Lecture Notes

Introduction to Control Systems

Herbert Werner

Contents

Introduction

Automatic control plays an important role in engineering and science. One can define a control system as an arrangement of physical components related in such a manner as to direct or regulate itself or another system. A special class of control systems contains a feedback structure, and these are the control systems that we will study in this course. Feedback means that the quantity of interest, the controlled variable, is compared with a reference value, and that appropriate control action is taken depending on this comparison. An example that illustrates this concept of a feedback control system is a household furnace controlled by a thermostat, shown in Figure 1. This diagram identifies the key components of the control system and their interaction. A more general representation is shown in Figure 2; the structure shown in this block diagram is common to many control systems. The central component is the *plant*, in the above example the house; the controlled variable is the room temperature and the actuator is the gas valve. The concept of a plant is very general indeed: it can be - for example - an airplane and the controlled variable the airplane heading, the actuators being in that case the rudder and the ailerons. It includes also non-technical systems - many examples of feedback loops can be found in biological systems, e.g. the human skin and its perspiration system, where the controlled variable is the skin temperature and the actuators are the sweat glands. In all cases the controlled variable is measured by some sensor, and used together with a reference signal to generate the input to a controller that causes control action. The sensors and the the actuators can be devices separate from the plant (like the gas valve in the room temperature control loop), or they can be considered part of the plant.
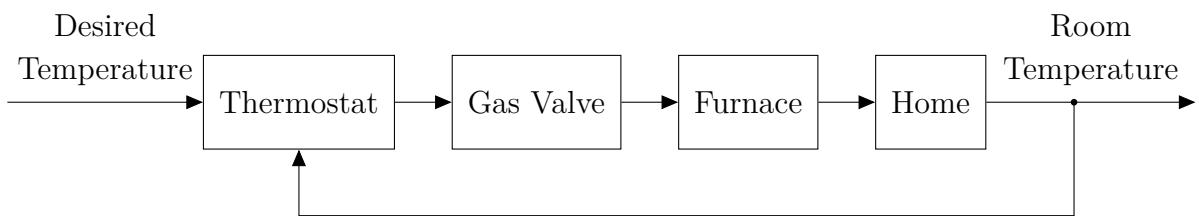

Figure 1: Room heating as an example for feedback control
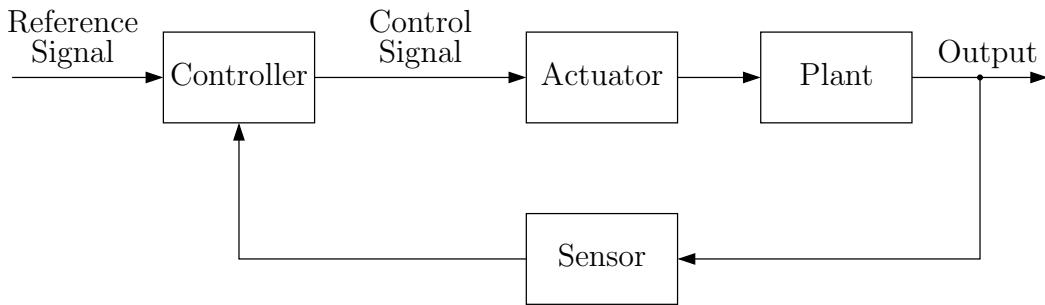
Figure 2: Structure of a closed-loop system

Because of its loop structure, such control systems are called *closed-loop systems*, in contrast to *open-loop systems* where no feedback is involved. The main advantage of closed-loop systems over open-loop ones is a considerable increase in accuracy: because the controlled variable is monitored permanently and control action taken when necessary, the system depends less on calibration, and is less sensitive to variations in the plant and its environment than an open-loop system. On the other hand, a serious problem with closed-loop systems is their tendency towards oscillation or even instability; for this reason closed-loop control systems need to be designed and tuned carefully.

This course will introduce basic methods for analysis and design of control systems. These methods can be applied to a huge variety of systems. For a systematic design of a control system usually two things are required: a mathematical model of the plant, and a control strategy that achieves a desired closed-loop behaviour. In this course, we will use linear, time-invariant (LTI) models to describe the plant. In real life, plants are in general nonlinear and time-varying, but in most cases it is possible to approximate their dynamic behaviour by an LTI model. The advantage of using LTI models is that one can employ efficient tools and the rich and well developed theory of linear control systems.

The course begins with an introduction to the basic tools for modelling the dynamic behaviour of linear systems in Chapter 1. The use of transfer function models for the analysis of linear systems is reviewed, with emphasis on the transient response of first and second order systems. In Chapter 2 we then turn to the analysis and the design of linear feedback systems. The study of control systems begins with an introduction to feedback, motivated by sensitivity reduction and disturbance rejection. Proportional, integral and derivative feedback are discussed. This overview is followed by a presentation of two classical techniques for analysis and design of control systems: methods based on root locus plots in Chapter 3, and frequency response techniques in Chapter 4. Here the emphasis is on concepts - for example students will be expected to learn how to sketch a root locus plot by hand, but the procedure for sketching a root locus plot consists of only four basic steps, because an exact root locus plot can (and will in practice) easily be generated using software tools like MATLAB. What is necessary however in order to verify and interpret the computer-generated plots, is to understand the underlying principles and

to develop a feeling for how the closed-loop dynamics depend on the controller parameters. Since controllers are usually implemented digitally, a very brief introduction to digital control is given in Chapter 5.

The last section of each chapter contains a number of exercises, which are an important part of this course and should be solved to gain a full understanding of the material presented in the chapter. Sometimes a derivation or proof is left as an exercise; more often, the exercises serve to illustrate the concepts developed in a chapter with examples. These are either pencil and paper examples or examples that should be solved with the help of MATLAB and SIMULINK. Some exercises or parts of exercises, which are of a more challenging nature and may be skipped at first reading, are marked with an asterisk.

Chapter 1

Linear Dynamic Systems

1.1 Differential Equation Models

The design of a feedback control system requires a mathematical model of the plant to be controlled. Such a model is needed for two purposes: a) to decide on a control strategy, and b) to analyse the response of the closed-loop system - analytically or in simulation trials - before a controller is applied to the real plant. To obtain a good model that captures the dynamic behaviour of the plant and is at the same time not too complex for the design process, is a difficult task that may take a considerable part of a control engineer's time. Often two different models are used for (a) and (b). Here we present the basic principles of modelling for the purpose of linear controller design, and illustrate these with simple examples.

The dynamic behaviour of a plant is usually described by ordinary differential equations, which are in many cases nonlinear. It is often possible to linearize these equations about the operating point of interest, see Example 1.3. In this course we will assume that the systems we are interested in have already been linearized and are modelled by linear ordinary differential equations.

When constructing a model, it is important to know what its input and output signals are. The *input signal* is an external stimulus, applied to the system in order to produce a desired response. The *output signal* is the response of the system variable of interest to this input.

Example 1.1

As an example, consider the mechanical system shown in Figure 1.1. The equation of motion for a mechanical system is based on Newton's law

$$F = ma,$$

where F is the vector sum of the forces applied to a body, a the vector acceleration of the body, and m its mass. In the example, a mass is connected to a spring with spring constant k and a damper with damping coefficient b. The input to the system is the external force $u(t)$, and we consider the output to be the displacement $y(t)$ of the mass. The displacement $y(t)$ is measured from the equilibrium position where the force applied by the spring is zero. Assuming that all components are linear, application of Newton's law yields the equation of motion

$$u(t) - ky(t) - b\dot{y}(t) = m\ddot{y}(t)$$

or

$$\ddot{y}(t) + \frac{b}{m}\,\dot{y}(t) + \frac{k}{m}\,y(t) \;=\; \frac{u(t)}{m}. \tag{1.1}$$

The second order differential equation (1.1) describes the dynamic behaviour of the system: the right hand side involves the input signal $u(t)$ and the left hand side the output signal $y(t)$. If the initial values $y(0)$ and $\dot{y}(0)$ and the input signal up to a given time t are known, then one can solve (1.1) and compute the output signal.
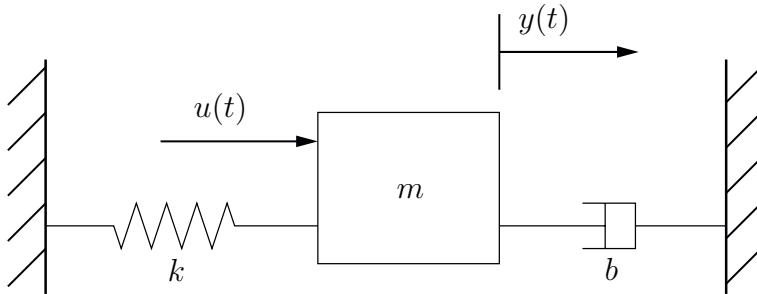

Figure 1.1: Mechanical system

Linear Time Invariant Systems

The equation of motion (1.1) is a *linear differential equation* - the left and right hand side represent linear combinations of output and input and their derivatives, respectively. Moreover, the coefficients (here functions of m, b and k) are *time invariant*, their values are constant. Systems that can be described by linear differential equations with time-invariant coefficients are called *linear time-invariant systems*, or *LTI systems*. The properties of linearity and time invariance can be characterized directly in terms of system behaviour.

Linearity

Consider two arbitrary input signals $u_1(t)$ and $u_2(t)$, and assume that $y_1(t)$ is the response to $u_1(t)$ and $y_2(t)$ the response to $u_2(t)$. Linearity of a system implies that for any two

input signals and for any real constants α_1 and α_2 the response to the signal $\alpha_1 u_1(t) + \alpha_2 u_2(t)$ is $\alpha_1 y_1(t) + \alpha_2 y_2(t)$.

Time Invariance

Consider an arbitrary input signal $u(t)$ and its response $y(t)$. Time-invariance of a system implies that for any time delay $d > 0$ the response to the delayed input $u(t-d)$ is $y(t-d)$.

Example 1.2

As a second example of modelling a dynamic system, consider the electrical circuit shown in Figure 1.2. It contains an inductance L, a resistance R and a capacitance C. Applying Kirchhoff's voltage law to the system, we obtain the equations

$$L\frac{di}{dt} + Ri + v_o = v_i$$
$$C\frac{dv_o}{dt} = i.$$

Eliminating the current i between these equations yields

$$LC\frac{d^2v_o}{dt^2} + RC\frac{dv_o}{dt} + v_o = v_i,$$

and identifying v_i with the input signal $u(t)$ and v_o with the output signal $y(t)$ leads to the equation

$$\ddot{y}(t) + \frac{R}{L}\,\dot{y}(t) + \frac{1}{LC}\,y(t) = \frac{1}{LC}\,u(t) \tag{1.2}$$
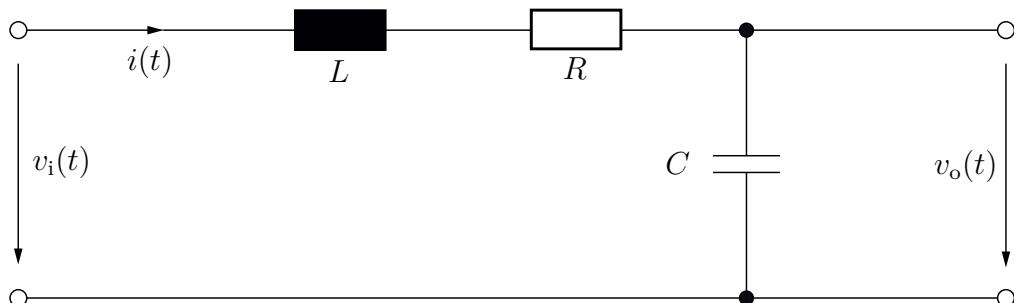

Figure 1.2: Electrical Circuit

Example 1.3

As a last example, consider the water tank shown in Figure 1.3. In steady state operation, water is flowing in and out at the same flow rate Q, and the height of the water level is H. As long as the variations in height are small, the system can be modelled as follows. Let q_i denote the deviation from the steady-state inflow rate, q_o the deviation from the outflow rate, and h the deviation from the steady-state water level. Assume that adjusting the

control valve causes a net inflow $q_i - q_o$, then the volume of stored water will change in an infinitesimal time interval dt by $(q_i - q_o)dt$. The capacitance C of the tank is defined as the change in volume of stored water divided by the resulting change of height, thus we have

$$Cdh = (q_i - q_o)dt.$$

On the right hand side of this first order differential equation we still have the outflow q_o, which depends non-linearly on h. However, if the deviation from the steady state values is not too large, one can approximately describe the system by a linear model. The resistance R of the load valve is then defined as the ratio of change in level difference to the resulting change in outflow rate, i.e.

$$R = \frac{h}{q_o}.$$

Eliminating the outflow q_o between the last two equations yields

$$RC\dot{h} + h = Rq_i,$$

and identifying the change in inflow q_i (determined by the control valve) with the input signal $u(t)$ and the change in height h with the output signal $y(t)$ leads to the first order differential equation

$$\dot{y}(t) + \frac{1}{RC}\,y(t) = \frac{1}{C}\,u(t) \tag{1.3}$$
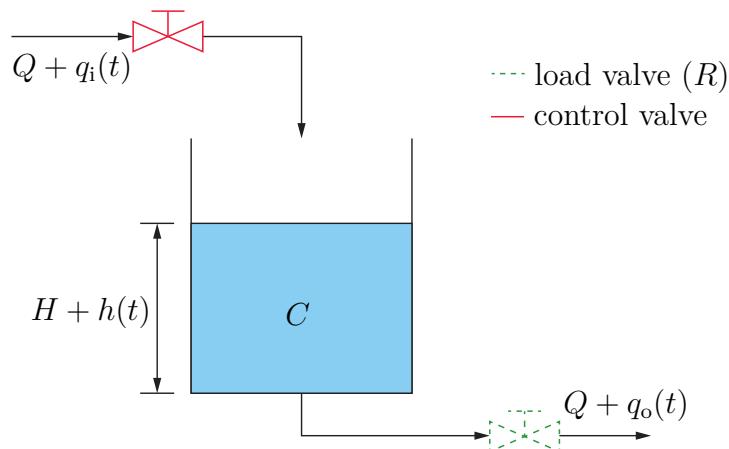
Figure 1.3: Water tank

State Space Models

The examples in this section illustrate how dynamic systems can be modelled by linear ordinary differential equations. The models in this section are of first and second order. When dealing with systems that are described by higher order differential equations, it may be more convenient to rewrite the differential equation in a more compact form. We now illustrate how a nth order linear differential equation model can be transformed into a first order vector differential equation model. We use again the spring-mass-damper system as an example to illustrate the idea. Recall the equation of motion

$$\ddot{y}(t) + \frac{b}{m}\,\dot{y}(t) + \frac{k}{m}\,y(t) \;=\; \frac{1}{m}u(t).$$

Introducing the velocity $v(t) = \dot{y}(t)$ as a second variable, we can write

$$\dot{y}(t) = v(t)$$
$$\dot{v}(t) = -\frac{b}{m}v(t) - \frac{k}{m}y(t) + \frac{1}{m}u(t)$$

Defining the *state vector*

$$x(t) = \begin{bmatrix} y(t) \\ v(t) \end{bmatrix}$$

we can express this in a more compact way as

$$\begin{bmatrix} \dot{y}(t) \\ \dot{v}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} y(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u(t) \tag{1.4}$$

and

$$y(t) = [1 \;\; 0] \begin{bmatrix} y(t) \\ v(t) \end{bmatrix} \tag{1.5}$$

Note that equations (1.4) and (1.5) are just an alternative way of writing the equation of motion that governs the plant dynamics. By introducing a new variable $v(t)$, we obtain a model that involves only first order derivatives; the original second order differential equation is now represented by a first order vector differential equation with a two-dimensional state vector. The same idea can be applied to higher order systems: a nth order differential equation leads to a first order vector differential equation with a n-dimensional state vector. The new variables which make up the state vector are called the *state variables* of the system. They define the state of the system at any given time. Equations (1.4) and (1.5) are an example of a *state space model* which in general has the form

$$\dot{x}(t) = Ax(t) + bu(t) \tag{1.6}$$
$$y(t) = cx(t) + du(t) \tag{1.7}$$

In the above example we have

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} y(t) \\ v(t) \end{bmatrix}$$

and

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}, \qquad b = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}, \qquad c = [1 \ 0], \qquad d = 0.$$

Equation (1.6) represents the dynamics of the plant, it is called the *state equation* . Equation (1.7) defines the plant output as a linear combination of state variables and input and is called the *output equation* or *measurement equation* . The $n \times n$ matrix A is called the *system matrix* ; b and c are n-dimensional column and row vectors, respectively. In the above example, there is no direct feedthrough from input to output, therefore we have $d = 0$.

Advanced techniques for an automated design of complex control systems are usually based on state space models. On the other hand, classical analysis and design techniques are based on the concept of a *transfer function.* In this course, we will study methods based on transfer function models. State space methods will be discussed extensively in the course "Control Systems Theory and Design".

1.2 Laplace Transform and Transfer Functions

In the previous section, linear systems were modelled by linear differential equations of the form

$$\frac{d^n}{dt^n} y(t) + a_{n-1} \frac{d^{n-1}}{dt^{n-1}} y(t) + \ldots + a_1 \dot{y}(t) + a_0 y(t) =$$
$$b_m \frac{d^m}{dt^m} u(t) + b_{m-1} \frac{d^{m-1}}{dt^{m-1}} u(t) + \ldots + b_1 \dot{u}(t) + b_0 u(t) \quad (1.8)$$

The examples considered so far were of first and second order and did not involve derivatives of the input signal $u(t)$ (a model involving the derivative of $u(t)$ will be presented in Example 1.7). In (1.8) the order of the highest derivative of output and input are n and m, respectively. Later in this chapter it will be shown that if a differential equation of the form (1.8) represents a physical system, we must have $n \geq m$.

In this section, we will discuss how the Laplace transform can be used to convert differential equations of the form (1.8) into frequency domain models - represented by *transfer functions* - that involve only algebraic operations.

We start with a brief review of the Laplace transform. We consider the transformation of piecewise continuous time domain signals $x(t)$ that are zero for negative times, i.e.

$$x(t) = 0, \quad t < 0.$$

The Laplace transform of $x(t)$ is defined as

$$\mathcal{L}[x(t)] = \int_0^\infty x(t) e^{-st} \, dt, \tag{1.9}$$

where s is a complex variable; we will later see that s can be interpreted as a complex frequency. We will also use the notation

$$X(s) = \mathcal{L}[x(t)]$$

for the Laplace transform of a time signal.

Existence of the Laplace Transform

The Laplace transform of a signal $x(t)$ as defined in (1.9) exists if the integral converges. This is the case for values of $s = \sigma + j\omega$ if

$$\lim_{t\to\infty} |x(t)e^{-\sigma t}| = 0. \tag{1.10}$$

If a real number σ exists that satisfies (1.10), the signal $x(t)$ as a function of time is said to be of *exponential order*. For example, the function Ke^{at} is of exponential order for all values of K and a and therefore has a Laplace transform. On the other hand, the function Ke^{t^2} is not of exponential order and does not have a Laplace transform.

The smallest value of σ that satisfies (1.10) for a given signal $x(t)$ is called the *abscissa of convergence*, and the region in the complex plane defined by $\mathrm{Re}(s) > \sigma$ is called the *region of convergence*. The signals we typically encounter in control applications are all of exponential order, and we will always assume that the integral in (1.9) is taken in the region of convergence. In practice, the question of existence and the region of convergence are usually no issues of concern.

Linearity

It follows from the definition that taking the Laplace transform is a linear operation, i.e. for two signals $x_1(t)$ and $x_2(t)$ and for any real constants α_1, α_2 we have

$$\mathcal{L}[\alpha_1 x_1(t) + \alpha_2 x_2(t)] = \alpha_1 \mathcal{L}[x_1(t)] + \alpha_2 \mathcal{L}[x_2(t)] = \alpha_1 X_1(s) + \alpha_2 X_2(s). \tag{1.11}$$

Unit Step

We now compute the Laplace transforms of three simple but important functions. First, we compute the Laplace transform of the *unit step function*

$$\sigma(t) = \begin{cases} 1, & t \geq 0, \\ 0, & t < 0. \end{cases}$$

Applying the definition (1.9) yields

$$\mathcal{L}[\sigma(t)] = \int_0^\infty \sigma(t)e^{-st}\,dt = \int_0^\infty e^{-st}\,dt$$
$$= -\frac{1}{s}e^{-st}\Big|_0^\infty = 0 + \frac{1}{s};$$

so we have the result

$$\mathcal{L}[\sigma(t)] = \frac{1}{s}.$$

Exponential

Next, consider the *exponential function*

$$x(t) = \begin{cases} e^{-at}, & t \geq 0, \\ 0, & t < 0. \end{cases}$$

which can also be written as $x(t) = e^{-at}\sigma(t)$. Its Laplace transform can be computed as

$$\mathcal{L}[e^{-at}\sigma(t)] = \int_0^\infty e^{-at}e^{-st}\,dt$$
$$= \int_0^\infty e^{-(s+a)t}\,dt = \frac{1}{s+a}.$$

Sinusoid

To compute the Laplace transform of the *sinusoidal function*

$$x(t) = \begin{cases} \sin bt, & t \geq 0, \\ 0, & t < 0, \end{cases}$$

we use the identity

$$\sin bt = \frac{1}{2j}\left(e^{jbt} - e^{-jbt}\right)$$

and the above results to obtain

$$\mathcal{L}[\sin bt\sigma(t)] = \frac{b}{s^2 + b^2}.$$

A collection of time functions and their Laplace transforms is listed in Table 1.1. In this table, all time-domain signals are multiplied by the unit step function $\sigma(t)$ because we consider only signals which are zero for $t < 0$.

Time Delay

Next, we review some properties of Laplace transforms. First, consider a time function $x(t)$ that is shifted by a time delay d. To compute the Laplace transform of the delayed signal $x(t - d)$, we use the definition of the Laplace transform

$$\mathcal{L}[x(t - d)] = \int_0^\infty x(t - d)e^{-st}\,dt$$

and a change of integration variables

$$t \;\rightarrow\; \tau = t - d$$

to obtain

$$\mathcal{L}[x(t-d)] = e^{-sd} \int_0^\infty x(\tau)e^{-s\tau}\, d\tau;$$

or

$$\mathcal{L}[x(t-d)] = e^{-sd}\mathcal{L}[x(t)]. \tag{1.12}$$

Thus, the Laplace transform of a signal delayed by time d is equal to the Laplace transform of the undelayed signal, multiplied by e^{-sd}.

Unit Impulse

A signal that plays an important role in control theory and more generally in system theory is the *unit impulse function* $\delta(t)$ (also called *Dirac delta function*). It is defined by

$$\delta(t) = 0, \qquad t \neq 0$$

and

$$\int_{-\infty}^t \delta(\tau)d\tau = \sigma(t).$$

Strictly speaking $\delta(t)$ is not a function but a distribution. (A distribution is a generalised function, constructed in a way that allows to differentiate functions whose derivatives do not exist.) The following definition of a unit impulse will be sufficient for our purpose: consider the rectangular pulse

$$r_\epsilon(t) = \begin{cases} \frac{1}{\epsilon}, & 0 \leq t \leq \epsilon, \\ 0 & \text{else,} \end{cases}$$

and take the limit as $\epsilon \to 0$, i.e.

$$\delta(t) = \lim_{\epsilon \to 0} r_\epsilon(t).$$

To compute the Laplace transform of $r_\epsilon(t)$, observe that the rectangular pulse can be written as

$$r_\epsilon(t) = \frac{1}{\epsilon}\Big(\sigma(t) - \sigma(t-\epsilon)\Big).$$

Using (1.12), we have

$$\mathcal{L}[\delta(t)] = \lim_{\epsilon \to 0} \mathcal{L}[r_\epsilon(t)] = \lim_{\epsilon \to 0} \frac{1 - e^{-s\epsilon}}{\epsilon s}$$

$$= \lim_{\epsilon \to 0} \frac{\frac{d}{d\epsilon}(1 - e^{-s\epsilon})}{\frac{d}{d\epsilon}(\epsilon s)} = \lim_{\epsilon \to 0} \frac{se^{-s\epsilon}}{s} = 1;$$

so we obtain the important result

$$\mathcal{L}[\delta(t)] = 1.$$

Table 1.1: Laplace transforms

Number	$x(t)$	$X(s)$
1	$\delta^{(n)}(t)$	s^n
2	$\delta'(t)$	s
3	$\delta(t)$	1
4	$\sigma(t)$	$\dfrac{1}{s}$
5	$t\,\sigma(t)$	$\dfrac{1}{s^2}$
6	$\dfrac{t^{n-1}}{(n-1)!}\,\sigma(t)$	$\dfrac{1}{s^n}$
7	$e^{-at}\,\sigma(t)$	$\dfrac{1}{s+a}$
8	$te^{-at}\,\sigma(t)$	$\dfrac{1}{(s+a)^2}$
9	$\sin bt\,\sigma(t)$	$\dfrac{b}{s^2+b^2}$
10	$\cos bt\,\sigma(t)$	$\dfrac{s}{s^2+b^2}$
11	$e^{-at}\sin bt\,\sigma(t)$	$\dfrac{b}{(s+a)^2+b^2}$
12	$e^{-at}\cos bt\,\sigma(t)$	$\dfrac{s+a}{(s+a)^2+b^2}$
13	$\dfrac{e^{-at}-e^{-bt}}{b-a}\,\sigma(t)$	$\dfrac{1}{(s+a)(s+b)}$
14	$\dfrac{(c-a)e^{-at}-(c-b)e^{-bt}}{b-a}\,\sigma(t)$	$\dfrac{s+c}{(s+a)(s+b)}$

Differentiation and Integration

One of the most useful properties of the Laplace transform is the fact that differentiation and integration in time domain are transformed into algebraic operations in frequency domain. To find the Laplace transform of the time derivative of a function $x(t)$, we start with the right hand side of the definition (1.9) and integrate by parts

$$\int_0^\infty x(t)e^{-st}\,dt = x(t)\left.\frac{e^{-st}}{-s}\right|_0^\infty - \int_0^\infty \frac{d}{dt}x(t)\frac{e^{-st}}{-s}\,dt$$

or

$$X(s) = \frac{x(0)}{s} + \frac{1}{s}\int_0^\infty \frac{d}{dt}x(t)e^{-st}\,dt.$$

Observing that the last integral is the Laplace transform of the time derivative of $x(t)$, we have

$$\mathcal{L}\left[\frac{d}{dt}x(t)\right] = sX(s) - x(0).$$

Applying the above procedure to $\frac{d}{dt}x(t)$ instead of $x(t)$ yields

$$\mathcal{L}\left[\frac{d^2}{dt^2}x(t)\right] = s^2X(s) - sx(0) - \dot{x}(0),$$

and repeating this n times leads to

$$\mathcal{L}\left[\frac{d^n}{dt^n}x(t)\right] = s^nX(s) - s^{n-1}x(0) - s^{n-2}\dot{x}(0) - s^{n-3}\ddot{x}(0) - \ldots - x^{(n-1)}(0). \qquad (1.13)$$

A formula for the Laplace transform of the integral of the time function $x(t)$ can be derived by integration of parts, similiar to the derivation of (1.13). We obtain

$$\mathcal{L}\left[\int_0^t x(\tau)\,d\tau\right] = \frac{1}{s}X(s). \qquad (1.14)$$

In practice, the initial values $x(0)$, $\dot{x}(0)$ etc are usually assumed to be zero. In that case, the Laplace transform of the nth time derivative simplifies to

$$\mathcal{L}\left[\frac{d^n}{dt^n}x(t)\right] = s^nX(s). \qquad (1.15)$$

Transfer Functions

It is the property (1.15) of the Laplace transform that makes it possible to convert differential equations in time domain into algebraic equations in frequency domain; this leads to the concept of a *transfer function*. Consider a plant with dynamics governed by a differential equation of the form (1.8).

Taking the Laplace transform on both sides and using (1.15), we have

$$s^n Y(s) + a_{n-1}s^{n-1}Y(s) + \ldots + a_1 s Y(s) + a_0 Y(s) =$$
$$b_m s^m U(s) + b_{m-1}s^{m-1}U(s) + \ldots + b_1 s U(s) + b_0 U(s)$$

or

$$\left(s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0\right)Y(s) = \left(b_m s^m + b_{m-1}s^{m-1} + \ldots + b_1 s + b_0\right)U(s).$$

Note that here we assumed zero initial values - this assumption is usually made when working with transfer functions. Dividing by the polynomial in s on the left hand side gives

$$Y(s) = \frac{b_m s^m + b_{m-1}s^{m-1} + \ldots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0} U(s),$$

and we see that in frequency domain, the output $Y(s)$ to a given input $U(s)$ is obtained simply by multiplying the input with the rational function

$$G(s) = \frac{b_m s^m + b_{m-1}s^{m-1} + \ldots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0}.$$

The function $G(s)$ represents the dynamic properties of a system and is called its transfer function. In general we have

$$Y(s) = G(s)U(s). \tag{1.16}$$

State Space Models and Transfer Functions

We can also use the the Laplace transform to obtain a transfer function from a given state space model (1.6) and (1.7). Assuming that the initial values of the state variables are zero, taking Laplace transforms in (1.6) we have

$$sX(s) = AX(s) + bU(s)$$

from which we obtain

$$X(s) = (sI - A)^{-1}bU(s).$$

Here $X(s)$ is a n-dimensional column vector containing the Laplace transforms of the state variables, and I represents the n by n identity matrix. Substituting the above in the Laplace transform of the output equation yields then

$$Y(s) = \left(c(sI - A)^{-1}b + d\right)U(s).$$

Thus, the transfer function expressed in terms of a state space model is

$$G(s) = c(sI - A)^{-1}b + d. \tag{1.17}$$

Final Value Theorem

If a signal $x(t)$ converges to a finite, constant value as $t \to \infty$, this value can be obtained from the Laplace transform $X(s)$. To see this, take the limit of the Laplace transform of $\frac{d}{dt}x(t)$ as $s \to 0$

$$\lim_{s \to 0} \int_0^\infty \left(\frac{d}{dt}x(t) \right) e^{-st} \, dt = \lim_{s \to 0}(sX(s) - x(0))$$

or

$$x(t)|_0^\infty = x(\infty) - x(0) = \lim_{s \to 0} sX(s) - x(0).$$

This result is known as the *Final Value Theorem*: assuming $\lim_{t \to \infty} x(t)$ exists, we have

$$\lim_{t \to \infty} x(t) = \lim_{s \to 0} sX(s). \tag{1.18}$$

It is important to keep in mind that (1.18) is meaningful only if the limit on the left hand side exists, i.e. if $x(t)$ converges to a finite, constant value as $t \to \infty$.

