

EE – Automatic Control



Control Theory and Practice
Advanced Course

Laboratory experiment: THE FOUR-TANK PROCESS

Anders Hansson, Ola Markusson och Magnus Åkerblad, December 1999
Revised by Jonas Wijk, December 2002

v.2014

1 Introduction

In this laboratory experiment we will control the four-tank process, which is a multivariable system. In particular we will investigate interactions between inputs and outputs as well as properties of non-minimum phase dynamics.

The experiment is divided into two occasions. On the first occasion, the modeling and the manual control is performed. This will give you numerical values of some important parameters. Relevant parts of the computer exercises are then repeated, using the identified parameters, to obtain model based controllers. On the second occasion, the control design is evaluated.

Signing up for the laboratory experiment: You should form groups of four students, and then sign up for the laboratory through BILDA. It is important that you sign up for the same four-tank process (A, B, C, D) on both laboratory occasions, since you will use the model obtained on the first occasion for the control design in the second occasion.

Preparations for occasion 1: Chapters 3.3, 3.5, 3.6, 6.5, 7.3-7.5, 7.7, 8.3, 10.5 in Glad, T. and Ljung, L.: "Control theory—multivariable and nonlinear methods". Read the lab instructions and solve the problems that can be solved in advance. Also read the Appendix. **Note: You need to be registered for the course in order to log on to the computers.**

Preparations for occasion 2: Four controllers should be prepared and brought to occasion 2. The controllers can be generated through repeating relevant parts of the computer exercises using the parameters identified during lab occasion 1.

One controller for the minimum phase case and one for the non-minimum phase case. In addition to this, construct a robustified controller (for each cases) using the Glover-McFarlane method. Bring the MATLAB scripts and generate the controllers on the lab computer. On page 8 it is described how to name and save the controllers.

Presentation: All problems in this exercise should be solved and be presented in a written report. The date when the report should be handed in is indicated on the course website. The report should be a full report, contain *abstract* and *conclusions*, relevant figures and tables with *captions* and *legends*, etc. Focus on explaining what you have done, and do not repeat theorems but cite them. Remember to write concise (a report is not a novel), be specific in your writing ("the overshoot is 20%" compared to "the overshoot is not so good"), and check your speling. The report will be graded based on both the content and presentation.

Additional information: Check the course website and the white-board in the lab for additional instructions. Don't forget to save your data to plot the figures for the report.

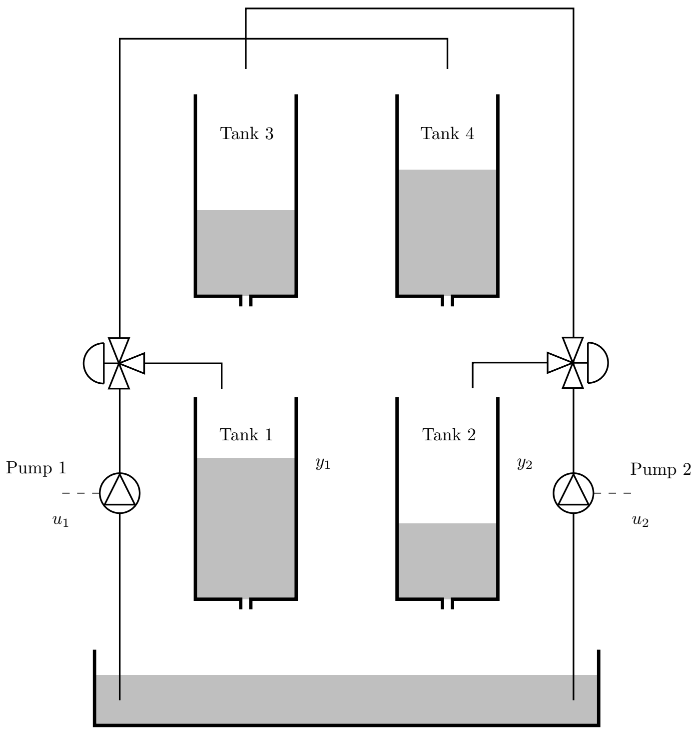


Figure 13: The four-tank process

2 Laboratory occasion 1

In the exercises below a physical model of the four-tank process will be constructed. Then, we will investigate manual control and coupling between the tanks. Performance limitations due to non-minimum dynamics will be investigated. Finally, we will design model based controllers, more specifically decentralized PI control and robust control using the Glover-McFarlane method.

