

Miroslav Fikar
Martin Klaučo
Radoslav Paulen

Theory of Automatic Control I
Practice Examples

STU in Bratislava, 2022

© prof. Ing. Miroslav Fikar, DrSc.,
Ing. MSc. Martin Klaučo, PhD.,
doc. Ing. Radoslav Paulen, PhD.

Reviewers: prof. Alena Kozáková, PhD., FEI STU
doc. Ing. Martin Gulán, PhD., SJF STU

Approved by the administration of FCFT STU in Bratislava.

ISBN 978-80-8208-085-1

Contents

- Foreword** **11**

- 1 Mathematical Modelling of Continuous-Time Systems** **13**
 - 1.1 Overview 13
 - 1.1.1 Linear Approximations of Nonlinear Systems 14
 - 1.2 Examples 16
 - 1.2.1 Liquid Level in a Tank 16
 - 1.2.2 Two Tanks with Interaction 20
 - 1.2.3 Two Tanks with a Pump 25
 - 1.2.4 Two Tanks in Series 27
 - 1.3 Practice Examples 29
 - 1.3.1 A One-Dimensional Car 29
 - 1.4 Problems 32
 - 1.5 MATLAB Snippets 36
 - 1.6 Python Snippets 37
 - 1.7 Bibliography 38

- 2 Input-Output Characteristics of Dynamic Systems** **39**
 - 2.1 Overview 39
 - 2.1.1 Transfer Functions 39
 - 2.1.2 The Algebra of Transfer Functions 40
 - 2.1.3 Converting State-Space Representations to Transfer-Functions Representations 41
 - 2.1.4 Converting Transfer-Function Representations to State-Space Representations 41
 - 2.2 Examples 43

2.2.1	Conversion of Second-Order Transfer Functions to the Controllable Canonical Form	43
2.2.2	Two Tanks with Interaction	46
2.2.3	Two Tanks with a Pump	49
2.3	Practice Examples	49
2.3.1	A One-Dimensional Car	49
2.3.2	Two Tanks in Series	51
2.4	Problems	57
2.5	MATLAB Snippets	60
2.6	Python Snippets	60
2.7	Bibliography	61
3	Poles, Zeros, and Their Effects on Process Dynamics	63
3.1	Overview	63
3.1.1	Step Response of a First-Order System	65
3.1.2	Effects of an Additional Pole on the Step Response	68
3.1.3	Effects of an Additional Zero on the Step Response	70
3.1.4	Step Response of a Second-Order System	71
3.2	Examples	74
3.2.1	Two Tanks with Interaction	74
3.2.2	Two Tanks with a Pump	76
3.3	Practice Examples	77
3.3.1	An Introductory Example	77
3.3.2	Two Tanks in Series	78
3.3.3	Mass on a Spring	80
3.4	Problems	82
3.5	MATLAB Snippets	85
3.6	Python Snippets	85
3.7	Bibliography	86
4	Frequency-Domain Characteristics of Dynamic Systems	87
4.1	Overview	88
4.1.1	Complex Numbers, Variables, and Functions	88
4.1.2	Frequency Responses and Characteristics	90
4.1.3	Stability Margins	91
4.2	Examples	92

4.2.1	Two Tanks with Interaction	97
4.2.2	Two Tanks with a Pump	99
4.3	Practice Examples	100
4.3.1	First-Order System	100
4.3.2	A One-Dimensional Car	103
4.3.3	Two Tanks in Series	104
4.4	Problems	105
4.5	MATLAB Snippets	107
4.6	Python Snippets	108
4.7	Bibliography	109
5	Control Performance	111
5.1	Overview	111
5.1.1	Time-Domain Specifications	111
5.1.2	Integral Criteria	112
5.1.3	Frequency-Domain Indices	113
5.1.4	Pole Placement	115
5.2	Examples	117
5.3	Practice Examples	121
5.3.1	Two Tanks in Series	121
5.4	Problems	129
5.5	MATLAB Snippets	129
5.6	Python Snippets	130
5.7	Bibliography	131
6	Closed-Loop Systems and PID Controllers	133
6.1	Overview	133
6.1.1	A Closed-Loop System	133
6.1.2	PID Controller	134
6.2	Examples	139
6.2.1	Two Tanks with Interaction	141
6.2.2	Two Tanks with a Pump	144
6.3	Practice Examples	145
6.3.1	First-Order System Control	145
6.3.2	Two Tanks in Series	148
6.4	Problems	151

6.5	MATLAB Snippets	152
6.6	Python Snippets	153
6.6.1	PI Control of Two Tanks with Interaction	153
6.7	Bibliography	155
7	Root-Locus Analysis and Design	157
7.1	Overview	157
7.1.1	General Features of a Root-Locus Plot	159
7.2	Examples	160
7.2.1	Construction of Root-Locus Plots	160
7.2.2	Two Tanks with Interaction	164
7.2.3	Two Tanks with a Pump	168
7.3	Practice Examples	169
7.3.1	Warm-up Examples	169
7.3.2	Two Tanks in Series: Root-Locus Construction	170
7.3.3	A One-Dimensional Car: Root-Locus Construction	171
7.3.4	Poles and Zeros of a PID Controller	172
7.3.5	A One-Dimensional Car: PID Root-Locus Design	173
7.3.6	Two Tanks in Series: PID Root-Locus Design	174
7.4	Problems	178
7.5	MATLAB Snippets	181
7.6	Python Snippets	182
7.7	Bibliography	183
8	PID Tuning	185
8.1	Overview	185
8.1.1	Experimental Methods	186
8.1.2	Model-Based Methods	188
8.2	Examples	192
8.3	Practice Examples	194
8.3.1	PID Controller Design for Two Tanks in Series	194
8.3.2	PID Controller Design for Two Tanks with a Pump	199
8.3.3	Fan-Flex System – Flexy2	202
8.4	Problems	205
8.5	MATLAB Snippets	206
8.6	Python Snippets	206

8.7 Bibliography 207

Bibliography **209**

Foreword

This textbook is mainly intended for undergraduate students in engineering in cybernetics at the Faculty of Chemical and Food Technology at the Slovak University of Technology in Bratislava. This text is used in seminars of the course *Theory of Automatic Control I* in the first year of engineering (master's) studies. It can also be of interest to engineers or applied scientists. It is focused on problems in the theory of automatic control, process dynamics, control performance, and controller design.

The mathematical level of the covered topics is oriented to students in chemical engineering with knowledge of mass and energy balances and modelling, differential equations, Laplace transform, that is, those who have completed an introductory course in cybernetics or process control.

The contents of the book can be divided into two major parts. The first part considers process modelling, time-domain, and frequency-domain properties. Process modelling is based on dynamic mass and energy balances for processes that are typical in chemical, food, and bio-engineering. These models can further be simplified to linear state-space and input-output representations that are sufficiently precise for controller design. The course and the book concentrate on single-input, single-output (SISO) continuous-time input-output models, while state-space representations, discrete-time, multivariate systems are considered in subsequent courses.

The properties of input-output models described by transfer functions include time-domain characteristics resulting from process dynamics represented by poles and zeros as well as frequency-domain characteristic, the Bode and Nyquist plots, and relative stability.

The second part of the book considers closed-loop systems: feedback control, its components, and definition of control performance indices. The most common controller type, a PID controller, is investigated together with its properties, issues, and practical aspects. Special focus is on the root-locus method, *i.e.*, characterisation of the closed-loop pole locations. Finally, selected controller design procedures are explained and compared.

The structure of each chapter is uniform: it starts with a list of new skills and knowledge that the students will acquire. Then, theoretical foundations are explained in an engineering fashion and are mostly declarative. It is assumed that a rigorous theoretical treatment is presented elsewhere – selected references are mentioned at the end of the chapter.

Theoretical concepts are further elaborated using solved examples and simulations. Two nonlinear process models are used throughout the book: two tanks with interaction as an example of a nonlinear stable process, and two tanks with a pump that constitute an example of an open-loop unstable process.

The third part of each chapter contains practice examples. These are intended to be solved in seminars with teaching assistants. Both simulations and real-time experiments are used. We provide two editions of the book: while the print version contains practice example with solutions, the electronic version can provide practice examples without solution so that students can investigate various paths towards the solutions and later compare their approaches with the ones in the book. Also, solved practice examples can be a base for work in later chapters.

Each chapter contains problems where solutions are provided without explanation. These can serve for self-study and usually examine some broader topics.

Each chapter concludes with small fragments of code from MATLAB and Python. These are meant to complement chapter content and to help in implementation of concrete topics.

The textbook is complemented with other learning materials. The youtube section for students at UIAM web page: <https://www.uiam.sk> contains links and metadata to lecture videos available at youtube channel of the institute: <https://www.youtube.com/c/UIAMFCHPTSTU>. The complete course lives at our Moodle server: <https://elearn.uiam.sk> and is available using guest access.

Material for the publication evolved for more than a decade. We would like to thank to many colleagues and doctoral students that contributed in some way and shaped the material. We were heavily influenced by Profs. Mikleš and Mészáros who had taught this course at our faculty in the past. We acknowledge assistance from Prof. Kvasnica, assoc. Prof. Bakošová and also our present colleagues at the Institute of Information Engineering, Automation and Mathematics for the collegial atmosphere. We would specifically wish to thank to Prof. Shardt from TU Ilmenau for his careful proofreading, suggestions to improve consistency, spelling, and language, and many constructive remarks. We also thank the reviewers, Prof. Alena Kozáková and assoc. Prof. Martin Gulán for their detailed notes and opinions that further greatly improved the manuscript.

Bratislava, July 2022

authors

Acknowledgement to Slovak grant agencies: the textbook would not appear without the support of the Scientific Grant Agency of the Slovak Republic under the grants 1/0585/19, 1/0545/20, 1/0691/21, 1/0297/22 and of the Slovak Research and Development Agency under the grants APVV-15-0007, APVV-20-0261, APVV-21-0019.

CHAPTER 1

Mathematical Modelling of Continuous-Time Systems

In this chapter, you will deepen your knowledge and learn new skills with the following goals:

- the importance and techniques of mathematical modelling of processes,
- the characterisation of processes and definition of corresponding systems,
- the description of systems evolving in time using differential equations,
- the linearisation of nonlinear models using the Taylor expansion,
- the determination of equilibrium points and steady states,
- the characterisation of linear state-space systems,
- the creation of block diagrams of systems and subsystems,
- the simulation of systems described by differential equations.

We will use process modelling and simulations of controlled processes for the study of their properties, for the synthesis of controllers, and for the characterisation of the properties of the resulting closed-loop system.

1.1 Overview

Mathematical models are constructed from the mass and energy balances. Dynamic mathematical models are used to describe the behaviour of phenomena (processes) that have some internal inertia. Such models consist of differential equations.

A mass balance of a process with time-varying inputs can be formulated as:

$$\left(\begin{array}{c} \text{sum of the mass flow rates} \\ \text{entering the system} \end{array} \right) = \left(\begin{array}{c} \text{sum of the mass flow} \\ \text{leaving the system} \end{array} \right) + \left(\begin{array}{c} \text{rate of the mass} \\ \text{accumulation in the system} \end{array} \right).$$

An analogous formulation for energy balances is:

$$\left(\begin{array}{c} \text{sum of the energy fluxes} \\ \text{entering the system} \end{array} \right) = \left(\begin{array}{c} \text{sum of the energy fluxes} \\ \text{leaving the system} \end{array} \right) + \left(\begin{array}{c} \text{rate of the energy} \\ \text{accumulation in the system} \end{array} \right).$$

If chemical reactions are concerned, mass and/or energy can seemingly appear or disappear and other (source and sink) terms have to be considered.

The accumulation term can be represented as a time derivative of the total mass or energy in the system.

In the next step, the model is analysed, its constants and parameters are specified. The model can be solved if the number of equations is equal to the number of unknown variables. All other variables have to be specified as known inputs: some of them as manipulated variables that can be changed by our interventions; others as variables that act as disturbances.

1.1.1 Linear Approximations of Nonlinear Systems

The majority of processes in the chemical and food industries are modelled by nonlinear models. However, if they operate in the vicinity of the steady state, where process variables change only a little and deviations from the steady states are small, nonlinear behaviour can be approximated by linear models. Such approximations can be obtained using Taylor expansion to give only linear terms. Such models are more suitable for control design as there exists a broad variety of control design methods for linear systems.

The linearity of a system is defined by studying its inputs and outputs (responses). A linear system has to obey the principle of superposition and homogeneity. If the response of the system to the input of magnitude u_1 is y_1 and the response to the input of magnitude u_2 is y_2 then the principle of superposition states that the response to the input of magnitude $u_1 + u_2$ is $y_1 + y_2$. Similarly, if the input is multiplied by a constant k to become ku_1 then the output is ky_1 , which is the principle of homogeneity.

In general, a function $f(x)$ is linear with respect to the variable x if its first-order derivative with respect to x is a constant.

For example the function $f(x) = 3x$ is linear with respect to x as its derivative is a constant $df(x)/dx = 3$. Functions $f(x) = 3x^2$, $f(x) = \sqrt{x}$, and $f(x) = 4/x^2$ are all nonlinear.

When considering a multivariate function, such as $f(x_1, x_2) = a\sqrt{bx_1} + cx_2^2$ (with a, b, c being constant parameters), is linear with respect to x_1 and nonlinear with respect to x_2 .

A general linearisation procedure for models described by a nonlinear differential equation consists of the following steps:

1. Given a nonlinear mathematical model described by a differential equation:

$$\frac{dz(t)}{dt} = \dot{z}(t) = f(z(t), v(t)). \quad (1.1)$$

2. Find the steady-state form of the model (or in general in some equilibrium-state form):

$$\frac{dz^s}{dt} = f(z^s, v^s) = 0. \quad (1.2)$$

Steady-state values of the time-varying variables will be denoted with the superscript "s" as v^s .

3. Define the deviation variables as the difference between the time-varying variables and their steady-state values. The usual notation uses $u = v - v^s$ for the deviation in the manipulated input, d for the deviation disturbance input, $x = z - z^s$ for the deviation state, and y for the deviation in the output. Subtract the approximated nonlinear model and the steady-state model, that is,

$$\frac{dz(t)}{dt} - \frac{dz^s}{dt} = \frac{d(z(t) - z^s)}{dt} = f(z, v) - f(z^s, v^s). \quad (1.3)$$

4. Approximate all nonlinear terms on the right-hand side of (1.1) using the first-order Taylor expansion about the steady state. For functions of a single variable $f(v)$ this is of the form

$$f(v) = f(v^s) + \left. \frac{df(v)}{dv} \right|_{v=v^s} (v - v^s), \quad (1.4)$$

and for functions of two variables $f(v, z)$, this is

$$f(v, z) = f(v^s, z^s) + \left. \frac{\partial f(v, z)}{\partial v} \right|_{v=v^s, z=z^s} (v - v^s) + \left. \frac{\partial f(v, z)}{\partial z} \right|_{v=v^s, z=z^s} (z - z^s), \quad (1.5)$$

and similarly for functions of multiple variables.

5. Replace the original variables with their corresponding deviational variables.

A general linearised mathematical model in the deviation variables with n dimensional vector of states $\mathbf{x}(t)$ (with initial condition $\mathbf{x}_0 = 0$), m dimensional vector of inputs $\mathbf{u}(t)$ and r dimensional vector of outputs $\mathbf{y}(t)$ can then be characterised by the state and output equations

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{x}(0) = \mathbf{0}, \quad (1.6)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \quad (1.7)$$

with matrices $\mathbf{A}[n, n]$, $\mathbf{B}[n, m]$, $\mathbf{C}[r, n]$, and $\mathbf{D}[r, m]$.

Matrices \mathbf{A} and \mathbf{B} can also be directly obtained from the original nonlinear model with inputs $\mathbf{v}(t)$ and states $\mathbf{z}(t)$ around some operating point $\mathbf{z}^s, \mathbf{v}^s$

$$\dot{\mathbf{z}}(t) = \mathbf{f}(\mathbf{z}(t), \mathbf{v}(t)), \quad (1.8)$$

as follows:

$$\mathbf{A} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{z}} \right|_{\mathbf{z}=\mathbf{z}^s}, \quad \mathbf{B} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{v}} \right|_{\mathbf{v}=\mathbf{v}^s}, \quad (1.9)$$

with deviational variables $\mathbf{x}(t) = \mathbf{z}(t) - \mathbf{z}^s$ and $\mathbf{u}(t) = \mathbf{v}(t) - \mathbf{v}^s$.

In the same manner, if there is a measured output $\mathbf{w}(t)$ with relation to the states and the inputs of the form

$$\mathbf{w}(t) = \mathbf{g}(\mathbf{z}(t), \mathbf{v}(t)), \quad (1.10)$$

matrices \mathbf{C} and \mathbf{D} are

$$\mathbf{C} = \left. \frac{\partial \mathbf{g}}{\partial \mathbf{z}} \right|_{\mathbf{z}=\mathbf{z}^s}, \quad \mathbf{D} = \left. \frac{\partial \mathbf{g}}{\partial \mathbf{v}} \right|_{\mathbf{v}=\mathbf{v}^s}, \quad (1.11)$$

and the deviational output is defined as $\mathbf{y}(t) = \mathbf{w}(t) - \mathbf{w}^s$.

Example. 1.1: Linearisation

Consider a dynamic mathematical model described by the differential equation of the form

$$\frac{dh(t)}{dt} = -h(t) + h(t)q(t) + q(t), \quad h(0) = h^s = 2. \quad (1.12)$$

Find a linearised state-space model using the general linearisation procedure.

The corresponding steady state is obtained by adding superscript “s” to all time-dependent variables

$$\frac{dh^s}{dt} = -h^s + h^s q^s + q^s. \quad (1.13)$$

Note, that the left-hand side term $\frac{dh^s}{dt}$ is equal to zero as the steady-state value h^s of the state variable does not depend on time. This means there is no accumulation of mass in the system.

Subtract the nonlinear and steady-state models

$$\frac{d(h(t) - h^s)}{dt} = -(h(t) - h^s) + h(t)q(t) - h^s q^s + (q(t) - q^s). \quad (1.14)$$

We can define deviation variables $x(t) = h(t) - h^s$, $u(t) = q(t) - q^s$. The term $h(t)q(t)$ is nonlinear and will be linearised using the Taylor expansion

$$\begin{aligned} h(t)q(t) &= h^s q^s + \left. \frac{\partial(hq)}{\partial h} \right|_{h=h^s, q=q^s} (h(t) - h^s) + \left. \frac{\partial(hq)}{\partial q} \right|_{h=h^s, q=q^s} (q(t) - q^s) \\ &= h^s q^s + q|_{h=h^s, q=q^s} (h(t) - h^s) + h|_{h=h^s, q=q^s} (q(t) - q^s) \\ &= h^s q^s + q^s (h(t) - h^s) + h^s (q(t) - q^s). \end{aligned} \quad (1.15)$$

The last equation is linear as q^s and h^s are constants. Substituting back into (1.14) gives

$$\begin{aligned} \frac{d(h(t) - h^s)}{dt} &= -(h(t) - h^s) + q^s (h(t) - h^s) + h^s (q(t) - q^s) + (q(t) - q^s), \\ \frac{dx(t)}{dt} &= -x(t) + q^s x(t) + h^s u(t) + u(t), \quad x(0) = 0. \end{aligned} \quad (1.16)$$

The linearised model has the initial condition equal to zero ($x(0) = 0$) as the deviation variable definition gives $x(0) = h(0) - h^s = 0$.

1.2 Examples

1.2.1 Liquid Level in a Tank

We want to study the dynamic properties of a tank with vertical walls and a cross-sectional area F that holds a liquid with the height (liquid level) $h(t)$ and the volume $V(t) = Fh(t)$. A model of the system is depicted in Figure 1.1. The liquid flows into the tank from its top. The volumetric flow rate of this stream $q_0(t)$ is adjustable by a valve. The exiting stream leaves

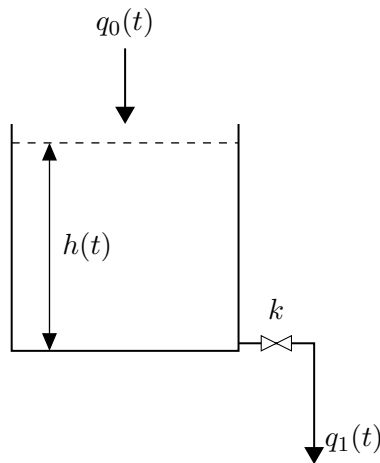


Figure 1.1: A schematic model of a tank.

the tank at the bottom of the tank and its flow rate is driven by gravity, *i.e.*, according to the Torricelli's law $q_1(t) = k\sqrt{h(t)}$, where k is the valve constant. We can assume that the density of the liquid ρ is constant.

The values of parameters and signals of the process at the initial time are $q_0(0) = q_0^s = 0.3 \text{ m}^3/\text{s}$, $h(0) = 0 \text{ m}$, $F = 0.5 \text{ m}^2$, and $k = 1.15 \text{ m}^{2.5}/\text{s}$.

1.2.1.1 Select the input and the state variables of the above described system.

Input variables: liquid inlet flow q_0 , state variables: height of the liquid in the tank $h(t)$.

1.2.1.2 Write the (dynamic) mass balance of the liquid in the tank and express it in terms of the input and state variables.

The mass balance of the system gives:

$$\frac{dm(t)}{dt} = m_0(t) - m_1(t), \quad (1.17)$$

where $m(t)$ is the mass of the liquid in the tank and $m_0(t)$ and $m_1(t)$ are the mass flows of the inflowing and the outflowing liquid, respectively.

We can further write, using the volume V and volumetric flow rates q_0, q_1 :

$$\frac{d\rho V(t)}{dt} = \rho q_0(t) - \rho q_1(t). \quad (1.18)$$

Using $V(t) = Fh(t)$ and re-arranging, we get

$$\frac{dh(t)}{dt} = \frac{q_0(t)}{F} - \frac{k}{F}\sqrt{h(t)}. \quad (1.19)$$

1.2.1.3 Find the steady-state value h^s of the liquid level in the tank for the given input q_0^s .

$$0 = \frac{q_0^s}{F} - \frac{k}{F} \sqrt{h^s} \quad \rightarrow \quad h^s = \left(\frac{q_0^s}{k} \right)^2. \quad (1.20)$$

1.2.1.4 If necessary, linearise the model.

The square root of the state variable constitutes a nonlinearity of the model. For the linearisation, we first define the deviational variables:

$$x(t) = h(t) - h^s, \quad u(t) = q_0(t) - q_0^s. \quad (1.21)$$

The linearised model is then given by:

$$\frac{dx(t)}{dt} = ax(t) + bu(t), \quad (1.22)$$

where

$$a = \left. \frac{\partial \left(\frac{q_0(t)}{F} - \frac{k}{F} \sqrt{h(t)} \right)}{\partial h(t)} \right|_{h^s, q_0^s} = -\frac{k}{2F\sqrt{h^s}}, \quad b = \left. \frac{\partial \left(\frac{q_0(t)}{F} - \frac{k}{F} \sqrt{h(t)} \right)}{\partial q_0(t)} \right|_{h^s, q_0^s} = \frac{1}{F}. \quad (1.23)$$

1.2.1.5 Create a Simulink model to simulate the nonlinear and linearised models for some given values of the initial conditions, parameters of the model, and inputs.

Simulation models of nonlinear and linearised models are built from the basic building blocks (e.g. gain, multiply, divide, sum, integrate, and square root) are shown in Figures 1.2 and 1.3, respectively. Another possibility is to use the S-function approach, where the process differential and output equations are specified directly.

1.2.1.6 Embed the nonlinear and linear systems as subsystems. Run the simulations and compare the responses of the nonlinear and linearised models.

The simulation for both nonlinear and linearised models is shown in Figure 1.4.

Assuming that the Scope block is configured to save all signals to a variable `h`, we can then use the following MATLAB snippet to plot the results, save them, and to compare the simulated liquid level responses in the tank (Figure 1.5, left).

```
plot(h(:,1), h(:,2), h(:,1), h(:,3), 'linewidth', 2);
xlabel('time [s]'); ylabel('level h(t) [m]'); grid on
legend('nonlinear', 'linear', 'location', 'southeast');
iam_save_pdf('h_t.pdf');
```

1.2.1.7 Run the same simulation as in the previous step. After $t = 2$ s, change the input to $q_0 = 0.2 \text{ m}^3/\text{s}$. Compare the responses of the nonlinear and linearised models.

The results are shown in Figure 1.5 (right).

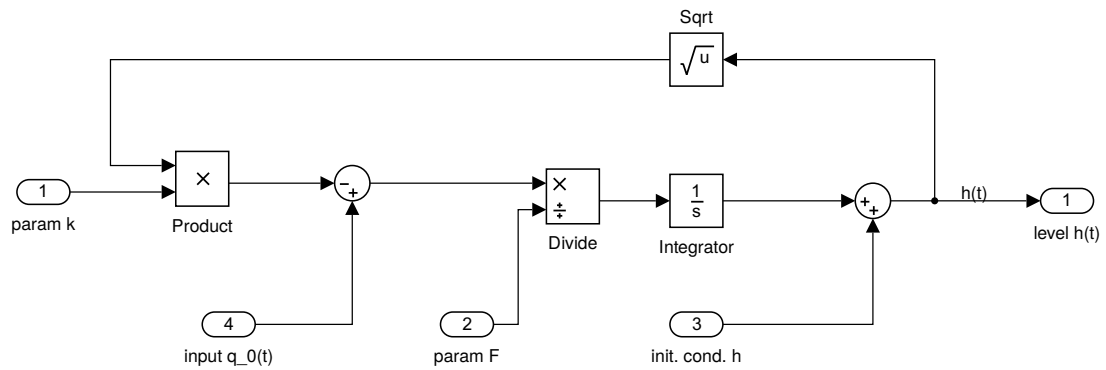


Figure 1.2: Simulink model of the nonlinear model of the tank.

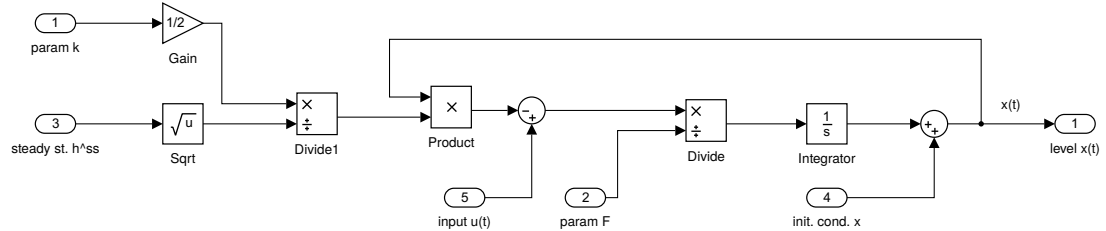


Figure 1.3: Simulink model of the linearised model of the tank.

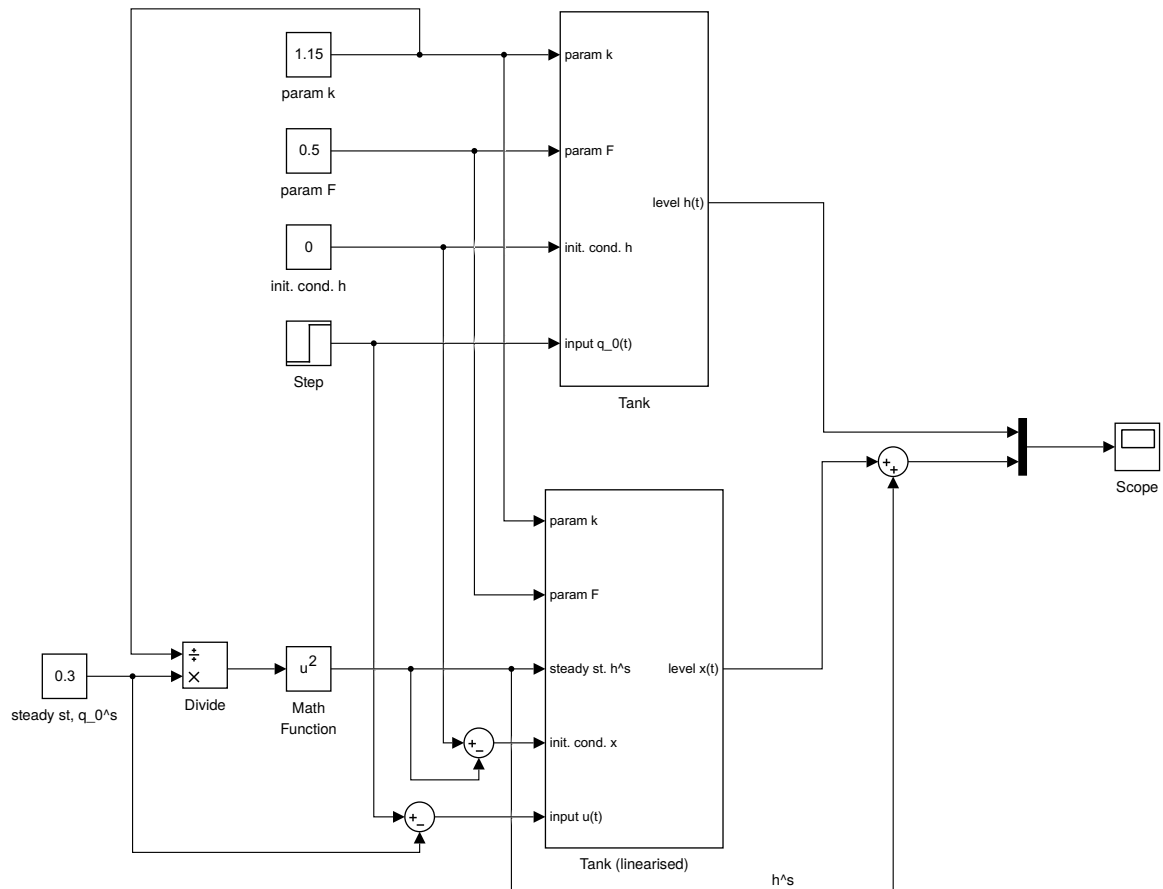


Figure 1.4: Simulation model of the nonlinear and linearised models of the tank.

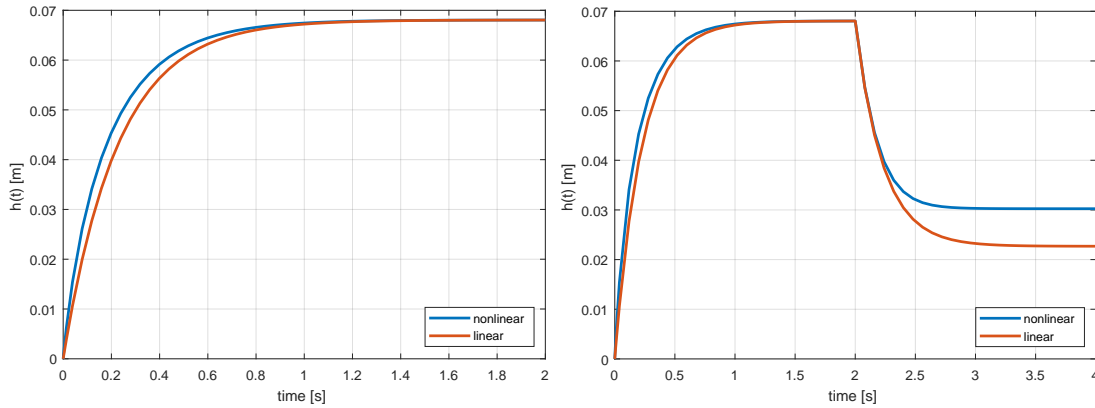


Figure 1.5: Comparison of the simulated levels of nonlinear and linearised models of the tank. (left) The first simulation (constant input and empty tank) and (right) the second simulation (continuation with a step change of the input at $t = 2$ s).

1.2.1.8 Plot and calculate steady-state values of the linearised model for a range of input flows $q_0^s \in [0, 2]$, if the operating point is $q_0^s = 0.3 \text{ m}^3/\text{s}$ with the corresponding level steady state $h^s = 0.068 \text{ m}$. Compare the results with the steady states of the nonlinear model for specific values of the input flow $q_0^s = (0.2, 0.3, 0.6, 0.9)$.

Steady-state values are calculated from the steady-state input/output characteristics (also called the static characteristics) defined by (1.20). To calculate the steady-state values of the linearised model, we need to obtain the linearised counterpart of the steady-state characteristics. The linearised characteristics can be obtained by applying the first order Taylor expansion from Section 1.1.1, namely the relations (1.4) to (1.20). The linearised version of the static characteristics is given as

$$h_L^s(q_0) \approx \left(\frac{q_0^s}{k}\right)^2 + \left.\frac{d\left(\frac{q_0}{k}\right)^2}{dq_0}\right|_{q_0=q_0^s} (q_0 + q_0^s), \quad (1.24)$$

which gives

$$h_L^s(q_0) \approx 0.454q_0 + 0.068. \quad (1.25)$$

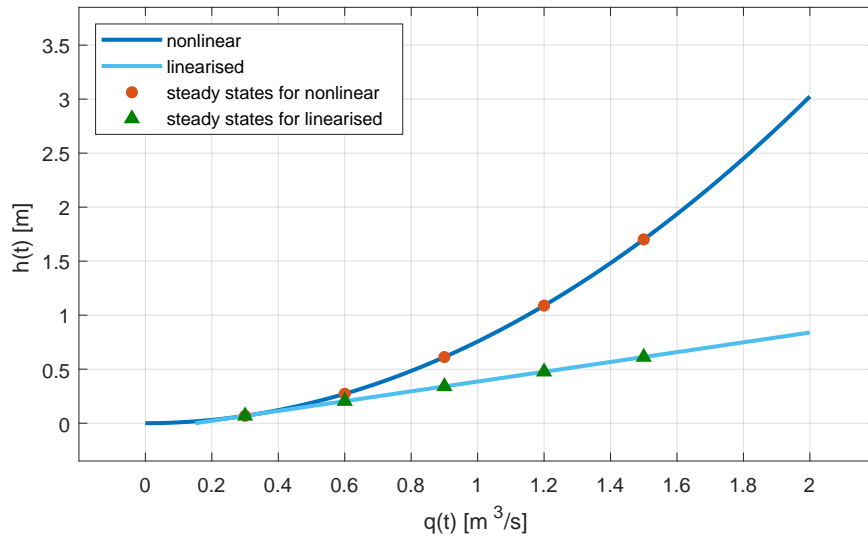
Once we plot the nonlinear static characteristics along with the linearised version, we can understand the impact of the linearisation on the potential differences between the linear and nonlinear dynamic models. The visual comparison is shown in Figure 1.6. The differences in the steady state of the linear and nonlinear models are given in Table 1.1. Note, that values for the input $q_0^s = 0.2 \text{ m}^3/\text{s}$ can also be observed in Figure 1.5 which compares the simulated dynamic responses.

1.2.2 Two Tanks with Interaction

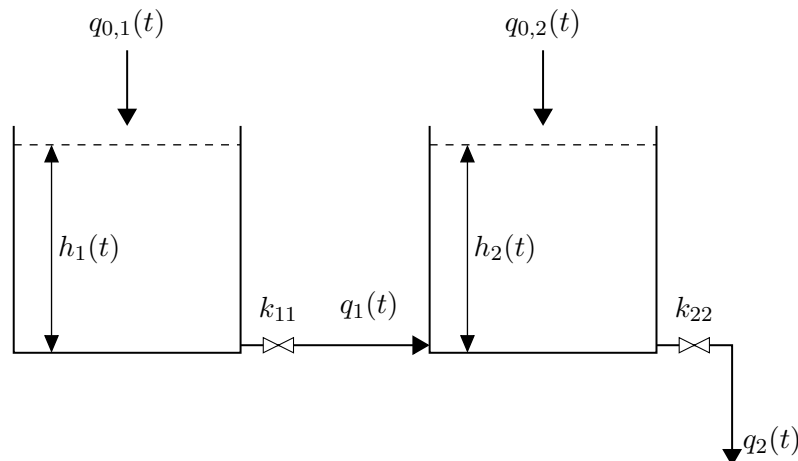
We study a process consisting of two tanks with interaction (see Figure 1.7). Each tank is structurally equivalent to the tank described in Section 1.2.1. Therefore, the process parameters include the cross-sectional areas F_1, F_2 , the valve constants k_{11}, k_{22} , and the density ρ . There are two adjustable liquid streams entering the tanks at their top: $q_{0,1}(t)$ to the first tank and $q_{0,2}(t)$

Table 1.1: Comparison of the steady-state values.

q_0^s	h^s	h_L^s
0.2	0.0302	0.0227
0.3	0.0681	0.0681
0.6	0.2722	0.2042
0.9	0.6125	0.3403

**Figure 1.6:** Comparison of the static characteristics of nonlinear and linearised models of a single tank system.

to the second tank. The exiting stream $q_1(t)$ leaves the first tank at the bottom and its flow rate is driven by gravity and difference between the levels in the tanks, *i.e.*, according to the Torricelli's law $q_1(t) = k_{11}\sqrt{h_1(t) - h_2(t)}$. The exiting stream $q_2(t)$ leaves the second tank at the bottom and its flow rate is given by $q_2(t) = k_{22}\sqrt{h_2(t)}$. We assume that we can only measure the level in the second tank and that the levels are initially at some steady state h_1^s, h_2^s .

**Figure 1.7:** A model of two tanks with interaction.

The values of parameters and variables of the process are $q_{0,1}^s = 0.9 \text{ m}^3/\text{s}$, $q_{0,2}^s = 0 \text{ m}^3/\text{s}$, $F_1 = 0.5 \text{ m}^2$, $F_2 = 0.6 \text{ m}^2$, $k_{11} = 0.8 \text{ m}^{2.5}/\text{s}$, and $k_{22} = 0.5 \text{ m}^{2.5}/\text{s}$.

1.2.2.1 Define a system and select its input, state, and output variables.

Input variables: liquid inlet flows $q_{0,1}(t)$, $q_{0,2}(t)$, state variables: heights of the liquid in the tanks $h_1(t)$, $h_2(t)$, output variables: measured height $h_2(t)$.

1.2.2.2 Write down the (dynamic) mass balances of the liquid in the tank and express it in terms of the input and state variables.

The mass balance of the system is

$$\frac{d\rho F_1 h_1}{dt} = \rho q_{0,1} - \rho q_1 = \rho q_{0,1} - \rho k_{11} \sqrt{h_1 - h_2}, \quad (1.26)$$

$$\frac{d\rho F_2 h_2}{dt} = \rho q_{0,2} + \rho q_1 - \rho q_2 = \rho q_{0,2} + \rho k_{11} \sqrt{h_1 - h_2} - \rho k_{22} \sqrt{h_2}, \quad (1.27)$$

$$\frac{dh_1}{dt} = \frac{q_{0,1}}{F_1} - \frac{k_{11}}{F_1} \sqrt{h_1 - h_2}, \quad h_1(0) = h_1^s, \quad (1.28)$$

$$\frac{dh_2}{dt} = \frac{q_{0,2}}{F_2} + \frac{k_{11}}{F_2} \sqrt{h_1 - h_2} - \frac{k_{22}}{F_2} \sqrt{h_2}, \quad h_2(0) = h_2^s. \quad (1.29)$$

Note, that this mathematical model implicitly assumes that $h_1 > h_2$ and that the liquid flows from the first tank to the second one. If this assumption is not valid (for example if $q_{0,1} = 0$ and $q_{0,2} > 0$), the model needs to be modified using the information on the sign of the expression $h_1 - h_2$.

$$s_h = \text{sgn}(h_1 - h_2) = \begin{cases} -1 & \text{if } h_1 - h_2 \geq 0, \\ 1 & \text{if } h_1 - h_2 < 0, \end{cases} \quad (1.30)$$

$$\frac{dh_1}{dt} = \frac{q_{0,1}}{F_1} - s_h \frac{k_{11}}{F_1} \sqrt{|h_1 - h_2|}, \quad h_1(0) = h_1^s, \quad (1.31)$$

$$\frac{dh_2}{dt} = \frac{q_{0,2}}{F_2} + s_h \frac{k_{11}}{F_2} \sqrt{|h_1 - h_2|} - \frac{k_{22}}{F_2} \sqrt{h_2}, \quad h_2(0) = h_2^s. \quad (1.32)$$

However, we will assume for further derivations that $h_1 > h_2$.

1.2.2.3 Derive the steady-state values of the process.

$$0 = \frac{q_{0,1}^s}{F_1} - \frac{k_{11}}{F_1} \sqrt{h_1^s - h_2^s}, \quad \Rightarrow \quad h_1^s = \left(\frac{q_{0,1}^s}{k_{11}} \right)^2 + h_2^s, \quad (1.33)$$

$$0 = \frac{q_{0,2}^s}{F_2} + \frac{k_{11}}{F_2} \sqrt{h_1^s - h_2^s} - \frac{k_{22}}{F_2} \sqrt{h_2^s}, \quad \Rightarrow \quad h_2^s = \left(\frac{q_{0,1}^s + q_{0,2}^s}{k_{22}} \right)^2. \quad (1.34)$$

1.2.2.4 Linearise the model if necessary.

Let us first decide whether linearisation is necessary: if the derivatives of the right-hand sides of the differential equations with respect to the state and input

variables are constant, the model is linear. Otherwise, it is nonlinear.

We define the deviational variables for the inputs, states, and outputs as

$$x_1(t) = h_1(t) - h_1^s, \quad x_2(t) = h_2(t) - h_2^s, \quad y_1 = x_2, \quad (1.35)$$

$$u_1(t) = q_{0,1}(t) - q_{0,1}^s, \quad u_2(t) = q_{0,2}(t) - q_{0,2}^s. \quad (1.36)$$

The model is nonlinear as differentiation of the square root terms with respect to tank levels is not constant. Linearisation of the nonlinear terms:

$$k_{11}\sqrt{h_1 - h_2} = k_{11}\sqrt{h_1^s - h_2^s} + \frac{k_{11}}{2\sqrt{h_1^s - h_2^s}}(h_1 - h_1^s) - \frac{k_{11}}{2\sqrt{h_1^s - h_2^s}}(h_2 - h_2^s) \quad (1.37)$$

$$= k_{11}\sqrt{h_1^s - h_2^s} + k_1(h_1 - h_1^s) - k_1(h_2 - h_2^s), \quad (1.38)$$

$$k_{22}\sqrt{h_2} = k_{22}\sqrt{h_2^s} + \frac{k_{22}}{2\sqrt{h_2^s}}(h_2 - h_2^s) = k_{22}\sqrt{h_2^s} + k_2(h_2 - h_2^s), \quad (1.39)$$

$$k_1 = \frac{k_{11}}{2\sqrt{h_1^s - h_2^s}}, \quad k_2 = \frac{k_{22}}{2\sqrt{h_2^s}}. \quad (1.40)$$

Then, the resulting linear equations are

$$\frac{dx_1}{dt} = \frac{1}{F_1}u_1 - \frac{k_1}{F_1}x_1 + \frac{k_1}{F_1}x_2, \quad x_1(0) = h_1(0) - h_1^s = 0, \quad (1.41)$$

$$\frac{dx_2}{dt} = \frac{1}{F_2}u_2 + \frac{k_1}{F_2}x_1 - \frac{k_1 + k_2}{F_2}x_2, \quad x_2(0) = h_2(0) - h_2^s = 0. \quad (1.42)$$

1.2.2.5 Define a linear state-space model, that is, matrices \mathbf{A} , \mathbf{B} , \mathbf{C} , \mathbf{D} according to Equations (1.6) and (1.7)

State-space description:

$$\mathbf{A} = \begin{pmatrix} -\frac{k_1}{F_1} & \frac{k_1}{F_1} \\ \frac{k_1}{F_2} & -\frac{k_1 + k_2}{F_2} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \frac{1}{F_1} & 0 \\ 0 & \frac{1}{F_2} \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} 0 & 1 \end{pmatrix}, \quad \mathbf{D} = 0. \quad (1.43)$$

1.2.2.6 Build a Simulink model from the basic building blocks to simulate the evolution of the liquid heights for some given values of the initial conditions and inputs. The Simulink model should compare the nonlinear model with its linear approximation.

The simulation models are shown in Figs. 1.8 and 1.9.

1.2.2.7 Run the simulation from the steady-state conditions. Change $q_{0,1} = 0.9q_{0,1}^s$ after the first 5 seconds. How do you expect the tank levels to behave? Plot the simulation results as state and input variables vs. time.

As the flow $q_{0,1}$ is reduced, we expect that tank levels will decrease as well. This change will be slower in the second tank as the tank level in the first tank will not decrease immediately. The results are shown in Figure 1.10.

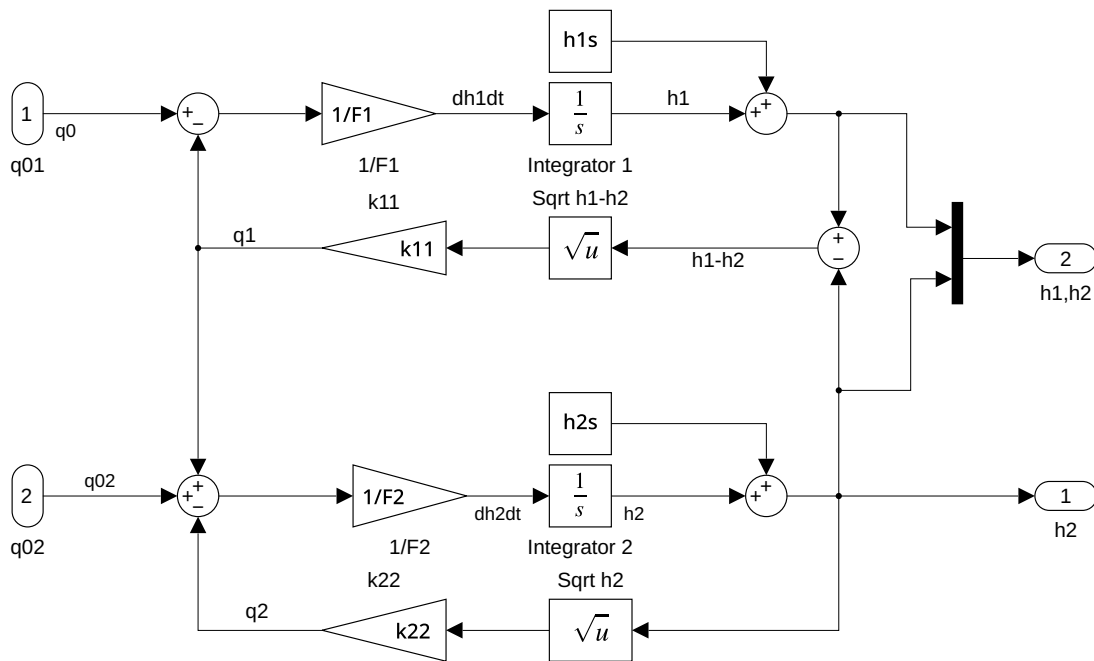


Figure 1.8: Simulink model of the nonlinear model of two tanks with interaction.

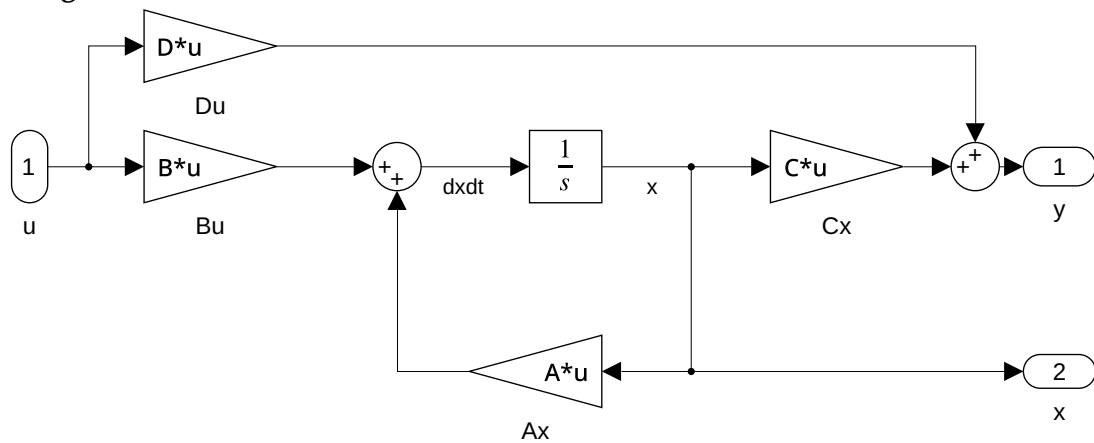


Figure 1.9: Simulink model of a linear state-space model described by matrices A, B, C, D .

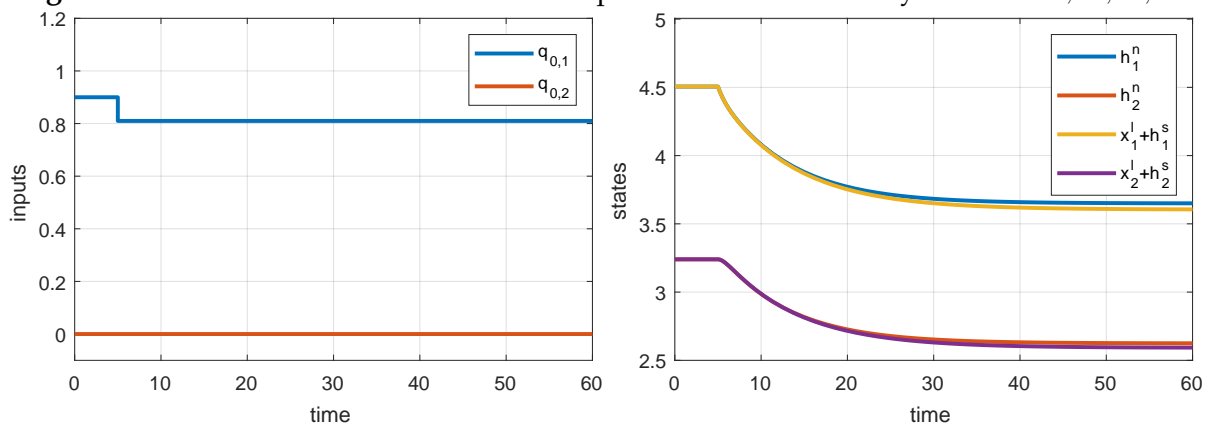


Figure 1.10: Comparison of the simulated levels of nonlinear and linearised models for two tanks with interaction (left – inputs, right – outputs).

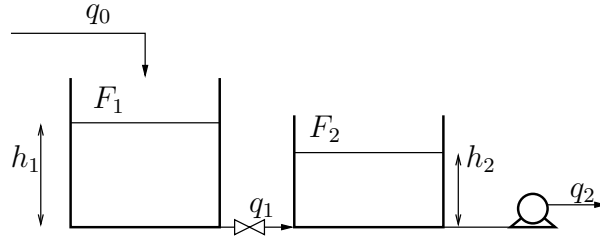


Figure 1.11: The model of two tanks with a pump.

1.2.3 Two Tanks with a Pump

We study the dynamic properties of the process again consisting of two tanks with interaction. However, now the exiting stream leaving the second tank at the bottom of the tank with flow rate q_2 is determined by a pump (see Figure 1.11).

Values of parameters and variables of the process are $F_1 = 0.5 \text{ m}^2$, $F_2 = 0.6 \text{ m}^2$, $k_{11} = 0.8 \text{ m}^{2.5}/\text{s}$, $q_0(0) = 0.9 \text{ m}^3/\text{s}$, $q_2(0) = q_0(0)$, $h_1(0) = 4.506 \text{ m}$, and $h_2(0) = 3.24 \text{ m}$.

1.2.3.1 Define a system and select its input, state, and output variables.

Input variables: liquid inlet flow q_0 , liquid outlet flow q_2 , state variables: heights of the liquid in the tanks $h_1(t)$, $h_2(t)$, output variables: measured height $h_2(t)$.

1.2.3.2 Write down the (dynamic) mass balances of the liquid in the tank and express it in terms of the input and state variables.

The mass balances of the system yield:

$$\frac{d\rho F_1 h_1}{dt} = \rho q_0 - \rho q_1 = \rho q_0 - \rho k_{11} \sqrt{h_1 - h_2}, \quad (1.44)$$

$$\frac{d\rho F_2 h_2}{dt} = \rho q_1 - \rho q_2 = \rho k_{11} \sqrt{h_1 - h_2} - \rho q_2, \quad (1.45)$$

$$\frac{dh_1}{dt} = \frac{q_0}{F_1} - \frac{k_{11}}{F_1} \sqrt{h_1 - h_2}, \quad h_1(0) = h_{1,0}, \quad (1.46)$$

$$\frac{dh_2}{dt} = \frac{k_{11}}{F_2} \sqrt{h_1 - h_2} - \frac{q_2}{F_2}, \quad h_2(0) = h_{2,0}. \quad (1.47)$$

1.2.3.3 Discuss the steady-state behaviour of the process.

It is not possible to find a steady-state value of h_2^s . In fact, arbitrary value of h_2 can stay constant as long as $q_1 = q_2$. The value of h_1^s can be obtained from

$$q_0^s = q_1^s = q_2^s = k_{11} \sqrt{h_1^s - h_2^s}, \quad h_1^s = h_2^s + \left(\frac{q_0^s}{k_{11}} \right)^2. \quad (1.48)$$

The process model includes a simple integrator.

1.2.3.4 Linearise the model if necessary.

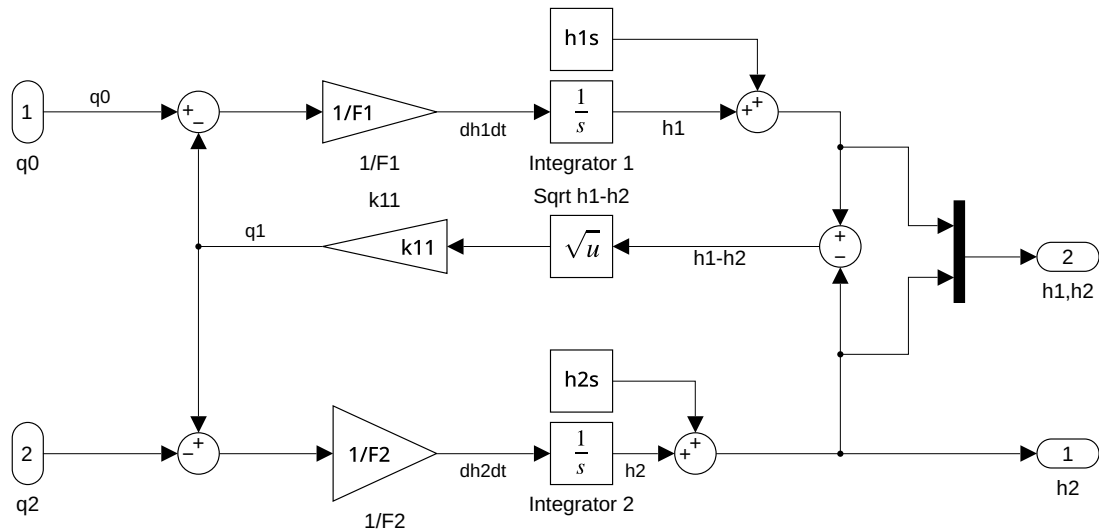


Figure 1.12: Simulink model of the nonlinear model of two tanks with pump.

Clearly, the model equations are nonlinear in h_1 and h_2 and linear in q_0 and q_2 .

We define deviational variables for the inputs, states, and outputs:

$$x_1(t) = h_1(t) - h_1(0), \quad u_1(t) = q_0(t) - q_0(0), \quad (1.49)$$

$$x_2(t) = h_2(t) - h_2(0), \quad u_2(t) = q_2(t) - q_2(0), \quad (1.50)$$

$$y_1 = x_2. \quad (1.51)$$

The resulting linear equations are ($k_1 = \frac{k_{11}}{2\sqrt{h_{10}-h_{20}}}$):

$$\frac{dx_1}{dt} = \frac{u_1}{F_1} - \frac{k_1}{F_1}x_1 + \frac{k_1}{F_1}x_2, \quad x_1(0) = h_1(0) - h_{10} = 0, \quad (1.52)$$

$$\frac{dx_2}{dt} = -\frac{u_2}{F_2} + \frac{k_1}{F_2}x_1 - \frac{k_1}{F_2}x_2, \quad x_2(0) = h_2(0) - h_{20} = 0. \quad (1.53)$$

1.2.3.5 Define a linear state-space system description, that is the matrices A , B , C , D as in Equations (1.6) and (1.7)

State-space description:

$$A = \begin{pmatrix} -\frac{k_1}{F_1} & \frac{k_1}{F_1} \\ \frac{k_1}{F_2} & -\frac{k_1}{F_2} \end{pmatrix}, \quad B = \begin{pmatrix} \frac{1}{F_1} & 0 \\ 0 & -\frac{1}{F_2} \end{pmatrix}, \quad C = (0 \ 1), \quad D = (0 \ 0). \quad (1.54)$$

1.2.3.6 Build a Simulink model from the basic building blocks that will simulate the evolution of the heights of the liquid for some given values of the initial conditions and inputs.

Simulation model is shown in Figure 1.12.

1.2.3.7 Run the simulation with the given initial conditions. Change $q_0 = 0.9q_0(0)$ at $t = 5$ s and $q_2 = 0.9q_2(0)$ at $t = 25$ s. How do you expect the tank levels to behave? Plot the simulation results as state and input variables vs. time.

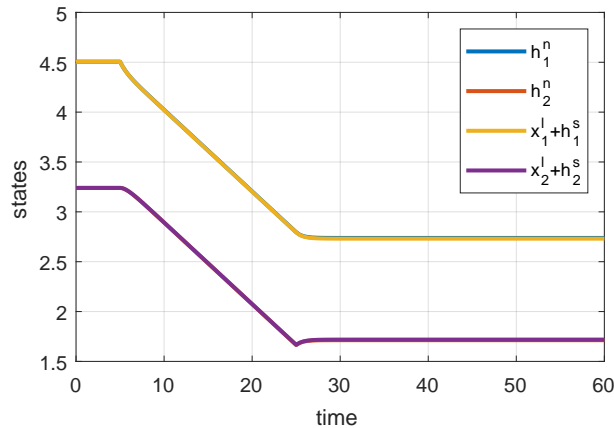


Figure 1.13: Comparison of simulated levels of nonlinear and linearised models for two tanks with pump.

Simulation results are shown in Figure 1.12. In this case, the difference between the nonlinear and linear models is practically negligible and the major impact on the dynamic properties is the integrating behaviour of the process.

1.2.4 Two Tanks in Series

We study a process consisting of two tanks connected in series. Each tank is structurally equivalent to the tank in Example 1.2.1. Liquid flows into the first tank with a flow rate of q_0 and exits from the tank by gravity with a flow rate of q_1 that then enters the second tank. We assume that we only can measure the level in the second tank.

1.2.4.1 Define the system by using serial connection of the two systems from Example 1.2.1. Connect the corresponding inputs, states, and outputs.

The first tank:

- input variable: liquid inlet flow q_0 ,
- state variable: height of the liquid in the tank $h_1(t)$.

The second tank:

- input variable: liquid inlet flow q_1 ,
- state variable: height of the liquid in the tank $h_2(t)$.

Two tanks (notice that the output variable of the first tank defines the input variable of the second tank $q_1 = k_{11}\sqrt{h_1}$. Thus both variables do not appear in the two-tank system description):

- input variable: liquid inlet flow q_0 ,
- state variables: heights of the liquid in the particular tanks $h_1(t)$ and $h_2(t)$,
- output variable: height of the liquid in the second tank $h_2(t)$.

1.2.4.2 Build the mathematical model from the individual models.

$$F_1 \frac{dh_1}{dt} = q_0 - k_{11} \sqrt{h_1}, \quad h_1(0) = h_1^s, \quad (1.55)$$

$$F_2 \frac{dh_2}{dt} = k_{11} \sqrt{h_1} - k_{22} \sqrt{h_2}, \quad h_2(0) = h_2^s, \quad (1.56)$$

with $h_1^s = (q_0^s/k_{11})^2$ and $h_2^s = (q_0^s/k_{22})^2$.

1.2.4.3 Define the linearised model by using the linear models from Example 1.2.1.

For both subsystems, define the deviational variables for the states and inputs:

$$x_1(t) = h_1(t) - h_1^s, \quad u_1(t) = q_0(t) - q_0^s, \quad (1.57)$$

$$x_2(t) = h_2(t) - h_2^s, \quad u_2(t) = q_1(t) - q_1^s, \quad (1.58)$$

and the linearised models

$$\frac{dx_1(t)}{dt} = a_1 x_1(t) + b_1 u_1(t), \quad a_1 = -\frac{k_{11}}{2F_1 \sqrt{h_1^s}}, \quad b_1 = \frac{1}{F_1}, \quad (1.59)$$

$$\frac{dx_2(t)}{dt} = a_2 x_2(t) + b_2 u_2(t), \quad a_2 = -\frac{k_{22}}{2F_2 \sqrt{h_2^s}}, \quad b_2 = \frac{1}{F_2}. \quad (1.60)$$

To connect both subsystems, note that u_2 is related to x_1 using the Taylor expansion of $q_1(t)$:

$$u_2 = q_1(t) - q_1^s = k_{11} \sqrt{h_1} - k_{11} \sqrt{h_1^s} = a_1 x_1. \quad (1.61)$$

Hence, the overall linearised model is

$$\frac{dx_1(t)}{dt} = a_1 x_1(t) + b_1 u_1(t), \quad (1.62)$$

$$\frac{dx_2(t)}{dt} = b_2 a_1 x_1(t) + a_2 x_2(t), \quad (1.63)$$

$$y(t) = x_2(t), \quad (1.64)$$

$$\mathbf{A} = \begin{pmatrix} a_1 & 0 \\ b_2 a_1 & a_2 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_1 \\ 0 \end{pmatrix} \quad (1.65)$$

$$\mathbf{C} = \begin{pmatrix} 0 & 1 \end{pmatrix}, \quad \mathbf{D} = 0. \quad (1.66)$$

1.2.4.4 Generalise the state-space description for n tanks in series.

If we assume that only the last tank contains a level sensor, the state-space model

contains n differential equations and is given as

$$\mathbf{A} = \begin{pmatrix} a_1 & 0 & 0 & 0 & \dots & 0 \\ b_2 a_1 & a_2 & 0 & 0 & \dots & 0 \\ 0 & b_3 a_2 & a_3 & 0 & \dots & 0 \\ \vdots & & & & & \vdots \\ 0 & \dots & 0 & 0 & b_n a_{n-1} & a_n \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad (1.67)$$

$$\mathbf{C} = (0 \quad \dots \quad 0 \quad 1), \quad \mathbf{D} = 0, \quad (1.68)$$

where

$$a_i = -\frac{k_{ii}}{2F_i \sqrt{h_i^s}}, \quad b_i = \frac{1}{F_i}. \quad (1.69)$$

1.3 Practice Examples

1.3.1 A One-Dimensional Car

We study a motion of a car. For simplicity, the car can only move in one dimension (Figure 1.14). Assume that the car is a point of mass $m = 1$ kg. We can directly influence its acceleration and can determine its properties: position and velocity.

1.3.1.1 Select the input variables and the state variables of the system.

Input variables: acceleration $u(t)$, state variables: position $x_1(t)$ and velocity $x_2(t)$.

1.3.1.2 Write down the equation of movement for the car (use Newton's second law of motion) and transform it to a set of first-order, ordinary differential equations.

$$\frac{d^2 x(t)}{dt^2} = m u(t) = u(t), \quad (1.70)$$

where $x(t)$ is the position of the car. First define:

$$x_1(t) = x(t), \quad (1.71)$$

$$x_2(t) = \frac{dx(t)}{dt}, \quad (1.72)$$

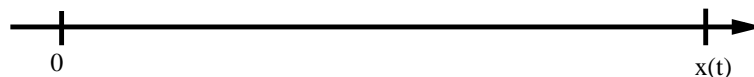


Figure 1.14: Motion of a one-dimensional car.

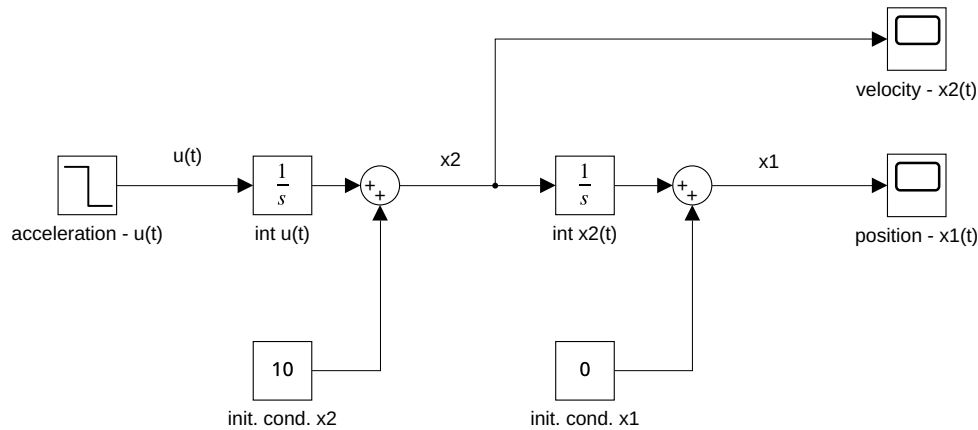


Figure 1.15: A Simulink model of the movement of the car.

which gives

$$\frac{dx_1(t)}{dt} = \frac{dx(t)}{dt} = x_2(t), \quad (1.73)$$

$$\frac{dx_2(t)}{dt} = \frac{d^2x(t)}{dt} = u(t). \quad (1.74)$$

1.3.1.3 Discuss the steady-state behaviour of the system.

The steady state only exists when $u(t) = u^s = 0$ and $x_2^s = x_2(t) = 0$. The value x_1^s of $x_1(t)$ is arbitrary at the steady state.

1.3.1.4 The obtained model is a double integrator. Linearise the model if necessary.

The model can be written as:

$$\frac{d}{dt} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \mathbf{A} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \mathbf{B} u(t) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u(t), \quad (1.75)$$

so it is linear.

1.3.1.5 Build a Simulink model that will simulate the movement of the car for some given values of the initial conditions and inputs.

The Simulink model is shown in Figure 1.15.

1.3.1.6 Run the simulation with initial conditions of 0 m for the position and 10 m/s for the velocity. Use the input value of 5 m/s² for the first 10 s and 0 m/s² for the next 10 s. Plot the simulation results as state variables vs. time.

(Optional) Add an input value of -5 m/s^2 between $t \in (20, 30)$.

(Optional) Check the correctness of the simulation results by analytically solving the model equations.

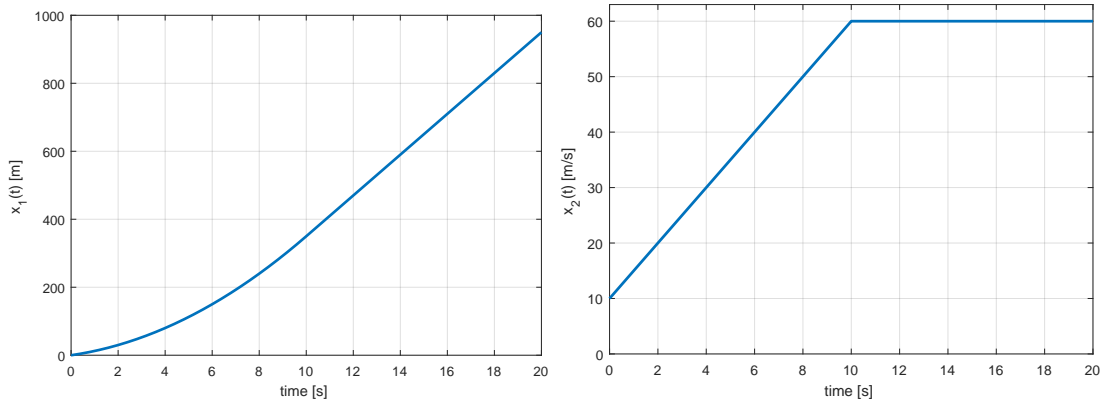


Figure 1.16: (left) Position and (right) velocity time trajectories of the car.

Run the simulation (Figure 1.15) and plot the results shown in Figure 1.16.

```
sim('double_integrator')
plot(tx1(:,1), tx1(:,2), 'LineWidth', 2);
xlabel('time [s]'); ylabel('x_1(t)'); grid on
% iam_save_pdf('x1_t.pdf');

plot(tx2(:,1), tx2(:,2), 'LineWidth', 2);
xlabel('time [s]'); ylabel('x_2(t)'); axis([0 20 0 63]); grid on
% iam_save_pdf('x2_t.pdf');
```

For the analytical solution of the model equations, we will make use of the fact that the input is piecewise constant. For the velocity, we can write:

$$\frac{dx_2(t)}{dt} = u \rightarrow \int_{x_{2,0}}^{x_2(t)} dx_2(t) = \int_0^t u dt \rightarrow x_2(t) = x_{2,0} + ut, \quad (1.76)$$

which means that if $u \neq 0$ the velocity is a linear function of time. The plot of the simulation results for $x_2(t)$ should therefore show a line with slope u . If $u = 0$, the velocity does not change with time.

For the position, we arrive at:

$$\frac{dx_1(t)}{dt} = x_2(t) \rightarrow \int_{x_{1,0}}^{x_1(t)} dx_1(t) = \int_0^t x_{2,0} + ut dt \rightarrow x_1(t) = x_{1,0} + x_{2,0}t + \frac{1}{2}ut^2, \quad (1.77)$$

which means that if $u \neq 0$ the position is a quadratic function of time. The plot of the simulation results for $x_2(t)$ should therefore show part of a parabola. If $u = 0$, the position is a linear function of time.

- 1.3.1.7 Create a Simulink subsystem in order to separate the structure of the dynamic system and the data that might change in different simulations.

The simulation model is shown in Figure 1.17.

(Optional) Simplify the Simulink model representation for maximal re-use (imagine using a triple or quadruple integrator).