Initial Value Theorem

There is a counterpart to the final value theorem: the *Initial Value Theorem*. It can be used to find the value of $x(t)$ as $t \to 0^+$, i.e. the limit of $x(t)$ when approaching zero from the right. While we assume that all signals are zero for $t < 0$, we will encounter signals that "jump" at $t = 0$ to non-zero values. To avoid the discontinuity at $t = 0$ when taking the Laplace transform, we slightly modify the Laplace integral and consider

$$\lim_{s \to \infty} \mathcal{L}_+[\dot{x}(t)] = \lim_{s \to \infty} \int_{0^+}^\infty \dot{x}(t)e^{-st}dt$$

Now the left hand side can be obtained from (1.13), and the right hand side approaches zero as $s \to \infty$. Thus we have

$$\lim_{s \to \infty} sX(s) - x(0^+) = 0$$

or

$$x(0^+) = \lim_{s \to \infty} sX(s). \tag{1.19}$$

Convolution Integral

Later in this section we will encounter the *convolution integral*

$$x_1(t) * x_2(t) = \int_0^t x_1(\tau)x_2(t - \tau)d\tau$$

for two given signals $x_1(t)$ and $x_2(t)$. If $X_1(s)$ and $X_2(s)$ are the Laplace transforms of $x_1(t)$ and $x_2(t)$, respectively, one can show that

$$\mathcal{L}[x_1(t) * x_2(t)] = X_1(s)X_2(s). \tag{1.20}$$

The proof is left to Problem 1.2.

Some important properties of Laplace transforms are summarized in Table 1.2.

Table 1.2: Properties of Laplace transforms

No.	Time Function	Laplace Transform	Property
1	$\alpha_1 x_1(t) + \alpha_2 x_2(t)$	$\alpha_1 X_1(s) + \alpha_2 X_2(s)$	Linearity
2	$x(t-d)$	$e^{-sd} X(s)$	Time delay
3	$\dfrac{d^n}{dt^n} x(t)$	$s^n X(s) - s^{n-1} x(0) -$ $s^{n-2} \dfrac{d}{dt} x(0) - \ldots - \dfrac{d^{n-1}}{dt^{n-1}} x(0)$	Differentiation
4	$\displaystyle\int_0^t x(\tau)\, d\tau$	$\dfrac{1}{s} X(s)$	Integration
5	$\displaystyle\lim_{t\to\infty} x(t)$	$\displaystyle\lim_{s\to 0} sX(s)$	Final-value theorem
6	$x(0^+)$	$\displaystyle\lim_{s\to\infty} sX(s)$	Initial-value theorem

Poles of Transfer Functions

Consider a system that can be modelled by the transfer function

$$G(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \ldots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0}.$$

The values of s at which the denominator of $G(s)$ takes the value zero, and therefore at which $G(s)$ becomes infinite, are called the *poles* of the transfer function $G(s)$. The locations of the poles in the complex plane determine the dynamic behaviour of the system; for this reason the denominator polynomial of the transfer function is called the *characteristic polynomial*. Setting the characteristic polynomial to zero yields the characteristic equation. The following examples illustrate the effect of the pole locations on the dynamic response of a system.

Example 1.4

Consider a system governed by the differential equation

$$\ddot{y}(t) + 5\dot{y}(t) + 6y(t) = 6u(t).$$

Its transfer function is

$$G(s) = \frac{6}{s^2 + 5s + 6} = \frac{6}{(s+2)(s+3)},$$

so the system has two poles, one at $s = -2$ and one at $s = -3$. We are interested in the effect a sudden change in the value of the input $u(t)$ will have on the output $y(t)$; thus we assume that the input is switched at time $t = 0$ from 0 to 1. This input signal can be modelled by the unit step function $\sigma(t)$. To compute the resulting output, we take the Laplace transform of the input and use (1.16) to obtain

$$Y(s) = G(s)\frac{1}{s} = \frac{6}{s(s+2)(s+3)}.$$
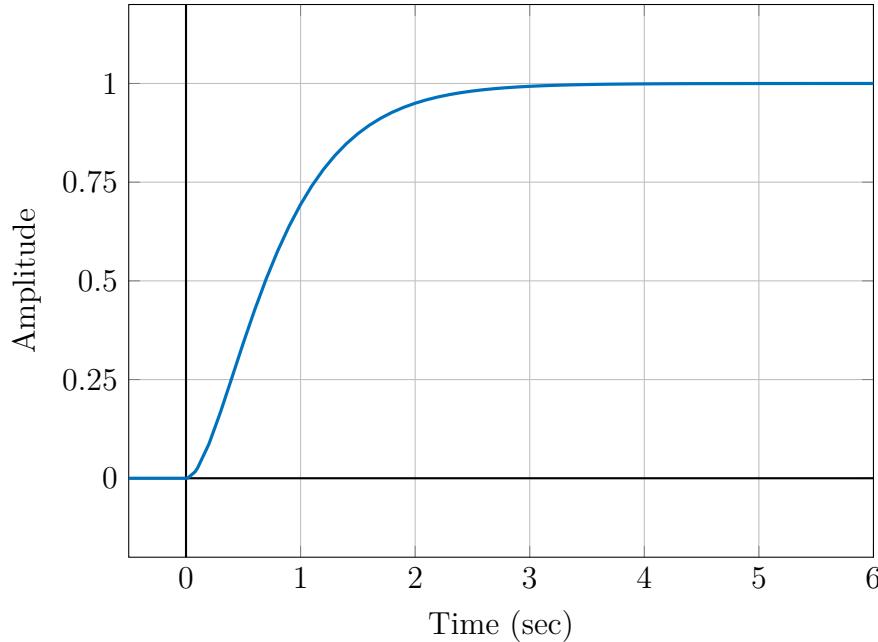
Figure 1.4: Step response for Example 1.4

The last expression is not contained in the table of Laplace transforms, but we can expand it into partial fractions

$$Y(s) = \frac{6}{s(s+2)(s+3)} = \frac{1}{s} - \frac{3}{s+2} + \frac{2}{s+3}.$$

The first term on the right hand side is again the Laplace transform of the unit step, and the remaining terms are the Laplace transforms of exponential functions. Using the Laplace table, we obtain the time domain expression of the plant response to a step change at the input

$$y(t) = \begin{cases} 1 - 3e^{-2t} + 2e^{-3t}, & t \geq 0, \\ 0, & t < 0. \end{cases} \tag{1.21}$$

This response is plotted in Figure 1.4. To show how the poles of the transfer function affect the response, the three components of the step response are shown separately in Figure 1.5, where $y_1(t) = \sigma(t)$, $y_2(t) = 2e^{-3t}\sigma(t)$ and $y_3(t) = -3e^{-2t}\sigma(t)$.
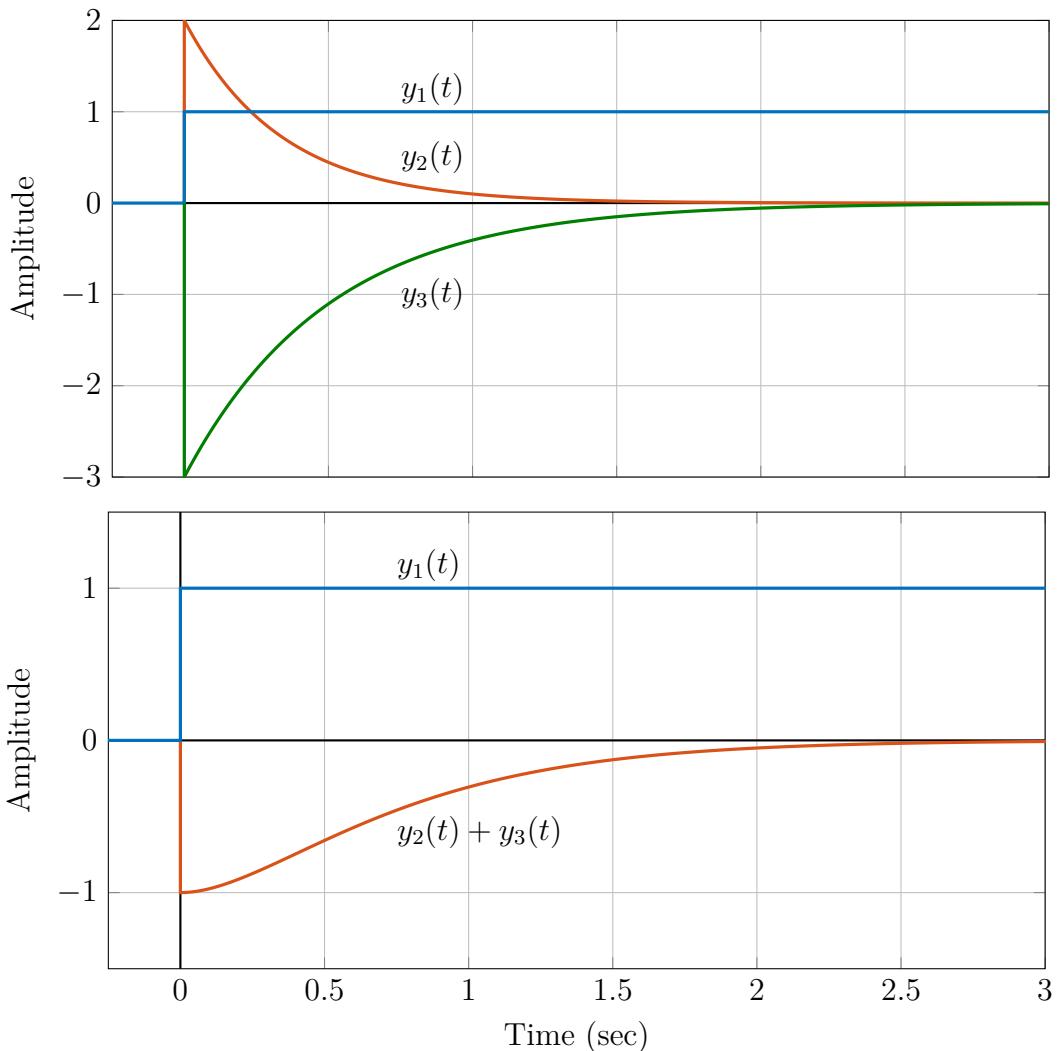
Figure 1.5: Components of step response for Example 1.4

The step function $y_1(t)$ is the *steady state response* of the plant to the step change, it represents the steady state effect of the external input. The exponentials $y_2(t)$ and $y_3(t)$ are components of the *transient response*, or the *characteristic response* of the system. They represent the transition from one equilibrium to another, and are determined by the location of the transfer function poles in the complex plane. Because the component $2e^{-3t}$ decays faster than $-3e^{-2t}$, we say that the pole at $s = -3$ is "faster" than the pole at $s = -2$; the values $1/3$ and $1/2$ are called the *time constants* of the responses $y_2(t)$ and $y_3(t)$, respectively.

In the previous example, both poles of the system are located on the real axis of the complex plane. In the following example, the poles of the transfer function are form a complex conjugate pair.

Example 1.5

Consider a system with transfer function

$$G(s) = \frac{10}{s^2 + 2s + 17}$$

The poles are located at $s = -1 \pm j4$. Applying a unit step input, and expanding the resulting output $Y(s)$ in partial fractions yields

$$Y(s) = \frac{10}{s(s+1-j4)(s+1+j4)} = \frac{A}{s} + \frac{B}{s+1-j4} + \frac{\bar{B}}{s+1+j4},$$

with $A = 1/1.7$, $B = -0.2941 + j0.0735$ and $\bar{B}$ the complex conjugate of B. The above can also be written as

$$Y(s) = \frac{A}{s} + \frac{C}{2}\frac{e^{j\varphi}}{s+1-j4} + \frac{C}{2}\frac{e^{-j\varphi}}{s+1+j4};$$

with $C = 2|B| = 0.6063$ and $\varphi = \arg B = 2.8966$. Using the Laplace table in the same
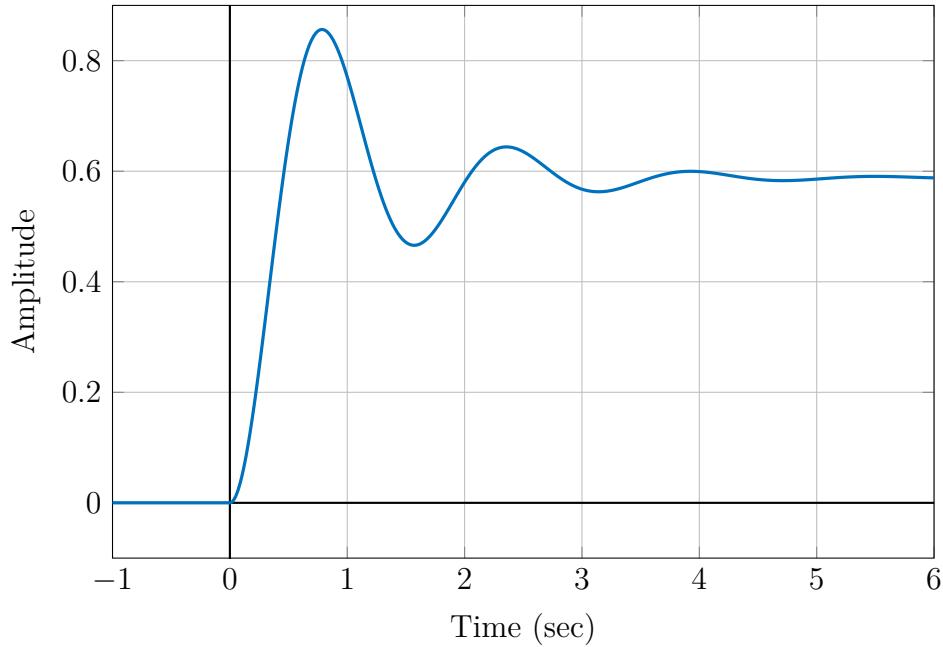
Figure 1.6: Step response for Example 1.5

way as in the previous example gives

$$y(t) = \left(A + \frac{C}{2}e^{j\varphi}e^{-t}e^{j4t} + \frac{C}{2}e^{-j\varphi}e^{-t}e^{-j4t}\right)\sigma(t)$$
$$= \left(A + \frac{C}{2}e^{-t}(e^{j(4t+\varphi)} + e^{-j(4t+\varphi)})\right)\sigma(t)$$
$$= \left(A + Ce^{-t}\cos(4t + \varphi)\right)\sigma(t).$$

The step response of this system is shown in Figure 1.6; the steady state response $y_1(t) = A\sigma(t)$ and the transient response $y_2(t) = Ce^{-t}\cos(4t + \varphi)\sigma(t)$ are shown separately in Figure 1.7. Note that the real part of the poles determines the rate of decay, whereas the imaginary part determines the frequency of the damped oscillation.
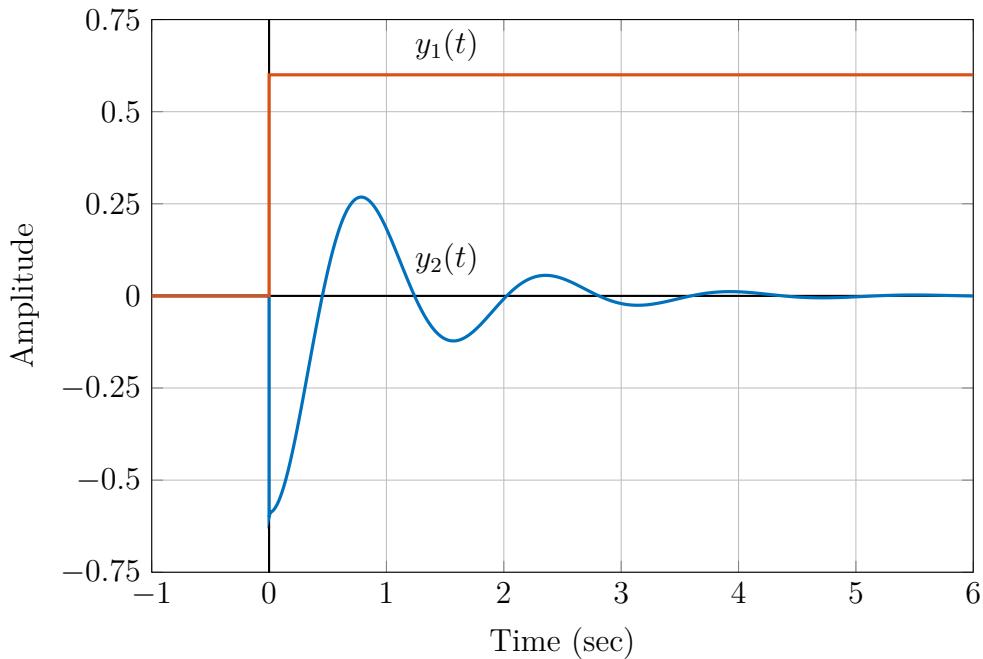
Figure 1.7: Steady state and transient response for Example 1.5

Static Gain

The last two examples can also be used to illustrate the final value theorem. The steady state value of the step response in Example 1.4 is found from Figure 1.4 to be 1, whereas in Example 1.5 it is 0.588. Evaluating $sY(s)$ at $s = 0$ for both examples confirms these values.

In general, we have for the step response

$$Y(s) = G(s)\frac{1}{s}$$

and for its steady state value

$$\lim_{t\to\infty} y(t) = \lim_{s\to 0} sG(s)\frac{1}{s} = \lim_{s\to 0} G(s).$$

Thus, the value of $G(0)$ - if it exists - represents the *static gain* (also referred to as *DC gain*) of the system.

Transfer Function, Impulse Response and Step Response

The transfer function $G(s)$ is a rational function of the complex variable s; it has the same form as the Laplace transform of a time signal. One might ask whether the transfer

function itself represents the response to a particular input signal. This is indeed the case: recall that the Laplace transform of the unit impulse function is 1, and use

$$Y(s) = G(s)U(s)$$

to compute the response to a unit impulse as

$$Y(s) = G(s)\mathcal{L}[\delta(t)] = G(s).$$

This shows that the transfer function is equal to the Laplace transform of the response to a unit impulse - the *impulse response* of the system.

Superposition and Convolution

Recalling from Section 1.2 that multiplication in frequency domain is equivalent to convolution in time domain, we can transform equation (1.16) back into time domain to obtain

$$y(t) = \int_0^t g(\tau)u(t - \tau)d\tau$$

or alternatively

$$y(t) = \int_0^t u(\tau)g(t - \tau)d\tau. \tag{1.22}$$

An interpretation of these convolution integrals is that the system response $y(t)$ to an arbitrary input $u(t)$ can be seen as the superposition of impulse responses weighted by the values of $u(t)$. This point is explored further in the Problem 1.12.

Pole Locations and Stability

We have seen that the transfer function $G(s)$ contains information about both steady state and transient behaviour: if it exists, the static gain is $\lim_{s \to 0} G(s)$, and the transient response is determined by the poles of the transfer function. The n poles of a plant described by a nth order linear differential equation are either real or come in complex conjugate pairs, so the transfer function can be written as a sum of n terms

$$G(s) = \frac{C_1}{s - p_1} + \frac{C_2}{s - p_2} + \ldots + \frac{C_\nu}{s - p_\nu} + \frac{\bar{C}_\nu}{s - \bar{p}_\nu} + \ldots$$

The examples considered so far show that real poles give rise to exponentially decaying components in the transient response, and the complex conjugate pole pairs give rise to oscillating components. For a real, negative pole at $p = \sigma < 0$, the exponential function e^{pt} decays with time constant $1/\sigma$. If a real pole is however positive, the response grows exponentially with time; a system having such a pole is said to be *unstable* . For a complex pole pair at $p = \sigma \pm j\omega$, the calculation in Example 1.5 shows that the transient response has the form $Ae^{\sigma t}\cos(\omega t + \varphi)$, it is an oscillation with frequency ω and a time-varying amplitude. If the real part σ is negative, the amplitude of the oscillation decays with
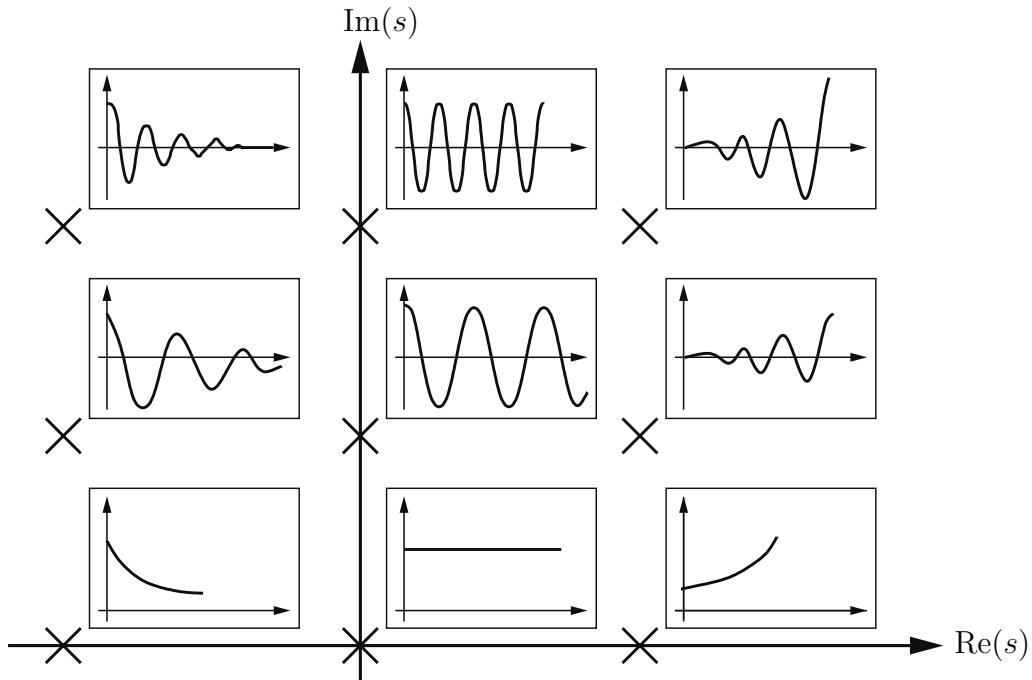
Figure 1.8: Components of the transient response corresponding to different pole locations

time constant $1/\sigma$, and if σ is positive, the amplitude grows exponentially with time and again a system having such a pole pair is said to be unstable. If σ is zero, the amplitude neither grows nor decays, and the system is said to be *marginally stable* .

We can summarize these results by stating that poles with a negative real part lead to components in the transient response that die out in time, whereas poles with a positive real part lead to a response that grows with time. In terms of pole locations in the complex plane, we find that poles in the left half plane correspond to exponentially decaying components in the transient response and poles in the right half plane to exponentially growing ones. A system is said to be stable if and only if it has all its poles in the left half plane.

Physical Realizability

The transfer function of a nth order system has the form

$$G(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \ldots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0}.$$

This system has n poles and m zeros, and we will now see that if it is physically realizable we must have $n \geq m$. To see why a physically realizable system cannot have more zeros than poles, consider the step response

$$Y(s) = G(s)\frac{1}{s}$$

From the initial value theorem it follows that

$$y(0^+) = \lim_{s \to \infty} sY(s) = \lim_{s \to \infty} G(s)$$

For a system with more zeros than poles, i.e. $m > n$, we have

$$y(0^+) = \lim_{s \to \infty} \left(b_m s^{m-n} + \alpha_{m-1} s^{m-1-n} + \ldots + \alpha_0 + \frac{\beta_{n-1} s^{n-1} + \ldots + \beta_1 s + \beta_0}{s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0} \right)$$

where the α_i and β_i are obtained by polynomial division. Since $y(0^+)$ will be infinite as $s \to \infty$, this transfer function cannot represent a physically realizable system.

On the other hand, if $n > m$ it is clear that

$$y(0^+) = \lim_{s \to \infty} G(s) = 0$$

which is a typical property of the step response of a physical system.

If $n = m$ we have

$$y(0^+) = \lim_{s \to \infty} \frac{b_n s^n + \ldots + b_0}{s^n + + \ldots + a_0} = b_n.$$

It is common in the literature to refer to a system as physically realizable when $n = m$. Note however that, strictly speaking, the resulting step response is not realizable, because the output variable of a physical system cannot "jump" instantaneously from zero to a non-zero value b_n. In practice, transfer functions with $n = m$ are often used to describe controller dynamics. This can be a reasonable approximation if the controller dynamics are much faster than the plant dynamics - compared with the dominant time constant of the plant response, the fast response of the controller at $t = 0$ may look like a step change. We will in this course refer to systems as physically realizable when $n \geq m$.

Systems with $n \geq m$ are called *proper*, and systems with $n > m$ are called *strictly proper*. A system with $n = m$ is called *bi-proper*. The difference $n - m$ between the number of poles and zeros of a system is called its *pole excess*.

Transfer Functions and Block Diagrams
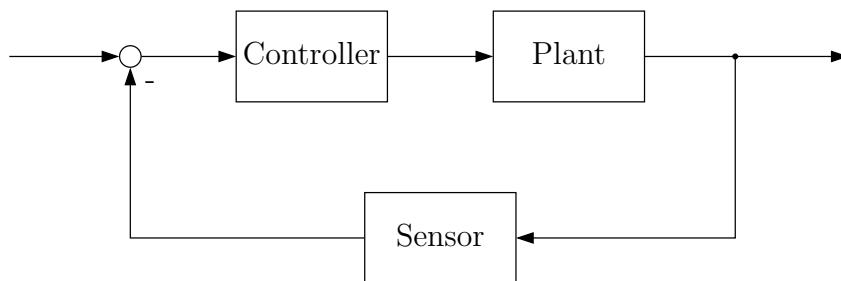
Figure 1.9: Typical control loop

Transfer functions are used to model the dynamic behaviour of the components in a control system. Figure 1.9 shows a typical control loop represented by a block diagram.

Block diagrams are often used in analysis and design of control systems, and each block in the diagram can be modelled by a transfer function. To be able to work with block diagrams, we need to know how to combine blocks and their transfer functions. Three basic configurations are series connection, parallel connection and a closed-loop structure.

A series connection of two transfer functions is shown in Figure 1.10. With $Y_1(s) = G_1(s)U_1(s)$ and $Y_2(s) = G_2(s)U_2(s)$ we have $U_2(s) = Y_1(s)$, and the overall transfer function is the product of the two transfer functions $G_2(s)G_1(s)$.

Figure 1.10: Series connection

Figure 1.11 shows a parallel connection of two transfer functions. It is easy to see that the overall transfer functions is the sum of the two transfer functions $G_1(s) + G_2(s)$.
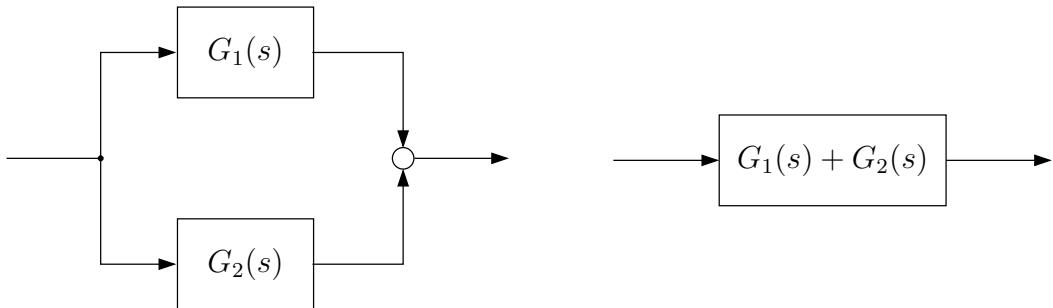
Figure 1.11: Parallel connection

Two transfer functions in a loop are shown in Figure 1.12, this configuration is important for studying control loops. We have

$$Y_1(s) = G_1(s)U_1(s) = G_1(s)(R(s) + Y_2(s))$$

or

$$Y_1(s) = G_1(s)R(s) + G_1(s)G_2(s)Y_1(s).$$

Thus

$$(1 - G_1(s)G_2(s))Y_1(s) = G_1(s)R(s),$$

so we can write

$$Y_1(s) = G_{cl}(s)R(s),$$

where $G_{cl}(s)$ is the *closed-loop transfer function*

$$G_{cl}(s) = \frac{G_1(s)}{1 - G_1(s)G_2(s)}.$$
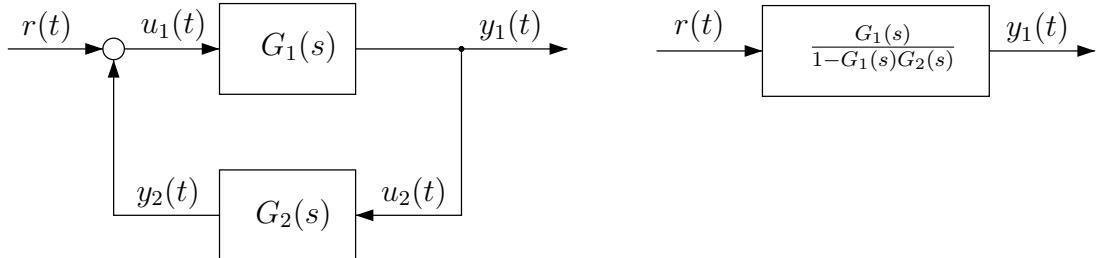
Figure 1.12: Closed loop configuration

An easy way to remember this formula is

$$\text{closed loop transfer function} \; = \; \frac{\text{forward gain}}{1 - \text{loop gain}} \; ,$$

where "forward gain" stands for the transfer function in the forward path, and "loop gain" for the transfer function seen when traversing the loop.