2.1 Modeling

Here the nonlinear differential equations which describe the four-tank process will be derived. The process is shown schematically in Figure 13. For each tank the following relation holds:

$$dV = (q_{in} - q_{out})dt$$

where dV is the change in water volume during the time dt. Divide this equation by dt and assume that $V = Ah$ where A is the cross section area of the tank and h its water level. Then we obtain

$$A\frac{dh}{dt} = q_{in} - q_{out}$$

For the outflow of water, Bernoulli's law holds:

$$q_{out} = a\sqrt{2gh}$$

where a is the cross section area of the outlet hole and $g = 981\text{cm/s}^2$. The flow q generated by a pump is considered proportional to the applied pump voltage u:

$$q = ku$$

where k is the constant. This flow is then divided according to

$$q_L = \gamma ku, \qquad q_U = (1-\gamma)ku, \qquad \gamma \in [0,1]$$

where γ indicates the setting of the valve which is connected to the pump. q_L denotes the flow to the lower tank and q_U is the flow to the upper tank.

Exercise 2.1.1. Show that the following equations describe the water levels in the four tanks.

$$\begin{aligned}
\frac{dh_1}{dt} &= -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} + \frac{\gamma_1 k_1}{A_1}u_1 \\
\frac{dh_2}{dt} &= -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{a_4}{A_2}\sqrt{2gh_4} + \frac{\gamma_2 k_2}{A_2}u_2 \\
\frac{dh_3}{dt} &= -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{(1-\gamma_2)k_2}{A_3}u_2 \\
\frac{dh_4}{dt} &= -\frac{a_4}{A_4}\sqrt{2gh_4} + \frac{(1-\gamma_1)k_1}{A_4}u_1
\end{aligned}$$

where index i in A_i, a_i and h_i refer to tank i and index j in k_j and γ_j refer to pump j and valve j.

....

....

Assume that the levels in the lower tanks are measured by sensors for which the output voltages y_i are proportional to the water levels h_i:

$$y_i = k_c h_i$$

where k_c is a constant.

Exercise 2.1.2. Write down the equations which describe an equilibrium u_1^0, u_2^0, h_1^0, h_2^0, h_3^0, h_4^0, y_1^0, y_2^0 for the tanks.

....

....

Let $\Delta u_i = u_i - u_i^0$, $\Delta h_i = h_i - h_i^0$ and $\Delta y_i = y_i - y_i^0$ denote the deviations from an equilibrium. Introduce

$$u = \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \end{bmatrix}, \quad x = \begin{bmatrix} \Delta h_1 \\ \Delta h_2 \\ \Delta h_3 \\ \Delta h_4 \end{bmatrix}, \quad y = \begin{bmatrix} \Delta y_1 \\ \Delta y_2 \end{bmatrix}.$$

Exercise 2.1.3. Show that the linearized system is given by

$$\frac{dx}{dt} = \begin{bmatrix} -\frac{1}{T_1} & 0 & \frac{A_3}{A_1T_3} & 0 \\ 0 & -\frac{1}{T_2} & 0 & \frac{A_4}{A_2T_4} \\ 0 & 0 & -\frac{1}{T_3} & 0 \\ 0 & 0 & 0 & -\frac{1}{T_4} \end{bmatrix} x + \begin{bmatrix} \frac{\gamma_1 k_1}{A_1} & 0 \\ 0 & \frac{\gamma_2 k_2}{A_2} \\ 0 & \frac{(1-\gamma_2)k_2}{A_3} \\ \frac{(1-\gamma_1)k_1}{A_4} & 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} k_c & 0 & 0 & 0 \\ 0 & k_c & 0 & 0 \end{bmatrix} x$$

where

$$T_i = \frac{A_i}{a_i}\sqrt{\frac{2h_i^0}{g}}.$$

. . . .

. . . .