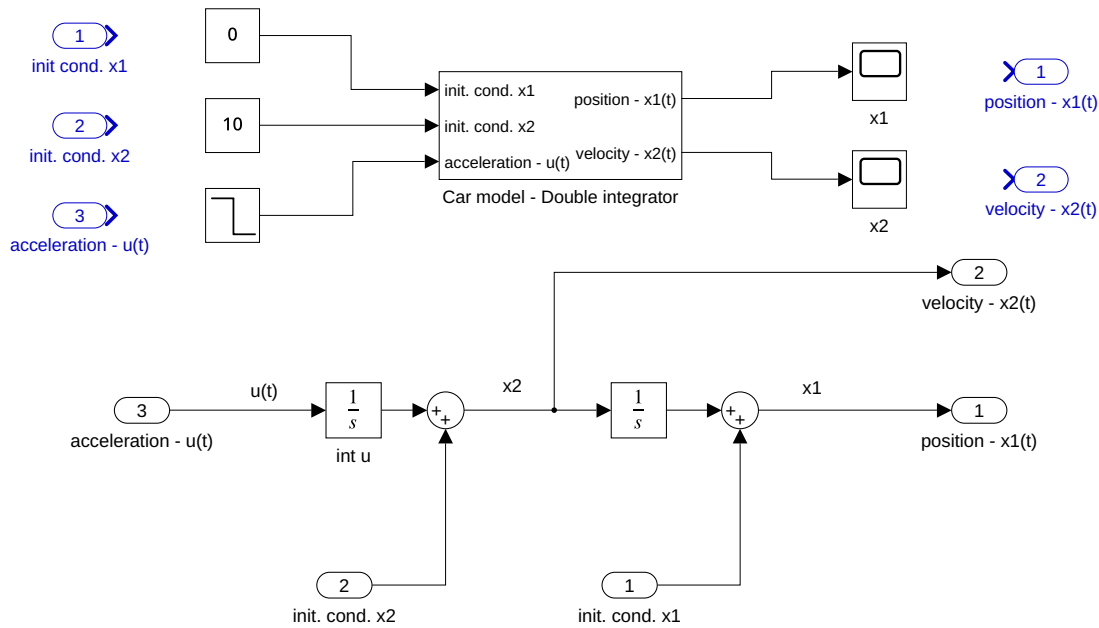


Figure 1.17: (top) A Simulink model with a subsystem and (bottom) the detail of the subsystem.

The simulation model with two simple integrator blocks is shown in Figure 1.18.

1.3.1.8 (Optional) Simulate the system with initial conditions $x_1(0) = 0$ m for the position and $x_2(0) = 10$ m/s for the velocity and with a controller whose control law is given by $u(t) = -K_1x_1(t) - K_2x_2(t)$ where $K_1 = 1, K_2 = 1.73$. Observe the results by plotting velocity vs. position.

We will re-use the model in Figure 1.17 and close the loop as in Figure 1.19.

1.4 Problems

Problem. 1.1: Linearisation

Consider a nonlinear function of the form $F(x) = x^3 + 3x^2 - x + 1$. Find a linear function $F_0(x)$ that approximates $F(x_0)$ at $x_0 = 2$.

$F_0(x) = 23x - 27$

Problem. 1.2: Linearisation

Consider a nonlinear model of a dynamic system of the form $\dot{y}(t) + 0.5y^2(t) + y(t) = v(t)$. Its steady state is characterised by the values v^s, y^s . Determine the corresponding linearised model in deviational variables defined by $x(t) = y(t) - y^s, u(t) = v(t) - v^s$.

$\dot{x}(t) + x(t)(y^s s + 1) = u(t)$

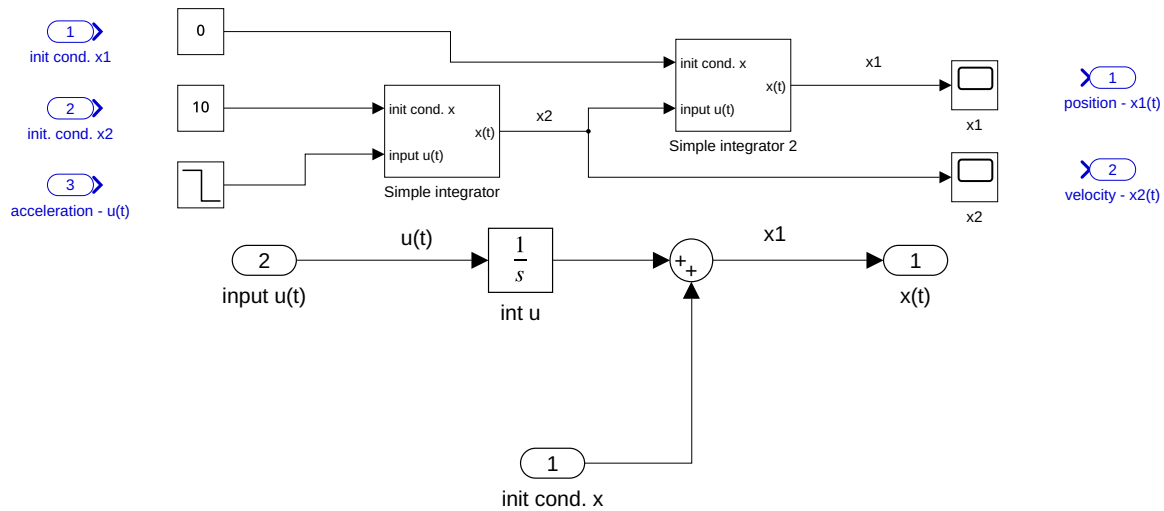


Figure 1.18: A Simulink model of two simple integrators.

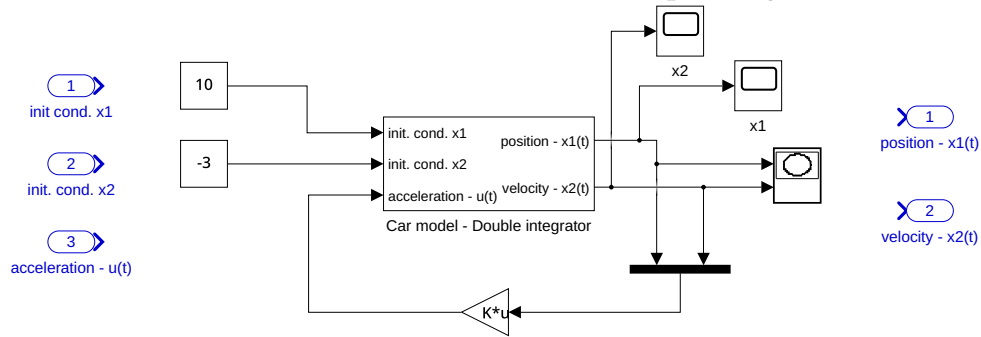


Figure 1.19: A Simulink model of the state-feedback control of car.

Problem. 1.3: Mass on a Spring

Consider a system consisting of mass $M = 1$ kg hanging on a spring (see Figure 1.20). The spring force is given by Hooke's Law, $F_s = k_s(x(t) - x_0)$, where k_s is the spring constant, $x(t)$ is the position of the mass (distance from the ceiling), and x_0 is the equilibrium position of the end of the spring with no mass attached. We assume the friction force is proportional to the velocity of the mass $F_r = k_r \dot{x}(t)$, where k_r is the friction coefficient. The position of the mass is measured, hence $y(t) = x(t)$.

In general, the mass is affected by the gravitational force F_g , which together with other forces creates a momentum displacement force F_m .

1. Create the mathematical model of the process that describes the position of the mass.
2. Define the deviational variables and transform the model to use them.

$$\ddot{x}(t) + k_r \dot{x}(t) + k_s x(t) = k_s x_0 + g, \tag{1.78}$$

$$y(t) = x(t) - x^s, \quad \dot{y}(t) = \dot{x}(t), \quad \ddot{y}(t) = \ddot{x}(t), \tag{1.79}$$

$$\ddot{y}(t) + k_r \dot{y}(t) + k_s y(t) = 0, \quad \dot{y}(0) = y(0) = 0. \tag{1.80}$$

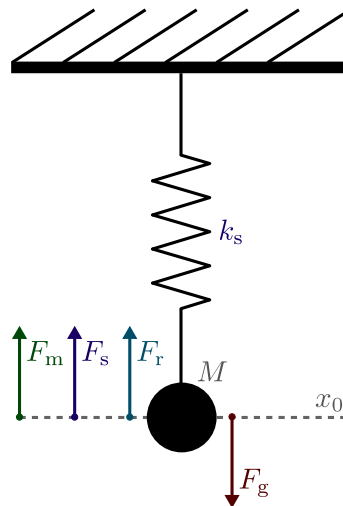


Figure 1.20: Mass-on-a-spring system.

Problem. 1.4: Mixing of Liquids

Consider a tank with m [kg] of salt water where m is constant and the mass fraction of salt in the water $w(t)$ can change with time. There are two inlet streams: $m_1(t)$ [kg/s] with salt mass fraction $w_1(t)$ and $m_2(t)$ [kg/s] with salt mass fraction $w_2(t)$. The water outlet contains the salt with the same concentration as in the tank.

1. Define a system and select its input, state, and output variables.
2. Write down the (dynamic) mass balances for the total mass and mass of the salt in the tank.
3. Calculate the steady-state salt concentration (mass) in the tank.
4. Linearise the model.
5. Define a linear state-space system if the salt concentration in the tank is considered to be measured.

$$\frac{dw}{dt} = \frac{m_1}{m}(w_1 - w) + \frac{m_2}{m}(w_2 - w), \quad w^s = \frac{m_1 w_1 + m_2 w_2}{m_1 + m_2}, \quad (1.81)$$

$$u_1 = m_1 - m_1^s, \quad u_2 = m_2 - m_2^s, \quad u_3 = w_1 - w_1^s, \quad u_4 = w_2 - w_2^s, \quad (1.82)$$

$$x = w - w^s, \quad (1.83)$$

$$m \frac{dx}{dt} = u_1 (w_1^s - w^s) + u_2 (w_2^s - w^s) + u_3 m_1^s + u_4 m_2^s - x (m_1^s + m_2^s), \quad (1.84)$$

$$\mathbf{A} = -\frac{m_1^s + m_2^s}{m}, \quad \mathbf{B} = \frac{1}{m} \begin{pmatrix} w_1^s - w^s \\ w_2^s - w^s \\ m_1^s \\ m_2^s \end{pmatrix}, \quad (1.85)$$

$$\mathbf{C} = 1, \quad \mathbf{D} = 0. \quad (1.86)$$

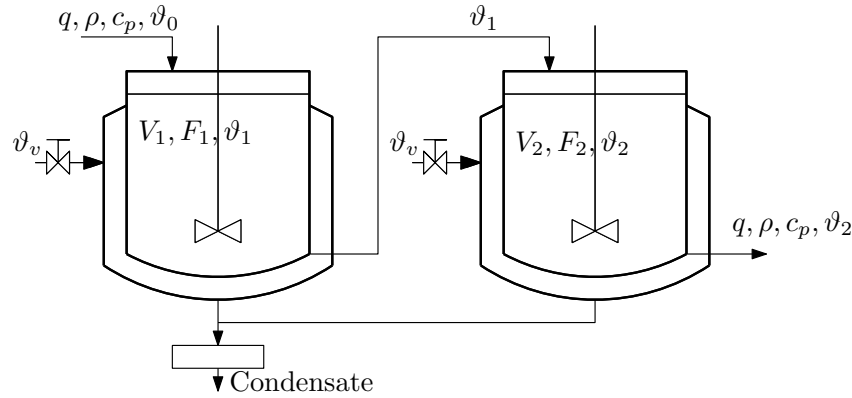


Figure 1.21: Two heat exchangers in series.

Problem. 1.5: Two Heat Exchangers in Series

Consider a series of heat exchangers shown in Figure 1.21. The inlet stream to the first heat exchanger enters with a temperature ϑ_0 [°C]. The heat exchangers are heated with the condensing vapour of temperature ϑ_v [°C]. The temperatures inside of units are ϑ_1, ϑ_2 [°C]. Constant process parameters include the volumetric flow rate of the inlet and outlet streams q [m³/s], density ρ [kg/m³], volumes V_1, V_2 [m³], liquid specific heat capacity c_p [J kg⁻¹ K⁻¹], the heat transfer areas F_1, F_2 [m²], and heat transfer coefficient α [W m⁻² K⁻¹].

Further assumptions for development of the model are negligible wall accumulation, zero heat losses, and well-mixed tanks.

1. Define a system and select its input, state, and output variables.
2. Write the dynamic energy balances.
3. Calculate the steady-state temperatures in the heat exchangers.
4. Define a linear state-space system if the temperature ϑ_2 is measured.

$$T_1 \frac{d\vartheta_1}{dt} = -\vartheta_1 + K_{11}\vartheta_v + K_{12}\vartheta_0, \quad (1.87)$$

$$T_2 \frac{d\vartheta_2}{dt} = -\vartheta_2 + K_{21}\vartheta_v + K_{22}\vartheta_1, \quad (1.88)$$

$$T_1 = \frac{V_1 \rho c_p}{q \rho c_p + \alpha F_1}, \quad K_{11} = \frac{\alpha F_1}{q \rho c_p + \alpha F_1}, \quad K_{12} = \frac{q \rho c_p}{q \rho c_p + \alpha F_1}, \quad (1.89)$$

$$T_2 = \frac{V_2 \rho c_p}{q \rho c_p + \alpha F_2}, \quad K_{21} = \frac{\alpha F_2}{q \rho c_p + \alpha F_2}, \quad K_{22} = \frac{q \rho c_p}{q \rho c_p + \alpha F_2}, \quad (1.90)$$

$$\vartheta_1^s = K_{11}\vartheta_v^s + K_{12}\vartheta_0^s, \quad \vartheta_2^s = K_{21}\vartheta_v^s + K_{22}\vartheta_1^s, \quad (1.91)$$

$$u_1 = \vartheta_v - \vartheta_v^s, \quad u_2 = \vartheta_0 - \vartheta_0^s, \quad x_1 = \vartheta_1 - \vartheta_1^s, \quad x_2 = \vartheta_2 - \vartheta_2^s, \quad (1.92)$$

$$y = x_2, \quad (1.93)$$

$$\mathbf{A} = \begin{pmatrix} -\frac{1}{T_1} & 0 \\ \frac{K_{22}}{T_2} & -\frac{1}{T_2} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \frac{K_{11}}{T_1} & \frac{K_{12}}{T_1} \\ \frac{K_{21}}{T_2} & 0 \end{pmatrix}, \quad \mathbf{C} = (0 \ 1), \quad \mathbf{D} = 0. \quad (1.94)$$

1.5 MATLAB Snippets

- Linearisation of the function $F(x) = x^3 + 3x^2 - x + 1$ about the operating point $x_0 = 2$ using symbolic operations

```
syms x x0
F = x^3 + 3 * x^2 - x + 1;
F0 = subs(F, x, x0) + (x-x0) * subs(diff(F, x), x, x0)
fplot(F, [-5, 5])
hold on
fplot(subs(F0, x0, 2), [0 5])
legend('F(x)', 'Flin(x)')
```

- Definition of a state-space model

```
A = [-1, -2; 1, -1]; B = [0.5; 0];
C = [0, 1]; D = 0;
sys = ss(A,B,C,D)
```

- Simulation of a state-space model response with nonzero initial conditions and zero input

```
x0 = [1; 2];
initial(sys, x0)
```

- Simulation of a state-space model response with nonzero initial conditions and nonzero input

```
t = 0:0.01:10;
u = ones(1, length(t));
% u = sin(t);
lsim(sys, u, t, x0)
```

- Solution of a set of differential equations

```
x0 = [1; 2];
t = 0:0.01:10;
f=@(t,x) [-x(1) - 2 * x(2) + 0.5; x(1) - x(2)]
[tn, xn]=ode45(f, t, x0);
plot(tn, xn(:, 2))
```

- Linearisation of a Simulink model MDL at the operating point IO

```
linsys = linearize mdl, io)
```

- Find the operating point in the Simulink model.

```
mdl = 'tanks'; open_system(mdl)
opspec = operspec(mdl)
op = findop(mdl, opspec)
```

- Save figures to a pdf file and crop the result. This function is used in the book to produce MATLAB figure files. It should be added to the path so that it is accessible in other chapters as well.

```
function [] = iam_save_pdf(filename, width, height)
% Save FILENAME (with extension pdf) of the current figure.
% Default values for figure dimensions are WIDTH=16cm, HEIGHT=11cm
% The final figure is cropped using external file pdfcrop (should exist)

if nargin == 1
    width = 16; height = 11;
end
fig = gcf;
FigurePaperUnits = 'centimeters'; FigurePaperPositionMode = 'manual';
FigurePaperPosition = [0 0 width height]; FigurePaperSize = [width height];
print(filename, '-painters', '-dpdf');
system(['pdfcrop ', filename, ' ', filename]);
end
```

- Save Simulink model to a pdf file and crop the result.

```
function [] = iam_save_slx_pdf(filename, slxfile, slxmodel)
% Save FILENAME (with extension pdf) of the model SLXMODEL in SLXFILE.
% The final figure is cropped using external file pdfcrop (should exist)
% If only one input parameter is specified, it is SLXMODEL.
if nargin == 1
    slxmodel = filename;
    slxfile = [filename, '.slx'];
    filename = [filename, '.pdf'];
end
open_system(slxfile)
print(['-s', slxmodel], filename, '-dpdf', '-painters');
system(['pdfcrop ', filename, ' ', filename]);
close_system(slxfile)
end
```

- Run MATLAB script before opening the Simulink model (initialise all model constants).
Simulink uses a mechanism of model callback functions at certain points (model opening, saving, closing, etc.). In the Simulink Editor, open the Property Inspector. In the Modelling tab, under Design, click Property Inspector. Here, choose a suitable callback and specify the MATLAB code or script that should be used, for example LoadFcn (only once) or InitFcn (at every simulation run).

1.6 Python Snippets

We also include one example for the Python programming language that models the process with one tank that reproduces the results shown in Figure 1.5. The python library `python-control` is used for automatic control, `numpy` for data manipulation, and `matplotlib` for graphs.

```
import control
import numpy as np
import math
import matplotlib.pyplot as plt

def iam_save_pdf(plt, filename, width=16, height=11):
    """Write actual plot to filename with default width = 16 and height = 11"""
    plt.rc('text', usetex=True)
    plt.rc('font', family='serif')
    plt.gcf().set_size_inches(width/2.54, height/2.54)
    plt.tight_layout()
    plt.savefig(filename)

def tank_derivative(t, x, u, params): # right hand side of differential equation
    F = params.get('F')
    k = params.get('k')

    h = x[0]
    q0 = u
    if h < 0: # numerical issues
        h = 0
    dh = (q0 - k * math.sqrt(h)) / F
    return dh

def tank_output(t, x, u, params): # output equation
    return x[0]
```

```

tank = control.NonlinearIOSystem(
    tank_derivative, tank_output, input='q0', output=('h'),
    states='h', params = {'F':0.5, 'k':1.15}, name='tank')

T = np.linspace(0, 4, 100)
q0 = np.array([
    0.3 if t <= 2 else
    0.2 for t in T])
q0s = 0.3
h0 = 0
hs, qs = control.find_eqpt(tank, h0, q0s)
t, h = control.input_output_response(tank, T, q0, h0, solve_ivp_method='BDF')
ss_sys = control.linearize(tank, hs, q0s)
tank_lin = control.LinearIOSystem(ss_sys)
t, hl = control.input_output_response(tank_lin, T, q0 - q0s, -hs)

plt.close('all')
plt.figure(1)
plt.step(T, q0)
plt.xlabel('time [s]')
plt.ylabel('flow $q_0(t)$ [m$^3$/s]')
plt.grid(True)
plt.title('Input flowrate $q_0$')
# iam_save_pdf(plt, "h_t2_q0.pdf")

plt.figure(2)
plt.plot(T, h, label='h-nonlinear')
plt.plot(T, hl+hs, label='h-linear')
plt.legend()
plt.xlabel('time [s]')
plt.ylabel('level $h(t)$ [m]')
plt.title('Tank level $h$')
plt.grid(True)
plt.show(block=False)
# iam_save_pdf(plt, "h_t2.pdf")

```

1.7 Bibliography

The theoretical background for modelling, simulation, linearisation can be found in Mikleš and Fikar (2007, Chapter 2), information in Slovak in Bakošová and Fikar (2008, Chapter 4) or Huba, Hubinský, and Žáková (2006, Chapter 1).

For additional information, see for example Ogata (2010, Chapters 2-4), Corriou (2004, Chapter 1), Bequette (2003, Chapter 2) or Dorf and Bishop (2008, Chapter 2). Process oriented point of view is emphasised in Shinskey (1979), Luyben (1990), Seborg, Edgar, Mellichamp, and Doyle (2016), Stephanopoulos (1984), and Marlin (1995).

Input-Output Characteristics of Dynamic Systems

In this chapter, you will review your knowledge and learn new terms with the following outcomes:

- definition of transfer functions and transfer matrices,
- conversions between differential equations and transfer functions,
- conversions between state-space and transfer-function representations,
- the algebra of transfer functions,
- the important state-space representations.

We will review the formal equivalence of the input-output and state-space representations.

2.1 Overview

2.1.1 Transfer Functions

Linear continuous single-input, single-output (SISO) systems with constant coefficients with input $u(t)$ and output $y(t)$ can be described by a differential equation of the form

$$\frac{d^n y(t)}{dt^n} + a_{n-1} \frac{d^{n-1} y(t)}{dt^{n-1}} + \cdots + a_0 y(t) = b_m \frac{d^m u(t)}{dt^m} + \cdots + b_0 u(t), \quad (2.1)$$

where we suppose that $u(t)$ and $y(t)$ are deviation variables and without loss of generality we set $a_n = 1$. Physical feasibility (causality) requires that $n \geq m$. We distinguish proper $n = m$ and strictly proper ($n > m$) systems.

After performing the Laplace transform of (2.1) and assuming zero initial conditions we get

$$(s^n + a_{n-1}s^{n-1} + \cdots + a_0)Y(s) = (b_m s^m + \cdots + b_0)U(s), \quad (2.2)$$

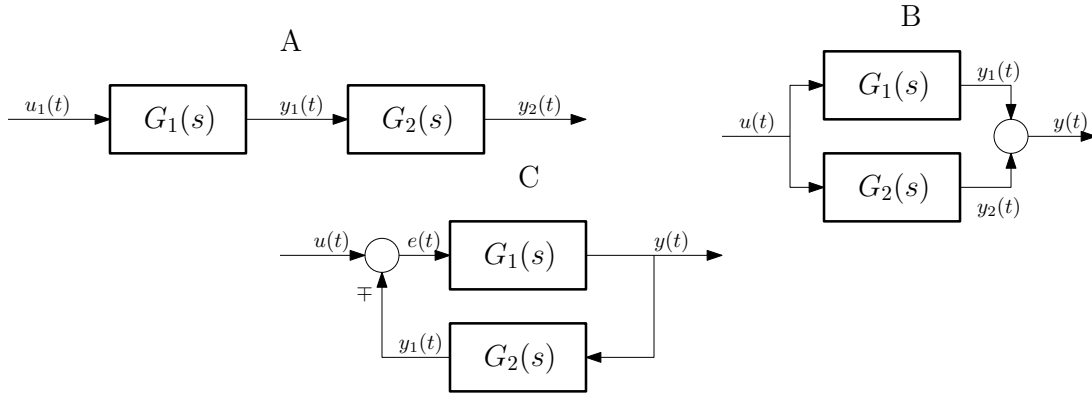


Figure 2.1: Algebra of transfer functions: serial (A), parallel (B), and feedback (C) connections.

or

$$G(s) = \frac{Y(s)}{U(s)} = \frac{B(s)}{A(s)}, \quad (2.3)$$

where

$$B(s) = b_m s^m + b_{m-1} s^{m-1} + \dots + b_0, \quad A(s) = s^n + a_{n-1} s^{n-1} + \dots + a_0. \quad (2.4)$$

$G(s)$ is called a transfer function of the system and is defined as the ratio of the Laplace transforms of the output and the input variables under zero initial conditions.

We can understand its behaviour when studying the relationship $Y(s) = G(s)U(s)$. This means that if the Laplace transform of the input is $U(s)$ then the Laplace transform of the output is described by $G(s)U(s)$.

The equation $Y(s) = G(s)U(s)$ in the time domain is a convolution integral and when assuming zero initial conditions, gives

$$y(t) = g(t) * u(t) = \int_0^T g(t - \tau)u(\tau)d\tau, \quad (2.5)$$

where $g(t)$ represents impulse response of the system (output of the system when the input is an impulse).

2.1.2 The Algebra of Transfer Functions

All block representations of dynamics systems given by partial transfer functions can be decomposed into three basic connections (Figure 2.1):

1. Serial connection: $Y_1(s) = G_1(s)U_1(s)$ and $Y_2(s) = G_2(s)Y_1(s)$. The resulting transfer function is given as

$$G(s) = \frac{Y_2(s)}{U_1(s)} = G_1(s)G_2(s). \quad (2.6)$$

2. Parallel connection: $Y_1(s) = G_1(s)U(s)$, $Y_2(s) = G_2(s)U(s)$, and $Y(s) = Y_1(s) + Y_2(s)$. The resulting transfer function is given as

$$G(s) = \frac{Y(s)}{U(s)} = G_1(s) + G_2(s). \quad (2.7)$$

3. Feedback connection: $Y(s) = G_1(s)E(s)$, $Y_1(s) = G_2(s)Y(s)$, and $E(s) = U(s) \mp Y_1(s)$. The resulting transfer function is given as

$$G(s) = \frac{Y(s)}{U(s)} = \frac{G_1(s)}{1 \pm G_1(s)G_2(s)}. \quad (2.8)$$

2.1.3 Converting State-Space Representations to Transfer-Functions Representations

The standard and natural description of a system with m inputs and r outputs (multiple-input, multiple-output – MIMO) is in the form of state-space model

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad (2.9)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t). \quad (2.10)$$

The Laplace transform gives

$$\mathbf{Y}(s) = \left(\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \right) \mathbf{U}(s) + \left(\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1} \right) \mathbf{x}_0. \quad (2.11)$$

Assuming zero initial conditions $\mathbf{x}_0 = \mathbf{0}$ gives

$$\mathbf{Y}(s) = \mathbf{G}(s)\mathbf{U}(s), \quad (2.12)$$

where

$$\mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}, \quad (2.13)$$

is a transfer matrix with dimensions $[r \times m]$. Its individual elements $G_{ij}(s)$, $i = 1, \dots, r$, $j = 1, \dots, m$ are transfer functions between the i -th output and j -th input, *i.e.* $G_{ij}(s) = Y_i(s)/U_j(s)$.

The matrix $\mathbf{G}(s)$ can also be written as

$$\mathbf{G}(s) = \frac{\mathbf{C}\text{adj}(s\mathbf{I} - \mathbf{A})\mathbf{B} + \mathbf{D}}{m(s)}, \quad (2.14)$$

where $m(s) = |s\mathbf{I} - \mathbf{A}|$ and the roots of the polynomial $m(s)$ are poles of $\mathbf{G}(s)$.

2.1.4 Converting Transfer-Function Representations to State-Space Representations

While the conversion of a state-space representation into a transfer function is unique, there are infinitely many possible state-space conversions and representations. We will study two representations in this section.

Controllable Canonical Form

Consider a system with a transfer function

$$G(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}, \quad n \geq m, \quad (2.15)$$

or in the time domain equivalently described by the following differential equation (we denote n differentiations of the variable $y(t)$ with respect to time by $y^{(n)}(t)$)

$$\begin{aligned} y^{(n)}(t) + a_{n-1}y^{(n-1)}(t) + \dots + a_1\dot{y}(t) + a_0y(t) &= \dots \\ \dots &= b_m u^{(m)}(t) + b_{m-1}u^{(m-1)}(t) + \dots + b_1\dot{u}(t) + b_0u(t). \end{aligned} \quad (2.16)$$

Let us introduce an auxiliary variable $z(t)$ and its Laplace image $Z(s)$ such that

$$\frac{Y(s)}{Z(s)} = b_m s^m + b_{m-1}s^{m-1} + \dots + b_1s + b_0, \quad (2.17)$$

$$\frac{Z(s)}{U(s)} = \frac{1}{s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0}. \quad (2.18)$$

Equation (2.18) corresponds to the following differential equation

$$z^{(n)}(t) + a_{n-1}z^{(n-1)}(t) + \dots + a_1\dot{z}(t) + a_0z(t) = u(t). \quad (2.19)$$

Now let us define state variables by the following relationships

$$x_1 = z, \quad x_2 = \dot{z}, \quad \dots \quad x_i = \frac{d^{i-1}z(t)}{dt^{i-1}}, \quad \dots \quad x_n = \frac{d^{n-1}z(t)}{dt^{n-1}}, \quad (2.20)$$

which implies

$$\frac{dx_i(t)}{dt} = x_{i+1}(t), \quad i = 1, 2, \dots, n-1. \quad (2.21)$$

Equation (2.19) can now be transformed into n first-order differential equations

$$\begin{aligned} \frac{dx_1(t)}{dt} &= x_2(t), \\ \frac{dx_2(t)}{dt} &= x_3(t), \\ &\vdots \\ \frac{dx_{n-1}(t)}{dt} &= x_n(t), \\ \frac{dx_n(t)}{dt} &= \frac{d^n z(t)}{dt^n} = -a_{n-1}x_n - \dots - a_1x_2(t) - a_0x_1(t) + u(t). \end{aligned} \quad (2.22)$$

If $n = m$ then Eq. (2.17) corresponds to

$$y(t) = b_0x_1(t) + b_1x_2(t) + \dots + b_{n-1}x_n(t) + b_n\dot{x}_n(t). \quad (2.23)$$

$\dot{x}_n(t)$ from the above equation can be obtained from Eq. (2.22) and gives

$$y(t) = (b_0 - a_0b_n)x_1(t) + (b_1 - a_1b_n)x_2(t) + \dots + (b_{n-1} - a_{n-1}b_n)x_n(t) + b_nu(t). \quad (2.24)$$

Equations (2.22) and (2.24) form a general state-space model (A , B , C , and D) given as

$$\mathbf{A}_c = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n-1} \end{pmatrix}, \quad \mathbf{B}_c = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}, \quad (2.25)$$

$$\mathbf{C}_c = (b_0 - a_0b_n \quad b_1 - a_1b_n \quad \dots \quad b_{n-1} - a_{n-1}b_n), \quad \mathbf{D}_c = b_n. \quad (2.26)$$

We can observe that if $m < n$ then $\mathbf{D}_c = \mathbf{0}$. This state-space representation is called the controllable canonical form.

Observable Canonical Form

Consider again the system with transfer function (2.15) and its representation as a differential equation (assuming $n = m$)

$$\begin{aligned} y^{(n)}(t) + a_{n-1}y^{(n-1)}(t) + \cdots + a_1\dot{y}(t) + a_0y(t) &= \cdots \\ \cdots &= b_nu^{(n)}(t) + b_{n-1}u^{(n-1)}(t) + \cdots + b_1\dot{u}(t) + b_0u(t). \end{aligned} \quad (2.27)$$

The observable canonical form is

$$\mathbf{A}_o = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & 0 & \cdots & 0 & -a_2 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -a_{n-1} \end{pmatrix}, \quad \mathbf{B}_o = \begin{pmatrix} b_0 - a_0b_n \\ b_1 - a_1b_n \\ \vdots \\ b_{n-2} - a_{n-2}b_n \\ b_{n-1} - a_{n-1}b_n \end{pmatrix}, \quad (2.28)$$

$$\mathbf{C}_o = (0 \ \cdots \ 0 \ 1), \quad \mathbf{D}_o = b_n. \quad (2.29)$$

Note that the observable canonical form can be obtained from the controllable canonical form as

$$\mathbf{A}_o = \mathbf{A}_c^T, \quad (2.30)$$

$$\mathbf{B}_o = \mathbf{C}_c^T, \quad (2.31)$$

$$\mathbf{C}_o = \mathbf{B}_c^T, \quad (2.32)$$

$$\mathbf{D}_o = \mathbf{D}_c. \quad (2.33)$$

For the derivation of this canonical form and its physical explanation see also Kailath 1980.

2.2 Examples

2.2.1 Conversion of Second-Order Transfer Functions to the Controllable Canonical Form

Consider a second-order dynamic system described by the following differential equation

$$\ddot{y}(t) + a_1\dot{y}(t) + a_0y(t) = b_2\ddot{u}(t) + b_1\dot{u}(t) + b_0u(t). \quad (2.34)$$

Apply the Laplace transform under zero initial conditions

$$s^2Y(s) + a_1sY(s) + a_0Y(s) = b_2s^2U(s) + b_1sU(s) + b_0U(s). \quad (2.35)$$

The resulting transfer function is then given as

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_2s^2 + b_1s + b_0}{s^2 + a_1s + a_0}. \quad (2.36)$$

2.2.1.1 Derive the controllable canonical form if $b_1 = b_2 = 0$.

If the numerator of the transfer function is a constant, we study the process

described by the transfer function

$$\frac{Y(s)}{U(s)} = \frac{b_0}{s^2 + a_1s + a_0}, \quad (2.37)$$

and by the differential equation

$$\ddot{y}(t) + a_1\dot{y}(t) + a_0y(t) = b_0u(t). \quad (2.38)$$

The controllable canonical form defines the first state as the output variable $y(t)$ and the other states as derivatives of the state with respect to time with a gradually increasing order. As the differential equation (or the denominator of the transfer function) is of order 2, there will be two states $x_1 = y$ and $x_2 = \dot{y}$. If we differentiate both states with respect to time we get $\dot{x}_1 = \dot{y}$ and $\dot{x}_2 = \ddot{y}$. Next, we substitute the defined states for $y(t)$ and its derivatives in the respective expressions

$$\dot{x}_1 = x_2, \quad (2.39)$$

$$\dot{x}_2 = b_0u - a_0x_1 - a_1x_2, \quad (2.40)$$

where the second equation results from the differential equation of the system (2.38). The state-space matrices A_c and B_c are then

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -a_0 & -a_1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ b_0 \end{pmatrix} u, \quad (2.41)$$

$$A_c = \begin{pmatrix} 0 & 1 \\ -a_0 & -a_1 \end{pmatrix}, \quad B_c = \begin{pmatrix} 0 \\ b_0 \end{pmatrix}. \quad (2.42)$$

Definition of matrices C_c and D_c follows from the fact that the first state is equal to the output, hence

$$y = x_1 = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad (2.43)$$

and

$$C_c = \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad D_c = 0. \quad (2.44)$$

2.2.1.2 Derive the controllable canonical form if $b_2 = 0$.

In this case, the system transfer function is strictly proper

$$\frac{Y(s)}{U(s)} = \frac{b_1s + b_0}{s^2 + a_1s + a_0}, \quad (2.45)$$

and described by the differential equation

$$\ddot{y}(t) + a_1\dot{y}(t) + a_0y(t) = b_1\dot{u}(t) + b_0u(t). \quad (2.46)$$

We will find the solution in two steps. Firstly, we will reduce the problem to the previous one (constant numerator of the transfer function).

To this end, we define an intermediate variable $z(t)$ and an intermediate transfer function

$$\frac{Z(s)}{U(s)} = \frac{1}{s^2 + a_1s + a_0}. \quad (2.47)$$

The relationship between $Z(s)$ and $Y(s)$ can be obtained if we divide the transfer function (2.45) by (2.47)

$$\frac{Y(s)}{U(s)} = \frac{b_1s + b_0}{s^2 + a_1s + a_0} \cdot \frac{Z(s)}{U(s)} = \frac{b_1s + b_0}{s^2 + a_1s + a_0}, \quad (2.48)$$

$$\frac{Y(s)}{Z(s)} = b_1s + b_0, \quad (2.49)$$

$$Y(s) = B(s)Z(s), \quad (2.50)$$

$$y(t) = b_1\dot{z}(t) + b_0z(t). \quad (2.51)$$

We define states as the respective time derivatives of the intermediate variable $x_1 = z$ and $x_2 = \dot{z}$ which gives rise to the same state equations and matrices A_c and B_c as before

$$\dot{x}_1 = x_2, \quad (2.52)$$

$$\dot{x}_2 = u - a_0x_1 - a_1x_2. \quad (2.53)$$

The output equation is obtained by substituting states for the derivatives of z

$$y = b_0z + b_1\dot{z} = b_0x_1 + b_1x_2. \quad (2.54)$$

The matrices of the state-space representation in a controllable canonical form are given as

$$A_c = \begin{pmatrix} 0 & 1 \\ -a_0 & -a_1 \end{pmatrix}, \quad B_c = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad (2.55)$$

$$C_c = (b_0 \quad b_1), \quad D_c = 0. \quad (2.56)$$

2.2.1.3 Derive the controllable canonical form if $b_2 \neq 0$.

This case involves a proper transfer function where the numerator and denominator have the same degree

$$\frac{Y(s)}{U(s)} = \frac{b_2s^2 + b_1s + b_0}{s^2 + a_1s + a_0}, \quad (2.57)$$

and the process is described by the differential equation

$$\ddot{y}(t) + a_1\dot{y}(t) + a_0y(t) = b_2\ddot{u}(t) + b_1\dot{u}(t) + b_0u(t). \quad (2.58)$$

Again, we define the intermediate variable $Z(s)/U(s) = 1/A(s)$, the corresponding differential equation:

$$\ddot{z}(t) + a_1\dot{z}(t) + a_0z(t) = u(t), \quad (2.59)$$

and the states $x_1 = z$ and $x_2 = \dot{z}$. State and output equations are then

$$\dot{x}_1 = x_2, \quad (2.60)$$

$$\dot{x}_2 = u - a_0x_1 - a_1x_2, \quad (2.61)$$

$$y = b_0z + b_1\dot{z} + b_2\ddot{z} = b_0x_1 + b_1x_2 + b_2(u - a_0x_1 - a_1x_2). \quad (2.62)$$

The matrices of the state-space representation in a controllable canonical form are given as

$$\mathbf{A}_c = \begin{pmatrix} 0 & 1 \\ -a_0 & -a_1 \end{pmatrix}, \quad \mathbf{B}_c = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad (2.63)$$

$$\mathbf{C}_c = (b_0 - a_0b_2 \quad b_1 - a_1b_2), \quad \mathbf{D}_c = b_2. \quad (2.64)$$

Another possibility to obtain the solution is to separate the absolute term and the strictly proper part by dividing the numerator by the denominator

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_2s^2 + b_1s + b_0}{s^2 + a_1s + a_0} \quad (2.65)$$

$$= b_2 + \frac{(b_1 - a_1b_2)s + (b_0 - a_0b_2)}{s^2 + a_1s + a_0}. \quad (2.66)$$

The absolute term represents the matrix \mathbf{D} and the strictly proper part corresponds to the triplet $(\mathbf{A}, \mathbf{B}, \mathbf{C})$.

2.2.2 Two Tanks with Interaction

We study the dynamic properties of two tanks with interaction from Example 1.2.2. There, we derived the state-space representation characterised by the matrices \mathbf{A} , \mathbf{B} , \mathbf{C} and \mathbf{D} :

$$\mathbf{A} = \begin{pmatrix} -\frac{k_1}{F_1} & \frac{k_1}{F_1} \\ \frac{k_1}{F_2} & -\frac{k_1+k_2}{F_2} \end{pmatrix} = \begin{pmatrix} -a & a \\ b & -c \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \frac{1}{F_1} & 0 \\ 0 & \frac{1}{F_2} \end{pmatrix} = \begin{pmatrix} d & 0 \\ 0 & e \end{pmatrix}, \quad (2.67)$$

$$\mathbf{C} = (0 \quad 1), \quad \mathbf{D} = (0 \quad 0). \quad (2.68)$$

2.2.2.1 Derive all the transfer functions using the Laplace transform of the state equations.

The process involves two inputs and one output. Therefore, two transfer functions can be derived

$$G_1(s) = \frac{Y(s)}{U_1(s)}, \quad G_2(s) = \frac{Y(s)}{U_2(s)}. \quad (2.69)$$

Matrices \mathbf{A} and \mathbf{B} correspond to state differential equations:

$$\frac{dx_1(t)}{dt} = -ax_1(t) + ax_2(t) + du_1(t), \quad (2.70)$$

$$\frac{dx_2(t)}{dt} = bx_1(t) - cx_2(t) + eu_2(t). \quad (2.71)$$

In the first step, we apply the Laplace transform and under zero initial conditions:

$$sX_1(s) = -aX_1(s) + aX_2(s) + dU_1(s), \quad (2.72)$$

$$sX_2(s) = bX_1(s) - cX_2(s) + eU_2(s). \quad (2.73)$$

As the output variable is defined by $y(t) = x_2(t)$, we solve the first equation for $X_1(s)$ and substitute the result into the second equation. This gives

$$Y(s) = X_2(s) = \frac{bd}{s^2 + s(a+c) + a(c-b)}U_1(s) + \frac{e(s+a)}{s^2 + s(a+c) + a(c-b)}U_2(s). \quad (2.74)$$

This results in two transfer functions G_1 for $(U_2(s) = 0)$ and G_2 for $(U_1(s) = 0)$

$$G_1(s) = \frac{Y(s)}{U_1(s)} = \frac{bd}{s^2 + s(a+c) + a(c-b)} = \frac{k_1}{F_1F_2s^2 + (k_1F_1 + k_1F_2 + k_2F_1)s + k_1k_2}, \quad (2.75)$$

$$G_2(s) = \frac{Y(s)}{U_2(s)} = \frac{e(s+a)}{s^2 + s(a+c) + a(c-b)} = \frac{F_1s + k_1}{F_1F_2s^2 + (k_1F_1 + k_1F_2 + k_2F_1)s + k_1k_2}. \quad (2.76)$$

2.2.2.2 Derive all the transfer functions using the matrix-based approach from (2.13).

This approach directly gives the overall transfer function matrix. Its derivation using MATLAB is given at the end of this example and also in Section 2.5.

$$s\mathbf{I} - \mathbf{A} = \begin{pmatrix} s+a & -a \\ -b & s+c \end{pmatrix}, \quad (2.77)$$

$$|s\mathbf{I} - \mathbf{A}| = s^2 + s(a+c) + a(c-b), \quad (2.78)$$

$$(s\mathbf{I} - \mathbf{A})^{-1} = \frac{1}{|s\mathbf{I} - \mathbf{A}|} \begin{pmatrix} s+c & a \\ b & s+a \end{pmatrix}, \quad (2.79)$$

$$\mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \quad (2.80)$$

$$= \frac{1}{s^2 + s(a+c) + a(c-b)} \begin{pmatrix} bd & e(s+a) \end{pmatrix} \quad (2.81)$$

$$= \begin{pmatrix} G_1(s) & G_2(s) \end{pmatrix}. \quad (2.82)$$

2.2.2.3 Derive a state-space representation from the previously derived transfer functions.

We observe that both transfer functions have the same denominator. Therefore, we can start by converting the common part into a state-space representation using intermediate variables $Z_1(s) = G(s)U_1(s)$ and $Z_2(s) = G(s)U_2(s)$

$$G(s) = \frac{Z_1(s)}{U_1(s)} = \frac{Z_2(s)}{U_2(s)} = \frac{1}{F_1F_2s^2 + (k_1F_1 + k_1F_2 + k_2F_1)s + k_1k_2}. \quad (2.83)$$

We convert the common transfer function into the controllable state-space representation (e.g., manipulating the expressions for $Z_1(s)/U_1(s)$) – see Section 2.1.4

$$\dot{x}_1 = x_2, \quad (2.84)$$

$$\dot{x}_2 = (u_1 - k_1 k_2 x_1 - (k_1 F_1 + k_1 F_2 + k_2 F_1) x_2) / (F_1 F_2), \quad (2.85)$$

$$z_1 = x_1. \quad (2.86)$$

We apply the similar procedure for $Z_2(s)/U_2(s)$ defining the states x_3 and x_4 , where $z_2 = x_4$.

Then, we can express the output as a combination of two signals

$$Y(s) = Y_1(s) + Y_2(s) \quad (2.87)$$

$$= k_1 Z_1(s) + (F_1 s + k_1) Z_2(s) \quad (2.88)$$

$$= k_1 G(s) U_1(s) + (F_1 s + k_1) G(s) U_2(s). \quad (2.89)$$

The overall state-space representation is then obtained by combining the partial controllable state-space representations of the common transfer function $G(s)$

$$\dot{x}_1 = x_2, \quad (2.90)$$

$$\dot{x}_2 = \frac{u_1 - k_1 k_2 x_1 - (k_1 F_1 + k_1 F_2 + k_2 F_1) x_2}{F_1 F_2}, \quad (2.91)$$

$$\dot{x}_3 = x_4, \quad (2.92)$$

$$\dot{x}_4 = \frac{u_2 - k_1 k_2 x_3 - (k_1 F_1 + k_1 F_2 + k_2 F_1) x_4}{F_1 F_2}, \quad (2.93)$$

$$y_1 = k_1 x_1, \quad (2.94)$$

$$y_2 = k_1 x_3 + F_1 x_4, \quad (2.95)$$

$$y = y_1 + y_2 = k_1 x_1 + k_1 x_3 + F_1 x_4. \quad (2.96)$$

2.2.2.4 Simulate the step responses of all the representations.

Step responses of all representations are the same even if the state-space representations do not share the same number of states.

```

q01s = 0.9; q02s = 0.0; F1 = 0.5; F2 = 0.6; k11 = 0.8; k22 = 0.5;
h2s = ((q01s+q02s)/k22)^2; h1s = (q01s/k11)^2 + h2s;
us = [q01s; q02s]; ys = h2s; xs = [h1s; h2s];
k1=k11/(2*sqrt(h1s-h2s)); k2=k22/(2*sqrt(h2s));
A = [-k1/F1 k1/F1; k1/F2 -(k1+k2)/F2]; B = [1/F1 0; 0 1/F2];
C = [0 1]; D = 0;
G1 = tf(k1, [F1*F2 k1*F1+k1*F2+k2*F1 k1*k2]);
G2 = tf([F1 k1], [F1*F2 k1*F1+k1*F2+k2*F1 k1*k2]);
An = [0 1 0 0
      -k1*k2/(F1*F2) -(k1*F1+k1*F2+k2*F1)/(F1*F2) 0 0
      0 0 0 1
      0 0 -k1*k2/(F1*F2) -(k1*F1+k1*F2+k2*F1)/(F1*F2)];
Bn = [0 0 ; 1 0; 0 0; 0 1];
Cn = [k1 0 k1 F1];
Dn = [0 0];
sys1 = ss(A,B,C,D); sys2 = [G1 G2]; sys3 = ss(An,Bn,Cn,Dn);
step(sys1, sys2, sys3); grid on

```


2.2.3 Two Tanks with a Pump

We study the dynamic properties of two tanks with interaction modelled in Section 1.2.3. The derived state-space representation is characterised by the matrices \mathbf{A} , \mathbf{B} , \mathbf{C} , and \mathbf{D} :

$$\mathbf{A} = \begin{pmatrix} -\frac{k_1}{F_1} & \frac{k_1}{F_1} \\ \frac{k_1}{F_2} & -\frac{k_1}{F_2} \end{pmatrix} = \begin{pmatrix} -a & a \\ b & -b \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \frac{1}{F_1} & 0 \\ 0 & -\frac{1}{F_2} \end{pmatrix} = \begin{pmatrix} d & 0 \\ 0 & -e \end{pmatrix}, \quad (2.97)$$

$$\mathbf{C} = \begin{pmatrix} 0 & 1 \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} 0 & 0 \end{pmatrix}. \quad (2.98)$$

2.2.3.1 Derive the transfer functions using the matrix-based approach from (2.13).

Application of (2.13) gives

$$s\mathbf{I} - \mathbf{A} = \begin{pmatrix} s+a & -a \\ -b & s+b \end{pmatrix}, \quad |s\mathbf{I} - \mathbf{A}| = s^2 + s(a+b), \quad (2.99)$$

$$(s\mathbf{I} - \mathbf{A})^{-1} = \frac{1}{|s\mathbf{I} - \mathbf{A}|} \begin{pmatrix} s+b & a \\ b & s+a \end{pmatrix}, \quad (2.100)$$

$$\mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \quad (2.101)$$

$$= \frac{1}{s^2 + s(a+b)} \begin{pmatrix} bd & -e(s+a) \end{pmatrix} \quad (2.102)$$

$$= \begin{pmatrix} \frac{k_1}{F_1 F_2 s^2 + (k_1 F_1 + k_1 F_2)s} & \frac{-F_1 s - k_1}{F_1 F_2 s^2 + (k_1 F_1 + k_1 F_2)s} \end{pmatrix}. \quad (2.103)$$

Note that the denominator has one root at $s = 0$ which indicates an integrating dynamics. Also, the negative sign of the second transfer function results from the inverse character of this process — an increase in the pump revolutions gives a higher flow out of the tank and thus, a level decrease in the second tank.

2.3 Practice Examples

2.3.1 A One-Dimensional Car

We study the movement of a car modelled in Section 1.3.1.

2.3.1.1 Recall the choice of the input and the state variables of the system. Select the output variables of the system while assuming that we measure both the position and velocity of the car. Write down the linear state-space representation of the system:

$$\frac{dx_1(t)}{dt} = x_2(t), \quad (2.104)$$

$$\frac{dx_2(t)}{dt} = u(t). \quad (2.105)$$

Input variables: acceleration $u(t)$, state variables: position $x_1(t)$ and velocity $x_2(t)$. The output variables are the both states, $y_1(t) = x_1(t)$ and $y_2(t) = x_2(t)$. The

state-space representation of the dynamic system is:

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}}_{\mathbf{A}} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{\mathbf{B}} u(t), \quad (2.106)$$

$$\begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}_{\mathbf{C}} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \underbrace{\begin{pmatrix} 0 \\ 0 \end{pmatrix}}_{\mathbf{D}} u(t). \quad (2.107)$$

2.3.1.2 Derive the transfer matrix of the system.

From the definition of the transfer matrix (Laplace transform of the state-space model with zero initial conditions), we obtain:

$$\mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} s & -1 \\ 0 & s \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \quad (2.108)$$

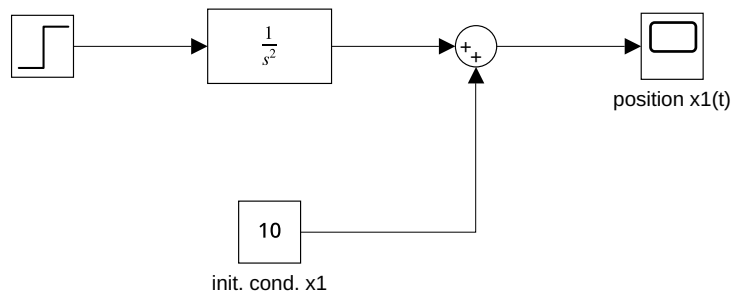
The inverse of the matrix can be found, using the determinant and the adjoint matrix, by:

$$\begin{pmatrix} s & -1 \\ 0 & s \end{pmatrix}^{-1} = \frac{1}{s^2} \begin{pmatrix} s & 1 \\ 0 & s \end{pmatrix}. \quad (2.109)$$

The resulting transfer function matrix is:

$$\mathbf{G}(s) = \frac{1}{s^2} \begin{pmatrix} s & 1 \\ 0 & s \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{s^2} \begin{pmatrix} 1 \\ s \end{pmatrix} = \begin{pmatrix} \frac{1}{s^2} \\ \frac{1}{s} \end{pmatrix} = \begin{pmatrix} G_{y_1 u}(s) \\ G_{y_2 u}(s) \end{pmatrix}. \quad (2.110)$$

2.3.1.3 Create a Simulink model that will simulate the position of the car using one of the derived transfer functions for some given values of the initial conditions and the inputs.



2.3.1.4 Run the simulation setting the initial conditions to be 10 m for the position and 0 m/s for the velocity. Verify the correctness of the results by comparing with the results of the simulation created in the previous chapter. Plot the simulation results as the output variable as a function of time.

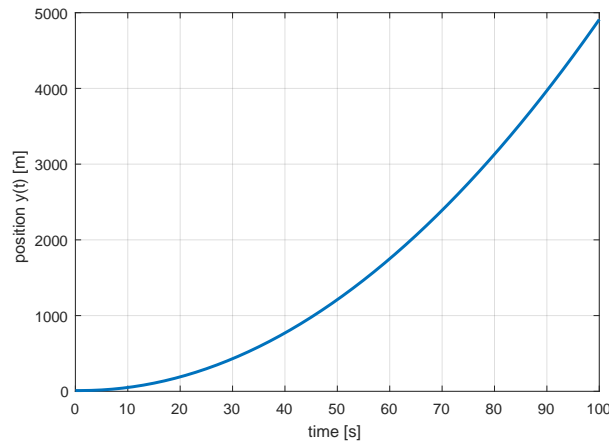


Figure 2.2: Time reponse of the car position.

```
plot(tx(:,1), tx(:,2), 'LineWidth', 2);
xlabel('time [s]'); ylabel('position y(t) [m]'); grid on
iam_save_pdf('x_t.pdf')
```

Simulation results are shown in Figure 2.2.

2.3.2 Two Tanks in Series

We study a system with two tanks in series described in Example 1.2.4 with a modification that both tanks have independent inlet flows with flow rates $q_{0,1}$ and $q_{0,2}$ as shown in Figure 2.3.

The values of parameters and initial signals of the process are: $q_{0,1}^s = 0.3 \text{ m}^3/\text{s}$, $q_{0,2}^s = 0.5 \text{ m}^3/\text{s}$, $F_1 = 0.5 \text{ m}^2$, $F_2 = 0.8 \text{ m}^2$, $k_{11} = 1.15 \text{ m}^{2.5}/\text{s}$, and $k_{22} = 1.3 \text{ m}^{2.5}/\text{s}$.

2.3.2.1 Select the input, state, and output variables of the system.

Input variables: liquid inlet flows $q_{0,1}$ and $q_{0,2}$, state variables: height of the liquid in the tanks $h_1(t)$ and $h_2(t)$, output variables: $h_2(t)$.

2.3.2.2 Write down the mass balance of the liquid in the tanks.

The mass balance of the first tank (from the previous chapter) is

$$\frac{dh_1(t)}{dt} = \frac{q_{0,1}(t)}{F_1} - \frac{k_{11}}{F_1} \sqrt{h_1(t)}. \quad (2.111)$$

The mass balance of the second tank is

$$\frac{dh_2(t)}{dt} = \frac{q_{0,2}(t)}{F_2} + \frac{k_{11}}{F_2} \sqrt{h_1(t)} - \frac{k_{22}}{F_2} \sqrt{h_2(t)}. \quad (2.112)$$

2.3.2.3 Recall the steady-state value of the liquid level in the tanks from the previous chapter. Adapt these expressions for the tanks in the system at hand.

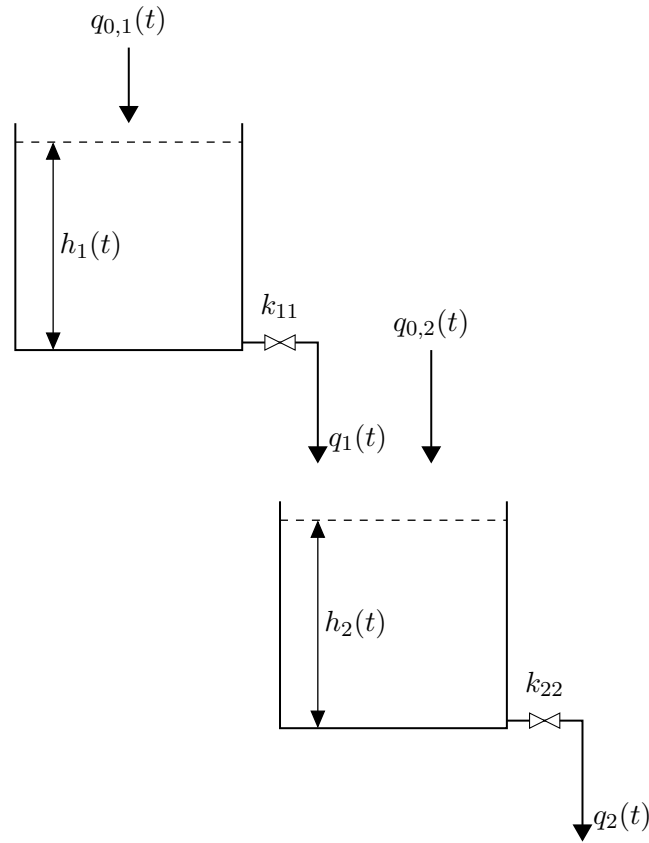


Figure 2.3: Schematic diagram of two tanks in series.

The steady-state value of the liquid level in the tank 1 can be found from

$$h_1^s = \left(\frac{q_{0,1}^s}{k_{11}} \right)^2. \quad (2.113)$$

From (2.112) and (2.113), the steady state of the level in the tank 2 is given by:

$$h_2^s = \left(\frac{q_{0,2}^s}{k_{22}} + \frac{k_{11}}{k_{22}} \sqrt{h_1^s} \right)^2 = \left(\frac{q_{0,1}^s + q_{0,2}^s}{k_{22}} \right)^2. \quad (2.114)$$

2.3.2.4 If necessary, linearise the model. Write down the state-space representation of the linearised model.

The derived model is nonlinear because of the presence of the square roots of the state variables in (2.111), (2.112). For the linearisation, we first define the deviation variables:

$$x_1(t) = h_1(t) - h_1^s, \quad u_1(t) = q_{0,1}(t) - q_{0,1}^s, \quad (2.115)$$

$$x_2(t) = h_2(t) - h_2^s, \quad u_2(t) = q_{0,2}(t) - q_{0,2}^s. \quad (2.116)$$

The linearised model is then given by:

$$\frac{dx_1(t)}{dt} = a_{11}x_1(t) + b_{11}u_1(t), \quad \frac{dx_2(t)}{dt} = a_{21}x_1(t) + a_{22}x_2(t) + b_{22}u_2(t), \quad (2.117)$$

where

$$a_{11} = -\frac{k_{11}}{2F_1\sqrt{h_1^s}} \stackrel{\text{def}}{=} -\frac{k_1}{F_1}, \quad b_{11} = \frac{1}{F_1}, \quad (\text{from the previous chapter}) \quad (2.118)$$

$$a_{21} = \left. \frac{\partial \frac{q_{0,2}(t)}{F_2} + \frac{k_{11}}{F_2}\sqrt{h_1(t)} - \frac{k_{22}}{F_2}\sqrt{h_2(t)}}{\partial h_1(t)} \right|_{h_1^s, h_2^s, q_{0,1}^s} = \frac{k_{11}}{2F_2\sqrt{h_1^s}} \stackrel{\text{def}}{=} \frac{k_1}{F_2}, \quad (2.119)$$

$$a_{22} = \left. \frac{\partial \frac{q_{0,2}(t)}{F_2} + \frac{k_{11}}{F_2}\sqrt{h_1(t)} - \frac{k_{22}}{F_2}\sqrt{h_2(t)}}{\partial h_2(t)} \right|_{h_1^s, h_2^s, q_{0,1}^s} = -\frac{k_{22}}{2F_2\sqrt{h_2^s}} \stackrel{\text{def}}{=} -\frac{k_2}{F_2}, \quad (2.120)$$

$$b_{22} = \left. \frac{\partial \frac{q_{0,2}(t)}{F_2} + \frac{k_{11}}{F_2}\sqrt{h_1(t)} - \frac{k_{22}}{F_2}\sqrt{h_2(t)}}{\partial q_{0,2}(t)} \right|_{h_1^s, h_2^s, q_{0,1}^s} = \frac{1}{F_2}. \quad (2.121)$$

The state-space model is then

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \underbrace{\begin{pmatrix} -\frac{k_1}{F_1} & 0 \\ \frac{k_1}{F_2} & -\frac{k_2}{F_2} \end{pmatrix}}_{\mathbf{A}} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \underbrace{\begin{pmatrix} \frac{1}{F_1} & 0 \\ 0 & \frac{1}{F_2} \end{pmatrix}}_{\mathbf{B}} \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix}, \quad (2.122a)$$

$$y(t) = \underbrace{\begin{pmatrix} 0 & 1 \end{pmatrix}}_{\mathbf{C}} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \underbrace{\begin{pmatrix} 0 & 0 \end{pmatrix}}_{\mathbf{D}} \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix}. \quad (2.122b)$$

2.3.2.5 Derive the transfer matrix of the system.

From the definition of the transfer matrix (Laplace transform of the state-space model with zero initial conditions), we obtain:

$$\mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} = \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} s + \frac{k_1}{F_1} & 0 \\ -\frac{k_1}{F_2} & s + \frac{k_2}{F_2} \end{pmatrix}^{-1} \begin{pmatrix} \frac{1}{F_1} & 0 \\ 0 & \frac{1}{F_2} \end{pmatrix} + \begin{pmatrix} 0 & 0 \end{pmatrix}. \quad (2.123)$$

The inverse of a matrix can be found using its determinant and adjoint matrix:

$$\begin{pmatrix} s + \frac{k_1}{F_1} & 0 \\ -\frac{k_1}{F_2} & s + \frac{k_2}{F_2} \end{pmatrix}^{-1} = \frac{1}{\left(s + \frac{k_1}{F_1}\right)\left(s + \frac{k_2}{F_2}\right)} \begin{pmatrix} s + \frac{k_2}{F_2} & 0 \\ \frac{k_1}{F_2} & s + \frac{k_1}{F_1} \end{pmatrix}, \quad (2.124)$$

which results in:

$$\mathbf{G}(s) = \begin{pmatrix} 0 & 1 \end{pmatrix} \frac{1}{\left(s + \frac{k_1}{F_1}\right)\left(s + \frac{k_2}{F_2}\right)} \begin{pmatrix} s + \frac{k_2}{F_2} & 0 \\ \frac{k_1}{F_2} & s + \frac{k_1}{F_1} \end{pmatrix} \begin{pmatrix} \frac{1}{F_1} & 0 \\ 0 & \frac{1}{F_2} \end{pmatrix} \quad (2.125)$$

$$= \begin{pmatrix} 0 & 1 \end{pmatrix} \frac{1}{\left(s + \frac{k_1}{F_1}\right)\left(s + \frac{k_2}{F_2}\right)} \begin{pmatrix} \frac{1}{F_1}\left(s + \frac{k_2}{F_2}\right) & 0 \\ \frac{k_1}{F_1 F_2} & \frac{1}{F_2}\left(s + \frac{k_1}{F_1}\right) \end{pmatrix} \quad (2.126)$$

$$= \begin{pmatrix} \frac{\frac{k_1}{F_1 F_2}}{\left(s + \frac{k_1}{F_1}\right)\left(s + \frac{k_2}{F_2}\right)} & \frac{\frac{1}{F_2}}{s + \frac{k_2}{F_2}} \end{pmatrix} = \begin{pmatrix} G_{yu_1}(s) & G_{yu_2}(s) \end{pmatrix}. \quad (2.127)$$

2.3.2.6 Verify the correctness of the derived transfer function $G_{yu_1}(s)$ applying the algebra of transfer functions on the individual transfer functions of the first and second tanks with respect to input $u_1(t)$ derived using the explicit Laplace transform.

The Laplace transform of the first linearised differential equation gives:

$$sX_1(s) = -\frac{k_1}{F_1}X_1(s) + \frac{1}{F_1}U_1(s) \Rightarrow \left(s + \frac{k_1}{F_1}\right)X_1(s) = \frac{1}{F_1}U_1(s). \quad (2.128)$$

We will define a temporary output variable $Y_1(s) = X_1(s)$. The respective transfer function can be derived as:

$$G_{y_1u_1}(s) = \frac{Y_1(s)}{U_1(s)} = \frac{\frac{1}{F_1}}{s + \frac{k_1}{F_1}}. \quad (2.129)$$

The Laplace transform of the second linearised differential equation gives:

$$sX_2(s) = \frac{k_1}{F_2}X_1(s) - \frac{k_2}{F_2}X_2(s) \Rightarrow \left(s + \frac{k_2}{F_2}\right)X_2(s) = \frac{k_1}{F_2}X_1(s). \quad (2.130)$$

Using the definition of the output variable $Y(s) = X_2(s)$, the transfer function is

$$G_{yy_1}(s) = \frac{Y}{Y_1(s)} = \frac{\frac{k_1}{F_2}}{s + \frac{k_2}{F_2}}. \quad (2.131)$$

The resulting transfer function is

$$G_{yu_1}(s) = \frac{Y(s)}{U_1(s)} = G_{y_1u_1}(s)G_{yy_1}(s) = \frac{\frac{k_1}{F_1F_2}}{\left(s + \frac{k_1}{F_1}\right)\left(s + \frac{k_2}{F_2}\right)}. \quad (2.132)$$

2.3.2.7 Verify the equivalence of the state-space representation and the transfer function $G_{yu_1}(s)$ using a simulation. Apply a unit step increase of $q_{0,1}$ at $t = 1$ and a unit step increase of $q_{0,2}$ at $t = 5$ of the simulation.

We build the Simulink model (Figure 2.4) with the parameters

```
q01s = 0.3; q02s = 0.5; k11 = 1.15; k22 = 1.3;
h1s = (q01s/k11)^2; h2s = (q01s/k22 + q02s/k22)^2;
k1 = k11/2/sqrt(h1s); k2 = k22/2/sqrt(h2s); F1 = 0.5; F2 = 0.8;
```

Simulation results can be plotted and are shown in Figure 2.5.

```
plot(th(:,1), th(:,2), th(:,1), th(:,3), 'linewidth', 2);
xlabel('time [s]'); ylabel('level h(t) [m]'); grid on
legend('SS', 'TF');
iam_save_pdf('h_t_ss_tf.pdf')
```

We can observe, that both dynamic models coincide until time 5 of the simulation. As the transfer-function model cannot handle the second input to the state-space system, the outputs differ if the second input is changed.

2.3.2.8 Repeat the previous simulation and compare the results obtained using the state-space and the nonlinear models. Rerun the simulation with a 10% change in process inputs.

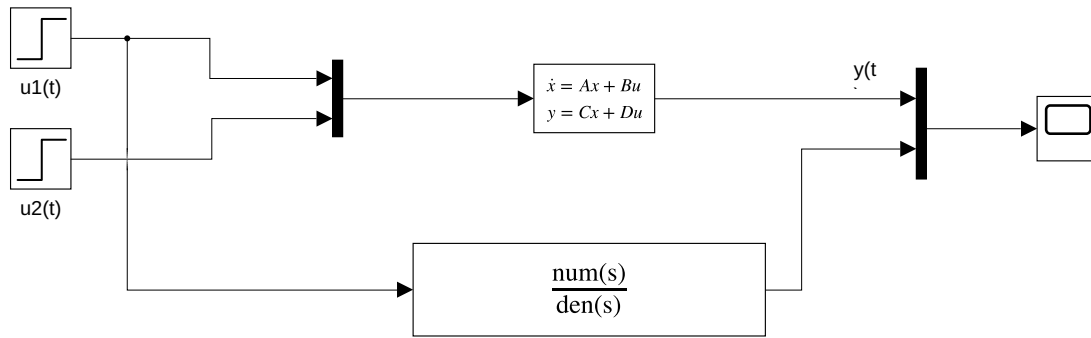


Figure 2.4: A Simulink model to compare the state-space and transfer-function models.

We build the nonlinear model in Simulink from the basic building blocks. For the linear model, we use the state-space block but care has to be taken of its inputs and outputs with respect to deviation variables. The final model is shown in Figure 2.6 and simulation results in Figure 2.7 (left). We can see that the models behave quite differently. This is caused by a large change in the operating point. Thus, the linearisation is then very inaccurate.

We repeat the simulation, but with a 10% change in the inlet flows. This time, both models show similar behaviour – Figure 2.7 (right). Thus, the linear state-space model can be used in place of the full nonlinear model.

2.3.2.9 Assume that the process is in the initial steady state and that the inlet $q_{0,1}$ is reduced to 90% of its original value. Calculate the new steady-state values of the level h_2 using the nonlinear and transfer function models.

The new steady state for the nonlinear model can be obtained from (2.114)

$$h_2^s = \left(\frac{q_{0,1}^s + q_{0,2}^s}{k_{22}} \right)^2 = \left(\frac{0.3 \cdot 0.9 + 0.5}{1.3} \right)^2 = 0.35 \text{ m.} \quad (2.133)$$

An approximate value can be also found from the linear model and transfer functions

$$Y(s) = G_{yu_1}(s)U_1(s) + G_{yu_2}(s)U_2(s). \quad (2.134)$$

Since $u_1 = q_{0,1} - q_{0,1}^s = -0.1q_{0,1}^s$ and $u_2 = q_{0,2} - q_{0,2}^s = 0$, it follows that

$$Y(s) = G_{yu_1}(s)U_1(s) = -\frac{\frac{k_1}{F_1 F_2}}{\left(s + \frac{k_1}{F_1}\right)\left(s + \frac{k_2}{F_2}\right)} \frac{0.1q_{1,0}^s}{s}, \quad (2.135)$$

$$y(\infty) = \lim_{t \rightarrow \infty} y(t) = \lim_{s \rightarrow 0} sY(s) = -\frac{0.1q_{1,0}^s}{k_2}, \quad (2.136)$$

$$h_{2,\text{lin}} = h_2^s + y(\infty) = 0.37 \text{ m.} \quad (2.137)$$

where we used the final value theorem for stable systems (relationship between the final value of a variable in the time domain and the corresponding expression in the Laplace transform) in (2.136).

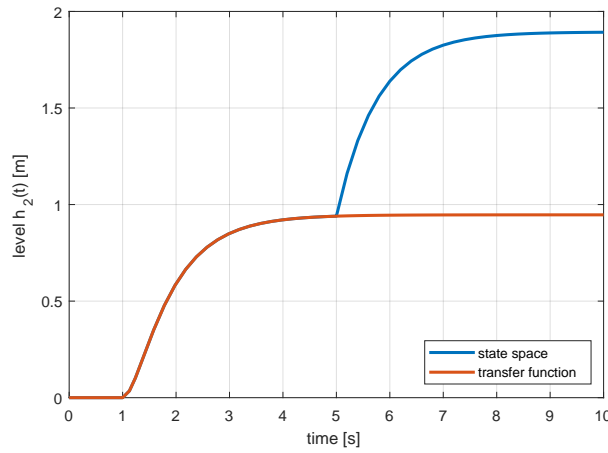


Figure 2.5: Comparison of linear models.