1.3   First and Second Order Systems

The transfer functions in Examples 1.1 and 1.2 have the form

$$G(s) = \frac{b_0}{s^2 + a_1 s + a_0}, \tag{1.23}$$

whereas in Example 1.3 we encounter the transfer function

$$G(s) = \frac{b_0}{s + a_0}. \tag{1.24}$$

These transfer functions represent systems described by linear first and second order differential equations. First and second order systems play an important role in feedback design, because it is often possible to simplify the model of a plant with more complex dynamic properties and to describe its dominant features by a first or second order model. For this reason, in this section we will study the properties of first and second order systems, and we will focus on the step response of these systems.

First Order Systems

First order systems are particularly simple to analyze. It is in fact straightforward to show that a system with transfer function (1.24) has the step response shown in Figure 1.13. The response is characterized by the static gain b_0/a_0, and by the *time constant* $\tau = 1/a_0$, which is the time it takes the output to reach 63.2% of its steady state value; see Exercise 1.3.
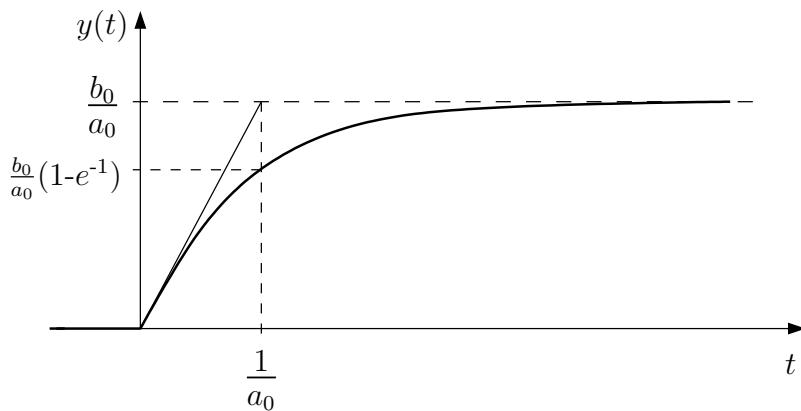
Figure 1.13: Step response of a first order system

Second Order Systems

To investigate the properties of second order systems, we rewrite (1.23) in terms of new parameters that are more directly related to the dynamic behaviour than the coefficients of the differential equations. These parameters are

- the *static gain* $K = G(0)$, which is easily seen to be b_0/a_0,

- the *natural frequency* ω_n, defined as $\omega_n = \sqrt{a_0}$,

- the *damping ratio* ζ, defined as $\zeta = a_1/(2\omega_n)$.

With these definitions, (1.23) can be written as

$$G(s) = K \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}. \tag{1.25}$$

The static gain has already been introduced, and we now discuss the influence of the natural frequency and the damping ratio on the system dynamics by investigating the step response of (1.25).

Damping Ratio

To study the effect of the damping ratio, we assume for simplicity that the natural frequency is 1; thus we have

$$G(s) = K \frac{1}{s^2 + 2\zeta s + 1}. \tag{1.26}$$

The poles of this transfer function are the solutions of the characteristic equation

$$s^2 + 2\zeta s + 1 = 0,$$

they are

$$s_{1,2} = -\zeta \pm j\sqrt{1 - \zeta^2}.$$

For $\zeta = 0$, we have a complex conjugate pole pair on the imaginary axis at $s_{1,2} = \pm j$, whereas for $\zeta = 1$ both poles are at $s_{1,2} = -1$. For values of ζ between 0 and 1, the poles are located on a circle of radius 1. As ζ is increased from 0 to 1, the poles move from $\pm j$ to -1 as shown in Figure 1.14. For $\zeta > 1$, both poles are real and when ζ is increased from 1 to ∞ one pole moves left towards $-\infty$ and the other one moves right towards 0.
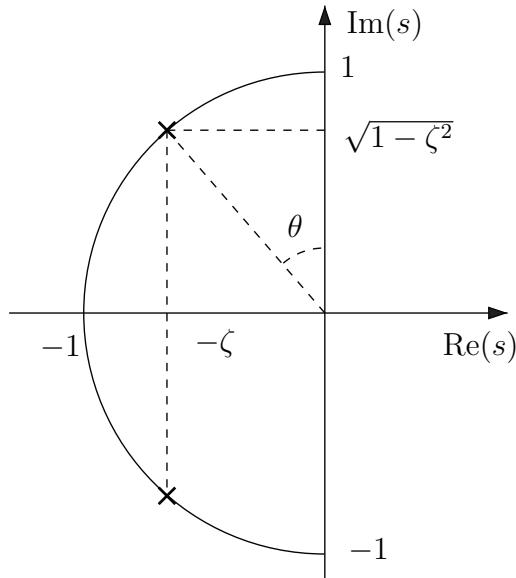
Figure 1.14: Poles of (1.26) when $\omega_n = 1$

Using the results of the previous section, we can infer the dynamic behaviour in time domain from the pole locations.

- When the damping ratio is zero, the transient response is pure oscillation - corresponding to a pole pair on the imaginary axis.

- When the damping ratio is greater than or equal to 1, we have exponential decay, determined by a pair of real poles.

- For damping ratios between zero and one, the transient response contains both oscillation and decay: the rate of decay is determined by the real part $-\zeta$ of the poles, and the frequency of oscillation by the imaginary part $\pm j\sqrt{1 - \zeta^2}$.

Figure 1.15 shows the step response of a plant with transfer function (1.26) for different damping ratios.

The damping ratio $\zeta = 1$ is called the *critical damping ratio* because it represents the boundary between oscillatory response and pure exponential decay. In practice, a damping ratio of 0.7 is often considered acceptable.
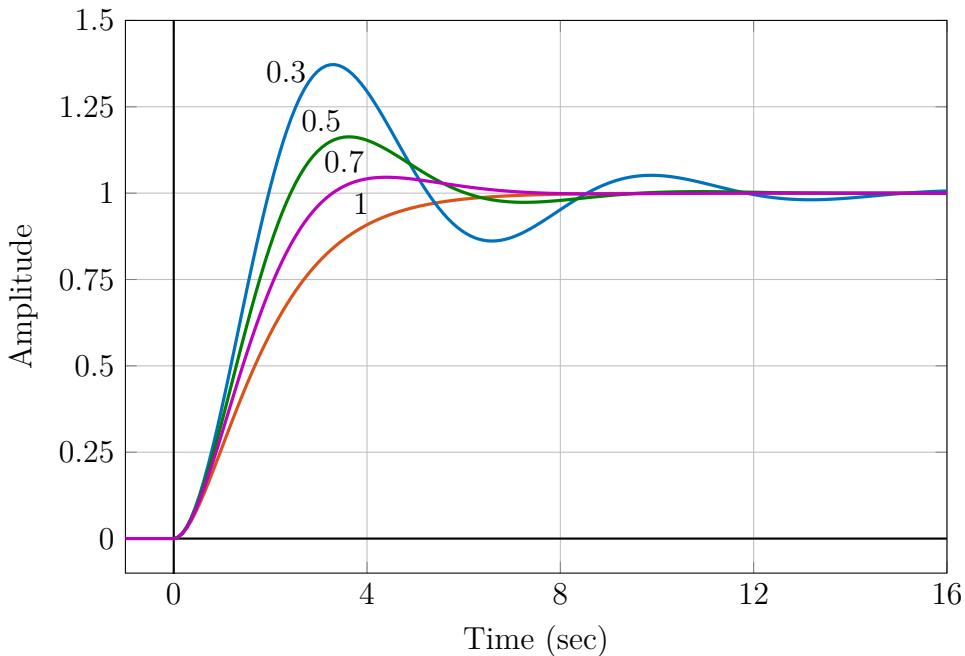
Figure 1.15: Step response of (1.26) for different damping ratios ($\omega_n = 1$)

Natural Frequency

For simplicity, the natural frequency was assumed to be $\omega_n = 1$ so far. The general situation where ω_n takes any positive value can be transformed into this simpler form by frequency scaling: in (1.25), divide numerator and denominator by ω_n^2 to get

$$G(s) = K \frac{1}{\left(\frac{s}{\omega_n}\right)^2 + 2\,\zeta\frac{s}{\omega_n} + 1}.$$

This is the same transfer function as (1.26), except for the fact that the complex variable s is replaced by the normalized variable s/ω_n. Let $s' = s/\omega_n$, then the results on the damping ratio discussed above are valid for any value of ω_n if in (1.26) we replace $G(s)$ by

$$G'(s') = K\frac{1}{s'^2 + 2\zeta s' + 1}$$

To revert from s' to s we need to multiply s' by ω_n. Figure 1.16 shows the resulting pole locations. Because everything is multiplied by ω_n, the poles are now located on a circle of radius ω_n. The real part of the poles is $-\omega_n\zeta$, and the imaginary part is now $\omega_d = \omega_n\sqrt{1 - \zeta^2}$. The quantity ω_d is the frequency of the damped oscillation, it is called the *damped natural frequency*.

In time domain, the natural frequency determines the time scale of the response. Figure 1.15 shows the step responses of a plant with $\omega_n = 1$. For the lightly damped response with $\zeta = 0.3$, the damped natural frequency ω_d is close to the natural frequency $\omega_n = 1$, and from Figure 1.15 it can be seen that the time between the first and second peak
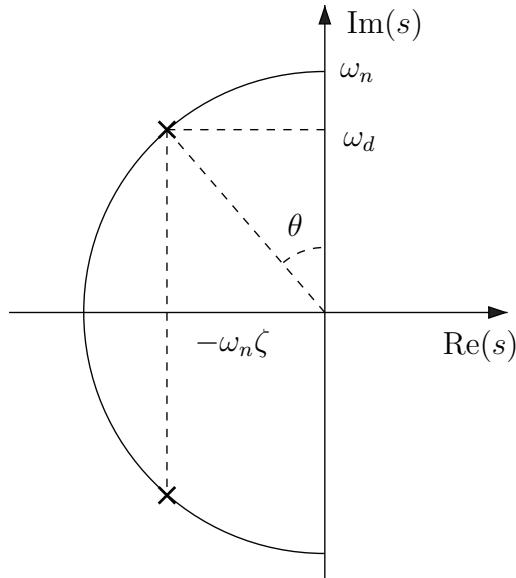
Figure 1.16: Pole location with natural frequency ω_n

is close to 2π. Figure 1.17 shows in comparison the response of a plant with the same damping ratio $\zeta = 0.3$ but $\omega_n = 3$. As is to be expected from the pole locations, decay rate and frequency of oscillation are three times faster.

The damping ratio of a given complex pole pair is clearly related to the angle θ shown in Figure 1.16: we have $\sin\theta = \zeta$. The angle θ is often used to characterize the damping ratio of a second order system.

Time Domain Specifiations

The above discussion has shown that rewriting a second order transfer function in terms of steady state gain, natural frequency and damping ratio provides information about the shape of the response. When designing controllers for second order systems, the requirements are most often expressed in terms of the desired closed-loop step response. Typical requirements are concerned with the speed of the response, the overshoot of the response, and the time it takes for oscillation to die out. Three parameters that are often used to measure these quantities are shown in Figure 1.18, they are

- the *peak overshoot M_p*

- the *rise time t_r*

- the *settling time t_s*.

The rise time is a measure of the initial speed of the response; peak overshoot and settling time are measures of amplitude and decay rate of the oscillation in the transient response.
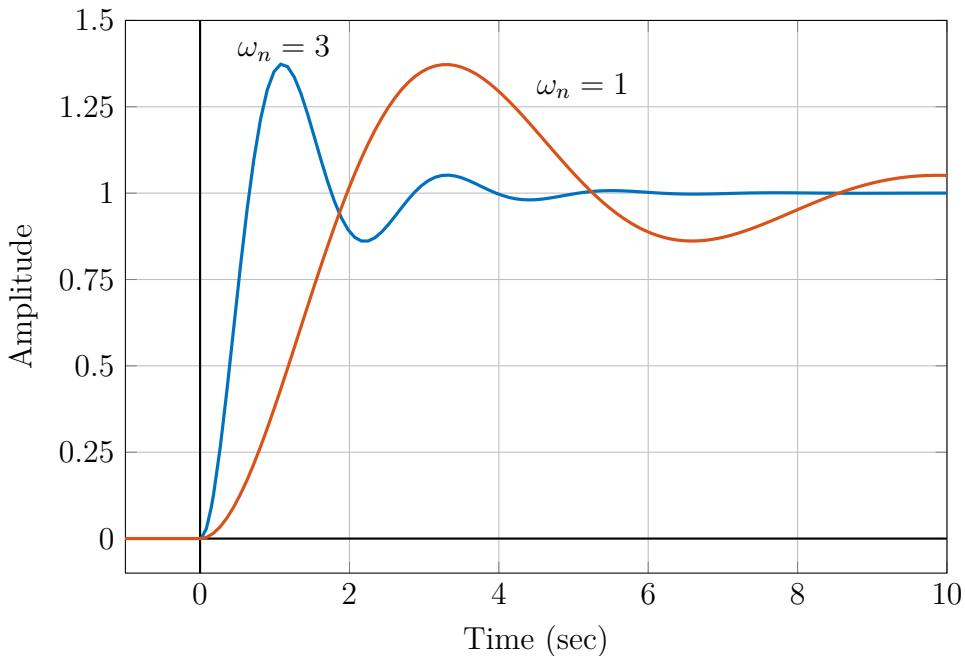
Figure 1.17: Step response with natural frequencies 1.0 and 3.0 (damping ratio $\zeta = 0.3$)

Peak Overshoot

The peak overshoot

$$M_p = \frac{y_{\max} - y_{ss}}{y_{ss}}$$

is a relative measure; for a system with transfer function (1.25) its value can be computed by setting the time derivative of the response to zero (see Exercise 1.15). The result is

$$M_p = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}}. \tag{1.27}$$

Figure 1.19 shows the value of the peak overshoot as a function of the damping ratio. For a rough estimate of the peak overshoot, this formula is sometimes replaced by the linear approximation

$$M_p \approx 1 - \frac{\zeta}{0.6}. \tag{1.28}$$

Rise Time

We define the rise time as the time it takes the output to move from 10% to 90% of its steady state value. Note that other definitions are also used in the literature. A rough estimate of the rise time can be obtained from Figure 1.15. For damping ratios of 0.7 and less, the rise times do not vary significantly. The damping ratio 0.5 corresponds to a rise
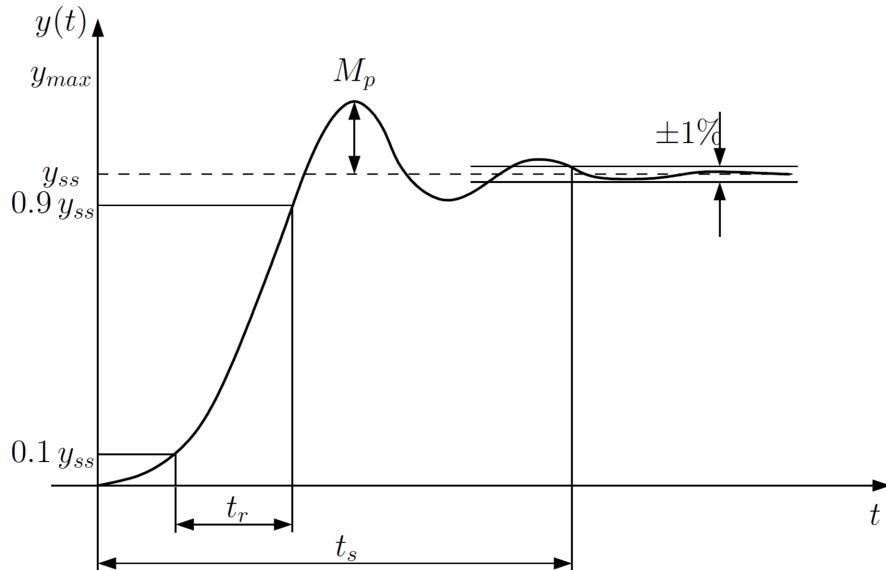
Figure 1.18: Time-domain specifiations for the step response

time of approximately 1.7, and taking that as an average (and recalling that this value is valid only for $\omega_n = 1$) leads to the estimate

$$t_r \approx \frac{1.7}{\omega_n}. \tag{1.29}$$

Settling Time

We define the settling time as the time t_s at which the output oscillation has decayed to a point where the deviation from its steady state value remains less than 1%, i.e.

$$\left| \frac{y(t) - y_{ss}}{y_{ss}} \right| < 0.01, \quad \forall t \geq t_s.$$

Note again that the settling time can be defined for different levels of steady state error; in this course however we will use the 1% settling time throughout. To estimate the settling time of a second order system, we observe that the oscillation in the transient response decays as $e^{-\omega_n \zeta t}$. Allowing a tolerance of 1%, the settling time t_s can be estimated by solving $e^{-\omega_n \zeta t_s} = 0.01$ for t_s, which gives

$$t_s \approx \frac{4.6}{\zeta \omega_n} = \frac{4.6}{\sigma}.$$

Example 1.6

Consider again Example 1.5 of Section 1.2 with

$$G(s) = \frac{10}{s^2 + 2s + 17}.$$
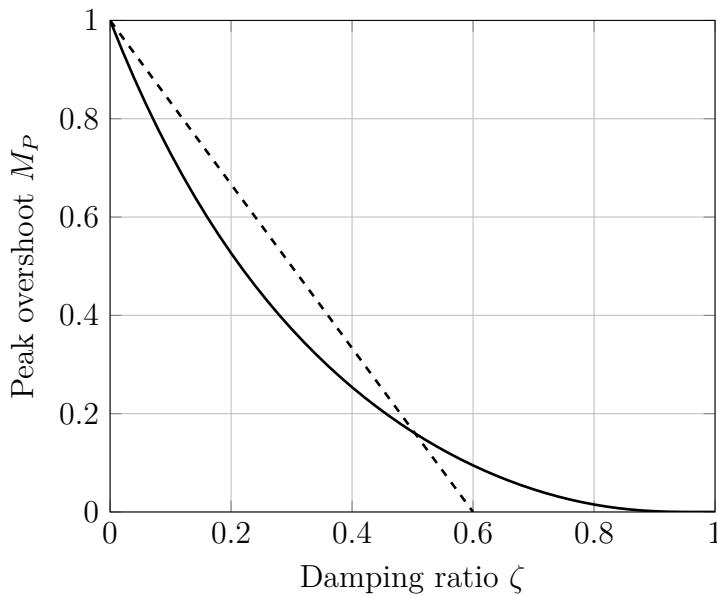
Figure 1.19: Peak overshoot vs. damping ratio and linear approximation

It is straightforward to rewrite this transfer function in the form of (1.25) as

$$G(s) = \frac{1}{1.7} \frac{17}{s^2 + 2 \cdot 0.24 \cdot 4.12 s + 17}$$

thus we have $K = 0.59$, $\omega_n = 4.1$ and $\zeta = 0.24$. Using the above approximations yields the estimates

$$M_p \approx 60\%, \quad t_r \approx 0.41, \quad t_s \approx 4.6,$$

which can be compared against the plot in Figure 1.6.

In this example, an estimate of the step response of the system has been extracted from its transfer function. On the other hand, if the step response of a system (1.25) has been measured, it can be used to estimate the transfer function.

Time Domain Specifications and Pole Regions

The above discussion shows that time domain conditions on a second order system such as a minimum speed of response, a maximum peak overshoot and a maximum settling time can be translated into regions in the complex plane where the poles should be located. Since an upper bound on overshoot implies a lower bound on the damping ratio, the poles must lie within a conic sector as shown in Figure 1.20. Conditions on settling time and rise time translate into the requirement that the poles are to the left of a vertical line and outside a circle, respectively. As we shall see in Chapter 4, poles too far the left are undesirable; for this reason a desirable pole region is usually also bounded to the left and has typically a shape as indicated in Figure 1.20.

More than Two Poles, Dominant Pole Pair
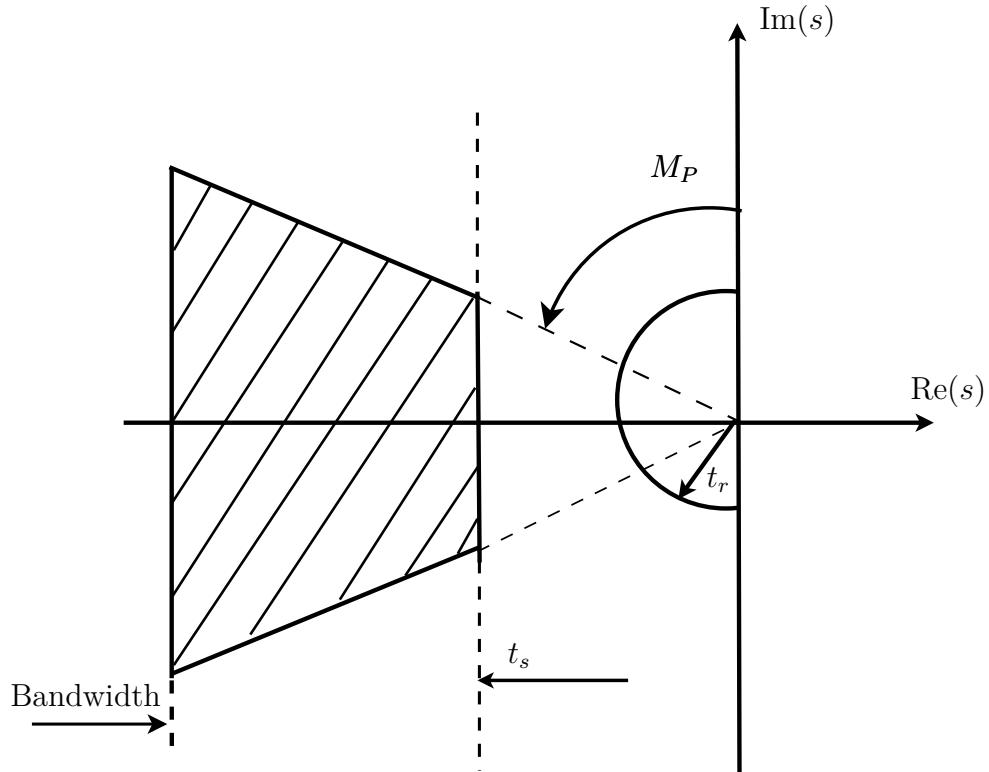
Figure 1.20: Design specifications expresses as desirable pole region (shaded)

The relationship between pole locations and shape of the step response discussed in this section has been derived for second order systems with a transfer function of the form (1.25). A natural question to ask is to what extent the above rules do still apply when a system has more than two poles. In practice, these rules are often applied to higher order systems. This will usually be a valid approximation when a higher order system is dominated by second order dynamics. To illustrate this, consider a third order system with the transfer function

$$G(s) = \frac{K}{(s^2 + 2\zeta s + 1)\left(\frac{s}{\alpha} + 1\right)}. \tag{1.30}$$

This system can be interpreted as a second order system in standard form (1.25) with an additional pole at $s = -\alpha$. The step response of the system (1.30) is shown in Figure 1.21 for $\zeta = 0.3$ and different values of α; the dotted line shows the response of the corresponding second order system without extra pole. It can be seen that additional poles far to the left of the second order pole pair (see Figure 1.22) such as for $\alpha = 5$ and $\alpha = 2$ do not significantly change the behaviour of the system. Only when the extra pole gets close to the second order pole pair, does the response change and displays a significantly reduced overshoot, as can be seen for the response when $\alpha = 0.5$. A system of order greater than two is said to have a *dominant pole pair* if its behaviour is dominated by a pair of complex conjugate poles and the remaining poles are sufficiently far to the

left. In this case - which occurs quite often in practise - all the rules discussed in this section can be applied to the dominant pole pair.
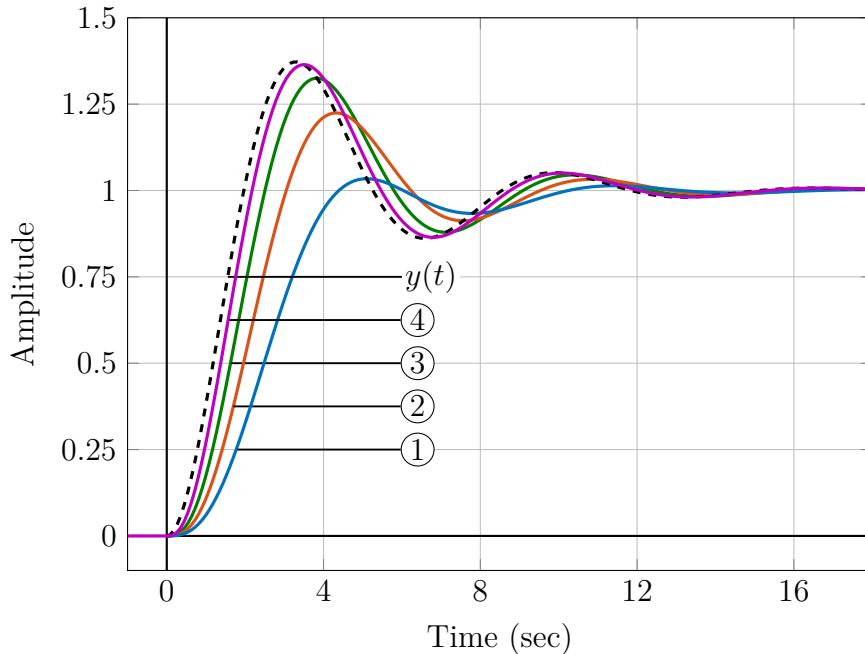
Figure 1.21: Step response of second order system ($y(t)$, dashed) and of third order systems with extra poles as shown in Figure 1.22 (solid (1)-(4))

Transfer Function Zeros

So far we studied rational transfer functions by investigating the denominator polynomial: the roots of the characteristic polynomial - the poles - determine the shape of the components of the transient response. In the examples we considered until now, the numerator of the transfer function was a constant. In general, however, the numerator of a transfer function can also be a polynomial in the variable s, as the following example illustrates.

Example 1.7

The transfer function of the RLC network in Figure 1.23 can be derived as in Example 1.2; it is

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0},$$

where

$$a_0 = b_0 = \frac{1}{LC}, \quad a_1 = b_1 = \frac{R}{L}.$$
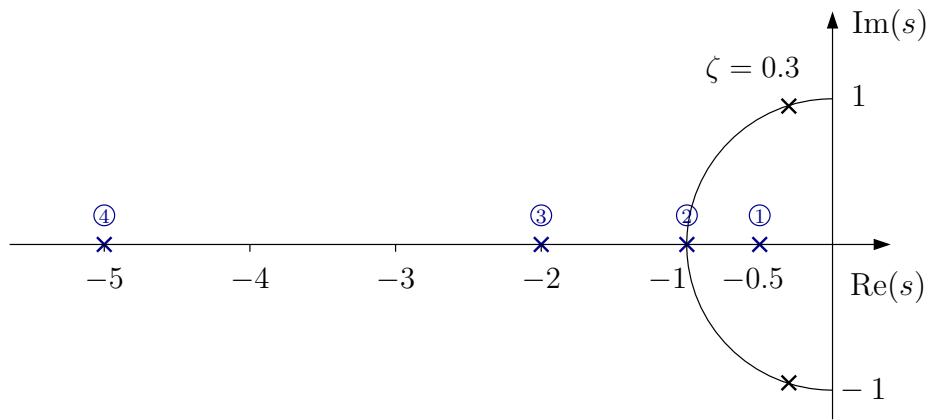
Figure 1.22: Dominant pole pair and additional pole at different locations
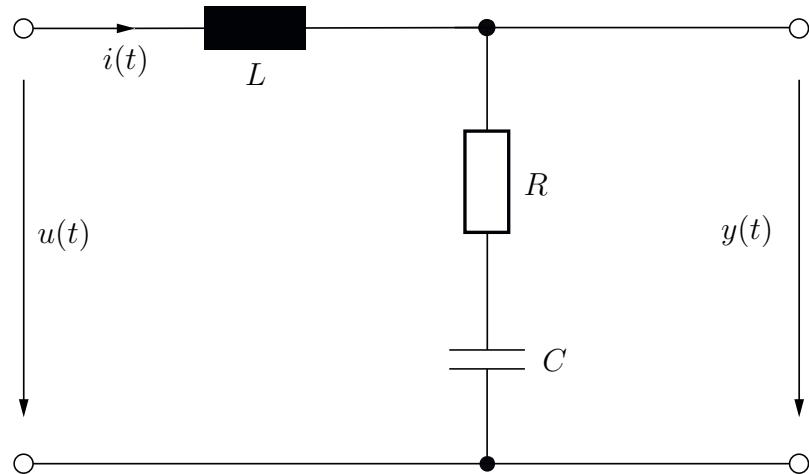
Figure 1.23: Electrical network with transfer function zero

The *poles* of the transfer function - the values of s at which the denominator is zero - determine the components of the transient response. We will now investigate the effect a *zero* of the transfer function - a value of s at which the numerator is zero - has on the shape of the response. The above transfer function has a zero at $s = -b_0/b_1$.

The effect of a zero on the step response is now illustrated with a numerical example.