Exercise 2.1.4. Show that the transfer matrix from u to y is given by

$$G(s) = \begin{bmatrix} g_{11}(s) & g_{12}(s) \\ g_{21}(s) & g_{22}(s) \end{bmatrix} = \begin{bmatrix} \frac{\gamma_1 k_1 c_1}{1+sT_1} & \frac{(1-\gamma_2)k_2 c_1}{(1+sT_3)(1+sT_1)} \\ \frac{(1-\gamma_1)k_1 c_2}{(1+sT_4)(1+sT_2)} & \frac{\gamma_2 k_2 c_2}{1+sT_2} \end{bmatrix}$$

where $c_i = T_i k_c / A_i$.

. . . .

. . . .

Exercise 2.1.5. The zeros of $G(s)$ are given by the zeros of

$$\det G(s) = \frac{k_1 k_2 c_1 c_2}{\Pi_{i=1}^4 (1+sT_i)} \left[(1+sT_3)(1+sT_4) - \frac{(1-\gamma_1)(1-\gamma_2)}{\gamma_1 \gamma_2} \right]$$

Show that $G(s)$ is minimum phase if $1 < \gamma_1 + \gamma_2 \leq 2$ and that $G(s)$ is non-minimum phase if $0 < \gamma_1 + \gamma_2 \leq 1$.

. . . .

. . . .

Exercise 2.1.6. Show that the RGA of $G(0)$ is given by

$$\begin{bmatrix} \lambda & 1-\lambda \\ 1-\lambda & \lambda \end{bmatrix}$$

where $\lambda = \gamma_1\gamma_2/(\gamma_1 + \gamma_2 - 1)$. In the minimum phase case we have $\gamma_1 = \gamma_2 = 0.625$ and in the non-minimum phase case we have $\gamma_1 = \gamma_2 = 1 - 0.625 = 0.375$. Calculate the RGA matrix for both these cases.

. . . .

. . . .

All tanks have cross section area $A = 15.52$ cm^2. However, their effective outlet hole areas vary slightly, and therefore these have to be determined experimentally. We will use different outlet hole sizes in the upper tanks depending on if we are studying the minimum phase or non-minimum phase case. (The outlet holes in the two lower tanks should always have the same size). This means that we will have to determine six (effective) outlet hole areas altogether.

The level sensors have the proportionality constant $k_c = 0.2 \frac{V}{cm}$. For the minimum phase case, $\gamma_1 = \gamma_2 = 0.625$ and for the non-minimum phase case we have $\gamma_1 = \gamma_2 = 1 - 0.625 = 0.375$.

In order to determine the remaining parameters, (a_1 a_2 $a_{3,min}$ $a_{3,nonmin}$ $a_{4,min}$ $a_{4,nonmin}$ k_1 k_2), experiments will be performed.

It is important to prepare proposals for suitable experiments (in order to solve the problems stated below) before performing the laboratory exercise.

Then, in order to perform the experiments, the four-tank process will be set up according to the following instructions:

1. Turn on the computer and login with your KTH-account[1].
2. Connect the minimum phase case according to the instructions in appendix A.2.
3. Double-click the icon “Quadrupletank” at the Desktop.
4. The program starts by asking if any controllers are to be loaded. Answer no by typing “n” and pressing enter.
5. Start the program by pressing the green Start button.
6. Turn on the two UPM’s by pushing the buttons at the back.
7. Choose 50 (% of maximum voltage) of the control signals in the boxes “Control sig. pump 1/2”, and check that water is pumped into all tanks.

Exercise 2.1.7. Propose a suitable experiment[2] to determine k_1 and k_2 $[\frac{cm^3}{s \cdot V}]$, and perform it.

. . . .

. . . .

. . . .

Hint: All measurement are given as percentage of their maximum value.

[1]You need to be registered for the course to get access to the computers.

[2]Tip: there might be air in the tubes, even when the pumps are on. Let the water flow and squeeze the tubes carefully to eliminate the air. A good idea is to run the experiment with u_1=u_2=7.5 V (50% of maximum voltage).

Exercise 2.1.8. Propose a suitable experiment to determine the four effective outlet hole areas a_i for the minimum phase case, and perform it. (In order to save time, we will determine $a_{3,nonmin}$ and $a_{4,nonmin}$ later.)

. . . .

. . . .

. . . .

2.2 Manual control

Solve the problems below for both the minimum phase and the non-minimum phase case. It is suitable to start with minimum phase.

Exercise 2.2.1. Set the pumps on 50 (% of maximum voltage). Wait until stationarity and read the levels on all four tanks from the scale indicated in cm on the four-tank process. Are the levels (fairly) in accordance with the calculations on the equations in Exercise 2.1.2[3]?