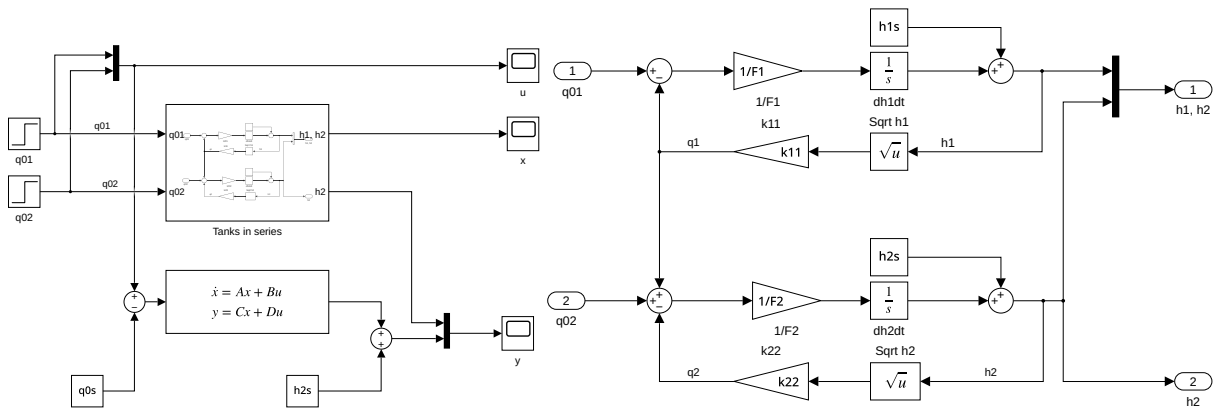


Figure 2.6: A Simulink model to compare the state-space and nonlinear models.

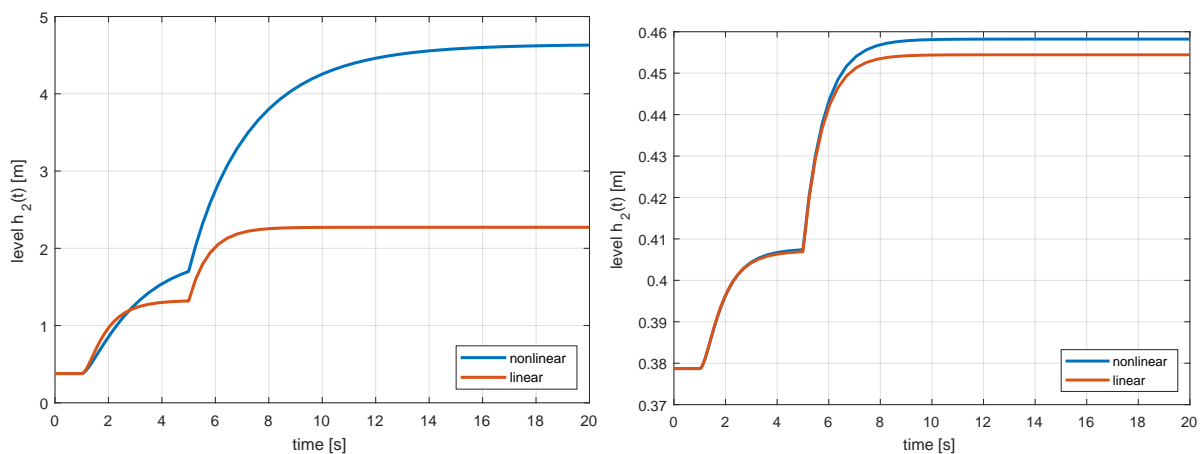


Figure 2.7: Comparison of linear and nonlinear models. (left) Large change in the inputs, (right) small change in the inputs.

2.3.2.10 (Optional) Derive the state-space representation of the transfer function $G_{yu_1}(s)$ in the controllable canonical form.

First, we rewrite the transfer function as:

$$G_{yu_1}(s) = \frac{\frac{k_1}{F_1 F_2}}{s^2 + \left(\frac{k_1}{F_1} + \frac{k_2}{F_2}\right)s + \frac{k_1 k_2}{F_1 F_2}} = \frac{Y(s)}{U_1(s)}. \quad (2.138)$$

We perform the inverse Laplace transform to obtain:

$$\ddot{y}(t) + \left(\frac{k_1}{F_1} + \frac{k_2}{F_2}\right)\dot{y}(t) + \frac{k_1 k_2}{F_1 F_2}y(t) = \frac{k_1}{F_1 F_2}u_1(t). \quad (2.139)$$

We define new state variables

$$\xi_1(t) = y(t), \quad \xi_2(t) = \dot{y}(t). \quad (2.140)$$

We can then write:

$$\dot{\xi}_1(t) = \xi_2(t), \quad \dot{\xi}_2(t) = -\left(\frac{k_1}{F_1} + \frac{k_2}{F_2}\right)\xi_2(t) - \frac{k_1 k_2}{F_1 F_2}\xi_1(t) + \frac{k_1}{F_1 F_2}u_1(t). \quad (2.141)$$

This finally gives:

$$\begin{pmatrix} \dot{\xi}_1(t) \\ \dot{\xi}_2(t) \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ -\frac{k_1 k_2}{F_1 F_2} & -\left(\frac{k_1}{F_1} + \frac{k_2}{F_2}\right) \end{pmatrix}}_{A_c} \begin{pmatrix} \xi_1(t) \\ \xi_2(t) \end{pmatrix} + \underbrace{\frac{k_1}{F_1 F_2}}_{B_c} u_1(t), \quad (2.142)$$

$$y(t) = \underbrace{\begin{pmatrix} 1 & 0 \end{pmatrix}}_{C_c} \begin{pmatrix} \xi_1(t) \\ \xi_2(t) \end{pmatrix} + \underbrace{0}_{D_c} u_1(t). \quad (2.143)$$

2.3.2.11 (Optional) Verify the equivalence of the input-output representation and the derived state-space model in the controllable canonical form.

We implement the model shown in Figure 2.8. Simulations show that the output trajectories are the same.

2.4 Problems

Problem. 2.1: A transfer function from a differential equation

Find a transfer function between the output $y(t)$ and the input $u(t)$ of the system described by the differential equation

$$y^{(3)}(t) + 3\ddot{y}(t) + 12\dot{y}(t) + 4y(t) = 2\ddot{u}(t) + u(t). \quad (2.144)$$

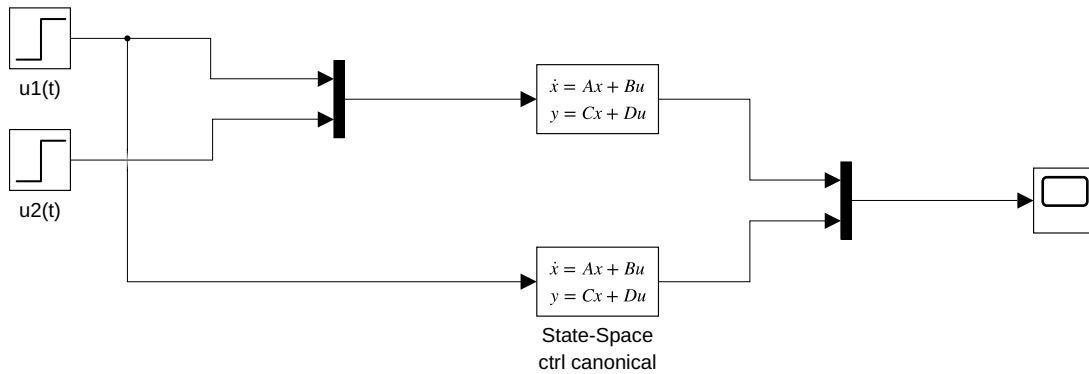


Figure 2.8: A simulation model to compare the linear state-space controllable form and the state-space representation (obtained from the mass balance).

$$G(s) = \frac{Y(s)}{U(s)} = \frac{2s^2 + 1}{s^3 + 3s^2 + 12s + 4}. \quad (2.145)$$

Problem. 2.2: A transfer function from a differential equation

Find transfer functions between the output (variable on the left-hand side) and the input (variable on the right-hand side) of systems described by the differential equations

1. $y^{(3)}(t) + 10\ddot{y}(t) + 30\dot{y}(t) + 20y(t) = 4\ddot{u}(t) + 5\dot{u}(t) + 6u(t)$,
2. $6z^{(5)}(t) + 20z^{(4)}(t) + 2z^{(3)}(t) = v^{(4)}(t) + 3\ddot{v}(t) + v(t)$,
3. $u^{(3)}(t) - 10\dot{u}(t) + 12u(t) = e(t)$,
4. $\dot{x}(t) + 3\sqrt{x(t)} = u(t)$.

$$G_1(s) = \frac{Y(s)}{U(s)} = \frac{4s^2 + 5s + 6}{s^3 + 10s^2 + 30s + 20}, \quad (2.146)$$

$$G_2(s) = \frac{Z(s)}{V(s)} = \frac{s^4 + 3s^2 + 1}{6s^5 + 20s^4 + 2s^3}, \quad (2.147)$$

$$G_3(s) = \frac{U(s)}{E(s)} = \frac{1}{s^3 - 10s + 12}. \quad (2.148)$$

The last problem is not linear, we cannot define the transfer function.

Problem. 2.3: A transfer matrix from a state-space representation

Find the transfer matrix for the system, whose state-space representation is given by the matrices

$$\mathbf{A} = \begin{pmatrix} -1 & 2 & 0 \\ 2 & -2 & 3 \\ 1 & -1 & -1 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}. \quad (2.149)$$

$$G(s) = \begin{pmatrix} \frac{s^2 + 7s + 27}{s^3 + 4s^2 + 4s - 5} & \frac{2s^2 + 12s + 40}{s^3 + 4s^2 + 4s - 5} \\ \frac{2s^2 + 15s + 16}{s^3 + 4s^2 + 4s - 5} & \frac{3s^2 + 22s + 25}{s^3 + 4s^2 + 4s - 5} \\ \frac{6s^2 + 30s + 39}{s^3 + 4s^2 + 4s - 5} & \frac{9s^2 + 45s + 60}{s^3 + 4s^2 + 4s - 5} \end{pmatrix}. \quad (2.150)$$

Problem. 2.4: A differential equation from a transfer function

Find the differential equation describing a PID controller with the output $u(t)$ and input $e(t)$ described by the transfer function

$$G(s) = \frac{U(s)}{E(s)} = \frac{2s^2 + 3s + 1}{s}. \quad (2.151)$$

Note that the degree of the transfer function numerator is higher than the degree of the denominator and the controller cannot be realised.

$$\dot{u}(t) = 2\ddot{e}(t) + 3\dot{e}(t) + e(t). \quad (2.152)$$

A standard representation of a PID controller is obtained when this equation is integrated to get the output $u(t)$

$$u(t) = 2\dot{e}(t) + 3e(t) + \int_0^T e(\tau) d\tau. \quad (2.153)$$

Problem. 2.5: Controllable and observable canonical forms from a transfer function

Find the controllable and observable canonical forms for the transfer function given as

$$G(s) = \frac{Y(s)}{U(s)} = \frac{2s^2 + 1}{s^3 + 3s^2 + 12s + 4}. \quad (2.154)$$

$$\mathbf{A}_c = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -4 & -12 & -3 & -1 \end{pmatrix}, \quad \mathbf{B}_c = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad (2.155)$$

$$\mathbf{C}_c = (1 \ 0 \ 2 \ 0), \quad \mathbf{D}_c = 0, \quad (2.156)$$

$$\mathbf{A}_o = \begin{pmatrix} 0 & 0 & 0 & -4 \\ 1 & 0 & 0 & -12 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & -1 \end{pmatrix}, \quad \mathbf{B}_o = \begin{pmatrix} 1 \\ 0 \\ 2 \\ 0 \end{pmatrix}, \quad (2.157)$$

$$\mathbf{C}_o = (0 \ 0 \ 0 \ 1), \quad \mathbf{D}_o = 0. \quad (2.158)$$

2.5 MATLAB Snippets

- Calculation of the symbolic expression for the transfer matrix of two tanks with interaction by (2.13)

```
syms a b c d e s
A = [-a a ; b -c], B=[d 0 ; 0 e], C=[0, 1], D=[0,0];
G = C*inv(s*eye(2)-A)*B+D;
pretty(G) % pretty print of G using ascii art
```

- Creation of a transfer function

```
G1 = tf([1 2],[1 0 10])
G1delay = tf([1 2],[1 0 10], 'OutputDelay', 1)
s = tf('s'); G2 = exp(-s)*(s+2)/(s^2+10)
G2oli = tf( {[1 2] ; 4} , {[1 0 10] ; [1 1 0]})
```

- Algebra of transfer functions

```
G1 = tf([1 2],[1 0 10]), G2 = tf(1,[1 2 1]),
G3series = G1*G2,
G3paralel = G1+G2,
G3fb = G1*G2/(1+G1*G2)
G3fb = minreal(G3fb) % reduce to a fraction of coprime polynomials
```

- Creation of a state-space representation, conversion of the state-space representation to a transfer matrix and access to its data

```
A=[-1 2 0; 2 -2 3; 1 -1 -1], B = [1 2; 2 3; 3 4]
C=[1 0 0 ; 0 1 0; 1 1 1], D=[0 0; 0 0; 0 0]
abcd = ss(A,B,C,D)
g = tf(abcd)
[num,den] = tfdata(g)
```

- Conversion of a transfer function to a state-space representation and access to its data

```
G1 = tf([1 2],[1 0 10])
abcd = ss(G1)
[A,B,C,D] = ssdata(abcd)
```

- MATLAB uses `canon()` function to perform conversion between canonical forms.

2.6 Python Snippets

- Calculation of the symbolic expression for the transfer matrix of two tanks with interaction by (2.13)

```
import sympy as sym
a, b, c, d, e, s = sym.symbols('a, b, c, d, e, s')
A = sym.Matrix([[ -a, a], [b, -c]])
B = sym.Matrix([[d, 0], [0, e]])
C = sym.Matrix([[0, 1]])
D = sym.Matrix([[0, 0]])
sia = sym.Matrix([[s, 0], [0, s]]) - A
G = C @ sia.inv() @ B + D
```

- Creation of a transfer function

```
import control
G1 = control.tf([1, 2],[1, 0, 10]); print(G1)
s = control.tf('s'); G2 = (s + 2) / (s**2 + 10); print(G2)
num = [[[1, 2]], [[4]]]; den = [[[1, 0, 10]], [[1, 1, 0]]];
G2oli = control.tf(num, den); print(G2oli)
```

- Algebra of transfer functions

```
from control.matlab import *
G1 = tf([1, 2],[1, 0, 10]); G2 = tf(1,[1, 2, 1])
G3series = G1 * G2
G3paralel = G1 + G2
G3fb = G1 * G2 / (1 + G1 * G2)
G3fb = minreal(G3fb) #reduce to a fraction of coprime polynomials
```

- Creation of a state-space representation, conversion from the state-space representation to a transfer function and access to its data

```
from control.matlab import *
A = [[-1, 2, 0], [2, -2, 3], [1, -1, -1]]
B = [[1, 2], [2, 3], [3, 4]]
C = [[1, 0, 0], [0, 1, 0], [1, 1, 1]]
D = [[0, 0], [0, 0], [0, 0]]
abcd = ss(A, B, C, D)
g = tf(abcd)
num, den = tfdata(g)
```

- Conversion of a transfer function to a state-space representation and access to its data

```
from control.matlab import *
G1 = tf([1, 2],[1, 0, 10])
abcd = ss(G1)
A, B, C, D = ssdata(abcd)
```

- python-control provides function `canonical_form(sys, form='reachable')`, where the form can be 'reachable', 'observable' or 'modal'.

2.7 Bibliography

Transfer functions play a pivotal role in continuous-time control analysis and design. The theoretical background for this chapter can be found in Mikleš and Fikar (2007, Chapter 3), information in Slovak in Bakošová and Fikar (2008, Chapter 2).

Further sources of information are for example Ogata (2010, Chapter 2), Golnaraghi (2010, Chapter 3), or Dorf and Bishop (2008, Chapter 2).

CHAPTER 3

Poles, Zeros, and Their Effects on Process Dynamics

In this chapter, you will study the basic characteristics of dynamic systems in time domain. In particular:

- determination of the step response for selected processes,
- characterisation of step responses,
- relationship between the locations of the poles and zeros and the time-domain characteristics of processes,
- relationship between the step, impulse, and ramp responses.

3.1 Overview

We already know how to model processes using differential equations or transfer functions and how to convert them to each other. If we want to study the behaviour of processes, we can examine their responses to standard input signals. The behaviour of the process is determined by its dynamics and by the chosen input signal. We can expect that two different input signals will result in different dynamic responses of the process. Also, if two different processes will respond to the same input signal, their responses will be different.

There are four commonly used standard input signals: step, impulse, ramp, and sine wave. In this chapter, we will work mainly with step input, but will consider the impulse and ramp signals. Process response to a sinusoid will be studied in the next chapter.

A transfer function can be represented as:

1. Ratio of two polynomials

$$G(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_0}, \quad n \geq m. \quad (3.1)$$

2. Poles and zeros

$$G(s) = \frac{b_m(s - z_1)(s - z_2) \dots (s - z_m)}{a_n(s - p_1)(s - p_2) \dots (s - p_n)}. \quad (3.2)$$

Roots of the characteristic polynomial (denominator of the transfer function) p_1, \dots, p_n are system poles. Roots of the numerator polynomial z_1, \dots, z_m are system zeros. Note that p_i or z_i can occur in complex conjugate pairs.

If the transfer function is stable, minimum phase, with real poles and zeros that are all negative, it is common to write the transfer function as

$$G(s) = k \frac{(s + z_1)(s + z_2) \dots (s + z_m)}{(s + p_1)(s + p_2) \dots (s + p_n)}. \quad (3.3)$$

with p_i and z_i being positive.

3. Time constants

$$G(s) = K \frac{(T_{N,1}s + 1)(T_{N,2}s + 1) \dots (T_{N,m}s + 1)}{(T_1s + 1)(T_2s + 1) \dots (T_ns + 1)}, \quad (3.4)$$

where $T_1, \dots, T_n, T_{N,1}, \dots, T_{N,m}$ are the time constants and K is the system (static) gain. This expression for transfer function can only be written if all poles and zeros are real. Time constants correspond to negative inverses of poles and zeros. Therefore, as time constants are in units of time, poles and zeros are in units of frequency. Note that this form of representation cannot capture any possible integrators or differentiators.

Static Gain

The static gain K of a stable system with a transfer function $G(s)$ is defined as the value of the output $y(t)$ at steady state ($t \rightarrow \infty$) if the system input is a unit step change. It can be determined using the Final Value Theorem

$$K = \lim_{t \rightarrow \infty} y(t) = \lim_{s \rightarrow 0} sY(s) = \lim_{s \rightarrow 0} sG(s)U(s) = \lim_{s \rightarrow 0} sG(s) \frac{1}{s} = G(0). \quad (3.5)$$

Note, that the above formula holds only for stable systems, *i.e.*, a transfer function with all poles having negative real parts. If we consider the transfer function (3.1) then

$$K = \frac{b_0}{a_0}. \quad (3.6)$$

If the process is in an initial steady state with output value y_0 and the input step change is $\Delta u = u_\infty - u_0$ then the static gain can be determined as the ratio of the changes in the output to the input

$$K = \frac{y_\infty - y_0}{u_\infty - u_0}, \quad (3.7)$$

where y_∞ is the value of the output at the new steady state.

Impulse Response

If the input to the system is the unit impulse (Dirac delta) with $U(s) = 1$ then the process reacts with an impulse response

$$Y(s) = G(s)U(s) = G(s), \quad y(t) = g(t), \quad (3.8)$$

where the impulse response $g(t)$ is also called the weighting function.

In general, the impulse response can be obtained as a time derivative of the step response.

3.1.1 Step Response of a First-Order System

Consider a first-order system described by the differential equation

$$a_1 \dot{y}(t) + a_0 y(t) = b_0 u(t), \quad (3.9)$$

and with the transfer function of the form

$$G(s) = \frac{b_0}{a_1 s + a_0} = \frac{K}{Ts + 1}, \quad K = \frac{b_0}{a_0}, \quad T = \frac{a_1}{a_0}, \quad (3.10)$$

$$= k \frac{1}{s + p}, \quad k = \frac{b_0}{a_1} = \frac{K}{T}, \quad p = \frac{a_0}{a_1} = \frac{1}{T}. \quad (3.11)$$

Let $u(t)$ be the unit step input with the corresponding Laplace image $U(s) = 1/s$. Then, assuming zero initial conditions, the system output is

$$Y(s) = G(s)U(s) = \frac{K}{s(Ts + 1)}. \quad (3.12)$$

The partial fraction expansion of this expression gives

$$\frac{K}{s(Ts + 1)} = \frac{k_1}{s} + \frac{k_2}{Ts + 1}. \quad (3.13)$$

There are several methods to obtain the coefficients k_1 and k_2 . One of them is to multiply the equation by the denominator $s(Ts + 1)$ and equate the powers of s on both sides. This gives

$$K = k_1(Ts + 1) + k_2 s, \quad (3.14)$$

from which it follows

$$s^0 : K = k_1, \quad (3.15a)$$

$$s^1 : 0 = Tk_1 + k_2. \quad (3.15b)$$

If the poles are real and distinct, we can multiply both sides of the equation by a pole and evaluate the resulting equation at the pole. For example, multiplication by s and evaluating at $s = 0$ directly gives k_1

$$\frac{K}{Ts + 1} = k_1 + \frac{k_2 s}{Ts + 1}, \quad (3.16)$$

$$K = k_1. \quad (3.17)$$

Similarly, we can multiply the equation by $(Ts + 1)$ and evaluate at $s = -1/T$ to get k_2

$$\frac{K}{s} = \frac{k_1(Ts + 1)}{s} + k_2, \quad (3.18)$$

$$\frac{K}{-\frac{1}{T}} = -KT = k_2. \quad (3.19)$$

Therefore, the step response in Laplace and time domains is given as

$$Y(s) = K \frac{1}{s} - KT \frac{1}{Ts + 1} = K \left(\frac{1}{s} - \frac{1}{s + \frac{1}{T}} \right), \quad (3.20)$$

$$y(t) = K \left(1 - e^{-\frac{t}{T}} \right). \quad (3.21)$$

Step responses for various values of K and T are shown in Figure 3.1. We can observe that increasing the value of T stretches the graph horizontally and increasing the value of K stretches it vertically.

The step response function (3.21) can be used to determine the values of the parameters K and T from the step response plot. Clearly, $K = y(\infty)$ which also follows from (3.21). As far as the time constant is concerned, there are several methods. We can, for example, observe that

$$y(T) = K(1 - e^{-1}) = 0.63K \approx \frac{2}{3}K. \quad (3.22)$$

Therefore, it can be obtained as the time when the step response reaches roughly 2/3 of its steady-state value.

There are several time-domain indices that can characterise the step response of a first-order system. All of them can be expressed as linear functions of T .

Settling time T_ϵ is given as the time after which the output value remains within some prescribed percentage distance ϵ around the steady state. Similarly to the preceding discussion (3.22)

$$1 - e^{-\frac{T_\epsilon}{T}} = 1 - \epsilon. \quad (3.23)$$

The usual considered values are $\epsilon = 0.02$ or $\epsilon = 0.05$, which gives

$$T_{0.05} \approx 3T, \quad T_{0.02} \approx 4T. \quad (3.24)$$

Therefore, another possibility to estimate the time constant from the step response is using the fact that it takes approximately 3–5 time constants for the step response to settle down (to reach the steady state).

Rise time T_c is the time required for the response to rise from 0% to $c\%$ of its final value. For underdamped responses, a time point T_{100} is considered (time when the output signal reaches its final value for the first time). For overdamped responses, a time point T_{90} is commonly used as the time elapsed between reaching 10% and 90% of the new steady-state output signal. T_{90} can be approximated as:

$$T_{90} = 2.2T. \quad (3.25)$$

Example. 3.1: Impulse and ramp responses of a first-order system

Derive the impulse and ramp responses of the system

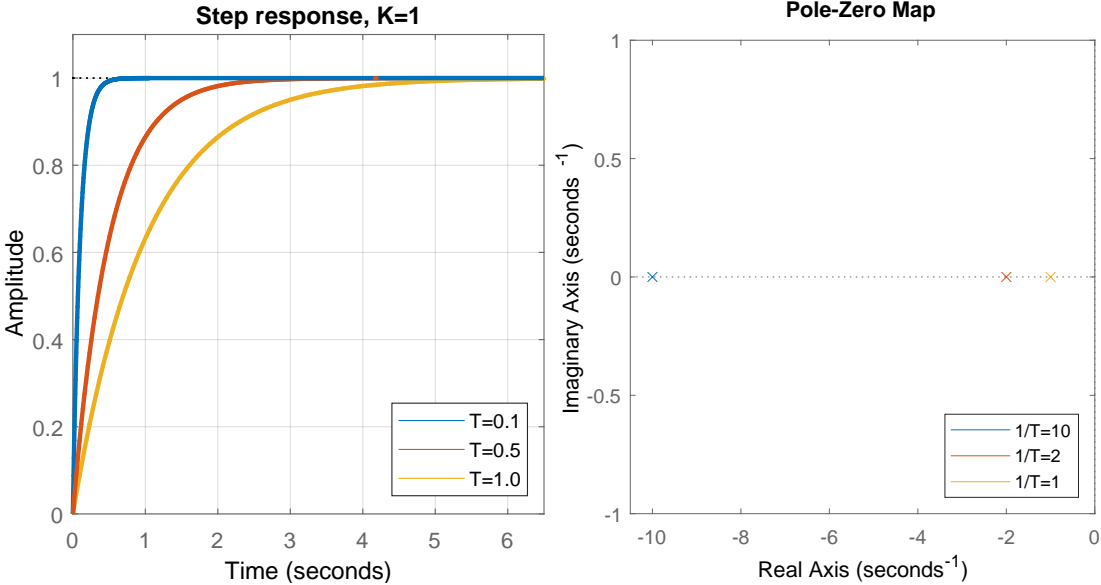
$$G(s) = \frac{K}{Ts + 1}. \quad (3.26)$$

and compare them with the step response of this system.

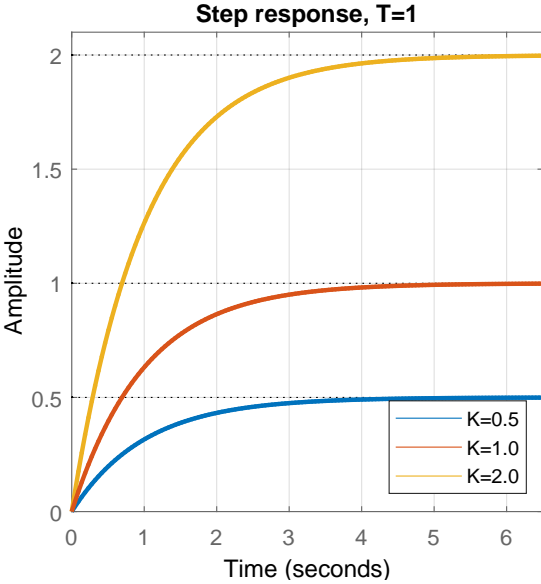
The impulse response $g(t)$ is given as

$$g(t) = \frac{K}{T} e^{-\frac{t}{T}}, \quad y(t) = g(t), \quad (3.27)$$

and can be obtained as the time derivative of expression (3.21).



(a) Response with varying time constant T . (b) Pole/zero map with varying time constant T .



(c) Response with varying gain K .

Figure 3.1: Step responses and pole/zero maps of a first-order system.

To derive the ramp response, we consider the input signal to the process $u(t) = t$ and its Laplace transform $U(s) = 1/s^2$. This gives

$$Y(s) = G(s)U(s) = \frac{K}{s^2(Ts + 1)} = \frac{K}{s^2} - \frac{K}{s} + \frac{KT^2}{Ts + 1}, \quad (3.28)$$

$$y(t) = K \left(t - 1 + Te^{-\frac{t}{T}} \right). \quad (3.29)$$

We note that the time derivative of the ramp response is the process step response.

Example. 3.2: Rectangular pulse response of a first-order system

Consider rectangular pulse at the input of the form

$$u(t) = \begin{cases} 0, & t < 0, \\ c, & 0 \leq t < t_1, \\ 0, & t \geq t_1. \end{cases} \quad (3.30)$$

Derive the response of a first-order system with gain K and time constant T .

The pulse is composed of two step changes: the first one with a magnitude c at time $t = 0$ and the second one with a magnitude $-c$ at time $t = t_1$. Therefore, its Laplace transform is given as

$$U(s) = \frac{c}{s} - \frac{c}{s}e^{-t_1s} = \frac{c}{s} \left(1 - e^{-t_1s} \right). \quad (3.31)$$

The Laplace transform of the response can also be expressed as the sum of two step responses

$$Y(s) = \frac{cK}{s(Ts + 1)} - \frac{cK}{s(Ts + 1)}e^{-t_1s}. \quad (3.32)$$

The inverse Laplace transform gives the desired expression for $y(t)$

$$y(t) = \begin{cases} cK \left(1 - e^{-t/T} \right), & t < t_1, \\ cK \left(1 - e^{-t/T} \right) - cK \left(1 - e^{-(t+t_1)/T} \right), & t \geq t_1. \end{cases} \quad (3.33)$$

3.1.2 Effects of an Additional Pole on the Step Response

Now, let us examine a combination of a first-order system with gain K and time constant $T = 1$ s in series with an additional first-order system with the unit gain and the time constant $1/p$ (or equivalently a stable pole $-p$). The combined transfer function is given as

$$G(s) = \frac{1}{(s + 1) \left(\frac{1}{p}s + 1 \right)}. \quad (3.34)$$

We can choose, without loss of generality, $K = 1$ as the gain just stretches the step response vertically without any other effects. We can expect that the step response will be for large values

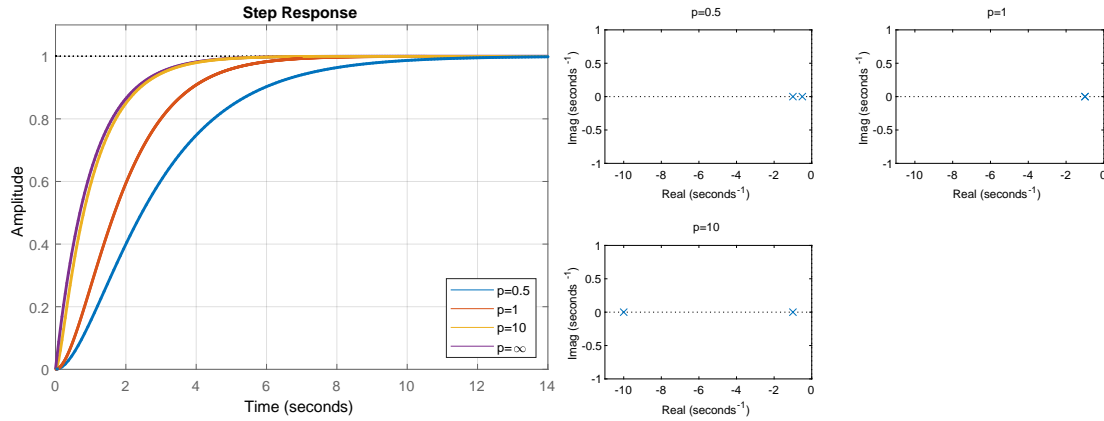


Figure 3.2: Step responses and pole/zero maps of a second-order system with a varying position of the pole p .

of p close to the step response of the first-order system. In that case, it should settle down in a new steady state within 3–5 seconds.

The actual step response will differ depending on whether the value of p is equal to or distinct from the value of the other pole (corresponding to $T = 1$).

$$Y(s) = G(s)U(s) = \frac{1}{s(s+1)\left(\frac{1}{p}s+1\right)} = \begin{cases} \frac{1}{s} - \frac{1}{s+1} - \frac{1}{(s+1)^2}, & \text{if } p = 1, \\ \frac{1}{s} - \frac{p}{p-1} \frac{1}{s+1} + \frac{1}{p(p-1)} \frac{1}{\frac{1}{p}s+1}, & \text{if } p \neq 1. \end{cases} \quad (3.35)$$

$$y(t) = \begin{cases} 1 - e^{-t} - te^{-t}, & \text{if } p = 1, \\ 1 - \frac{p}{p-1}e^{-t} - \frac{1}{p-1}e^{-pt}, & \text{if } p \neq 1. \end{cases} \quad (3.36)$$

Figure 3.2 shows the step responses and pole locations for systems with various values of p . If $p \rightarrow \infty$, we recover the original first-order system. We can see that the step response with $p = 10$ is very close to that of the first-order system. The pole $p \gg 1$ is very fast and its effect vanishes quickly. In this case, the pole $p = 1$ dominates the step response. If the value of p is close to one, the step response moves to the right (the process response is slower) and exhibits a clearly visible inflection point. For very small values of p , the pole at p would dominate the step response and the step response would again resemble a first-order system.

Dominant time constant T_{DOM} If one time constant is at least 10 times higher than other ones, it dominates the dynamic properties of the process. The corresponding pole is then the *dominant* pole (or a couple of dominant poles if they are complex conjugates) and it is the closest pole to the origin.

A practical rule of thumb is that $T_\epsilon \approx (3 \text{ to } 5)T_{\text{DOM}}$.

In general, an additional pole slows down the process response if it is comparable with the other poles. Its effects on the step response will be negligible if it is at least 10 times faster than other poles. On the other hand, it will dominate the dynamic properties if it is at least 10 times slower than the other poles.

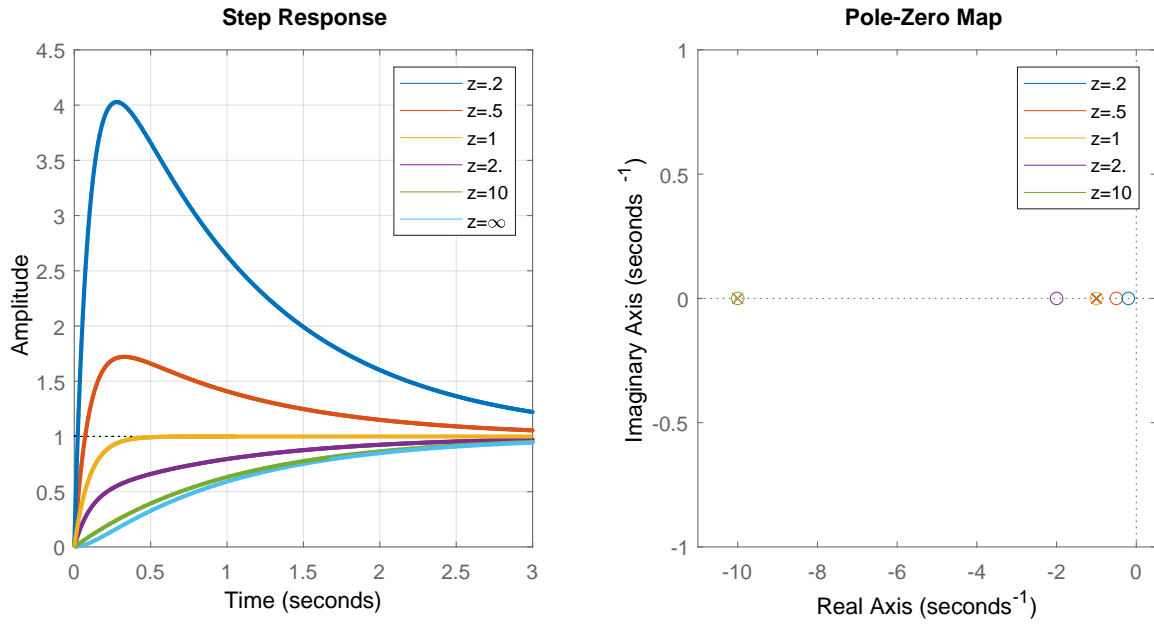


Figure 3.3: Step responses and pole/zero maps of a second-order system with a varying position of a stable real zero z in the complex plane.

3.1.3 Effects of an Additional Zero on the Step Response

Let us now consider adding a zero to the process transfer function and study its influence on the step response. We will distinguish two cases: a stable and an unstable zero.

Let us consider a system with two poles p_1 and p_2 as in the previous section and a zero z . We can derive the step response of the system without zero $y_0(t)$ and the influence of the zero

$$G(s) = \frac{s + z}{(s + p_1)(s + p_2)} = (s/z + 1) \frac{1}{(s/p_1 + 1)(s/p_2 + 1)}, \quad (3.37)$$

$$Y(s) = G(s)U(s) = (s/z + 1) \frac{1}{(s/p_1 + 1)(s/p_2 + 1)} \frac{1}{s}, \quad (3.38)$$

$$Y_0(s) = G_0(s)U(s) = \frac{1}{(s/p_1 + 1)(s/p_2 + 1)} \frac{1}{s}, \quad (3.39)$$

$$y(t) = y_0(t) + \frac{1}{z} \dot{y}_0(t). \quad (3.40)$$

As the derivative $\dot{y}_0(t)$ is positive, the smaller the zero, the faster the step response is.

Stable Zero

Figure 3.3 shows the step responses of a process with poles at $p_1 = 1$ and $p_2 = 10$, hence the pole p_1 is dominant. Without the zero ($z \rightarrow \infty$), the step response resembles the step response of the first-order system with $T = 1$. Actually, the values of zeros $z = 10$ and $z = 1$ cancel one of the poles and leave the first-order system with time constant $T = 1$ and $T = 0.1$, respectively.

In general, the presence of a stable zero reduces the rise time. As far as the settling time is concerned, the step response settles the fastest if the zero cancels out the dominant pole. If the zero is very small (in the sense of its absolute value) and dominant (the closest to the imaginary axis), it results in an overshoot and the settling time increases again.

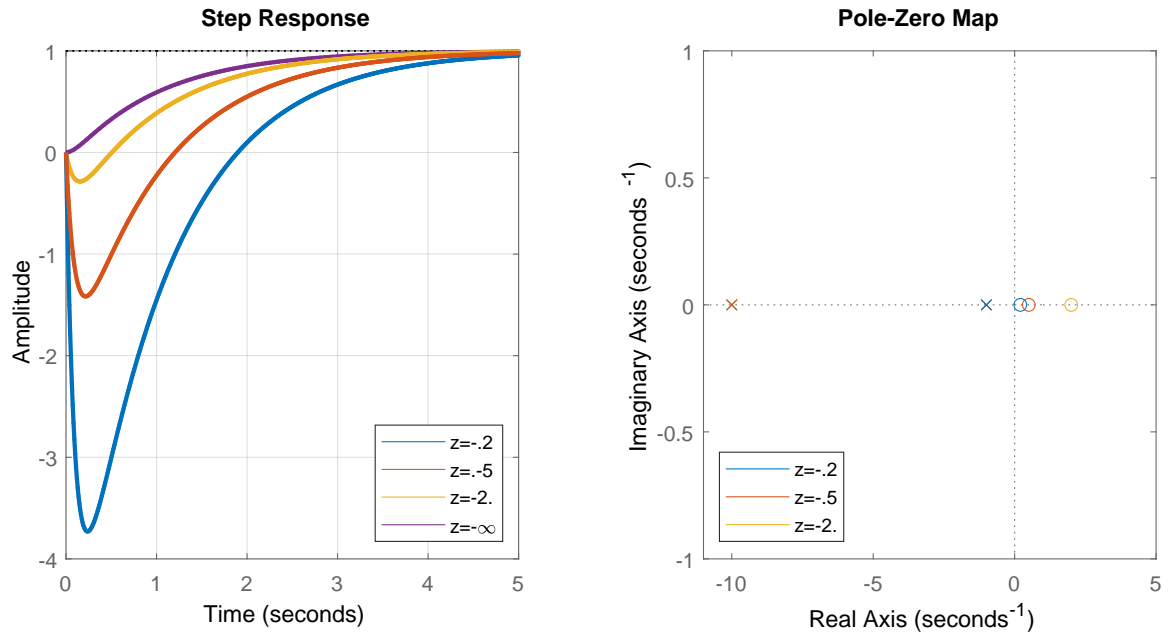


Figure 3.4: Step responses and pole/zero maps of a second-order system with a varying position of an unstable zero z .

Unstable Zero

Negative value of the process zero results in the initial part of the step response taking the opposite direction to the final part. This is shown in Figure 3.4. Again, the largest effect is caused by the dominant (smallest) unstable zero the closest to the imaginary axis.

3.1.4 Step Response of a Second-Order System

Let us now consider systems with a dominant pair of complex conjugate poles. Such systems can be described by the transfer function (here normalised with $K = 1$)

$$G(s) = \frac{1}{T^2 s^2 + 2T\zeta s + 1} = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}, \quad \zeta \in (0, 1), \quad (3.41)$$

where T is the time constant, $\omega_0 = 1/T$ is the undamped natural frequency, and ζ is the damping ratio (relative damping, damping coefficient). If $\zeta \geq 1$ we speak about an overdamped system with two real poles. If $\zeta \in (0, 1)$ the system is underdamped and its poles occur as a complex conjugate pair given as

$$s_{1,2} = -\zeta\omega_0 \pm j\omega_0 P, \quad P = \sqrt{1 - \zeta^2}. \quad (3.42)$$

We can calculate the magnitude (distance from the origin) and phase (angle from the origin) of the complex poles

$$|s_{1,2}| = \sqrt{(-\zeta\omega_0)^2 + (\omega_0 P)^2} = \omega_0, \quad (3.43)$$

$$\phi(s_1) = \tan^{-1} \left(\frac{\omega_0 P}{-\zeta\omega_0} \right) = -\tan^{-1} \frac{P}{\zeta}, \quad \phi(s_2) = \tan^{-1} \frac{P}{\zeta}. \quad (3.44)$$

While the magnitude on ω_0 only, the phase depends on ζ only.

Let us now derive the step response by partial fractioning the Laplace transform of the output $Y(s)$ to

$$Y(s) = G(s)U(s) = \frac{\omega_0^2}{s(s^2 + 2\zeta\omega_0s + \omega_0^2)} = \frac{k_1}{s} + \frac{k_2s + k_3}{s^2 + 2\zeta\omega_0s + \omega_0^2} \quad (3.45)$$

$$= \frac{1}{s} - \frac{s + 2\omega_0}{s^2 + 2\zeta\omega_0s + \omega_0^2} \quad (3.46)$$

$$= \frac{1}{s} - \frac{s + \omega_0}{(s + \zeta\omega_0)^2s + (\omega_0P)^2} - \frac{\zeta}{P} \frac{\omega_0P}{(s + \zeta\omega_0)^2s + (\omega_0P)^2}. \quad (3.47)$$

The inverse Laplace transform then gives the step response

$$y(t) = 1 - \frac{1}{P} e^{-\zeta\omega_0t} (P \cos(\omega_0Pt) + \zeta \sin(\omega_0Pt)) \quad (3.48)$$

$$= 1 - \frac{1}{P} e^{-\zeta\omega_0t} \sin(\omega_0Pt + \varphi), \quad \varphi = \cos^{-1} \zeta. \quad (3.49)$$

Step responses and pole/zero maps for various values of the parameters ζ and ω_0 are shown in Figure 3.5. The response of the system is damped, oscillating with the frequency ω_0P (period of oscillations $2\pi/\omega_0P$). We note that if ω_0 is fixed, the damping ratio ζ stretches the graph vertically and the poles lie on the circular arc with radius ω_0 . If ζ is close to one, the step response resembles the one with two real poles. As ζ approaches zero, oscillations are more pronounced and the system is closer to instability.

If ζ is fixed, the undamped natural frequency ω_0 stretches the graph horizontally. The poles lie on the lines running at an angle depending on ζ only.

The rise time is determined for underdamped systems as T_{100} from the condition $y(t) = y_\infty$. This gives

$$T_{100} = \frac{1}{\omega_0P} \left[\pi - \tan^{-1} \left(\frac{P}{\zeta} \right) \right]. \quad (3.50)$$

The settling time can just be approximated. We start from (3.49) and over/underapproximate the function by

$$y(t) \approx 1 \pm \frac{1}{P} e^{-\zeta\omega_0t}, \quad (3.51)$$

as the values of the sine function lie within $[-1, 1]$. The condition $y(T_\epsilon) = 1 \pm \epsilon$ gives

$$T_\epsilon \approx \frac{1}{\zeta\omega_0} \ln \frac{1}{\epsilon P}, \quad T_{0.02} \approx \frac{4}{\zeta\omega_0}, \quad T_{0.05} \approx \frac{3}{\zeta\omega_0}. \quad (3.52)$$

We can define some additional characteristics of the oscillatory step responses:

Maximum overshoot e_{\max} or σ is usually defined as the ratio between the maximum and steady-state outputs in percents

$$e_{\max} = \sigma = \frac{y_{\max} - y_\infty}{y_\infty - y_0} \times 100\%. \quad (3.53)$$

Peak time T_σ is the time when the maximum overshoot occurs.

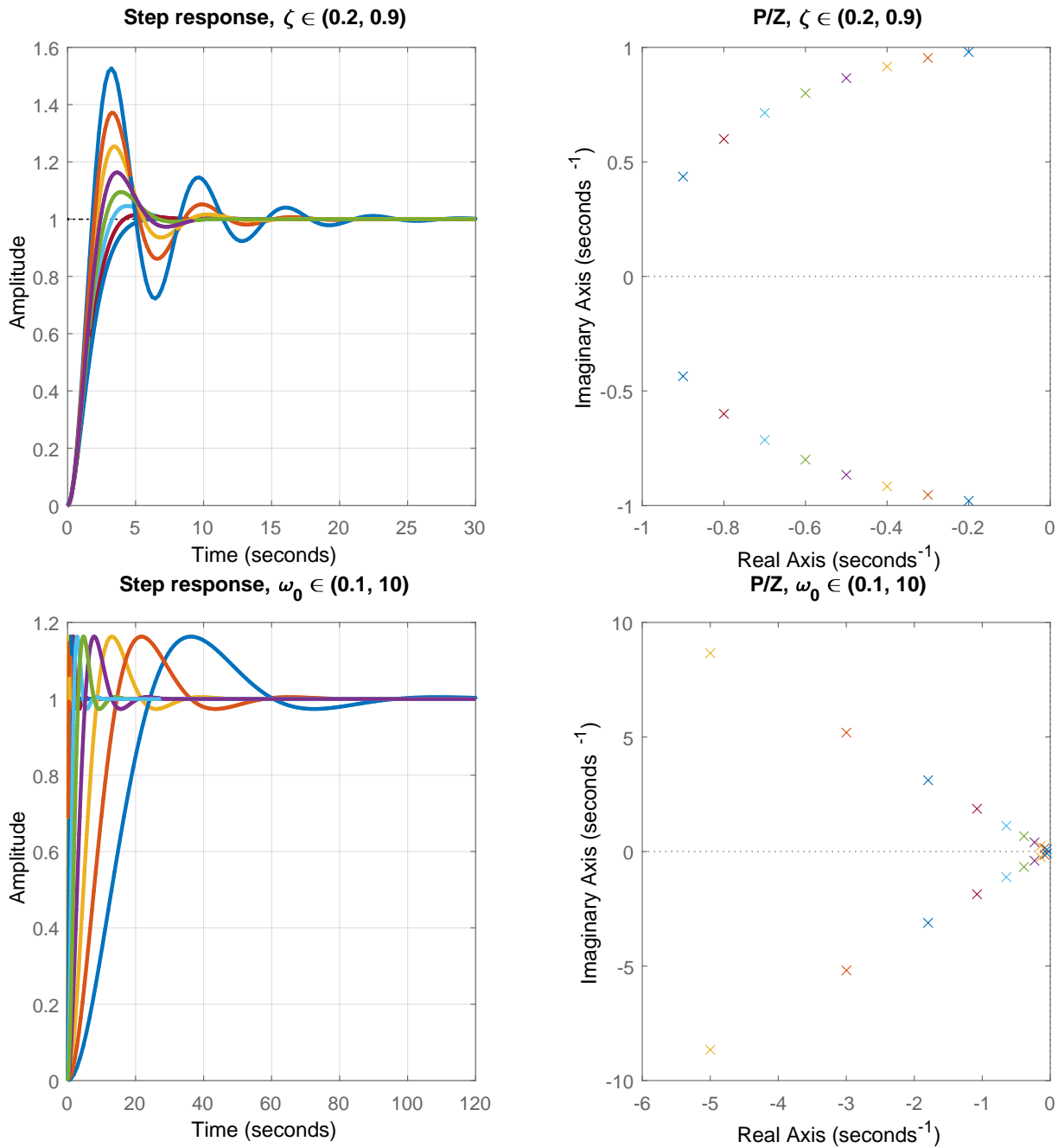


Figure 3.5: Step responses and pole/zero maps of a second-order underdamped system for various values of ζ and ω_0 . Smaller ζ results in a less damped response, smaller ω_0 makes the response faster.

Both the aforementioned characteristics can be derived from the step response setting the derivative of the output $y(t)$ with respect to time equal to zero. The derivative of (3.48) gives

$$\sin(\omega_0 P t) = 0 \quad \Rightarrow \quad T_\sigma = \frac{\pi}{\omega_0 P}. \quad (3.54)$$

If ω_0 is fixed, the peak time reduces with decreasing ζ and converges towards $T_\sigma = \pi/\omega_0 = \pi T$. The value of the dominant time constant can be estimated in a similar way as for the first-order system and its settling time, $T \approx (3 \text{ to } 4)T_\sigma$.

Substituting back the expression for T_σ into (3.48) gives

$$y(T_\sigma) = 1 + e^{-\frac{\pi\zeta}{P}} \quad \Rightarrow \quad e_{\max} = e^{-\frac{\pi\zeta}{P}} \quad \Rightarrow \quad \zeta = \frac{|\ln e_{\max}|}{\sqrt{\pi^2 + \ln^2 e_{\max}}}, \quad (3.55)$$

where we can see that the maximum overshoot is a function of the damping ratio only. The value of ζ can thus be determined from the maximum overshoot.

3.2 Examples

3.2.1 Two Tanks with Interaction

We study the dynamic properties of two tanks with interaction modelled in Example 1.2.2. The process parameters are as follows: $q_{0,1}^s = 0.9 \text{ m}^3/\text{s}$, $q_{0,2}^s = 0 \text{ m}^3/\text{s}$, $F_1 = 0.5 \text{ m}^2$, $F_2 = 0.6 \text{ m}^2$, $k_{11} = 0.8 \text{ m}^{2.5}/\text{s}$, and $k_{22} = 0.5 \text{ m}^{2.5}/\text{s}$.

We have also derived two possible transfer functions of the system (Eqs. (2.75) and (2.76)) given as

$$G_1(s) = \frac{Y(s)}{U_1(s)} = \frac{k_1}{F_1 F_2 s^2 + (k_1 F_1 + k_1 F_2 + k_2 F_1)s + k_1 k_2} = \frac{7.2}{6.075s^2 + 9.326s + 1}, \quad (3.56)$$

$$G_2(s) = \frac{Y(s)}{U_2(s)} = \frac{F_1 s + k_1}{F_1 F_2 s^2 + (k_1 F_1 + k_1 F_2 + k_2 F_1)s + k_1 k_2} = \frac{10.12s + 7.2}{6.075s^2 + 9.326s + 1}. \quad (3.57)$$

3.2.1.1 Rewrite the transfer functions in terms of zeros and poles. Draw the poles and zeros in the complex plane.

Both transfer functions are second order with two stable poles $p_1 = -0.12$, $p_2 = -1.42$ and a gain $K = 7.2$. The second transfer function also has one stable zero $z_1 = -0.71$.

$$G_1(s) = \frac{1.19}{(s + 1.42)(s + 0.12)}, \quad (3.58)$$

$$G_2(s) = 1.67 \frac{s + 0.71}{(s + 1.42)(s + 0.12)}. \quad (3.59)$$

```
g1 = tf(7.2, [6.075 9.326 1]); g2 = tf([10.12 7.2], [6.075 9.326 1]);
zpk(g1), zpk(g2)
pzmap(g1); figure; pzmap(g2)
```

Pole-zero maps of both transfer functions are shown in Figure 3.6.

3.2.1.2 Discuss the step response corresponding to the transfer functions and estimate the settling times. Plot the step responses to confirm your findings.

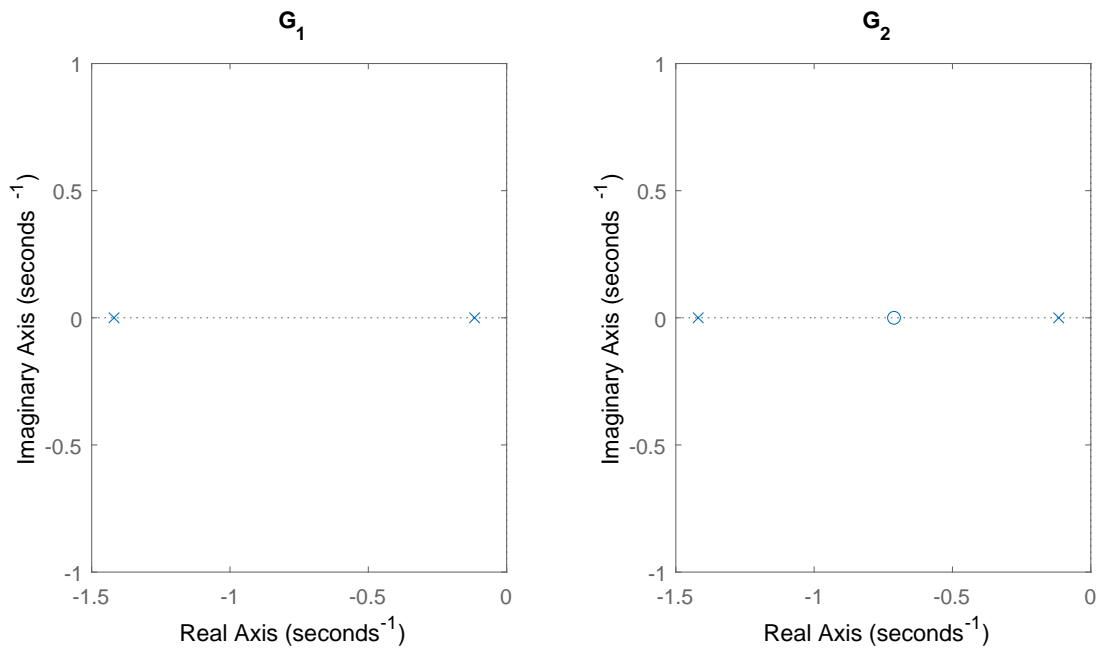


Figure 3.6: Pole-zero map of tanks with interaction.

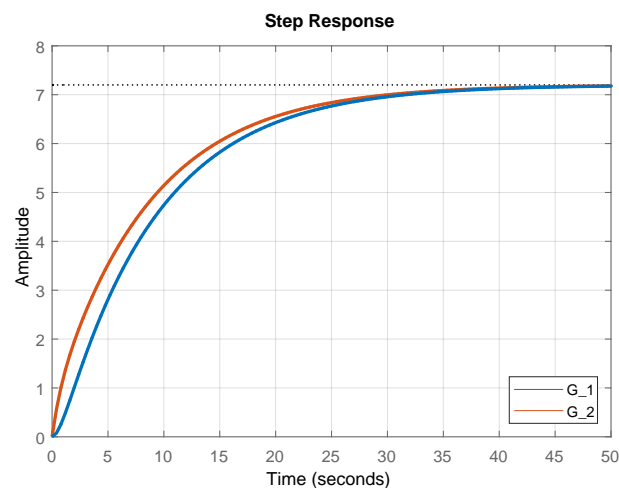


Figure 3.7: Step responses of two tanks with interaction.

Both transfer functions have stable real poles, hence, their step responses will be overdamped (no oscillations). As far as $G_1(s)$ is concerned, pole p_1 is dominant with the corresponding time constant $T_1 = -1/p_1 = 8.6$ s. We can expect that the step response will resemble that of the first-order system with the time constant T_1 and the settling time approximately 30 s.

The step response of $G_2(s)$ will be influenced by the additional zero. This is located between the poles and will speed up the response a bit. However, the pole p_1 remains dominant and will still influence the response.

```
step(g1,g2)
legend('G_1', 'G_2')
```

Both step responses are shown in Figure 3.7.

3.2.1.3 Determine the damping ratio ζ and the natural frequency ω_0 of both transfer functions.

The natural frequency can be obtained if the denominator polynomial is monic (*i.e.* the coefficient at the highest degree of s is equal to one) and written as $s^2 + 2\zeta\omega_0s + \omega_0^2 = 0$

$$6.075s^2 + 9.326s + 1 = 0 \Rightarrow s^2 + 1.54s + 0.17 = 0 \Rightarrow \omega_0^2 = 0.17.$$

The coefficient of s^1 can be used to determine ζ , since it corresponds to $2\zeta\omega_0$

$$\zeta = \frac{1.54}{2\omega_0} = 1.89.$$

The system is overdamped as $\zeta > 1$.

3.2.2 Two Tanks with a Pump

We study the dynamic properties of two tanks with a pump modelled in Example 1.2.3. The process parameters are: $q_{0,1}^s = 0.9 \text{ m}^3/\text{s}$, $q_{0,2}^s = 0 \text{ m}^3/\text{s}$, $F_1 = 0.5 \text{ m}^2$, $F_2 = 0.6 \text{ m}^2$, $k_{11} = 0.8 \text{ m}^{2.5}/\text{s}$, and $k_{22} = 0.5 \text{ m}^{2.5}/\text{s}$.

We have also derived two possible transfer functions between the output and two inputs (2.103)

$$G_1(s) = \frac{k_1}{F_1F_2s^2 + (k_1F_1 + k_1F_2)s}, \quad (3.60)$$

$$G_2(s) = \frac{-F_1s - k_1}{F_1F_2s^2 + (k_1F_1 + k_1F_2)s}. \quad (3.61)$$

3.2.2.1 Derive the step response corresponding to the transfer functions.

Both transfer functions have a pole at the origin and are thus unstable.

$$Y_1(s) = G_1(s) = \frac{k_1}{s^2(F_1F_2s + k_1F_1 + k_1F_2)}, \quad (3.62)$$

$$Y_2(s) = G_2(s) = \frac{-F_1s - k_1}{s^2(F_1F_2s + k_1F_1 + k_1F_2)}. \quad (3.63)$$

Applying the inverse Laplace transform gives the desired expressions for the step response (see Section 3.5):

$$y_1(t) = \frac{t}{F_1 + F_2} - \frac{F_1F_2}{k_1(F_1 + F_2)^2} \left(1 - e^{-\frac{k_1(F_1+F_2)}{F_1F_2}t} \right), \quad (3.64)$$

$$y_2(t) = -\frac{t}{F_1 + F_2} - \frac{F_1^2}{k_1(F_1 + F_2)^2} \left(1 - e^{-\frac{k_1(F_1+F_2)}{F_1F_2}t} \right). \quad (3.65)$$

3.2.2.2 Discuss asymptotic behaviour of the step responses.

The first term in both (3.64) and (3.65) is a ramp function, the second term is a

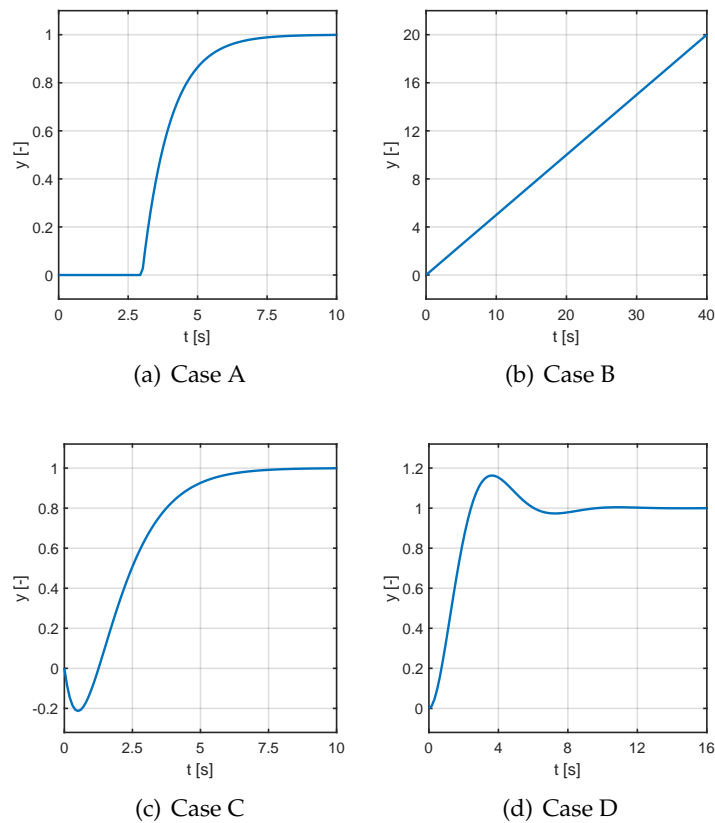


Figure 3.8: Step response of the four dynamic systems.

constant and the third term is a stable exponential with the time constant

$$T = \frac{F_1 F_2}{k_1 (F_1 + F_2)}, \quad (3.66)$$

which goes to zero after $(3 \text{ to } 5)T$. The first term dominates, thus the step response is asymptotically a linear function with the slope $\pm 1/(F_1 + F_2)$.

3.3 Practice Examples

3.3.1 An Introductory Example

Figure 3.8 shows step responses of four different systems after a perturbation by the step function at time $t_0 = 0$ from zero initial conditions.

Match the given transfer functions with the corresponding step response. Justify your answers briefly. Characterise the step response in terms of stability, oscillatory behaviour, time delay, and minimum/nonminimum phase.

$$G_1(s) = \frac{0.5}{s}, \quad G_2(s) = \frac{1}{s^2 + s + 1}, \quad G_3(s) = \frac{1-s}{(s+1)^2}, \quad (3.67)$$

$$G_4(s) = \frac{e^{-3s}}{s+1}, \quad G_5(s) = \frac{1+s}{(s+1)^2}. \quad (3.68)$$

$$A - G_4(s), B - G_1(s), C - G_3(s), D - G_2(s).$$

3.3.2 Two Tanks in Series

We study the nonlinear process with two tanks in series, for which we have derived the transfer function matrix in Example 2.3.2.

The derived transfer function between the input $u_1(t)$ (the deviation variable of the inflow $q_{0,1}(t)$ from its steady state) and the output $y(t)$ (the deviation variable of the liquid level in the second tank $h_2(t)$ from its steady state) is of the form:

$$G(s) = \frac{\frac{k_1}{F_1 F_2}}{\left(s + \frac{k_1}{F_1}\right) \left(s + \frac{k_2}{F_2}\right)} = \frac{k_1}{(F_1 s + k_1)(F_2 s + k_2)} \quad (3.69)$$

$$= \frac{\frac{1}{k_2}}{\left(\frac{F_1}{k_1} s + 1\right) \left(\frac{F_2}{k_2} s + 1\right)} = \frac{\frac{1}{k_2}}{\frac{F_1 F_2}{k_1 k_2} s^2 + \left(\frac{F_1}{k_1} + \frac{F_2}{k_2}\right) s + 1}. \quad (3.70)$$

3.3.2.1 Transform the transfer function into the form:

$$G(s) = \frac{K\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}, \quad (3.71)$$

and define ω_0 , ζ , and K in terms of the process parameters. Discuss the meaning of the process parameters to the system dynamics. How can you change the steady state (gain) of the system?

$$G(s) = \frac{\frac{k_1}{F_1 F_2}}{\left(s + \frac{k_1}{F_1}\right) \left(s + \frac{k_2}{F_2}\right)} = \frac{\frac{k_1}{F_1 F_2}}{s^2 + \left(\frac{k_1}{F_1} + \frac{k_2}{F_2}\right) s + \frac{k_1 k_2}{F_1 F_2}} = \frac{K\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}. \quad (3.72)$$

From this, it follows:

$$\omega_0^2 = \frac{k_1 k_2}{F_1 F_2}, \quad (3.73)$$

$$2\zeta\omega_0 = \frac{k_1}{F_1} + \frac{k_2}{F_2} \Rightarrow 2\zeta\sqrt{\frac{k_1 k_2}{F_1 F_2}} = \frac{k_1}{F_1} + \frac{k_2}{F_2} \Rightarrow \zeta = \frac{\frac{k_1}{F_1} + \frac{k_2}{F_2}}{2\sqrt{\frac{k_1 k_2}{F_1 F_2}}}, \quad (3.74)$$

$$K\omega_0^2 = \frac{k_1}{F_1 F_2} \Rightarrow K\frac{k_1 k_2}{F_1 F_2} = \frac{k_1}{F_1 F_2} \Rightarrow K = \frac{1}{k_2}. \quad (3.75)$$

The steady-state gain of the system can be changed solely by the valve constant at the outlet of the second tank. The overall system dynamics is given by the cross-sectional areas of the tanks and by the values of valve constants. The damping ratio ζ is always greater than 1 (notice here the relationship between the arithmetic and geometric mean of two numbers).

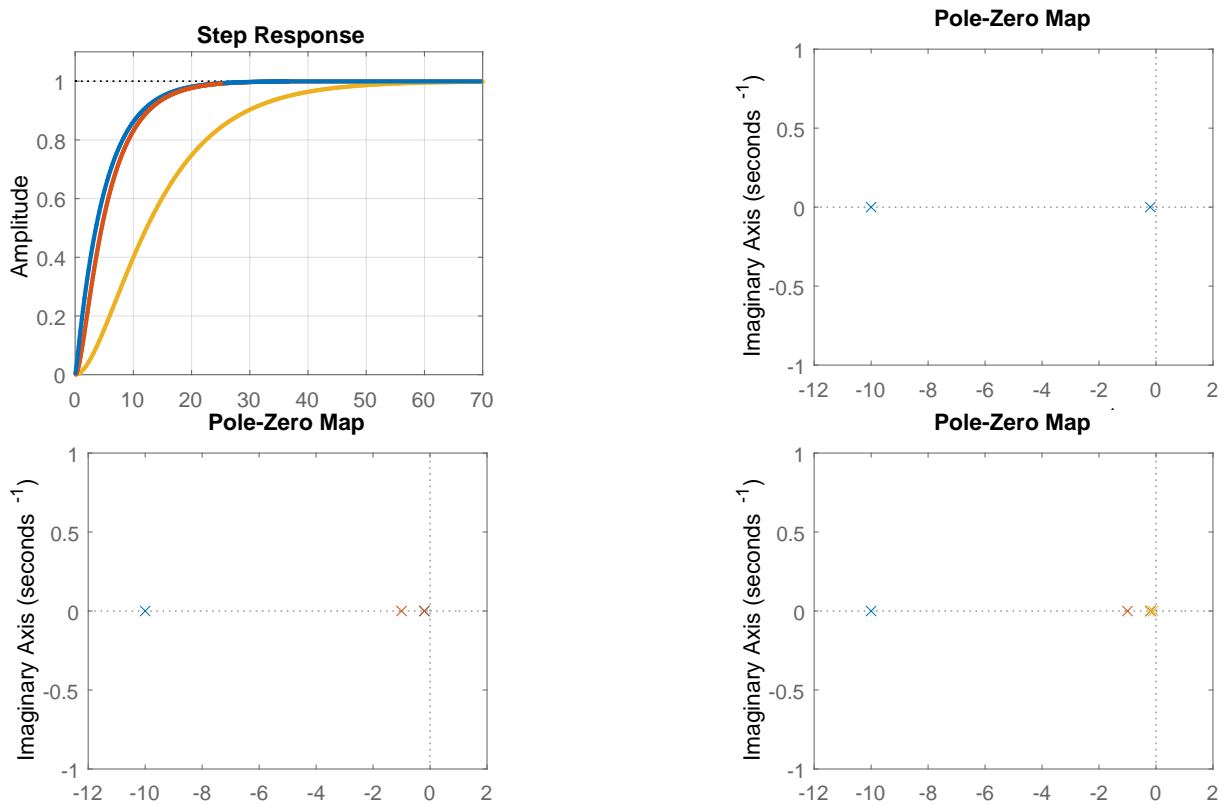


Figure 3.9: Two tanks in series: step responses and cumulative pole-zero maps for various values of F_2 .

3.3.2.2 Let us study a system where the valve constants are $k_1 = k_2 = 1 \text{ m}^2/\text{s}$ and the cross-sectional area of the first tank is $F_1 = 5 \text{ m}^2$. Compare the step responses of three systems, where the cross-sectional area of the second tank is $F_2 = 0.1 \text{ m}^2$, $F_2 = 1 \text{ m}^2$, and $F_2 = 10 \text{ m}^2$, respectively. Discuss the results. Would the results change if the cross-sectional area of the second tank was $F_2 = 5 \text{ m}^2$ and $F_1 = 0.1 \text{ m}^2$, $F_1 = 1 \text{ m}^2$, and $F_1 = 10 \text{ m}^2$, respectively? What physical change in the system would you recommend to make so that the overall response of the system becomes faster but without oscillations and without a change in the steady-state gain?

The results would not change if we exchanged values of F_1 and F_2 . This would only change the behaviour of particular tanks but not the overall system behaviour as the poles would remain unchanged by the transformation.

If we want to make the response of the system faster, we need to change the slowest (dominant) pole. This can be done by decreasing the cross-sectional area of the tank with the slowest pole or by increasing the valve constant (opening the valve more) of the tank with the slowest pole, but only in the case if this tank is the first in the series.

Figure 3.9 shows the step responses and pole-zero maps for $F_2 \in \{0.1, 1, 10\} \text{ m}^2$.

3.3.2.3 (Optional) Show the recommendations from the previous point by numerical simulation. Try to make the system response twice as fast.

If $F_2 = 1 \text{ m}^2$ then we can set $F_1 = 2.5 \text{ m}^2$ or $k_1 = 2 \text{ m}^2/\text{s}$.

3.3.2.4 Consider the steady-state process with parameters and initial values as follows: $q_{0,1}^s = 0.3 \text{ m}^3/\text{s}$, $q_{0,2}^s = 0.5 \text{ m}^3/\text{s}$, $F_1 = 0.5 \text{ m}^2$, $F_2 = 0.8 \text{ m}^2$, $k_{11} = 1.15 \text{ m}^{2.5}/\text{s}$, and $k_{22} = 1.3 \text{ m}^{2.5}/\text{s}$.

Calculate the maximal liquid height in the second tank if the process is subject to following trajectory of $q_{0,1}$

$$q_{0,1}(t) = \begin{cases} 0.3 \text{ m}^3/\text{s}, & t < 5 \text{ s}, \\ 0.33 \text{ m}^3/\text{s}, & t \in [5, 7) \text{ s}, \\ 0.2 \text{ m}^3/\text{s}, & t \geq 7 \text{ s}. \end{cases} \quad (3.76)$$

Confirm your findings in a simulation using the nonlinear model. (Hint: calculate the final response of the linearised model as a weighted sum of two step responses at $t = 5 \text{ s}$ and $t = 7 \text{ s}$, respectively, using the inverse Laplace transform).