Example 1.8

Consider again the system of Example 1.4 with transfer function

$$G_1(s) = \frac{6}{(s+2)(s+3)}.$$

The step response was computed in the previous section as

$$y_1(t) = \left(1 - 3e^{-2t} + 2e^{-3t}\right)\sigma(t). \tag{1.31}$$

A plot of the response is shown in Figure 1.4. With an additional zero at $s = -1$, the transfer function becomes

$$G_2(s) = \frac{6(s+1)}{(s+2)(s+3)},$$

and the step response can be computed as

$$y_2(t) = \left(1 + 3e^{-2t} - 4e^{-3t}\right)\sigma(t). \tag{1.32}$$

Both responses are shown in Figure 1.24; it can be seen that the presence of the zero leads to a significantly different shape.
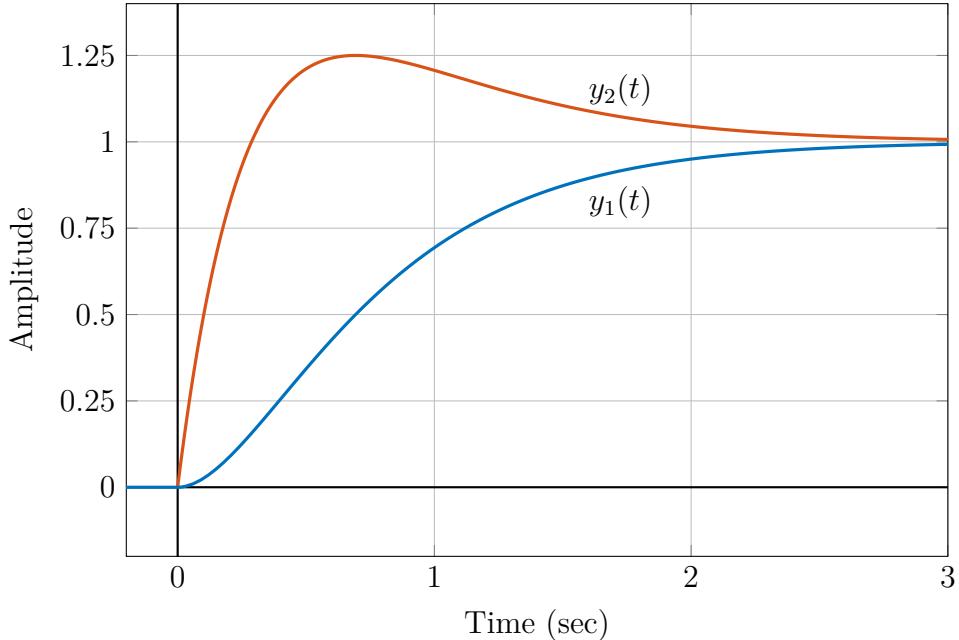
Figure 1.24: Step responses $y_1(t)$ of $G_1(s)$ and $y_2(t)$ of $G_2(s)$

This example illustrates the different ways in which poles and zeros of a transfer function influence the shape of the response. The poles determine the characteristic features of the transient response: whether it is fast or slow, whether or not there is oscillation in the response, and whether or not the system is stable. Each pole represents one of
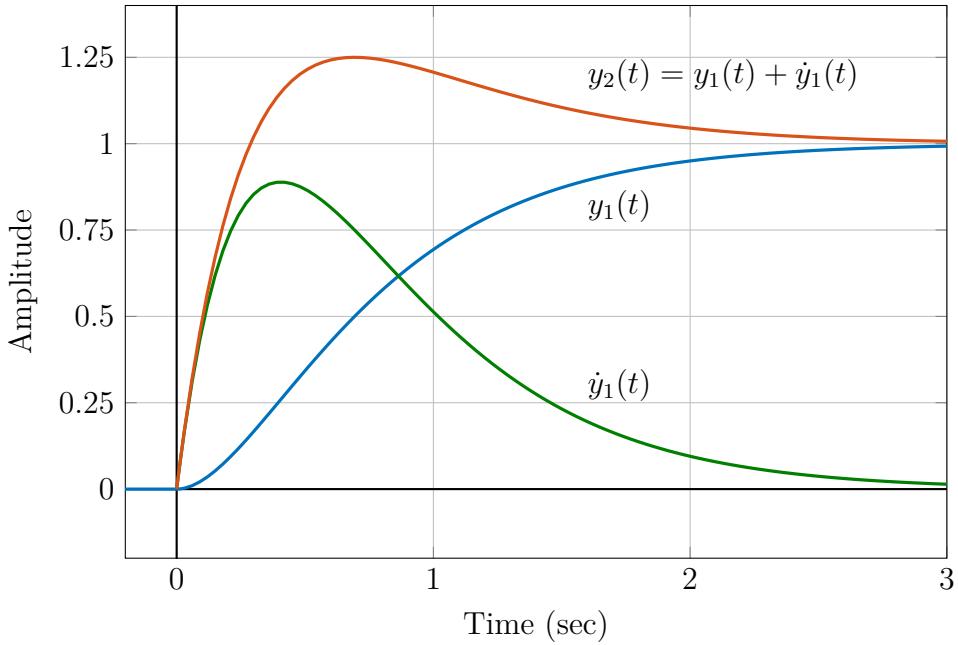
Figure 1.25: Step response with left half plane zero ($\alpha = 1$)

the components from which the transient response is formed. On the other hand, the zeros have an effect on the relative weight of each component, on how much each pole contributes to the response.

This can be seen when comparing the step response (1.31) with (1.32) in Example 1.8. In this example, the main effect of adding a zero to the transfer function $G_1(s)$ is that the dominant (the slower) component e^{-2t} becomes positive and the faster component e^{-3t} negative; this results in overshoot in the step response. Adding a zero in the left half plane has in general a tendency to increase the overshoot. To see this, consider the two transfer functions

$$G_1(s) = \frac{b_0}{s^2 + a_1 s + a_0}, \qquad G_2(s) = \frac{b_0(\frac{s}{\alpha} + 1)}{s^2 + a_1 s + a_0}.$$

The transfer function $G_2(s)$ has the same poles and the same static gain as $G_1(s)$, but it has an additional zero at $s = -\alpha$. This zero is in the left half plane if $\alpha > 0$. $G_2(s)$ can also be written as

$$G_2(s) = \frac{b_0}{s^2 + a_1 s + a_0} + \frac{1}{\alpha} s \frac{b_0}{s^2 + a_1 s + a_0} = G_1(s) + \frac{1}{\alpha} s G_1(s).$$

The Laplace transforms of the step responses are

$$Y_1(s) = G_1(s)\frac{1}{s} \quad \text{and} \quad Y_2(s) = G_1(s)\frac{1}{s} + \frac{1}{\alpha} s G_1(s)\frac{1}{s},$$

respectively, or

$$Y_2(s) = Y_1(s) + \frac{1}{\alpha} s Y_1(s).$$

In time domain, the step response of the plant with an additional zero is therefore

$$y_2(t) = y_1(t) + \frac{1}{\alpha}\dot{y}_1(t).$$

Thus, the effect of adding a left half plane zero on the response is that its derivative - scaled by $1/\alpha$ - is added to the response. Figure 1.25 shows both responses together with the derivative of y_1.

If an additional *right* half plane zero is included in the dynamics, i.e. if $\alpha < 0$, the derivative of the step response without zero is subtracted instead of added. This causes undershoot in the step response, as illustrated in Figure 1.26, where the step response $y_3(t)$ of a transfer function $G_3(s)$ with a zero at $s = 1$ ($\alpha = -1$) is shown.
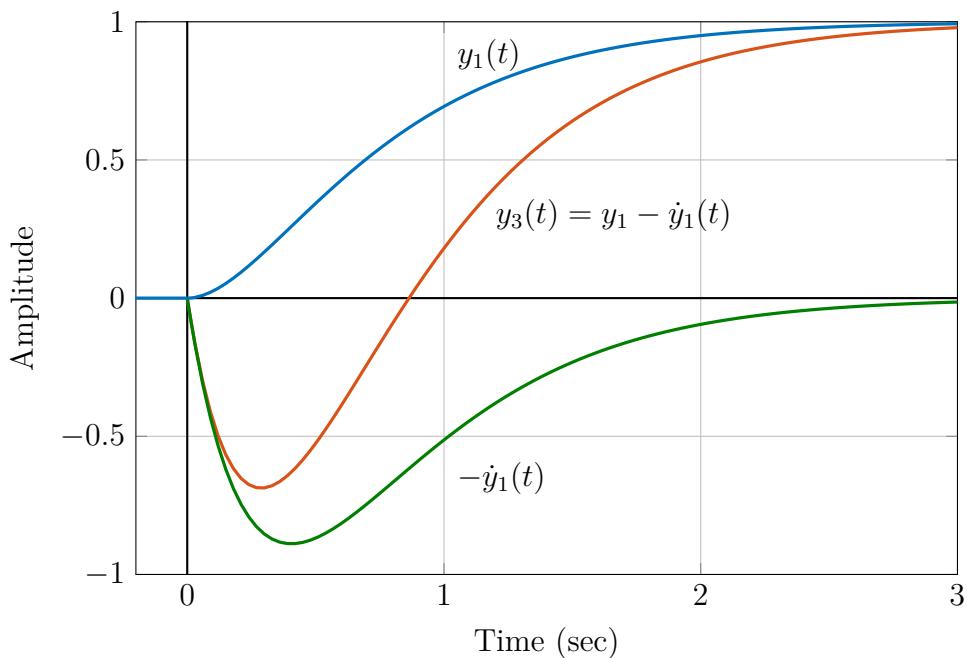
Figure 1.26: Step response with right half plane zero ($\alpha = -1$)

1.4   Modelling a DC Motor

In this section we derive a simplified model of a DC motor, that will be used frequently in this course to illustrate concepts and ideas.

The electric circuit of the armature of a DC motor is shown in Figure 1.27. To keep the model simple, we will first neglect the inductance in the armature circuit as well as the friction; this simplification and a more accurate model will be investigated in Exercises 1.5 and 1.6. We will however consider the effect of an external load torque on the motor speed $\Omega = \dot{\theta}$.
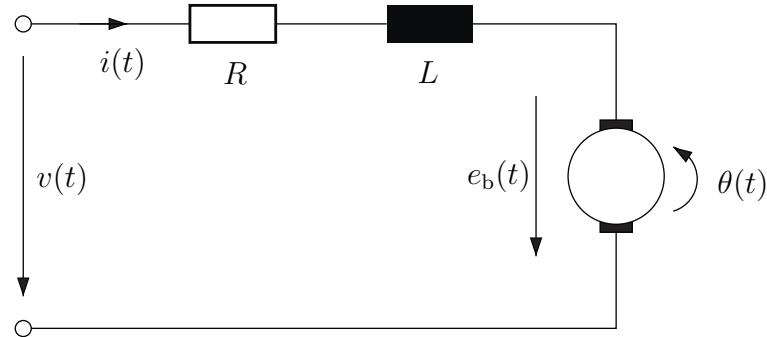
Figure 1.27: DC motor

To control the motor speed, the armature voltage v is used. Under the simplifying assumptions given above, the motor can be described by the equations

$$J\dot{\Omega} = T_m + T_l$$
$$v = Ri + e_b,$$

where in the first equation J is the motor shaft inertia, T_m the motor torque generated in the rotor, and T_l an external load torque. In the second equation R is the armature resistance, i the armature current and e_b the back emf. Assuming that the field is constant, we have

$$T_m = K_m i \quad \text{and} \quad e_b = K_g \Omega$$

where K_m is the motor constant and K_g the generator constant.

Substituting for the back emf and solving for i yields

$$i = \frac{1}{R}(v - K_g \Omega).$$

When this expression is used in the torque equation, we obtain

$$J\dot{\Omega} + \frac{K_m K_g}{R}\Omega = \frac{K_m}{R}v + T_l,$$

or after dividing by the coefficient of Ω

$$\frac{JR}{K_m K_g}\dot{\Omega} + \Omega = \frac{1}{K_g}v + \frac{R}{K_m K_g}T_l.$$

Introducing new constants

$$\tau = \frac{JR}{K_m K_g}, \quad K_o = \frac{1}{K_g}, \quad K_l = \frac{R}{K_m} \tag{1.33}$$

and taking the Laplace transform, we have

$$\tau s \Omega(s) + \Omega(s) = K_0(V(s) + K_l T_l(s))$$
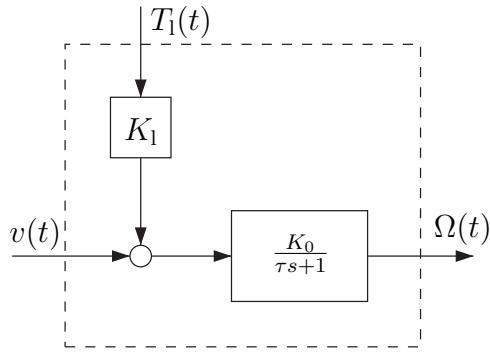
Figure 1.28: Model of a DC motor

or

$$\Omega(s) = \frac{K_o}{\tau s + 1}\Big(V(s) + K_l T_l(s)\Big). \tag{1.34}$$

A block diagram of this model is shown in Figure 1.28. The plant has two inputs and one output. The input voltage v is used to control the speed, it is called the *control input*. The load torque T_l cannot be influenced and thus cannot be utilized for control; this input is called a *disturbance input*.

Exercises for Chapter 1

Exercises

Problem 1.1

Compute the Laplace transform of the unit ramp function

$$x(t) = t\sigma(t).$$

Hint: Use integration by parts.

Problem 1.2

Prove the convolution theorem

$$\mathcal{L}[x_1(t) * x_2(t)] = \mathcal{L}[x_1(t)] \cdot \mathcal{L}[x_2(t)].$$

Problem 1.3

Consider a dynamic system governed by the first order differential equation

$$\dot{y}(t) + a_0 y(t) = b_0 u(t)$$

a) Write down the transfer function and compute the response $y(t)$ of the system to a unit step $u(t) = \sigma(t)$.

b) Sketch the step response.

c) Compute the slope $\dot{y}(0)$ of the step response at $t = 0$.

d) The time constant τ of a first order system is defined as the time at which a tangent fitted at $y(0)$ reaches the steady state value of the step response. Compute τ for the above system.

e) For $a_0 = 1$ and $b_0 = 2$, compare your sketch with the step response generated by MATLAB.

MATLAB file: `Problem1_3_1stOrderStepResponse.mlx`

Problem 1.4

Consider a system governed by the second order differential equation

$$\ddot{y}(t) + 3\dot{y}(t) + 2y(t) = 2u(t)$$

a)   Write down the transfer function and compute the step response.

b)   Sketch the step response.

c)   Compare your sketch with the step response generated by MATLAB.
Try to find out what happens to the shape of the response when you vary the values
of the coefficients of the differential equation.

MATLAB file: `Problem1_4_2ndOrderStepResponse.mlx`

Problem 1.5
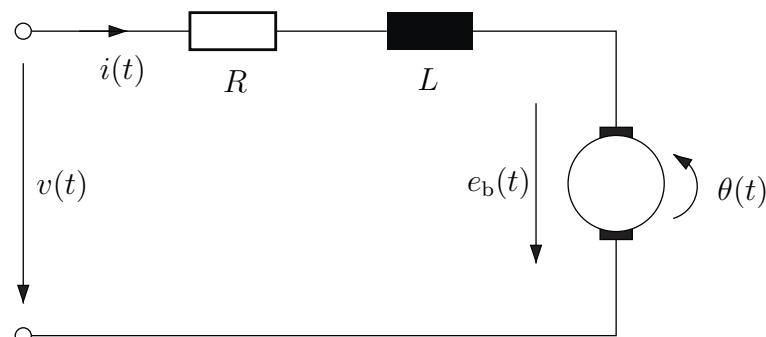
Figure 1.29: DC motor

Table 1.3: List of variables

Variable	Name	Variable	Name
v	armature voltage	T_{m}	motor torque
i	armature current	θ	shaft angle
R	armature resistance	$\Omega = \dot{\theta}$	rotor speed
L	armature inductance	b_{v}	viscious friction coefficient
e_{b}	back emf voltage	J	inertia of the motor
K_{m}	motor constant	K_{g}	generator constant
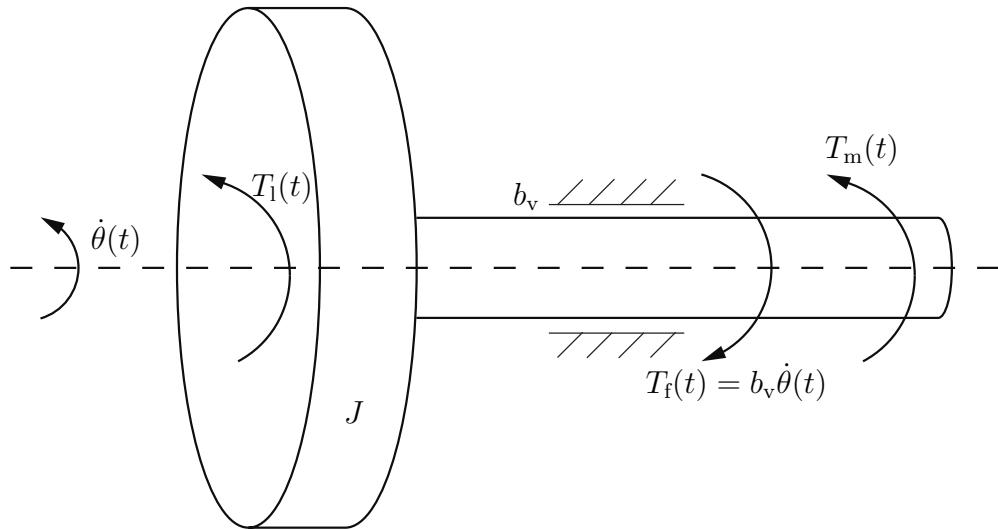
Figure 1.30: Mechanical subsystem of the DC motor

Figure 1.29 shows the electric circuit of the armature of a DC motor.
A simplified first order model has been derived in Section 1.4. In this problem the model
(1.34) will be extended to include the armature inductance as well as viscous friction.
Assume for now that there is no load torque, i.e. $T_l = 0$.

In order to form the equation of motion, you have to use the principle of angular momentum, from which you obtain:

$$J\ddot{\theta} + b_v\dot{\theta} = T_m = K_m i.$$

The electrical model can be described as follows:

$$L\frac{di}{dt} + Ri = v - e_b = v - K_g\dot{\theta}$$

a)   From the given equations, derive a state space model:

$$\dot{x}(t) = Ax(t) + bu(t), \quad y(t) = cx(t) + du(t)$$

Choose as input the armature voltage ($u = v$), as state vector $x = [\Omega \ \ i]^T$, and as
output the angular velocity ($y = \Omega$).

b)   Compute the transfer function corresponding to the state space model in (a).

c)   We now define the plant output to be the shaft angle θ instead of the angular
velocity $\Omega = \dot{\theta}$. Modify the transfer function and the state space model accordingly.

Problem 1.6

a)   Verify that the model obtained in Problem (1.5.b) reduces to the simplified model
(1.34) when $b_v = 0$ and $L = 0$. Assume $T_l = 0$.

b) Assume again that b_v is neglegible and therefore the electric power equals the mechanical power: $P_e = e_b \cdot i = P_m = T \cdot \dot{\theta}$. Which relationship between the motor and generator constants K_m and K_g does this imply?

Problem 1.7

Introduction to Motor Experiments:

1) Download the MATLAB/SIMULINK files from *Stud.IP*, unpack the files and save the folder in *Documents → MATLAB*

2) Start MATLAB

3) The new folder with the MATLAB files has to be in your *current folder*. Right-click the folder within MATLAB and choose the option 'Add to Path → Selected Folders and Subfolders'

4) Open `"Model_Problem1_7_OpenLoopStepInput.slx"`

5) Change the COM Port number (Look for this in the document *DC Motor User's Guide* in *Stud.IP*)

6) Connect the motor with your PC as well as with a power source 1.7 leave it all with out motor symbol

7) Start simulation

In the following task all unknown motor parameters - the motor constant K_m, the electrical resistance R and the moment of inertia J - will be identified experimentally. This will determine the numerical values of the coefficients in the simplified transfer function model (1.34).

To obtain the motor constant, one can exploit the relationships used already in Problem 1.5.a). By solving for Ω, neglecting the inductance L of the motor and assuming that $K = K_m = K_g$, one obtains

$$\Omega = v \cdot \frac{1}{K} - \frac{R \cdot i}{K}. \tag{1.35}$$

The template `"Model_Problem1_7_OpenLoopStepInput.slx"` can be used to perform experiments.

a) Consider equation (1.35) in steady state, and assume that the current is constant. With this assumption, Ω in (1.35) depends affinely on v. Apply different input voltage steps (from 0 to 6V) and measure the corresponding steady state velocity Ω. Fit a line to your data points to determine $K = K_m$. Note the hint at the end of the excercise.

b) 🖿 What is the unit of the static gain of the simplified transfer function model? How can one determine the static gain experimentally? Measure the static gain.

c) Calculate the motor constant and compare it with the results obtained in Problem 1.7.a.

d) 🖿 Now the value of the electrical resistance R is to be be determined from equation (1.35). In which case is it possible to obtain R easily by plotting the armature current i against the armature voltage v? At which angular velocity can such a measurement be carried out conveniently (e.g. by manually intervening in the experiment)? Note the hint at the end of the excercise.

Determine R experimentally.

e) 🖿 To identify the moment of inertia J, consider the first order dynamics

$$G(s) = \frac{\Omega(s)}{U(s)} = \frac{K_0}{\tau s + 1}.$$

of the motor. Determine the time constant τ by experiments with step inputs.

f) Determine the numerical value of J from equation (1.33) using the values of the electrical resistance R and the motor constant K_m as well as the time constant τ determined in the previous subtasks.

Hints:

- *You can touch the disc or even stop it.*

- *In order to show a regression line use the basic fitting tool. Open the figure window → click 'tool' → 'basic fitting' and choose the figure which fits best. You can also display the equation for the regression line.*

SIMULINK files:

- `Model_Problem1_7_OpenLoopRampInput.slx`
- `Model_Problem1_7_OpenLoopStepInput.slx`

Problem 1.8   🖿

Validate the motor model obtained in Problem 1.7: Plot the simulated step response of the model together with the experimental step response of the DC motor. Both plots are shown in one figure. Compare the results.

SIMULINK file: `Model_Problem1_8_ParameterTest.slx`

Problem 1.9

a)   Compute a state-space model of the system given by the differential equation

$$\ddot{y}(t) + 3\dot{y}(t) + 2y(t) = 2u(t)$$

Define state variables $x_1 = y$, $x_2 = \dot{y}$

b)   Assuming zero initial values, simulate the step response of the state-space model.

MATLAB file: `Problem1_9_DifferentialEquationWithoutInitialConditions.mlx`

Problem 1.10

Compute the output $y(t)$ of the system in Problem 1.9 with the initial values:

$$y(0) = 3, \quad \dot{y}(0) = 0, \quad \text{and} \quad u(t) = 2\sigma(t)$$

Sketch the solution and compare it with the step response in Problem 1.4.

MATLAB file: `Problem1_10_DifferentialEquationWithInitialConditions.mlx`

Problem 1.11

Consider a system with transfer function

$$G(s) = \frac{b_0}{s^2 + a_1 s + a_0} = \frac{b_0}{(s - p_1)(s - p_2)}$$

and two real poles $p_1, p_2 < 0$.

a)   Compute the steady state and the transient component of the step response.

b)   Using your result in (a), show that $y(0^+) = 0$, i.e. the output signal does not "jump" at $t = 0$.

c)   Use the initial value theorem to show the same result.

d)   What can you say about $\dot{y}(0+)$?

e)   What is $\dot{y}(0^+)$ for a system with an extra zero

$$H(s) = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0}?$$

Problem 1.12

Consider the circuit in Figure 1.31.
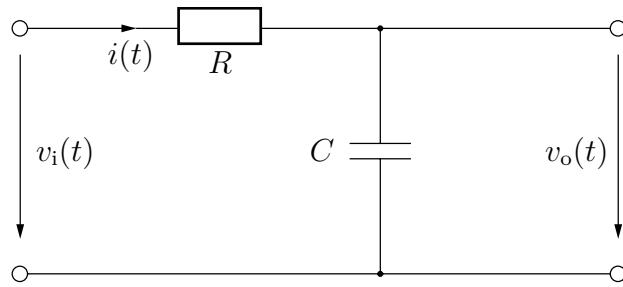
Figure 1.31: Electrical circuit

The input voltage applied to this circuit is shown in Figure 1.32

a) Compute the response $v_o(t)$ of the circuit when the input voltage $v_i(t)$ is applied.

b) Show that the response to the rectangular pulse in Figure 1.32 approaches the impulse response when $\epsilon \to 0$.
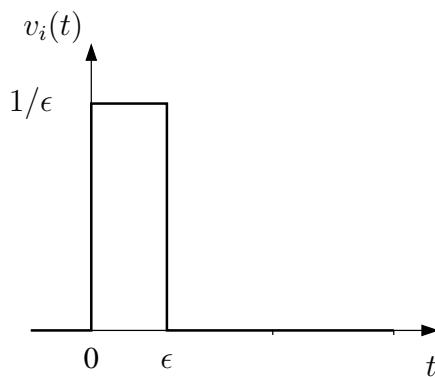
Figure 1.32: Input voltage for circuit

MATLAB file: `Problem1_12_RCModel.mlx`

Problem 1.13 *

In this exercise we use the model of a DC motor derived in Problem 1.5.

a) Using the numerical values $J = 0.01$, $b = 0.001$, $K_m = K_g = 1$, $R = 10$, $L = 1$, and taking the motor speed $\dot\theta$ as output, simulate the response to rectangular inputs $u(t) = r_\epsilon(t)$ for $\epsilon = 0.2$, $\epsilon = 0.1$ and $\epsilon = 0.05$. Compare these responses with the impulse response of the system.

b)   Using the approximated impulse response found in   a) (for $\epsilon = 0.2$, 0.1, 0.05) find the response of the system for the armature voltage as shown in Figure 1.33. (Hint: Use convolution integral (1.22)).

c)   Simulate the exact response to the armature voltage shown in Figure 1.33.
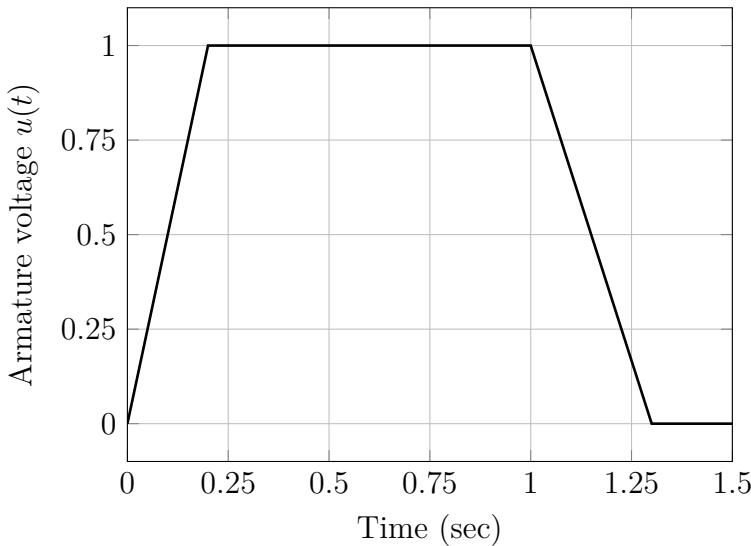
Figure 1.33: Armature voltage

MATLAB file: `Problem1_13_DiscreteConvolution.mlx`

Problem 1.14

Consider the mass-spring-damper system in Figure 1.1. The equation of motion is given in Equation (1.1).

a)   Recalling the standard form of a second order system

$$G(s) = K \frac{\omega_n^2}{s^2 + 2\omega_n \zeta s + \omega_n^2}$$

express the natural frequency ω_n, the static gain K and the damping ratio ζ in terms of the system parameters k, m and b.

b)   For the numerical values $m = 1$, $k = 50$ and $b = 4$, compute ω_n, K and ζ. Simulate the step response with MATLAB. Vary the values of the physical parameters and check by simulation how these changes affect gain, damping and natural frequency, and how they influence the shape of the step response.

c)   Repeat this exercise for the RLC circuit in Figure 1.2, and compare the results.

MATLAB file: `Problem1_14_MassSpringDamperSystem.mlx`

Problem 1.15

Consider a second order system with transfer function

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad \text{where } \zeta < 1.$$

a) Compute the response $h(t)$ of the system to a unit step.

   Hint: Rewrite the Laplace transform $H(s)$ of the step response as

$$H(s) = \frac{1}{s} - \frac{s + 2\zeta\omega_n}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

$$= \frac{1}{s} - \frac{s + \zeta\omega_n}{(s + \zeta\omega_n)^2 + \omega_d^2} - \frac{\zeta\omega_n}{(s + \zeta\omega_n)^2 + \omega_d^2}$$

   where ω_d denotes the damped natural frequency

$$\omega_d = \omega_n\sqrt{1 - \zeta^2}$$

   and use the table of Laplace transforms.

b*) Derive the Formula (1.27) for peak overshoot

$$M_p = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}}.$$

   from $\dot{h}(t) = 0$. For this purpose, show first that the time t_p at which the response reaches its peak value is given by

$$t_p = \frac{\pi}{\omega_d}$$

Problem 1.16

The deviation $y(t)$ of the spring-mass-damper system shown in Figure 1.34 after a step change in the applied force $u(t) = 5\sigma(t)$ at time $t = 0$ from 0 to 5 is plotted in Figure 1.35.

Use the step response to estimate the values of k, m and b. Validate your estimate by simulation.