. . . .

. . . .

Exercise 2.2.2. Study the step responses (the two outputs) from one input at a time for the two cases (minimum and non-minimum phase). Does the system seem to be coupled? Is this in accordance with the indications of the RGA?

. . . .

. . . .

Exercise 2.2.3. Choose suitable reference levels for the two lower tanks, for example 15 cm (60% of full tank). Try to manually set the pump voltages so that the values displayed on the computer screen become equal to the reference values[4]. How long is the transient time?

Hint: Patience is required for the non-minimum phase case. (If you have not succeeded after 10 minutes, skip it and move on.)

. . . .

. . . .

[3] The level sensors are not calibrated exactly, and therefore the value you read from the scale does not correspond exactly to the value displayed on the screen.

[4] Because of sensitive technique of measurement, it might happen that the signals from the level sensors are subject to small jumps every now and then, so called "offset jumps".

1. Connect the components according to the non-minimum phase case described in appendix A.2.
2. Go back to Exercise 2.1.8 and determine $a_{3,nonmin}$ and $a_{4,nonmin}$.
3. Now repeat the exercises above for the non-minimum phase case.

Exercise 2.2.4. For the above exercises, what are the most important differences between the minimum phase and the non-minimum phase case?

. . . .

. . . .

The first laboratory occasion is now completed, but don't forget to:

1. Log out from the computer, and turn off the UPM:s.
2. Make sure that the laboratory spot is nice and clean.
3. Do the preparation tasks for the next laboratory occasion (they are described in the next section.)

3 Calculation of controllers

Before laboratory occasion 2, four controllers will be calculated.

Exercise 3.1.1. The values of the effective outlet hole areas, the k_i and γ_i (i = 1,2) that you obtained will now be used to calculate controllers for the next laboratory occasion. Therefore, change the values in the files `minphase.m` and `nonminphase.m`. After that, repeat relevant parts of the computer exercises to obtain four controllers. Two controllers for the minimum phase case and two for the non-minimum phase case. In each phase-case, the first controller should be the decentralized controller which you thought was the best when performing the computer exercise. Do not forget to motivate your choice. The second controller should be a robustified Glover-McFarlane controller.

. . . .

. . . .

. . . .

The controllers are saved as `.MAT` files using the function `save` (type `help save` for more information). The files must be named `reg1.MAT`, `reg2.MAT`, `reg3.MAT` and `reg4.MAT` and must contain state space representations of controllers. The state space matrices must be named A, By, C and Dy.

Note: It is preferred, and a huge advantage, to bring a MATLAB script file that generates the controllers in the lab. There are some incompatibilities between different MATLAB versions, and it also lets you recompute the parameters at the second occasion if the four tank process has changed.

If the controller F is available on transfer function form, the following MATLAB code can be used to save it:

```
F=ss(F,'min');
[A,By,C,Dy]=ssdata(F);
save regX.MAT A By C Dy
```

4 Laboratory occasion 2

4.1 Decentralized control

The exercises below require that you have repeated the design process in the computer exercises with the parameters obtained on the previous lab occasion. Make sure that small step responses and load disturbances do not cause saturation of the control signals. Start by setting up the four-tank process in the following way:

1. Use the same laboratory equipment as in laboratory occasion 1 to connect the four-tank process.
2. Turn on the computer and login with your KTH-account[5].
3. Connect the minimum phase case according to the instructions in appendix A.2.
4. Generate and save the four controllers (according to the instructions in section 3).
5. Double-click the icon "Quadrupletank" on the Desktop.
6. The program starts by asking if you want to load controllers. Answer yes by typing "y" and then press enter. Locate and select `reg1.MAT` (the remaining controllers will then load automatically).
7. Start the program by pushing the green Start button.
8. Turn on both UPM:s by pushing the button at the back.
9. Choose 50 (% of maximum voltage) of the control signals in the boxes "Control sig. pump 1/2", and check that water is being pumped into all tanks.

Solve the exercises below for both the minimum phase and non-minimum phase case. It is suitable to start with the minimum phase case.