The transfer function between h_2 and $q_{0,1}$ is given by (2.127) and we can express the output $y(t)$ using the inverse Laplace transform considering $u(t) = 1$

$$G_{yu_1}(s) = \frac{\frac{k_1}{F_1 F_2}}{\left(s + \frac{k_1}{F_1}\right)\left(s + \frac{k_2}{F_2}\right)} = \frac{5.51}{s^2 + 5.729s + 5.82}, \quad (3.77)$$

$$Y(s) = G_{yu_1}(s)U(s) = \frac{5.51}{s^3 + 5.729s^2 + 5.82s}, \quad (3.78)$$

$$y(t) = \begin{cases} 0.4e^{-4.4t} - 1.35e^{-1.32t} + 0.95, & t \geq 0, \\ 0, & t < 0. \end{cases} \quad (3.79)$$

As the system is linear, its response to two step changes is given as a weighted sum of the individual responses. In our case, this gives

$$h_2(t) = h_2^s + 0.03y(t - 5) - 0.13y(t - 7). \quad (3.80)$$

The maximum height will be achieved at time $t = 7$ and is given as $h_2(7) = 0.4 \text{ m}$. The nonlinear process will reach the maximum height at the same time and its value will differ from the linear approximation by a few percent.

3.3.3 Mass on a Spring

Consider a system of a unit mass hanging from a spring as described in Problem 1.3.

3.3.3.1 Write down mathematical model of the system such that one can infer the position.

From the force balance:

$$F_g = F_m + F_s + F_r, \quad (3.81)$$

where F_g is the force due to gravity. Using the expressions for forces, we obtain:

$$g = k_s(x(t) - x_0) + \ddot{x}(t) + k_r\dot{x}(t), \quad (3.82)$$

which can be rewritten as:

$$\ddot{x}(t) + k_r \dot{x}(t) + k_s x(t) = k_s x_0 + g. \quad (3.83)$$

3.3.3.2 Rewrite the model such that the initial conditions when set to zero would represent the steady state.

The steady state can be found for conditions $\ddot{x}(t) = \dot{x}(t) = 0$ as:

$$k_s x^s = k_s x_0 + g \quad \Rightarrow \quad x^s = x_0 + \frac{g}{k_s}. \quad (3.84)$$

Then, we can write for the deviation variable $y(t)$ and its derivatives:

$$y(t) = x(t) - x^s, \quad \dot{y}(t) = \dot{x}(t), \quad \ddot{y}(t) = \ddot{x}(t). \quad (3.85)$$

Finally

$$\ddot{y}(t) + k_r \dot{y}(t) + k_s y(t) = 0, \quad \dot{y}(0) = y(0) = 0. \quad (3.86)$$

3.3.3.3 Transform the model into a transfer function of the form:

$$G(s) = \frac{K\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}, \quad (3.87)$$

and define ω_0 , ζ , and K in terms of the process parameters.

$$G(s) = \frac{0}{s^2 + k_r s + k_s} = \frac{K\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \quad (3.88)$$

From this, it follows that

$$\omega_0^2 = k_s, \quad (3.89)$$

$$2\zeta\omega_0 = k_r \quad \Rightarrow \quad 2\zeta\sqrt{k_s} = k_r \quad \Rightarrow \quad \zeta = \frac{k_r}{2\sqrt{k_s}}, \quad (3.90)$$

$$K\omega_0^2 = 0 \quad \Rightarrow \quad K = 0. \quad (3.91)$$

3.3.3.4 Simulate the response of the system where (a) $k_s = 10$ and $k_r = 1$, (b) $k_s = 100$ and $k_r = \sqrt{10}$, (c) $k_s = 10$ and $k_r = 0$ and (d) $k_s = 10$ and $k_r = 10$. Comment on the system response regarding the values of ω_0 and ζ .

We will first convert the model to a state-space form with states $x_1 = y$, $x_2 = \dot{y}$

$$\dot{x}_1 = x_2, \quad (3.92)$$

$$\dot{x}_2 = -k_s x_1 - k_r x_2. \quad (3.93)$$

```
% Case A
ks = 10; kr = 1;
initial(ss([0 1; -ks -kr], [], [1 0], 0), [1 0], 30)

% Case B
```

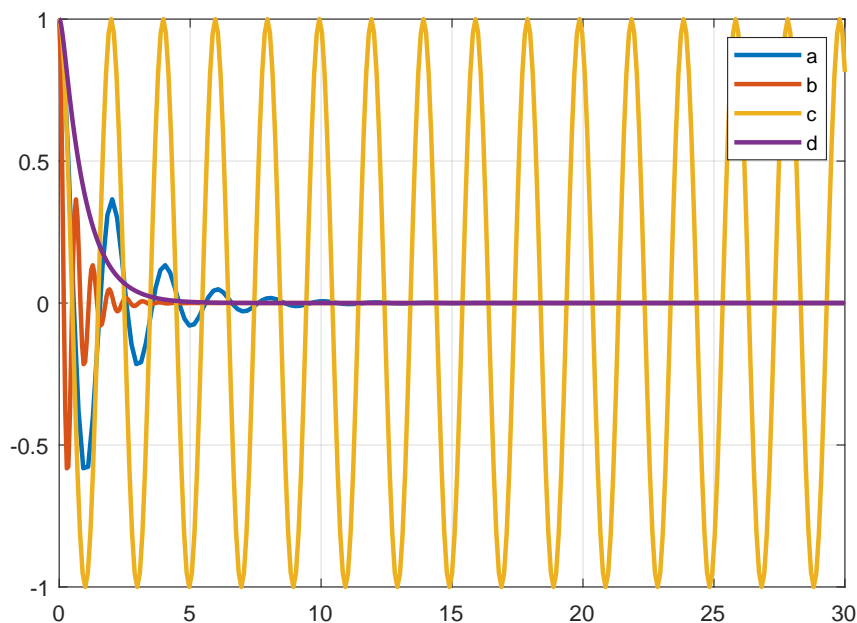


Figure 3.10: Time response of the position of the mass-spring process for different values of the parameters.

```

ks = 100; kr = 1*sqrt(10);
initial(ss([0 1; -ks -kr], [], [1 0], 0), [1 0], 30)

% Case C
ks = 10; kr = 0;
initial(ss([0 1; -ks -kr], [], [1 0], 0), [1 0], 30)

% Case D
ks = 10; kr = 10;
initial(ss([0 1; -ks -kr], [], [1 0], 0), [1 0], 30)

```

Time response of the output variable is shown in Figure 3.10. The Cases A and C exhibit the same frequency of oscillations with $\omega_0 = 3.33$, the case A oscillates three times faster with $\omega_0 = 10$. Damping ratio is the same in the cases A and B ($\zeta = 0.16$) and ten times higher in the case D. Therefore, the cases A and B exhibit the same decay rate and the case D is overdamped. Finally, $\zeta = 0$ characterises a marginally stable system with sustained oscillations.

3.4 Problems

Problem. 3.1: Reluctance area of the first-order system

Consider a step response of the first-order system with the gain K and time constant T . Calculate analytically area A between this curve and line $r(t) = K$. Confirm your results using numerical integration in MATLAB.

$A = KT$. This information can also be used to determine the value of the time constant T experimentally.

```
T = 3; K = 4;
g = tf(K, [T 1]);
[y,t]=step(g);
area = trapz(t, K-y) % = 12.0006
T1 = area/K % = 3.0002
```

Problem. 3.2: Additional pole/zero with a transfer function involving complex conjugate poles

Examine the step responses of the systems of the form

$$G_z(s) = \frac{\omega_0^2(s+z)}{s^2 + 2\zeta\omega_0s + \omega_0^2}, \quad (3.94)$$

$$G_p(s) = \frac{\omega_0^2}{(s+p)(s^2 + 2\zeta\omega_0s + \omega_0^2)}, \quad (3.95)$$

where $\zeta \in (0, 1)$ and z and p are a stable zero and pole, respectively. What happens if these become dominant? Examine the maximum overshoot, rise, and settling times.

An additional zero makes the step response faster and an additional pole makes the step response slower. A dominant pole/zero will reduce the influence of the complex poles.

Problem. 3.3: Lead-lag system

A lead-lag system is characterised by one pole and one zero

$$G(s) = \frac{T_1s + 1}{T_2s + 1}. \quad (3.96)$$

Plot the step responses and consider both cases: $T_1 > T_2$ and $T_1 < T_2$.

```
T1 = 1; T2 = 2;
G1 = tf([T1 1], [T2 1]); G2 = tf([T2 1], [T1 1]);
step(G1, G2)
```

As the system is proper (the same degree of the numerator and denominator), the step response exhibits a discontinuity. If a lead ($T_1 < T_2$) is used in series with another system, it increases the speed of the overall response. As the static gain of the lead-lag system is equal to one, this system only modifies the transient response.

Problem. 3.4: Time-delay system

A system with a time delay is characterised by the transfer function

$$G(s) = G_0(s)e^{-T_D s}. \quad (3.97)$$

Plot the step responses of the first-order system with gain $K = 1$, time constant $T = 1$, and time delays $T_D \in \{0.1, 1, 5\}$. Approximate the time delay by a fraction of two polynomials (Padé approximation) in s .

```
T = 1; K = 1;
G0 = tf(K, [T 1]);
G1 = tf(K, [T 1], 'InputDelay', 0.1);
G2 = tf(K, [T 1], 'InputDelay', 1);
G3 = tf(K, [T 1], 'InputDelay', 5);
step(G0, G1, G2, G3)
```

Time delay shifts the step response to the right. If it is smaller than the time constant, the step response remains dominated by the time constant. In that case, the first- or the second-order Padé approximation can be used

$$e^{-T_D s} \approx \frac{1 - \frac{T_D}{2} s}{1 + \frac{T_D}{2} s} \quad (3.98)$$

$$\approx \frac{1 - \frac{T_D}{2} s + \frac{T_D^2}{12} s^2}{1 + \frac{T_D}{2} s + \frac{T_D^2}{12} s^2} \quad (3.99)$$

```
T = 1; K = 1; TD=0.1;
G0 = tf(K, [T 1]); G1 = tf(K, [T 1], 'InputDelay', 0.1);
G1d = tf([-TD/2 1], [TD/2 1]); G1a = G0 * G1d;
step(G1, G1a)
```

Problem. 3.5: Oscillatory step response

Estimate the static gain, damping ratio, and undamped natural frequency from a step response that exhibits an overshoot $y_{\max} = 3.2$ at $t = 2$ s and settles down at $y_{\infty} = 2$.

$K = 2, \zeta = 0.16, \omega_0 = 1.59$.

```
K = 2; zeta = 0.16; om0 = 1.59;
g = tf(K*om0^2, [1 2*zeta*om0 om0^2]);
step(g)
```

Problem. 3.6: Step and impulse responses

Compare the step response of the first-order system with gain K and time constant T with the impulse response of an unstable process with a transfer function of the form

$$G(s) = \frac{K}{s(Ts + 1)}. \quad (3.100)$$

The responses are the same.

Problem. 3.7: Ramp and step responses

Compare the ramp response of the first-order system with gain K and time constant T with a step response of an unstable process with a transfer function of the form

$$G(s) = \frac{K}{s(Ts + 1)}. \quad (3.101)$$

The responses are the same.

3.5 MATLAB Snippets

- Inverse Laplace transform using symbolic operations to derive the step response (3.64) and (3.65)

```
syms k1 F1 F2 s
g1 = k1/(F1*F2*s^2 + k1*(F1+F2)*s); y1 = g1/s;
yt1 = simplify(ilaplace(y1))
g2 = -(F1*s+k1)/(F1*F2*s^2 + k1*(F1+F2)*s); y2 = g2/s;
yt2 = simplify(ilaplace(y2))
```

- Polynomial and pole-zero representations of transfer functions

```
num = [1 3 2]; den=[1 2 3 1];
g1 = tf(num, den), g2 = zpk(g1)
```

- Step response can be calculated and/or plotted using `step` or `stepplot` (Figure 3.1).

```
K = 1; T = [0.1 0.5 1.0];
G1 = tf(K, [T(1) 1]); G2 = tf(K, [T(2) 1]); G3 = tf(K, [T(3) 1]);
h = stepplot(G1, G2, G3);
setoptions(h, 'XLim', [0, 6.5], 'YLim', [0, 1.1], 'grid', 'on')
legend('T=0.1', 'T=0.5', 'T=1.0', 'Location', 'southeast');
title('Step response, K=1');
```

Similarly for impulse response: `impulse` or `impzplot`

- There is no built-in command for the ramp response as this can be obtained from the step response of a system with the transfer function $G(s)/s$
- Transfer function from K , ω_0 , and ζ

```
K = 2; zeta = 0.16; om0 = 1.59;
g = tf(K*om0^2, [1 2*zeta*om0 om0^2]);
[num, den] = ord2(om0, zeta);
g2 = tf(K*om0^2*num, den)
```

- Pole-zero map (see Figure 3.1)

```
pzmap(G1, G2, G3); axis([-10.5 0 -1 1]);
legend('1/T=10', '1/T=2', '1/T=1', 'Location', 'southeast');
```

This plot can be customised with `pzplot`.

- Static gain of a transfer function $K = G(0)$

```
K = dcgain(g);
```

3.6 Python Snippets

- Inverse Laplace transform using symbolic operations to derive step responses (3.64) and (3.65)

```
import sympy as sym
s = sym.symbols('s')
F1, F2, k1, t = sym.symbols('F1, F2, k1, t', positive=True)
a = F1 * F2; b = k1 * (F1 + F2)
g1 = k1 / (a * s**2 + b * s)
y1 = g1 / s
```

```
sym.inverse_laplace_transform(y1, s, t)
g2 = -(F1 * s + k1) / (a * s**2 + b * s)
y2 = g2 / s;
sym.inverse_laplace_transform(y2, s, t)
```

- Polynomial and pole-zero representations of transfer functions

```
from control.matlab import *
num = [1, 3, 2]; den = [1, 2, 3, 1]
g1 = tf(num, den)
z, p, k = tf2zpk(num, den)
g2 = zpk2tf(z, p, k)
```

- Step response can be calculated and/or plotted using `step` (Figure 3.1).

```
from control.matlab import *
import matplotlib.pyplot as plt
K = 1; T = [0.1, 0.5, 1.0];
G1 = tf(K, [T[0], 1]); G2 = tf(K, [T[1], 1]); G3 = tf(K, [T[2], 1]);
y1, t1 = step(G1, 7); y2, t2 = step(G2, 7); y3, t3 = step(G3, 7);
plt.figure(1)
plt.plot(t1, y1, t2, y2, t3, y3); plt.legend(['T=0.1', 'T=0.5', 'T=1.0'])
plt.grid(); plt.xlabel('Time (seconds)'); plt.ylabel('Amplitude')
plt.title('Step response, K=1')
plt.show()
```

Similarly for impulse responses `impulse`. There also exist functions for a response with nonzero initial conditions (`initial`) and a simulation with an arbitrary input (`lsim`)

- Poles, zeros, and pole-zero map (see Figure 3.1)

```
from control.matlab import *
import matplotlib.pyplot as plt
num = [1, 3, 2]; den = [1, 2, 3, 1]; g1 = tf(num, den);
z = zero(g1); p = pole(g1)
p, z = pzmap(g1, plot=True, grid=True)
plt.show()
```

- Static gain of a transfer function $K = G(0)$

```
K = dcgain(g1)
```

3.7 Bibliography

Understanding of dynamic properties of processes represents a crucial knowledge for process control engineering. The theoretical background for this chapter can be found in Mikleš and Fikar (2007, Chapter 4), information in Slovak in Bakošová and Fikar (2008, Chapter 3).

Additional information can be found for example in Seborg, Edgar, Mellichamp, and Doyle (2016, Chapters 5-6), or Stephanopoulos (1984, Chapter 3).

CHAPTER 4

Frequency-Domain Characteristics of Dynamic Systems

Expected outcomes of this chapter are:

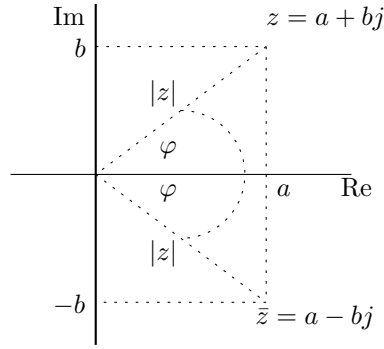
- refresh of knowledge on complex numbers, their representations, and mathematical operations involving them,
- understand the properties of dynamic systems in frequency domain, frequency responses of dynamic systems and their graphical representations: Bode and Nyquist diagrams,
- ability to sketch Bode diagrams for basic transfer function components of linear systems (consisting of gain, real and complex poles, integrators, derivators, delays),
- understanding the relationships between poles, zeros, and frequency-domain properties.

In the previous chapter, we studied the characteristics of dynamic systems in the time domain and the responses to step, impulse, or ramp signals. While it is possible to determine time-domain properties for the first and the second order systems, it is difficult to do so for higher-order systems. Moreover, controller design methods that set time-domain characteristics are mostly iterative and their use was possible only due to the power of digital computers.

Another commonly used input signal is a sine wave as a function of frequency ω . We will study the asymptotic time response of processes to this (harmonic) signal and relate them to frequency-domain transfer functions where the variable s is substituted for the complex variable $j\omega$. We will see that it is possible to predict time-domain properties based on their frequency-domain counterparts.

While two different dynamic systems can exhibit similar time-domain characteristics (*e.g.* step responses, rise time, settling time), their differences will clearly show in frequency domain. Moreover, powerful methods exist for open- or closed-loop analysis, controller design using frequency-domain tools.

Frequency-domain analysis does not mean that the process input is sinusoidal. Rather, we will be able to study the controlled process, controller, or closed-loop system from a different point of view.

Figure 4.1: Complex numbers z and \bar{z} .

4.1 Overview

4.1.1 Complex Numbers, Variables, and Functions

A complex number z and its conjugate form \bar{z} can be represented in algebraic, trigonometric, and polar forms as

$$z = a + bj = |z|(\cos \varphi + j \sin \varphi) = |z|e^{j\varphi}, \quad (4.1)$$

$$\bar{z} = a - bj = |z|(\cos \varphi - j \sin \varphi) = |z|e^{-j\varphi}, \quad (4.2)$$

where j is defined by $j = \sqrt{-1}$, thus, $j^2 = -1$. Real-valued parameters a and b of the algebraic form are real and imaginary parts that can be thought of as coordinates of a point in a two-dimensional space (Figure 4.1). The second equality represents the trigonometric and the last equality the polar form expressed by the magnitude (amplitude) and phase (angle). Clearly, the magnitude or absolute value (distance from the origin) $|z|$ and phase or angle φ (to horizontal, real axis) can be determined as

$$|z| = \sqrt{a^2 + b^2}, \quad \cos \varphi = \frac{a}{|z|}, \quad \sin \varphi = \frac{b}{|z|}, \quad \tan \varphi = \frac{b}{a}. \quad (4.3)$$

Care has to be taken when determining the phase using the last formula to detect the correct quadrant (as with $a = -1, b = -1$).

Useful formulae for magnitude and phase of the composite complex numbers arising from multiplication and division are (consider polar form $z_i = |z_i|e^{j\varphi_i}, i = 1, 2, 3$)

$$z = \frac{z_1 z_2}{z_3} = \frac{|z_1|e^{j\varphi_1}|z_2|e^{j\varphi_2}}{|z_3|e^{j\varphi_3}} = \frac{|z_1||z_2|}{|z_3|}e^{j(\varphi_1+\varphi_2-\varphi_3)} = |z|e^{-j\varphi}, \quad (4.4)$$

$$|z| = \frac{|z_1||z_2|}{|z_3|}, \quad \varphi = \varphi_1 + \varphi_2 - \varphi_3. \quad (4.5)$$

Let us now consider a complex variable $s = \sigma + j\omega$ and a function $G(s)$. It follows that $G(s)$ is also a complex-valued function that can be represented by its real/imaginary parts, or by the magnitude and phase

$$G(s) = \mathcal{R}(G(s)) + j\mathcal{I}(G(s)) = |G(s)|e^{j\varphi(G(s))}. \quad (4.6)$$

Example. 4.1: Complex variable

Characterise the real and imaginary parts, magnitude, and phase of the complex function

$$G(s) = \frac{s+2}{s(s+1)}, \quad (4.7)$$

for $s = 3j$.

The real and imaginary parts can be obtained by substituting for $s = 3j$ and by eliminating j from the denominator

$$G(3j) = \frac{3j+2}{3j(3j+1)} = \frac{3j+2}{-9+3j} \frac{(-9-3j)}{(-9-3j)} = \frac{-9-33j}{81+9} = -\frac{9}{90} - \frac{33}{90}j = -\frac{3}{30} - \frac{11}{30}j. \quad (4.8)$$

The absolute value can either be calculated from the real and imaginary parts or as the absolute value of the numerator divided by the absolute value of the denominator

$$|G(3j)| = \sqrt{\left(\frac{3}{30}\right)^2 + \left(\frac{11}{30}\right)^2}, \quad |G(3j)| = \frac{|3j+2|}{|3j||3j+1|} = \sqrt{\frac{13}{90}}. \quad (4.9)$$

The phase (or angle) can either be calculated from the real and imaginary parts or as the sum of phases in the numerator minus the sum of phases in the denominator

$$\tan \varphi(G(3j)) = \frac{11}{3}, \quad \varphi(G(3j)) = \varphi(3j+2) - \varphi(3j) - \varphi(3j+1). \quad (4.10)$$

Example. 4.2: Complex function

Characterise the real and imaginary parts, magnitude, and phase of the complex function

$$G(s) = \frac{K}{Ts+1}, \quad (4.11)$$

for the real positive parameters K and T and for $s = j\omega$, where $\omega \in \langle 0, \infty \rangle$.

The real and imaginary parts can be obtained by substituting $s = j\omega$ and removing j from the denominator

$$G(j\omega) = \frac{K}{Tj\omega+1} = \frac{K}{1+T\omega j} \frac{1-T\omega j}{1-T\omega j} = \frac{K(1-T\omega j)}{1+T^2\omega^2} = \frac{K}{1+T^2\omega^2} - \frac{KT\omega}{1+T^2\omega^2}j. \quad (4.12)$$

The absolute value can either be calculated from the real and imaginary parts or as the absolute value of the numerator divided by the absolute value of the denominator

$$|G(j\omega)| = \frac{K}{\sqrt{1+T^2\omega^2}}. \quad (4.13)$$

The phase can either be calculated from the real and imaginary parts or as the difference of the phase in the numerator (zero) and the phase in the denominator

$$\tan \varphi(G(j\omega)) = -T\omega. \quad (4.14)$$

4.1.2 Frequency Responses and Characteristics

As we have seen in the example above, a transfer function can be represented as a function of a complex variable j and frequency ω . We can characterise its magnitude and phase as functions of frequency.

Another physical observation can be derived when introducing harmonic signal of the form $u(t) = a \sin(\omega t)$ to a process described by a transfer function $G(s)$. The process response settles down to a harmonic signal of the same frequency ω but with a different magnitude after some transient period and it is delayed by a phase shift corresponding to the magnitude and phase of the transfer function at that particular frequency:

$$y(t) = b \sin(\omega t + \varphi), \quad (4.15)$$

where

$$b = a|G(j\omega)|, \quad (4.16)$$

$$\varphi = \varphi(G(j\omega)). \quad (4.17)$$

The most popular representations of transfer function frequency-domain properties are the frequency characteristics – plotted for values of frequency either $\omega \in (0, \infty)$ or $\omega \in (-\infty, \infty)$

- Magnitude: $|G(j\omega)| = f(\log_{10} \omega)$ or $20 \log_{10} |G(j\omega)| = f(\log_{10} \omega)$,
- Phase: $\arg(G(j\omega)) = \varphi(\omega) = f(\log_{10} \omega)$,
- Magnitude/Phase: $G(j\omega) = f(\omega)$.

The first two plots are usually shown together and called Bode diagram. The third one is called the Nyquist diagram.

Bode diagrams use semilog ω -axis. Magnitude is plotted in decibels (dB, $20 \log_{10} |G|$), which is again in a logarithmic scale. The main advantage of this representation lies in the fact that the absolute values of individual components of the transfer function (containing poles, zeros, gain) can be summed in both magnitude and phase.

Consider for example a system with the transfer function given as

$$G(s) = \frac{B(s)}{A(s)} = K \frac{s^k (1 + \frac{s}{z_1}) \cdots (1 + \frac{s}{z_m})}{s^l (1 + \frac{s}{p_1}) \cdots (1 + \frac{s}{p_n})}. \quad (4.18)$$

Its logarithmic magnitude and phase can then be written as a sum of individual components:

$$20 \log |G(j\omega)| = 20 \log |K| + 20(k-l) \log \omega + \sum_{i=1}^m 20 \log \left| 1 + j \frac{\omega}{z_i} \right| - \sum_{i=1}^n 20 \log \left| 1 + j \frac{\omega}{p_i} \right|, \quad (4.19)$$

$$\varphi(\omega) = 90(k-l) + \sum_{i=1}^m \varphi \left(1 + j \frac{\omega}{z_i} \right) - \sum_{i=1}^n \varphi \left(1 + j \frac{\omega}{p_i} \right). \quad (4.20)$$

Moreover, the individual contributions of simple poles and zeros to the magnitude can conveniently be approximated by asymptotes whereby any asymptote can only have a slope that is an integer multiple of 20 dB per decade. This is shown in Figure 4.2, and 4.3 (complex poles are represented by the polynomial $s^2 + 2\zeta s + 1$ where $\zeta \in (0, 1)$).

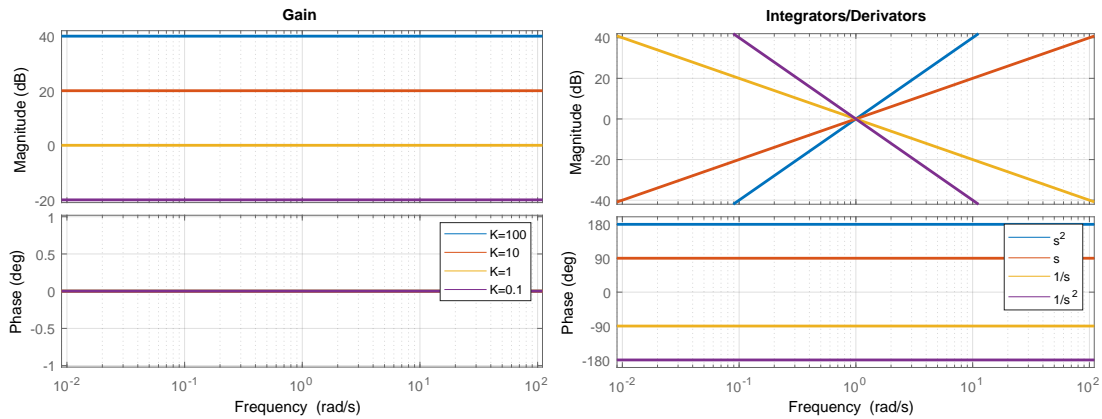


Figure 4.2: (left) Bode diagrams of component transfer functions: gain, (right) integrator/derivative.

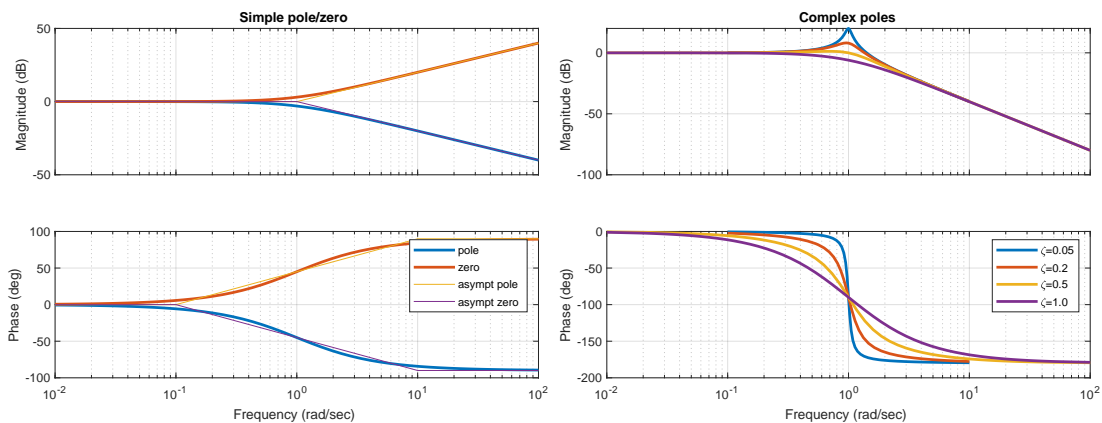


Figure 4.3: (left) Bode diagrams of component transfer functions: simple pole/zero, (right) two complex poles.

4.1.3 Stability Margins

When considering a transfer function of the controlled process and the controller connected in series, we speak of open-loop transfer function and denote the respective system transfer function as $G_o(s)$.

- The gain crossover point in the frequency-domain plot of $G_o(j\omega)$ is the point at which $|G_o(j\omega)| = 1$ or $20 \log |G_o(j\omega)| = 0$ dB. The frequency at the gain crossover point is called the gain crossover frequency ω_g .
- The phase crossover point in the frequency-domain plot of $G_o(j\omega)$ is the point at which $\varphi(G_o) = -180^\circ$. The frequency at the phase crossover point is called the phase crossover frequency ω_p .
- The gain margin g_m is the amount of gain K in decibels (dB) that has to be added to the open-loop system at frequency ω_p before the closed-loop system becomes unstable ($|G_o(\omega_p)| = 1$), that is,

$$|G_o(\omega_p)|K = 1, \quad g_m = 20 \log K = -20 \log (|G_o(\omega_p)|). \quad (4.21)$$

- The phase margin φ_p is defined as the angle in degrees by which the Nyquist diagram of the open-loop system has to be rotated about the origin so that the gain crossover passes

the $-1 + 0j$ point and the closed-loop system becomes unstable, that is,

$$\varphi(G_o(\omega_g)) + \varphi_p = -180, \quad \varphi_p = -180 - \varphi(G_o(\omega_g)). \quad (4.22)$$

Gain and phase margins along with the corresponding frequencies are shown in the Bode diagram in Figure 4.4. The same information in the Nyquist diagram is shown in Figure 4.5.

4.2 Examples

Example. 4.3: Bode diagrams of the basic dynamic components

Derive and examine the magnitude and phase frequency characteristics of a gain, a chain of integrators, a simple pole/zero, and complex conjugate poles that are in Figures 4.2, 4.3.

- Gain $G(s) = K$: $20 \log |G(j\omega)| = 20 \log |K|$ and $\varphi = 0^\circ$ (for positive K). This corresponds to a horizontal line in the magnitude plot which is positive if $K > 1$. Negative values in decibels represent the value of $1/K$ as

$$20 \log K = -20 \log \frac{1}{K}. \quad (4.23)$$

Gain does not change the phase for positive values. If negative, then the phase shift is -180° .

- Multiple integrators $G(s) = 1/s^n$:

$$20 \log |G(j\omega)| = 20 \log \frac{1}{|j\omega|^n} = 20 \log \frac{1}{\omega^n} = -20n \log \omega, \quad (4.24)$$

$$\varphi = \varphi(1) - \varphi((j\omega)^n) = 0^\circ - n90^\circ = -n90^\circ. \quad (4.25)$$

The magnitude plot is a straight line that crosses the point ($\omega = 10^0$, $20 \log |G| = 0$ dB) with a slope of $-20n$. Hence, if the frequency increases 10 times, the magnitude in decibels decreases $20n$. We speak of a decrease of $20n$ decibels per decade.

Each integrator exhibits a constant phase shift of -90° over the entire frequency range.

Note that multiple derivators $G(s) = s^n$ just change the sign of both frequency characteristics. Magnitude plots increase $20n$ dB per decade while the phase shift is constant and positive.

- Simple zero $G(s) = s/z + 1$:

$$20 \log |G(j\omega)| = 20 \log \left| \frac{j\omega}{z} + 1 \right| = \begin{cases} 0, & \text{if } \omega \ll z, \\ 20 \log \omega - 20 \log z, & \text{if } \omega \gg z, \end{cases} \quad (4.26)$$

$$\varphi = \varphi(1) - \varphi \left(\frac{j\omega}{z} + 1 \right) = \begin{cases} 0^\circ, & \text{if } \omega \ll z, \\ -90^\circ, & \text{if } \omega \gg z. \end{cases} \quad (4.27)$$

The magnitude frequency characteristics can be approximated by asymptotes and divided into two regions. For small frequencies, it is constant and stays at 0 dB while

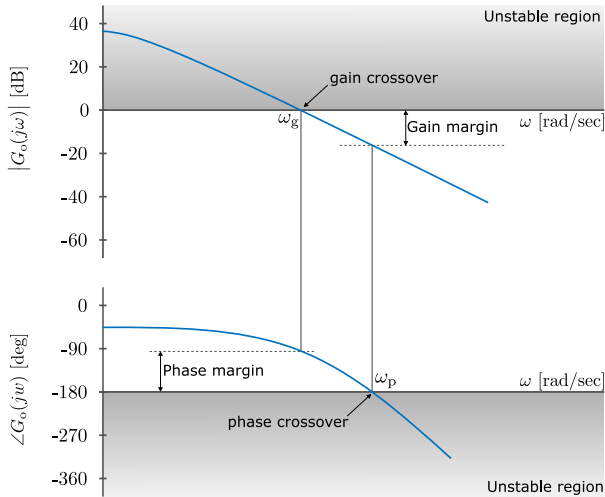


Figure 4.4: Relative stability in Bode diagram

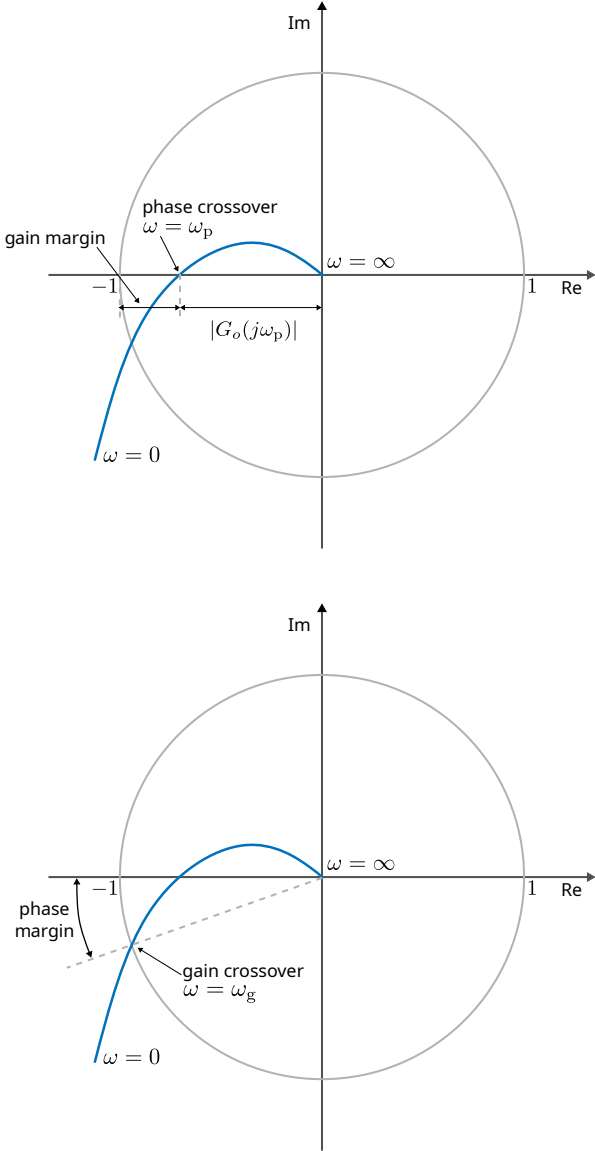


Figure 4.5: Relative stability in Nyquist diagram

for large frequencies it is a straight line that increases at a rate of 20 dB per decade. These two asymptotes meet at the corner frequency $\omega_c = z$, where asymptotes differ from the real magnitude value by about 3 dB.

Phase frequency characteristics can also be approximated by asymptotes. It is zero for small frequencies and 90° for large frequencies. The middle section can be approximated by a straight line starting from the zero asymptote one decade before the corner frequency and ending one decade after the corner frequency.

Note that a simple pole $G(s) = 1/(s/p + 1)$ just changes the sign of both frequency characteristics. The magnitude graph decreases 20 dB per decade for large frequencies defined by the corner frequency $\omega_c = p$ and the phase graph converges to -90° .

- Complex conjugate poles with a damping ratio $\zeta \in (0, 1)$ and a time constant T (or natural frequency $\omega_0 = 1/T$)

$$G(s) = \frac{1}{T^2 s^2 + 2\zeta T s + 1} = \frac{1}{\left(\frac{s}{\omega_0}\right)^2 + 2\zeta \frac{s}{\omega_0} + 1}, \quad (4.28)$$

$$20 \log |G(j\omega)| = -20 \log \sqrt{\left(1 - \frac{\omega^2}{\omega_0^2}\right)^2 + \left(2\zeta \frac{\omega}{\omega_0}\right)^2} \quad (4.29)$$

$$= \begin{cases} 0, & \text{if } \omega \ll \omega_0, \\ -40 \log \omega - 40 \log \omega_0, & \text{if } \omega \gg \omega_0, \end{cases} \quad (4.30)$$

$$\varphi = \tan^{-1} \left(\frac{2\zeta \frac{\omega}{\omega_0}}{1 - \frac{\omega^2}{\omega_0^2}} \right) = \begin{cases} 0^\circ, & \text{if } \omega \ll \omega_0, \\ -90^\circ, & \text{if } \omega = \omega_0, \\ -180^\circ, & \text{if } \omega \gg \omega_0. \end{cases} \quad (4.31)$$

The presence of two poles indicates that the magnitude frequency characteristics can be approximated by an initial asymptote at 0 dB at smaller frequencies and by an asymptote declining by 40 dB per decade at higher frequencies. Again, these two asymptotes intersect at the corner frequency $\omega_c = \omega_0$.

Complex conjugate poles can give rise to a local maximum at the resonant frequency ω_r (called a resonance peak) near the corner frequency. This can be obtained by taking the derivative of the magnitude with respect to ω and setting it to zero. This gives:

$$\omega_r = \omega_0 \sqrt{1 - 2\zeta^2}, \quad (4.32)$$

$$|G(j\omega_r)| = \frac{1}{2\zeta \sqrt{1 - \zeta^2}}. \quad (4.33)$$

The maximum of the magnitude plot occurs only for smaller values of the damping ratio $\zeta < 1/\sqrt{2} \approx 0.7$ and its value depends on ζ only.

The resonant frequency is always smaller than the corner frequency but converges towards it if the damping ratio decreases.

Phase plot can be approximated by constant lines of 0° for small frequencies and -180° for large frequencies. It is equal to -90° at the corner frequency $\omega_c = \omega_0$.

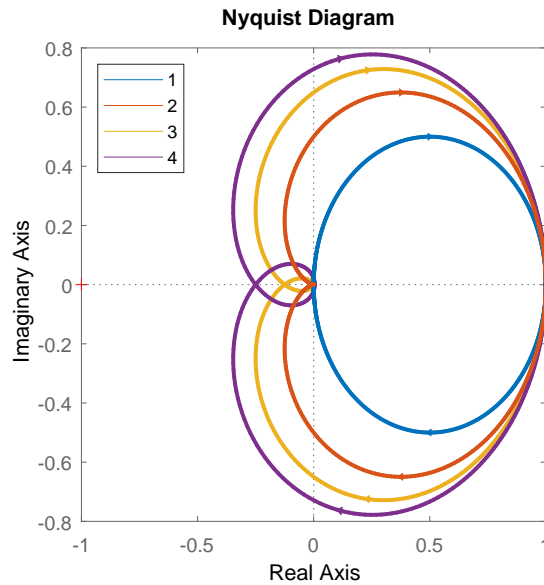


Figure 4.6: Nyquist diagrams of the first-, second-, third-, and fourth-order systems.

Example. 4.4: Nyquist diagrams of multiple systems

Examine the Nyquist diagrams of the following four dynamic systems

$$G_n(s) = \frac{1}{(s+1)^n}, \quad n = 1, 2, 3, 4. \quad (4.34)$$

We derive the magnitude and phase frequency characteristics for the transfer functions $G_n(j\omega)$

$$|G_n(j\omega)| = \frac{1}{|1+j\omega|^n} = \frac{1}{(\sqrt{1+\omega^2})^n}, \quad (4.35)$$

$$\varphi_n = -n \tan^{-1} \omega. \quad (4.36)$$

The main difference between the systems lies in the phase characteristics. While $G_1(j\omega)$ decreases from the initial 0° to -90° , the increasing order of the system adds an additional integer multiples of -90° decrease. The resulting Nyquist diagrams are shown in Figure 4.6. Note that MATLAB shows these plots both for positive and negative frequencies – the plot is symmetric along the real axis. We usually describe only the part with positive frequencies.

While $G_1(j\omega)$ lies entirely in the fourth quadrant (and its mirror image along the real axis in the first quadrant), $G_2(j\omega)$ terminates in the third quadrant, $G_3(j\omega)$ in the second quadrant, and $G_4(j\omega)$ in the first quadrant. Note also the point $-1+0j$ marked with a red cross. It represents the gain equal to one and the phase of -180° . If the frequency response passes this point, the system is marginally stable. We can see that with increasing order the system moves closer to this point indicating possible stability issues.

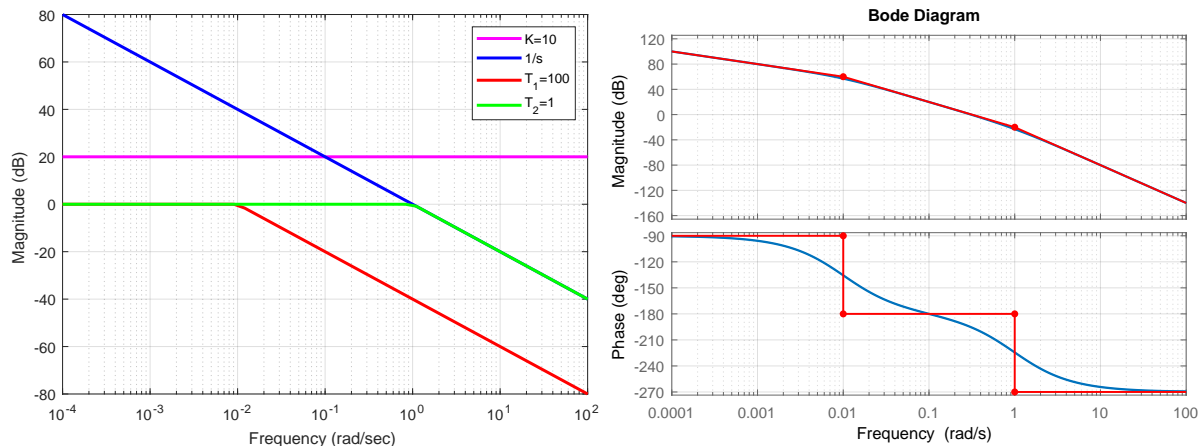


Figure 4.7: Bode diagrams of the third-order system. (left) Magnitude – asymptotes of the individual components, (right) Bode diagram with asymptotes.

Example. 4.5: Unstable third-order system

Examine the Bode diagram of the following system consisting of a gain, integrator, and two poles

$$G(s) = \frac{K}{s(T_1s + 1)(T_2s + 1)}. \quad (4.37)$$

The magnitude and the phase can be obtained as follows:

$$20 \log |G(\omega)| = 20 \log |K| - 20 \log \omega - 20 \log \sqrt{(T_1\omega)^2 + 1} - 20 \log \sqrt{(T_2\omega)^2 + 1}, \quad (4.38)$$

$$\varphi(\omega) = -90^\circ - \tan^{-1}(T_1\omega) - \tan^{-1}(T_2\omega). \quad (4.39)$$

The magnitude-frequency characteristics will be a sum of the characteristics of the four individual systems. The magnitude in decibels of each pole is equal to zero at low frequencies. At the corner frequency $\omega_c = 1/T_i$, an asymptote start decreasing by 20 dB per decade. The integrator decreases by 20 dB per decade over the entire ω span and crosses the point $(\omega = 10^0, 20 \log |G| = 0 \text{ dB})$. Finally, the gain K shifts the magnitude plot up or down depending on the value of $20 \log |K|$.

Similarly, phases of the individual components are: the gain K has a phase of 0° and the integrator has a phase of -90° . Phase of each pole starts at 0° , and bends down around its corner frequency and ends at -90° .

Let us assume the values $K = 10$, $T_1 = 100$, $T_2 = 1$. Magnitudes of the individual components are shown in the left part of Figure 4.7. When summed up at each frequency, they give the final magnitude plot shown in the right part of the figure.

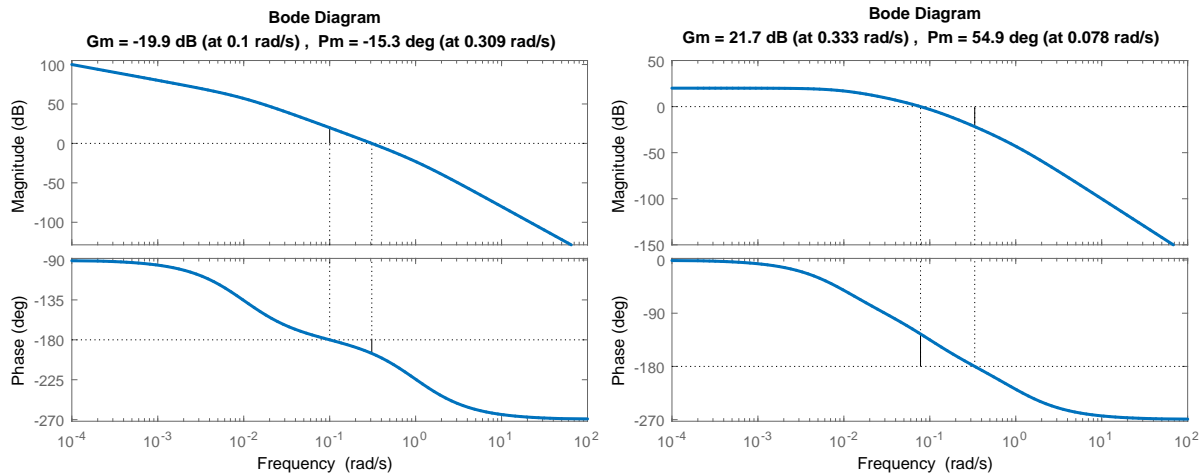


Figure 4.8: Stability margins of 3rd order systems. (left) unstable, (right) stable.

Example. 4.6: Stability margins of a stable and unstable third-order system

Let us study the stability margins of the following transfer functions:

$$G_1(s) = \frac{10}{s(s+1)(100s+1)}, \quad G_2(s) = \frac{10}{(s+1)(10s+1)(100s+1)}. \quad (4.40)$$

We will use the command `margin` that adds stability margins to the Bode diagram.

```
g1 = tf(10, [100 101 1 0])
margin(g1);
g2 = tf(10, [1000 1110 111 1]);
margin(g2);
```

The result is shown in Figure 4.8. Two horizontal lines are added to the plots: for the magnitude plot, one line with the value 0 dB that corresponds to the gain equal to one and in the phase plot, one line with the value corresponding to -180° . Subsequently, two vertical lines are plotted: the one where the magnitude plot crosses the horizontal line 0 dB that extends to the phase plot and marks the complement to -180° . If the phase plot crosses this line from above, we observe a positive phase margin. Otherwise, the closed-loop system is unstable due to the phase.

The second vertical line rises from the intersection of the phase plot with the line corresponding to -180° up to the magnitude plot. Then we can read the difference in decibels. If it is negative, the closed-loop system is unstable due to the gain.

4.2.1 Two Tanks with Interaction

We study the dynamic properties of two tanks with interaction modelled in Example 1.2.2. The process parameters are as follows: $q_{0,1}^s = 0.9 \text{ m}^3/\text{s}$, $q_{0,2}^s = 0 \text{ m}^3/\text{s}$, $F_1 = 0.5 \text{ m}^2$, $F_2 = 0.6 \text{ m}^2$, $k_{11} = 0.8 \text{ m}^{2.5}/\text{s}$, and $k_{22} = 0.5 \text{ m}^{2.5}/\text{s}$.

We have also derived two possible transfer functions between the output and the two inputs

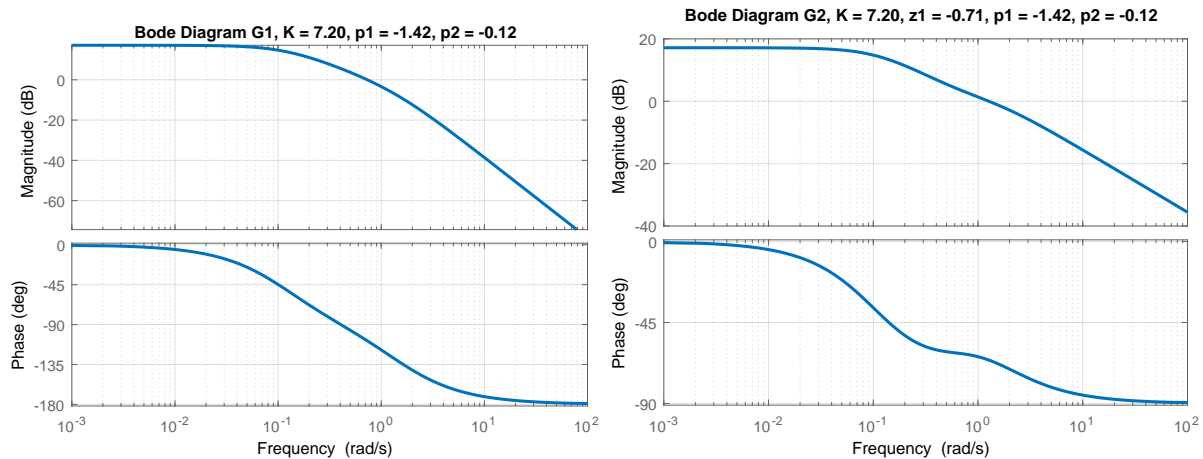


Figure 4.9: Bode diagrams for two tanks with interaction. (left) $G_1(s)$, (right) $G_2(s)$.

(equations (2.75), (2.76)) given as

$$G_1(s) = \frac{Y(s)}{U_1(s)} = \frac{k_1}{F_1 F_2 s^2 + (k_1 F_1 + k_1 F_2 + k_2 F_1)s + k_1 k_2} = \frac{7.2}{6.075s^2 + 9.326s + 1}, \quad (4.41)$$

$$G_2(s) = \frac{Y(s)}{U_2(s)} = \frac{F_1 s + k_1}{F_1 F_2 s^2 + (k_1 F_1 + k_1 F_2 + k_2 F_1)s + k_1 k_2} = \frac{10.12s + 7.2}{6.075s^2 + 9.326s + 1}. \quad (4.42)$$

4.2.1.1 Compare the Bode diagrams for both transfer functions and discuss the differences.

Both transfer functions are second order, with two stable poles $p_1 = -0.12$ and $p_2 = -1.42$ and a gain $K = 7.2$. In addition, the second transfer function involves one stable zero $z_1 = -0.72$.

The resulting Bode diagrams are shown in Figure 4.9. At smaller frequencies, the magnitude plots have a constant value $20 \log K = 17.15$ until the first pole p_1 starts to influence the plot and bends the magnitude down towards a decrease of 20 dB per decade. For $G_1(s)$, the magnitude decreases further and near the second pole p_2 starts to follow the final asymptote with a slope of 40 dB per decade.

On the other hand, $G_2(s)$ has a zero z_1 that counteracts the 20 dB decrease back to a constant value. However, the second pole cancels the effect of the zero for larger frequencies and the final asymptote decreases by 20 dB per decade.

A similar situation occurs with the phase plot. Both start at 0° . For $G_1(s)$, the phase plot ends at -180° whereas for $G_2(s)$ at -90° . In both cases, we can see the effect of the relative degree: an excess of the number of poles to the number of zeros.

4.2.1.2 Find the selected points of the frequency response of a nonlinear process.

A frequency-response characteristics can be determined experimentally using various methods. One point of the frequency response can be obtained by feeding a harmonic signal $u(t) = a \sin(\omega t)$ (as deviation from the steady state) and calculating the ratio of the output signal magnitude to the input signal magnitude as well as the phase shift. This can be repeated for a range of frequencies.

It is possible to determine the frequency characteristics from a single experiment

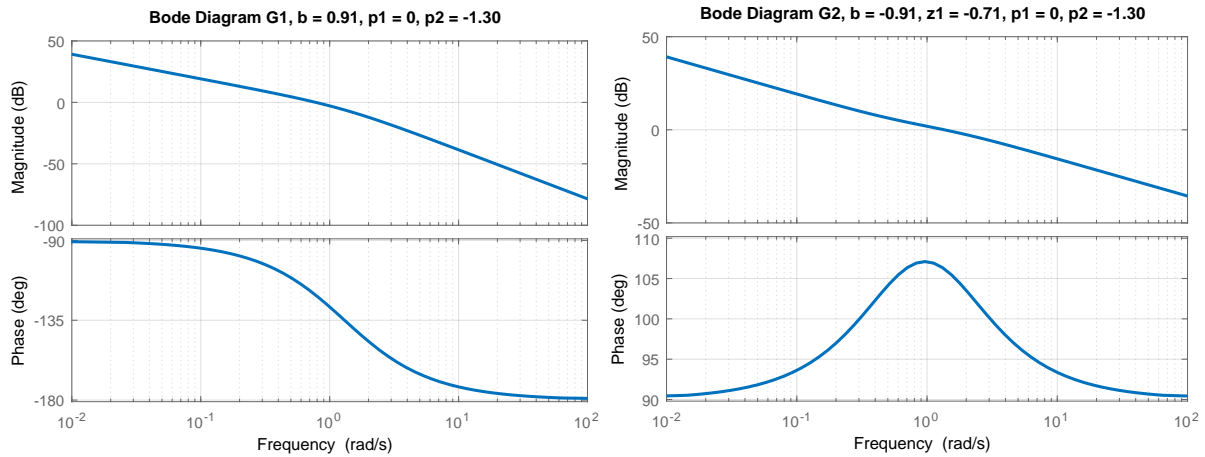


Figure 4.10: Bode diagrams for two tanks with a pump. (left) $G_1(s)$, (right) $G_2(s)$.

using more sophisticated methods based on Fourier analysis. However, it is beyond the scope of this book.

4.2.2 Two Tanks with a Pump

We study the dynamic properties of two tanks with interaction modelled in Example 1.2.2. The process parameters are as follows: $q_{0,1}^s = 0.9 \text{ m}^3/\text{s}$, $q_{0,2}^s = 0.9 \text{ m}^3/\text{s}$, $F_1 = 0.5 \text{ m}^2$, $F_2 = 0.6 \text{ m}^2$, $k_{11} = 0.8 \text{ m}^{2.5}/\text{s}$, and $k_{22} = 0.5 \text{ m}^{2.5}/\text{s}$.

We have also derived two transfer functions between the output and the two inputs (equations (2.103)) given as

$$G_1(s) = \frac{Y(s)}{U_1(s)} = \frac{k_1}{F_1 F_2 s^2 + (k_1 F_1 + k_1 F_2)s} = \frac{0.91}{0.77s^2 + s}, \quad (4.43)$$

$$G_2(s) = \frac{Y(s)}{U_2(s)} = \frac{-F_1 s - k_1}{F_1 F_2 s^2 + (k_1 F_1 + k_1 F_2)s} = \frac{-1.28s - 0.91}{0.77s^2 + s}. \quad (4.44)$$

Compare the Bode diagrams for both transfer functions and discuss the differences.

Both transfer functions are second order, with one unstable pole $p_1 = 0$, one stable pole $p_2 = -1.42$, and a numerator which evaluated at $s = 0$ equals $b = \pm 0.91$. $G_2(s)$ also has one stable zero $z_1 = -0.71$.

The resulting Bode diagrams are shown in Figure 4.10. The integrator ($1/s$) results in a decrease of 20 dB per decade from the start of the magnitude-frequency plot. The additional pole at $p_2 = -1.3$ bends the magnitude further down to 40 dB per decade (left). The occurrence of a zero in $G_2(s)$ effectively suppresses the pole and the magnitude plot on the right-hand side resembles a single integrator.

The phase plot of $G_1(s)$ starts at -90° and ends at -180° . On the other hand, the stable zero of $G_2(s)$ keeps its phase plot practically constant at $-270^\circ = 90^\circ$. Compared to a single integrator which stays at -90° , the negative sign in the numerator adds an additional -180° .

4.3 Practice Examples

4.3.1 First-Order System

Consider a first-order linear dynamic system without time delay with a gain K and a time constant T given as

$$G(s) = \frac{K}{Ts + 1}. \quad (4.45)$$

4.3.1.1 Derive the response of the system $y(t)$ if the input is $u(t) = A \sin(\omega t)$. Discuss the results.

Using the transfer function of the system:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{K}{Ts + 1}, \quad (4.46)$$

we obtain:

$$Y(s) = \frac{K}{Ts + 1} U(s). \quad (4.47)$$

Next, we introduce the Laplace transform of the input:

$$u(t) = A \sin(\omega t) \quad \xrightarrow{\mathcal{L}(\cdot)} \quad U(s) = \frac{A\omega}{s^2 + \omega^2}. \quad (4.48)$$

and obtain:

$$Y(s) = \frac{K}{Ts + 1} \frac{A\omega}{s^2 + \omega^2}. \quad (4.49)$$

Partial fractioning gives

$$\frac{K}{Ts + 1} \frac{A\omega}{s^2 + \omega^2} = \frac{C}{Ts + 1} + \frac{Ds + E}{s^2 + \omega^2}, \quad (4.50)$$

which leads to the relationship

$$KA\omega = C(s^2 + \omega^2) + (Ds + E)(Ts + 1) \quad (4.51)$$

$$= (C + DT)s^2 + (D + ET)s + C\omega^2 + E. \quad (4.52)$$

We compare left- and right-hand coefficients to get:

$$0 = C + DT \quad \Rightarrow \quad C = -DT = (KA\omega - C\omega^2)T^2, \quad (4.53)$$

$$0 = D + ET \quad \Rightarrow \quad D = -ET = -(KA\omega - C\omega^2)T, \quad (4.54)$$

$$KA\omega = C\omega^2 + E \quad \Rightarrow \quad E = KA\omega - C\omega^2. \quad (4.55)$$

Re-arranging gives

$$C + C\omega^2 T^2 = KA\omega T^2 \quad \Rightarrow \quad C = \frac{KA\omega T^2}{1 + \omega^2 T^2}, \quad (4.56)$$

$$D = -\left(KA\omega - \frac{KA\omega^3 T^2}{1 + \omega^2 T^2}\right) T = -\frac{KA\omega T}{1 + \omega^2 T^2}, \quad (4.57)$$

$$E = KA\omega - \frac{KA\omega^3 T^2}{1 + \omega^2 T^2} = \frac{KA\omega}{1 + \omega^2 T^2}. \quad (4.58)$$

The final Laplace transform of the output is

$$Y(s) = \frac{KA\omega}{1 + \omega^2 T^2} \left(\frac{T^2}{Ts + 1} + \frac{-Ts + 1}{s^2 + \omega^2} \right) \quad (4.59)$$

$$= \frac{KA\omega}{1 + \omega^2 T^2} \left(\frac{T^2}{Ts + 1} + \frac{-Ts}{s^2 + \omega^2} + \frac{1}{s^2 + \omega^2} \right). \quad (4.60)$$

Using inverse Laplace transform, we obtain:

$$y(t) = \frac{KA\omega}{1 + \omega^2 T^2} \left(T e^{-\frac{t}{T}} - T \cos(\omega t) + \frac{1}{\omega} \sin(\omega t) \right). \quad (4.61)$$

Here the first term vanishes over time. Now we can use a trigonometric identity

$$a_1 \cos(b) + a_2 \sin(b) = \sqrt{a_1^2 + a_2^2} \sin \left(b + \tan^{-1} \left(\frac{a_1}{a_2} \right) \right). \quad (4.62)$$

We obtain the intermediate result

$$y(t) = \frac{KA\omega}{1 + \omega^2 T^2} \sqrt{T^2 + \left(\frac{1}{\omega} \right)^2} \sin(\omega t + \underbrace{\tan^{-1}(-T\omega)}_{\varphi}), \quad (4.63)$$

which upon simplification gives

$$y(t) = \frac{KA}{\sqrt{1 + \omega^2 T^2}} \sin(\omega t + \varphi). \quad (4.64)$$

Both the magnitude and the phase depend on the frequency ω . The magnitude is given by $|\frac{KA}{\sqrt{1 + \omega^2 T^2}}|$ and the phase is $\tan^{-1}(-T\omega)$. The magnitude and phase both decrease as the frequency increases.

- 4.3.1.2 Verify the observations made by simulating the response of the system with $K = 2$ and $T = 10$ to the input $u(t) = A \sin(\omega t)$ with $A = 1$ and three different frequencies $\omega = \{0.01, 0.1, 10\}$. Discuss the results.

We will make use of the Simulink model shown in Figure 4.11 and apply different input frequencies $\omega = \{0.01, 0.1, 10\}$. The results are shown in Figure 4.12.

The simulation results confirm the theoretical findings. At small frequencies, the magnitude of the input is given by $\approx |KA|$. An increase of the input frequency decreases the magnitude towards 0. The phase is close to 0 at small frequencies and then decreases towards -90° . At $\omega = 0.1$, it is -45° .

- 4.3.1.3 Plot the frequency response of the system using the Bode and Nyquist diagrams. Discuss the results.

The Bode and Nyquist diagrams of the system can be created in MATLAB and are shown in Figure 4.13. The vertical line represents the pole location of the system at $s = 0.1$.

```
g = tf(2, [10 1]);
```

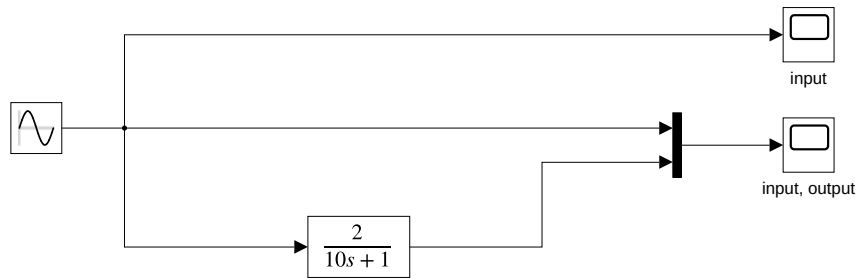


Figure 4.11: Simulation of response of the first-order system to a harmonic signal.

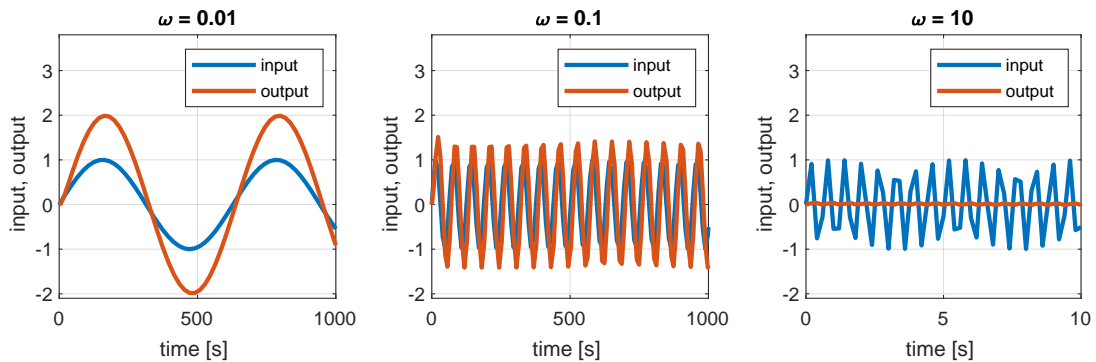


Figure 4.12: Simulation results of the response of the first-order system to a harmonic signal.

```
bode(g); grid
figure; nyquist(g)
```

Bode diagram: the magnitude plot shows a decay of 20 dB per decade after reaching the frequency of the pole. The phase plot reaches -90° which shows the presence of one pole and no zeros.

Nyquist diagram: MATLAB plots two curves which are symmetric about the x-axis. The upper one is for negative frequencies $(-\infty, 0)$ and the lower one for positive frequencies $(0, \infty)$. The positive plot starts at $K + 0j$ for $\omega = 0$ and asymptotically approaches the point $0 + 0j$ for $\omega \rightarrow \infty$. The phase shift is always negative and varies from 0° at low frequencies to -90° at high frequencies.

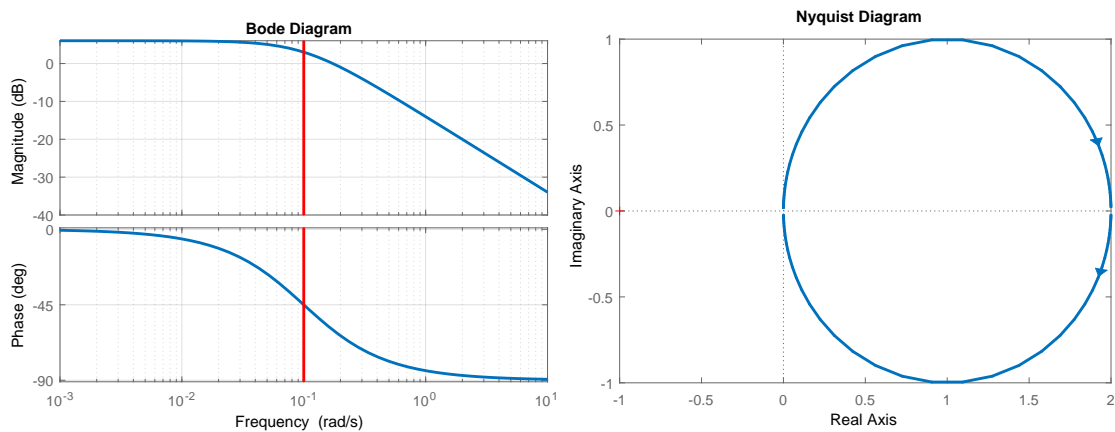


Figure 4.13: Bode and Nyquist diagrams for the first-order system.

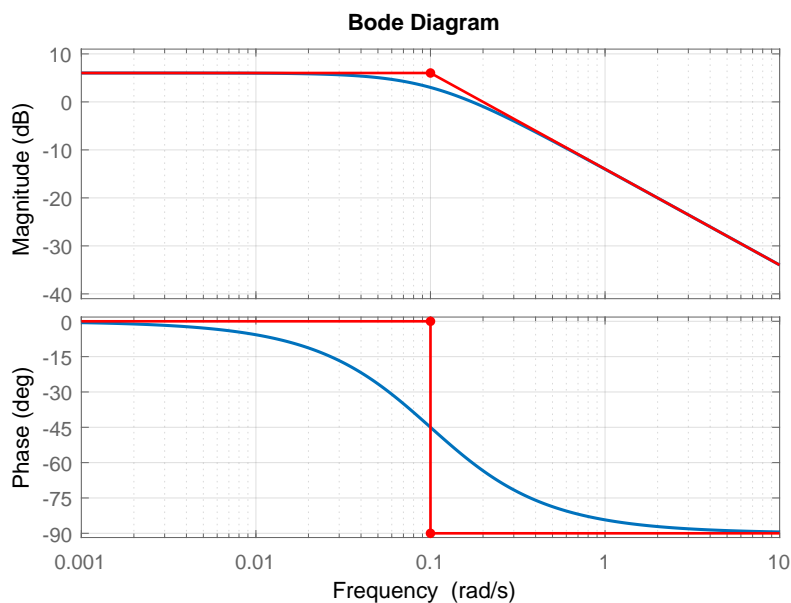


Figure 4.14: Bode diagram of the first-order system with the asymptotes in the magnitude and phase plots.

4.3.1.4 Draw the asymptotes that correspond to the beginning and the end of the magnitude and phase plots. Comment on the results.

The asymptotes of the magnitude plot of the Bode diagram are lines with an increase/decrease by integer multiple of 20 dB per decade (the top part of Figure 4.14). The maximum deviation between the Bode diagram and the asymptotes is about 3 dB and occurs at the pole location (at the corner frequency).

The lower plot shows the horizontal asymptotes on the phase plot.

4.3.2 A One-Dimensional Car

We study the motion of a car modelled in Section 1.3.1.

Plot the Bode diagram. Discuss the results.

```
g = tf(1, [1 0 0])
bode(g)
grid
h1=findall(gcf); hline=findobj(gcf, 'Type', 'line');
for i=1:length(hline)
    hline(i).LineWidth=2;
end
iam_save_pdf('bod_plt_car.pdf');
```

The phase plot is constant at $-\pi$, which suggests the presence of a double integrator (Figure 4.15). This is confirmed by the magnitude plot, which crosses zero at the frequency $\omega = 1$ and decreases by 40 dB per decade.

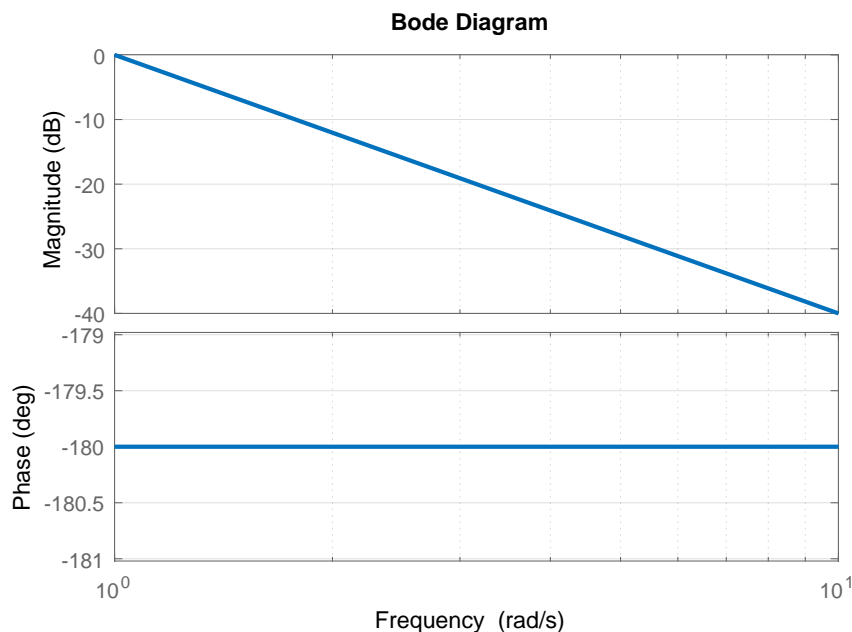


Figure 4.15: Bode diagram of the one-dimensional car.