MATLAB file: `Problem1_16_MassSpringDamperSystemIdentification.mlx`
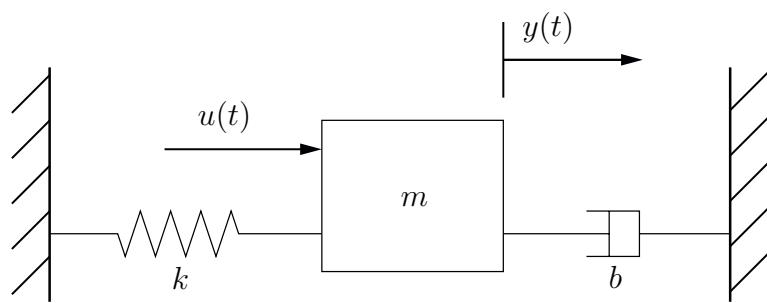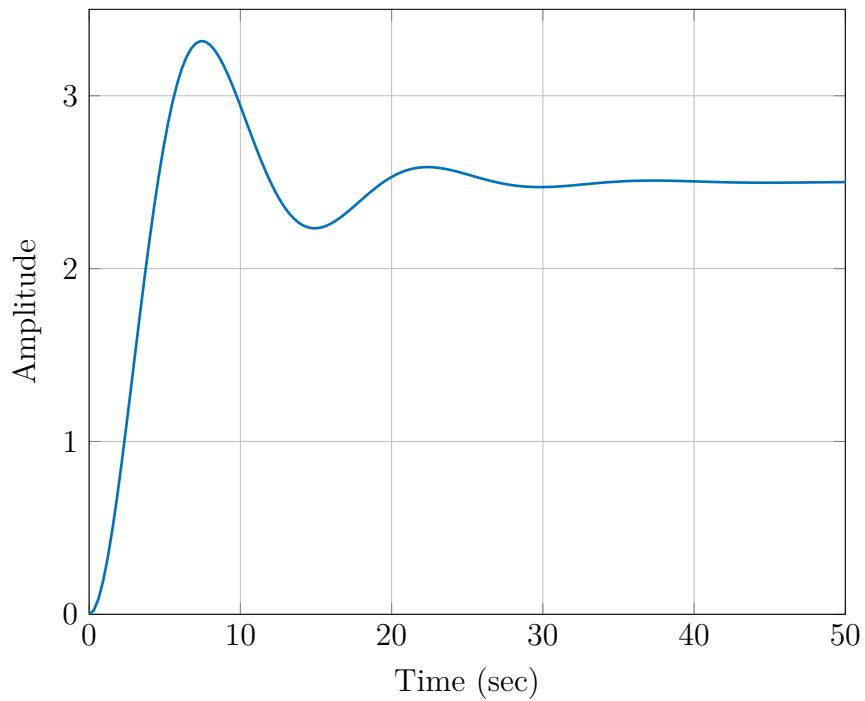
Figure 1.34: Mechanical system

Figure 1.35: Response to $u(t) = 5\sigma(t)$

Chapter 2

Feedback Control

This chapter will introduce the idea of feedback and discuss the advantage of feedback control over open-loop control. After discussing the basic types of feedback - proportional, integral and derivative - we will study under what conditions a control loop can track a given reference input with zero steady state tracking error. The last section presents a way of checking whether a system is stable without computing the poles, which can also be used to determine the range of gain values for which a control loop is stable.

2.1   Feedback Control and Sensitivity

We begin by illustrating the difference between open-loop control and closed-loop control with a simple design example. The plant to be controlled is an armature controlled DC motor. A model of a DC motor was derived in Section 1.4 in the previous chapter.

We will now compare an open-loop and a closed-loop approach to speed control. Consider again the simplified motor model (1.34 and the block diagram in Figure 1.27. Let Ω_d be the desired motor speed. In this section, we consider only steady state behaviour of the control system and ignore transient effects.

Open-Loop Control

An open-loop configuration is shown in Figure 2.1. The controller is connected in series with the plant, and the controller input is the desired speed Ω_r. We consider the simplest possible type of controller: an adjustable gain K_c. The steady state gain of the plant (obtained by setting $s = 0$) is K_0, therefore in steady state - assuming there is no load torque ($T_l = 0$) - the motor speed is

$$\Omega = K_0 v.$$

Setting the controller gain to

$$K_c = \frac{1}{K_0}$$

leads to the desired result $\Omega = \Omega_r$.
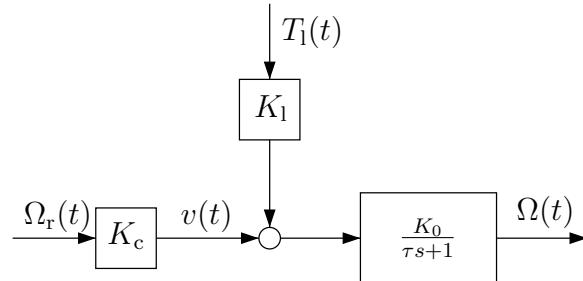
Figure 2.1: Open-loop control

Closed-Loop Control

The closed-loop configuration is shown in Figure 2.2; here the controller input is not the desired speed, but the difference between desired and actual speed. Again assuming $T_l = 0$, the closed-loop transfer function is

$$\frac{\Omega(s)}{\Omega_r(s)} = \frac{K_c K_0}{\tau s + 1 + K_c K_0},$$

and letting G_0 denote the steady state closed-loop gain, we have

$$G_0 = \frac{K_c K_0}{1 + K_c K_0}.$$

In steady state operation, the motor speed is

$$\Omega = \frac{K_c K_0}{1 + K_c K_0}\Omega_r.$$

Therefore, in contrast to open-loop control, the closed-loop controller cannot set the actual speed exactly to its desired value. However, when the gain product $K_c K_0$ is much larger than 1, the steady state error $\Omega - \Omega_r$ will be small ($K_c K_0 = 100$ for example leads to a 1% error).

Uncertain Plant Parameters, Sensitivity

At this point it seems that open-loop control is superior to closed-loop control, because it can set the speed exactly. An important point to observe however is that this is only possible when the steady state gain of the plant K_0 is known exactly. In practical applications, plant data are only known up to a given limit of accuracy. Moreover, plant parameters often depend on operating conditions and may vary considerably. To study
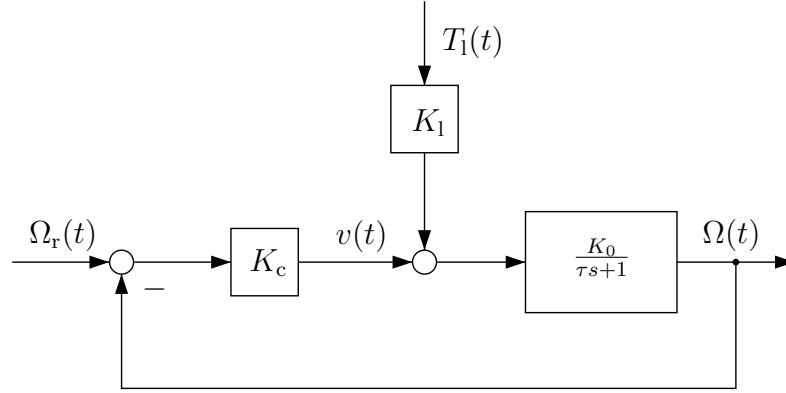
Figure 2.2: Closed-loop control

the effect of uncertainty and variation of plant parameters on the controlled output, we now consider the situation where the designer assumes the steady state plant gain to be K_0, whereas the actual gain is $K_0 + \Delta K_0$.

In the open-loop configuration, the motor speed under this assumption is

$$\Omega = (K_0 + \Delta K_0)\frac{1}{K_0}\Omega_r.$$

Introducing the speed error

$$\Delta\Omega = \frac{\Delta K_0}{K_0}\Omega_r,$$

we find

$$\frac{\Delta\Omega}{\Omega_r} = \frac{\Delta K_0}{K_0}.$$

Thus, an error of say 10% in the plant gain would lead to a 10% error in motor speed.

In the closed-loop configuration, the closed loop steady state gain would change to

$$G_0 + \Delta G_0 = \frac{K_c(K_0 + \Delta K_0)}{1 + K_c(K_0 + \Delta K_0)}.$$

To compare the relative error with the open-loop configuration, we use a linear approximation: if ΔK_0 is small,

$$\Delta G_0 \approx \frac{dG_0}{dK_0}\Delta K_0$$

is a reasonable estimate of the change in steady state gain caused by the error in the plant gain. It follows that the relative speed error is

$$\frac{\Delta\Omega}{\Omega_r} = \frac{\Delta G_0}{G_0} \approx \left(\frac{K_0}{G_0}\frac{dG_0}{dK_0}\right)\frac{\Delta K_0}{K_0}.$$

The factor between the relative error in plant gain and the resulting relative error in closed-loop steady state gain is called the *sensitivity* of the control system, denoted by S. We have

$$S = \frac{K_0}{G_0}\frac{dG_0}{dK_0},$$

and calculating the derivative yields

$$S = \frac{K_0}{\frac{K_c K_0}{1 + K_c K_0}} \frac{K_c(1 + K_c K_0) - K_c K_0 K_c}{(1 + K_0 K_c)^2}$$

or

$$S = \frac{1}{1 + K_c K_0}.$$

When the gain product $K_c K_0$ is large, the sensitivity S is small and the effect of an error in the plant gain on the controlled output is reduced considerably; e.g. with $K_c K_0 = 100$ a 10% error in plant gain leads to a 0.1% error in motor speed.

Disturbance Rejection

Now we consider the effect of a load torque on the motor speed. Ideally the controller should maintain the desired speed independent of the load. In an open-loop control scheme, when the load Torque T_l is nonzero, the motor speed is

$$\Omega = K_0(\frac{1}{K_0}\Omega_r + K_l T_l) = \Omega_r + K_0 K_l T_l.$$

Thus, the open-loop speed error due to the load torque is

$$\Delta\Omega_{ol} = K_0 K_l T_l.$$

In the closed-loop configuration of Figure 2.2, the transfer function from T_l to Ω is

$$\frac{\Omega(s)}{T_l(s)} = \frac{K_l K_0}{\tau s + 1 + K_c K_0},$$

and in steady state operation we have

$$\Omega = \frac{K_c K_0}{1 + K_c K_0}\Omega_r + \frac{K_l K_0}{1 + K_c K_0}T_l.$$

The first term on the right hand side is the motor speed that would be reached when the load torque is zero. The second term is the closed-loop speed error due the load torque

$$\Delta\Omega_{cl} = \frac{K_l K_0}{1 + K_c K_0}T_l.$$

Comparing this with the open-loop result, we find

$$\Delta\Omega_{cl} = \frac{1}{1 + K_c K_0}\Delta\Omega_{ol}$$

or

$$\Delta\Omega_{cl} = S\Delta\Omega_{ol}.$$

Thus, in closed loop the effect of a disturbance load on the speed is reduced by the same factor S as the effect of plant parameter errors; if the gain product $K_c K_0$ is 100, the closed-loop error is reduced to 1% of the open-loop error.

Open Loop vs. Closed Loop

We have compared the performance of open-loop and closed-loop control with respect to two different control objectives:

a) the controlled output should follow a reference input as closely as possible (this is called the *tracking problem*)

b) the controlled output should be held at its desired value in spite of external disturbances (this is the *disturbance rejection problem*).

Figure 2.3 shows the two configurations. The transfer functions from reference input and disturbance input, respectively, to the controlled output are shown in the Table 2.1.
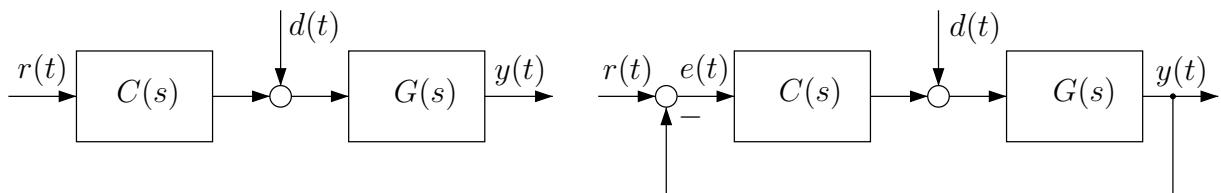
Figure 2.3: Open-loop and closed-loop control

In a tracking problem, a large steady state gain of the controller $C(s)$ in a closed-loop configuration will bring the controlled output close to the reference input; the error will be almost independent of changes in the plant transfer function G. In open loop, a relative change in the plant gain will cause the same relative change in the controlled output.

In open loop, the controller cannot reduce the effect of an external disturbance on the output, whereas in closed loop, a large controller gain leads to a significant reduction of disturbance effects.

So far, only steady state behaviour was discussed. The transient behaviour of a closed-loop system is studied in the next section. But from the transfer functions in Table 2.1 we observe the following. In open-loop control, the transient part of the output y is partly determined by the poles of the plant transfer function G. Thus, if the plant has slow or lightly damped poles, the corresponding transient responses will be present in the output. In closed-loop control, the closed-loop poles are the zeros of $1 + GC$; these can be completely different from the plant poles, and the controller can be designed to achieve desired pole locations and a desired transient response. This will be discussed in detail in the following chapters.

Table 2.1: Tracking and disturbance rejection in open and closed loop control

Objective	Open loop	Closed loop
tracking $(d = 0)$	$Y = GCR$	$Y = \dfrac{GC}{1 + GC}R$
disturbance rejection $(r = 0)$	$Y = GD$	$Y = \dfrac{G}{1 + GC}D$

2.2 Types of Feedback

In this section, three basic types of feedback are introduced: proportional feedback, derivative feedback and integral feedback.
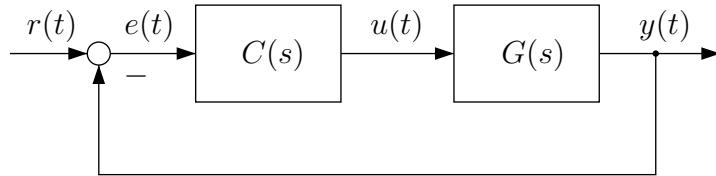

Figure 2.4: Feedback loop

Proportional Feedback

The type of feedback employed in the previous section is proportional feedback: the control input is linearly proportional to the control error. The control law is

$$u(t) = K_{\mathrm{P}}(r(t) - y(t)) = K_{\mathrm{P}}e(t).$$

The steady state behaviour of the closed-loop system with such a controller was already discussed; we now turn to its transient behaviour. Assume that the plant in Figure 2.4 is a second order system with transfer function

$$G(s) = \frac{1}{s^2 + a_1 s + a_0}$$

and the controller is a proportional gain, i.e. $C(s) = K_{\mathrm{P}}$. The closed-loop transfer function is

$$G_{cl}(s) = \frac{Y(s)}{R(s)} = \frac{K_{\mathrm{P}}}{s^2 + a_1 s + a_0 + K_{\mathrm{P}}}, \tag{2.1}$$

and comparison with the plant transfer function

$$s^2 + a_1 s + a_0 + K_{\mathrm{P}} = 0$$

shows the only difference is that a_0 is replaced by $a_0 + K_P$. Comparing this with the standard form of the characteristic equation of a second order system

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$$

we see that the effect of K_P apart from the steady state gain is a change of the natural frequency ω_n. It is also helpful to recall the mechanical example introduced in Section 1.1 and shown in Figure 1.1. The characteristic equation of this system is (compare equation (1.1))

$$s^2 + \frac{b}{m}s + \frac{k}{m} = 0$$

where k is the spring constant, b the damping coefficient and m the mass. The coefficient k/m represents the force of the spring that restores the system to its equilibrium, whereas the coefficient b/m represents the effect of the damper. In this example, using proportional feedback to control the system is equivalent to changing its spring constant.

To study the effect of proportional feedback on the closed-loop poles, we initially make the simplifying assumption $a_0 = 0$ (in terms of the above example, that means the plant itself has no spring and the spring represents the controller). The closed-loop poles are then

$$s_{1,2} = -\frac{a_1}{2} \pm \sqrt{\left(\frac{a_1}{2}\right)^2 - K_P}.$$
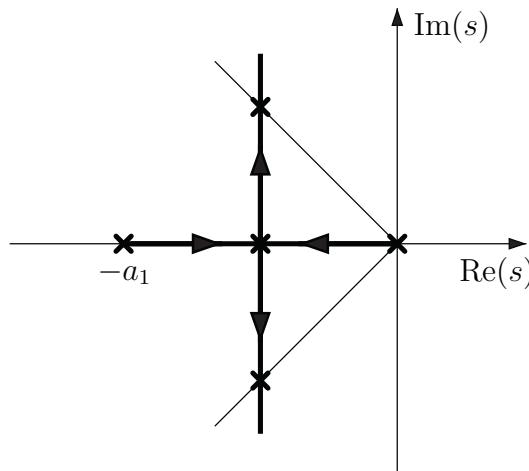
Figure 2.5: Closed-loop pole locations when $a_0 = 0$

The location of the poles for different values of K_P is shown in Figure 2.5. When $K_P = 0$, the roots are $s_1 = 0$ and $s_2 = -a_1$, which are the open-loop poles of the plant without control. (Note that as $K_P \to 0$, the closed-loop poles approach the open-loop poles but at the same time the controller gain goes to zero.) When K_P is increased, the poles

move towards each other along the negative real axis, and for $K_P = a_1^2/4$ they meet at $s = -a_1/2$. When K_P is further increased, the poles become complex

$$s_{1,2} = -\frac{a_1}{2} \pm j\sqrt{K_P - \left(\frac{a_1}{2}\right)^2}$$

with real part $-a_1/2$, and imaginary part increasing with K_P. When $K_P = a_1^2/2$, the poles are

$$s_{1,2} = -\frac{a_1}{2} \pm j\frac{a_1}{2},$$

and the angle between the poles and the imaginary axis is 45°.

When we remove the assumption $a_0 = 0$, we have to distinguish two cases: the open-loop poles can be real or complex. In both cases, the behaviour of the closed-loop poles is similar to that when $a_0 = 0$; both cases are shown in Figure 2.6. The plots in Figures 2.5 and 2.6 are called *root locus plots*, the use of such plots for analysis and design of controllers will be studied in Chapter 3.
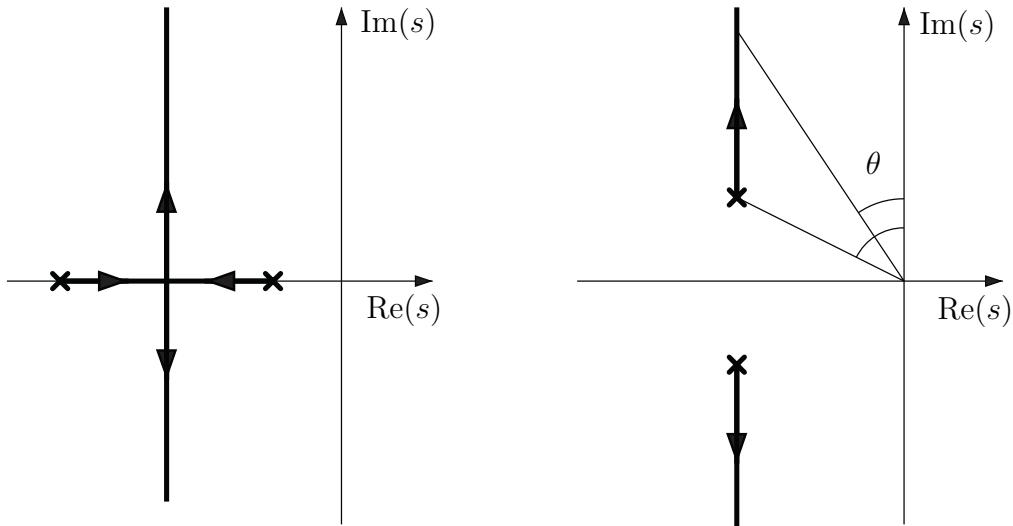
Figure 2.6: Closed-loop pole locations with real open-loop pole pair (left) and complex open-loop pole pair (right)

When we considered the steady state error under proportional feedback, we found that from a steady state point of view, the feedback gain should be large in order to make the error small. The present discussion shows that making the feedback gain large has an undesired effect on the transient behaviour: as indicated in Figure 2.6, a higher gain K_P leads to a smaller angle θ between poles and imaginary axis. But from Section 1.3 we know that $\sin \theta = \zeta$, therefore a large proportional feedback gain leads to a poor transient response with low damping ratio, large peak overshoot and oscillation. In terms of the mechanical example cited above, this is a consequence of the fact that proportional feedback can only modify the stiffness of the spring, it cannot change the damping coefficient.

We will now see that the latter can be achieved by a combination of proportional feedback and derivative feedback.

Proportional plus Derivative Feedback

The control law for proportional plus derivative feedback (also referred to as *PD control*) is

$$u(t) = K_{\mathrm{P}}(e(t) + T_{\mathrm{D}}\dot{e}(t)),$$

where $e = r - y$ is the control error. Taking Laplace transforms, we have

$$U(s) = K_{\mathrm{P}}(1 + T_{\mathrm{D}}s)E(s),$$

thus the controller transfer function is

$$C(s) = K_{\mathrm{P}}(1 + T_{\mathrm{D}}s). \tag{2.2}$$

The parameter T_{D} is called the *derivative time* .

To study the effect of PD control on the closed-loop poles, we consider again the plant with transfer function

$$G(s) = \frac{1}{s^2 + a_1 s + a_0}.$$

With this plant and the controller transfer function given above, the closed-loop transfer function of the feedback system in Figure 2.4 is

$$G_{cl}(s) = \frac{K_{\mathrm{P}}(1 + T_{\mathrm{D}}s)}{s^2 + a_1 s + a_0 + K_{\mathrm{P}}(1 + T_{\mathrm{D}}s)}.$$

The characteristic equation is

$$s^2 + (a_1 + K_{\mathrm{P}}T_{\mathrm{D}})s + a_0 + K_{\mathrm{P}} = 0,$$

it shows an additional degree of freedom in design offered by PD control: the proportional gain K_{P} can still be used to reduce steady state error and to change the natural frequency (recall the standard form of the characteristic equation for a second order system). In addition, the parameter T_{D} can be used to change the damping ratio independently of the natural frequency.

Integral Feedback

The third type of feedback is integral feedback. The reason for using integral feedback is to bring the steady state error to zero. This is best explained in frequency domain. The steady state error is zero when the closed-loop steady state gain from r to y is 1; from (2.1) we have

$$G_{cl}(0) = \frac{K_{\mathrm{P}}}{a_0 + K_{\mathrm{P}}} \quad \to \quad 1 \quad \text{as} \quad K_{\mathrm{P}} \to \infty.$$

The steady state error under proportional feedback becomes zero only when the feedback gain becomes infinite. Apart from not being realizable, a large proportional gain is also undesirable because it can lead to a poor transient response. An important point to observe here is that the gain needs to be infinite *only in steady state*, i.e. when $s = 0$. If the gain is infinite for $s = 0$ but finite for $s \neq 0$, a zero steady state error can be achieved while avoiding undesirable effects on the transient response. These considerations suggest to include a factor $1/s$ in the controller transfer function. This factor has precisely the required properties: in steady state it is infinite, and for nonzero values of s it is finite.

Thus, we consider the controller transfer function

$$C(s) = K_P \frac{1}{T_I s}. \tag{2.3}$$

which has the property $C(0) = \infty$. From

$$U(s) = \frac{K_P}{T_I s} E(s).$$

we obtain

$$u(t) = \frac{K_P}{T_I} \int_{t_0}^{t} e(\tau) \, d\tau.$$

The fact that the control input is generated by integrating the control error gives this type of feedback its name. The parameter T_I is called the *integral time* or *reset time.*

To study the effect of integral feedback on the closed-loop behaviour, we consider the first order plant model

$$G(s) = \frac{K_0}{\tau s + 1}.$$

With the controller transfer function $C(s)$ in (2.3), the closed-loop transfer function is

$$G_{cl} = \frac{Y(s)}{R(s)} = \frac{K_P K_0}{T_I s(\tau s + 1) + K_P K_0},$$

and setting $s = 0$ shows that

$$G_{cl}(0) = \frac{K_P K_0}{K_P K_0} = 1,$$

thus $y(t) = r(t)$ as $t \to \infty$, i.e. the steady state error is indeed zero.

The characteristic equation is

$$s^2 + \frac{1}{\tau} s + \frac{K_P K_0}{T_I \tau} = 0.$$

Increasing the controller gain K_P/T_I leads to a faster decay of the steady state error, but the characteristic equation shows that increasing this gain also leads to a low damping ratio. In conclusion, integral feedback brings the steady state error to zero, but on its own produces poor transient behaviour.

In practice, the three types of feedback introduced here are used in combination, and the resulting control law is known as *PID control.*

PID Control
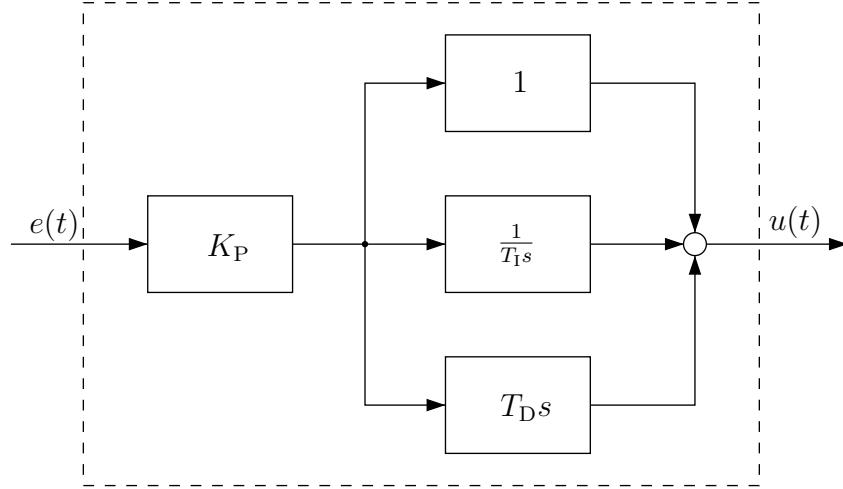
Figure 2.7: PID controller

Combining proportional, integral and derivative feedback leads to the control law

$$u(t) = K_{\mathrm{P}} \left(e(t) + \frac{1}{T_{\mathrm{I}}} \int_{t_0}^{t} e(\tau) \, d\tau + T_{\mathrm{D}} \dot{e}(t) \right).$$

The transfer function of this controller is

$$C(s) \;=\; K_{\mathrm{P}} \left(1 + T_{\mathrm{D}} s + \frac{1}{T_{\mathrm{I}} s} \right). \tag{2.4}$$

A block diagram of a PID controller is shown in Figure 2.7.

PID control was developed in the 1930s and has been widely used since. Most commercially available industrial controllers today are of this type. There are three basic design parameters: proportional gain, integral time and derivative time; these parameters are the tuning knobs of a PID controller. Roughly speaking, a small integral time T_{I} brings the steady state error quickly to zero, but at the same time reduces the damping ratio; increasing T_{D} increases the damping ratio, and increasing K_{P} increases the natural frequency and thus the speed of the response.

A simple example that illustrates the effect of the three tuning parameters is shown in Figure 2.8. A second order system with transfer function

$$G(s) = \frac{1}{s^2 + 0.7s + 1}$$

is subjected to a unit disturbance step at plant input at time $t = 1$. With proportional feedback $K_{\mathrm{P}} = 1$, a steady state error of 0.5 remains, and there is a significant overshoot.
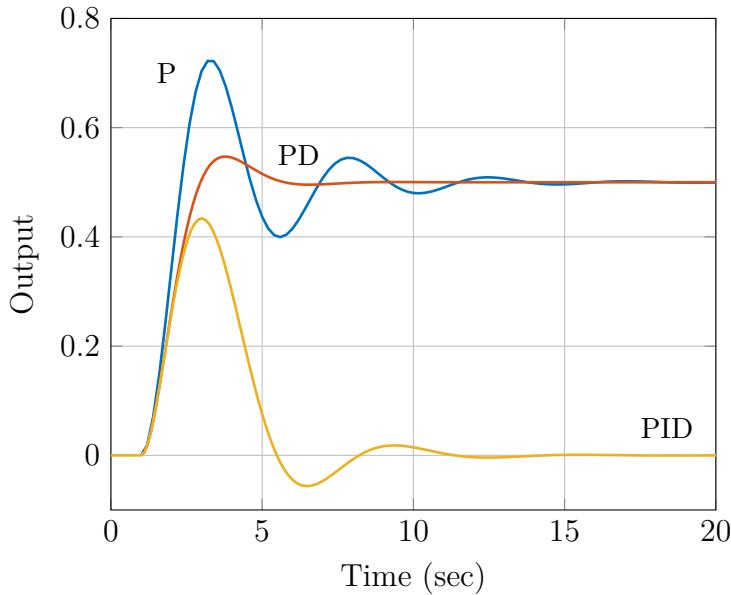
Figure 2.8: Response to an input disturbance with P, PD and PID control. Desired is to bring the output back to zero - this is achieved only by the PID controller.

Adding derivative feedback ($T_\mathrm{D} = 1$) reduces overshoot and oscillation, but leaves the steady state error unaltered. Adding integral action ($T_\mathrm{I} = 1$) finally brings the steady state error to zero, but slightly reduces the damping ratio.

PI Control

In practice, a combination of proportional and integral feedback only is also used, the resulting control law is known as PI control. The controller has the transfer function

$$C(s) = K_\mathrm{P} \left(1 + \frac{1}{T_\mathrm{I}s}\right). \tag{2.5}$$

Note that this controller has in addition to the pole at $s = 0$ also a zero at $s = -1/T_\mathrm{I}$.