Exercise 4.1.1. Wait until stationarity. Choose the best decentralized PI controller. Choose Automatic in the popup menu "Operational Mode". Make sure that you work with small deviations from these levels, about 5 percentage points. Investigate the system's response from a step in one of the reference signals. What is the rise time and the overshoot? Also, investigate the system's response to different load disturbances: pour a cup of water in one of the lower tanks; open an extra outlet in one of the upper tanks. How long time does it take for the controller to eliminate the load disturbances?

. . . .

. . . .

. . . .

[5] You need to be registered for the course to get access to the computers.

Now connect the non-minimum phase case according to the instructions in appendix A.2. Then repeat the exercise above for the non-minimum phase case. (Problems can occur when opening an extra outlet in one of the upper tanks. In that case, specify what kind of problems that you get.)

Exercise 4.1.2. For the exercises above, what are the most important differences between the minimum phase and non-minimum phase case?

. . . .

. . . .

. . . .

4.2 Robust control

The exercises below require that you have repeated the design procedures in computer exercise 4 using the parameters obtained previously in this laboratory experiment.

Solve the problems below for both the minimum phase and the non-minimum phase case. It is suitable to start with the latter, since the laboratory equipment now is connected according to the non-minimum phase case.

Exercise 4.2.1. Wait until stationarity. Choose the Glover-McFarlane controller calculated according to the instructions in computer exercise 4. Choose Automatic in the pop up menu "Operational Mode". Make sure that you work with small deviations from these levels. Investigate the system's response to a step in one of the reference signals. What is the rise time and the overshoot? Also, investigate responses from different load disturbances: pour a cup of water in one of the lower tanks; open an extra outlet in one of the upper tanks. How long time does it take for the controller to eliminate the load disturbances?

. . . .

. . . .

. . . .

Exercise 4.2.2. What are the most important differences in performance when comparing the different controllers?

. . . .

. . . .

. . . .

Connect the minimum phase case according to the instructions in appendix A.2. Then repeat the exercises above for the minimum phase case.

Exercise 4.2.3. For the above exercises, what are the most important differences between the minimum phase and the non-minimum phase case?

. . . .

. . . .

The second laboratory occasion is now completed, but don't forget to:

1. Log out from the computer, and turn off the UPM:s.
2. Make sure that the laboratory spot is nice and clean.
3. Write the report!

A Manual for the four-tank process

This manual is divided into four parts. The first part describes how to connect the process. The second describes how to connect the minimum phase and non-minimum phase case. The third part deals with the graphic user interface. The fourth part deals with disconnecting the process.

Note: You will need to perform the steps in appendix A.2.

A.1 How to connect the components of the four-tank process

The four-tank process consists of two double-tank processes connected to each other. The double-tanks are used for the laboratory experiments in the basic control course. Now we will describe how to connect the components of the four-tank process.

- Find the two double-tank processes that you are going to use. (Your group letter decides which two process that you are going to use, see "Signing up for the laboratory experiment" on page 1.) From now on, we will denote these two process "left" and "right", as seen from the front. Carefully remove the blue water bowls. Release the right process by disconnecting the two cords connected to it. Carefully put the right process as close to the left process as possible. (Put both processes at the same table, so that there is no differences in altitude between them). On top of the cupboard at the back of the hall, there is a bigger water bowl which you will place under both the two lower tanks. Fill it with water, almost to the top.

- We will only use the computer of the left process, and we will soon connect the measurement and control signals. However, we will use the UPM:s (Universal Power Module) of both processes. There is an I/O card which belongs to the computer, and you find it at the back of the computer. We will use the card's analogue Input and Output sockets. The left process should already be correctly connected, but to be sure we will verify it. Its control signal [6] should be connected to the card's "Analog Output kanal 0". Its level sensor for the lower tank[7] should be connected to the card's "Analog Input kanal 5". The remaining three level sensor cords should not be connected. Disconnect the cords which are connected to the I/O card of the right computer (not the broad gray flat band cables). Move the UPM of the right process closer to the left one, so that its cords can be connected to the I/O card of the left computer. Connect the control signal of the right process to the card's "Analog Output kanal 1" and the lower tank level sensor of the right process to "Analog input kanal 4". Finally, connect the two non-connected cords [8] from the right UPM to the right process. If you are unsure: check once again that you have connected the process correctly!

[6] The black cord which is connected to "From D/A" at the UPM.