4.3.3 Two Tanks in Series

We study a system with two tanks in series (Example 1.2.4), for which we have derived the transfer matrix in Section 2.3.2.

4.3.3.1 Plot the Bode diagram and discuss the results. The parameters of the system are $k_{11} = 1.15 \text{ m}^{2.5}/\text{s}$, $k_{22} = 1.3 \text{ m}^{2.5}/\text{s}$, $F_1 = 0.5 \text{ m}^2$, and $F_2 = 0.8 \text{ m}^2$. Use the input values of $q_{0,1} = 0.3 \text{ m}^3/\text{s}$ and $q_{0,2} = 0.5 \text{ m}^3/\text{s}$ to define the steady state.

```

q01s = 0.3; q02s = 0.5;
k11 = 1.15; k22 = 1.3;
h1s = (q01s/k11)^2; h2s = (q01s/k22 + q02s/k22)^2;

k1 = k11/2/sqrt(h1s); k2 = k22/2/sqrt(h2s);
F1 = 0.5; F2 = 0.8;

g = tf(k1/(F1*F2), [1 (k1/F1+k2/F2) k1*k2/(F1*F2)]);
figure; bode(g);
grid on;

iam_save_pdf('bod_plt_tank.pdf');

keyboard % select the first plot

iam_save_pdf('bod_plt_tank.pdf');

poles = pole(g);
hold on
plot(-[poles(1), poles(1)], [-180, 180])
plot(-[poles(2), poles(2)], [-180, 180])

keyboard % select the first plot
% plot the asymptotes by hand

```

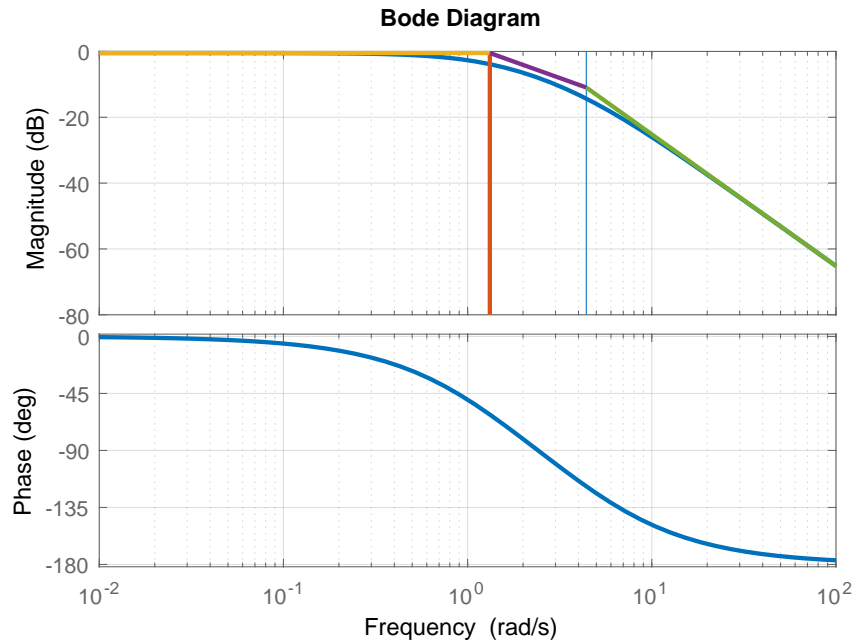



Figure 4.16: Bode diagram of two liquid tanks in series.

```
hold on
plot(-[poles(1), poles(1)], [-180, 180])
plot(-[poles(2), poles(2)], [-180, 180])

iam_save_pdf('bod_plt_tank.pdf');
```

Similar conclusions can be made as in the case of a first-order system. Yet, we can observe a decrease of the phase up to $-\pi$ for increasing frequencies (Figure 4.16). We can also observe that for high frequencies the decrease in magnitude is 40 dB per decade. These two observations suggest the presence of two poles. When we plot the poles and asymptotes of the contribution of each pole and the gain, we clearly see that the asymptotes intersect at the frequencies that correspond to the poles of the transfer function.

4.4 Problems

Problem. 4.1: Systems with a time delay and with a nonminimum phase zero

Compare the frequency characteristics of the following systems

$$G_{11}(s) = \frac{1}{s+1}, \quad G_{12}(s) = \frac{1}{s+1}e^{-5s}, \quad (4.65)$$

$$G_{21}(s) = \frac{s+1}{s+10}, \quad G_{22}(s) = \frac{s-1}{s+10}. \quad (4.66)$$

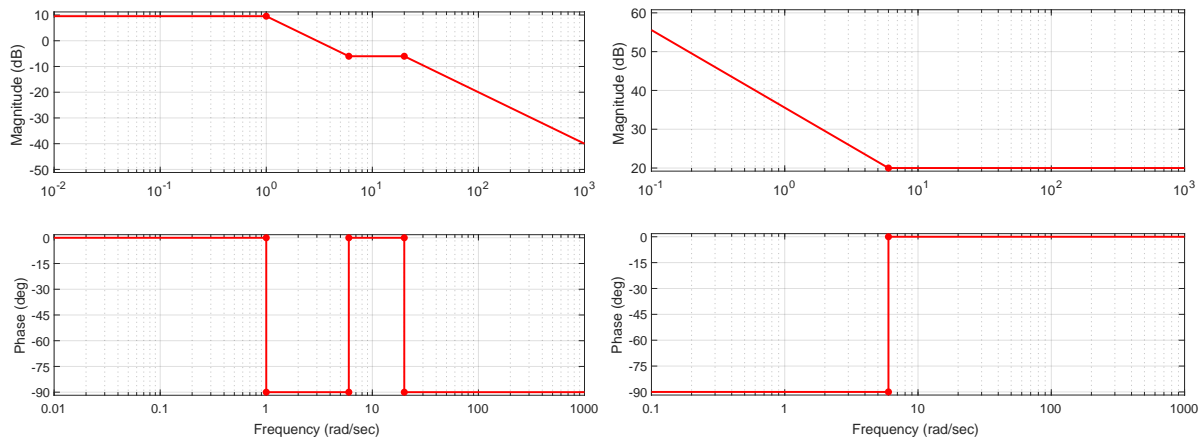


Figure 4.17: Identify respective transfer functions from the given asymptotes.

```
G11 = tf(1, [1 1]); G12 = tf(1, [1 1], 'InputDelay', 5);
G21 = tf([1 1], [1 10]); G22 = tf([1 -1], [1 10]);
bode(G11); figure; bode(G21);
figure; bode(G21); figure; bode(G22);
```

Note that a time delay or nonminimum phase behaviour do not change the magnitude plot. Hence, unlike for minimum phase systems where we are able to derive the phase plot based on the known magnitude plot, this is not possible in the case of nonminimum phase system.

Problem. 4.2: Lead-lag system

A lead-lag system is characterised by one pole and one zero

$$G(s) = \frac{T_1 s + 1}{T_2 s + 1}. \quad (4.67)$$

Plot the Nyquist and Bode diagrams considering both cases: $T_1 > T_2$ and $T_1 < T_2$,

```
T1 = 1; T2 = 2;
G1 = tf([T1 1], [T2 1]); G2 = tf([T2 1], [T1 1]);
bode(G1, G2); figure; nyquist(G1, G2)
```

Problem. 4.3: Asymptotes in the Bode diagrams

Draw asymptotes of the magnitude and phase plots of the Bode diagram for systems with the following transfer functions

$$G_1(s) = \frac{0.1s + 1}{s(s^2 + 3s + 2)}, \quad G_2(s) = \frac{10}{s^2(s^2 + 3s + 2)}. \quad (4.68)$$

Verify your results using Bode diagrams.

Problem. 4.4: Identification of systems from asymptotes

Derive respective transfer functions from the Bode diagram asymptotes shown in Figure 4.17. Verify your results using Bode diagrams.

The resulting transfer functions are

$$G_1(s) = \frac{10(s+6)}{(s+1)(s+20)},$$

$$G_2(s) = \frac{10(s+6)}{s}.$$

Problem. 4.5: Stability margins

Find stability margins for the following systems

$$G_1(s) = \frac{10(s+6)}{(s+1)(s+20)}, \quad G_2(s) = \frac{10(s+6)}{s}, \quad G_3(s) = \frac{5}{(s+1)(2s+1)(5s+1)}. \quad (4.69)$$

$$g_{m,1} = \infty, \quad \varphi_{m,1} = 126^\circ,$$

$$g_{m,2} = \infty, \quad \varphi_{m,2} = \infty,$$

$$g_{m,3} = 8 \text{ dB}, \quad \varphi_{m,3} = 33^\circ.$$

4.5 MATLAB Snippets

- Complex number representations

```
z = complex(a, b); z1 = a+b*j; z2=c+d*i; % instance of complex numbers
zbar = conj(z)
re = real(z); im = imag(z); magn = abs(z); phase = angle(z)
zn = magn * exp(j * phase) % zn == z
```

- Evaluation of a frequency response at a certain frequency based on the transfer function

```
g = tf(2, [10 1])
omega = 4
evalfr(g, omega*j)
freqresp(g, omega)
```

- Bode and Nyquist plots, stability margins

```
K = 2; T = 10; g1 = tf(K, [T 1]); g2 = tf(K, [T^2 2*T 1])
bode(g1); % plot Bode diagram
[magnitude, phase, frequency] = bode(g1); % do not plot, store graphs
bode(g1, g2);
nyquist(g1)
margin(g1)
```

- Customisation of Bode diagrams

```
K = 2; T = 10; g1 = tf(K, [T 1])
h = bodeplot(tf(2, [10 1]));
bodeoptions % see a list of changeable items
setoptions(h, 'FreqUnits', 'Hz', 'PhaseVisible', 'off'); % plot magnitude only
setoptions(h, 'PhaseVisible', 'on', 'MagVisible', 'off'); % plot phase only
```

- Plotting the asymptotes on Bode diagram¹

```
s = tf('s'); g = K/(T*s+1)
w = logspace(-3, 1);
asymp(g, w(1), w(end))
```

- Manually plotting the Bode diagram with the asymptotes

```
w = logspace(-2, 2);
[m1, p1]=bode(1, [1 1], w); [m2, p2] = bode([1, 1], 1, w);
subplot(211)
semilogx(w, 20 * log10(m1)); hold on; semilogx(w, 20 * log10(m2))
semilogx([0.01 1 100], [0 0 40]); semilogx([0.01 1 100], [0 0 -40])
grid on; hold off
xlabel('Frequency (rad/sec)'), ylabel('Magnitude (dB)')

subplot(212)
semilogx(w, p1); hold on; semilogx(w, p2)
semilogx([0.01 0.1 10 100], [0 0 90 90]);
semilogx([0.01 0.1 10 100], [0 0 -90 -90])
grid on; hold off
xlabel('Frequency (rad/sec)'), ylabel('Phase (deg)')
legend('pole', 'zero', 'asympt pole', 'asympt zero')
```

4.6 Python Snippets

- Complex number representations

```
z = complex(3, 4); z1 = 3+4j; z2 = 1+2j; # instance of complex numbers
zbar = z.conjugate()
re = z.real; im = z.imag; magn = abs(z)
import cmath, math, numpy
phase = cmath.phase(c)
print('Phase in Degrees =', numpy.degrees(phase))
zn = cmath.polar(z) # zn is z in polar coordinates
```

- Evaluation of a frequency response at a certain frequency based on the transfer function

```
from control.matlab import *
g = tf([2], [10, 1])
omega = 4; s = 1j * omega
evalfr(g, s)
frequencies = [0.1, 1, 10]
mag, phase, omega = freqresp(g, frequencies)
```

- Bode and Nyquist plots, stability margins

```
from control.matlab import *
K = 2; T = 10; g1 = tf(K, [T, 1]); g2 = tf(K, [T**2, 2*T, 1])
bode(g1, plot=True); plt.show()
magnitude, phase, frequency = bode(g1) # do not plot, store graphs
bode(g1, g2, plot = True); plt.show()
nyquist(g1)
gm, pm, wg, wp = margin(g1)
```

- Manually plotting the Bode diagram with the asymptotes

```
from control.matlab import *
import matplotlib.pyplot as plt
import numpy as np
import math
w = logspace(-2, 2)
```

¹<https://uk.mathworks.com/matlabcentral/fileexchange/10183-bode-plot-with-asymptotes>

```
g1 = tf(1, [1, 1]); g2 = tf([1, 1], 1)
m1, p1, om1 = bode(g1, w); m2, p2, om2 = bode(g2, w)
plt.close('all')
plt.subplot(2, 1, 1)
plt.semilogx(om1, 20 * np.log10(m1), om2, 20 * np.log10(m2))
plt.semilogx([0.01, 1, 100], [0, 0, 40]); plt.semilogx([0.01, 1, 100], [0, 0,
-40])
plt.grid()
plt.xlabel('Frequency (rad/sec)'); plt.ylabel('Magnitude (dB)')

plt.subplot(2,1,2)
p2 += 2 * math.pi
plt.semilogx(w, p1, w, p2)
plt.semilogx([0.01, 0.1, 10, 100], [0, 0, math.pi/2, math.pi/2]);
plt.semilogx([0.01, 0.1, 10, 100], [0, 0, -math.pi/2, -math.pi/2])
plt.grid()
plt.xlabel('Frequency (rad/sec)'); plt.ylabel('Phase (deg)')
plt.legend(['pole', 'zero', 'asympt pole', 'asympt zero'])
plt.show()
```

4.7 Bibliography

Although chemical process control is mostly studied in time domain, frequency representation can also be useful. The theoretical background for this chapter can be found in Mikleš and Fikar (2007, Chapter 4), information in Slovak in Huba, Hubinský, and Žáková (2006, Chapter 5), Mikleš and Hutla (1986, Chapter 2).

For complementary information, see for example Ogata (2010, Chapter 7), Golnaraghi (2010, Chapter 8), or Dorf and Bishop (2008, Chapter 8).

Control Performance

The main objective of the control design is to guarantee closed-loop stability and performance. In this chapter, we will introduce various specifications to assess performance. Mainly,

- we review the already discussed time-domain specifications and add some new ones,
- we introduce optimal control measures: integral cost functionals, performance indices ,
- we discuss frequency-domain indices both in open- and closed-loop setup,
- we introduce standard polynomials.

We will investigate how are these specifications related to the placement of poles and zeros in the complex plane.

5.1 Overview

5.1.1 Time-Domain Specifications

The most commonly used performance indices are the time-domain ones that specify the speed of the response. It is sought to make them as small as possible:

Settling time T_ϵ is given as the minimum time after which the output value remains within some prescribed percentile distance ϵ from the steady state. The usually values considered are about 2 – 5%.

Rise time T_b is the time required for the response to rise from 0% to $b\%$ of its final value. For underdamped responses, the value T_{100} is considered (indicating time when the output reaches its final value for the first time). For overdamped responses, T_{90} is commonly used as the time elapsing between 0% and 90% of the new steady state output.

Dominant time constant T_{DOM} If one time constant is at least 10 times slower than other ones, it dominates the dynamic properties of the process. The corresponding pole is then the

dominant pole (or a couple of dominant poles if they are complex conjugates) and it is the closest pole to the origin of the complex plane.

Peak time T_σ is the time when the maximum overshoot occurs.

Feedback performance ratio is the ratio of the closed-loop to the open-loop settling times.

Other time-domain performance indices include:

Maximum overshoot e_{\max} or σ is usually defined in percentage as a ratio of the maximum (y_{\max}) and steady state (new steady state: y_∞ , previous steady state: y_0) outputs

$$e_{\max} = \sigma = \frac{y_{\max} - y_\infty}{y_\infty - y_0} \times 100\%. \quad (5.1)$$

The recommended value is $e_{\max} < 25\%$ for the underdamped response specifications. Note that this definition is suitable for setpoint changes. If the step response is overdamped and without overshoot then $e_{\max} = 0$. If a step disturbance is considered, the absolute value of e_{\max} can be used (Figure 5.1(b)).

Decay rate is defined as the ratio between amplitudes of two subsequent peaks of an underdamped response. The recommended value is up to 30%.

Steady-state control error $e(\infty)$ (**bias, offset**) is the difference between the setpoint and the process output in the steady state. For some processes, a nonzero value can be accepted, otherwise a zero value is usually desired.

A graphical representation of selected time-domain indices is shown in Figure 5.1. We can distinguish between the closed-loop response to a step change in the setpoint or the disturbance.

Note that Figure 5.1(a) also depicts the times T_u (intersection of the time axis and the tangent to the response at the inflection point) and T_n (intersection of the output steady-state value and the tangent to the response at the inflection point). These indices can be used for process identification from its step response.

Time-domain indices are an intuitive and preferred way to specify closed-loop performance. If the controlled process is of first- or second-order, we have seen that there is a direct relationship between these indices and process characteristics. However, for higher-order systems, we can only approximate their behaviour. Also, the actual controller design in time domain is not so straightforward.

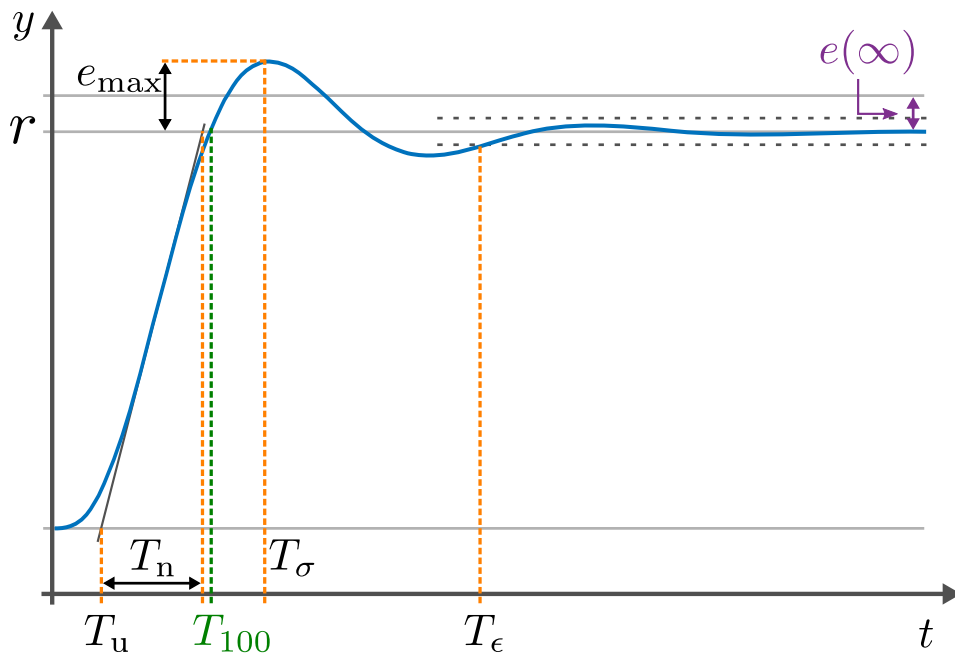
5.1.2 Integral Criteria

Integral criteria possess one advantage over the time-domain specifications. Instead of multiple indices, only one is sufficient to characterise the performance as a function of the control error that should ideally be minimised. The respective general cost function can be defined as

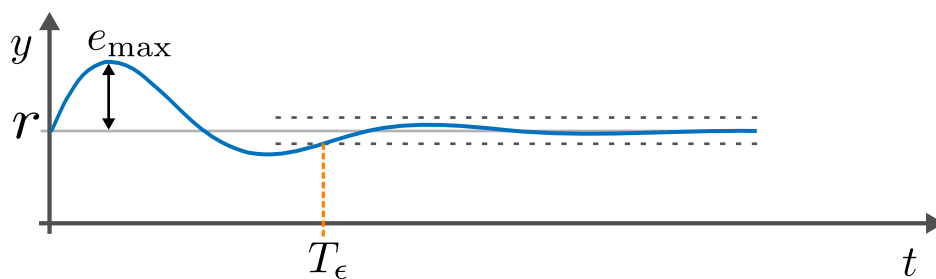
$$J = \int_0^\infty f[e(t)]dt. \quad (5.2)$$

Most often, the following forms are used:

IAE – integral absolute value of error $f = |e(t)|$: This form is suitable for oscillatory responses. However, from the computational point of view, it can be difficult to implement as the absolute value is nondifferentiable.



(a) Response to a step change in the setpoint.



(b) Response to a step change in the disturbance.

Figure 5.1: Time-Domain specifications.

ITAE – integral time multiplied absolute value of error $f = |e(t)|t$: the additional time term improves the settling time.

ISE – integral squared value of error $f = e^2(t)$: similar to IAE. Due to squaring, large values of control error are penalised more heavily than the small ones. Closed-loop system usually shows larger settling times as compared to IAE cost. However, squared error is mathematically convenient for analytical purposes.

Q – quadratic $f = e^2(t) + \lambda u^2(t)$: penalises both the manipulated variable (control effort) and the control error (control quality) and simply adjusts the trade-off between speed and robustness of the controller.

5.1.3 Frequency-Domain Indices

One of the drawbacks of the time-domain performance indices is that they do not consider stability issues. This is more easily performed in the frequency domain, where we can distinguish between open- and closed-loop criteria. Another advantage of the frequency-domain approach is that higher-order systems pose no problems and that certain types of controllers can be tuned

very efficiently. On the other hand, it can be difficult to specify frequency-domain criteria because their relationship to the time-domain performance is not so obvious.

Let us recall the open-loop frequency-domain indices from Section 4.1.3:

Gain margin (g_m) is the amount of gain in decibels (dB) that can be added to the open-loop system before the closed-loop system becomes unstable. If step responses are considered, the recommended values of g_m are between -12 to -20 dB, for disturbance rejection between -4 to -9 dB.

Phase margin (φ_p) is the phase (angle) by which the Nyquist plot of the open-loop system has to be rotated about the origin so that the gain crossover passes the point $-1 + 0j$ and the closed-loop system becomes unstable. The recommended values of φ_p are for setpoint tracking between 40° to 60° and for disturbance rejection between 20° to 50° .

For an underdamped second-order system (3.41), we can derive the gain-crossover frequency ω_g and phase margin as

$$\frac{\omega_g}{\omega_0} = \sqrt{\sqrt{4\zeta^4 + 1} - 2\zeta^2}, \quad (5.3)$$

$$\varphi_p = \arctan\left(2\zeta \frac{\omega_0}{\omega_g}\right). \quad (5.4)$$

Useful approximations for higher-order systems related to other indices and parameters are

$$\omega_g T_{50} \approx 1.5 - \frac{e_{\max}[\%]}{250}, \quad (5.5)$$

$$70 \approx \varphi_p[^\circ] + e_{\max}[\%], \quad (5.6)$$

$$\zeta \approx \frac{\varphi_p}{100[^\circ]}, \quad \zeta \in (0.0, 0.6). \quad (5.7)$$

When the closed-loop transfer function between the output and the setpoint is considered, several frequency-domain performance criteria can be taken into account:

Bandwidth ω_b is defined as the frequency range where the magnitude is approximately constant compared to the value at some specified frequency (usually zero) and differs by not more than -3 dB, so that the larger the bandwidth, the faster the response.

Resonant peak is the maximum of the magnitude frequency response. Recommended values are between 1.1 to 1.5. This indicates relative stability. For the second-order system see (4.33).

Resonant frequency ω_r is the frequency at which the resonance peak occurs. See (4.32) for the second-order system.

Bandwidth is defined by $|G(j\omega_b)| = 1/\sqrt{2}|G(0)|$. For the underdamped second-order system (3.41),

$$\omega_b = \omega_0 \sqrt{1 - 2\zeta^2 + \sqrt{1 + (1 - 2\zeta^2)^2}}, \quad (5.8)$$

$$\tan \varphi_b = \frac{2\zeta \sqrt{1 - 2\zeta^2 + \sqrt{1 + (1 - 2\zeta^2)^2}}}{2\zeta^2 - \sqrt{1 + (1 - 2\zeta^2)^2}}, \quad (5.9)$$

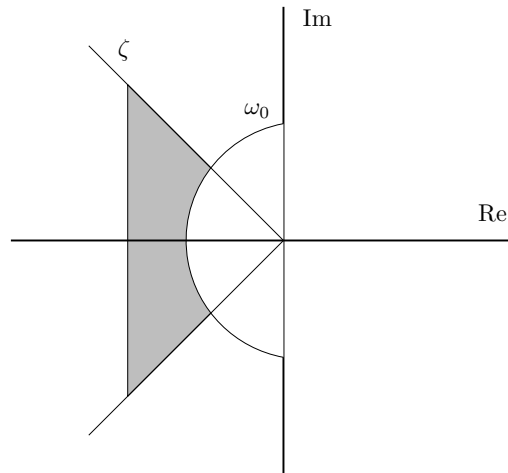


Figure 5.2: Area of desired pole locations.

which are functions of the damping ratio ζ . If higher-order systems are considered, there are some useful approximations that relate these indices to the transfer function parameters:

$$\frac{\omega_b}{\omega_0} \approx 1.8 - 1.1\zeta, \quad \zeta \in (0.3, 0.8), \quad (5.10)$$

$$|\varphi_b| \approx \pi - 2.23\zeta, \quad \zeta \in (0, 1), \quad (5.11)$$

$$\omega_b T_{50} \approx 2.3, \quad \zeta \in (0.3, 0.8). \quad (5.12)$$

5.1.4 Pole Placement

Finally, we can express the desired closed-loop performance directly by placing the closed-loop transfer-function poles and zeros. We usually start by inspecting the poles and zeros of the controlled system. A general rule of thumb is to move dominant poles more to the left, but not too far as that would make the closed-loop system response too fast with large values of control actions.

Let us recall properties of a second order system in terms of its dominant poles:

- the distance from the imaginary axis: $\zeta\omega_0$, if too small, the response sluggish and/or oscillatory ,
- the distance from the origin is equal to ω_0 (3.43),
- the angle to the horizontal axis depends on ζ (3.44), it is recommended that $\zeta > 0.7$ which represents less than 45° .

Desired pole locations are shown in Figure 5.2 as area within the desired angles (as a function of ζ) and outside of semicircle characterised by the frequency ω_0 .

One of possible approaches is to define the n -th closed-loop characteristic polynomial as:

- A binomial expansion form with a real stable pole $s = -\omega_0$ with a multiplicity of n .

$$\begin{aligned} & s + \omega_0 \\ & s^2 + 2\omega_0 s + \omega_0^2 \\ & s^3 + 3\omega_0 s^2 + 3\omega_0^2 s + \omega_0^3 \\ & s^4 + 4\omega_0 s^3 + 6\omega_0^2 s^2 + 4\omega_0^3 s + \omega_0^4 \end{aligned}$$

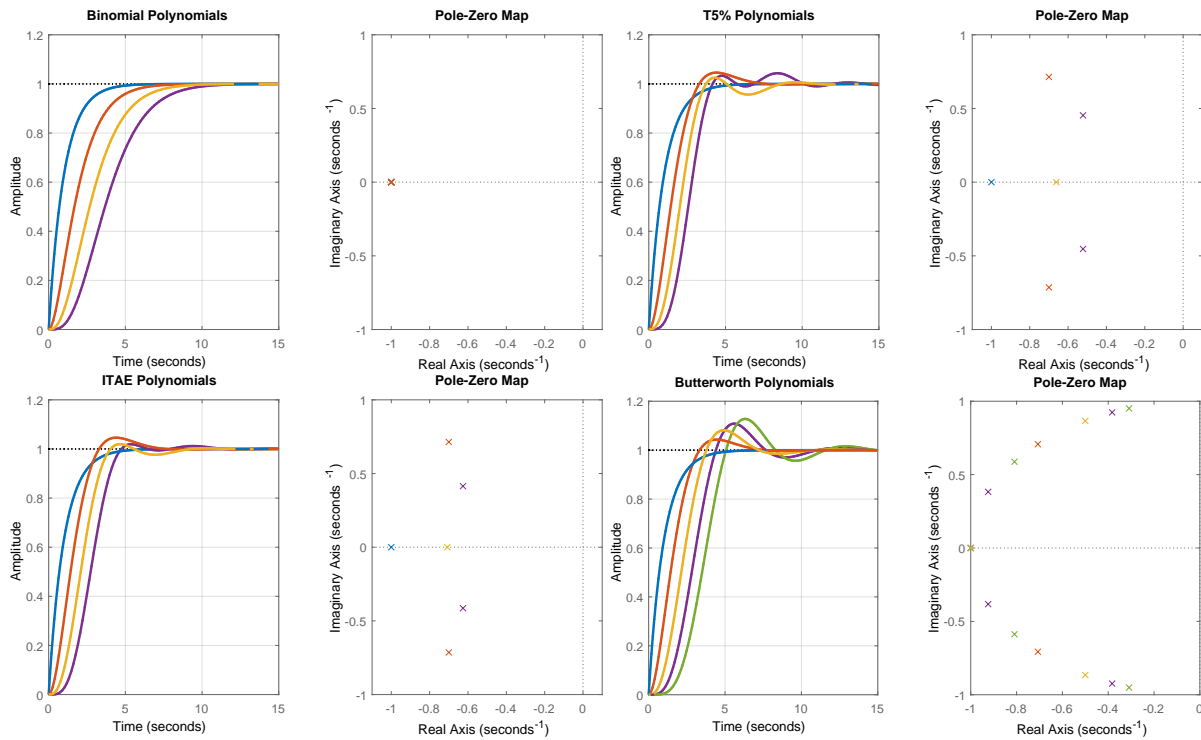


Figure 5.3: Step responses and corresponding pole locations of systems with standard polynomials in the denominator: (top left) binomial, (top right) minimal $T_{5\%}$, (bottom left) ITAE, (bottom right) Butterworth).

- Minimal $T_{5\%}$ – the fastest transient response with a maximum overshoot of 5%.

$$\begin{aligned}
 & s + \omega_0 \\
 & s^2 + 1.4\omega_0 s + \omega_0^2 \\
 & s^3 + 1.55\omega_0 s^2 + 2.10\omega_0^2 s + \omega_0^3 \\
 & s^4 + 1.60\omega_0 s^3 + 3.15\omega_0^2 s^2 + 2.45\omega_0^3 s + \omega_0^4
 \end{aligned}$$

- Minimum of the ITAE cost function.

$$\begin{aligned}
 & s + \omega_0 \\
 & s^2 + 1.4\omega_0 s + \omega_0^2 \\
 & s^3 + 1.75\omega_0 s^2 + 2.15\omega_0^2 s + \omega_0^3 \\
 & s^4 + 2.1\omega_0 s^3 + 3.4\omega_0^2 s^2 + 2.7\omega_0^3 s + \omega_0^4
 \end{aligned}$$

- Butterworth polynomials (roots are located on a semicircle in the left half-plane with the radius ω_0 symmetrically distributed with respect to the imaginary axis).

$$\begin{aligned}
 & s + \omega_0 \\
 & s^2 + 1.4\omega_0 s + \omega_0^2 \\
 & s^3 + 2\omega_0 s^2 + 2\omega_0^2 s + \omega_0^3 \\
 & s^4 + 2.6\omega_0 s^3 + 3.4\omega_0^2 s^2 + 2.6\omega_0^3 s + \omega_0^4
 \end{aligned}$$

Step responses and pole locations for standard polynomials with $\omega_0 = 1$ are shown in Figure 5.3.

Note that standard polynomials only approximate the desired behaviour, assuming there are no transfer function zeros. Obviously, the closed-loop dynamics is also influenced by its zeros.

When control is expensive, we can place the dominant closed-loop poles at the stable plant poles and at mirrors (the reflection about the y-axis) of unstable plant poles. On the other hand, when

control is cheap, we can place dominant closed-loop poles at stable plant zeros and mirrors of unstable plant zeros, while placing the other nondominant poles according to the Butterworth pattern.

5.2 Examples

Example 5.1: Time-domain indices

Consider an unstable process with the transfer function

$$G_S(s) = \frac{1}{s(s + 0.5)}, \quad (5.13)$$

controlled with a unit-gain proportional controller. Determine the rise time, peak time, maximum overshoot, settling time, and steady-state control error if there is a unit-step change in the setpoint.

The closed-loop transfer function between the setpoint and the process output with the controlled process and controller is given by

$$G_{cl}(s) = \frac{G_S(s)G_c(s)}{1 + G_S(s)G_c(s)}, \quad (5.14)$$

where $G_c(s)$ is the controller transfer function. This gives in our case the second-order transfer function (3.41)

$$G_{cl}(s) = \frac{1}{s^2 + 0.5s + 1}, \quad (5.15)$$

with complex conjugate poles with $\omega_0 = 1$ and $\zeta = 0.25$ and the gain $K_p = 1$. Therefore, we can use formulae from Section 3.1.4 ($P = \sqrt{1 - \zeta^2}$).

$$T_{100} = \frac{1}{\omega_0 P} \left[\pi - \tan^{-1} \left(\frac{P}{\zeta} \right) \right] = 1.88, \quad (5.16)$$

$$T_{0.02} \approx \frac{4}{\zeta \omega_0} = 16, \quad T_{0.05} \approx \frac{3}{\zeta \omega_0} = 12, \quad (5.17)$$

$$T_\sigma = \frac{\pi}{\omega_0 P} = 3.24, \quad (5.18)$$

$$e_{\max} = e^{-\frac{\pi \zeta}{P}} = 44\%. \quad (5.19)$$

The steady-state control error for a unit-step change in the setpoint can be calculated using the Final Value Theorem

$$y(\infty) = \lim_{s \rightarrow 0} sY(s) = \lim_{s \rightarrow 0} sG_{cl}(s) \frac{1}{s} = G_{cl}(0) = 1. \quad (5.20)$$

Therefore, $e(\infty) = 1 - y(\infty) = 0$.

```
Gs = tf(1, [1, 0.5, 0]);   Gc = 1;
Gcl = Gs * Gc / (1 + Gs * Gc);
step (Gcl)
```

Example. 5.2: Controller design with time-domain indices

Consider again the unstable process with the transfer function

$$G_S(s) = \frac{1}{s(s + 0.5)}, \quad (5.21)$$

controlled with a proportional controller. Determine the gain of the controller so that the settling time is 10 s and the maximum overshoot of the output variable is 10% if there is a unit-step change in the setpoint.

The closed-loop transfer function between the setpoint and the process output with the controlled process and controller is given as

$$G_{cl}(s) = \frac{G_S(s)G_c(s)}{1 + G_S(s)G_c(s)}, \quad (5.22)$$

where $G_c(s) = K_p$ is the controller transfer function. This gives, in our case, a second-order transfer function (3.41)

$$G_{cl}(s) = \frac{K_p}{s^2 + 0.5s + K_p}, \quad (5.23)$$

with complex conjugate poles with $\omega_0 = \sqrt{K_p}$ and $\zeta = 0.25/\omega_0$. The expressions for $T_{0.05}$ and e_{\max} are functions of ζ and ω_0

$$T_{0.05} \approx \frac{3}{\zeta\omega_0} \iff \omega_0 = \frac{3}{\zeta T_{0.05}}, \quad (5.24)$$

$$e_{\max} = e^{-\frac{\pi\zeta}{P}} \iff \zeta = \frac{|\ln e_{\max}|}{\sqrt{\pi^2 + \ln^2 e_{\max}}}. \quad (5.25)$$

Then, (5.25) gives $\zeta = 0.59$ and (5.24) gives $\omega_0 = 0.51$. Therefore, $K_p = \omega_0^2 = 0.26$.

The Simulink model and the resulting step response are shown in Figure 5.4. We can see that the desired indices are fulfilled.

```
Gp = tf(1, [1, 0.5, 0]);   Gc = 0.26;
Gcl = Gp * Gc / (1 + Gp * Gc);
step (Gcl)
```

Example. 5.3: Gain and phase margins

Let us consider a process with the transfer function

$$G(s) = \frac{20}{s^2 + 10s + 100}, \quad (5.26)$$

controlled with an integral controller $G_c(s) = 20/s$. Determine the gain and phase margins.

We will first derive the expressions for the magnitude and phase of the open-loop system

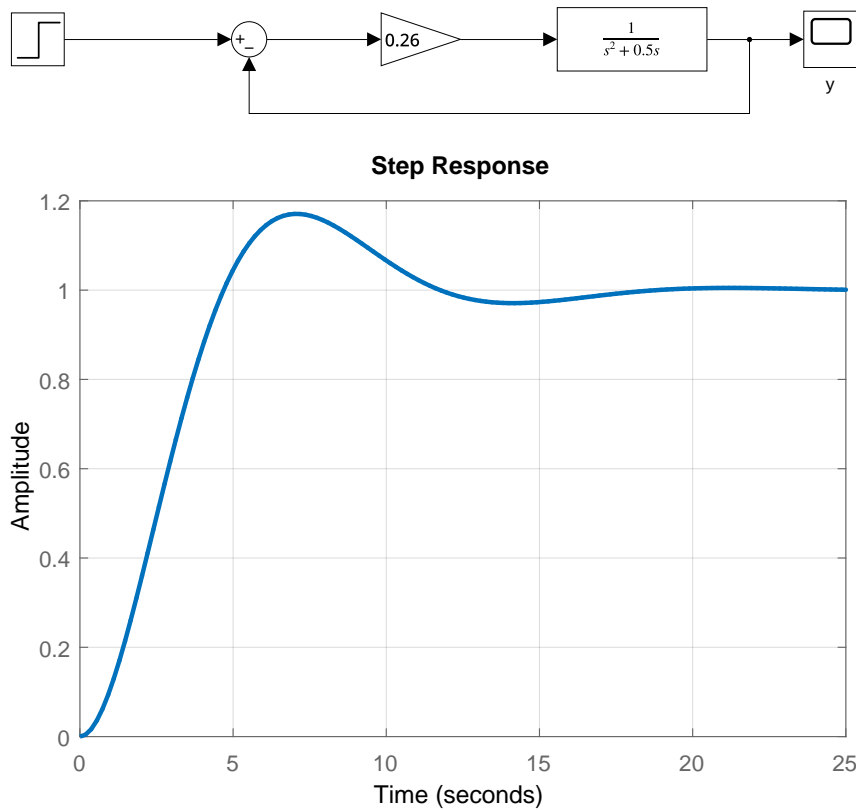


Figure 5.4: A Simulink model and step response for Example 5.2.

$G(s)G_c(s)$ with the transfer function of the form

$$G_o(s) = \frac{400}{s(s^2 + 10s + 100)}. \quad (5.27)$$

Substituting $s = j\omega$, the magnitude and phase of the transfer function are given as

$$|G_o(j\omega)| = \left| \frac{400}{(j\omega)^3 + 10(j\omega)^2 + 100(j\omega)} \right| \quad (5.28)$$

$$= \frac{400}{|-10\omega^2 + j(100\omega - \omega^3)|} = \frac{400}{\sqrt{100\omega^4 + (100\omega - \omega^3)^2}}, \quad (5.29)$$

$$\varphi = \varphi(400) - \varphi(-10\omega^2 + j(100\omega - \omega^3)) = -\tan^{-1} \frac{100 - \omega^2}{10\omega}. \quad (5.30)$$

The gain margin is defined for the frequency ω_g when the phase is equal to $\varphi = -\pi$. This gives

$$-\pi = -\tan^{-1} \frac{100 - \omega_g^2}{10\omega_g} \Rightarrow \frac{100 - \omega_g^2}{10\omega_g} = \tan \pi \Rightarrow \omega_g = \sqrt{100} = 10. \quad (5.31)$$

The gain margin is given by the reciprocal of the magnitude of the transfer function at this frequency $1/|G_o(j\omega_g)$ and it is usually defined in decibels, hence

$$g_m = 20 \log_{10} \frac{1}{|G_o(j\omega_g)|} = -20 \log_{10} |G_o(j\omega_g)| = -20 \log_{10} 0.4 = 7.96 \text{ dB}. \quad (5.32)$$

The phase margin is defined at the frequency ω_p when $|G_o(j\omega_p)|$ is equal to one, that is,

$$\frac{400}{\sqrt{100\omega_p^4 + (100\omega_p - \omega_p^3)^2}} = 1. \quad (5.33)$$

Manipulating this equation gives a sixth-order equation (or the third order equation in the variable ω_p^2)

$$\omega_p^6 - 100\omega_p^4 + 100^2\omega_p^2 - 400^2 = 0. \quad (5.34)$$

Solving this equation gives a single real positive root $\omega_p = 4.35$ and the corresponding phase $\phi_{p_m} = 61.8^\circ$.

```
g = tf(20, [1 10 100]); gc = tf(20, [1 0]);
go = g * gc;
margin(go)
```

Example. 5.4: Determination of the closed-loop bandwidth

Let us again consider the process with the transfer function

$$G(s) = \frac{20}{s^2 + 10s + 100}, \quad (5.35)$$

controlled with an integral controller $G_c(s) = 20/s$. Determine the closed-loop bandwidth and estimate the rise time T_{50} .

The closed-loop transfer function is given as

$$G_{cl}(s) = \frac{G(s)G_c(s)}{1 + G(s)G_c(s)} = \frac{G_o(s)}{1 + G_o(s)} = \frac{400}{s^3 + 10s^2 + 100s + 400}. \quad (5.36)$$

The magnitude of this transfer function is given as

$$\begin{aligned} |G_{cl}(j\omega)| &= \left| \frac{400}{(j\omega)^3 + 10(j\omega)^2 + 100(j\omega) + 400} \right| \\ &= \frac{400}{|(400 - 10\omega^2) + j(100\omega - \omega^3)|} = \frac{400}{\sqrt{(400 - 10\omega^2)^2 + (100\omega - \omega^3)^2}}. \end{aligned} \quad (5.37)$$

The magnitude value at $s = 0$ is $|G_{cl}(0)| = 1$. To find the bandwidth frequency ω_b the magnitude should be 3 dB less than the initial one, *i.e.* -3 dB ($1/\sqrt{2}G(0) = 0.707 \approx -3$ dB)

$$\frac{1}{\sqrt{2}} = \frac{400}{\sqrt{(400 - 10\omega_b^2)^2 + (100\omega_b - \omega_b^3)^2}} \Rightarrow \omega_b = 9.8. \quad (5.39)$$

To estimate T_{50} , we will use the relationship $T_{50}\omega_b \approx 2.3$ which gives $T_{50} = 0.23$. This can be confirmed by plotting the closed-loop step response.

5.3 Practice Examples

5.3.1 Two Tanks in Series

We study a system with two tanks in series, for which we have derived the transfer function matrix in Section 2.3.2.

The derived transfer function between the input $u_1(t)$ (the deviation variable of the inflow $q_{0,1}(t)$ from its steady state) and the output $y(t)$ (the deviation variable of the liquid level in the second tank $h_2(t)$ from its steady state) is of the form:

$$G(s) = \frac{\overbrace{\frac{1}{k_2}}^K \overbrace{\frac{k_1 k_2}{F_1 F_2}}^{\omega_0^2}}{s^2 + 2 \underbrace{\frac{\frac{k_1}{F_1} + \frac{k_2}{F_2}}{2\sqrt{\frac{k_1 k_2}{F_1 F_2}}}}_{\zeta} \underbrace{\sqrt{\frac{k_1 k_2}{F_1 F_2}}}_{\omega_0} s + \underbrace{\frac{k_1 k_2}{F_1 F_2}}_{\omega_0^2}} = \frac{\frac{k_1}{F_1 F_2}}{s^2 + \left(\frac{k_1}{F_1} + \frac{k_2}{F_2}\right) s + \frac{k_1 k_2}{F_1 F_2}}. \quad (5.40)$$

We remind the reader, that the parameters of the two-tank-in-series system are given in Section 2.3.2, for convenience are presented again here:

```
q01s = 0.3; q02s = 0.5; k11 = 1.15; k22 = 1.3;
h1s = (q01s/k11)^2; h2s = (q01s/k22 + q02s/k22)^2;
k1 = k11/2/sqrt(h1s); k2 = k22/2/sqrt(h2s); F1 = 0.5; F2 = 0.8;
```

5.3.1.1 Consider the tracking problem. Formulate the closed-loop transfer function with a negative feedback and a proportional controller with a gain K_p .

$$G_{cl}(s) = \frac{K_p G(s)}{1 + K_p G(s)} = \frac{K_p \frac{\frac{k_1}{F_1 F_2}}{s^2 + \left(\frac{k_1}{F_1} + \frac{k_2}{F_2}\right) s + \frac{k_1 k_2}{F_1 F_2}}}{1 + K_p \frac{\frac{k_1}{F_1 F_2}}{s^2 + \left(\frac{k_1}{F_1} + \frac{k_2}{F_2}\right) s + \frac{k_1 k_2}{F_1 F_2}}} = \frac{K_p \frac{k_1}{F_1 F_2}}{s^2 + \left(\frac{k_1}{F_1} + \frac{k_2}{F_2}\right) s + \frac{k_1 k_2}{F_1 F_2} + K_p \frac{k_1}{F_1 F_2}}. \quad (5.41)$$

5.3.1.2 Re-arrange the transfer function obtained into the form

$$G_{cl}(s) = \frac{K_{cl} \omega_{0,cl}^2}{s^2 + 2\zeta_{cl} \omega_{0,cl} s + \omega_{0,cl}^2}, \quad (5.42)$$

and define ω_0 , ζ , and K in terms of the process parameters. Discuss the meaning of the process parameters for the dynamics of the system.

It follows that:

$$\omega_{0,cl}^2 = \frac{k_1 k_2}{F_1 F_2} + K_p \frac{k_1}{F_1 F_2}, \quad (5.43)$$

$$2\zeta_{cl}\omega_{0,cl} = \frac{k_1}{F_1} + \frac{k_2}{F_2} \Rightarrow 2\zeta_{cl}\sqrt{\frac{k_1 k_2}{F_1 F_2} + K_p \frac{k_1}{F_1 F_2}} = \frac{k_1}{F_1} + \frac{k_2}{F_2} \quad (5.44)$$

$$\Rightarrow \zeta_{cl} = \frac{\frac{k_1}{F_1} + \frac{k_2}{F_2}}{2\sqrt{\frac{k_1 k_2}{F_1 F_2} + K_p \frac{k_1}{F_1 F_2}}}, \quad (5.45)$$

$$K_{cl}\omega_0^2 = K_p \frac{k_1}{F_1 F_2} \Rightarrow K_{cl} \left(\frac{k_1 k_2}{F_1 F_2} + K_p \frac{k_1}{F_1 F_2} \right) = K_p \frac{k_1}{F_1 F_2} \quad (5.46)$$

$$\Rightarrow K_{cl} = \frac{K_p \frac{k_1}{F_1 F_2}}{\frac{k_1 k_2}{F_1 F_2} + K_p \frac{k_1}{F_1 F_2}} = \frac{K_p}{k_2 + K_p}. \quad (5.47)$$

For small values of K_p , the coefficient ζ_{cl} will be greater than one and the steady-state gain K_{cl} will be smaller than one, so the closed-loop step response will be aperiodic and the output will not reach the reference value. On the other hand, for larger values of K_p the value of ζ_{cl} will be positive and less than one and the steady-state gain K_{cl} will approach one, so the closed-loop response will be periodic but the steady-state control error will be smaller.

5.3.1.3 Verify the observations from simulations. Consider a system with the valve constants $k_1 = k_2 = 1 \text{ m}^2/\text{s}$ and the cross-sectional areas $F_1 = 5 \text{ m}^2$ and $F_2 = 1 \text{ m}^2$, for the first and the second tank, respectively. Compare the performance under different controllers by plotting the closed-loop step responses, the closed-loop poles, and evaluating the following performance criteria: maximum overshoot, rise time, $T_{0.05}$, and the bandwidth frequency.

Tune the controllers such that:

- $\zeta_{cl} = 1.2$,
- $\zeta_{cl} = 1$,
- we have a steady-state error of 10%,
- we have a steady-state error of 1%,
- we have a maximum overshoot of 25%.

Discuss the results obtained.

Maximum overshoot:

$$e_{\max} = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}}. \quad (5.48)$$

Rise time:

$$T_{100} = \frac{1}{\omega_0 P} \left[\pi - \tan^{-1} \left(\frac{P}{\zeta} \right) \right]. \quad (5.49)$$

Settling time:

$$T_\epsilon \approx \frac{1}{\zeta\omega_0} \ln \frac{1}{\epsilon P}, \quad (5.50)$$

$$T_{0.05} \approx \frac{3}{\zeta\omega_0}. \quad (5.51)$$

Bandwidth frequency ω_b (the larger ω_b means the faster response)

$$\omega_b = \omega_0 \sqrt{1 - 2\zeta^2 + \sqrt{1 + (1 - 2\zeta^2)^2}}. \quad (5.52)$$

We obtain from (5.45):

$$K_p = \frac{\left(\frac{k_1 + k_2}{F_1 + F_2}\right)^2 - \frac{k_1 k_2}{F_1 F_2}}{\frac{k_1}{F_1 F_2}}, \quad (5.53)$$

which gives the controller gain that corresponds to the chosen ζ_{cl} .

Using this:

a) $K_p = 0.25,$

b) $K_p = 0.8.$

Results are shown in Figure 5.5.

We can obtain from (5.47):

$$K_{cl} \left(\frac{k_1 k_2}{F_1 F_2} + K_p \frac{k_1}{F_1 F_2} \right) = K_p \frac{k_1}{F_1 F_2} \Rightarrow K_p = \frac{K_{cl} \frac{k_1 k_2}{F_1 F_2}}{(1 - K_{cl}) \frac{k_1}{F_1 F_2}}, \quad (5.54)$$

which gives the controller gain that corresponds to the chosen K_{cl} .

Using this for $K_{cl} = 0.9$ and $K_{cl} = 0.99$ we get:

c) $K_p = 9,$

d) $K_p = 99.$

Results are shown in Figure 5.6.

Damping ratio ζ_{cl} is obtained from (3.55)

$$\zeta_{cl} = \frac{|\ln e_{\max}|}{\sqrt{\pi^2 + \ln^2 e_{\max}}}, \quad (5.55)$$

which gives the controller gain that corresponds to the chosen K_{cl} .

Using this we get:

e) $\zeta = 0.4037, K_p = 10.044.$ Results are shown in Figure 5.7.

K_p	maximum overshoot	rise time	$T_{0.05}$	bandwidth frequency
0.25	0	∞	12.3	0.25
0.80	0	∞	7.80	0.39
9.00	23%	1.57	5.16	1.91
99.00	65%	0.38	4.50	6.86
10.04	25%	1.46	5.14	2.04

```

k1 = 1; k2 = 1; F1 = 5; F2 = 1;

xi = 1.2;
Kp = ((k1/F1+k2/F2)/2/xi)^2 - k1*k2/F1/F2)/(k1/F1/F2)
G = tf(Kp*k1/F1/F2, [1 (k1/F1+k2/F2) k1*k2/F1/F2+Kp*k1/F1/F2]);
step(G); grid on
% pause
save_pdf(4, 4, 20, 'plt_xi12.pdf');
!pdfcrop plt_xi12.pdf plt_xi12.pdf

w0 = sqrt(k1*k2/F1/F2 + Kp*k1/F1/F2);
emax = exp(-pi*xi/sqrt(1-xi^2))
T100 = 1/w0/sqrt(1-xi^2)*(pi - atan(sqrt(1-xi^2)/xi))
T005 = 1/xi/w0*log(1/0.05/sqrt(1-xi^2))
T005 = 3/xi/w0
wb = w0*sqrt(1-2*xi^2+sqrt(1+(1-2*xi^2)^2))
bode(G)

xi = 1;
Kp = ((k1/F1+k2/F2)/2/xi)^2 - k1*k2/F1/F2)/(k1/F1/F2)
G = tf(Kp*k1/F1/F2, [1 (k1/F1+k2/F2) k1*k2/F1/F2+Kp*k1/F1/F2]);
step(G); grid on
% pause
save_pdf(4, 4, 20, 'plt_xi1.pdf');
!pdfcrop plt_xi1.pdf plt_xi1.pdf

w0 = sqrt(k1*k2/F1/F2 + Kp*k1/F1/F2);
emax = exp(-pi*xi/sqrt(1-xi^2))
T100 = 1/w0/sqrt(1-xi^2)*(pi - atan(sqrt(1-xi^2)/xi))
T005 = 1/xi/w0*log(1/0.05/sqrt(1-xi^2))
T005 = 3/xi/w0
wb = w0*sqrt(1-2*xi^2+sqrt(1+(1-2*xi^2)^2))
bode(G)

Kc1 = 0.9;
Kp = Kc1*k1*k2/F1/F2/(1-Kc1)/(k1/F1/F2)
G = tf(Kp*k1/F1/F2, [1 (k1/F1+k2/F2) k1*k2/F1/F2+Kp*k1/F1/F2]);
step(G); grid on
% pause
save_pdf(4, 4, 20, 'plt_kc109.pdf');
!pdfcrop plt_kc109.pdf plt_kc109.pdf

w0 = sqrt(k1*k2/F1/F2 + Kp*k1/F1/F2);
xi = (k1/F1 + k2/F2)/2/sqrt(k1/F1*k2/F2 + Kp*k1/F1/F2)
emax = exp(-pi*xi/sqrt(1-xi^2))
T100 = 1/w0/sqrt(1-xi^2)*(pi - atan(sqrt(1-xi^2)/xi))
T005 = 1/xi/w0*log(1/0.05/sqrt(1-xi^2))
T005 = 3/xi/w0
wb = w0*sqrt(1-2*xi^2+sqrt(1+(1-2*xi^2)^2))
bode(G)

Kc1 = 0.99;
Kp = Kc1*k1*k2/F1/F2/(1-Kc1)/(k1/F1/F2)
G = tf(Kp*k1/F1/F2, [1 (k1/F1+k2/F2) k1*k2/F1/F2+Kp*k1/F1/F2]);
step(G); grid on
% pause
save_pdf(4, 4, 20, 'plt_kc1099.pdf');
!pdfcrop plt_kc1099.pdf plt_kc1099.pdf

w0 = sqrt(k1*k2/F1/F2 + Kp*k1/F1/F2);

```

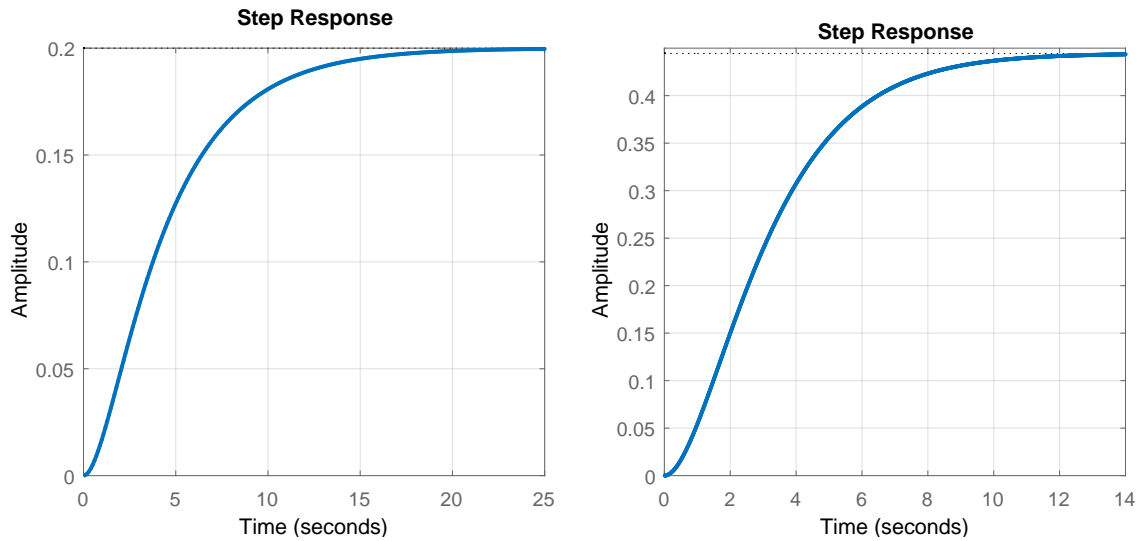


Figure 5.5: (left) Closed-loop step responses with $\zeta_{cl} = 1.2$ and (right) $\zeta_{cl} = 1$.

```

xi = (k1/F1 + k2/F2)/2/sqrt(k1/F1*k2/F2 + Kp*k1/F1/F2)
emax = exp(-pi*xi/sqrt(1-xi^2))
T100 = 1/w0/sqrt(1-xi^2)*(pi - atan(sqrt(1-xi^2)/xi))
T005 = 1/xi/w0*log(1/0.05/sqrt(1-xi^2))
T005 = 3/xi/w0
wb = w0*sqrt(1-2*xi^2+sqrt(1+(1-2*xi^2)^2))
bode(G)

emax = 0.25;
xi = abs(log(emax))/sqrt(pi^2 + log(emax)^2)
Kp = ((k1/F1+k2/F2)/2/xi)^2 - k1*k2/F1/F2/(k1/F1/F2)
G = tf(Kp*k1/F1/F2, [1 (k1/F1+k2/F2) k1*k2/F1/F2+Kp*k1/F1/F2]);
step(G); grid on
% pause
save_pdf(4, 4, 20, 'plt_over025.pdf');
!pdfcrop plt_over025.pdf plt_over025.pdf

w0 = sqrt(k1*k2/F1/F2 + Kp*k1/F1/F2);
xi = (k1/F1 + k2/F2)/2/sqrt(k1/F1*k2/F2 + Kp*k1/F1/F2)
emax = exp(-pi*xi/sqrt(1-xi^2))
T100 = 1/w0/sqrt(1-xi^2)*(pi - atan(sqrt(1-xi^2)/xi))
T005 = 1/xi/w0*log(1/0.05/sqrt(1-xi^2))
T005 = 3/xi/w0
wb = w0*sqrt(1-2*xi^2+sqrt(1+(1-2*xi^2)^2))

```

5.3.1.4 (Optional) Evaluate the ISE and IAE indices under a controller that guarantees damping ratio of the closed-loop response $\zeta_{cl} = 1.2$ and under the controller that gives a steady-state error of 1%.

5.3.1.5 Consider the disturbance-rejection problem. Plot the closed-loop response to the unit step in the disturbance $u_2(t)$, which is the deviation variable between the inflow into the second tank and the steady-state value of the inflow q_2^s . Compare the closed-loop responses of the systems under the controllers from the previous question.

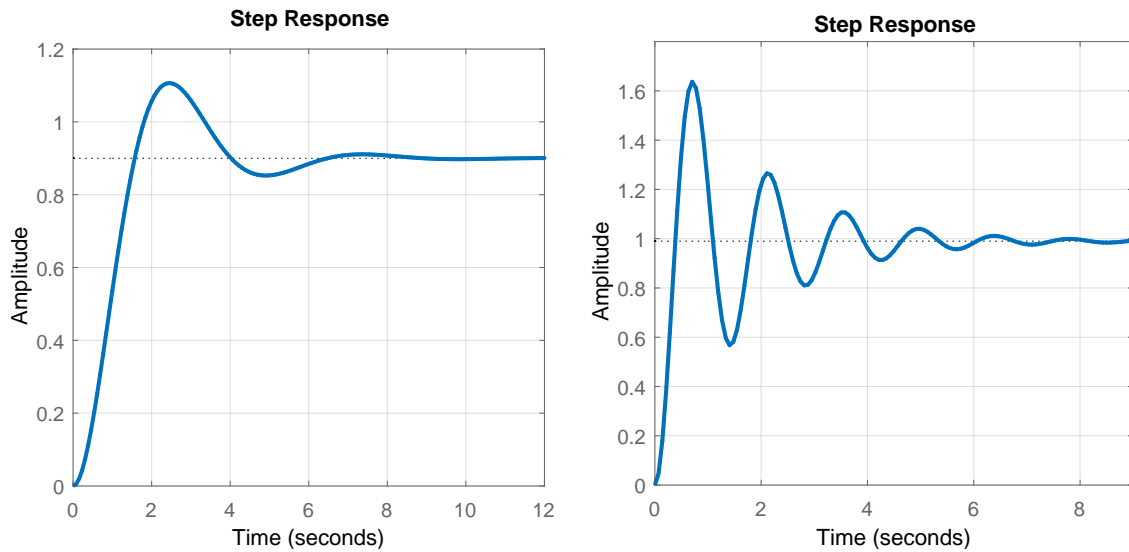


Figure 5.6: (left) Closed-loop step responses with $K_{cl} = 0.9$ and (right) $K_{cl} = 0.99$.

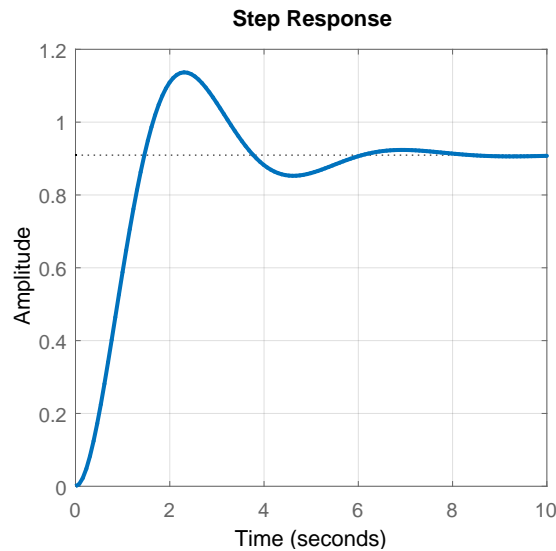


Figure 5.7: Closed-loop step response with $K_p = 10.04$ for a 25% overshoot.

$$G_{cl,d}(s) = \frac{G_d(s)}{1 + K_p G} = \frac{\frac{1}{F_2}}{s + \frac{k_2}{F_2}} \frac{k_1}{F_1 F_2} \frac{1}{s^2 + \left(\frac{k_1}{F_1} + \frac{k_2}{F_2}\right)s + \frac{k_1 k_2}{F_1 F_2}} \quad (5.56)$$

$$= \frac{\frac{1}{F_2} \left(s + \frac{k_1}{F_1}\right)}{s^2 + \left(\frac{k_1}{F_1} + \frac{k_2}{F_2}\right)s + \frac{k_1 k_2}{F_1 F_2} + K_p \frac{k_1}{F_1 F_2}} \quad (5.57)$$

```
xi = 1.2;
Kp = ((k1/F1+k2/F2)/2/xi)^2 - k1*k2/F1/F2)/(k1/F1/F2);
G = tf([1/F2 k1/F1/F2], [1 (k1/F1+k2/F2) k1*k2/F1/F2+Kp*k1/F1/F2]);
step(G)
```

```

save_pdf(4, 4, 20, 'plt_dist_xi12.pdf');
!pdfcrop plt_dist_xi12.pdf plt_dist_xi12.pdf

xi = 1;
Kp = ((k1/F1+k2/F2)/2/xi)^2 - k1*k2/F1/F2)/(k1/F1/F2);
G = tf([1/F2 k1/F1/F2], [1 (k1/F1+k2/F2) k1*k2/F1/F2+Kp*k1/F1/F2]);
step(G)
save_pdf(4, 4, 20, 'plt_dist_xi1.pdf');
!pdfcrop plt_dist_xi1.pdf plt_dist_xi1.pdf

Kc1 = 0.9;
Kp = Kc1*k1*k2/F1/F2/(1-Kc1)/(k1/F1/F2);
G = tf([1/F2 k1/F1/F2], [1 (k1/F1+k2/F2) k1*k2/F1/F2+Kp*k1/F1/F2]);
step(G)
save_pdf(4, 4, 20, 'plt_dist_kc109.pdf');
!pdfcrop plt_dist_kc109.pdf plt_dist_kc109.pdf

Kc1 = 0.99;
Kp = Kc1*k1*k2/F1/F2/(1-Kc1)/(k1/F1/F2);
G = tf([1/F2 k1/F1/F2], [1 (k1/F1+k2/F2) k1*k2/F1/F2+Kp*k1/F1/F2]);
step(G)
save_pdf(4, 4, 20, 'plt_dist_kc1099.pdf');
!pdfcrop plt_dist_kc1099.pdf plt_dist_kc1099.pdf

emax = 0.25;
xi = abs(log(emax))/sqrt(pi^2 + log(emax)^2);
Kp = ((k1/F1+k2/F2)/2/xi)^2 - k1*k2/F1/F2)/(k1/F1/F2);
G = tf([1/F2 k1/F1/F2], [1 (k1/F1+k2/F2) k1*k2/F1/F2+Kp*k1/F1/F2]);
step(G)
save_pdf(4, 4, 20, 'plt_dist_over025.pdf');
!pdfcrop plt_dist_over025.pdf plt_dist_over025.pdf

```

Results are shown in Figure 5.8.

5.3.1.6 Consider the full nonlinear model of the process ($q_{0,1}^s = 0.3 \text{ m}^3/\text{s}$, $q_{0,2}^s = 0.5 \text{ m}^3/\text{s}$, $F_1 = 0.5 \text{ m}^2$, $F_2 = 0.8 \text{ m}^2$, $k_{11} = 1.15 \text{ m}^{2.5}/\text{s}$, and $k_{22} = 1.3 \text{ m}^{2.5}/\text{s}$). Implement proportional feedback control of the level h_2 with feed flow $q_{0,1}$ and unmeasurable disturbance $q_{0,2}$. The desired value of $h_2(t)$ can be described by

$$h_2^r(t) = \begin{cases} h_2^s, & t \leq 1, \\ 1.1h_2^s, & t \in (1, 10], \\ 0.8h_2^s, & t \in (10, 30]. \end{cases} \quad (5.58)$$

The disturbance trajectory is

$$q_{0,2}(t) = \begin{cases} q_{0,2}^s, & t \leq 20, \\ 0.9q_{0,2}^s, & t \in (20, 30]. \end{cases} \quad (5.59)$$

The manipulated variable is constrained by $0.05 \leq q_{0,1}(t) \leq 1$. Choose controllers b), c), d) and compare the results with the results obtained in the previous steps.

The simulation model is shown in Figure 5.9.

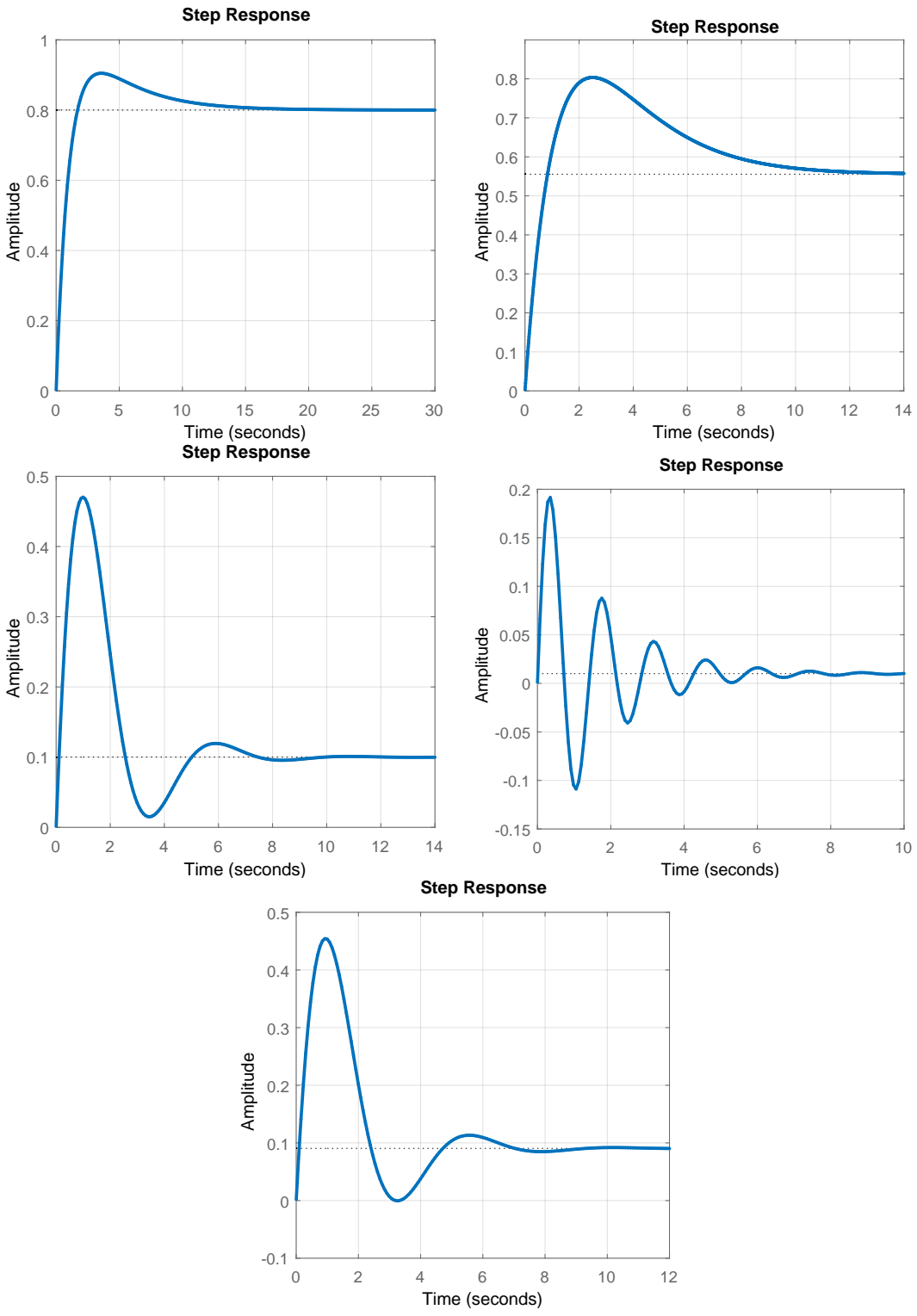


Figure 5.8: Closed-loop disturbance step responses.