Implementation

The $T_\mathrm{D}s$ block in Figure 2.7 represents a differentiator and is not physically realizable. A step change in the input to this block would result in a delta impulse as output. Apart from being not realizable, such a block has an undesired effect on the closed-loop performance, because a differentiator amplifies high frequency measurement noise. In practice, this block is replaced by a block with transfer function

$$\frac{T_\mathrm{D}s}{1 + \gamma T_\mathrm{D}s} \tag{2.6}$$

where γ is a small positive constant. Here with a typical value of $\gamma \approx 0.1$. The step response of this block is an exponentially decaying pulse with pulse height $1/\gamma$ and time

constant γT_{D}. For small values of gamma this is a good approximation of derivative feedback. The pole at $-1/(\gamma T_{\mathrm{D}})$ has the effect of a low pass filter and attenuates high frequency noise.

The transfer function of a PD controller where the $T_{\mathrm{D}}s$ block has been replaced by (2.6 is then

$$C(s) = K_{\mathrm{P}} \left(1 + \frac{T_{\mathrm{D}}s}{1 + \gamma T_{\mathrm{D}}s} \right). \tag{2.7}$$

We will refer to a PD controller in this form as a *real PD controller*, whereas we call the controller (2.2) an *ideal PD controller*.

Similarly, the PID controller (2.4) is referred to as an *ideal PID controller*, in order to distinguish it from a *real PID controller*

$$C(s) = K_{\mathrm{P}} \left(1 + \frac{1}{T_{\mathrm{I}}s} + \frac{T_{\mathrm{D}}s}{1 + \gamma T_{\mathrm{D}}s} \right). \tag{2.8}$$

Ziegler-Nichols Tuning Rules

There are a number of simple, heuristic tuning rules for PID controllers. In process control, the tuning rules developed by Ziegler and Nichols are widely used. We briefly present two methods.

Ziegler-Nichols Method 1

The first method is based on the observation that many plants in process control systems have a step response with the shape shown in Figure 2.9.
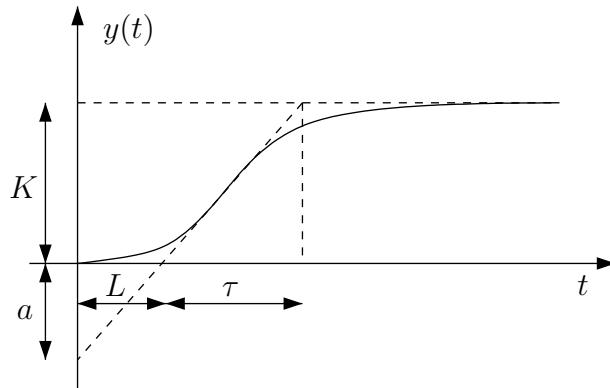
Figure 2.9: Parameters for Ziegler-Nichols tuning rules

If a plot of the step response is available, then one can fit a tangent at the inflection point of the response. The shape of the response is characterized by the values of the steady state gain K, the time lag L and the time constant τ. The value a is determined by the slope of the tangent and by the time lag; note that $a = KL/\tau$. Having read the values of L and a from the plot, the table below provides the setting of a P, PI or PID controller.

The rules in this table were not derived theoretically, but are the result of extensive simulation trials. They provide a reasonable setting for controller parameters when $0.1 < L/\tau < 1$. When the lag L is too large, a controller designed for large time delays is required, and when L is too small, the values in the table lead to a very large proportional gain.

Controller	K_P	T_I	T_D
P	$1/a$	∞	0
PI	$0.9/a$	$3L$	0
PID	$1.2/a$	$2L$	$L/2$

Ziegler-Nichols Method 2

A second method works as follows: if the plant is stable and has pole excess of at least three, we close a proportional feedback loop around the plant. For sufficiently small gains the closed-loop system will be stable. Increasing the gain will lead to a point where the loop becomes unstable. The gain at which this happens is called the *critical gain* K_{cr}. There will be a steady oscillation, and the period of that oscillation is called the *critical period* P_{cr}. Having determined the critical gain and the critical period of a plant, the controller gains can be obtained from the following table.

Controller	K_P	T_I	T_D
P	$0.5 \cdot K_{cr}$	∞	0
PI	$0.45 \cdot K_{cr}$	$0.83 \cdot P_{cr}$	0
PID	$0.6 \cdot K_{cr}$	$0.5 \cdot P_{cr}$	$0.125 \cdot P_{cr}$

Controllers tuned according to these rules often result in lightly damped, oscillatory closed-loop systems, but the parameter values are useful as starting values for subsequent fine tuning. Ziegler-Nichols tuning is illustrated in Exercise 2.5.

Integrator Windup Effect

Every actuator in a control system has physical limits to the range of the control input that can be applied to the plant. If the controller generates a control signal that exceeds these limits, the actuator is driven into saturation. In that case the feedback loop is effectively cut open. Moreover, if the control law includes integral action, the integrator keeps integrating the control error and may charge up to high values until the error changes sign. This can have a significant effect on the transient behavior of the closed-loop system.

A PID control loop with actuator saturation is shown in Figure 2.10.
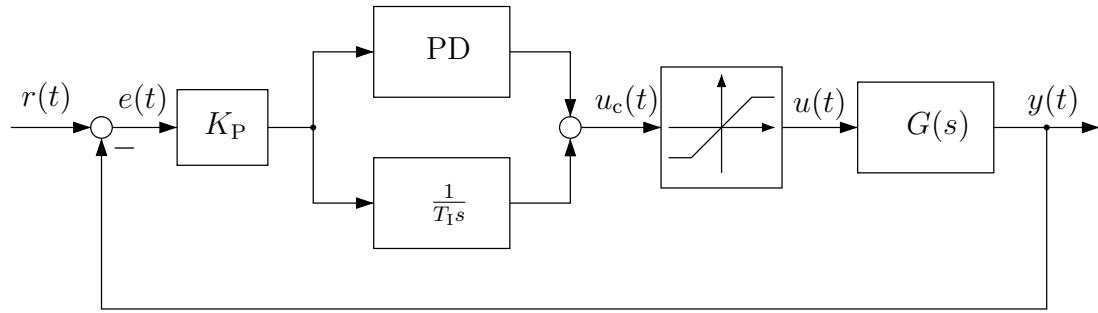
Figure 2.10: PID control loop with actuator saturation

Actuator saturation is represented by a block between controller and plant. The control signal u_c is generated by the controller, and the signal u represents the control input that is actually applied to the plant. The relationship between u and u_c is shown in Figure 2.11.
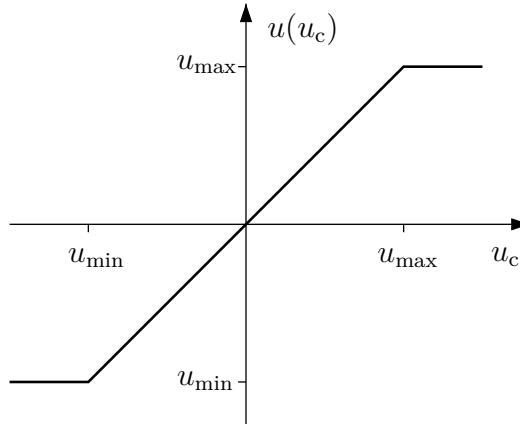
Figure 2.11: Actuator saturation

The effect that saturation has on the step response of a PID control loop is illustrated in Figure 2.12. After a step change from 0 to 1 of the reference input, the controller generates a control signal u_c that exceeds the actuator range, which is limited to $-1 < u < 1$. The integrator is being charged up ("winding up") until the control error $r - y$ changes sign and becomes negative at $t = 1.1$, even though the control system is effectively running in open loop. It takes a significant time for the negative error to wind down the integrator and - supported by the PD component of the controller - to bring the control output back to its physical limits. The result is a huge overshoot; this effect is referred to as *integrator windup*.

The problem is that in the open-loop mode caused by saturation, the integrator represents an unstable element. A variety of techniques can be used to overcome this problem; a simple method of doing this is illustrated in Figure 2.13.

The difference between the control signal u_c generated by the controller and the actual control input u is used for feedback around the integrator. When the actuator is not sat-
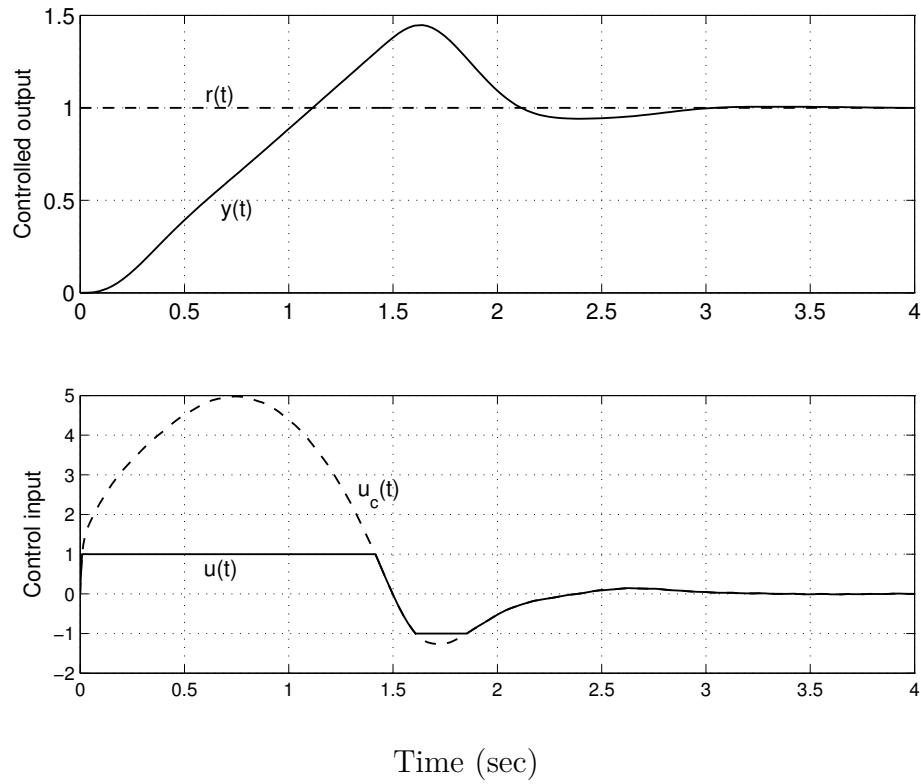
Figure 2.12: Step response with windup effect

urated the integrator operates normally. If saturation occurs, the feedback loop stabilizes the integrator and makes it behave like a first order system, which can be tuned via the feedback gain K_{aw}. Figure 2.14 illustrates how the step response shown in Figure 2.12 changes if the antiwindup configuration of Figure 2.13 is used. Tuning a PID controller with antiwindup is explored in Exercise 2.7.

2.3   Steady State Tracking Error and System Types

In this section we further investigate the steady state tracking error of closed-loop systems and classify feedback systems by their capability of tracking inputs with zero steady state error. In addition to step inputs we will also consider tracking of ramp inputs.

We begin with a few examples. Consider the feedback system in Figure 2.15 with plant transfer function

$$G(s) = \frac{K_0}{\tau s + 1}$$

and proportional feedback $C(s) = K_\mathrm{P}$. The closed-loop transfer function is

$$G_{cl}(s) = \frac{K_\mathrm{P} K_0}{\tau s + 1 + K_\mathrm{P} K_0},$$
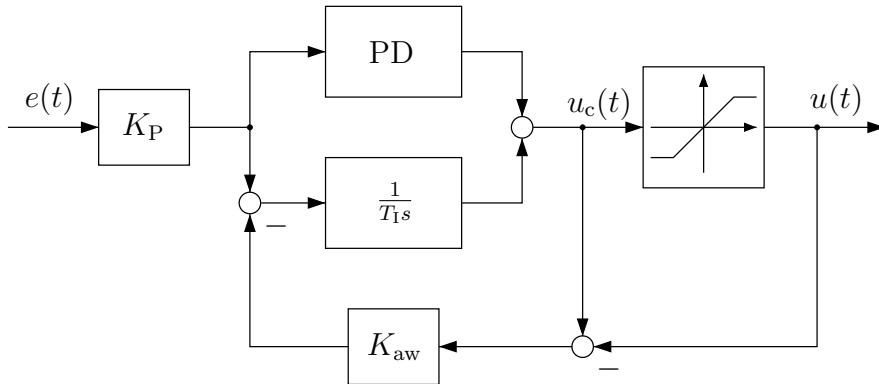
Figure 2.13: Anti-windup configuration

and the closed-loop steady state gain is

$$G_{cl}(0) = \frac{K_P K_0}{1 + K_P K_0} \neq 1,$$

therefore there will be a non-zero steady state error in the response to a constant reference input. When the plant transfer function is changed to

$$G(s) = \frac{K_0}{s(\tau s + 1)},$$

the closed-loop transfer function becomes

$$G_{cl}(s) = \frac{K_P K_0}{s(\tau s + 1) + K_P K_0},$$

and the steady state gain is

$$G_{cl}(0) = \frac{K_P K_0}{K_P K_0} = 1,$$

thus the steady state error is zero. Obviously this is due to the factor $1/s$ in the plant transfer function which indicates integral behaviour of the plant. The effect is the same as that of integral feedback discussed in the previous section, and the example illustrates that a factor $1/s$ in the forward path enables the feedback system to reach a constant setpoint with zero steady state error. Because it makes no difference whether this factor appears in the plant transfer function or in the controller, we simplify the feedback system in Figure 2.15 by combining plant and controller transfer function into the *loop transfer function $L(s) = G(s)C(s)$*, and consider the feedback system shown in Figure 2.16.

Steady State Error to a Ramp Input

So far we considered steady state errors to unit steps as reference inputs. A different type of input that is also of practical interest is the unit ramp input $r(t) = t\sigma(t)$. A ramp input as reference occurs for example when a position control system is required to track an object that is moving with constant velocity.
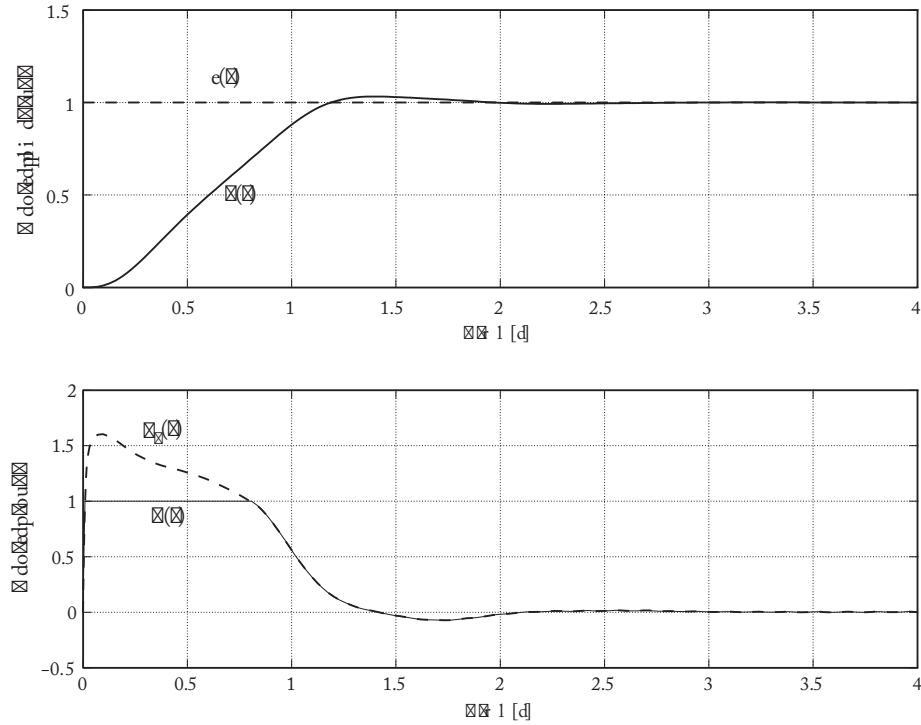
Figure 2.14: Step response with anti-windup configuration
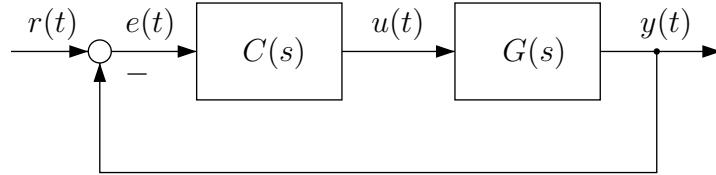
Figure 2.15: Feedback system

To compute the steady state error to a unit ramp input, we need to apply the final value theorem of the Laplace transform to the steady state error; therefore we need to express the steady state error $e(t)$ in terms of the reference input $r(t)$. We find the transfer function from r to e in Figure 2.16 by applying the rule for closed-loop transfer functions (forward gain divided by 1 minus loop gain) to be

$$\frac{E(s)}{R(s)} = \frac{1}{1 + L(s)}.$$

Now we return to the previous example. In terms of Figure 2.16 we have

$$L(s) = K_{\mathrm{P}}G(s) = \frac{K_{\mathrm{P}}K_0}{s(\tau s + 1)}.$$

We saw that the feedback system can follow a step change with zero steady state error. The Laplace transform of the unit ramp is $1/s^2$, thus the error to a unit ramp input is

$$E(s) = \frac{1}{1 + L(s)}R(s) = \frac{s(\tau s + 1)}{s(\tau s + 1) + K_{\mathrm{P}}K_0}\frac{1}{s^2},$$
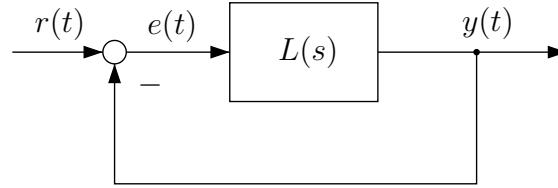
Figure 2.16: Simplified representation of the feedback system in Figure 2.15

and in steady state

$$e_\infty = \lim_{s \to 0} sE(s) = s\frac{s(\tau s + 1)}{s(\tau s + 1) + K_\mathrm{P} K_0} \frac{1}{s^2}$$

or

$$e_\infty = \frac{1}{K_\mathrm{P} K_0} \neq 0.$$

Hence, this feedback system can follow a ramp input only with a nonzero steady state error, and the error is small when the gain product $K_\mathrm{P} K_0$ is large.

Now consider a plant with transfer function

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

With proportional feedback as before, we have

$$L(s) = \frac{K_\mathrm{P}(s + 3)^2}{s^2(s + 1)} = \frac{n_L(s)}{d_L(s)}$$

and with

$$\frac{1}{1 + L(s)} = \frac{d_L(s)}{d_L(s) + n_L(s)}$$

we have

$$e_\infty = \lim_{s \to 0} sE(s) = \lim_{s \to 0} s\frac{1}{1 + L(s)} R(s)$$

$$= \lim_{s \to 0} s\frac{s^2(s + 1)}{s^2(s + 1) + K_\mathrm{P}(s + 3)^2} \frac{1}{s^2}$$

or

$$e_\infty = 0.$$

Because of the factor $1/s^2$ in the forward path, this feedback system can follow a ramp input with zero steady state error. Obviously, the capability of tracking reference inputs with zero steady state error is related to the power of the factor $1/s$ in the forward path, and this observation leads to the following classification of feedback systems.

Definition 2.1

Consider the feedback system shown in Figure 2.16. If k is the largest integer such that the transfer function L(s) can be written in the form

$$L(s) = \frac{n_L(s)}{s^k d_L(s)},$$

i.e. if the denominator has a factor s^k, then this feedback system is called a type k system *.*

With this definition, the examples considered in this section are a type 0 system, a type 1 system and a type 2 system, respectively. A useful measure of the capability of tracking reference inputs is provided by the concept of *error constants* .

Position Error Constant

The position error constant of the feedback system in Figure 2.16 is defined as

$$K_{pos} = \lim_{s \to 0} L(s) = \lim_{s \to 0} \frac{n_L(s)}{s^k d_L(s)}.$$

Because the steady state error to a unit step input is given by

$$e_\infty = \lim_{s \to 0} sE(s) = \lim_{s \to 0} s \frac{1}{1 + L(s)} \frac{1}{s} = \frac{1}{1 + \lim_{s \to 0} L(s)},$$

we can express the steady state error (in response to a unit step) in terms of the position error constant as

$$e_\infty = \frac{1}{1 + K_{pos}}.$$

For type 0, type 1 and type 2 systems, it is straightforward to compute the following position error constants and steady state errors (in response to a unit step).

Type	K_{pos}	e_∞
0	$L(0)$	$\frac{1}{1+K_{\text{pos}}}$
1	∞	0
2	∞	0

Velocity Error Constant

The velocity error constant is defined as

$$K_{vel} = \lim_{s \to 0} sL(s) = \lim_{s \to 0} \frac{n_L(s)}{s^{k-1} d_L(s)}. \tag{2.9}$$

Therefore the steady state error to a unit ramp input is

$$
\begin{aligned}
e_\infty &= \lim_{s\to 0} sE(s) = \lim_{s\to 0} s\frac{1}{1+L(s)}\frac{1}{s^2} \\
&= \lim_{s\to 0}\frac{1}{s+sL(s)} = \frac{1}{\lim_{s\to 0} sL(s)}
\end{aligned}
$$

or

$$
e_\infty = \frac{1}{K_{vel}}.
$$

The velocity error constants and the steady state errors to a unit ramp input for different system types are shown in the following table.

Type	K_{vel}	e_∞
0	0	∞
1	$n_{\mathrm{L}}(0)/d_{\mathrm{L}}(0)$	$1/K_{vel}$
2	∞	0

These results are illustrated by the simulated response of the feedback systems in the examples of this section to a ramp input, shown in Figure 2.17 (numerical values are $K_0 = \tau = 1$).
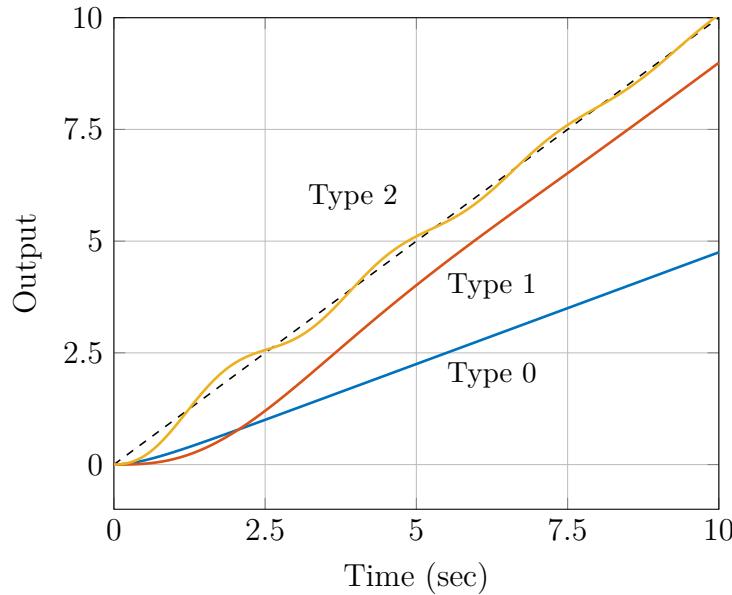
Figure 2.17: Responses of different system types to a unit ramp reference input (dashed)

Internal Model Principle

The fact that the number of integrators in the open-loop transfer function determines the capability of tracking step or ramp inputs with zero steady state error, is a particular

case of a more general principle: the open-loop transfer function must contain a model of the input signal for the feedback system to be able to track it with a zero tracking error. This can be achieved by including a model of the reference input in the controller.

Consider again the feedback system in Figure 2.15, and let

$$G(s) = \frac{b(s)}{a(s)}, \qquad C(s) = \frac{n_c(s)}{d_c(s)}$$

where $n_c(s)$ and $d_c(s)$ denote the numerator and denominator polynomial of the controller transfer function, respectively. Also, let

$$R(s) = \frac{n_r(s)}{d_r(s)}$$

be the Laplace transform of the reference input. The Laplace transform of the control error e is then

$$E(s) = \frac{1}{1 + G(s)C(s)} R(s) = \frac{d_c(s)a(s)}{d_c(s)a(s) + n_c(s)b(s)} \cdot \frac{n_r(s)}{d_r(s)} \qquad (2.10)$$

$$= \frac{C_1}{s - p_1} + \ldots + \frac{C_n}{s - p_n} + \frac{C_{n+1}}{s - r_1} + \ldots + \frac{C_{n+q}}{s - r_q}$$

where p_i are the n closed-loop poles, and r_i are the q roots of $d_r(s)$. Assuming that the closed-loop system is stable, the first n terms in the above expansion represent the transient response and will die out. However, if the reference input is such that $d_r(s)$ has roots on the imaginary axis (as is the case for a step, ramp or sinusoid), the corresponding terms in the partial fraction expansion will not die out but represent a steady state error. The only way for these terms to disappear is to be cancelled in (2.10) by a factor of $d_c(s)a(s)$. This can be achieved by including roots of $d_r(s)$ on the imaginary axis among the controller poles. For example, if a sinusoidal reference input

$$R(s) = \frac{A}{s^2 + \omega^2}$$

is to be tracked, the controller should include a pole pair at $\pm j\omega$.

2.4 Routh's Stability Test

The concepts of system type and error constants introduced in the previous section are meaningful only when the closed-loop system is stable: they refer to the steady state, i.e. the state when transient components of the response have died out. Feedback systems, when not properly tuned, can easily become unstable, and when designing a feedback system, it is important to know how to check its stability.

Stability was discussed in Chapter 1, where we stated that a system is stable if all its poles are in the left half plane. Thus one can check the stability of a system by computing

the poles of its transfer function. This is easily done for first or second order systems, but is hardly possible by hand for high order systems. In this section we present a method that can be used to check stability by simple calculations, without explicitly computing the poles.

Consider the transfer function

$$G(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \ldots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0} = \frac{b(s)}{a(s)}$$

where $n > m$. The stability of the system is determined by the roots of the characteristic equation

$$a(s) = s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0 = 0.$$

A simple test that can be performed by inspection is to check the sign of the coefficients a_i of the characteristic polynomial $a(s)$. If $G(s)$ is stable, all coefficients are positive, and if one or more coefficients are negative or zero, the system is unstable. To make this statement plausible, consider as an example a third order system with characteristic polynomial $a(s)$ and assume the poles are real. This polynomial can be factored so that the characteristic equation is

$$(s + p_1)(s + p_2)(s + p_3) = 0,$$

and the system will only be stable if all p_i's are positive (the poles are at $-p_i$). Expanding the product yields

$$s^3 + (p_1 + p_2 + p_3)s^2 + (p_1 p_2 + p_1 p_3 + p_2 p_3)s + p_1 p_2 p_3 = 0,$$

and because only addition and multiplication is involved, the only way for any of the coefficients to become negative is that at least one of the p_i's is negative, which in turn implies that the system is unstable. The same result can be shown to hold for polynomials of any order and with complex roots.

The condition that all coefficients of the characteristic polynomial must be positive is a *necessary condition* for stability, but it is not *sufficient*: even when all coefficients are positive, the system can be unstable. We now turn to a *necessary and sufficient condition* for stability, the *Routh criterion* (developed in 1875 by the mathematician J. Routh).

The Routh criterion is based on the so-called *Routh array*: the coefficients of the characteristic polynomial are arranged in two rows according to

s^n	1	a_{n-2}	a_{n-4}	a_{n-6}	0
s^{n-1}	a_{n-1}	a_{n-3}	a_{n-5}	a_{n-7}	0
	c_1	c_2	c_3	0	
	d_1	d_2	d_3	0	
	e_1	e_2	0		
	f_1	f_2	0		
	g_1	0			
	h_1				

Here it is assumed that a_{n-7} is the last coefficient (i.e. $n = 7$). From the first two rows, the subsequent rows are computed as follows.

$$c_1 = \frac{-\begin{vmatrix} 1 & a_{n-2} \\ a_{n-1} & a_{n-3} \end{vmatrix}}{a_{n-1}} = \frac{a_{n-1}a_{n-2} - a_{n-3}}{a_{n-1}}$$

$$c_2 = \frac{-\begin{vmatrix} 1 & a_{n-4} \\ a_{n-1} & a_{n-5} \end{vmatrix}}{a_{n-1}} = \frac{a_{n-1}a_{n-4} - a_{n-5}}{a_{n-1}}$$

$$c_3 = \frac{-\begin{vmatrix} 1 & a_{n-6} \\ a_{n-1} & a_{n-7} \end{vmatrix}}{a_{n-1}} = \frac{a_{n-1}a_{n-6} - a_{n-7}}{a_{n-1}}$$

$$c_4 = \frac{-\begin{vmatrix} 1 & 0 \\ a_{n-1} & 0 \end{vmatrix}}{a_{n-1}} = 0$$

$$d_1 = \frac{-\begin{vmatrix} a_{n-1} & a_{n-3} \\ c_1 & c_2 \end{vmatrix}}{c_1} = \frac{c_1 a_{n-3} - c_2 a_{n-1}}{c_1}$$

$$d_2 = \frac{-\begin{vmatrix} a_{n-1} & a_{n-5} \\ c_1 & c_3 \end{vmatrix}}{c_1} = \frac{c_1 a_{n-5} - c_3 a_{n-1}}{c_1}$$

The array terminates with $n+1$ rows, and a necessary and sufficient condition for stability is that all elements in the first column are positive. If not all elements in the first column are positive, then the number of unstable poles is equal to the number of sign changes.