[7] The white socket, (marked with 2), at the broad socket with 4 channels which is connected to "To A/D" at the UPM.

[8] The black cord connected to "To Load" at the UPM and the gray cord.

A.2 Minimum phase and non-minimum phase settings

Depending on which phase case you are working with, you should use different outlet holes of the upper tanks, and connect the cords differently. Below you find a description of this procedure.

For each double-tank process, there are three extra outlet plugs (besides those that are already screwed under the tanks). One located far to the left which has no holes, one in the middle which has a small hole and finally one to the right which has a large hole.

In the minimum phase case, each pump pumps most of the water into "its own" lower tank, and only a smaller fraction of water into the upper tank at the other side. With this setting, we obtain a γ larger than 0.5. In the non-minimum phase case we have the opposite situation, and γ is therefore less than 0.5.

A.2.1 The minimum phase case is connected in the following way

- Put something on top of the two lower tanks, for example a paper towels. (To make sure that nothing falls down into them [9].)
- Use the wrench attached to the process to unscrew the plugs under both the upper tanks. Place each screw among the set of extra screws on each process (to the left of the plug located at the far right side).
- Put the small outlet holes in both the upper tanks. It is **not** necessary to pull tight. (Remember that the processes are fragile.)
- The four tubes, which will be connected according to the instructions below, have to end approximately 27 cm above the bottom of the upper tanks[10]. It is a good idea to pull the tubes through the holes located at the top of the four-tank process, so that the tubes go partly at the back of the process.
- Pull the tube from "Out 1" at the left process to the extra plastic pipe next to tank 3, so that its water falls directly into tank 1. (The numbering of the tanks is given in Figure 1 on page 2.)
- Pull the tube from "Out 2" at the left process to tank 4.
- Pull the tube from "Out 1" at the right process to the extra plastic pipe next to tank 4, so that its water falls directly down in tank 2.
- Pull the tube from "Out 2" at the right process to tank 3.

[9] If anything falls into the tanks, one has to disconnect tubes and cords from that process. Then one has to tilt the process very carefully so that the item that fell into the tank falls out.

[10] If the tubes end at different altitudes, the constants k_1, k_2, γ_1 and γ_2 can be affected, because the driving force acting on the water is changed.

A.2.2 The non-minimum phase setting is obtained in the following way

- Put something on top of the two lower tanks, for example a paper towels. (To make sure that nothing falls down into them).
- Unscrew the plugs located under the two upper tanks. For each process, place the plug among the set of extra screws (to the right of the plug located at the far left side).
- Put the medium size holes in the two upper tanks. It is **not** necessary to pull tight. (Remember that the processes are fragile.)
- The four tubes, which will be connected according to the instructions below, have to end approximately 27 cm above the bottom of the upper tanks. It is a good idea to pull the tubes through the holes located at the top of the four-tank process, so that the tubes go partly at the back of the process.
- Pull the tube from "Out 2" at the left process to the extra plastic pipe next to tank 3, so that its water flows directly into tank 1.
- Pull the tube from "Out 1" at the left process to tank 4.
- Pull the tube from "Out 2" at the right process to the extra plastic pipe next to tank 4, so that its water falls directly into tank 2.
- Pull the tube from "Out 1" at the right process to tank 3.