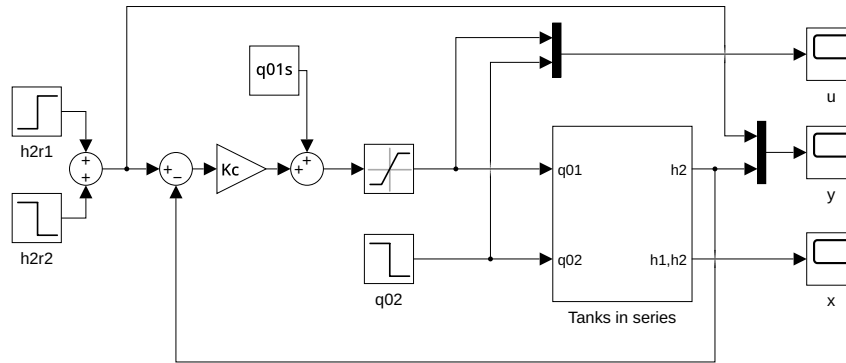


Figure 5.9: A model of feedback control loop with a proportional controller of the two-tanks-in-series process.

5.4 Problems

Problem. 5.1: Time-domain performance indices for a higher-order system

Examine the step response of the system with the transfer function

$$G(s) = \frac{0.5s^2 + 0.6s + 6}{s^4 + 4s^3 + 6s^2 + 8s + 6},$$

using MATLAB and determine the rise time, peak time, maximum overshoot, and settling time.

Optionally, find the dominant complex conjugate poles and the approximate time-domain indices using the second-order-system formulae.

$$T_{100} = 2.29, T_{\sigma} = 3.25, e_{\max} = 33.4\%, T_{0.02} = 24.19.$$

5.5 MATLAB Snippets

- Time-domain specifications (rise time, settling time, peak time, ...)

```
g = tf(1, [1 1 1])
stepinfo(g)
stepinfo(g, 'RiseTimeLimits', [0,1])
```

- Calculate closed-loop bandwidth ω_b , resonant peak, and frequency

```
gcl = tf(400, [1 10 100 400])
omb = bandwidth(gcl)
[gpeak, fpeak] = getPeakGain(gcl), gpeak_dB = 20 * log10(gpeak)
```

- Calculate the integral indices using numerical integration

```
gcl = tf(400, [1 10 100 400])
[y, t] = step(gcl); r = ones(size(t)); plot(t, r, t, y)
e = r - y;
IAE = trapz(t, abs(e))
ISE = trapz(t, e.^2)
ITAE = trapz(t, t.*abs(e))
ITSE = trapz(t, t.*e.^2)
```

- Find the time performance indices for Problem 5.1

```

g = tf([0.5 0.6 6], [1 4 6 8 6])
[y, t] = step(g);
plot(t, y)
grid
i = find(y >= 1); % if needed, improve using interpolation here
i100 = i(1); T100 = t(i100)
[m,i] = max(y);
sigma = (y(i) - 1) / (1 - y(1))
isigma = i; Tsigma = t(isigma)
epsilon = 0.02;
i = find(abs(y - 1) < epsilon);
ii = find(ishange(i, 'linear'));
i002 = i(ii(end)); T002 = t(i002)
hold
plot([t(1) t(end)], (1 + epsilon)*ones(1, 2), '--', [t(1) t(end)], (1 - epsilon)
    )*ones(1, 2), '--')
plot(T100, y(i100), '*', Tsigma, y(isigma), '*', T002, y(i002), '*')

```

5.6 Python Snippets

- Calculate closed-loop bandwidth ω_b , resonant peak, and frequency

```

from control.matlab import *
import matplotlib.pyplot as plt
import numpy as np
import math
gcl = tf(400, [1, 10, 100, 400])
m, p, om = bode(g)
i = np.where(m < m[0] / math.sqrt(2))
omb = om[i[0][0]]
ii = np.argmax(m); gpeak = m[ii]; fpeak = om[ii]
gpeak_dB = 20 * math.log10(gpeak)

```

- Calculate the integral indices using numerical integration

```

from control.matlab import *
import numpy as np
import matplotlib.pyplot as plt
gcl = tf(400, [1, 10, 100, 400])
y, t = step(gcl); r = np.ones(len(t))
plt.plot(t, r, t, y); plt.show()
e = w - y
IAE = np.trapz(abs(e), t)
ISE = np.trapz(e**2, t)
ITAE = np.trapz(t * abs(e), t)
ITSE = np.trapz(t * e**2, t)

```

- Find the time performance indices for Problem 5.1

```

from control.matlab import *
import matplotlib.pyplot as plt
import numpy as np
import math
g = tf([0.5, 0.6, 6], [1, 4, 6, 8, 6])
y, t = step(g)
i = np.where(y >= 1) # when needed, improve using interpolation here
i100 = i[0][0]; T100 = t[i100]
i = np.argmax(y); m = y[i]
sigma = (m - 1) / (1 - y[0])
isigma = i; Tsigma = t[isigma]
epsilon = 0.02

```

```
i = np.where (abs(y - 1) < epsilon)
ii = i[0]; iii = np.where(np.diff(ii) - 1 > 0)
iiii = iii[0][-1] # last nonzero difference
i002 = ii[iiiii + 1] - 1 # index of last y outside
T002 = t[i002]
plt.plot(t,y)
plt.plot([t[0], t[-1]], [1 + epsilon, 1 + epsilon], 'k--')
plt.plot([t[0], t[-1]], [1 - epsilon, 1 - epsilon], 'k--')
plt.plot(T100, y[i100], 'r*', Tsigma, y[isigma], 'r*', T002, y[i002], 'r*')
plt.grid()
plt.show()
```

5.7 Bibliography

Control performance can be specified using various methods. The theoretical background for this chapter can be found in Mikleš and Fikar (2007, Chapter 7), information in Slovak in Bakošová and Fikar (2008, Chapter 5), Mikleš and Hutla (1986, Chapter 2).

For supplemental information, see for example Unbehauen (1986, Chapter 8), Golnaraghi (2010, Chapter 9), or Dorf and Bishop (2008, Chapter 5).

CHAPTER 6

Closed-Loop Systems and PID Controllers

This chapter deals with the properties of closed-loop control systems and reviews the PID controller, its parameters and properties. It is assumed that students already understand stability, e.g., are acquainted with stability criteria for polynomials. In addition to stability, disturbance rejection and setpoint tracking problems are introduced. Different PID controller structures are discussed, together with anti-windup strategies and bumpless transfer between controllers.

6.1 Overview

6.1.1 A Closed-Loop System

A physical process is controlled by a feedback controller and constitutes a closed-loop, feedback system. This system consists of the following components: a controlled process, a measurement device (a sensor), a controller, and an actuator (a final control element). In addition, there is a block that compares the desired reference (setpoint) value $r(t)$ with the measured value of the process output $y_m(t)$. Figure 6.1(a) shows the corresponding block diagram with the individual components represented by transfer functions. The presence of an external signal $d(t)$ which acts as a disturbance is assumed with its influence specified by a transfer function $G_d(s)$.

We can derive an expression for the process output $y(t)$ as a function of the inputs $r(t)$ and $d(t)$ using block algebra

$$Y(s) = \frac{G_p(s)G_a(s)G_c(s)}{1 + G_m(s)G_p(s)G_a(s)G_c(s)}R(s) + \frac{G_d(s)}{1 + G_m(s)G_p(s)G_a(s)G_c(s)}D(s) \quad (6.1)$$

$$= G_{yr}(s)R(s) + G_{yd}(s)D(s), \quad (6.2)$$

$$G_{yr}(s) = \frac{G_p(s)G_a(s)G_c(s)}{1 + G_m(s)G_p(s)G_a(s)G_c(s)}, \quad (6.3)$$

$$G_{yd}(s) = \frac{G_d(s)}{1 + G_m(s)G_p(s)G_a(s)G_c(s)}. \quad (6.4)$$

The common denominator of both transfer functions is the characteristic equation of the closed-

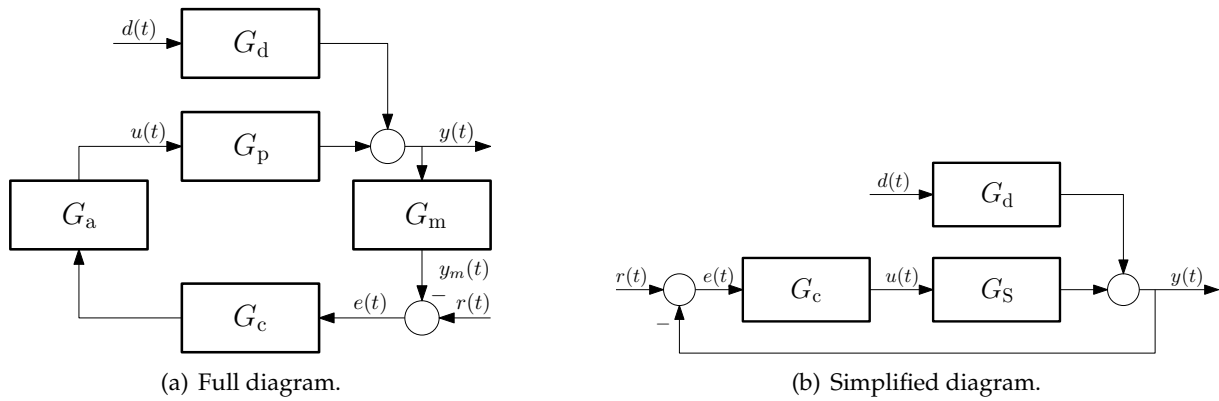


Figure 6.1: A block diagram of the closed-loop system.

loop system

$$1 + G_m(s)G_p(s)G_a(s)G_c(s) = 0. \quad (6.5)$$

Three basic closed-loop features are:

1. Stability – roots of the characteristic equation should have negative real parts.
2. Tracking – process output follows the setpoint, thus $G_{yr}(0) \approx 1$.
3. Regulation – process output should be insensitive to disturbances, thus $G_{yd}(0) \approx 0$.

Quite often, we group the process itself together with the measurement device and the final control element. This can be represented by a simplified block diagram shown in Figure 6.1 (right). In this case, expressions for the process output and the closed-loop transfer functions become

$$Y(s) = \frac{G_S(s)G_c(s)}{1 + G_S(s)G_c(s)}R(s) + \frac{G_d(s)}{1 + G_S(s)G_c(s)}D(s) = G_{yr}(s)R(s) + G_{yd}(s)D(s), \quad (6.6)$$

$$G_{yr}(s) = \frac{G_S(s)G_c(s)}{1 + G_S(s)G_c(s)}, \quad G_{yd}(s) = \frac{G_d(s)}{1 + G_S(s)G_c(s)}. \quad (6.7)$$

The common denominator of both transfer function is the closed-loop characteristic equation

$$1 + G_S(s)G_c(s) = 0. \quad (6.8)$$

6.1.2 PID Controller

Proportional Control

The output signal of this controller is proportional to the actual control error, $u(t) = K_p e(t)$ and its transfer function is thus given as

$$G_c(s) = K_p, \quad (6.9)$$

where K_p is called the proportional gain. As the transfer function is static, the controller does not change the number of closed-loop poles, but it can influence their positions.

As the control signal can only operate between its minimum u_{\min} and maximum u_{\max} values, we can define a proportional band P_p as the range of outputs where $u(t)$ is between these limits. Usually $u_{\min} = 0$ and $u_{\max} = 100\%$, which gives

$$u_{\max} = K_p e_{\max}, \quad u_{\min} = K_p e_{\min}, \quad P_p = e_{\max} - e_{\min}, \quad P_p = \frac{100\%}{K_p}. \quad (6.10)$$

Controllers with a small proportionality band act almost as on/off controllers.

Example 6.2 derives expressions for steady-state error for a first-order system. We can see that, both for tracking and regulation, there is a nonzero steady-state error that depends on the value of controller gain, which is reduced for large values of K_p . This is a common issue of proportional controller. On the other hand, a faster controller can also bring the closed-loop system to instability, as will be shown later.

To reduce the steady-state control error for a particular setpoint, a bias correction u_b can be used:

$$u(t) = K_p e(t) + u_b. \quad (6.11)$$

Proportional plus Integral Control

One possibility to deal with the issue of a steady-state tracking error under a proportional control is to make the bias u_b variable with time and to follow the actual control signal $u(t)$ via a first-order system with a time constant T_i :

$$u(t) = K_p e(t) + u_b(t), \quad u_b(t) = \frac{1}{T_i s + 1} u(t). \quad (6.12)$$

Elimination of $u_b(t)$ results in a PI (proportional + integral) controller with the transfer function and control law

$$G_c(s) = \frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} \right), \quad (6.13)$$

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) d\tau. \quad (6.14)$$

The additional parameter T_i is the integral time constant.

The PI controller increases the order of the closed-loop system (adds one pole) and also adds one stable zero that can influence the performance. In addition, the controller pole at $s = 0$ eliminates the steady-state error as shown in Example 6.3.

Example 6.1: Need for PI controller

Consider a first-order system with static gain K and time constant T . We would like to design a controller that perfectly tracks the setpoint (*i.e.*, the closed-loop gain equals to one), such that the closed-loop dynamics corresponds to a first-order system with a time constant $T_{cl} < T$.

The tracking-problem transfer function should be of the form

$$G_{yr}(s) = \frac{1}{T_{cl}s + 1} = \frac{G_S(s)G_c(s)}{1 + G_S(s)G_c(s)}. \quad (6.15)$$

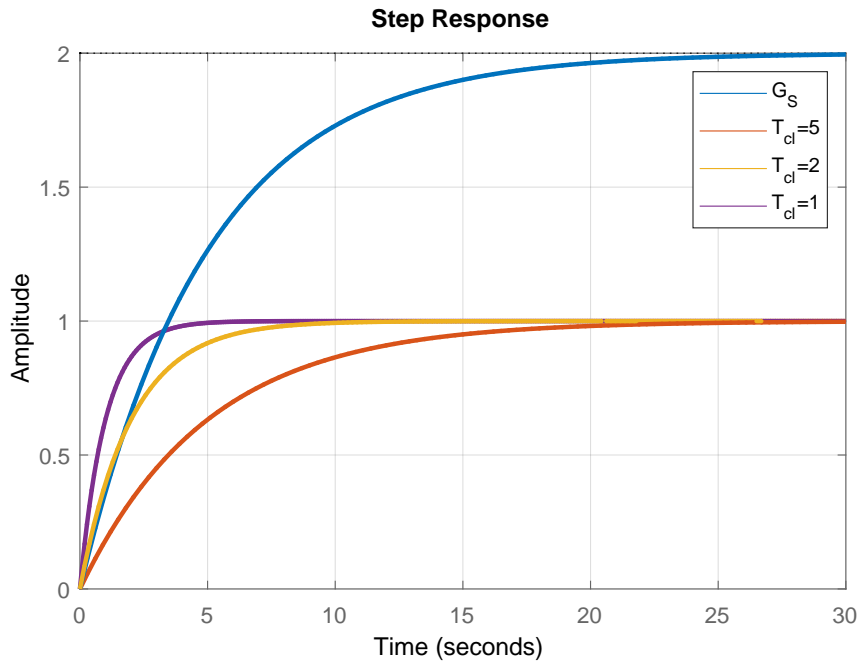


Figure 6.2: The first-order system ($K = 2, T = 5$) closed-loop control using the PI controller.

The controller transfer function is then

$$G_c(s) = \frac{T}{T_{cl}K} \left(1 + \frac{1}{T_s} \right), \quad (6.16)$$

which is a PI controller. The derived transfer function also suggests suitable values of controller parameters: $T_i = T$ and $K_p = k/K$ where $k = T/T_{cl} > 1$ to reduce the settling time. Thus, the closed-loop system will have the same settling time as the controlled process if $T_{cl} = T$ and $K_p = 1/K$; will be twice faster if $T_{cl} = T/2$ and $K_p = 2/K$; and five times faster if $T_{cl} = T/5$ and $K_p = 5/K$. Simulation results together with the step response of the uncompensated system G_S are shown in Figure 6.2.

Proportional plus Derivative Control

Another enhancement of the pure proportional control is to act not only on the control error but also on its rate of change. The controller then anticipates the future behaviour of the control error. If there is no change in the control error, the derivative controller does not act. Therefore, it is never used alone but at least with a proportional controller:

$$u(t) = K_p \left(e(t) + T_d \frac{de(t)}{dt} \right), \quad G_c(s) = K_p(1 + T_d s). \quad (6.17)$$

Here, T_d is a derivative time constant of the controller serving as a prediction of the control error $e(t + T_d)$. The PD controller thus realises the anticipatory action.

An ideal PD controller is not feasible as the controller output at time t cannot anticipate exactly the future behaviour ($e(t + T_d)$) as it would be a noncausal system. The derivative part can be delayed by filtering, e.g., using a first-order filter. This gives a causal controller transfer function

of the form

$$G_c(s) = K_p \left(1 + \frac{T_d s}{1 + \frac{T_d}{N} s} \right), \quad (6.18)$$

with $N \in [5, 20]$.

The derivative part of the PD controller has no effect on the steady-state error. The controller adds one stable zero and one stable pole to the closed-loop thus increasing its order. It can improve the speed of control as well.

Proportional plus Integral plus Derivative Control

The complete controller is known as PID controller and it is used in more than 90% of all control loops. It adds together all three parts. Each of them looks at a different time window and treats the tracking error in a different way: proportional acts on the present (value), integral acts on the (sum of the) past (values), and derivative on the future (predicted values).

PID Structures

There are several basic PID controller structures combining proportional, integral, and derivative actions:

- without interaction (ideal, standard PID)

$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right), \quad (6.19)$$

- with interaction (serial form)

$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} \right) (1 + T_d s), \quad (6.20)$$

- parallel form

$$G_c(s) = K_p + \frac{1}{T_i s} + T_d s. \quad (6.21)$$

Modifications of the standard PID controller include separate actions for setpoints and disturbances, leading to controllers with additional parameters and two-degree-of-freedom (2DoF) structure:

- PI-D: This structure can avoid a phenomena called “derivative kick”. It occurs due to instantaneous change of the tracking error caused by the setpoint adjustment. This results in a large value of the derivative part and sudden saturation of the controller output. As a remedy, it is assumed that the setpoint is constant, and therefore $de/dt = -dy/dt$:

$$U(s) = K_p \left(1 + \frac{1}{T_i s} \right) R(s) - K_p \left(1 + \frac{1}{T_i s} + T_d s \right) Y(s). \quad (6.22)$$

- I-PD: In addition to the elimination of the derivative kick, a sudden change of the controller output is avoided also for the proportional part (setpoint kick). The integral part cannot be turned off.

$$U(s) = K_p \frac{1}{T_i s} R(s) - K_p \left(1 + \frac{1}{T_i s} + T_d s \right) Y(s). \quad (6.23)$$

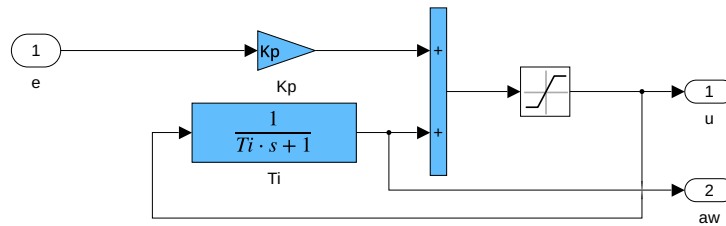


Figure 6.3: An anti-windup model AW1 with PI controller.

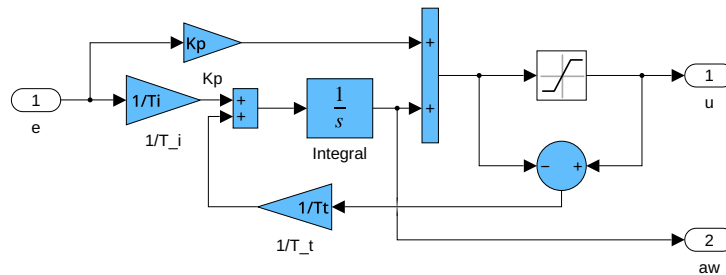


Figure 6.4: An anti-windup model with back-integration (AW2).

- 2DoF PID: $b, c < 1, (b \in (0.3, 0.8))$

$$U(s) = K_p \left(b + \frac{1}{T_i s} + c T_d s \right) R(s) - K_p \left(1 + \frac{1}{T_i s} + T_d s \right) Y(s). \quad (6.24)$$

This is a generalisation of PI-D and I-PD controllers.

Practical Issues

If controller with integral action saturates at its minimum or maximum output due to a large setpoint change or disturbance, a phenomenon called integral windup occurs. The controller is unaware of the fact that the calculated value of plant input was not applied to the process and, instead, its saturated value was used, and keeps increasing the integral part of the control signal. Afterwards, when eventually the controlled output reaches the setpoint and the sign of the control error reverses, the integral action is reduced only slowly. This means that the controller still remains on the saturated value even if it should not. Only once the integral term becomes sufficiently small (its absolute value), does the control signal not exhibit saturation anymore.

There are several possibilities to deal with windup. The first one (AW1) uses the original PI configuration (6.12), where the bias u_b acts on the saturated control signal and is implemented in Figure 6.3

$$u(t) = K_p e(t) + u_b(t), \quad u_b(t) = \frac{1}{T_i s + 1} u_{\text{SAT}}(t). \quad (6.25)$$

The second option (AW2) modifies the control error signal entering the integral term and back-integrates the difference between the calculated and the saturated control signals (Figure 6.4). This configuration involves an additional tuning parameter T_t (or a gain $1/T_t$) which is usually chosen similar to T_i or as the geometric mean of T_i and T_d , i.e., $\sqrt{T_i T_d}$.

The second solution introduces a PID controller feature called external feedback. Here, the input signals to the controller do not include only the setpoint and the controlled output but also the actual value of the manipulated variable applied. This is useful in situations where multiple

controllers are used, for example manual/automatic or heating/cooling. The transfer between controllers should be smooth and an external feedback can be used for this purpose. Therefore, the inactive controller uses the output of the active controller as its external signal and follows it. If this controller is activated, its output is the same as the actual one. Such a switch of controllers is called "bumpless transfer".

6.2 Examples

Example. 6.2: Control error for a first-order system using a P controller

Consider a first-order system with a static gain K and a time constant T controlled by a proportional controller with a gain K_p . Assume that the disturbance transfer function is also first order with a gain K_d and a time constant T_{pd}

$$G_S = \frac{K}{Ts + 1}, \quad G_c = K_p, \quad G_d = \frac{K_d}{T_{pd}s + 1}. \quad (6.26)$$

Derive the expressions for the steady-state error for the tracking and regulation problems if there be a step change in either the setpoint $r(t) = A$ or the disturbance $d(t) = A$.

Disturbance rejection problem: We consider $d(t) = A$ and $r(t) = 0$. The steady-state error is given as $e(\infty) = r(\infty) - y(\infty)$. We apply the Final Value Theorem

$$y(\infty) = \lim_{s \rightarrow 0} sY(s), \quad Y(s) = G_{yd}(s)D(s), \quad (6.27)$$

$$G_{yd}(s) = \frac{G_d(s)}{1 + G_S(s)G_c(s)} = \frac{\frac{K_d}{T_{pd}s + 1}}{1 + \frac{K}{Ts + 1}K_p} = \frac{K_d(Ts + 1)}{(T_{pd}s + 1)(Ts + 1 + KK_p)}, \quad (6.28)$$

$$y(\infty) = \lim_{s \rightarrow 0} s \frac{K_d(Ts + 1)}{(T_{pd}s + 1)(Ts + 1 + KK_p)} \frac{A}{s} = \frac{K_d}{1 + KK_p} A \neq 0, \quad (6.29)$$

$$e(\infty) = r(\infty) - y(\infty) = 0 - \frac{K_d}{1 + KK_p} A = -\frac{K_d}{1 + KK_p} A. \quad (6.30)$$

Setpoint tracking problem: We consider $d(t) = 0$ and $r(t) = A$. It follows that

$$y(\infty) = \lim_{s \rightarrow 0} sY(s), \quad Y(s) = G_{yr}(s)R(s), \quad (6.31)$$

$$G_{yr}(s) = \frac{G_S(s)G_c(s)}{1 + G_S(s)G_c(s)} = \frac{\frac{K}{Ts + 1}K_p}{1 + \frac{K}{Ts + 1}K_p} = \frac{KK_p}{Ts + 1 + KK_p}, \quad (6.32)$$

$$y(\infty) = \lim_{s \rightarrow 0} s \frac{KK_p}{Ts + 1 + KK_p} \frac{A}{s} = \frac{KK_p}{1 + KK_p} A < A, \quad (6.33)$$

$$e(\infty) = r(\infty) - y(\infty) = A - \frac{KK_p}{1 + KK_p} A = \frac{1}{1 + KK_p} A. \quad (6.34)$$

Simulation results for both cases are shown in Figure 6.5. Increasing K_p reduces the control error and the closed-loop time constant (settling time).

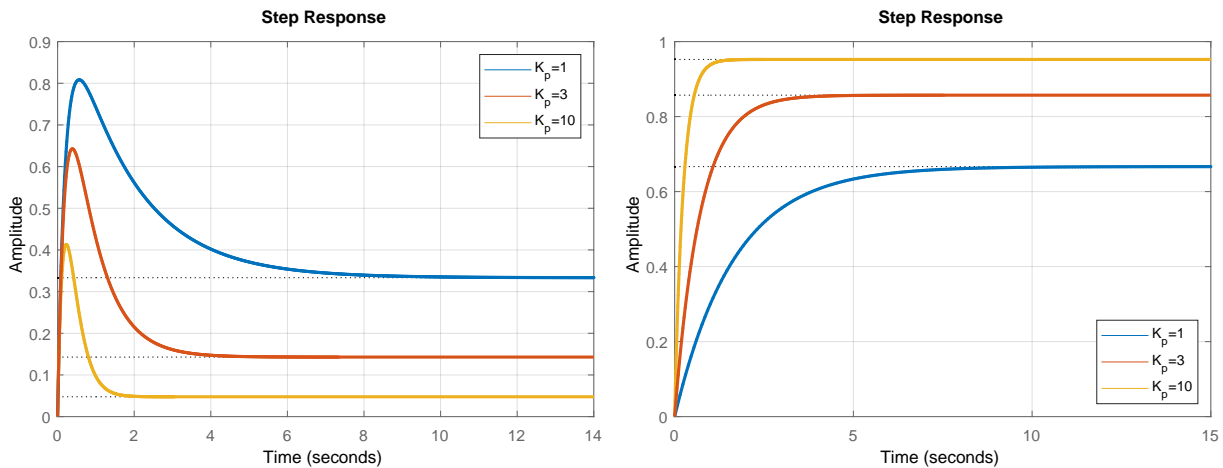


Figure 6.5: Responses of the first-order system controlled by a P controller: (left) regulation and (right) tracking problems ($K = 2$, $T = 5$, $K_d = 1$, and $T_{pd} = 0.2$).

Example. 6.3: Control error for a first-order system using a PI controller

Consider a first-order system with a static gain K and time constant T controlled by a PI controller with a gain K_p and a integral constant T_i . Assume that the disturbance transfer function is also first order with a gain K_d and a time constant T_{pd}

$$G_S = \frac{K}{Ts + 1}, \quad G_c = K_p \left(1 + \frac{1}{T_i s} \right), \quad G_d = \frac{K_d}{T_{pd}s + 1}. \quad (6.35)$$

Derive the steady-state error for the tracking and regulation problems if there be a step change in either the setpoint $r(t) = A$ or the disturbance $d(t) = A$.

Disturbance rejection problem: We consider $d(t) = A$ and $r(t) = 0$. The steady-state error is given as $e(\infty) = r(\infty) - y(\infty)$. We apply the Final Value Theorem

$$y(\infty) = \lim_{s \rightarrow 0} sY(s), \quad Y(s) = G_{yd}(s)D(s), \quad (6.36)$$

$$G_{yd}(s) = \frac{K_d(Ts + 1)s}{(T_{pd}s + 1)(Ts^2 + s + KK_p s + \frac{KK_p}{T_i})}, \quad (6.37)$$

$$y(\infty) = \lim_{s \rightarrow 0} s \frac{K_d(Ts + 1)s}{(T_{pd}s + 1)(Ts^2 + s + KK_p s + \frac{KK_p}{T_i})} \frac{A}{s} = \frac{0}{\frac{KK_p}{T_i}} A = 0, \quad (6.38)$$

$$e(\infty) = r(\infty) - y(\infty) = 0 - 0 = 0. \quad (6.39)$$

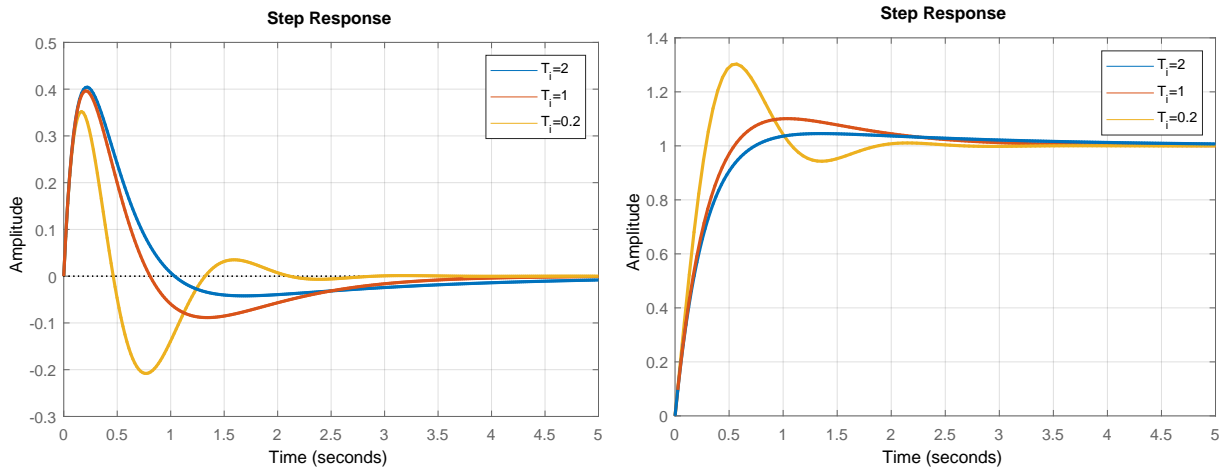


Figure 6.6: Responses of the first-order system controlled by a PI controller: (left) regulation and (right) tracking problems ($K = 2$, $T = 5$, $K_d = 1$, $T_{pd} = 0.2$, and $K_p = 10$).

Setpoint tracking problem: We consider $d(t) = 0$ and $r(t) = A$. It follows that

$$y(\infty) = \lim_{s \rightarrow 0} sY(s), \quad Y(s) = G_{yr}(s)R(s), \quad (6.40)$$

$$G_{yr}(s) = \frac{G_S G_c}{1 + G_S G_c} = \frac{K(K_p s + \frac{K_p}{T_i})}{T s^2 + s + K(K_p s + \frac{K_p}{T_i})}, \quad (6.41)$$

$$y(\infty) = \lim_{s \rightarrow 0} s \frac{K(K_p s + \frac{K_p}{T_i})}{T s^2 + s + K(K_p s + \frac{K_p}{T_i})} \frac{A}{s} = A, \quad (6.42)$$

$$e(\infty) = r(\infty) - y(\infty) = A - A = 0. \quad (6.43)$$

Simulation results for both cases are shown in Figure 6.6. Note how the decreasing value T_i (increasing the integral action) makes the dominant zero leading to an overshoot and oscillations.

6.2.1 Two Tanks with Interaction

We study PI control of two tanks with interaction modelled in Example 1.2.2. The process parameters are: $q_{0,1}^s = 0.9 \text{ m}^3/\text{s}$, $q_{0,2}^s = 0 \text{ m}^3/\text{s}$, $F_1 = 0.5 \text{ m}^2$, $F_2 = 0.6 \text{ m}^2$, $k_{11} = 0.8 \text{ m}^{2.5}/\text{s}$, and $k_{22} = 0.5 \text{ m}^{2.5}/\text{s}$.

The manipulated and disturbance variables are $q_{0,1}$ and $q_{0,2}$, respectively. The controlled variable is the level h_2 measured with a certain precision (modelled by additive white noise). The desired reference is 20% above the initial steady state up to $t = 25 \text{ s}$ and 30% of the initial steady state afterwards. The disturbance is zero up to $t = 40 \text{ s}$ and $0.1 \text{ m}^3/\text{s}$ afterwards. The valve on the manipulated variable cannot be fully closed leading to the lower limit $q_{0,1,\min} = 0.05 \text{ m}^3/\text{s}$. The fully open valve gives the upper limit $q_{0,1,\max} = 2.0 \text{ m}^3/\text{s}$.

We had derived the process gain and time constants in Example 3.2.1: $K = 7.2$, $T_1 = 8.6$, and $T_2 = 0.7$.

6.2.1.1 Design a Simulink model corresponding to the problem description and simulate the

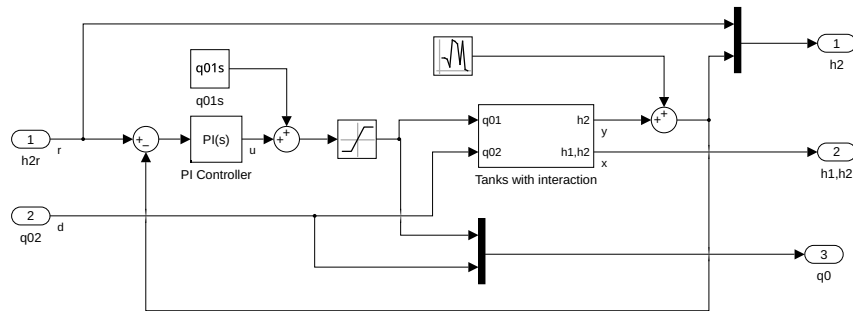


Figure 6.7: PI control of two tanks with interaction: a Simulink model.

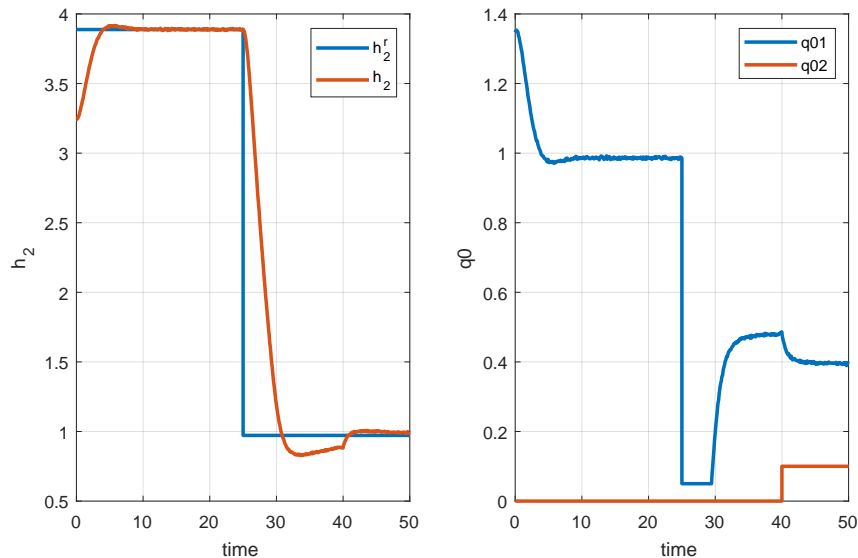


Figure 6.8: PI control of two tanks with interaction: simulation results.

closed-loop system response.

To propose PI constants, we will start with guidelines from Example 6.1. The proportional gain will be $K_p = 5/K$ and the integral time constant will be between the time constants of the process, thus $T_i = 2$.

The model is shown in Figure 6.7 and the simulation results are shown in Figure 6.8. In the first half of the simulation, we can see that the controller behaves satisfactorily as it does not use a fully open valve. It is able to achieve a settling time of about 15 s with a reasonable overshoot. Also, disturbance rejection after $t = 40$ s is fast. The integral part of the controller removes the steady-state control error.

Note that the constant $q_{0,1}^s$ added to the output of the PI controller in Figure 6.7 is not strictly necessary, and the controller can operate without it. It only influences the initial behaviour of the controller. If it were not included, the controller would start from zero initially decreasing the controlled level. As a remedy, industrial controllers include external feedback to guarantee bumpless transfer when the controller starts or resumes its operation.

The tracking between $t \in (25, 40)$ demonstrates the problem of integral windup. The controller output is at the lower limit even after time $t = 30$ s when the

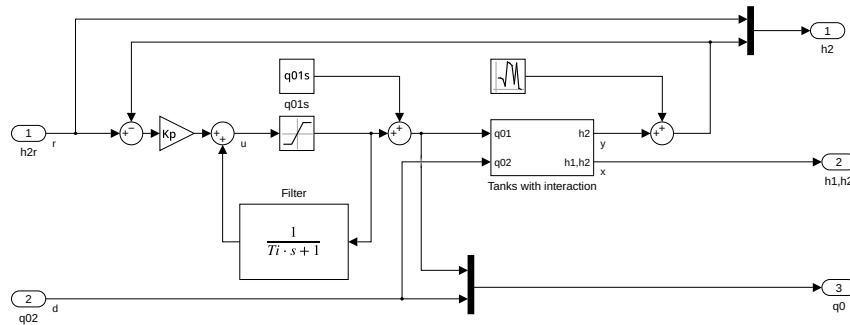


Figure 6.9: PI control of two tanks with interaction: a Simulink model with anti-windup enhancement AW1.

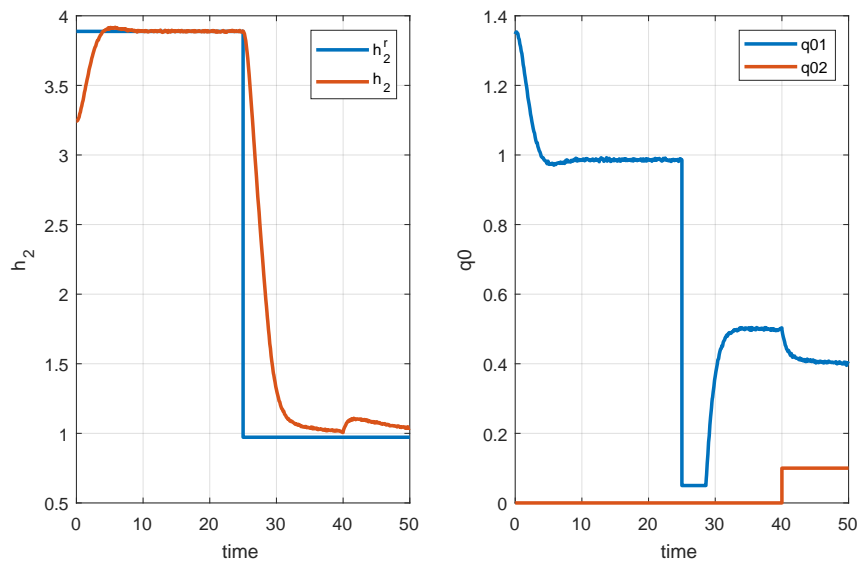


Figure 6.10: PI control of two tanks with interaction: simulation results with anti-windup enhancement AW1.

controlled output crosses the setpoint. This causes a substantial overshoot and almost drains the tank.

6.2.1.2 Design a Simulink model with the anti-windup structure AW1 and simulate the closed-loop system response.

The model is shown in Figure 6.9 and the simulation results are shown in Figure 6.10. When we compare the results with the previous case, we can observe that the first half of the simulation is the same. The saturation block uses the inner controller signal and has to be set to $(0.05 - q_{0,1}^s, 2 - q_{0,1}^s)$.

Tracking during $t \in (25, 40)$ shows a correct action of the controller. It opens the control valve much sooner than before and the overshoot is similar as in the unconstrained case.

6.2.1.3 Design a Simulink model with the anti-windup enhancement AW2 and simulate the closed-loop system response.

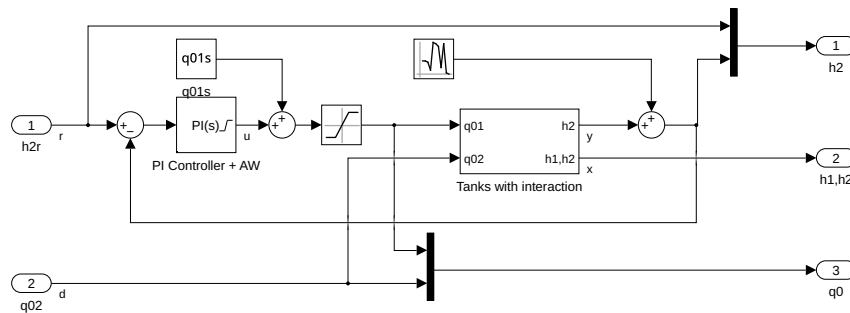


Figure 6.11: PI control of two tanks with interaction: a Simulink model with anti-windup enhancement AW2.

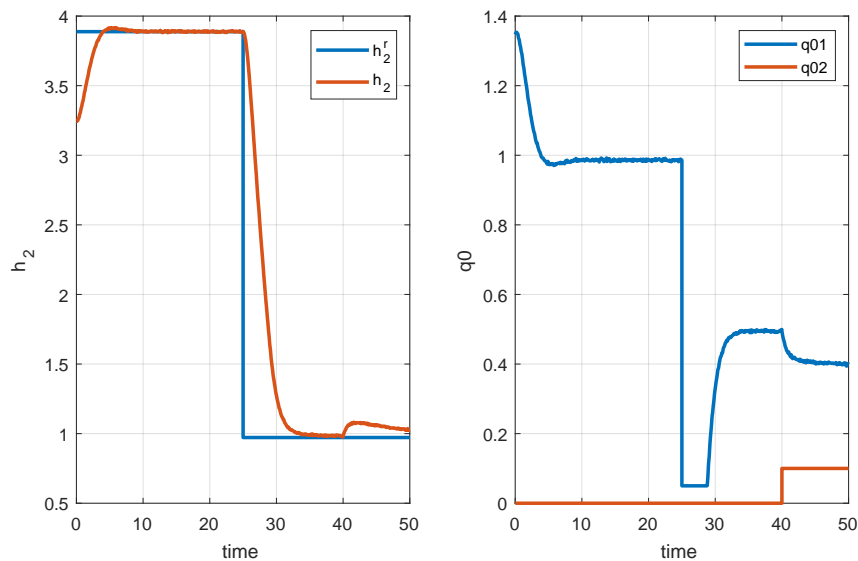


Figure 6.12: PI control of two tanks with interaction: simulation results with anti-windup enhancement AW2.

The model is shown in Figure 6.11. Simulink provides two anti-windup solutions in the PID controller block, the solution AW2 is called a back-calculation and the so-called clamping method. We have chosen the back-calculation coefficient $1/T_i$. Simulation results are shown in Figure 6.12. When we compare the result with the uncompensated case, we can observe again that the first half of the simulation is the same as we again use the addition of $q_{0,1}^s$. However, note that the inner limits within the PI controller have been set as $(0.05 - q_{0,1}^s, 2 - q_{0,1}^s)$.

Tracking during $t \in (25, 40)$ shows the correct action of the controller, and it behaves practically the same as AW1. The time the controller spends at the constraint can be fine-tuned by changing the value of the back-calculation coefficient.

6.2.2 Two Tanks with a Pump

We study the proportional control of two tanks with a pump modelled in Example 1.2.3. The process parameters are: $q_{0,1}^s = 0.9 \text{ m}^3/\text{s}$, $q_{0,2}^s = 0.9 \text{ m}^3/\text{s}$, $F_1 = 0.5 \text{ m}^2$, $F_2 = 0.6 \text{ m}^2$, $k_{11} = 0.8 \text{ m}^{2.5}/\text{s}$, and $k_{22} = 0.5 \text{ m}^{2.5}/\text{s}$.

The manipulated and disturbance variables are $q_{0,2}$, $q_{0,1}$, respectively. The controlled variable is the level h_2 and it is measured with a certain precision (modelled by an additive white noise). The desired setpoint is 20% above the initial steady-state level up to time $t = 25$ s and 30% of the initial steady-state level afterwards. The disturbance is $q_{0,1}^s$ up to $t = 40$ s and $0.9q_{0,1}^s$ afterwards. The valve on the manipulated variable cannot be fully closed leading to the lower limit $q_{0,2,\min} = 0.05 \text{ m}^3/\text{s}$ and fully open valve gives an upper limit $q_{0,2,\max} = 2.0 \text{ m}^3/\text{s}$.

6.2.2.1 Design a Simulink model corresponding to the problem description and simulate the closed-loop system response with a P controller. The proposed controller is reversely acting with a gain $K_p = -2$.

The model is shown in Figure 6.13 and the simulation results in Figure 6.14. We can see that the P controller stabilises the closed-loop system, and the pump always finds a correct value corresponding to the actual input disturbance in $q_{0,1}$. The process exhibits integrating behaviour. However, steady-state tracking error can be observed due to the nonlinear characteristics of the process. Its amount can be controlled by the appropriate choice of the proportional gain. Larger values, however, increase the sensitivity of the controller to the measurement noise.

6.2.2.2 Simulate the closed-loop system with a PI controller with an anti-windup enhancement. The proposed controller is reversely acting using the control law

$$G_c = -2 \left(1 + \frac{1}{2s} \right). \quad (6.44)$$

Simulation results are shown in Figure 6.15. Compared to the previous results, the PI controller structure results in zero steady-state control error. If anti-windup were not implemented, we would observe large overshoot in the process output at $t = 25$ s. The observed overshoot is due to the dominant zero of the controller.

6.3 Practice Examples

6.3.1 First-Order System Control

We study a first-order system

$$G(s) = \frac{K}{Ts + 1}, \quad (6.45)$$

controlled by P and PI controllers.

6.3.1.1 Formulate a transfer function of the closed-loop system. Find its poles, zeros, the static gain, and the time constant. Discuss the effect of tuning parameter(s) on the control performance. Treat P and PI cases separately. *PI hint: For simplicity start with $T_i = T$.*

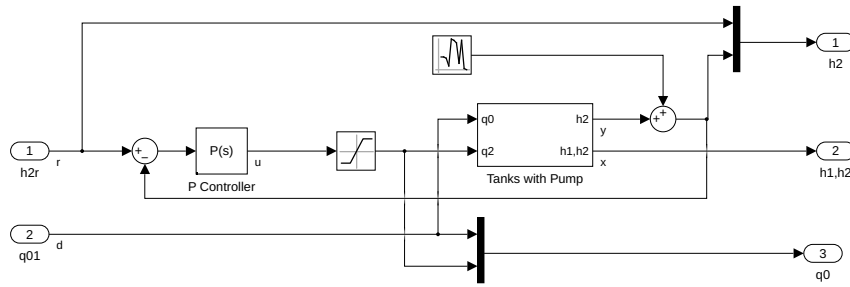


Figure 6.13: Control of two tanks with a pump: a Simulink model.

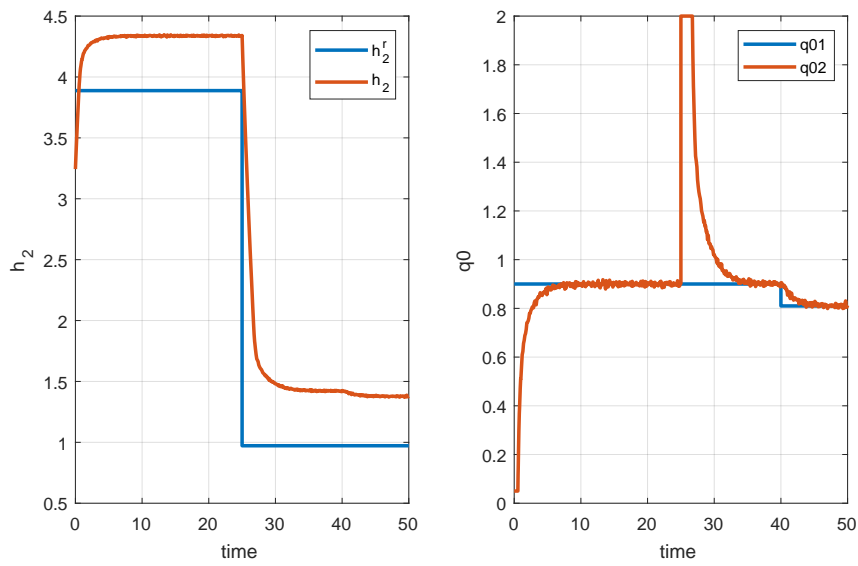


Figure 6.14: P control of two tanks with a pump: simulation results.

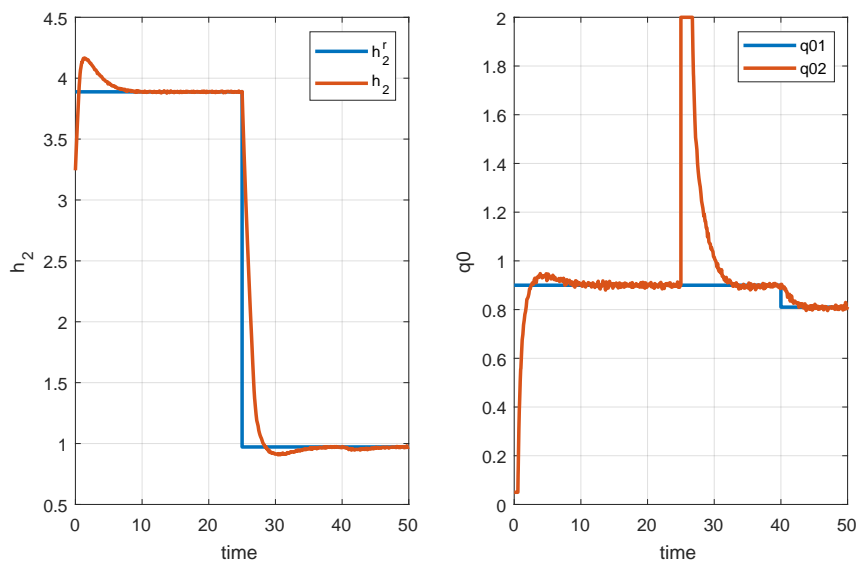


Figure 6.15: PI control of two tanks with a pump: simulation results.

- P controller: The closed-loop transfer function is:

$$G_{cl}(s) = \frac{G_S(s)G_c(s)}{1 + G_S(s)G_c(s)} = \frac{K_c \frac{K}{Ts+1}}{1 + K_c \frac{K}{Ts+1}} = \frac{K_c K}{Ts + K_c K + 1}. \quad (6.46)$$

The transfer function in (6.46) has no zeros. The closed-loop system is a first-order system with the pole

$$s = -\frac{1}{T} - \frac{K_c K}{T}. \quad (6.47)$$

Now, we are able to analyse the effect of the controller gain on closed-loop system's pole. Assuming that $K > 0$ and $T > 0$, with increasing K_c the value of the pole becomes more negative. Therefore, the closed-loop dynamics is faster.

To analyse the static gain and the time constant, we re-arrange the transfer function into the following form:

$$G_{cl}(s) = \frac{K_c K}{Ts + K_c K + 1} = \frac{\frac{K_c K}{1 + K_c K}}{\frac{T}{1 + K_c K} s + 1}, \quad (6.48)$$

where the static gain is

$$K_{cl} = \frac{K_c K}{1 + K_c K}, \quad (6.49)$$

and the time constant is

$$T_{cl} = \frac{T}{1 + K_c K}. \quad (6.50)$$

For high values of the controller gain in (6.49), the static gain K_{cl} asymptotically approaches 1. For smaller values of the controller gain, the static gain is always less than 1. By increasing the controller gain in (6.50), the time constant T_{cl} decreases. As a consequence, the dynamics of the system becomes faster.

- PI controller: The closed-loop transfer function is:

$$G_{cl}(s) = \frac{G_S(s)G_c(s)}{1 + G_S(s)G_c(s)} = \frac{\frac{K_c(T_i s + 1)}{T_i s} \frac{K}{Ts+1}}{1 + \frac{K_c(T_i s + 1)}{T_i s} \frac{K}{Ts+1}}. \quad (6.51)$$

Assuming $T_i = T$, the transfer function (6.51) is simplified to:

$$G_{cl}(s) = \frac{\frac{K_c K}{T s}}{1 + \frac{K_c K}{T s}} = \frac{\frac{K_c K}{T s}}{\frac{T s + K_c K}{T s}} = \frac{K_c K}{T s + K_c K}. \quad (6.52)$$

The closed-loop transfer function has no zeros and one pole:

$$s = -\frac{K_c K}{T}. \quad (6.53)$$

Analogously to the previous case, when considering $K > 0$ and $T > 0$, by increasing the controller gain, the pole becomes more negative and the dynamics of the system increases.

The static gain of the closed-loop system is:

$$K_{cl} = \frac{K_c K}{K_c K} = 1, \quad (6.54)$$

which means that the tuning of the controller gain has no effect on the static gain and it is always equal to 1.

The closed-loop time constant is

$$T_{cl} = \frac{T}{K_c K}. \quad (6.55)$$

By increasing the controller gain, the time constant T_{cl} decreases and therefore the closed-loop dynamics is faster.

If we assume that $T_i \neq T$, the transfer function is:

$$G_{cl}(s) = \frac{\frac{K_c(T_i s + 1)}{T_i s} \frac{K}{T s + 1}}{1 + \frac{K_c(T_i s + 1)}{T_i s} \frac{K}{T s + 1}} = \frac{K_c K (T_i s + 1)}{T_i s (T_i s + 1) + K_c K (T_i s + 1)} \quad (6.56)$$

$$= \frac{\frac{K_c K}{T_i T} (T_i s + 1)}{s^2 + \frac{T_i (K_c K + 1)}{T T_i} s + \frac{K_c K}{T_i T}}. \quad (6.57)$$

By comparing the transfer function (6.57) with the general form,

$$G_{cl}(s) = \frac{K_{cl} \omega_{0,cl}^2 (T_i s + 1)}{s^2 + 2\xi_{cl} \omega_{0,cl} s + \omega_{0,cl}^2}, \quad (6.58)$$

we are able to analyse the closed-loop dynamics and the static gain. The static gain of the closed-loop system is

$$K_{cl} = 1, \quad (6.59)$$

which means that the tuning the controller gain has no effect on the static gain and is always 1. The frequency $\omega_{0,cl}$ is

$$\omega_{0,cl} = \sqrt{\frac{K_c K}{T T_i}}, \quad (6.60)$$

which means that by increasing the controller gain K_c , the frequency increases and the closed-loop dynamics becomes faster.

6.3.1.2 Verify the theoretical analysis using numerical simulations.

6.3.1.3 (Optional) If available, implement experimental control of the Flexy2 device.

6.3.2 Two Tanks in Series

We study the process with two tanks in series from Example 1.2.4 with parameters $q_{0,1}^s = 0.3 \text{ m}^3/\text{s}$, $q_{0,2}^s = 0.5 \text{ m}^3/\text{s}$, $F_1 = 0.5 \text{ m}^2$, $F_2 = 0.8 \text{ m}^2$, $k_{11} = 1.15 \text{ m}^{2.5}/\text{s}$, $k_{22} = 1.3 \text{ m}^{2.5}/\text{s}$. The process is in the initial steady state at $t = 0$.

Table 6.1: Comparison of the PI controllers.

	$K_c = 10, T_i = 1$	$K_c = 1, T_i = 10$
σ	61%	0
T_{100}	0.59	-
T_{005}	2.47	14.77
ISE	0.07	0.34

The manipulated and disturbance variables are $q_{0,1}$ and $q_{0,2}$, respectively. The controlled variable is the level h_2 . The desired reference is 100% above the initial steady state up to $t = 25$ s and 70% of the initial steady state afterwards. The disturbance is zero after $t = 40$ s. The manipulated variable is limited between $0.1 \text{ m}^3/\text{s}$ and $2.0 \text{ m}^3/\text{s}$.

6.3.2.1 Calculate the process steady state, the transfer functions between the output and the input variables, their gains, poles, zeros, and time constants.

The process steady state is defined by (2.113), (2.114) as $h_1^s = 0.07$ m and $h_2^s = 0.38$ m. The transfer function between the output and the first input is:

$$G_{yu_1}(s) = \frac{\frac{k_1}{F_1 F_2}}{(s + \frac{k_1}{F_1})(s + \frac{k_2}{F_2})} = \frac{5.51}{s^2 + 5.73s + 5.82}. \quad (6.61)$$

Its poles are

$$s_1 = -\frac{k_1}{F_1} = -4.41, \quad s_2 = -\frac{k_2}{F_2} = -1.32, \quad (6.62)$$

which means that the system is stable and overdamped.

The time constants are

$$T_1 = \frac{F_1}{k_1} = 0.23, \quad T_2 = \frac{F_2}{k_2} = 0.76. \quad (6.63)$$

Finally, the gain of the system is $K = 0.95$.

The transfer function between the output and the second input is:

$$G_{yu_2} = \frac{\frac{1}{F_2}}{s + \frac{k_2}{F_2}} = \frac{1.25}{s + 1.32}. \quad (6.64)$$

The pole of the system is $s = -k_2/F_2 = -1.32$, which means that the system is stable and overdamped. The time constant is $T = F_2/k_2 = 0.76$. Finally, the gain of the system is $K = 0.95$.

6.3.2.2 Design a feedback control strategy with a PI controller. Observe quality indicators (overshoot, rise time, settling time, ISE) for various values of the controller parameters.

The solution for two values of controller parameters (up to $t = 25$ s) is summarised in Table 6.1 and shown in Figure 6.16.

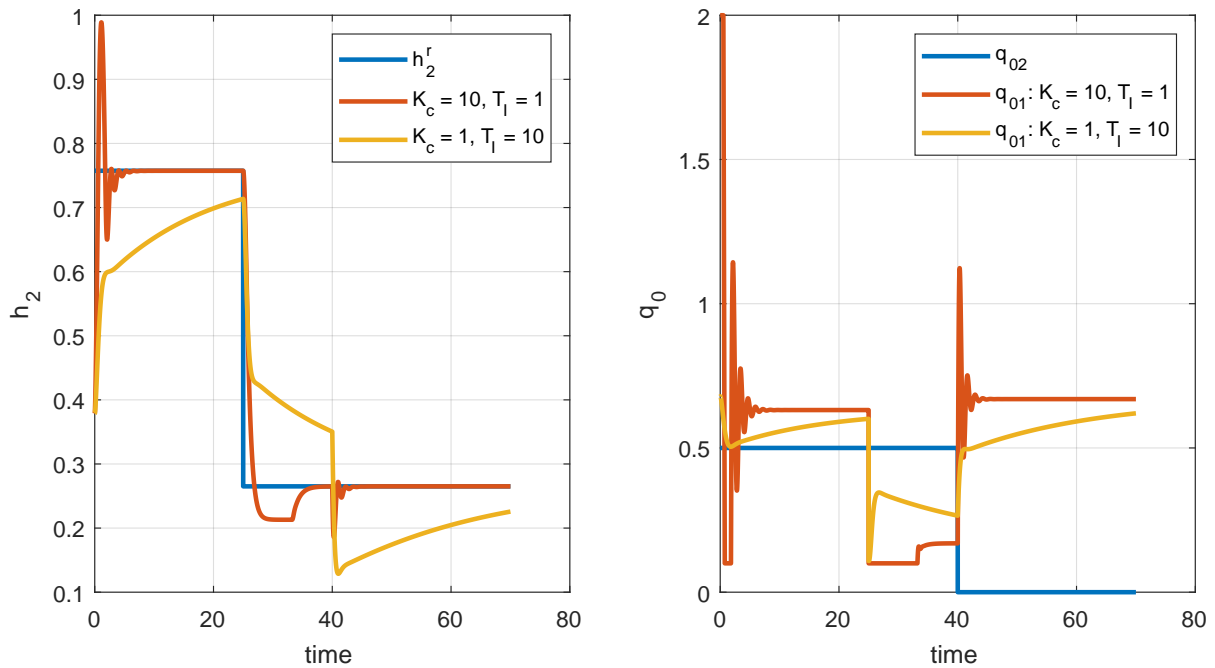


Figure 6.16: Control of two tanks in series, comparison of PI controllers – (left) trajectory of controlled variable $h_2(t)$, (right) manipulated variable $q_{0,1}(t)$.

6.3.2.3 Design a feedback control strategy with a PID controller and assume the measurement noise is present in the controlled output. Choose the PI part of the controller as the best one from the previous step. Observe the performance indices (overshoot, rise time, settling time, ISE, steady-state control error) for various values of the derivative time constant.

Increasing the derivative term in the controller improves overshoot and settling time. However, the presence of measurement noise causes jitter – the manipulated variable oscillates heavily.

6.3.2.4 Modify the PI controller to include anti-windup compensation AW2. Compare your solution based on Figure 6.4 with the PI controller without AW.

Comparison of simulations with and without anti-windup is shown in Figure 6.17. The controller parameters are $K_c = 10$, $T_i = 1$ and the AW parameter $T_t = 1$. We can see that controller actions in regulation and tracking are comparable except after $t > 25$ s where the original output variable suffers from windup.

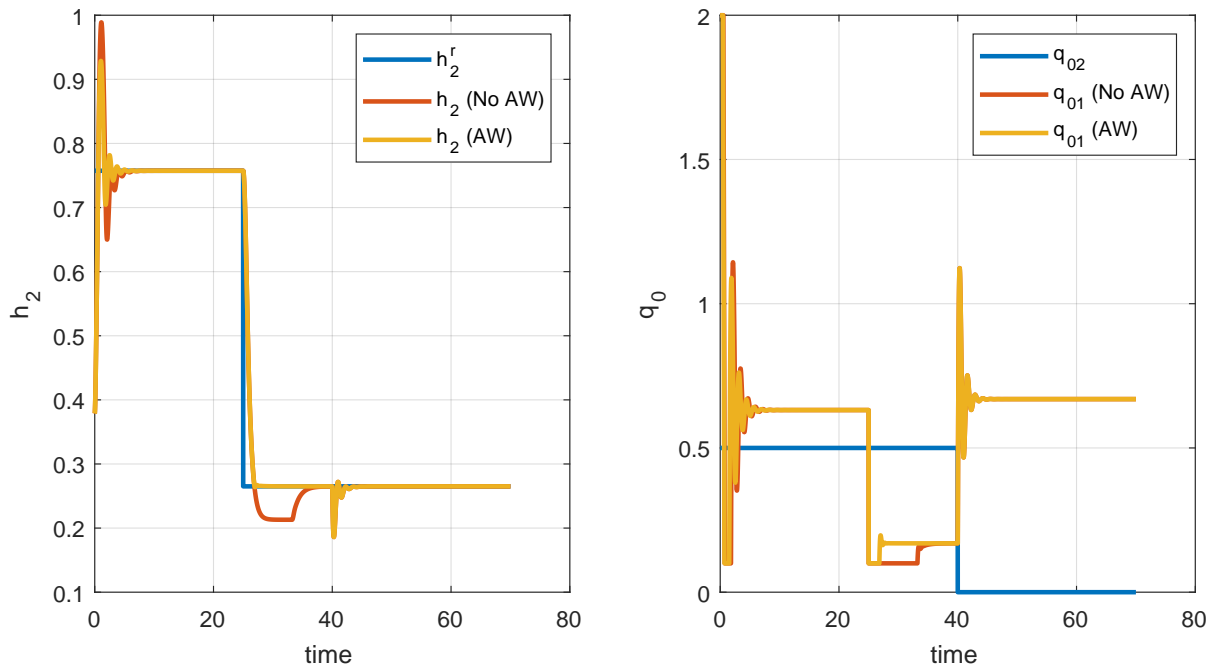


Figure 6.17: Control of two tanks in series, comparison of PI controllers with and without anti-windup – (left) time responses of controlled variable $h_2(t)$, (right) manipulated variable $q_{0,1}(t)$.

6.4 Problems

Problem. 6.1: Offset-free control of a two tanks system with a pump

Consider the two tanks system with a pump (Example 1.2.3). The process transfer function was determined in (2.103) as

$$G(s) = \frac{Y(s)}{U(s)} = \frac{k_1}{s(F_1 F_2 s + k_1 F_1 + k_1 F_2)}. \quad (6.65)$$

Determine the steady-state error if the process is controlled either by a proportional controller with a gain K_p or by a PI controller with parameters K_p and T_i .

As the loop contains an integrator in the process transfer function, the steady-state error is zero as long as the closed-loop system is stable.

Problem. 6.2: Steady-state tracking error

Calculate the steady-state tracking error of a controlled system

$$G_S(s) = \frac{7}{s^3 + 4s^2 + 3s + 2}, \quad (6.66)$$

and the error of disturbance rejection for the disturbance transfer function

$$G_d(s) = \frac{4}{s + 2}. \quad (6.67)$$

Assume control with a proportional controller with $K_p = 1$. Check whether the closed-loop system is stable. Consider two cases: step change in the setpoint with $r(\infty) = 1$ and step change in the disturbance with $d(\infty) = 1$.

The closed-loop system is stable, $e_r(\infty) = 2/9$, $e_d(\infty) = 4/9$.

```
gs = tf(7,[1 4 3 2]); gd = tf(4,[1 2]); gc = 1;
gyr = gs*gc/(1+gs*gc); gyr=minreal(gyr);
gyd = gpr/(1+gs*gc); gyd = minreal(gyd);
[num,den]=tfdata(gyr,'v'); roots(den)
er = 1-dcgain(gyr); ed = dcgain(gyd)
```

Problem. 6.3: Stability

Consider the closed-loop system from the previous problem with a proportional controller with $K_p = 4$. Check whether the closed-loop system is stable. Determine the range of values K_p for which the closed-loop system is stable.

Hint: Use Routh-Hurwitz stability criterion to find the critical controller gain. $K_p < 10/7$.

Problem. 6.4: Setpoint weighting

Consider again the controlled system

$$G_S(s) = \frac{7}{s^3 + 4s^2 + 3s + 2}, \quad (6.68)$$

and the PI controller $G_c(s) = 0.5 + 0.1/s$. Although the closed-loop system is stable, the tracking behaviour is rather oscillatory. Apply a 2DoF controller (6.24) to improve the tracking behaviour.

```
gs = tf(7,[1 4 3 2]); gc0 = pid(0.5, 0.1);
gyr0 = gs*gc0/(1+gs*gc0);
b = 0.3; gcf = pid(0.5*b, 0.1); gcb = gc0;
gyr = gs*gcf/(1+gs*gcb); step(gyr0,gyr)
```

6.5 MATLAB Snippets

- The PID controller transfer function

$$G_c(s) = K_p + \frac{K_i}{s} + \frac{K_d s}{T_f s + 1}.$$

```
gc = pid(Kp);
gc = pid(Kp, Ki);
gc = pid(Kp, Ki, Kd);
gc = pid(Kp, Ki, Kd, Tf);
```

with default values $K_p = 1$, $K_i = 0$, $K_d = 0$, and $T_f = 0$.

- Closed-loop stability

```
gs = tf(7, [1 4 3 2]); gc = 4;
gyr = gs * gc / (1 + gs * gc); gyr = minreal(gyr);
[num, den] = tfdata(gyr, 'v'); pole(gyr), roots(den), isstable(gyr), pzmap(gyr)
```

- Feedback connection of two systems

```
gs = tf(7, [1 4 3 2]); gc = 1;
gyr = gs * gc / (1 + gs * gc); gyr = minreal(gyr);
gyr2 = feedback(gs * gc, 1);
step(gyr, gyr2)
```

6.6 Python Snippets

- PID controller transfer function

$$G_c(s) = K_p + \frac{K_i}{s} + \frac{K_d s}{T_f s + 1}$$

```
from control.matlab import *
gc = Kp
gc = tf([Kp, Ki], [1, 0])
gc = tf([Kd + Tf * Kp, Tf * Ki + Kp, Ki], [Tf, 1, 0])
```

- Closed-loop stability

```
from control.matlab import *
gs = tf(7, [1, 4, 3, 2]); gc = 4;
gyr = gs * gc / (1 + gs * gc); gyr = minreal(gyr);
[num,den]=tfdata(gyr); roots(den[0][0])
pzmap(gyr); plt.show()
```

- Feedback connection of two systems

```
gs = tf(7, [1, 4, 3, 2]); gc = 1
gyr = gs * gc / (1 + gs * gc); gyr = minreal(gyr)
gyr2 = feedback(gs * gc, 1)
step(gyr); step(gyr2)
```

6.6.1 PI Control of Two Tanks with Interaction

We will model both the process and the PI controller as nonlinear systems. The reason for a nonlinear PI controller is the anti-windup structure AW1 (Figure 6.3). Both nonlinear systems are defined by differential equations, output equation, and by final `NonlinearIOSystem`. The feedback connection is implemented using the class `InterconnectedSystem`.

The program produces simulation results shown in Figs. 6.10.

```
import control
import numpy as np
import matplotlib.pyplot as plt
import math

def tanks2i_rhs(t, x, u, params):
    F1, F2, k11, k22 = 0.5, 0.6, 0.8, 0.5
    h1, h2 = x
    q01, q02 = u
    h2 = 0 if h2 < 0 else h2 # numerical issues
```

```

h1 = h2 if h1 < h2 else h1
dh1 = (q01 - k11 * math.sqrt(h1 - h2)) / F1
dh2 = (q02 + k11 * math.sqrt(h1 - h2) - k22 * math.sqrt(h2)) / F2
return [dh1, dh2]

def tanks2i_out(t, x, u, params):
    return x[1]

tanks2i_io = control.NonlinearIOSystem(
    tanks2i_rhs, tanks2i_out, inputs=('q01','q02'), outputs=('h2'),
    states=('h1', 'h2'), name='tanks2i')

def pi_rhs(t, x, u, params):
    ti = params.get('ti')
    r = u[0]; y = u[1]; ub = x[0]
    usat = pi_out(t, x, u, params)
    return (usat - ub) / ti

def pi_out(t, x, u, params):
    kp = params.get('kp');
    umin = params.get('umin', 0.0); umax = params.get('umax', 100.0)
    r = u[0]; y = u[1]; ub = x[0]
    return np.clip(kp * (r - y) + ub, umin, umax)

piaw_io = control.NonlinearIOSystem(
    pi_rhs, pi_out, name='piaw',
    inputs = ['r', 'y'], outputs = ['u'], states = ['x'],
    params = {'kp':0.6944, 'ti':8.6216, 'umin':0.05, 'umax':2.0})

sum_noise = control.summing_junction(inputs=['y', 'n'],
                                     output='ynoise',
                                     name='sumn')

cls_io = control.interconnect(
    (tanks2i_io, piaw_io, sum_noise),
    name='clsio',
    connections=(
        ['sumn.y', 'tanks2i.h2'],
        ['tanks2i.q01', 'piaw.u'],
        ['piaw.y', 'sumn.ynoise']),
    inplist=('piaw.r', 'tanks2i.q02', 'sumn.n'),
    inputs=('r', 'q02', 'n'),
    outlist=('sumn.ynoise', 'tanks2i.q01'),
    outputs=('h2n', 'u'))

q01s, q02s = 0.9, 0
hs, qs = control.find_eqpt(tanks2i_io, [2, 1], [q01s, q02s])
T = np.linspace(0, 50, 200)
noise = np.random.normal(0, 0.005, T.shape)
q02 = [
    0 if t <= 40 else
    .1 for t in T]
r = [
    1.2 * hs[1] if t <= 25 else
    0.3 for t in T]
t, y = control.input_output_response(cls_io, T, [r, q02, noise], [hs[0], hs[1], q01s

```

```
    ])  
  
plt.close('all')  
plt.figure(1)  
plt.step(T, r, label='setpoint'); plt.plot(t, y[0], label='h2'); plt.legend()  
plt.figure(2)  
plt.plot(t, y[1], label='q01'); plt.step(T, q02, label='q02'); plt.legend()  
plt.show(block=False)
```

6.7 Bibliography

The theoretical background for this chapter is fairly standard and can be found in most of textbooks on automatic control. For example in Mikleš and Fikar (2007, Chapter 7), information in Slovak in Bakošová and Fikar (2008, Chapter 5).