The following example illustrates the procedure. To test a transfer function with denominator polynomial

$$s^3 + 2s^2 + s + 4 = 0,$$

we form the Routh array

$$\begin{array}{c|ccc} s^3 & 1 & 1 & 0 \\ s^2 & 2 & 4 & 0 \\ s^1 & c_1 & 0 \\ s^0 & d_1 \end{array}$$

where

$$c_1 = \frac{-\begin{vmatrix} 1 & 1 \\ 2 & 4 \end{vmatrix}}{2} = \frac{2 - 4}{2} = -1$$

$$d_1 = \frac{-\begin{vmatrix} 2 & 4 \\ -1 & 0 \end{vmatrix}}{-1} = \frac{-4 - 0}{-1} = 4.$$

The elements in the first column $(1,\ 2,\ -1,\ 4)$ show two sign changes (from 2 to -1 and from -1 to 4), indicating two unstable roots. (The roots are $s_{1,2} = 0.16 \pm j1.3$ and $s_3 = -2.3$.)

As a second example, consider the polynomial

$$s^3 + 2s^2 + 4s + 4 = 0.$$

The Routh array is

$$
\begin{array}{cccc}
s^3 & 1 & 4 & 0 \\
s^2 & 2 & 4 & 0 \\
s^1 & 2 & 0 & \\
s^0 & 4 & &
\end{array}
$$

All elements in the first column are positive, indicating that the system is stable. (The roots are $s_{1,2} = -0.35 \pm j1.7$ and $s_3 = -1.3$.)

The Routh test was developed to check whether the roots of a polynomial are in the left half plane, without having to explicitly compute them. The next example demonstrates that the Routh array can also be used to calculate the stability range of feedback gains. Consider the feedback system shown in Figure 2.18.
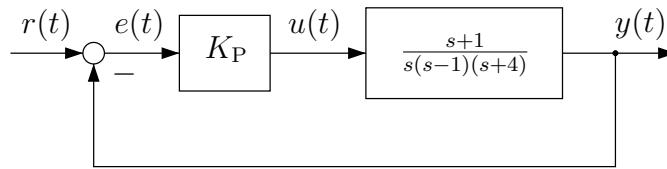
Figure 2.18: Proportional feedback system

The closed-loop transfer function is

$$\frac{Y(s)}{R(s)} = \frac{K_\mathrm{P}(s+1)}{s(s-1)(s+4) + K_\mathrm{P}(s+1)}$$

The plant has an unstable pole at $s = 1$, and we are interested in the range of values of K_P that make the closed-loop system stable. From the characteristic equation

$$s^3 + 3s^2 + (K_\mathrm{P} - 4)s + K_\mathrm{P}$$

we form the Routh array

$$
\begin{array}{cccc}
s^3 & 1 & K_\mathrm{P} - 4 & 0 \\
s^2 & 3 & K_\mathrm{P} & 0 \\
s^1 & c_1 & 0 & \\
s^0 & d_1 & &
\end{array}
$$

with

$$c_1 = \frac{3(K_\mathrm{P} - 4) - K_\mathrm{P}}{3} = \frac{2K_\mathrm{P} - 12}{3}$$

$$d_1 = \frac{c_1 K_P - 0}{c_1} = K_P.$$

The closed-loop system is stable if and only if all elements in the first column are positive, this requires $K_P > 0$ and

$$\frac{2}{3} K_P - 4 > 0$$

therefore the closed-loop system is stable if and only if $K_P > 6$.

Second and Third Order Systems

As discussed above, a necessary condition for a system to be stable is that all coefficients of the characteristic polynomial are positive. When the Routh test is applied to a second order system with the characteristic equation

$$s^2 + a_1 s + a_0 = 0$$

it is straightforward to check that this is also a *sufficient* condition for stability. For a third order system with the characteristic equation

$$s^3 + a_2 s^2 + a_1 s + a_0 = 0$$

applying this test shows that a sufficient condition is that all coefficients are positive and satisfy $a_2 a_1 > a_0$.

Exercises for Chapter 2

Exercises

Problem 2.1

In this task the motor speed $\Omega = \dot{\theta}$ is to be controlled using proportional feedback. The closed–loop system is assumed to be as shown in Figure 2.4. The plant model

$$G(s) = \frac{K_0}{\tau s + 1} \quad \text{with} \quad \tau = \frac{JR}{K^2} \quad \text{and} \quad K_0 = 1/K$$

is taken as a first order model with parameter values obtained in Problem 1.7 ($K \approx 0.0476, R \approx 24, \tau \approx 1.1$).

a) Calculate the position error constant as a function of K_P, and then the smallest value of K_P for which the steady state error is less than 10%.

b) Now assume that a steady state error of 10% is not acceptable, and that it must be reduced to 5%. Calculate the required value of K_P.

Problem 2.2 ⌷M⌷

a) ⌷M⌷ Tune a P controller in simulation: open a new Simulink model and generate a closed-loop system as in Figure 2.4 with the given plant transfer function. Choose $C(s) = K_P = 0.1$, and simulate the response to a reference step with a final value of 47.75 (at this speed the inner dots on the rotating disc of the motor will stand still). Find out for which values of K_P the steady state error is less than 10%.

b) ⌷M⌷ Validate the controller obtained in (2.1 a) experimentally on the real motor: replace the transfer function block with the DC motor block or use the given template. Compare the experimental step response with the simulated one.

c) ⛊ Validate the P controller with the new value of K_P obtained in (2.1 b) experimentally on the real motor, and compare the steady state error with the expected one. Can you observe effects in the experimental step response that you would not expect of a proportional feedback loop with a first order plant model?

SIMULINK file: `Model_Problem2_2_PControllerVelocity.slx`

Problem 2.3 ⛊

Now instead of the motor speed $\Omega = \dot{\theta}$ the motor angle θ is to be controlled with a proportional controller.

a) Adjust the numerical plant model of Problem 1.7.f

$$G(s) = \frac{21}{1.1s + 1}$$

such that the output is the **motor angle** θ by using a free integrator.

b) ⛊ Use the given SIMULINK model to determine K_{cr} and P_{cr} for the Ziegler-Nichols Method 2 [p. 62] experimentally. Use your results to design a proportional controller with gain K_P.

c) ⛊ Validate the P controller with gain from (b) experimentally: Apply a reference step input $r(t) = 2\pi\sigma(t)$, such that the motor is required to make a full 360° rotation. Comment on steady state error and peak overshoot.

   Hint: Be careful to reset the motor in each experiment to the zero initial position.

d) ⛊ Calculate the value of K_P that is required to achieve the fastest possible response but no overshoot. This can be achieved by using two real and identical poles. Simulate the response of the motor with this gain.

e) ⛊ Validate the controller with K_P from (d) experimentally and compare experimental and simulation results. Comment on the steady state error and explain it.

f) ⛊ Since none of the results in (c) and (e) were convincing, we assume now, that a rise time t_r of less than 0.5 s is desired. This chosen rise time is a trade-off between both considered extreme cases. Calculate the value of K_P that is required to achieve this rise time. Simulate the step response with this gain. Perform an experiment to obtain the step response of the real motor. Compare the simulated and experimental response, and comment on the achieved performance. Comment on the overshoot.

Hint: The following equations can be useful:

$$G_{cl}(s) = \frac{K_P K}{JRs^2 + K^2 s + K_P K} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

$$\omega_n \approx \frac{1.7}{t_r}$$

SIMULINK file: `Model_Problem2_3_PControllerAngle.slx`

Problem 2.4 ⓜ

Consider again the model obtained in Problem 2.3.a. In this problem a PD controller is to be designed to control the motor angle.

a) Try to improve the performance achieved with a P controller in Problem 2.3.f by using an ideal PD controller with transfer function (2.2) $C(s) = K_P(1 + T_D s)$. Use $K_P = 0.44$ from this problem, start with $T_D = 0.5$ and fine-tune the derivative time in the range $T_D \in [0.3, \ldots, 0.8]$ by simulating the step response. Apply a reference step input $r(t) = 2\pi\sigma(t)$. How does the choice of T_D affect the rise time and the peak overshoot?

Hint: When simulating the closed-loop response, note that an ideal PD controller is not realizable and cannot be implemented on its own; the product $C(s)G(s)$ is however realizable.

b) ⓜ Repeat the simulation in (a) but this time insert a saturation block with actuator limits $[-8 \ldots 8]$ V before the motor input. What do you observe with regard to the peak overshoot?

Furthermore, validate the controllers from (a) with different values for T_D experimentally. For this purpose you have to implement a real PD controller with transfer function (2.7); choose $\gamma = 0.01$. You may use the template `"Model_Problem2_4_PDControllerAngle.slx"`. Compare simulated and experimental response and comment on the difference.

Hint: Be careful to reset the motor in each experiment to the zero initial position.

SIMULINK files:

- `Model_Problem2_4_PDControllerAngleut.slx`
- `Model_Problem2_4_PDControllerAngleWithSaturation.slx`

Problem 2.5 ⓜ

In this problem a PID controller will be used to control the shaft angle of the DC motor. The model used here is again the one obtained in Problem 2.3.a with the transfer function

$$G(s) = \frac{21}{s(1.1s + 1)} \ .$$

a) Use the values of critical gain ($K_{\text{crit}} = 3.8$) and critical period ($P_{\text{crit}} = 0.74s$) obtained in Problem 2.3.b and the Ziegler-Nichols table [p. 62] to determine start values for the parameters K_{P}, T_{I} and T_{D} of a real PID controller with transfer function (2.8) . Choose $\gamma = 0.1$.

b) Fine-tune your PID controller by simulating the closed-loop step response. The control objectives are

 - peak overshoot less of than 20%,

 - rise time below 0.3 s.

c) ⬛

 i) Validate your controller from (b) experimentally by measuring the response to reference step inputs $r(t) = 2\pi\sigma(t)$ rad . You can use the SIMULINK template `"Model_Problem2_5_PIDControllerAngle.slx"` for this purpose.

 ii) Plot and compare the experimental and the simulated closed-loop response $y(t)$. Plot and compare also the control input signal $u(t)$ used in simulation and in the experiment.

 iii) Do the simulated and the experimental responses match? Explain the differences.

 Hint: Keep in mind that in the experiment only voltages from -8 V to 8 V can be applied to the plant.

d) Now insert a 'Saturation' block with appropriate limits in your SIMULINK model for the simulation and compare again the simulated with the experimental response.

SIMULINK files:

 - `Model_Problem2_5_PIDControllerAngle.slx`
 - `Model_Problem2_5_PIDControllerAngleWithSaturationt.slx`

Problem 2.6 🛩

In this problem the limits of a given system with the transfer function $G(s) = \frac{2}{s^2+s}$ will be considered.

a) Use the given SIMULINK model to design a control loop as in Figure 2.10 using:

 - $r(t) = \sigma(t)$
 - $K_{\text{P}} = 1$
 - $T_{\text{I}} = 1.25$

The saturation block has the limits $-1\,V$ and $+1\,V$. Use the real PD block $\frac{8.6s+16.8}{s+10}$.

b)   Use your SIMULINK model to discuss the relationship between control input $u(t)$
     which is applied to the plant and the control signal $u_c(t)$ which is generated by the
     controller. Also consider the control output $y(t)$ and input $r(t)$ and explain the
     results.

c)   Add a feedback gain K_{aw} to your SIMULINK model as shown in Figure 2.13. Tune
     the gain block K_{aw} to improve the step response. Compare your result with the
     results from part b) and describe the effect of K_{aw}.

SIMULINK file: `Model_Problem2_6_IntegratorWindup.slx`

Problem 2.7

In this problem an anti-windup scheme will be used to overcome the performance limita-
tions encountered in Problem 2.5 c).

Implement a PID controller with anti-windup function and test it experimentally by
simulating step responses. For this purpose, use the given SIMULINK template and your
controller parameters from Problem 2.5 b) ($K_{\mathrm{P}} = 1.2, T_{\mathrm{I}} = 0.44$ and $T_{\mathrm{D}} = 0.38$) as start
values. Set the anti-windup gain to $K_{\mathrm{aw}} = 0.8$. Tune the controller parameters for good
performance (peak overshoot and rise time).

Compare the peak overshoot and the rise time obtained experimentally with that achieved
in Problem 2.5 c) ($t_{\mathrm{r}} = 0.27s$ and $M_{\mathrm{p}} = 0.4$).

SIMULINK file: `Model_Problem2_7_PIDAntiWindup.slx`

Problem 2.8

Consider the control system shown in Figure 2.19, where disturbances acting on plant
input and output are shown. Assume initially that proportional feedback is used, i.e.
$C(s) = K_{\mathrm{P}}$ and the transfer function of the plant is given by

$$G(s) = \frac{1}{s(s+1)}\,.$$
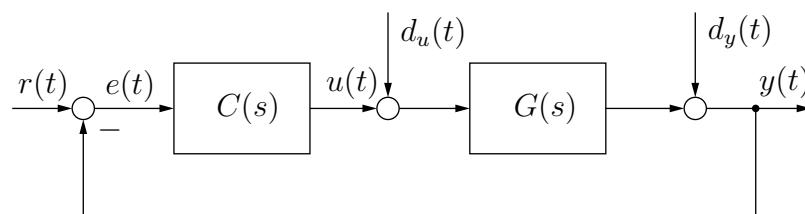
Figure 2.19: Control loop with disturbances

a)   What is the steady state error e_∞ when $r(t) = \sigma(t)$ and $d_u(t) = d_y(t) = 0$?

b)   What is the steady state error in response to a step disturbance d_y acting on the plant output, when $r(t) = d_u(t) = 0$ and $d_y(t) = \sigma(t)$? What is the steady state error in response to a step disturbance d_u acting on the plant input, when $r(t) = d_y(t) = 0$ and $d_u(t) = \sigma(t)$?

c)   Propose a controller transfer function $C(s)$ that achieves zero steady state error in response to a ramp input disturbance $(d_u(t) = t\sigma(t),\ r(t) = d_y(t) = 0)$.

d)   Propose a controller structure $C(s)$ that achieves zero steady state error in response to a sinusoidal disturbance $d_u(t) = sin(\omega_d t)\sigma(t)$, by $r(t) = d_y(t) = 0$ and $\omega_d = 1$.

–   Try to design the controller with two poles. Is the system stable?

–   Which form should have the numerator?

Hint: Use $(5s^2 + 1s + 5)$ *for the numerator. You will do a full design of the controller* $C(s)$ *in the next chapter in Problem 3.6.*

Simulink file: `Model_Problem2_8_InternalModelPrincipleSin.slx`

Problem 2.9

Consider again the angle control problem in Exercise 2.3, where $G(s) = \frac{21}{s(1.1s+1)}$. This time the closed-loop system is required to track a unit ramp as reference input.

a)   Design a P controller $C(s) = K_P$ for this purpose. Which value of K_P is required when $r(t) = t\sigma(t)$, and a steady state error of less than 0.15 is desired?

b)   Design a controller that achieves $e(t) \to 0$, and simulate the closed-loop ramp response.
Hint: You do not have to use a P controller.

Simulink files:

•   `Model_Problem2_9_PAngleRamp.slx`
•   `Model_Problem2_9_PDAngleRamp.slx`
•   `Model_Problem2_9_PIDAngleRamp.slx`

Chapter 3

Root Locus Design

3.1 Root Locus Plots

In this chapter we present a graphical method that can be used to gain information about the *location* of the closed-loop poles without computing them. The method consists of a number of simple rules which can be applied step by step to sketch the *root locus* of the closed-loop system. The root locus is the locus of the closed-loop poles in the complex plane when one parameter - usually a proportional feedback gain - is varied from zero to infinity.

The root locus of a feedback system was already introduced in the first example in Section 2.2, where the limitations of proportional feedback were discussed. Figure 2.5 shows the trajectory of the poles of the closed-loop system (2.1), when the proportional gain K_P takes values between zero and infinity. This root locus plot was constructed by computing the closed-loop poles as a function of K_P. We now present a simple method for sketching the root locus without solving the characteristic equation.

To find the locus of the closed-loop poles, it is not important to distinguish between the transfer function of the plant and of the controller. In order to simplify the following procedure, we assume that the feedback system under consideration has only one block in the forward path as shown in Figure 3.1. The *loop transfer function $L(s)$* can be thought of as the product of the plant transfer function $G(s)$ and the controller transfer function $C(s)$, with a proportional gain K pulled out as a parameter that can be varied.

The purpose of the following procedure is to estimate the location of the closed-loop poles of the feedback system in Figure 3.1 when K is changed from 0 to ∞. We start with the characteristic equation

$$1 + KL(s) = 0,$$

which can also be written as

$$L(s) = -\frac{1}{K}.$$

In general, the transfer function $L(s)$ takes complex values; but the latter form of the characteristic equation shows that if s is a closed-loop pole for some gain K, then $L(s)$ must be real and negative, because K is real and positive. Another way of saying that $L(s)$ is real and negative is to say that the phase angle $\arg L(s)$ is 180°. We conclude that s can only be a closed-loop pole of the feedback system in Figure 3.1 for some gain K between zero and infinity, if the transfer function $L(s)$ has a phase angle of 180°. This leads to the following definition of the root locus.
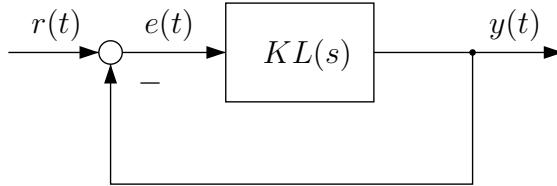
Figure 3.1: Simplified feedback system

Definition 3.1   *The root locus is the locus of points in the s-plane where*

$$\arg L(s) = 180^o + l360^o, \quad l = 0, \pm 1, \pm 2, \dots$$

Note that adding or subtracting integer multiples of 360° does not change the phase angle. We will use the above phase condition to identify points in the s-plane that can be closed-loop poles for some positive value of K. For this purpose, we use the fact that the phase angle of $L(s)$ is equal to the sum of the phase angles of its zeros and the negative phase angles of its poles. To illustrate this, consider an open-loop transfer function $L(s)$ with one zero and three poles

$$L(s) = \frac{K_0(s - z)}{s(s - p_1)(s - p_2)}.$$

Assuming that $K_0 > 0$, we have for the phase angle of $L(s)$

$$\arg L(s) = \arg(s - z) - \arg s - \arg(s - p_1) - \arg(s - p_2).$$

This situation is shown in Figure 3.2. A test point s_0 is marked, and the phase angle of $L(s_0)$ is determined by the angles of the zero and the poles. We have

$$\arg L(s_0) = \psi - \phi_0 - \phi_1 - \phi_2.$$

For example, with values estimated from the plot $\psi \approx 70^o$, $\phi_0 \approx 125^o$, $\phi_1 \approx 25^o$ and $\phi_2 \approx 40^o$, the phase angle of $L(s_0)$ is -120°. As a consequence, the test point s_0 does not belong to the root locus, it cannot be a closed-loop pole for any gain because the phase angle is not 180°.

We will now use this idea to develop a four-step procedure for sketching the root locus of the feedback system shown in Figure 3.1. The steps will be illustrated with an example.
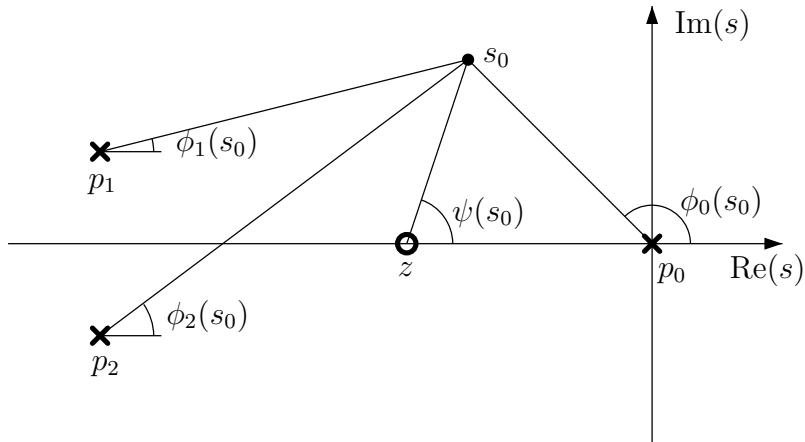
Figure 3.2: Checking the phase condition for a test point

Example 3.1

Consider the open-loop transfer function

$$L(s) = \frac{1}{s(s^2 + 4s + 8)}.$$

There are three open-loop poles: one at $s = 0$, and a pole pair at $s = -2 \pm j2$.

Step 1

The first step is to mark the open-loop poles and zeros in the s-plane. For the above example, the poles are shown in Figure 3.3. The only information needed in the following steps for sketching the root locus is the location of the open-loop poles and zeros.

Note that formally the open-loop poles can be seen as the poles of the feedback loop when $K \to 0$, but in the feedback loop shown in Figure 3.1 the forward gain and therefore the closed-loop transfer function from r to y will approach zero at the same time. The root locus techniques discussed here apply in fact to feedback loops that are more general than that in Figure 3.1, they provide information about the closed-loop poles and therefore about the *loop dynamics*, independent of where input and output signals enter or leave the loop, respectively.

Step 2

The second step is to determine which parts of the real axis belong the root locus. A test point s_0 is marked in Figure 3.3 on the positive real axis. The phase contribution from the complex conjugate pole pair is zero, because the angles ϕ_1 and ϕ_2 add up to zero. Since complex poles or zeros always come in conjugate pairs, it is clear that only open-loop poles or zeros on the real axis contribute to the phase angle of $L(s)$ along the real axis.
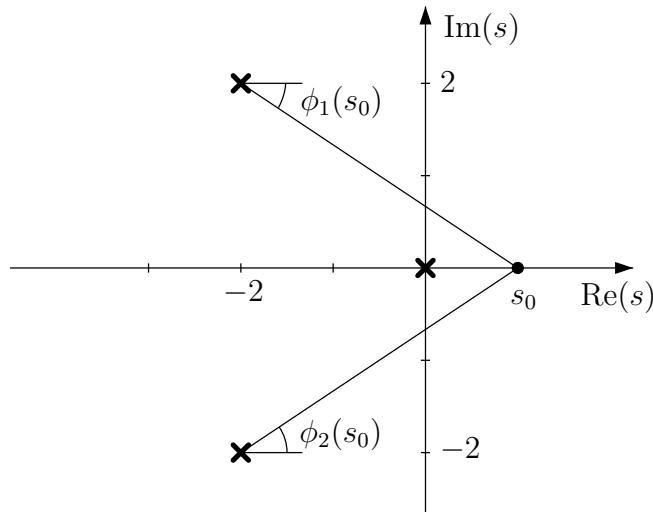
Figure 3.3: Test point on the real axis

To see how the open-loop poles and zeros on the real axis determine which parts of the real axis belong to the root locus, consider the two cases shown in Figure 3.4. In the first case, there are two poles on the real axis. For a test point to the right of both poles, the phase angle is 0° because both pole angles are zero. Hence, points to the right of both open-loop poles cannot be closed-loop poles and do not belong to the root locus. Between the two-poles, the phase contribution from the left pole is still zero, but the angle with the right pole is 180°. Thus, the phase contribution of that pole is -180° (which is the same as 180° because adding multiples of 360° does not change the phase), and the part of the real axis between the two poles belongs to the root locus. To the left of the left pole, the phase contributions of both poles add up to 0°, therefore this part of the real axis does not belong to the root locus.
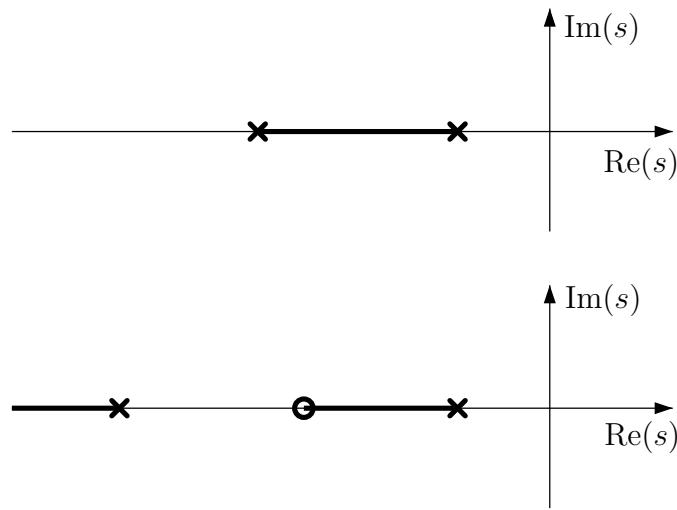
Figure 3.4: Poles and zeros on the real axis

In the second case, a zero is located between two poles. For the same reason as before, points to the right of the right pole do not belong to the root locus: the phase contribution from poles and the zero is 0°. The section between the zero and the right pole belongs to the root locus, because the pole angle is 180°. The section between the zero and the left pole is not on the root locus: both the angle to the right pole and to the zero are 180°, they add up to zero. Finally, to the left of the left pole, angles with the zero and with both poles are 180°, so the phase angle is -180°, therefore the negative real axis to the left of the left pole is part of the root locus.

The above examples show that whether or not a point on the real axis belongs to the root locus is determined by the poles and zeros to the right of that point. A pole or a zero to the right contributes 180° to the phase angle (it is not necessary to distinguish between poles and zeros in this case), whereas a pole or a zero to the left contributes 0°. Because two poles or zeros to the right add up to zero, we can summarize step 2 as follows:

- All points on the real axis to the left of an odd number of poles and zeros belong to the root locus.

Applying this rule to Example 3.1, we find that the entire negative real axis belongs to the root locus because there is only one pole on the real axis, located in the origin.

Step 3

The third step in constructing a root locus sketch is to determine the closed-loop poles as $K \to \infty$. The root locus consists of the n trajectories along which the closed-loop poles move as K is varied from 0 to ∞. We know that for $K = 0$ the closed-loop poles coincide with the open-loop poles, now we will find the opposite end of the trajectories. From the characteristic equation

$$L(s) = -\frac{1}{K}.$$

it is clear that

$$K \to \infty \quad \Rightarrow \quad L(s) = \frac{b(s)}{a(s)} = 0.$$

There are two ways for $L(s)$ to become zero: either s is a zero of the numerator polynomial $b(s)$, or $|s| \to \infty$ (recall that if $\deg a(s) = n$, $\deg b(s) = m$ then $n > m$ for physically realizable systems).

Example 3.2

To illustrate this point, consider

$$L(s) = \frac{s+1}{s^3 + 4s^2 + 8s}$$

This transfer function will be zero either if $s = -1$ or if $|s| \to \infty$.
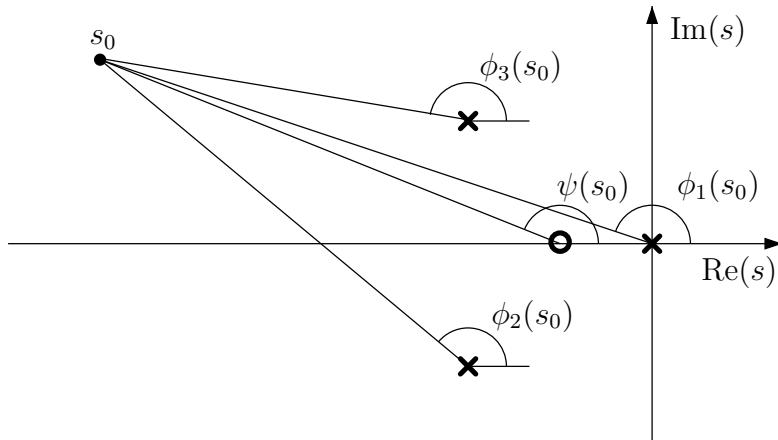
Figure 3.5: Test point for step 3

These two ways of $L(s)$ to become zero indicate the two possibilities of where the n closed-loop poles can go as $K \to \infty$: they either approach one of the m zeros of the loop transfer function, or they approach infinity. Because $n > m$, there will be $n - m$ closed-loop poles that go to infinity. Strictly speaking, it is the magnitude of s that becomes infinite, and we are interested in the phase angle under which this happens. To illustrate how the root locus phase condition can be used to find these angles, we return to Example 3.2. The poles and zeros are shown in Figure 3.5, and for the test point s_0 we have

$$\arg L(s_0) = \psi - \phi_1 - \phi_2 - \phi_3.$$

For a test point far away from the origin, i.e. $|s_0|$ very large, these angles are approximately the same, and as $|s_0| \to \infty$, we have

$$\psi = \phi_1 = \phi_2 = \phi_3.$$
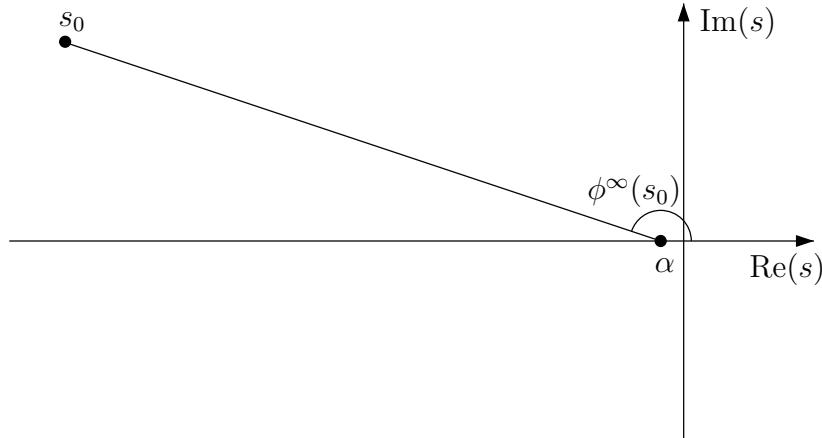
Figure 3.6: Test point far away from origin

This situation is illustrated in Figure 3.6; the poles and zeros now appear clustered at one point α on the real axis, and ϕ^∞ denotes the common angle. Because m zeros and n poles are clustered in the same place, the phase contribution of m poles is cancelled by that of the m zeros, and the net phase contribution is $-(n-m)\phi^\infty$. The phase condition for s_0 to be on the root locus is therefore

$$\arg L(s_0) = -(n-m)\phi^\infty = 180^o + l360^o$$

which can be solved for ϕ_l^∞

$$\phi_l^\infty = \frac{180^o}{n-m} + l\frac{360^o}{n-m}, \quad l = 0, \ldots, n-m-1$$

The fact that adding or subtracting multiples of 360° does not change anything in the root locus phase condition, leads to $n-m$ different possible values of ϕ_l^∞ for which s_0 can be on the root locus: for $n-m = 1$ we have $\phi_0^\infty = 180^o$, for $n-m = 2$ we have $\phi_0^\infty = 90^o$, $\phi_1^\infty = 270^o$, for $n-m = 3$ we have $\phi_0^\infty = 60^o$, $\phi_1^\infty = 180^o$, $\phi_2^\infty = 300^o$ etc.