A.3 The graphic user interface

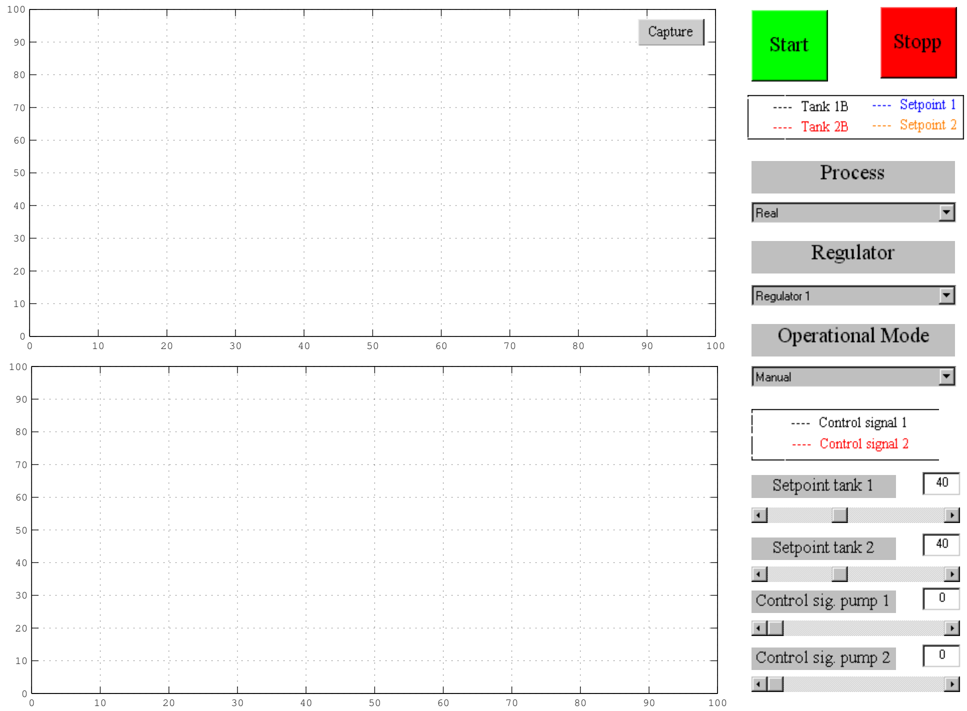


Figure 14: The graphic user interface.

- The interface is opened by double-clicking on the "Quadrupletank" MATLAB icon on the desktop.
- The program starts by asking if you want to load controllers. Choose `yes` or `no` depending on if you have designed controllers or not.
- Click on the green or red button at the top to the right to start or stop the process.
- Click on the popup menu `Regulator` to choose controller. (This popup menu is only displayed if you have loaded controllers.)
- The tank levels and the reference signals are plotted in the upper graph, and the control signals are plotted in the lower. The upper graph is scaled in percentage of full tank, so that 100 corresponds to 25 cm. The lower graph is scaled in percentage of maximal control signal, so that 100 corresponds to 15 V.
- In the popup menu `Operational Mode` you can switch between manual and automatic control.
- In order to change the manual control signal you could either pull the handle or type directly in the box.

- To change the reference value you can either pull the handle or type directly in the box.
- By right-clicking at an axis, a dialog box for zooming is opened. You can also use the capture button to study and zoom collected data.
- The `Time Offset` at the bottom to the left shows the time between 0 and the time displayed on the x axis. To obtain the true time at the x axis, you therefore add the offset value. (The time $t = 0$ is the time when you start the program using the green start button.)
- The Capture button is used to study collected data during operation. Click on the button to obtain a figure with reference signals and measured signals. (Use the zoom tool in the figure menu to zoom).
- Save data by clicking on the capture button. Then you obtain a box where you can save data. The data (time, measured signals, reference signals and control signals) is saved as `data.mat`. If you already have a file with that name, the name becomes `data1.mat` etc up to `data3.mat`. (After that you have to use another folder). The data is loaded into MATLAB with the function `load dataX.mat`.

A.4 How to disconnect the four-tank process

When disconnecting the four-tank process it is very important that the middle size holes are screwed in the upper tanks. Check this and therefore go through the items below:

1. Empty the large water bowl and put it on top of the cupboard at the back of the hall.
2. Disconnect the tubes between two double-tank processes.
3. Make sure that no tube is connected to “Out 2”.
4. Connect a tube to “Out 1”, put it through some hole so that it comes out at the back of the process, and then through the hole which is just above the upper tank. Then put the tube into the upper tank. Do this for both processes.
5. Remove the cables of the right UPM from the I/O card of the left computer. Also, disconnect the right process by disconnecting the two cords connected to it.
6. Put back each process (with its UPM) at its original spot and put a blue water bowl under each double-tank process.
7. Connect the level sensors to each tank[11] at the card’s “Analog Input kanal 4” and “Analog Input kanal 5” respectively. (The yellow at 4 and the white at 5.)
8. Connect each control signal [12] to “Analog Output kanal 0” at each I/O card.
9. Finally, connect the two non-connected cords[13] from the right UPM to the right process.

[11] The yellow and white sockets, (marked with 1 and 2 respectively), at the broad cord with 4 channels connected to “To A/D” at the UPM

[12] The black cord connected to “From D/A” at the UPM.

[13] The black cord connected to “To Load” at the UPM and the gray cord.