For complementary information, see for example Seborg, Edgar, Mellichamp, and Doyle (2016, Chapters 8, 9), Marlin (1995, Chapter 8), Åström and Hägglund (1995).

Root-Locus Analysis and Design

In previous chapters, we have seen that the location of poles (and zeros) of a closed-loop system determines its performance and its absolute and relative stability. Root-locus technique studies the roots of the closed-loop characteristic equation as a function of a single control parameter. We will study methods for sketching the plots of the roots-locus function and for generating them using a computer. The controller structure and also the controller parameters can be designed and tuned using the root-locus plots.

Based on this chapter, you should :

- understand the concept of root-locus plots in control system analysis and design,
- learn the basic rules that help to sketch root-locus plots,
- be able to generate root-locus plots using software,
- to analyse the root-locus plots,
- design PID controllers based on the root-locus analysis.

7.1 Overview

Consider a characteristic equation of a feedback control system with a controller $G_c(s)$ and process $G(s)$

$$1 + G_c G = 1 + \bar{G}_o = 1 + K G_o = 0, \quad (7.1)$$

where \bar{G}_o denotes the open-loop transfer function consisting of the process and controller. It is assumed that there is a single parameter K of this transfer function and we wish to study its influence on the closed-loop pole locations. We assume that K can be any nonnegative number.

The transfer function $G_o(s)$ can be expressed as a fraction of two polynomials or in terms of poles and zeros

$$G_o = \frac{B(s)}{A(s)} = \frac{(s - z_1) \cdots (s - z_m)}{(s - p_1) \cdots (s - p_n)}. \quad (7.2)$$

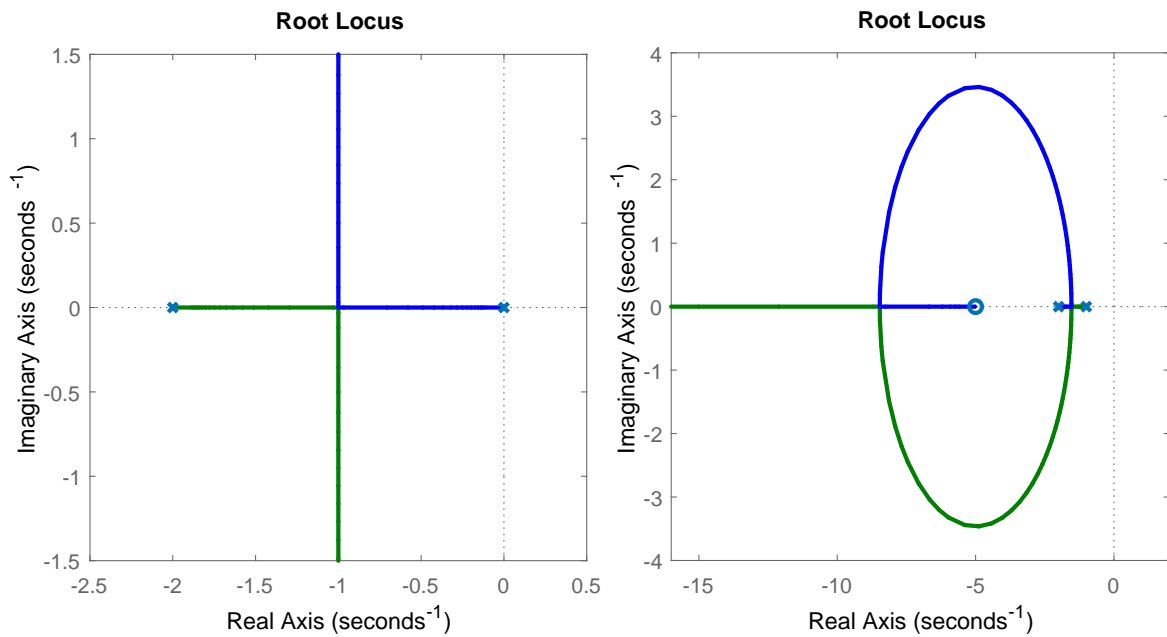


Figure 7.1: (left) Root-locus plot for Example 7.1 and (right) Example 7.2

Therefore, the characteristic equation can be expressed as a polynomial $f(s)$

$$1 + KG_o = 1 + K \frac{B(s)}{A(s)} = 0, \quad \Rightarrow \quad f(s) = A(s) + KB(s) = 0. \quad (7.3)$$

Example. 7.1: First-order system with an integral controller

Consider a process with a transfer function $G(s) = 1/(s + 2)$ controlled by an integral controller $G_c = K/s$. Calculate the closed-loop poles as functions of the gain K .

The characteristic equation of the closed-loop system is given as

$$1 + K \frac{1}{s(s+2)} = 0, \quad s^2 + 2s + K = 0. \quad (7.4)$$

The roots are functions of K :

$$s_{1,2} = \begin{cases} -1 \pm \sqrt{1 - K}, & K \leq 1, \\ -1 \pm j\sqrt{K - 1}, & K > 1. \end{cases} \quad (7.5)$$

Therefore, we can claim that the closed-loop system will never be unstable, as the poles contain negative real parts or are negative real. Also, the system is overdamped for $K \leq 1$ and underdamped for $K > 1$. The location of the roots start at the open-loop poles and moves as two branches.

Figure 7.1 (left) shows the root-locus plot obtained with the simple command

```
rlocus(1, [1 2 0]) % 1/(s(s+2)) -> 1/(s^2 + 2s + 0)
```

Notice the the two branches (one for each root/pole) in blue and green.

Example. 7.2: Second-order system with a PD controller

Consider a process with a transfer function $G(s) = 1/(s+1)(s+2)$ controlled by a PD controller $G_c = K(1+0.2s)$. Calculate the closed-loop poles as functions of the gain K .

The characteristic equation of the closed-loop system is given as

$$0 = 1 + K \frac{1 + 0.2s}{(s+1)(s+2)}, \quad (7.6)$$

$$0 = s^2 + (3 + 0.2K)s + (2 + K). \quad (7.7)$$

The process characteristic equation is quadratic with a discriminant

$$D = (3 + 0.2K)^2 - 4(2 + K). \quad (7.8)$$

Positive values of the discriminant give rise to two real negative roots for either small values of K , $K < 0.36$, or large values $K > 70$. The closed loop possesses two complex conjugate roots with negative real parts for the intermediate values of K . If $K \rightarrow \infty$, one real root goes to $-\infty$, the other one converges towards zero at $z = -1/0.2 = -5$.

Figure 7.1 (right) shows the root-locus plot obtained with the command

```
rlocus([0.2 1], [1 3 2])
```

Root-locus conditions follow from (7.1) as closed-loop poles must satisfy this equation. Since it is a function of the complex variable s , it can be rewritten into magnitude and phase conditions:

$$KG_o = -1 + 0j, \quad (7.9)$$

$$|KG_o| = 1, \quad (7.10)$$

$$\varphi(KG_o) = \pi \pm 2k\pi, \quad k = 0, 1, 2, \dots \quad (7.11)$$

7.1.1 General Features of a Root-Locus Plot

We consider a transfer function $G_o(s)$ from (7.2) with n poles and m zeros. We can state the following rules that are valid for the closed-loop pole locations as functions of the parameter K and help to sketch the root-locus plots:

- R1 : The number of loci (branches) is equal to n .
- R2 : The root locus on the real axis exists if there is an odd number of poles and zeros to the right of it on the real axis.
- R3 : The root locus is symmetric about the real axis. It is attracted by zeros and repelled by poles.
- R4 : The root loci originate at the poles of G_o and they end either at the zeros (m) or at infinity ($n - m$).
- R5 : $n - m$ branches converge to the asymptotes. The asymptotes intersect the real axis at the centre of gravity σ_w with angles ϕ_k

$$\sigma_w = \frac{\sum_{k=1}^n p_k - \sum_{k=1}^m z_k}{n - m}, \quad \phi_k = \frac{\pi + 2k\pi}{n - m}, \quad k = 1, \dots, n - m. \quad (7.12)$$

R6 : Break-away (break-in) points – points σ_k on the real axis where the loci meet and leave

$$\sum_{i=1}^n \frac{1}{\sigma_k - p_i} = \sum_{i=1}^m \frac{1}{\sigma_k - z_i}, \quad \text{or } = 0 \text{ for } m = 0, \quad (7.13)$$

$$\frac{dK}{ds} = 0. \quad (7.14)$$

Note, that this is only a necessary condition, any solution found has to be checked so that it is a positive value of K .

R7 : The value of K at point s is

$$K = -\frac{\prod_{i=1}^n (s - p_i)}{\prod_{i=1}^m (s - z_i)}, \quad \text{or } K = -\prod_{i=1}^n (s - p_i) \quad \text{if } m = 0. \quad (7.15)$$

R8 : Stability limit K_c

$$K_c \prod_{i=1}^m (j\omega_c - z_i) + \prod_{i=1}^n (j\omega_c - p_i) = 0. \quad (7.16)$$

7.2 Examples

7.2.1 Construction of Root-Locus Plots

Consider five different dynamic systems with the following transfer functions:

$$G_1(s) = \frac{1}{s + 2}, \quad (7.17a)$$

$$G_2(s) = \frac{s + 8}{s^2 + 6s + 8}, \quad (7.17b)$$

$$G_3(s) = \frac{s + 4}{s^2 + 10s + 16}, \quad (7.17c)$$

$$G_4(s) = \frac{s + 2}{s^2 + 12s + 32}, \quad (7.17d)$$

$$G_5(s) = \frac{1}{s^3 + 14s^2 + 56s + 64}. \quad (7.17e)$$

7.2.1.1 A proportional controller is chosen to adjust the input variable. Discuss the influence of the value of a gain of the controller for stability, damping ratio, and response rate of the system. Use the root-locus plot to support your arguments.

The final root-locus plots for all examples are shown in Figure 7.4.

(a) $G_1(s) = \frac{1}{s+2}$

R1 : There will be one branch in the root locus.

R2 : Every point on the real axis to the left of the pole at -2 is a part of the root locus.

R3 : As there exists only one branch, the root locus will remain on the real axis.

R4 : As there are no zeros, the one branch will terminate at infinity.

This practically concludes the construction of the root-locus plot as we are ready to draw the complete diagram and verify it using a computer.

```
G1 = tf(1, [1 2]);
figure; rlocus(G1)
iam_save_pdf('plt_rl_g1.pdf', 11, 11)
```

Influence of the increasing value of a gain of the controller:

- on stability: system will be always (theoretically) stable,
- on damping ratio: response will always be overdamped (non-oscillatory),
- on response rate of the system: system response will always speed up.

(b) $G_2(s) = \frac{s+8}{s^2+6s+8}$

R1 : There will be two branches of the root locus.

R2 : Every point on the real axis between $-\infty$ and -8 and between -4 and -2 is part of the root locus.

R3 : It will suffice to find one of the two branches, the second one is its mirror image.

R4 : The branches commence at the points -4 and -2 . As there is one zero present at the point -8 , one branch will terminate at -8 and the other branch will terminate at (minus) infinity.

R5 : There is only one asymptote with the angle $\phi_1 = (\pi + 2\pi) = \pi$.

R6 : There are two points where branches meet on the real axis:
 $1/(\sigma_k+4)+1/(\sigma_k+2) = 1/(\sigma_k+8) \rightarrow$
 $(\sigma_k+2)(\sigma_k+8) + (\sigma_k+4)(\sigma_k+8) =$
 $(\sigma_k+2)(\sigma_k+4) \rightarrow$
 $\sigma_k^2 + 16\sigma_k + 40 = 0 \rightarrow \sigma_1 =$
 $-12.8990, \sigma_2 = -3.1010.$

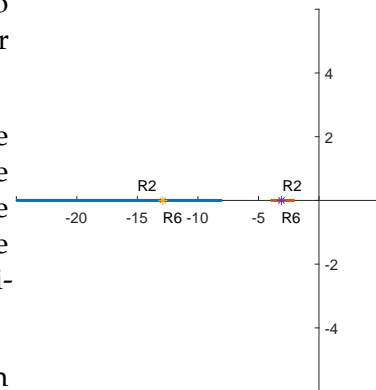


Figure 7.2: Root-locus sketch for $G_2(s)$.

We are ready to sketch the important parts of the root-locus diagram. The actual diagram can be plotted using the commands below.

```
G2 = tf([1 8], [1 6 8]);
figure; rlocus(G2)
iam_save_pdf('plt_rl_g2.pdf', 11, 11)
```

Influence of the increasing value of a gain of the controller:

- on stability: system will be always (theoretically) stable,

- on damping ratio: response will exhibit underdamped behavior for middle range of gain values,
- on response rate of the system: system response will speed up nearly always; for high range of gain values, the response settling will become slower.

$$(c) G_3(s) = \frac{s+4}{s^2+10s+16}$$

- R1 : There will be two branches of the root locus.
- R2 : Every point on the real axis between $-\infty$ and -8 and between -4 and -2 is a part of the root locus.
- R3 : It will suffice to find one of the two branches, the second one is its mirror image.
- R4 : The branches commence at the points -8 and -2 . As there is one zero present at -4 , one branch will terminate at the point -4 and the other branch will terminate at infinity.

This practically concludes the construction of root-locus plot. As there are no break-away points (two poles never meet each other), we are ready to draw the complete diagram.

```
G3 = tf([1 4], [1 10 16]);
figure; rlocus(G3)
iam_save_pdf('plt_rl_g3.pdf', 11, 11)
```

Influence of the increasing value of a gain of the controller:

- on stability: system will be always (theoretically) stable,
- on damping ratio: response will always be overdamped (non-oscillatory),
- on response rate of the system: system response will always speed up, yet there will be a limit to response settling rate.

$$(d) G_4(s) = \frac{s+2}{s^2+12s+32}$$

- R1 : There will be two branches of the root locus.
- R2 : Every point on the real axis between $-\infty$ and the point -8 and between the points -4 and -2 is a part of the root locus.
- R3 : It will suffice to find one of the two branches, the second one is its mirror image.
- R4 : The branches commence at -8 and -4 . As there is one zero present at the point -2 , one branch will terminate at -2 and the other branch will terminate at infinity.

This practically concludes the construction of root-locus plot. As there are no break-away points (two poles never meet each other), we are ready to draw the complete diagram.

```
G4 = tf([1 2], [1 12 32]);
figure; rlocus(G4)
iam_save_pdf('plt_rl_g4.pdf', 11, 11)
```

Influence of the increasing value of a gain of the controller:

- on stability: system will be always (theoretically) stable,
- on damping ratio: response will always be overdamped (non-oscillatory),
- on response rate of the system: system immediate response will always speed up, yet the response settling rate will slow down.

$$(e) G_5(s) = \frac{1}{s^3 + 14s^2 + 56s + 64}$$

R1 : There will be three branches of the root locus.

R2 : Every point on the real axis between $-\infty$ and the point -8 and between the points -4 and -2 is part of the root locus.

R3 : It will suffice to find one of the two branches with complex values, the second one is its mirror image. The third branch will reside on the real axis.

R4 : The branches commence at the points -8 , -4 and -2 . As there are no zeros present, all three branches will terminate at infinity.

R5 : Asymptotes start from $\sigma_w = (-8 - 4 - 2)/3 = -4.6667$ with angles $\phi_1 = (\pi + 2\pi)/3 = \pi$, $\phi_2 = (\pi + 4\pi)/3 = 5\pi/3$, $\phi_3 = (\pi + 6\pi)/3 = \pi/3$

R6 : $1/(\sigma_k + 8) + 1/(\sigma_k + 4) + 1/(\sigma_k + 2) = 0$, $\sigma_1 = -6.4305$ (not valid; part of the root locus for $K < 0$), $\sigma_2 = -2.9028$.

We are ready to sketch the important parts of the root-locus plot. The actual diagram can be plotted using the commands below.

```
G5 = tf(1, [1 14 56 64]);
figure; rlocus(G5)
iam_save_pdf('plt_rl_g5.pdf', 11, 11)
```

We can also find the critical value of a gain K_c and the corresponding critical frequency ω_c from R8. It gives a complex-valued equation that can be separated into real and imaginary parts:

$$0 = K_c - \omega_c^3 j - 14\omega_c^2 + 56\omega_c j + 64, \quad (7.18)$$

$$\text{Re: } 0 = K_c - 14\omega_c^2 + 64, \quad (7.19)$$

$$\text{Im: } 0 = -\omega_c^3 + 56\omega_c, \quad (7.20)$$

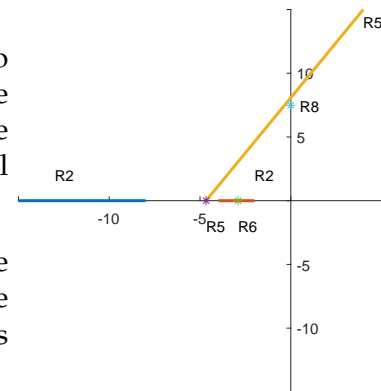


Figure 7.3: Root-locus sketch for $G_5(s)$.

and the resulting values (for $K_c > 0$) of $K_c = 720$, $\omega_c = \sqrt{56}$. Note, that the point $(0, \omega_c)$ is below the asymptote, not on it.

Another possibility to find the critical gain is to use the Routh reduction algorithm for stability and look for condition when any of the coefficients (functions of K_c) is zero.

Influence of the increasing value of a gain of the controller:

- on stability: system will be always stable unless the gain exceeds its critical value K_c ,
- on damping ratio: response will exhibit overdamped behavior only for low range of gain values,
- on response rate of the system: system immediate response will speed up always; for higher range of gain values, the response settling will become slower and eventually settling will disappear (instability).

7.2.1.2 Verify some of the conclusions made. Calculate controller gains at break-away points. Plot a closed-loop response of the system characterised by $G_2(s)$ with a proportional controller. Next, use a controller with a gain $K_p = 0.1$, $K_p = 3$, and $K_p = 500$.

We use the break-away points from R6 and we calculate the gains from R7

$$K = -\frac{\prod_{i=1}^n (\sigma_k - p_i)}{\prod_{i=1}^m (\sigma_k - z_i)} = -\frac{(\sigma_k + 2)(\sigma_k + 4)}{\sigma_k + 8} = (0.2, 19.8). \quad (7.21)$$

MATLAB code is shown below and the simulation results in Figure 7.5. Note the time axis of each of the plots.

```
figure
Kp = 0.1; G = Kp*G2/(1+Kp*G2);
step(G); title('Step Response, Kp = 0.1');
iam_save_pdf('plt_rep_g3Kp01.pdf', 20, 12)

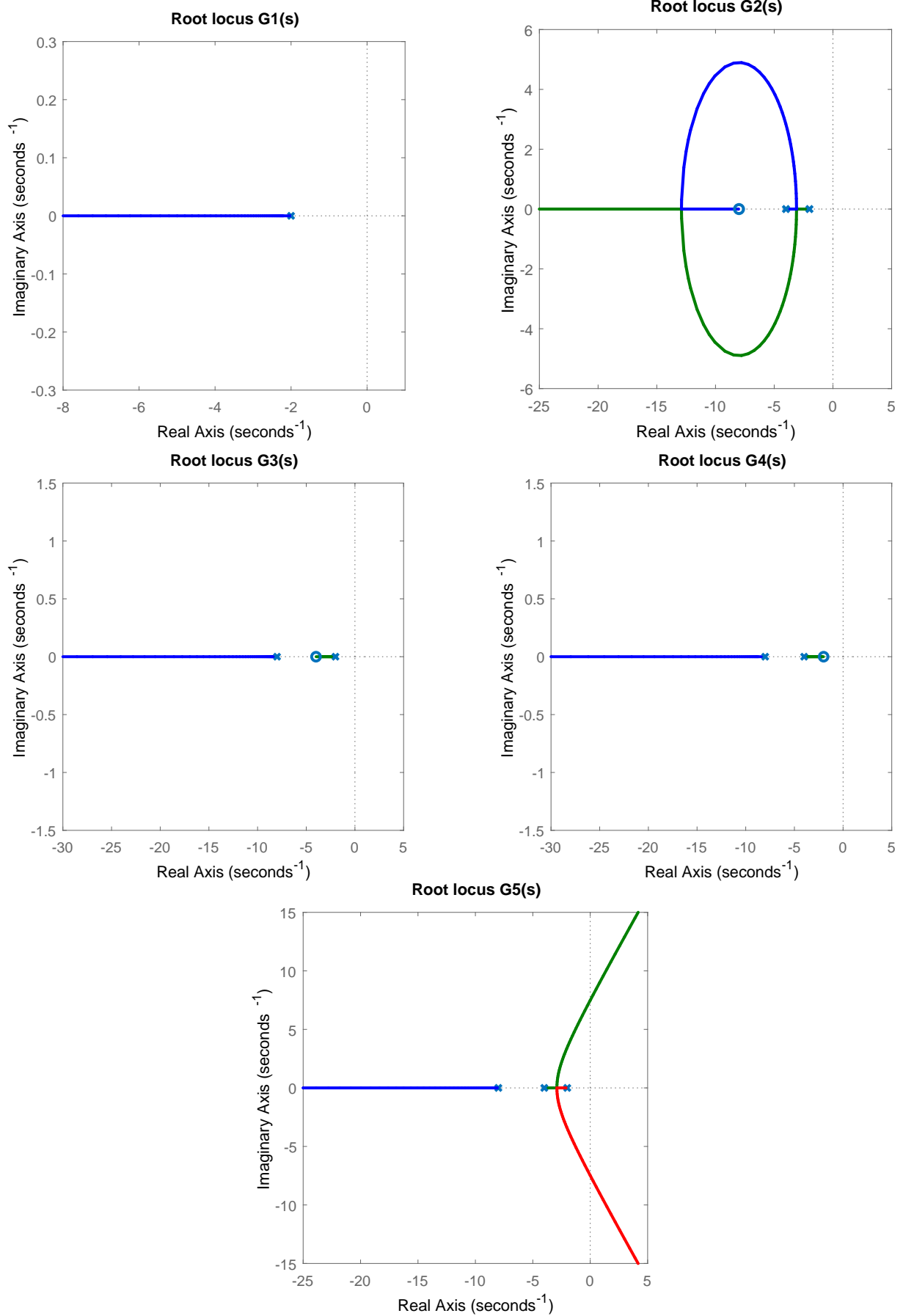
figure
Kp = 3; G = Kp*G2/(1+Kp*G2);
step(G); title('Step Response, Kp = 3');
iam_save_pdf('plt_rep_g3Kp3.pdf', 20, 12)

figure
Kp = 500; G = Kp*G2/(1+Kp*G2);
step(G); title('Step Response, Kp = 500');
iam_save_pdf('plt_rep_g3Kp500.pdf', 20, 12)
```

7.2.2 Two Tanks with Interaction

We study the dynamic properties of two tanks with interaction that we modelled in Example 1.2.2. The process parameters are as follows: $q_{0,1}^s = 0.9 \text{ m}^3/\text{s}$, $q_{0,2}^s = 0 \text{ m}^3/\text{s}$, $F_1 = 0.5 \text{ m}^2$, $F_2 = 0.6 \text{ m}^2$, $k_{11} = 0.8 \text{ m}^{2.5}/\text{s}$, and $k_{22} = 0.5 \text{ m}^{2.5}/\text{s}$.

The transfer function between the output h_2 and manipulated input q_{01} (in deviation variables)

**Figure 7.4:** Root-locus plot for $G_1 - G_5$.

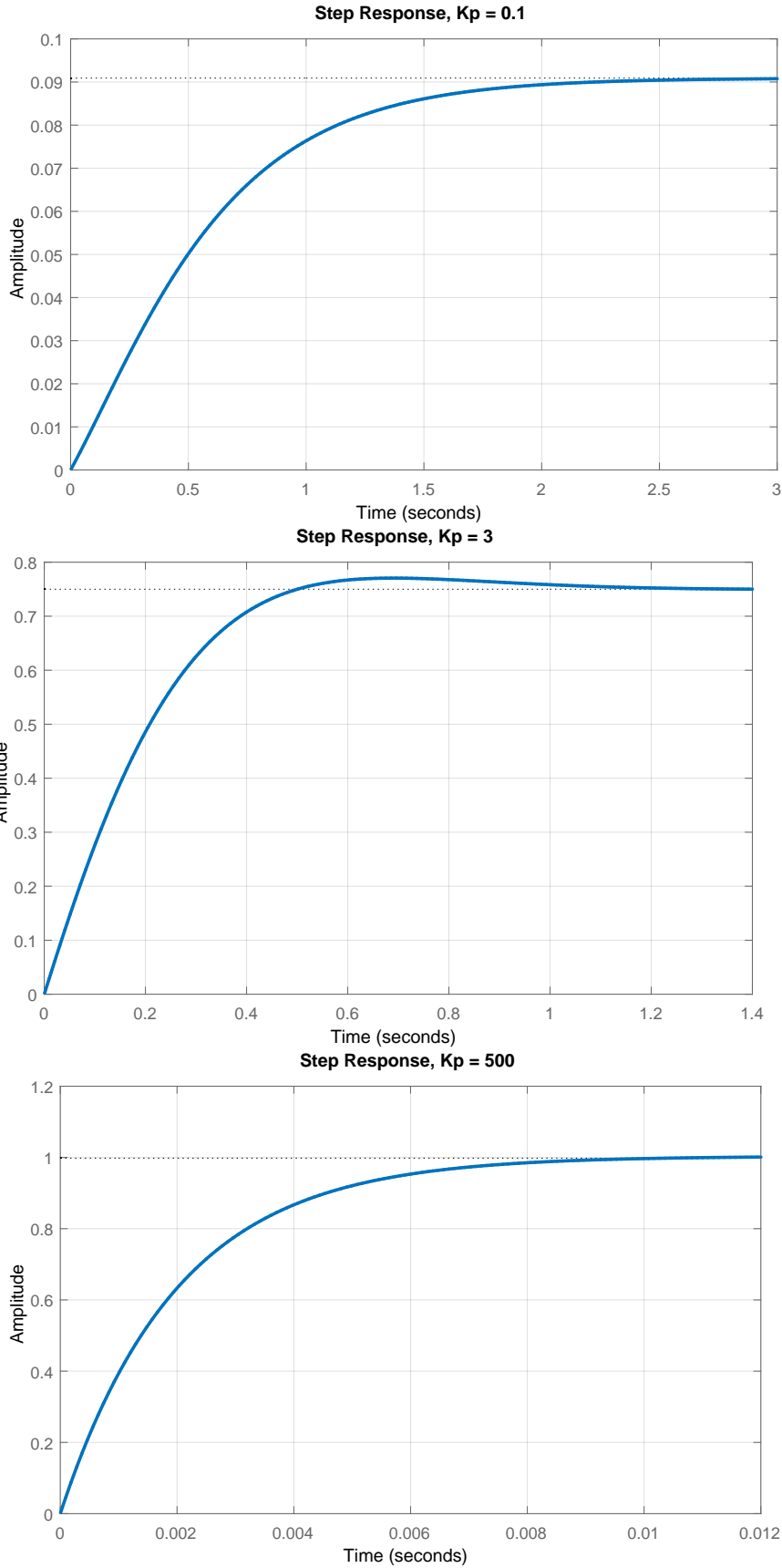


Figure 7.5: Feedback control of the process described by $G_2(s)$ with a proportional controller; $K_p = 0.1$, $K_p = 3$, and $K_p = 500$.

is given by (2.75)

$$G_1(s) = \frac{Y(s)}{U_1(s)} = \frac{k_1}{F_1 F_2 s^2 + (k_1 F_1 + k_1 F_2 + k_2 F_1)s + k_1 k_2} \quad (7.22)$$

$$= \frac{7.2}{6.075s^2 + 9.326s + 1} = \frac{1.19}{(s + 1.42)(s + 0.12)}. \quad (7.23)$$

7.2.2.1 Assume that a proportional controller manipulates the input variable. Find the maximum K_p using root locus so that the closed-loop system remains overdamped.

The process contains two stable real poles and root-locus plot (similar to the one in Figure 7.1 left) will exhibit an overdamped response for small values of K_p with the closed-loop poles coming closer to each other. The double pole can be found using rule R6 and the open-loop poles $p_1 = -1.42$ and $p_2 = -0.12$:

$$\frac{1}{\sigma_k - p_1} + \frac{1}{\sigma_k - p_2} = 0, \quad \sigma_1 = -0.77. \quad (7.24)$$

The corresponding value of the controller gain can be found using rule R7:

$$K = -(\sigma_1 - p_1)(\sigma_1 - p_2) = 0.4246, \quad K_{p1} = \frac{K}{1.19} = 0.36. \quad (7.25)$$

7.2.2.2 Assume again that a proportional controller manipulates the input variable. Find a value of K_p using a root-locus plot such that the closed-loop system exhibits a 20% overshoot.

The overshoot depends only on the damping ratio ζ (3.55):

$$\zeta = \frac{|\ln e_{\max}|}{\sqrt{\pi^2 + \ln^2 e_{\max}}} = 0.46. \quad (7.26)$$

We know that the damping ratio determines the angle of the poles and we are looking for the intersection of the root-locus plot with the line from the origin under this angle. The slope is given by

$$\tan \varphi = \frac{\sqrt{1 - \zeta^2}}{\zeta} = 1.95. \quad (7.27)$$

The intersection with the root locus gives the roots $s = -0.77 \pm 1.5j$ and the desired gain

$$K = -(s - p_1)(s - p_2) = 2.67, \quad K_{p2} = \frac{K}{1.19} = 2.25. \quad (7.28)$$

7.2.2.3 Try to reduce the steady-state error by adding a lag compensator to the controller while keeping the other performance criteria the same.

A lag compensator is of the form

$$G_a(s) = \frac{s - z}{s - p}, \quad p \ll z. \quad (7.29)$$

In our case, the lag zero z can be chosen as the dominant stable pole p_2 of the

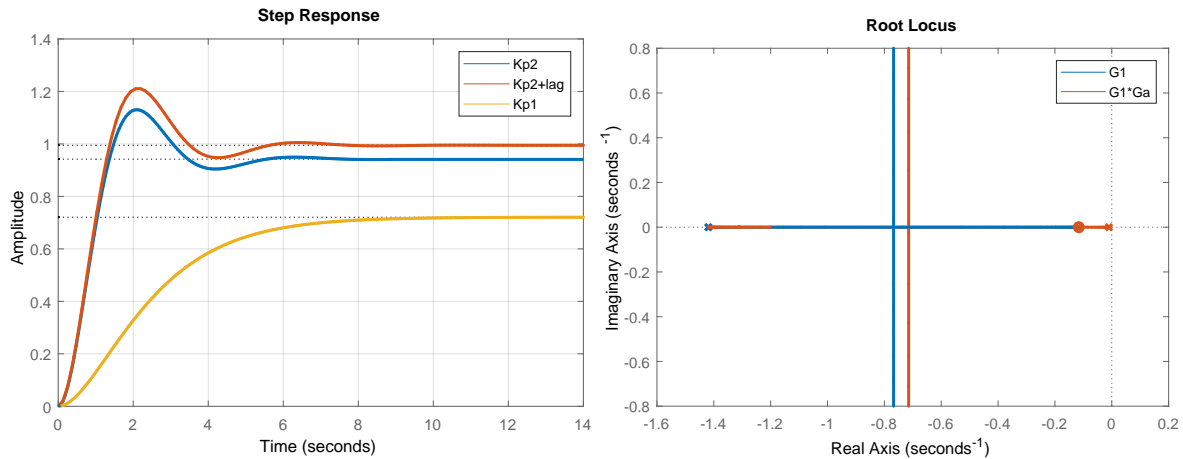


Figure 7.6: (left) Feedback control and (right) root-locus plot of the two tanks with interaction.

process and the pole of the compensator as $p = -0.01$. The resulting permanent tracking error was reduced from 6% to less than 1%.

Step responses of the closed-loop system with all three controllers and the root-locus plot of the original and lag-compensated process are shown in Figure 7.6. The root-locus plots are very similar and the same holds for the step responses.

7.2.3 Two Tanks with a Pump

We study the dynamic properties of two tanks with a pump that we modelled in Example 1.2.3. The process parameters are as follows: $q_{0,1}^s = 0.9 \text{ m}^3/\text{s}$, $q_{0,2}^s = 0.9 \text{ m}^3/\text{s}$, $F_1 = 0.5 \text{ m}^2$, $F_2 = 0.6 \text{ m}^2$, $k_{11} = 0.8 \text{ m}^{2.5}/\text{s}$, and $k_{22} = 0.5 \text{ m}^{2.5}/\text{s}$.

The transfer function between the output h_2 and manipulated input q_{02} (in deviation variables) is given by (2.103)

$$G_1(s) = \frac{Y(s)}{U_1(s)} = \frac{k_1}{F_1 F_2 s^2 + (k_1 F_1 + k_1 F_2)s} = \frac{1.185}{s^2 + 1.30s}. \quad (7.30)$$

7.2.3.1 Assume that a proportional controller manipulates the input variable. Find K_p using the root-locus plot such that the settling time is about 7 s.

The process contains one pole at the origin and one stable real pole. The root-locus plot will be similar to the previous one and a stable closed-loop system will result for any choice of K_p . The settling time is about $4/(\zeta\omega)$ with $\zeta\omega$ being the real part of the (dominant) closed-loop pole. Therefore

$$s = -\zeta\omega = -\frac{4}{T_{0.02}} = -0.57, \quad (7.31)$$

$$K = -(s - p_1)(s - p_2) = 0.41, \quad (7.32)$$

$$K_p = \frac{K}{1.19} = 0.35. \quad (7.33)$$

The position of the pole on the root-locus plot suggests overdamped response.

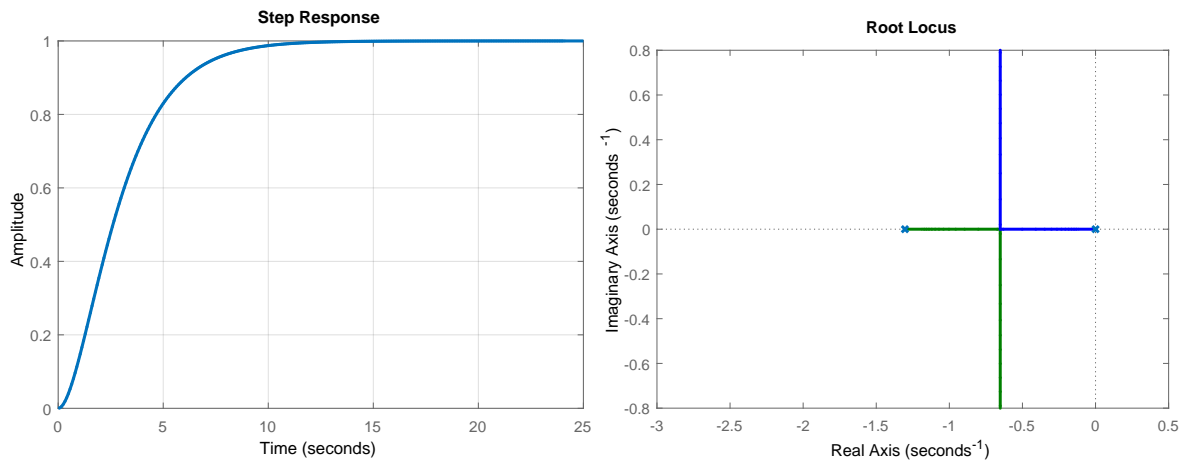


Figure 7.7: (left) Feedback control and (right) root-locus plot of the two tanks with a pump.

A step response of the closed-loop system and the root-locus plot are shown in Figure 7.7.

7.2.3.2 From the root-locus plot, we observe that the underdamped poles have real values of approximately -0.7 which has an influence on the settling time. If we choose an overshoot of 5% we get $K_{p1} = 0.75$. Let us now design a lead compensator that will move the underdamped poles to the right and reduce the settling time. The compensator is of the form

$$G_e(s) = \frac{s - z}{s - p}, \quad p > z. \quad (7.34)$$

Design the compensator so that the step response of the closed-loop system has the same overshoot but the settling time will be smaller.

We will place the compensator zero at the pole p_2 and the pole $p = 2p_2$. This will move the vertical part of the root-locus branch of the underdamped poles to the left (Figure 7.8, right) and we will find a controller gain that will provide the same overshoot: $K_{p2} = 3$.

Figure 7.8 (left) compares the step response of the uncompensated and compensated closed-loop system. We can see that the settling time has been halved.

7.3 Practice Examples

7.3.1 Warm-up Examples

- Plot roots of the closed-loop system with a proportional controller. The controlled system is described with $G_2(s)$ from Example 7.2.1. Consider 17 controller gains from the logarithmically spaced interval (0.01, 100).

See Section 7.5.

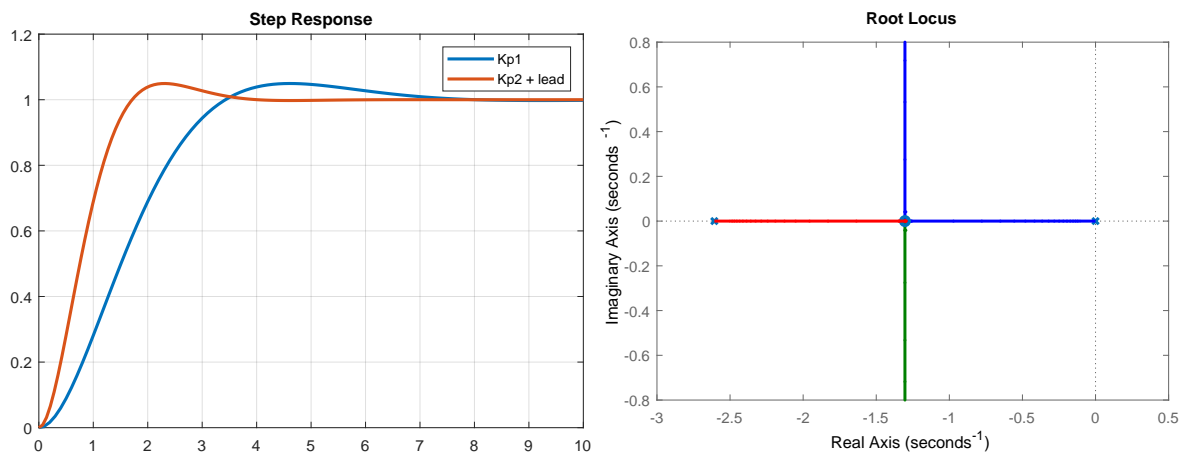


Figure 7.8: (left) Feedback control with/without a lead compensator and (right) root-locus plot with lead compensator of two tanks with a pump.

- Is it possible to find the gain of a proportional controller that will stabilise the following system $G(s) = \frac{1}{(s+2)(s-1)}$?

No. Examination of the root-locus plot (`rlocus(tf(1, [1, -1, -2]))`) shows that at least one closed-loop pole is always unstable.

7.3.2 Two Tanks in Series: Root-Locus Construction

We study a system with two tanks in series for which we have derived the transfer function matrix in Section 2.3.2.

7.3.2.1 A proportional controller is chosen to adjust the input variable. Discuss the influence of the value of the gain for stability, damping ratio, and response rate of the system. Use the root-locus plot to support your arguments.

R1 : There will be two branches of the root locus.

R2 : Every point on the real axis between the points -1 and -0.2 is a part of the root locus.

R3 : It will suffice to find one of the two branches, the second one is its mirror image.

R4 : The branches commence at the points -1 and -0.2 . As there are no zeros present in the system, both branches will terminate at infinity.

R5 : $\sigma_w = (-1 - 0.2)/2 = -0.6$, $\phi_1 = (\pi + 2\pi)/2 = 3\pi/2$, $\phi_2 = (\pi + 4\pi)/2 = 5\pi/2$.

R6 : $1/(\sigma_k + 1) + 1/(\sigma_k + 0.2) = 0 \rightarrow (\sigma_k + 0.2) + (\sigma_k + 1) = 0 \rightarrow \sigma_1 = -0.6$.

We are ready to construct the root-locus plot which is shown in Figure 7.9.

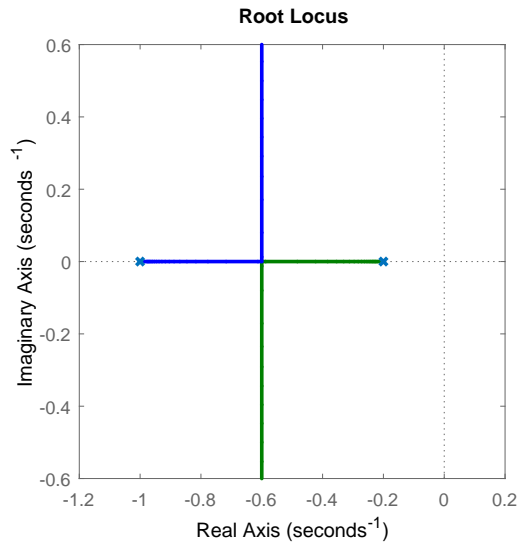


Figure 7.9: Root-locus plot for two tanks in series.

```
k1 = 1; k2 = 1; F1 = 5; F2 = 1;
G = tf(k1/F1/F2, [1 (k1/F1+k2/F2) k1*k2/F1/F2]);
figure; rlocus(G)
iam_save_pdf('plt_rl_tanks.pdf', 11, 11)
```

Influence of the increasing value of a gain of the controller:

- on stability: system will be always (theoretically) stable,
- on damping ratio: response will always be overdamped only for low values of gain,
- on response rate of the system: system response rate will increase in the low range of gain values; for middle to high range of gain values the response settling rate will become approximately constant.

7.3.3 A One-Dimensional Car: Root-Locus Construction

We study the movement of a car that we modelled in Section 1.3.1.

7.3.3.1 A proportional controller is chosen to adjust the input variable. Discuss the influence of the value of the gain for stability, periodicity, and response rate of the system. Use the root-locus plot to support your arguments.

R1 : There will be two branches as there are two poles at the origin.

R2 : As there are two poles at the origin, none of the points on the real axis is part of the root locus.

R3 : It will suffice to find one of the two branches, the second one is its mirror image.

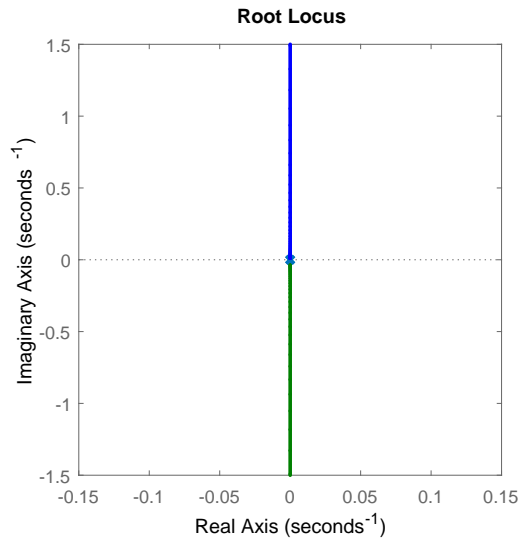


Figure 7.10: Root-locus plot for a one-dimensional car.

R4 : As there are no zeros, the two branches will terminate at infinity.

R5 : $\sigma_w = 0$, $\phi_1 = (\pi + 2\pi)/2 = 3\pi/2$, $\phi_2 = (\pi + 4\pi)/2 = 5\pi/2$.

R6 : This rule does not apply as there are no points on the real axis (R2).

This concludes the construction of root locus as we are now ready to draw the complete diagram as shown in Figure 7.10.

Influence of the increasing value of a gain of the controller:

- on stability: system will be always unstable,
- on damping ratio: response will always be underdamped,
- on response rate of the system: system response rate will not be affected by the gain values.

7.3.4 Poles and Zeros of a PID Controller

Consider the transfer function of a general PID controller

$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + \frac{T_d s}{T_n s + 1} \right) = K_p + \frac{K_p}{T_i} \frac{1}{s} + K_p T_d \frac{s}{T_n s + 1}. \quad (7.35)$$

7.3.4.1 Identify the poles and zeros of the controller. Discuss the admissible range of values of the poles and zeros.

The transfer function of the controller can be expressed as

$$G_c(s) = K_p \frac{(T_d T_i + T_i T_n) s^2 + (T_i + T_n) s + 1}{T_i s (T_n s + 1)}. \quad (7.36)$$

- Poles occur at: 0 and $-1/T_n$,
- Zeros occur at: $-\frac{T_i+T_n \pm \sqrt{(T_i+T_n)^2 - 4(T_d+T_n)T_i}}{2(T_d T_i + T_i T_n)}$,

which means that two zeros and one pole can be freely selected (zeros can even be complex conjugate) based on the values of T_i , T_d , and T_n .

7.3.4.2 Identify the number of poles and zeros of the different possible controller structures (P, I, PI, D, causal D, ...).

controller	#poles	#zeros
P		
I		
PI		
...		
PID _c		

controller	#poles	#zeros
P	0	0
I	1	0
PI	1	1
D	0	1
D _c	1	1
PD	0	1
PD _c	1	1
PID	1	2
PID _c	2	2

7.3.5 A One-Dimensional Car: PID Root-Locus Design

We study the movement of a car that we modelled in Section 1.3.1.

7.3.5.1 Plot the root-locus plots for all the controllers from the previous task. Use $T_i = 1$ s, $T_d = 1$ s, and $T_n = 0.1$ s.

```
close all
g = tf(1, [1 0 0]); % double integrator system
Ti = 1; Td = 1; Tn = 0.1;

subplot(3,3,1); gc = 1; rlocus(g*gc); title('P controller')
subplot(3,3,2); gc = tf(1, [Ti 0]); rlocus(g*gc); title('I controller')
subplot(3,3,3); gc = tf([Ti 1], [Ti 0]); rlocus(g*gc); title('PI
controller')
subplot(3,3,4); gc = tf([Td 0], 1); rlocus(g*gc); title('D controller')
subplot(3,3,5); gc = tf([Td 0], [Tn 1]); rlocus(g*gc); title('D_c
```

```

    controller')
subplot(3,3,6); gc = tf([Td 1], 1); rlocus(g*gc); title('PD controller'
)
subplot(3,3,7); gc = tf([Tn+Td 1], [Tn 1]); rlocus(g*gc); title('PD_c
controller')
subplot(3,3,8); gc = tf([Td*Ti Ti 1], [Tn 0]); rlocus(g*gc); title('PID
controller')
subplot(3,3,9); gc = tf([(Tn+Td)*Ti Tn+Ti 1], [Ti*Tn Ti 0]); rlocus(g*
gc); title('PID_c controller')
h1=findall(gcf); hline=findobj(gcf, 'Type', 'line');
for i=1:length(hline)
    hline(i).LineWidth=2;
end
iam_save_pdf('pid_rl_dinteg.pdf', 22, 22);

```

The resulting root-locus-plot matrix is shown in Figure 7.11.

7.3.5.2 Which of the controllers can you use to successfully adjust the car's acceleration and control its position? Test the best controller in a simulation. Plot both the control output and control input and discuss the results.

All the controllers with nonzero derivative part can be used to control the car. We will test the PD_c controller with a gain of 2.66. The corresponding Simulink model is shown in Figure 7.12 and the closed-loop results in Figure 7.13.

```

sim integ_pid

subplot(1,2,1); plot(y(:,1), y(:,2)); xlabel('time'); ylabel('y(t)');
    grid on
subplot(1,2,2); plot(u(:,1), u(:,2)); xlabel('time'); ylabel('u(t)');
    grid on

h1=findall(gcf); hline=findobj(gcf, 'Type', 'line');
for i=1:length(hline)
    hline(i).LineWidth=2;
end
iam_save_pdf('plt_ctrl_pdr_dinteg.pdf', 30, 11);

```

7.3.6 Two Tanks in Series: PID Root-Locus Design

We study a system with two tanks in series for which we have derived the transfer function matrix in Section 2.3.2.

7.3.6.1 Tune a real PID controller such that you compensate:

- the slowest pole,
- the fastest pole,
- both the poles.

Plot the corresponding root-locus plot. Plot the step response of the closed-loop system. Make a comparison with a corresponding proportional controller. Try controllers with different gains: $K_p = 5$ and $K_p = 50$.

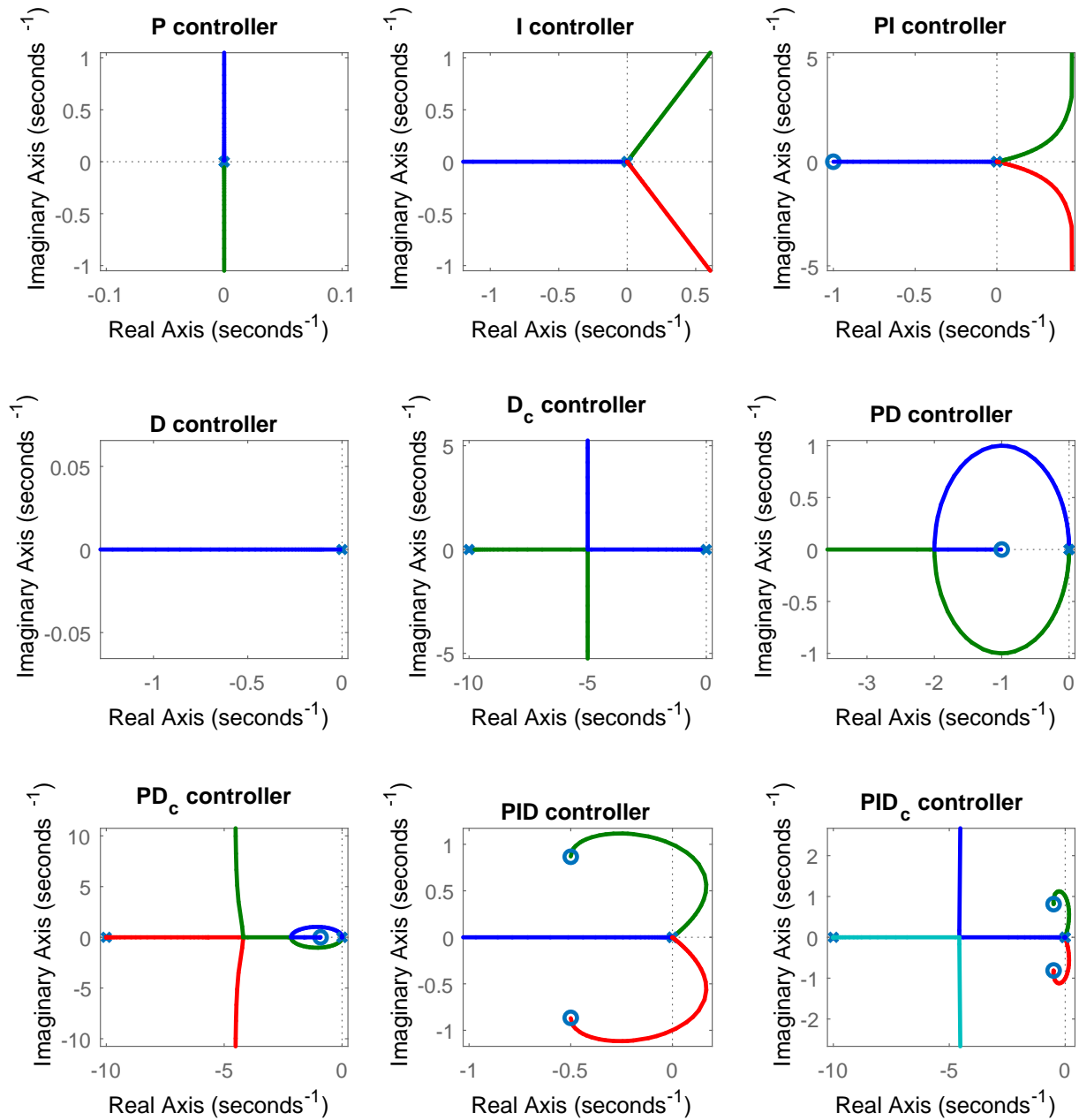


Figure 7.11: Root-locus plot of a double integrator with different PID controllers.

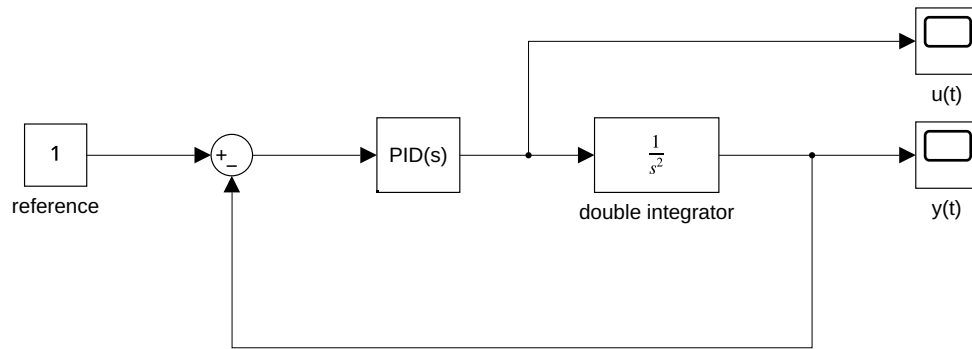


Figure 7.12: A Simulink model for double-integrator PID control.

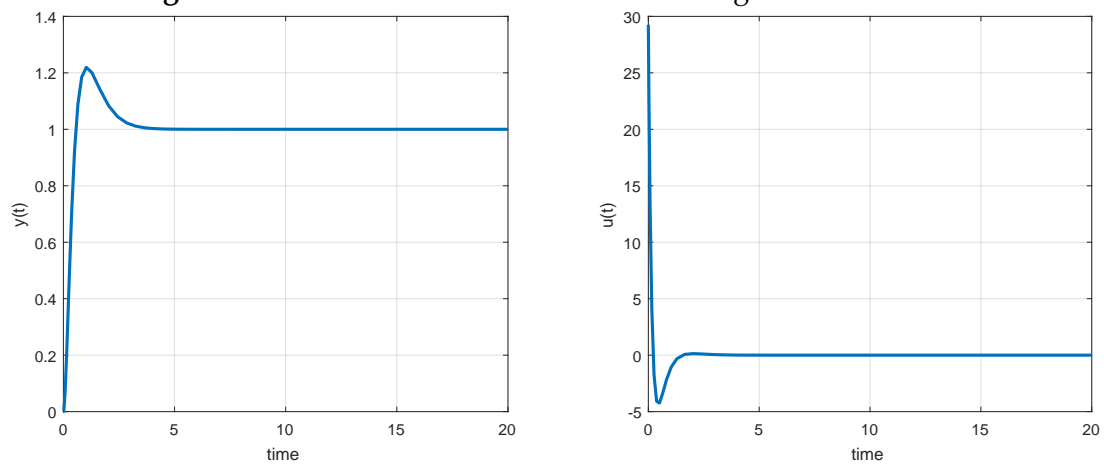


Figure 7.13: Results of PID control of a double integrator (left: position, right: acceleration).

```

syms Ti Td Tn s p1 p2

collect(1 + 1 / Ti / s + Td * s / (Tn * s + 1))

den = (Ti * Tn * s^2 + Ti * s);
num = (Td * Ti + Ti * Tn) * s^2 + (Ti + Tn) * s + 1;

solve(num, s)

eqp1 = p1 + (Ti + Tn + (Ti^2 - 2 * Ti * Tn - 4 * Td * Ti + Tn^2)^(1/2))
    / (2 * (Td * Ti + Ti * Tn));
eqp2 = p2 + (Ti + Tn - (Ti^2 - 2 * Ti * Tn - 4 * Td * Ti + Tn^2)^(1/2))
    / (2 * (Td * Ti + Ti * Tn));

sol = solve(eqp1, eqp2, Ti, Td)
sol.Ti, sol.Td
close all
% G(s) = k1/F1/F2/(s+k1/F1)/(s + k2/F2);
k1 = 1; k2 = 1; F1 = 5; F2 = 1;
g = tf(k1/F1/F2, poly([-k1/F1 -k2/F2]));
axr1 = [-2 0.2 -0.5 0.5];
% P controller
subplot(4, 3, 1)
gc = 1;
rlocus(g*gc)
axis(axr1); title('Root Locus, P')

```



```

subplot(4, 3, 2)
K = 5;
step(gc*K*g/(1+gc*K*g)); title('P, K_p=5'); grid on
v=axis; axis([v(1), v(2), 0, 2])

subplot(4, 3, 3)
K = 50;
step(gc*K*g/(1+gc*K*g)); title('P, K_p=50'); grid on
v=axis; axis([v(1), v(2), 0, 2])

% PI controller
subplot(4, 3, 4)
Ti = 1/(k1/F1);
gc = tf([Ti 1],[Ti 0]);
rlocus(g*gc); title('Root Locus, PI_1')
axis(axrl)

subplot(4, 3, 5)
K = 5;
step(gc*K*g/(1+gc*K*g)); title('PI_1, K_p=5'); grid on
v=axis; axis([v(1), v(2), 0, 2])

subplot(4, 3, 6)
K = 50;
step(gc*K*g/(1+gc*K*g)); title('PI_1, K_p=50'); grid on
v=axis; axis([v(1), v(2), 0, 2])

% PI controller
subplot(4, 3, 7)
Ti = 1/(k2/F2);
gc = tf([Ti 1],[Ti 0]);
rlocus(g*gc); title('Root Locus, PI_2')
axis(axrl)

subplot(4, 3, 8)
K = 5;
step(gc*K*g/(1+gc*K*g)); title('PI_2, K_p=5'); grid on
v=axis; axis([v(1), v(2), 0, 2])

subplot(4, 3, 9)
K = 50;
step(gc*K*g/(1+gc*K*g)); title('PI_2, K_p=50'); grid on
v=axis; axis([v(1), v(2), 0, 2])

% PID controller
subplot(4, 3, 10)
p1 = -k1/F1; p2 = -k2/F2;
Tn = 1/2;
Ti = -(p1 + p2 + Tn*p1*p2)/(p1*p2);
Td = -(Tn*p1 + Tn*p2 + Tn^2*p1*p2 + 1)/(p1 + p2 + Tn*p1*p2);
gc = tf([Td*Ti+Ti*Tn Ti+Tn 1],[Ti*Tn Ti 0]);
rlocus(g*gc); title('Root Locus, PID')
axis(axrl)

subplot(4, 3, 11)
K = 5;
step(gc*K*g/(1+gc*K*g)); title('PID, K_p=5'); grid on
v=axis; axis([v(1), v(2), 0, 2])

subplot(4, 3, 12)

```

```

K = 50;
step(gc*K*g/(1+gc*K*g)); title('PID, K_p=50'); grid on
v=axis; axis([v(1), v(2), 0, 2])
hl=findall(gcf); hline=findobj(gcf, 'Type', 'line');
for i=1:length(hline)
    hline(i).LineWidth=2;
end

iam_save_pdf('plt_rl_ctrl_tanks_pid.pdf', 25, 30);

```

The results are shown in Figure 7.14.

7.4 Problems

Problem. 7.1: Fourth-order system

Construct the root-locus plot for the open-loop system

$$G_o(s) = \frac{Y(s)}{E(s)} = \frac{K(s+1)}{s(s+2)(s+3)^2}, \quad (7.37)$$

and determine the critical gain K_c .

$$K_c = 96.1$$

Problem. 7.2: Two tanks with a pump

We derived the transfer function for two tanks with a pump

$$G_{y,u_1}(s) = \frac{1}{0.8437s^2 + 1.1s}. \quad (7.38)$$

Construct the root-locus plot if this process is controlled by

- P controller
- PI controller of the form $K(1 + 5/s)$
- PI controller of the form $K(1 + 0.5/s)$

What can be said about the stability of the closed-loop system with these controllers? What value of integral time constant T_i separates stable and unstable configurations?

The second controller cannot stabilise the process. The critical integral time constant is given as the reciprocal value of the stable process pole.

Problem. 7.3: System with two poles and one zero

Consider a system with two stable poles (either real or complex conjugate) and one stable zero. Sketch root-locus plots for all possible different scenarios.

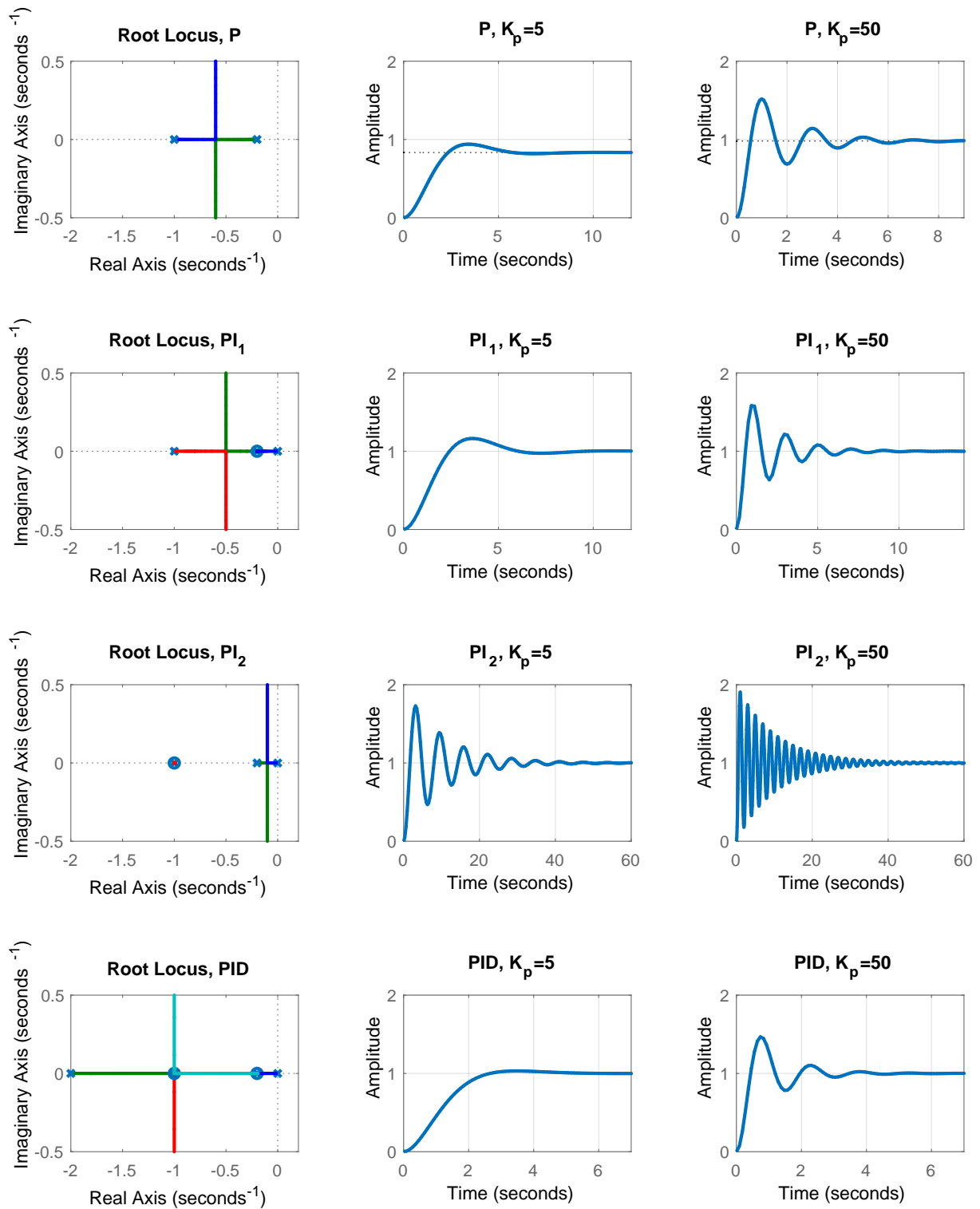


Figure 7.14: Different PID controllers with the corresponding root loci (left-hand column) and simulation results (centre and right-hand column) for two-tank control.

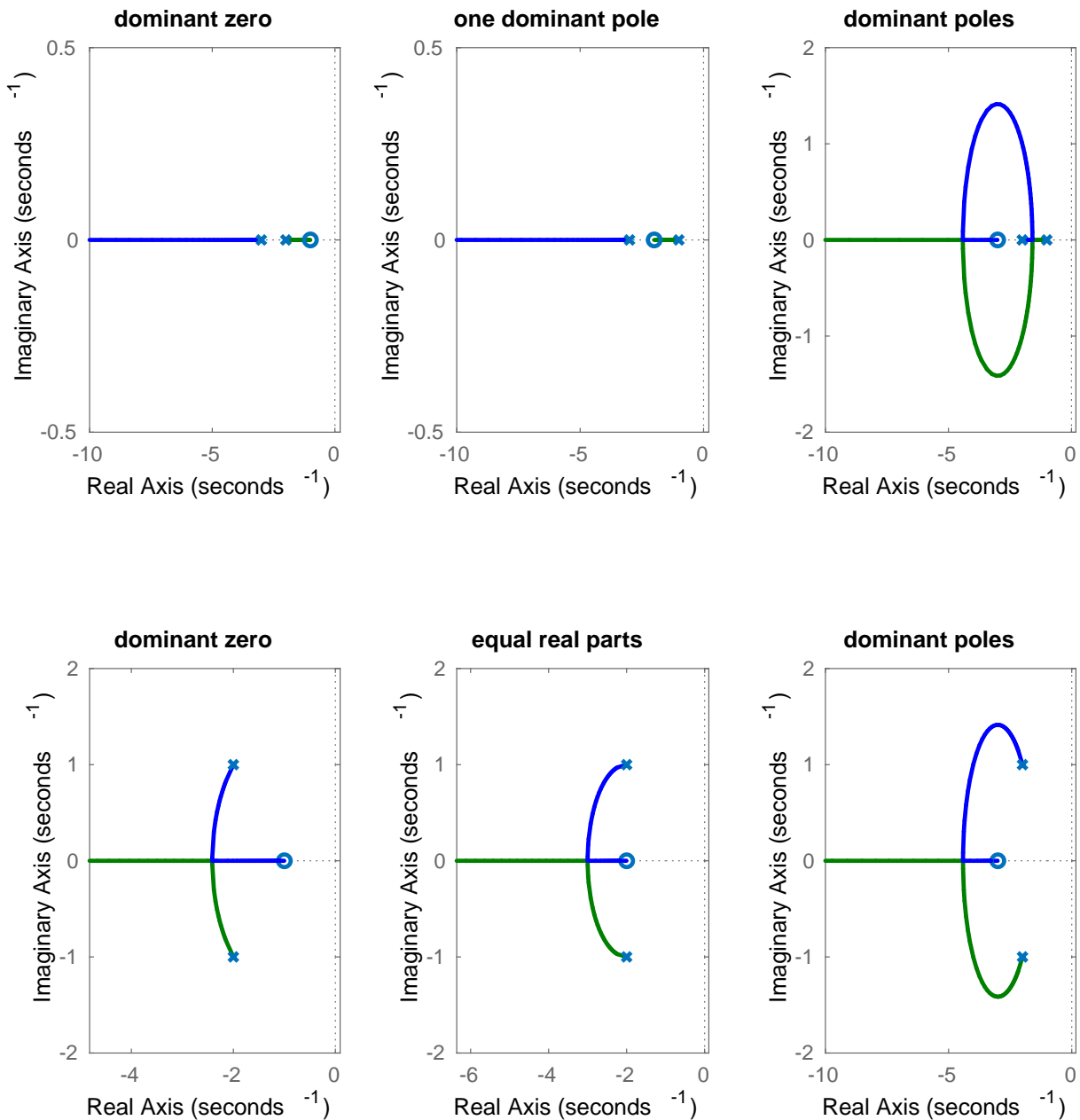


Figure 7.15: Root loci corresponding to different locations of real/complex poles and stable zero.

Solution is visualised in Fig. 7.15.

Problem. 7.4: Overshoot

Consider a system with a transfer function of the form

$$G(s) = \frac{18}{(s+1)(s+3)(s+6)}. \quad (7.39)$$

Design a proportional controller giving an overshoot of 10% using root-locus plot. Try to fix the proportional gain and to tune the derivative time constant of a PD controller to

improve the settling time.

Proportional controller: $K_p = 1.4$.

Problem. 7.5: Lead compensation

Consider the system from the previous problem with a transfer function of the form

$$G(s) = \frac{18}{(s+1)(s+3)(s+6)}. \quad (7.40)$$

Try to approximate the resulting PD controller by the controller of the form

$$G_c(s) = K \frac{s+z}{z+p}, \quad (7.41)$$

where $z \ll p$ (with the same zero and the pole ten times smaller). Plot the root-locus plots for both controllers.

Problem. 7.6: Lag compensation

Consider a system with a transfer function of the form

$$G(s) = \frac{4}{(s+1)(s+4)}. \quad (7.42)$$

Design a PI controller with a stable zero $z = -0.1$ using the root-locus plot that will result in damping ratio $\zeta = 0.7$.

Alternatively, try to find a controller of the form

$$G_c(s) = K \frac{s+z}{z+p}, \quad (7.43)$$

with the same damping ratio $\zeta = 0.7$ and with $z \gg p$ that approximates the PI controller (for example $p = 0.01$, $z = 0.1$). Such a controller (lag compensator) does not suffer from windup effects. What is the difference in the steady-state error between the proportional controller and the lag controller?

Damping ratio determines the angle of the closed-loop poles to the origin. Its intersection with the root-locus plot provides the desired gain K .

7.5 MATLAB Snippets

- Basic root-locus plot, obtaining root-locus values as vectors or matrices

```
G2 = tf([1 8], [1 6 8]);
rlocus(G2);
[roots, gains] = rlocus(G2);
```

- Root-locus plot with damping ratio lines and customisation

```
s=tf('s'); G2=(s + 8) / ((s + 2) * (s + 4));
h = rlocusplot(G2);
p = getoptions(h);
p.grid = 'on';
p.XLimMode = 'manual'; p.xlim = {[ -20 4]};
p.title.FontSize = 12; p.title.String = 'Root Locus G2(s)';
setoptions(h,p);
```

- Root-locus plot with emphasis of the poles from 17 controller gains from the logarithmically spaced interval (0.01, 100) (Section 7.3.1)

```
G2 = tf([1 8], [1 6 8]); K = logspace(-2, 2, 17);
poles=rlocus(G2, K);
plot(real(poles), imag(poles), 'rx', 'Markersize', 10); hold on
rlocus(G2)
hold off
```

- Read the gain, pole, damping ratio, overshoot, and frequency from root-locus plot: click on the desired point of the `rlocus` results.
- Store selected gain and poles from root-locus plot

```
G2 = tf([1 8], [1 6 8]);
rlocus(G2);
[k, poles] = rlocfind(G2); % click the desired poles
```

- Plot a desired damping ratio and frequency on the root-locus plot

```
zeros = -8; poles = [-2 -4]; gain = 1;
G2 = zpk(zeros, poles, gain);
rlocus(G2);
omega = 10; zeta = 0.8;
sgrid(zeta, omega)
```

- Interactive plots of closed-loop system: step, root-locus plot, and Bode diagrams. Drag the closed-loop poles in the root-locus plot to observe changes in other plots.

```
G2 = tf([1 8], [1 6 8]);
controlSystemDesigner(G2);
```

7.6 Python Snippets

- Basic root-locus plot, get values into vectors or matrices

```
from control.matlab import *
import matplotlib.pyplot as plt
G2 = tf([1, 8], [1, 6, 8])
roots, gains = rlocus(G2)
rlocus(G2)
plt.show()
```

- Root-locus plot with damping ratio lines and customisation

```
s = tf('s'); G2 = (s + 8) / ((s + 2) * (s + 4))
rlocus(G2, xlim = [-20, 4], grid = True)
plt.title('Root Locus G2(s)', fontsize = 12); plt.show()
```

- Root-locus plot with emphasis of the poles from 17 controller gains from the logarithmically spaced interval (0.01, 100) (Section 7.3.1)

```
G2 = tf([1, 8], [1, 6, 8]); K = np.logspace(-2, 2, 18)
rlocus(G2)
poles, gains = rlocus(G2, K, plot = False)
plt.scatter(poles.real, poles.imag, marker = 'x', color = 'red')
plt.show()
```

- Read the gain, pole, and damping ratio: click on the desired point of the `rlocus` results.
- Plots of the system: step, root-locus plot, Bode diagrams, and margins.

```
G2 = tf([1, 8], [1, 6, 8])
sisotool(G2)
plt.show()
```

7.7 Bibliography

A good treatment of root-locus, lead and lag compensations can be found in Ogata (2010, Chapter 6), Golnaraghi (2010, Chapter 7), information in Slovak in Huba, Hubinský, and Žáková (2006, Chapter 7).

PID Tuning

In this chapter you will:

- learn selected experimental and model-based methods for PID tuning,
- understand the role of individual controller parameters for control performance,
- apply your knowledge to the real-time control design of a laboratory device.

A general tuning procedure chooses controller structure and parameters so that some performance criteria of the closed-loop system are satisfied.

The selection of tuning methods presented here is by no means complete. These can be thought of as a starting point for tuning. The controller parameters obtained are usually further refined.

8.1 Overview

Let us briefly review the individual parts of a PID controller and their effect on control performance. Assume the controller in the parallel form

$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right). \quad (8.1)$$

Table 8.1: Impact of the individual PID components on control performance.

Response	Rise Time	Overshoot	Settling Time	S-S Error
K_p	↓	↑	–	↓
$1/T_i$	↓	↑	↑	0
T_d	–	↓	↓	–

In general, from Table 8.1, it follows that

- proportional control improves the rise time,
- derivative control improves the overshoot,
- integral control eliminates the steady-state error,

and

- a standalone P controller is used when a steady-state offset can be accepted or if the process contains a free integrator (for example, a tank with a pump),
- the most common controller is the PI controller, because it can remove the steady-state offset,
- a PID controller can be used if there is not much measurement noise and we would like to speed-up the closed-loop response compared to that of a PI controller,
- a PD controller can increase the controller gain (and reduce steady-state offset of a standalone P controller) without compromising closed-loop stability.

There are two classes of tuning methods: those based on experimental observation of input-output behaviour of the process and those based on a process model.

8.1.1 Experimental Methods

All methods presented in this section, namely

- trial-and-error method,
- Ziegler-Nichols – steady-state oscillations,
- quarter-decay method,
- good-gain method,

return the parameters of a PID controller in parallel form (8.1).

Trial-and-Error Method

Typical PID controller parameters can be determined as follows:

- Start with a P controller with a small gain: $T_i \rightarrow \infty, T_d = 0$.
 - Increase K_p until steady-state oscillations occur (without saturation) with K_{pc} .
 - Set $K_p = 0.5K_{pc}$, decrease T_i until steady-state oscillations occur with T_{ic} . Then, set $T_i = 3T_{ic}$.
 - Increase T_d until steady-state oscillations occur with T_{dc} . Then, set $T_d = 0.33T_{dc}$.
-

Ziegler-Nichols – Steady-State Oscillations

Start with a P controller and increase its gain until steady-state oscillations occur with a gain K_{pc} and a period T_c . Based on the controller structure, the final controller parameter values are set according to Table 8.2.

Table 8.2: PID tuning formulae based on the “Ziegler-Nichols – steady-state-oscillations method”.

Controller	K_p	T_i	T_d
P	$0.5K_{pc}$		
PI	$0.4K_{pc}$	$0.8T_c$	
PID	$0.6K_{pc}$	$0.5T_c$	$0.125T_c$
PID ₁	$0.33K_{pc}$	$0.5T_c$	$T_c/3$
PID ₂	$0.2K_{pc}$	$0.5T_c$	$T_c/3$
PI (TLM)	$0.31K_{pc}$	$2.2T_c$	
PID (TLM)	$0.45K_{pc}$	$2.2T_c$	$0.16T_c$

PID₁ gives smaller overshoot, while PID₂ is underdamped. TLM stands for the Tyreus-Lyuben method.

Quarter-Decay Method

Start with a P controller and increase its gain until the step response exhibits a damping ratio of 0.25 (magnitudes of subsequent peaks, see Figure 8.1(a)) with a gain K_{p4} and a period of oscillations T_{c4} . Based on the controller structure, the final controller parameter values are set according to Table 8.3.

Table 8.3: PID tuning formulae based on the quarter-decay method.

Controller	K_p	T_i	T_d
P	K_{p4}		
PI	$0.9K_{p4}$	T_{c4}	
PID	$1.2K_{p4}$	$0.6T_{c4}$	$0.15T_{c4}$

Compared with the previous methods, the quarter-decay method is superior to previous ones that it does not drive the closed-loop system to instability.

Good-Gain Method

Start with a P controller and increase its gain until the step response exhibits an overshoot and an observable undershoot with the gain K_{pg} . Illustration of such a step response can be found in Figure 8.1(b). The time T_g is defined as the difference between times of the undershoot and overshoot.

The controller parameters are then: $K_p = 0.8K_{pg}$, $T_i = 1.5T_g$, and $T_d = 0.25T_i$. This tuning is based on requirements on the damping ratio and the overshoot: $\zeta = 0.6$, $\sigma = 10\%$.

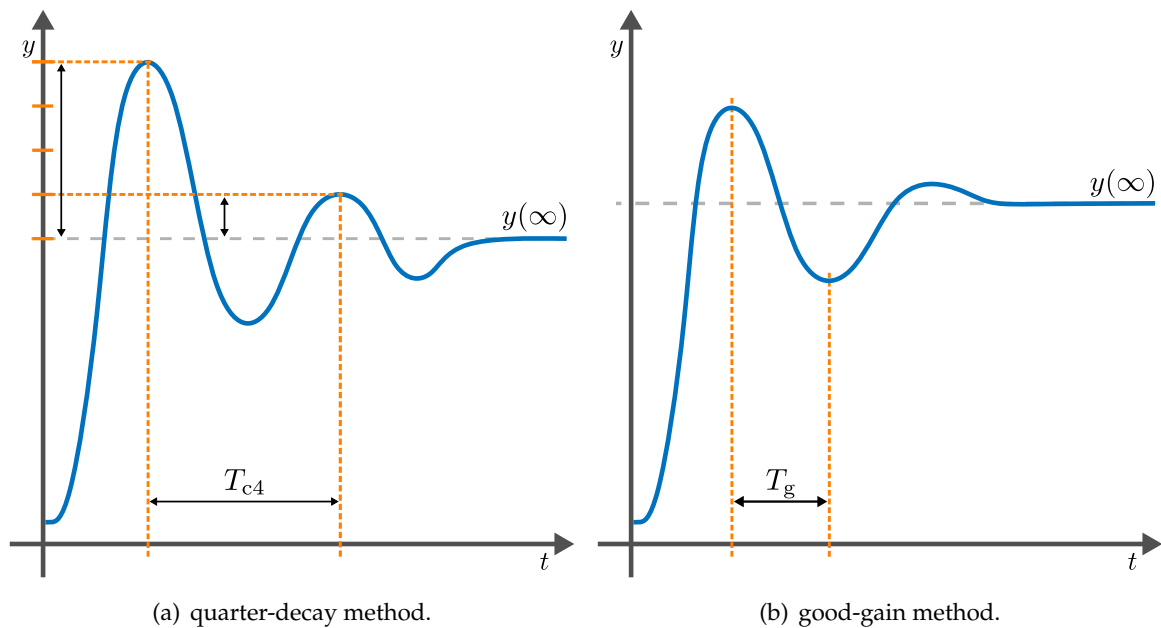


Figure 8.1: Parameters determined by the quarter-decay method and the good-gain method from the step response of closed-loop system.

8.1.2 Model-Based Methods

Ziegler-Nichols – Step Response

Step responses can be used as a non-parametric process model (Figure 8.2). It is assumed that the process is in the steady state y_0 with an input u_0 changed at time t_0 to a value u_∞ to reach the final value y_∞ . A tangent line in the inflection point of the step response is drawn to define times t_1 (intersection of the tangent with y_0) and t_2 (intersection of the tangent with y_∞).

Based on this information, the static gain and the times t_u and t_n are determined as follows:

$$K = \frac{y(\infty) - y(0)}{u(\infty) - u(0)}, \quad t_u = t_1 - t_0, \quad t_n = t_2 - t_1, \quad (8.2)$$

while the controlled parameters are given by Table 8.4. Note, that the form of the PID controller considered by this design is the parallel form (8.1). Furthermore, this method holds only for cases when $t_1 \neq t_0$, *i.e.*, for at least second-order systems.

Table 8.4: PID tuning formulae based on the “Ziegler-Nichols – step-response method”.

Controller	K_p	T_i	T_d
P	$\frac{1}{K} \frac{t_n}{t_u}$		
PI	$\frac{0.9}{K} \frac{t_n}{t_u}$	$3.33t_u$	
PID	$\frac{1.2}{K} \frac{t_n}{t_u}$	$2t_u$	$0.5t_u$

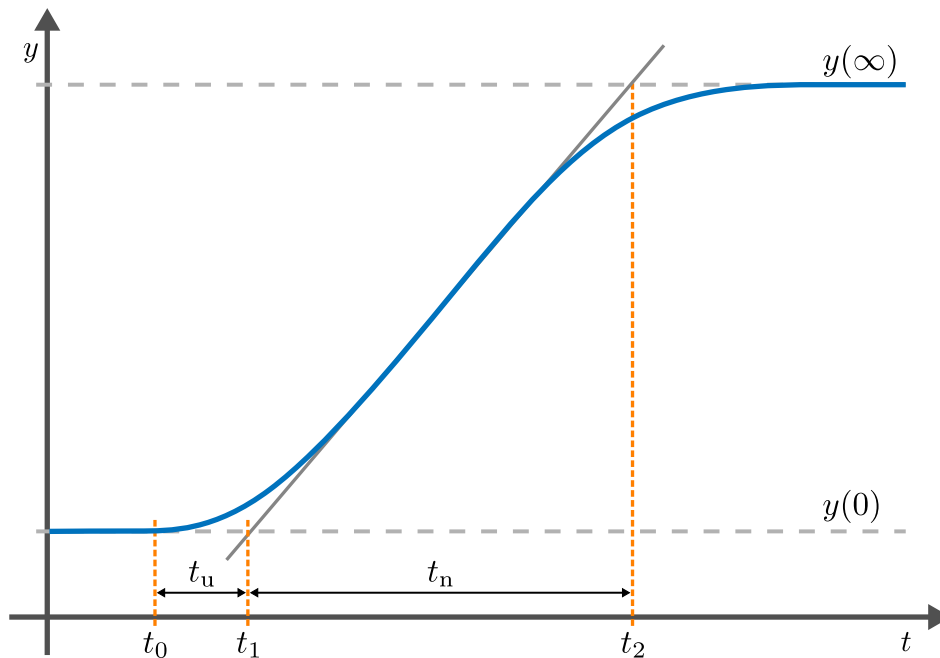


Figure 8.2: Step response of a nonoscillatory stable system with a step change occurring at time t_0 .

SIMC – Skogestad

This method considers a PID controller in the serial form

$$G_c = K_p \left(1 + \frac{1}{T_i s} \right) (1 + T_d s). \quad (8.3)$$

The controller structure and parameters depend on the process model and are given in Table 8.5.

Table 8.5: PID tuning formulae based on the SIMC method.

PID	Process	K_p	T_i	T_d
I	$K e^{-T_D s}$		$2KT_D$	
PI	$\frac{K}{Ts + 1} e^{-T_D s}$	$\frac{T}{2KT_D}$	$\min(T, 8T_D)$	
PID	$\frac{K}{(T_1 s + 1)(T_2 s + 1)} e^{-T_D s}$	$\frac{T_1}{2KT_D}$	$\min(T_1, 8T_D)$	T_2
PI	$\frac{K}{s} e^{-T_D s}$	$\frac{1}{2KT_D}$	$8T_D$	
PID	$\frac{K}{s(T_2 s + 1)} e^{-T_D s}$	$\frac{1}{2KT_D}$	$8T_D$	T_2
PID	$\frac{K}{s^2} e^{-T_D s}$	$\frac{1}{16KT_D^2}$	$8T_D$	$8T_D$

All models are limited to contain only the gain K in the numerator and a nonzero time delay T_D . Skogestad proposes to reduce other models using the so-called half rule to either first- or

second-order models. If the original model is of the form

$$G_S(s) = \frac{1}{\prod_{i=1}^n (T_{i0}s + 1)} e^{-T_{D0}s}, \quad (8.4)$$

with time constants sorted in decreasing order ($T_{i0} \geq T_{i+1,0}$), then the first-order approximation is given as

$$G_S(s) \approx \frac{1}{T_1s + 1} e^{-T_D s}, \quad T_1 = T_{10} + \frac{T_{20}}{2}, \quad T_D = T_{D0} + \frac{T_{20}}{2} + \sum_{i=3}^n T_{i0}. \quad (8.5)$$

Similarly, the second-order model can be approximated by

$$G_S(s) \approx \frac{1}{(T_1s + 1)(T_2s + 1)} e^{-T_D s}, \quad T_1 = T_{10}, \quad T_2 = T_{20} + \frac{T_{30}}{2}, \quad T_D = T_{D0} + \frac{T_{30}}{2} + \sum_{i=4}^n T_{i0}. \quad (8.6)$$

If $T_{D0} = 0$ then $T_1 = T_{10}/2$, $T_D = T_{10}/2$.

Pole Placement

If the desired closed-loop poles are known, the controller parameters can be found by equating the desired closed-loop characteristic polynomial with the one containing unknown controller parameters. This gives a set of nonlinear equations. The controller structure has to be chosen in such a way that the following holds:

- the number of closed-loop poles has to be equal to the number of controller parameters,
- proportional and derivative actions do not increase the degree of the characteristic polynomial,
- integral action increases the degree of the characteristic polynomial.

Example. 8.1: Controller structure for a first-order system

Consider a controlled process with a first-order transfer function

$$G(s) = \frac{K}{Ts + 1}, \quad (8.7)$$

with a stable pole $s = -1/T$. Find which of the PID controller structures are applicable using the pole-placement method and derive the expressions for the controller parameters.

When assuming a proportional controller $G_c(s) = K_p$, the closed-loop transfer function between the output and setpoint can be derived as

$$G_{yw}(s) = \frac{KK_p}{Ts + (1 + KK_p)} = \frac{\frac{KK_p}{T}}{s + \frac{1 + KK_p}{T}}. \quad (8.8)$$

We can see that the closed-loop characteristic equation is a first-order polynomial. Therefore, it is possible to choose one pole which should be located to the left of the original

pole. We will characterise it with a frequency ω_0

$$\omega_0 > \frac{1}{T}. \quad (8.9)$$

This gives a characteristic equation

$$s + \omega_0 = 0. \quad (8.10)$$

For a given ω_0 , we can compare coefficients at the same powers of s to get

$$\omega_0 = \frac{1 + KK_p}{T}. \quad (8.11)$$

Therefore, the controller gain is defined by

$$K_p = \frac{\omega_0 T - 1}{K}. \quad (8.12)$$

If we consider a general PID controller of the form

$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) = \frac{K_p (T_i s + 1 + T_i T_d s^2)}{T_i s}, \quad (8.13)$$

the closed-loop transfer function is of the form

$$G_{yw}(s) = \frac{KK_p(T_i s + 1 + T_i T_d s^2)}{(T + KK_p T_d) s^2 + (1 + KK_p) s + KK_p / T_i}. \quad (8.14)$$

The characteristic polynomial is a second-order polynomial, so that two poles are to be chosen. The PID controller has three parameters hence one of them can be omitted. This is possible only if $T_d = 0$ (recall that the integration part increases the number of poles so removing it would decrease the number of poles to be placed). Then, the characteristic equation can be simplified as

$$s^2 + \frac{1 + KK_p}{T} s + \frac{KK_p}{TT_i} = 0, \quad (8.15)$$

and to

$$s^2 + \frac{1 + KK_p}{T} s + \frac{KK_p}{TT_i} = s^2 + 2\zeta\omega_0 s + \omega_0^2, \quad (8.16)$$

for chosen values of ζ and ω_0 . Equating the coefficients at respective powers of s gives

$$2\zeta\omega_0 = \frac{1 + KK_p}{T}, \quad \omega_0^2 = \frac{KK_p}{TT_i}, \quad (8.17)$$

and the values of the PI controller parameters are given by

$$K_p = \frac{2\zeta\omega_0 T - 1}{K}, \quad T_i = \frac{2\zeta\omega_0 T - 1}{T\omega_0^2}. \quad (8.18)$$

Therefore, we can conclude that the pole-placement method is applicable to controller design of a first-order system with PID controller structures: P and PI.

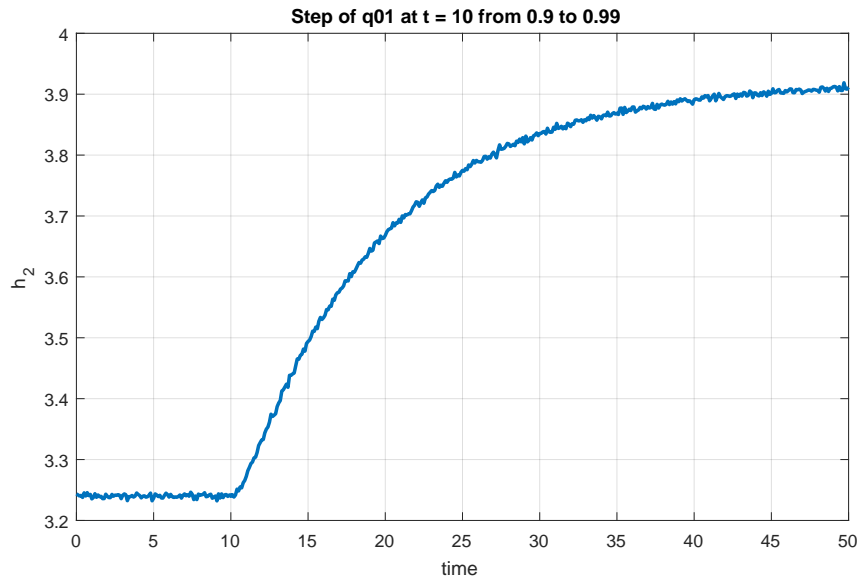


Figure 8.3: Step response of two tanks with interaction.

8.2 Examples

Example 8.2: PID control of two tanks with interaction

We study the PID control of two tanks with interaction modelled in Example 1.2.2. It is desired to tune the controller parameters using two methods and to compare their performance:

1. the Ziegler-Nichols – step response,
2. The SIMC method.

Implement the controller anti-windup feature to handle the flow constraints $0.05 \text{ m}^3/\text{s} \leq q_{0,1}(t) \leq 2 \text{ m}^3/\text{s}$.

1. We apply a 10% step change in the flow $q_{0,1}$ at $t = 10 \text{ s}$ from the steady-state $q_{0,1}^s = 0.9 \text{ m}^3/\text{s}$ to $q_{0,1} = 0.99 \text{ m}^3/\text{s}$. The resulting step response is shown in Figure 8.3.

We can estimate the inflection point and tangent line to the step response at this point. This determines the time points $t_1 = 10.5$ and $t_2 = 22$. The initial and final values of the process output are $y_0 = 3.24$, $y_\infty = 3.91$. Based on these values, the process gain K and times t_u and t_n are given as (note that in this case it is quite difficult to estimate the inflection point as the step response is very similar to a first-order system without an inflection point)

$$K = \frac{y(\infty) - y(0)}{u(\infty) - u(0)} = 7.44, \quad t_u = t_1 - t_0 = 0.5, \quad t_n = t_2 - t_1 = 11.5. \quad (8.19)$$

If we choose the PI control structure, the controller parameter values can be read

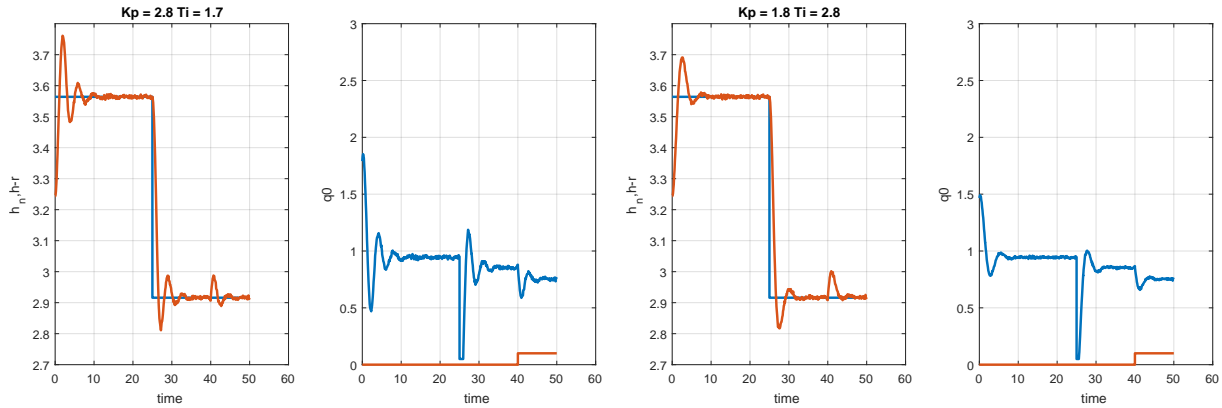


Figure 8.4: PI control of the two tanks with interaction process: (left) Ziegler-Nichols, (right) SIMC.

from the Ziegler-Nichols table (Table 8.4):

$$K_p = \frac{0.9 t_n}{K t_u} = 2.8, \quad T_i = 3.33 t_u = 1.7. \quad (8.20)$$

The process is controlled using the anti-windup AW2 configuration shown in Figure 6.11. Time trajectories of the controlled output and both inputs are shown in Figure 8.4 (left). We can observe that the Ziegler-Nichols rules provide controller yielding a rather oscillatory performance, but the controller can handle well both tracking (time points 0 s and 25 s) and regulation ($t = 40$ s) problems.

2. To apply the SIMC method, we need the process model. We can either estimate the model from the step response (as a first-order process with delay) or we can use the knowledge about its transfer function

$$G(s) = \frac{7.2}{(8.6s + 1)(0.7s + 1)}. \quad (8.21)$$

The model is not among the supported structures, hence we apply the half rule and reduce the model to

$$G(s) = \frac{7.2}{9s + 1} e^{-0.35s}. \quad (8.22)$$

The corresponding structure is a PI controller with parameters

$$K_p = \frac{T}{2KT_D} = 1.8, \quad T_i = \min(T, 8T_D) = 2.8, \quad (8.23)$$

and its performance is shown in Figure 8.4 (right). We can see that the response is not so oscillatory but still sufficiently fast.

8.3 Practice Examples

8.3.1 PID Controller Design for Two Tanks in Series

Consider the process of two tanks in series, whose nonlinear model was derived in Section 1.2.4 and its linear model is given by two transfer functions (Section 2.3.2). To test designed controllers with the nonlinear simulation model, consider a 20% step change on the manipulated variable.

8.3.1.1 Discuss which model-based methods can be applied to design a PID controller for both transfer functions.

Dynamic relationships between the input flow $q_{0,1}(t)$ and the levels $h_1(t)$ and $h_2(t)$, respectively, are described by the following transfer functions

$$G_1(s) = \frac{1.25}{s + 1.32}, \quad (8.24a)$$

$$G_2(s) = \frac{5.51}{s^2 + 5.73s + 5.82}. \quad (8.24b)$$

We can consider the pole placement method (PPM) and the SIMC method. As far as SIMC method is concerned, we have to reduce the transfer function (8.24b) to a first order system with a time delay.

8.3.1.2 Design proportional and PI controllers for the first-order transfer function and compare the control performance using the nonlinear model.

The value of the controller gain for a first-order system is given by (8.12), or we can derive the relationship between the desired pole and the value P of the proportional controller as

$$P = \frac{pT - 1}{K}, \quad (8.25)$$

where p stands for desired absolute value of the stable closed-loop pole. For the choice $p = 3$ (two to three times faster than the open-loop pole $p = 1.32$) the controller gain is evaluated as $P = 1.34$.

```
p = 3; % choice of a stable pole
[n, d] = tfdata(G1, 'v');
T = 1/d(end);
K = n(end)/d(end);
P = (p*T-1)/K; Kp = K;
```

The PI controller is designed according to the formula in (8.18). To solve the set of equations for the coefficients of the PI controller, we may use the Symbolic toolbox as follows

```
syms P I
p1 = 1/T; p2 = 3;
sol = solve( (1+K*P)/T == p1 + p2, K*I/T == p1*p2, P, I );
P = double(sol.P); I = double(sol.I);
Kp = P; Ki = P/I;
```

One of the desired poles is set the same as the process pole and the other one as before ($p = 3$). The coefficients for this PI controller are $P = 2.4$, and $I = 3.16$. The

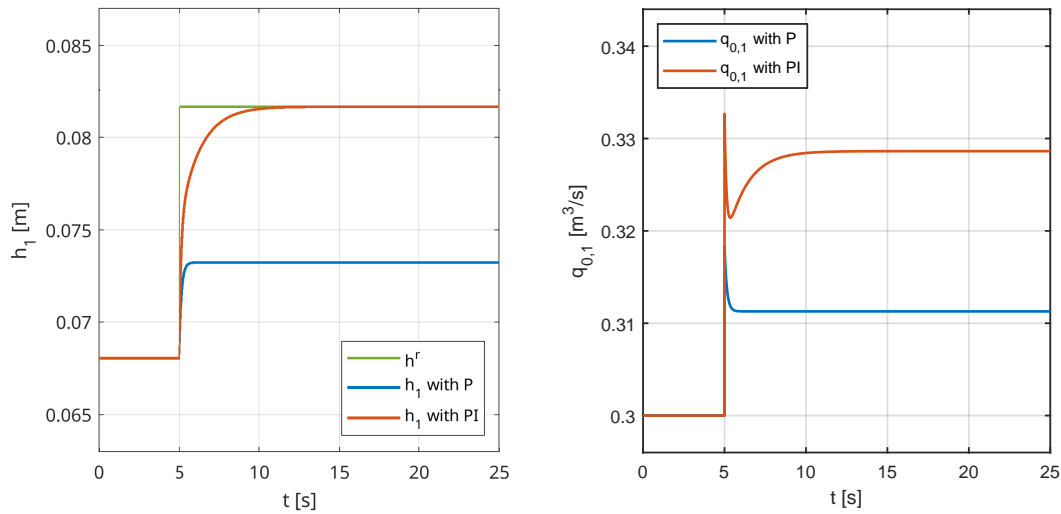


Figure 8.5: Comparison of P and PI control designed by pole-placement method based on $G_1(s)$ from (8.24a).

comparison of the proportional and PI controllers designed by the pole-placement method is shown in Figure 8.5. We can see that PI controller is considerable slower due to chosen poles.

8.3.1.3 Consider the Ziegler-Nichols – step response method for the design of PI and PID controllers. Perform simulations solely based on the nonlinear model. To generate the step response consider a step change on the input flow from $q_{0,1}^s \rightarrow 1.3 \cdot q_{0,1}^s$.

To adopt the formulas from the Ziegler-Nichols – step response method, we extract the static gain K , and the values t_u , and t_n . We obtain them from an open loop simulation with the nonlinear model. The step response is depicted in Figure 8.6. Values of the reported variables are $t_1 = 0.11$ s, $t_2 = 1.40$ s, hence

$$K = 0.99, \quad t_u = 0.11 \text{ s}, \quad t_n = 1.26 \text{ s}. \quad (8.26)$$

Coefficients of the PI and PID controllers are given by expressions in Table 8.4; for this case they have been evaluated as

$$\text{PI: } K_p = 11.77, \quad T_i = 0.36, \quad (8.27a)$$

$$\text{PID: } K_p = 2.62, \quad T_i = 0.22, \quad T_d = 0.63. \quad (8.27b)$$

The control performance with the nonlinear model is shown in Figure 8.7. As expected, the response is rather oscillatory.

8.3.1.4 Design a PID controller with the quarter-decay method using the nonlinear model to get K_{p4} and T_{c4} .

We perform a set of simulations involving a proportional controller and the nonlinear model of the two-tanks system to obtain oscillatory behaviour with

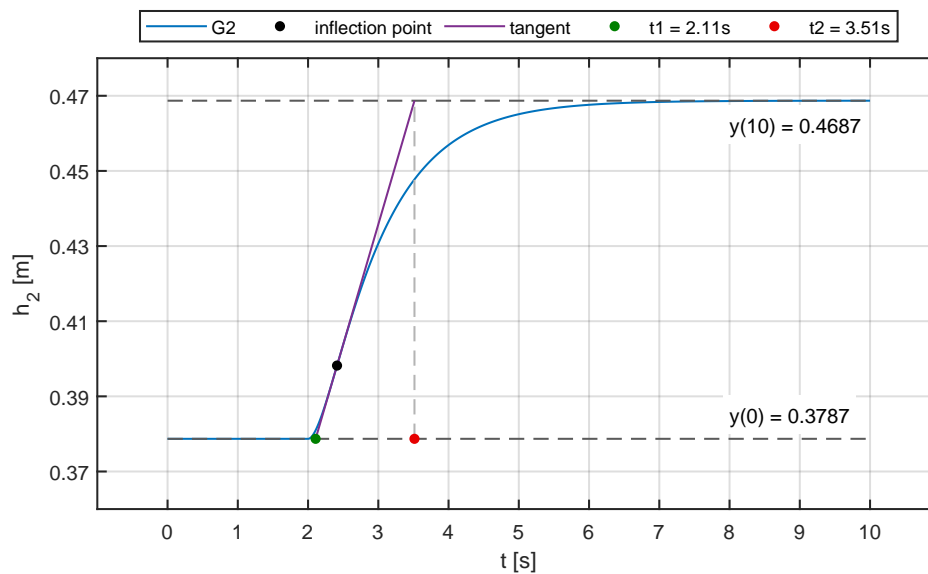


Figure 8.6: Extraction of the parameters required by the Ziegler-Nichols – step response method.

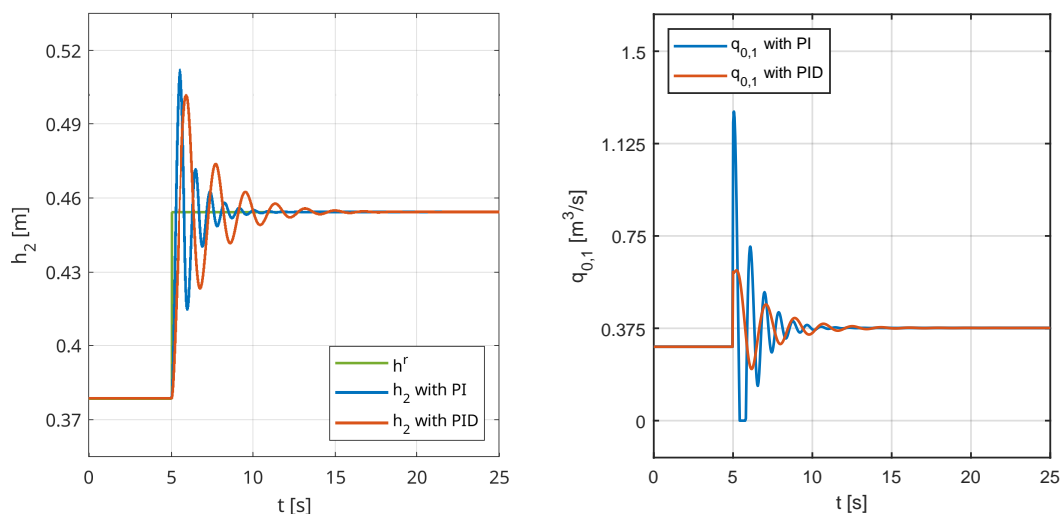


Figure 8.7: Comparison of P and PI controllers designed by the Ziegler-Nichols – step response method.

at least two peaks. The coordinates of the peaks from the step response can be extracted using the function `findpeaks()`. Verify that the Simulink model contains a saturation block after calculating the control action with the P controller. The following script depicts a way to extract information about the peaks from the simulated profiles systematically:

```
P_pool = [15, 35, 50];
for k = 1:length(P_pool)
    P = P_pool(k);
    simout = sim(sim_model, sim_time);
    y = simout.h(:, 3); % variable "h" is set in the Scope block
    [peak_val, peak_idx] = findpeaks(y);
```

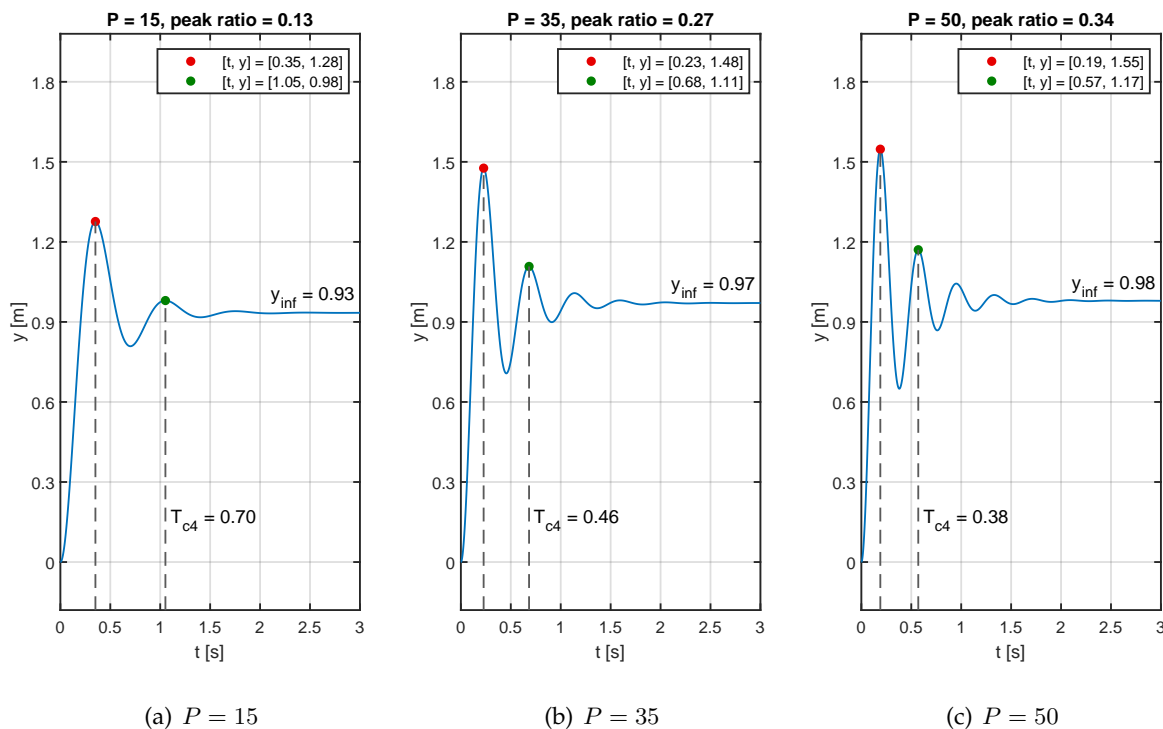


Figure 8.8: Closed-loop step responses with $G_2(s)$ from (8.24b), under different proportional controllers.

end

Hint: set the PID block in the Simulink model to pure P controller and set the value of the controller to the variable “P”. Depending on the version of the MATLAB/Simulink, it may be required to change the solver options in Simulink. Suggested value for maximum step time is at 10^{-3} , while relative and absolute tolerance may be set to 10^{-4} .

We deal with a nonlinear model where a saturation of the control signal plays a vital role since the inlet flow must be nonnegative. We cannot excite the nonlinear model sufficiently enough to achieve the quarter decay of the first two subsequent peaks for any choice of the gain value P of the controller. As an evidence, we offer an analysis, where we increase the value of the gain and measure the exact peak ratio. Such analysis is shown in Figure 8.9, which shows that even if we increase the value of the gain, the peak ratio will never reach the required value of 25%. We also compare the results if we consider the linearised model given by $G_2(s)$ from (8.24b), without saturations.

We also offer a solution based on the linearised model. However, we note that experimental methods should be used primarily directly with nonlinear models and laboratory devices. We remind that the peak ratio is computed with respect to the steady-state value of the step response. For this case, we consider a set of proportional controllers with $P = [15, 35, 50]$ for which we evaluate time coordinates of the first two peaks, the absolute values of peaks, the steady-state values of the step response, and the ratios. Individual values are shown in Figure 8.8.

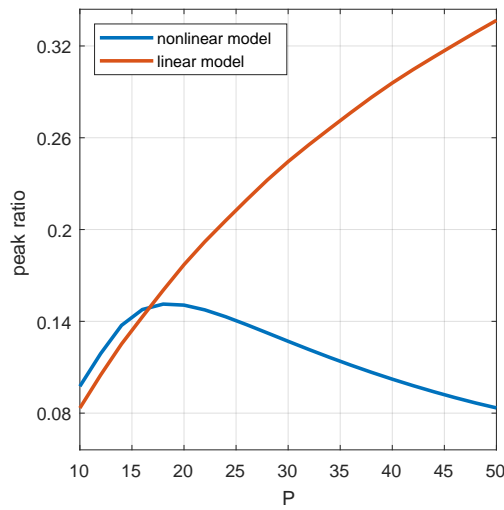


Figure 8.9: Peak ratio for the nonlinear model calculated based on system excitation with proportional controller.

Coefficients of PI and PID controllers are given by expressions in Table 8.3, and for our case are evaluated as follows

$$\text{PI: } K_p = 31.46, \quad T_i = 0.46, \quad (8.28a)$$

$$\text{PID: } K_p = 42, \quad T_i = 0.28, \quad T_d = 0.07. \quad (8.28b)$$

Control performance under both controllers is compared with the nonlinear model. Responses of the controllers to changes in setpoints are presented in Figure 8.10.

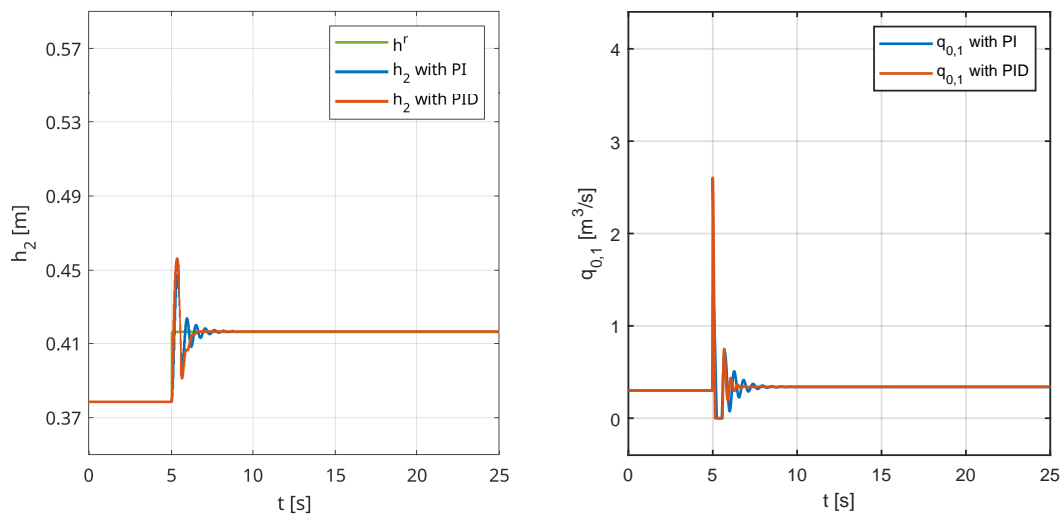


Figure 8.10: Comparison of P and PI controllers designed using the quarter-decay method for $G_2(s)$ from (8.24b).

8.3.2 PID Controller Design for Two Tanks with a Pump

Consider the model of two tanks with a pump from the Section 2.2.3. Recall that two transfer functions have been derived for this system.

8.3.2.1 Discuss which model-based methods can be used to design a PID controller for both transfer functions.

Transfer functions explored in this question map changes in the pump suction force and changes in liquid levels in individual tanks. These transfer functions are

$$G_1(s) = \frac{0.56}{0.4s^2 + 0.72s}, \quad (8.29a)$$

$$G_2(s) = \frac{-0.5s - 0.56}{0.4s^2 + 0.72s}. \quad (8.29b)$$

Figure 1.11 shows a schematic diagram of such a system. From the analysis of respective step responses (Figure 8.11) and the transfer function poles in (8.29), which are $\{0, -1.81\}$, we conclude that

- Ziegler-Nichols – step response method is *not applicable*, since we cannot extract values like t_u ,
- SIMC method is *not applicable*, since the transfer functions do not include time delay T_D . However, if a small time delay will be considered, PI or PID controllers can be designed according to formulae in Table 8.5,
- Pole-placement method is the only feasible approach for the model-based controller design, since it does not impose any specific restrictions.

Note, that the step response of the transfer function $G_2(s)$ (8.29b) has inverse behaviour, which is caused by the negative gain of this part of the process.

8.3.2.2 Determine if we can design a PID controller for both transfer functions using the quarter-decay and good-gain methods. Discuss the results.

To apply experimental methods like the quarter-decay method and good-gain methods, we must first perform a series of closed-loop simulations with a proportional controller. The controller should be tuned in such a way that an oscillatory response is obtained.

By increasing the gain of the controller, oscillatory response is obtained for the transfer function $G_1(s)$. Therefore, the good-gain method and quarter-decay methods can be used to design a controller for this system. An illustration for the good-gain method is shown in Figure 8.12(a). Here, the resulting coefficients of the PID controller are given as

$$\text{PID} : K_p = 2.4, \quad T_i = 2.57, \quad T_d = 0.64. \quad (8.30)$$

On the other hand, none of the mentioned methods are applicable for controller design for $G_2(s)$. From the step-response analysis in Figure 8.12(b), we conclude

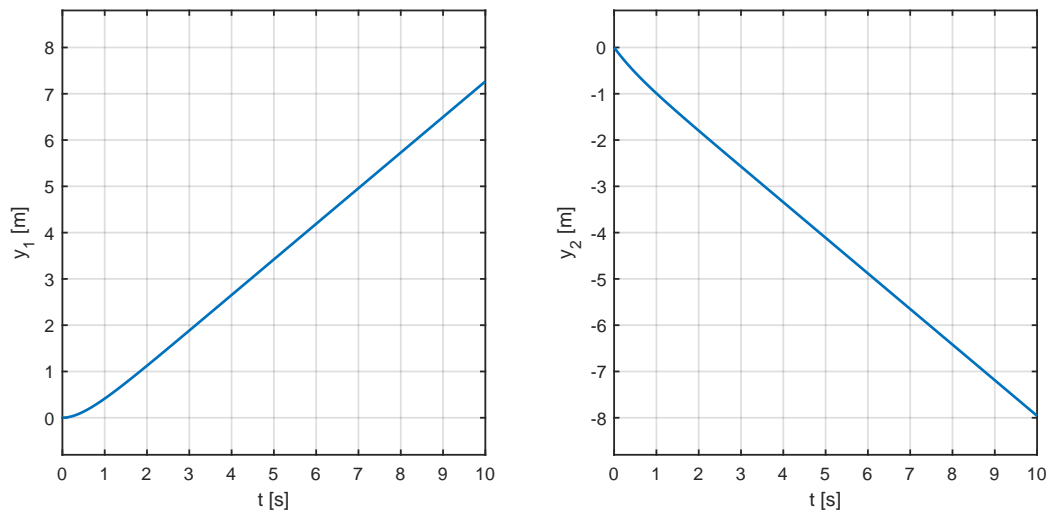
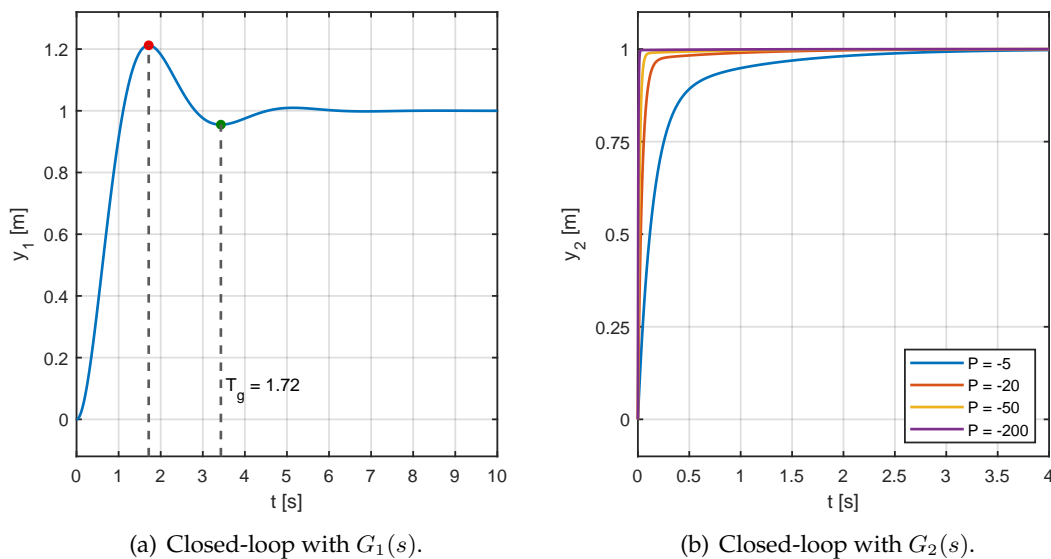


Figure 8.11: Unit step response corresponding to the dynamic behaviour given in (8.29).

that the oscillatory closed-loop response is not reached even with very high gains.



(a) Closed-loop with $G_1(s)$.

(b) Closed-loop with $G_2(s)$.

Figure 8.12: Closed-loop step responses with $G_1(s)$, and $G_2(s)$ from (8.29), under different proportional controllers.

8.3.2.3 Use the function `pidTuner()` in MATLAB to design a P, PI, and PID controllers. Compare the control performance using the nonlinear model.

The function `pidTuner` is a built-in MATLAB toolbox enabling interactive design of different PID controller types. We direct the reader to Figure 8.13, which provides an example of the interactive controller tuning interface. The behaviour of the closed-loop is primarily adjusted by two sliders in the top-middle part, and

by the type of the controller in the top-left part of the pidTuner toolbox. The actual parameters of the PID controller are displayed in the bottom-right corner. For convenient variable handling, the tuned controllers can be exported to a specific variable with the “Export” button in the top-right corner.

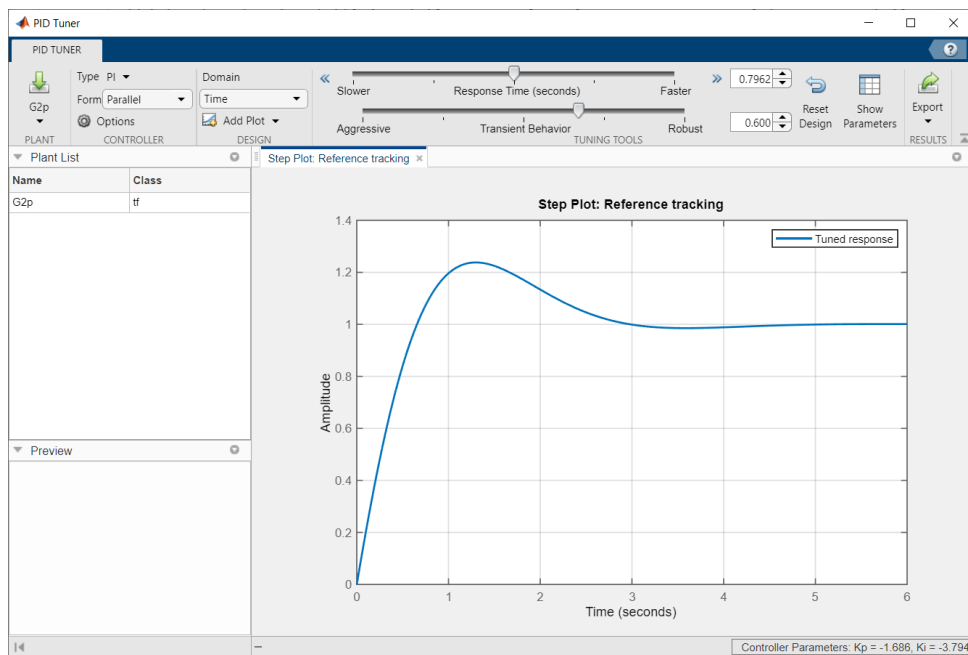


Figure 8.13: Tuning of the PID controller via `pidTuner()`.

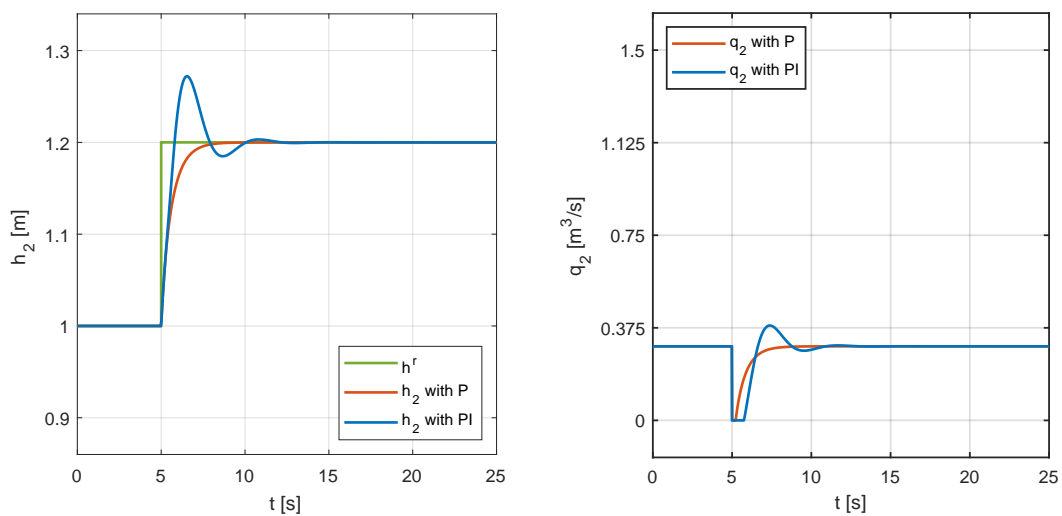


Figure 8.14: Control performance under P and PI controllers obtained using the pidTuner Toolbox.

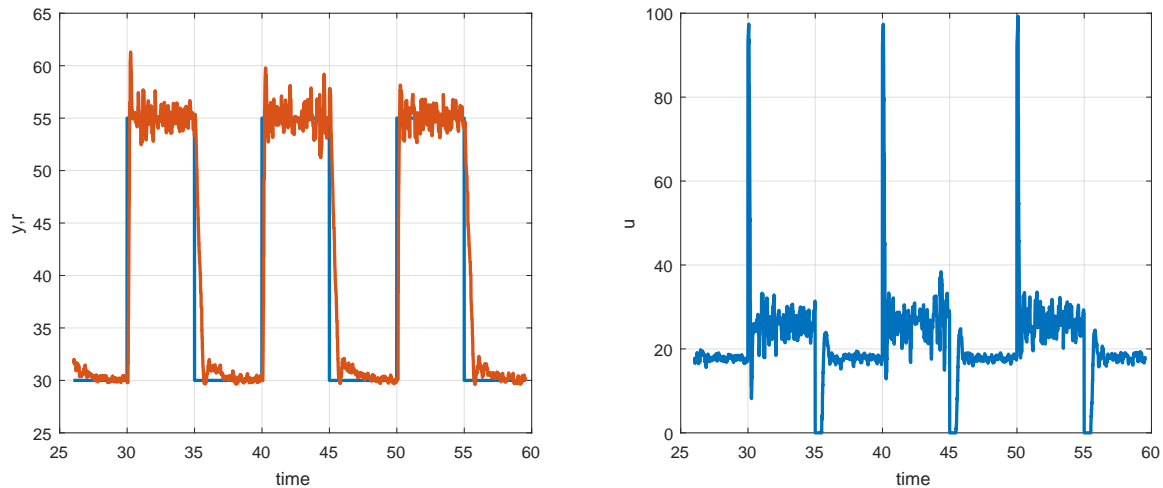


Figure 8.15: Tracking performance of the Flexy plant.

8.3.3 Fan-Flex System – Flexy2

We perform an experimental study on a PID controller design and tuning for the Flexy plant (version 2). Flexy is an open-source hardware device for process control training and education. It represents a simple dynamic system with one actuator (fan) and one sensor (flex resistor). The system allows us to measure and control the air flow produced by the fan. The air flows around an obstacle that is mounted on a flexible strip. The strip bends as a result of the increased hydrodynamic pressure. The corresponding change in the electric resistance of the flex sensor is measured. Flexy is equipped with a microcontroller unit that allows us to continuously control the fan speed and measure the bend of the flex resistor. Dimensions of the device allow us to use the standard short-format Arduino-type boards (e.g. Uno, Leonardo, 101, Zero, Yun). Flexy is also equipped with one potentiometer knob as an additional input. Users can define the functionality of the knob in a program of the microcontroller (e.g. control setpoint, fan control, manual parameter change).

A good closed-loop performance under the designed controller is represented by the tracking performance shown in Figure 8.15.

8.3.3.1 Identify the system using the Strejc's method.

We performed an experiment, where we first set the input to 30% for 10 seconds and then increased the input to 60%. The resulting output is shown in Figure 8.16.

We can find the necessary parameters using the following code:

```
% get the value of y(0) and y(inf)
u0 = 30; uinf = 60;
y0 = 26.98; yinf = 57.98;

% interesting time points
t0 = 10;

% get inflection point line and its parameters
TD = 10.1 - t0 = 0.1;
Tuf = 10.1 - TD - t0 = 0;
Tus = TD + Tuf = 0.1;
Tn = 10.4 - Tus - t0 = 0.3;
```

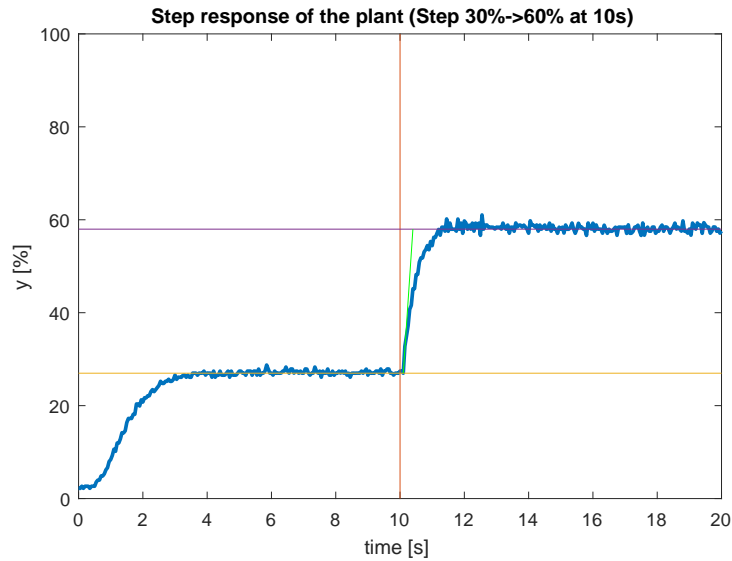


Figure 8.16: Step response of the Flexy plant.

```
% we choose the first-order system
T = Tn * gn = 0.3;
Z = (yinf - y0) / (uinf - u0) = 1.0333;

% the identified transfer function
G = tf(Z, [T 1], 'InputDelay', TD)
```

8.3.3.2 Design P, PI, and PID controllers using the model-based methods (Ziegler-Nichols and SIMC). Test the designed controllers. *Hint: If necessary, implement an anti-windup method.*

MATLAB code to tune controllers using the Ziegler-Nichols and the SIMC methods is given below and the experimental results are shown in Figure 8.18 (the simulation model is shown in Figure 8.17).

```
% parameters of the model
Z = 1 + 0.1 / 3; T = 0.3; TD = 0.1;

% the identified transfer function
G = tf(Z, [T 1], 'InputDelay', TD);

% Ziegler-Nichols - step response
tu = 0; % this is problematic
tu = TD; % practical choice
tn = T;

%% P Controller
% K = tn/tu/Z

%% PI Controller
K = 0.9 / Z * tn / tu; TI = tu * 10 / 3; P = K; I = 1 / TI;

%% PID Controller
% K = 1.2 / Z * tn / tu; TI = 2 * tu; TD = tu / 2

%% Get real-time measurements
```

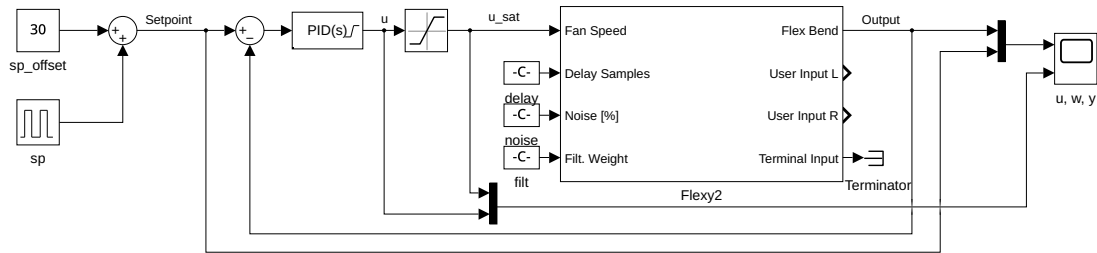


Figure 8.17: Simulation model for PID control of the Flexy plant.

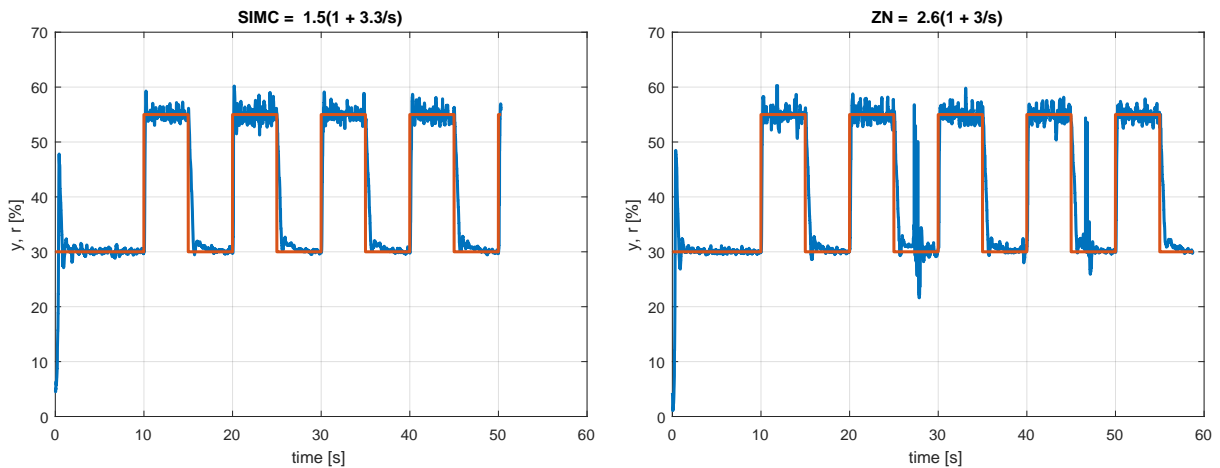


Figure 8.18: Experimental results: SIMC and ZN methods.

```

%% sim('flexy2_PID_control')
%% save plt_pi_zn.mat ScopeData

figure
load plt_pi_zn.mat ScopeData
t = ScopeData{1}.Values.Time;
yr = ScopeData{1}.Values.Data;
plot(t,yr, 'LineWidth', 2); title(['ZN = ' num2str(P,2) '(1 + '
    num2str(I,2) '/s)']); xlabel('time [s]'); ylabel('y, r [%]'); grid
on
iam_save_pdf('flexy2_PID_zn.pdf')

% SIMC
% PI Controller %G_S = \frac{k}{Ts+1} e^{-T_d s}
K = T / 2 / Z / TD; TI = min(T, 8 * TD); P = K; I = 1 / TI;

%% sim('flexy2_PID_control')
%% save plt_pi_simc.mat ScopeData

figure
load plt_pi_simc.mat ScopeData
t = ScopeData{1}.Values.Time;
yr = ScopeData{1}.Values.Data;
plot(t,yr, 'LineWidth', 2); title(['SIMC = ' num2str(P, 2) '(1 + '
    num2str(I, 2) '/s)']); xlabel('time [s]'); ylabel('y, r [%]'); grid
on
iam_save_pdf('flexy2_PID_simc.pdf')

```

8.3.3.3 Tune P, PI, and PID controllers using the experimental methods (the Ziegler-Nichols, the

quarter-decay method, and the good-gain method. *Hint: You can use the knob on the Flexy plant to provide the controller gain. You may use the range of values for the knob from 0 to 10.*

8.3.3.4 (Optional) Design a PI controller using the trial-and-error method.

8.4 Problems

Problem. 8.1: Controller structure for a second-order system with pole placement design

Consider a process with a second-order transfer function

$$G(s) = \frac{b_1s + b_0}{s^2 + a_1s + a_0}. \quad (8.31)$$

Using the pole placement design, find which PID controller structures are applicable and optionally derive the expressions for the controller parameters.

Suitable controller structures are PD and PID.

Problem. 8.2: Process with time delay

Consider a process with a first-order transfer function

$$G(s) = \frac{K}{Ts + 1} e^{-T_D s}. \quad (8.32)$$

Design a PI controller for the chosen values of K and T without assuming a time delay T_D . Then, increase the value of T_D as a multiple of T : $T_D = (0.1, 0.4, 0.7, 1, 2, 5)T$ and observe the closed-loop performance.

In general, processes with time delay can be separated into lag dominant ($T > T_D$) and time-delay dominant ($T < T_D$). Moreover, for excessive delays ($T \ll T_D$), standard feedback configuration cannot be used. Instead, time-delay compensation methods, such as the Smith predictor, are used.

Problem. 8.3: A PD controller

Consider a process of the form

$$G(s) = \frac{200}{s^3 + 11s^2 + 10s}. \quad (8.33)$$

Design a PD controller with a maximum overshoot of 20% and the settling time under 1 s. What happens to the performance if there is a constraint $|u| \leq 1$?

The controller is $G_c(s) = 0.53(1 + 1.18s)$. If there are constraints on u , the closed-loop system will be overdamped without overshoot and the settling time doubles.

8.5 MATLAB Snippets

- PID tuner in MATLAB. It can basically tune any PID controller structure, without proportional/derivative kick, and a controller with two degrees of freedom as well. It can evaluate various control performance criteria: rise and settling times, overshoot, gain and phase margins, and closed-loop stability.

```
pidTuner(sys,type)
pidTuner(sys,Cbase)
pidTuner(sys)
pidTuner
```

It is also available from the PID block in Simulink (click *Tune* in the *Main* tab).

- Flexy initialisation script

```
%% FLEXY2
% 2021-09-29

%% Prerequisites
addpath('RealTime_Pacer') % if not included in the PATH
slCharacterEncoding('UTF-8') % helps avoid warnings

%% Initialization and communication PORT identification
% f = Flexy2 %% windows
f = Flexy2('/dev/ttyACM0') %% Linux

%% Double calibration
f.calibrate;

%% Close CLI connection to enable Simulink communication
f.close

%% Initialize communication
% Sampling time
Ts = 0.01;
% Calibration
calibrate = 1;
% Port = 'COM6';
Port = f.ComPort;

%% Simulink model
%open('flexy2_PI_control_demo.slx');
```

8.6 Python Snippets

- Flexy script

```
# this is required module
import Flexy
# this is needed only for sleep function
from time import sleep

# create instance of Flexy2
f = Flexy.Flexy2('/dev/ttyACM0')

# if needed, calibrate Flexy2
print('Calibrating Flexy ...')
f.calibrate()
sleep(10)
```

```
# set the rate of data collecting
f.setInternalSamplingFrequency(20)

# lets turn on the fan on 50%
f.setFanSpeedPerc(50)

# collect and display some data
for i in range(0,50):
    print(str(i/10)+"s bend perc: "+str(f.getBendPerc()))
    sleep(.1)

f.setFanSpeedPerc(100)

for i in range(51,100):
    print(str(i/10)+"s bend perc: "+str(f.getBendPerc()))
    sleep(.1)

# turn off the fan
f.off()

# disconnect from the device
f.close()
```

8.7 Bibliography

The theoretical background for this chapter can be found in Mikleš and Fikar (2007, Chapter 7), information in Slovak in Bakošová and Fikar (2008, Chapter 6).

For complementary information on control and identification, see for example Marlin (1995, Chapter 9), Åström and Hägglund (1995), Shinskey (1979), Coughanouwr and Koppel (1965), Douglas (1972), Seborg, Edgar, Mellichamp, and Doyle (2016), Fikar and Mikleš (1999), and Shardt (2022).

Bibliography

- Åström, K. J. and T. J. Häggglund (1995). *PID Controllers: Theory, Design, and Tuning*. 2nd ed. Research Triangle Park: Instrument Society of America.
- Bakošová, M. and M. Fikar (2008). *Riadenie procesov*. Vydavateľstvo STU.
- Bequette, D. W. (2003). *Process Control. Modelling, Design, and Simulation*. Prentice Hall.
- Corriou, J.-P. (2004). *Process Control. Theory and Applications*. Springer Verlag.
- Coughanouwr, D. R. and L. B. Koppel (1965). *Process System Analysis and Control*. New York: McGraw-Hill.
- Dorf, R. C. and R. H. Bishop (2008). *Modern Control Systems*. 11th ed. Pearson Education International.
- Dostál, P., J. Mikleš, and A. Mészáros (1983). *Teória automatického riadenia. Príklady z teórie automatického riadenia II*. Bratislava: ES SVŠT.
- Douglas, J. M. (1972). *Process Dynamics and Control*. New Jersey: Prentice Hall, Inc.
- Fikar, M. and J. Mikleš (1999). *Identifikácia systémov*. STU Press.
- Golnaraghi F. and. Kuo, B. C. (2010). *Automatic Control Systems*. 9th ed. Wiley.
- Huba, M., P. Hubinský, and K. Žáková (2006). *Teória systémov*. STU Bratislava.
- Kailaith, T. (1980). *Linear Systems*. Englewood Cliffs, New Jersey: Prentice Hall.
- Koppel, L. B. (1968). *Introduction to Control Theory with Application to Process Control*. Englewood Cliffs, New Jersey: Prentice Hall.
- Luyben, W. L. (1990). *Process Modelling, Simulation and Control for Chemical Engineers*. 2nd ed. Singapore: McGraw Hill.
- Marlin, T. E. (1995). *Process Control*. Singapore: McGraw-Hill.
- Mikleš, J., P. Dostál, and A. Mészáros (1994). *Riadenie technologických procesov*. Bratislava: STU.
- Mikleš, J. and M. Fikar (2007). *Process Modelling, Identification, and Control*. Berlin: Springer Verlag.
- Mikleš, J. and V. Hutla (1986). *Teória automatického riadenia*. Bratislava: ALFA, SNTL.
- Ogata, K. (2010). *Modern Control Engineering*. 5th ed. Prentice Hall.
- Seborg, D. E., T. F. Edgar, D. A. Mellichamp, and F. J. Doyle (2016). *Process Dynamics and Control*. 4th ed. New York: Wiley.
- Shardt, Y. A. W. (2022). *Statistics for Chemical and Process Engineers: A Modern Approach*. 2nd ed. Cham: Springer Verlag.
- Shinskey, F. G. (1979). *Process Control Systems*. New York: McGraw-Hill.
- Skogestad, S. and I. Postlethwaite (1996). *Multivariable Feedback Control – Analysis and Design*. Wiley.
- Smith, C. A (2002). *Automated Continuous Process Control*. New York: Wiley.
- Smith, C. A. and A. B. Corripio (1997). *Principles and Practice of Automatic Process Control*. New York: Wiley.
- Stephanopoulos, G. (1984). *Chemical Process Control, An Introduction to Theory and Practice*. Englewood Cliffs, New Jersey: Prentice Hall.

Unbehauen, H. (1986). *Regelungstechnik I*. Braunschweig/Wiesbaden: Vieweg.

Ziegler, J. G. and N. B. Nichols (1942). "Optimum settings for automatic controllers". In: *Trans. ACME* 64.8, pp. 759–768.