To draw the asymptotes of the trajectories as $K \to \infty$, we also need to find the point α where they emanate from the real axis. For this purpose we use the following property of polynomials. A degree n polynomial (here the open loop characteristic polynomial $a(s)$) can be written as the product of n linear factors

$$s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0 = (s-p_1)(s-p_2)(s-p_3)\ldots(s-p_n) =$$
$$s^n + (-p_1 - p_2 - p_3 - \ldots - p_n)s^{n-1} + \ldots$$

Here the p_i are the open loop poles, and comparing the last expression with the first shows that

$$a_{n-1} = -\sum_{i=1}^{n} p_i \tag{3.1}$$

i.e. the coefficient a_{n-1} equals the negative sum of the open loop poles.

Now consider the closed-loop characteristic equation

$$1 + K\frac{b(s)}{a(s)} = 0$$

or

$$\bar{a}(s) = a(s) + Kb(s) = 0, \tag{3.2}$$

where $\bar{a}(s)$ denotes the closed-loop characteristic polynomial. Using again the above property of polynomials, we obtain

$$\bar{a}_{n-1} = -\sum_{i=1}^{n} r_i \tag{3.3}$$

where the r_i are the closed-loop poles. Now from (3.2) written as

$$s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0 + K(s^m + b_{m-1}s^{m-1} + \ldots + b_0) =$$
$$s^n + \bar{a}_{n-1}s^{n-1} + \ldots + \bar{a}_0$$

we see that if $m < n-1$, we have $\bar{a}_{n-1} = a_{n-1}$. Because of (3.1) and (3.3), this implies that the sum of the closed-loop poles equals the sum of the open-loop poles. But as $K \to \infty$, the closed-loop poles are the m open-loop zeros z_i together with $n - m$ poles which go to infinity and which add up to $(n - m)\alpha$. Therefore we have

$$\sum_{i=1}^{n} r_i = \sum_{i=1}^{n} p_i = \sum_{i=1}^{m} z_i + (n - m)\alpha$$

Solving for α yields

$$\alpha = \frac{\sum_{i=1}^{n} p_i - \sum_{i=1}^{m} z_i}{n - m}$$

Because complex poles and zeros always come in conjugate pairs, the imaginary parts in the sums add up to zero, and the above can be simplified to

$$\alpha = \frac{\sum_{i=1}^{n} \operatorname{Re} p_i - \sum_{i=1}^{m} \operatorname{Re} z_i}{n - m}$$
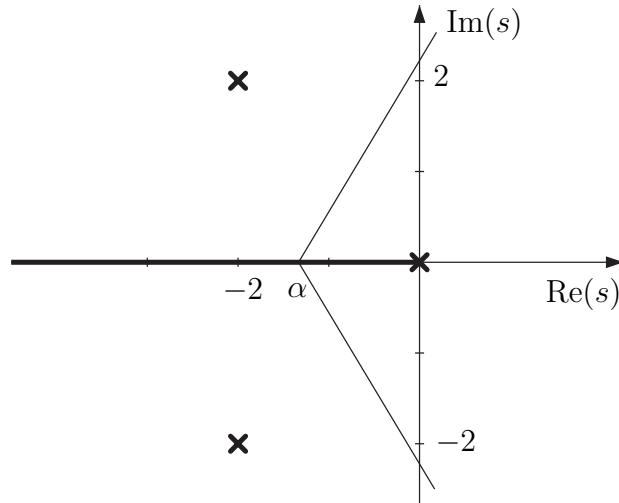
Figure 3.7: Information obtained in step 2 and 3

To illustrate the application of step 3, we return to Example 3.1. We have $n - m = 3 - 0 = 3$, therefore the angles of the asymptotes are $\phi_0^\infty = 60^o$, $\phi_1^\infty = 180^o$, $\phi_2^\infty = 300^o$. Using the above formula to find α gives

$$\alpha = \frac{-2 - 2 - 0}{3} = -1.33$$

Figure 3.7 shows the information gathered so far. The three branches of the root locus start at the open-loop poles, and as $K \to \infty$ the closed-loop poles move towards infinity along the asymptotes. One of these is the negative real axis, along which the pole at the origin moves towards $-\infty$, whereas the two complex conjugate poles move along the 60° and the 300° asymptotes towards infinity. What is still missing at this point is information about the root locus for small values of K, in particular we do not know yet under what

angle the root locus departs from the two complex poles. This information is obtained in step 4.

Step 4

To find the angles of departure , we choose a test point s_0 in the vicinity of the open-loop pole at $-2 + j2$, as shown in Figure 3.8. The phase condition for s_0 to be on the root locus is

$$\arg L(s_0) = -\phi_1 - \phi_2 - \phi_3 = 180^o + l360^o.$$

The angle ϕ_3 is the angle of departure (denoted by ϕ_3^0) when s_0 is close to the pole. The phase contribution of the other poles does not change significantly if we move the test point along a small circle around the pole at $-2 + j2$. From Figure 3.8 we find

$$\arg L(s_0) = -135^o - 90^o - \phi_3^0 = 180^o + l360^o,$$

and solving for ϕ_3^0 gives

$$\phi_3^0 = -405^o + l360^o = -45^o$$

if we take $l = 1$. Thus, the root locus departs at -45° as shown in Figure 3.8, and by symmetry the locus at the lower pole departs at $+45^o$. This information makes it possible to sketch the complete root locus as shown in the same Figure.
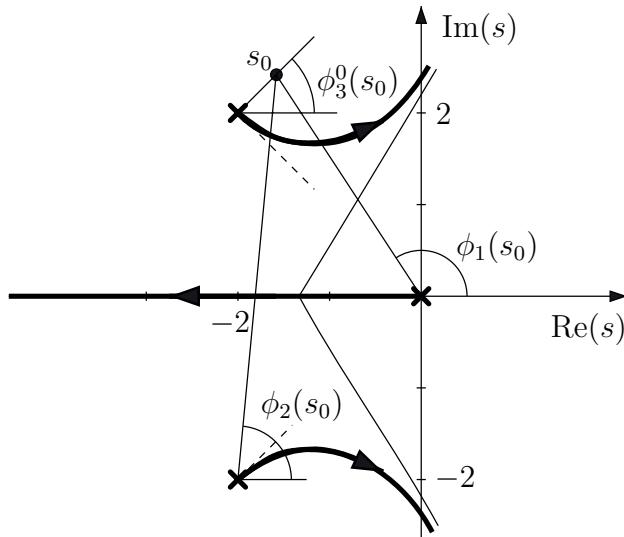
Figure 3.8: Root locus sketch

A comment is in order at this point: the four steps described so far do not give precise information about the locus at small gains, they do not tell how exactly the branches of the locus approach the asymptotes. If precise information is required, it can be taken from a computer-generated root locus. The idea of the 4-step procedure presented here is to gain qualitative information about the closed-loop system without computing closed-loop poles; in fact with a little experience it is possible to guess the shape of the root locus just from looking at the open-loop poles and zeros.

In some cases an additional step may give further useful information. Consider again the example discussed in Section 2.2, where

$$L(s) = \frac{1}{s^2 + s}$$
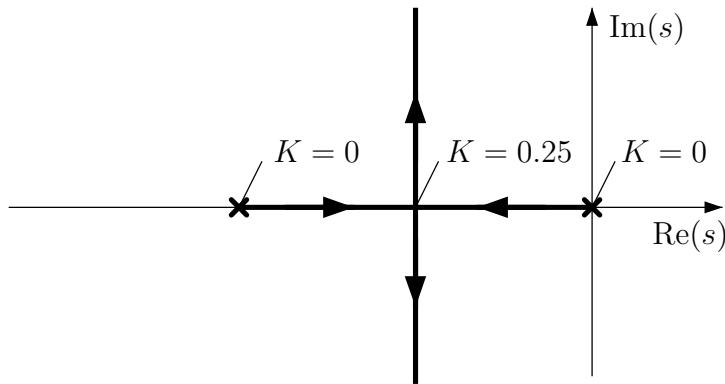
Figure 3.9: Breakaway point

The root locus is shown again in Figure 3.9 (now it can be easily sketched by applying the four steps above without computing the roots). The closed-loop poles move from their open-loop position towards each other, come together at $s = -0.5$, and depart from the real axis. The branching point at -0.5 is called a *break-away point* ; if two symmetric branches of the locus move back to the real axis it is called a *break-in point* . For low order systems, the location of break points can be computed by hand using the following observation. At $s = -1$ we have $K = 0$, and along the real axis towards the origin the gain increases until $s = -0.5$, where $K = 1/4$ (see the discussion in Section 2.2). Moving further towards the origin, the gain decreases back to $K = 0$ at the origin. Thus, the gain has a maximum at the breakaway point (at a break-in point, it has a minimum). This observation suggests to search for a break point by setting the derivative of K with respect to s to zero. We have

$$K = -\frac{1}{L(s)} = -\frac{a(s)}{b(s)}$$

Therefore

$$\frac{dK}{ds} = \frac{d}{ds}\left(-\frac{a(s)}{b(s)}\right) = -\frac{1}{b^2}\left(b\frac{da}{ds} - a\frac{db}{ds}\right) = 0$$

or

$$b\frac{da}{ds} - a\frac{db}{ds} = 0$$

is a condition for s to be a break point. In the present example

$$b(s) = 1 \qquad\qquad a(s) = s^2 + s$$
$$\frac{db}{ds} = 0 \qquad\qquad \frac{da}{ds} = 2s + 1$$

and solving $2s + 1 = 0$ yields $s = -0.5$, which is indeed the break point. Note however that, in contrast to the four steps above, this step involves solving for the roots of a polynomial, which may not be possible to do by hand for higher order systems.

Finally we summarize the steps for sketching a root locus.

Step 1

Mark open-loop poles and zeros in the s-plane.

Step 2

Mark real axis portion of the root locus to the left of an odd number of poles and zeros.

Step 3

Find asymptotes for the $n - m$ root locus branches that go to infinity:

$$\alpha = \frac{\sum_{i=1}^{n} \operatorname{Re} p_i - \sum_{i=1}^{m} \operatorname{Re} z_i}{n - m}, \qquad \phi_l^{\infty} = \frac{180^o}{n - m} + l\frac{360^o}{n - m}, \quad l = 0, \ldots, n - m - 1$$

Step 4

Use the phase condition on a test point along a small circle around an open-loop pole to find the angle of departure:

$$\arg L(s) = \sum_{i=1}^{m} arg(s - z_i) - \sum_{i=1}^{n} arg(s - p_i) = 180^o + l360^o$$

If applicable: **Step 5**

Compute breakaway and break-in points from

$$b\frac{da}{ds} - a\frac{db}{ds} = 0$$

3.2   Root Locus Examples

In this section, we show by way of examples how to apply the procedure for sketching a root locus. In particular, we interpret the effect of PD control in terms of root locus plots.

Example 3.3

We want to sketch the root locus of a feedback system with the open loop transfer function

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

Step 1 is to mark the two poles at $s = 0$ and the zero at $s = -1$. According to the rule of *step 2*, the part of the negative real axis to the left of the point -1 belongs to the root locus.

Because $L(s)$ has two poles and one zero, we have $n - m = 1$, and *step 3* yields

$$\phi_1^\infty = 180^o.$$

In this case, there is only one asymptote that coincides with the negative real axis, and we do not need to compute the intersection α with the real axis.
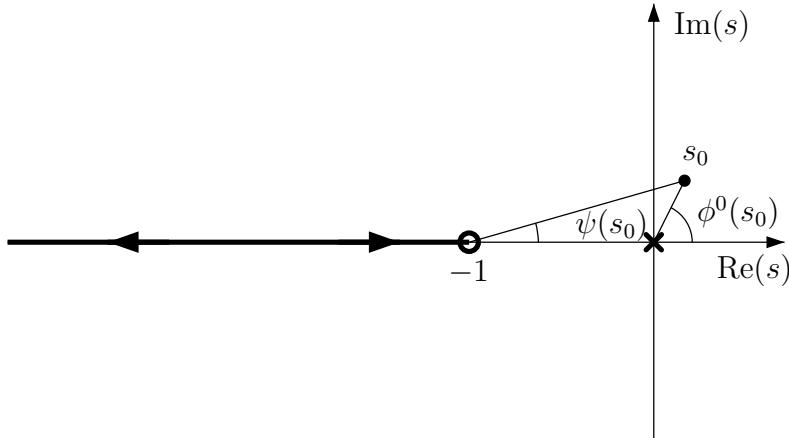
Figure 3.10: Example 3.3, angle of departure

Step 4 is to compute the angles of departure from the two poles at the origin. The information gathered so far is shown in Figure 3.10. A test point s_0 close to the poles at the origin is marked, with angle ϕ^0. The phase contribution ψ from the zero is approximately 0°. Because there are two poles, the phase contribution is $-2\phi^0$, therefore the phase condition is

$$\arg L(s_0) = \psi - 2\phi^0 = -2\phi^0 = 180^o + l360^o,$$

and solving for ϕ^0 yields

$$\phi^0 = -90^o + l180^o \qquad \rightarrow \qquad \phi_0^0 = -90^o, \ \phi_1^0 = 90^o.$$

The two branches of the root locus leave the origin in the direction of the positive and negative imaginary axis. This raises the question of stability: if the root locus stays on the imaginary axis or moves into the right half plane, the closed-loop system becomes unstable. We can form a Routh array to check stability: from the characteristic equation

$$s^2 + Ks + K = 0,$$

the Routh array is

$$
\begin{array}{c|cc}
s^2 & 1 & K \\
s^1 & K & 0 \\
s^0 & c_1 &
\end{array}
$$

with

$$c_1 = \frac{K^2 - 0}{K} = K.$$

This implies that for $K > 0$ the closed-loop system is stable, and we can conclude that the root locus stays in the left half plane for all $K > 0$.

At this stage we know that the two branches of the root locus start at the origin with departure angles $\pm 90^o$, and that for large K both branches are on the negative real axis, with one pole approaching the zero at -1 and the other pole approaching $-\infty$. To complete the root locus sketch, we need to know the shape of the locus branches before they return to the real axis.
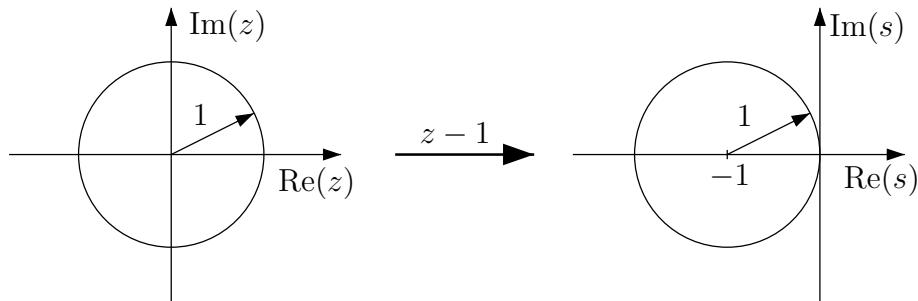
Figure 3.11: Change of variables $s \to z$

In fact the two root locus branches form a circle centered on the zero at -1. To see this, we consider the characteristic equation

$$s^2 + K_P(s + 1) = 0.$$

At this point it is convenient to introduce a change of variables: we replace the variable s by $z = s + 1$ and obtain

$$(z - 1)^2 + K_P z = z^2 + (K_P - 2)z + 1 = 0$$

thus

$$z_{1.2} = \frac{2 - K_P}{2} \pm \sqrt{\left(\frac{2 - K_P}{2}\right)^2 - 1}$$

or for $K_P \leq 4$

$$z_{1.2} = \frac{2 - K_P}{2} \pm j \sqrt{1 - \left(\frac{2 - K_P}{2}\right)^2}.$$

Since $|z| = \sqrt{(\text{Re } z)^2 + (\text{Im } z)^2}$, for $0 \leq K_P \leq 4$ the roots are located on the unit circle in the complex plane. If we now reverse the change of variables, we see that the locus of the roots is a circle with radius 1 centered on -1 (see Figure 3.11).

Similarly it can be shown that if a second order loop transfer function has a zero to the left of two real poles, the root locus will take the form shown in Figure 3.12, where

$$r = \sqrt{|z - p_1||z - p_2|}.$$

This pattern occurs frequently in control applications and will be further discussed below.
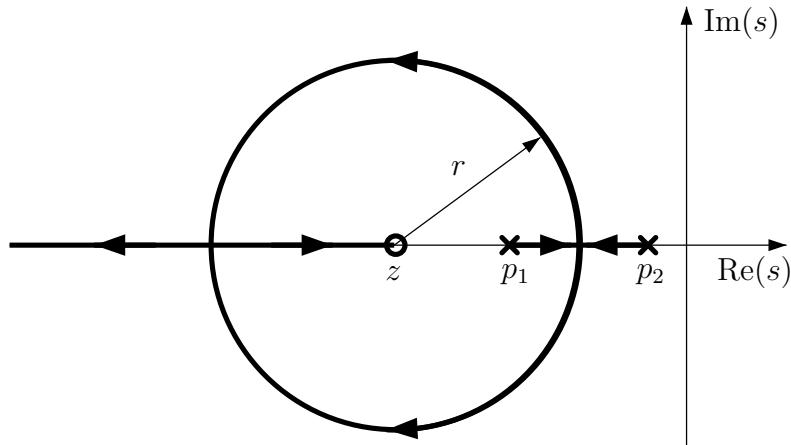
Figure 3.12: Root locus circle pattern

The complete root locus for Example 3.3 is shown in Figure 3.13. We can apply *step 5* to verify the breakpoints:

$$b(s) = s + 1 \qquad\qquad a = s^2$$
$$\frac{db}{ds} = 1 \qquad\qquad \frac{da}{ds} = 2s$$

therefore

$$b\frac{da}{ds} - a\frac{db}{ds} = s^2 + 2s = 0 \quad \rightarrow \quad s_1 = 0, \ s_2 = -2$$

where s_1 is the breakaway point at the origin and s_2 is the break-in point.
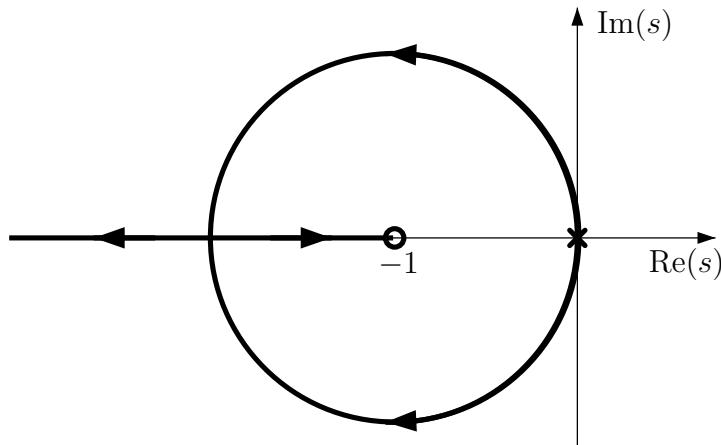
Figure 3.13: Root locus for Example 3.3

Example 3.4

We consider again the feedback system of Example 3.3, but with an additional pole at $s = -5$ of the loop transfer function, thus

$$L(s) = \frac{s+1}{s^2(s+5)}.$$

Applying the rule of *step 2* we find that not the entire negative real axis to the left of the point -1, but only the section between -1 and -5 belongs to the root locus. Because there are three poles and one zero $(n - m = 2)$, two closed-loop poles must go to infinity as $K \to \infty$. Applying *step 3* we find

$$\phi_0^\infty = 90^o, \quad \phi_1^\infty = 270^o, \quad \alpha = \frac{-5 - (-1)}{2} = -2$$

In *step 4*, the departure angles can be computed as in the previous example. Because the additional pole at -5 contributes 0° to the phase angle of $L(s_0)$ for a test point close to the origin, the departure angles are the same as before, i.e. $\pm 90^o$. The complete root locus sketch is shown in Figure 3.14; comparing it with Figure 3.13 shows that the effect of the additional pole is to push the root locus towards the right.
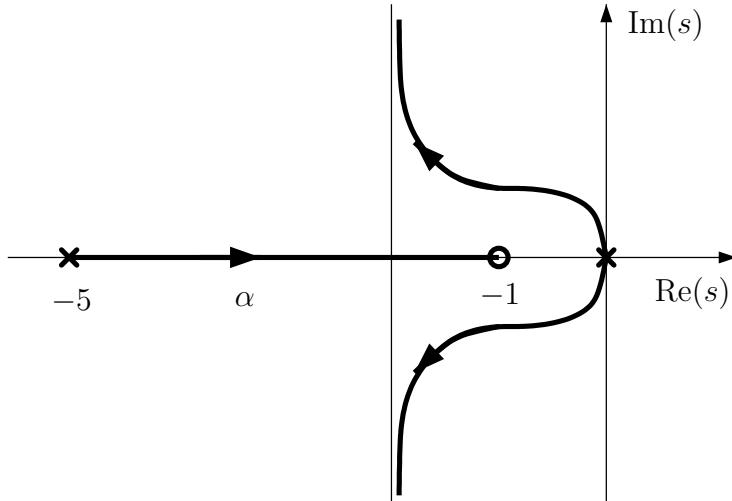
Figure 3.14: Root locus for Example 3.4

Example 3.5 (PD Control)

Consider the feedback system in Figure 3.15. The plant transfer function is

$$G(s) = \frac{10}{s(s+3)}$$

and a controller is to be designed for tracking and disturbance rejection. With proportional feedback

$$C(s) = K_P$$
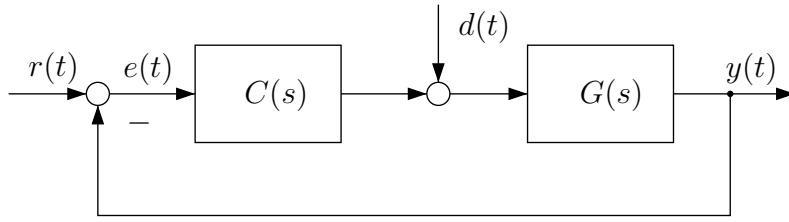
Figure 3.15: Feedback system in Example 3.5

the root locus is readily sketched as shown in Figure 3.16; here we take $K = K_P$ and $L(s)$ as the plant transfer function. The real axis portion of the root locus is the section between 0 and -3, and applying step 3 gives with $n - m = 2$ the angles $\phi_{0.1}^{\infty} = \pm 90^o$ and $\alpha = -1.5$.
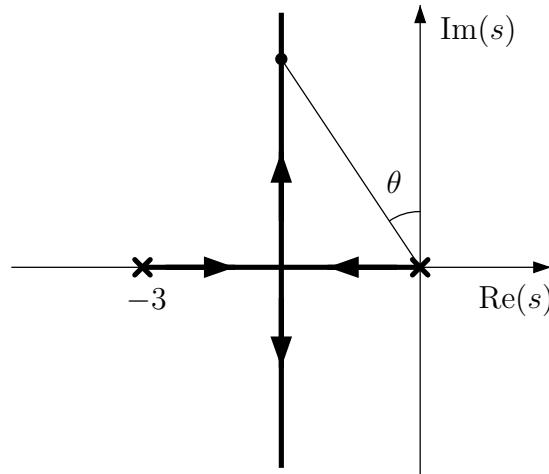
Figure 3.16: Root locus for Example 3.5 with proportional feedback

The problem of using proportional feedback alone was already discussed in Section 2.2: to make the steady state error due to a constant disturbance small requires a large gain, but a large gain results in a low damping ratio with large overshoot. Adding derivative feedback leads to the PD control law

$$C(s) = K_P(1 + T_D s),$$

and we have

$$C(s)G(s) = K_P \frac{10(1 + T_D s)}{s(s + 3)} = KL(s)$$

where again the controller gain K_P is taken as the root locus parameter K.

Adding derivative feedback is equivalent to adding a zero to the loop transfer function. The zero is located at $s = -1/T_D$, where T_D is a parameter to be chosen by the designer. We know from Example 3.3 in this section that a zero to the left of two poles will cause the two root locus branches in Figure 3.13 to form a circle centered at the zero, thus

pulling the root locus to the left and allowing large gains with a high damping ratio. We choose (somewhat arbitrarily) to place a zero at $s = -6$, this corresponds to the choice $T_D = 1/6$. The resulting root locus is shown in Figure 3.17, the breakaway and break-in points are

$$s_1 = -1.8, \qquad s_2 = -10.2$$
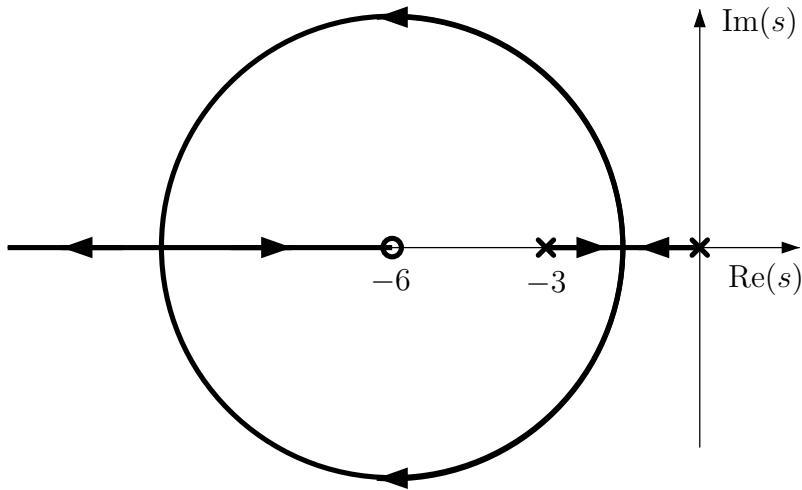
Figure 3.17: Root locus for Example 3.5 with PD control

As discussed in Section 2.2, a PD controller is not physically realizable, and in practice one would replace the term $T_D s$ by the term

$$\frac{T_D s}{1 + \gamma T_D s}$$

where γ is a small positive number. The resulting controller is

$$C(s) = K_P \left(1 + \frac{T_D s}{1 + \gamma T_D s} \right) = K_P \frac{1 + (1 + \gamma) T_D s}{1 + \gamma T_D s}$$

and for $\gamma \ll 1$, we have approximately

$$C(s) = K_P \frac{1 + T_D s}{1 + \gamma T_D s}$$

As seen in Example 3.4, the effect of an additional pole is to push the root locus to the right. To see what happens if the additional pole is not far enough to the left of the zero, i.e. if γ is not small enough, assume the pole is located at $s = -10$ (this corresponds to $\gamma = 0.6$). Applying step 3 yields

$$\phi_{0.1}^{\infty} = \pm 90^o, \qquad \alpha = \frac{-10 - 3 - (-6)}{2} = -3.5$$
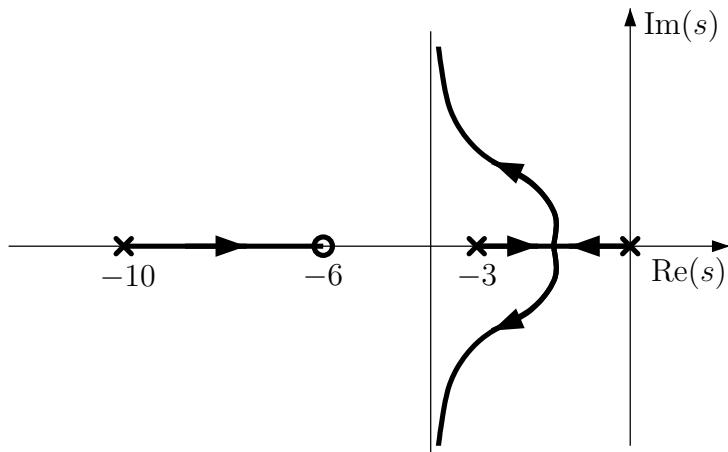
Figure 3.18: Root locus sketch for PD control with additional pole at $s = -10$

and the root locus sketch in Figure 3.18 shows that the desired effect of PD control - allowing a high gain *and* a large damping ratio - is lost. Obviously the problem is that the pole at -10 is too close to the zero. Shifting it to -30, we obtain

$$\alpha = \frac{-30 - 3 - (-6)}{2} = -13.5$$

and shifting it further left to -60 we have

$$\alpha = \frac{-60 - 3 - (-6)}{2} = -28.5$$

Computer generated root locus plots for both case are shown in Figures 3.19 and 3.20, respectively. They show that if the pole is far enough to the left, the root locus is close to that of PD control for a large range of gains. However, placing the pole too far to the left leads to amplification of measurement noise, and the choice of its location represents a trade-off between a good transient response and low noise sensitivity.

Root Locus for Negative Gains (0°-Root Locus)

When root locus plots are generated using a computer, one may find situations where the plots appear to contradict the rules for sketching a root locus introduced in this chapter. This would happen for example if the controller zero in the previous example is pulled into the right half plane. To see the reason for this apparent contradiction, we return to the closed-loop characteristic equation

$$L(s) = -\frac{1}{K}$$

from which the phase condition and the subsequent rules were derived. One important assumption we made so far is that $K > 0$. Moreover, we assumed that in the loop transfer function

$$L(s) = \frac{b_m s^m + \ldots + b_0}{s^n + a_{n-1} s^{n-1} + \ldots + a_0}$$
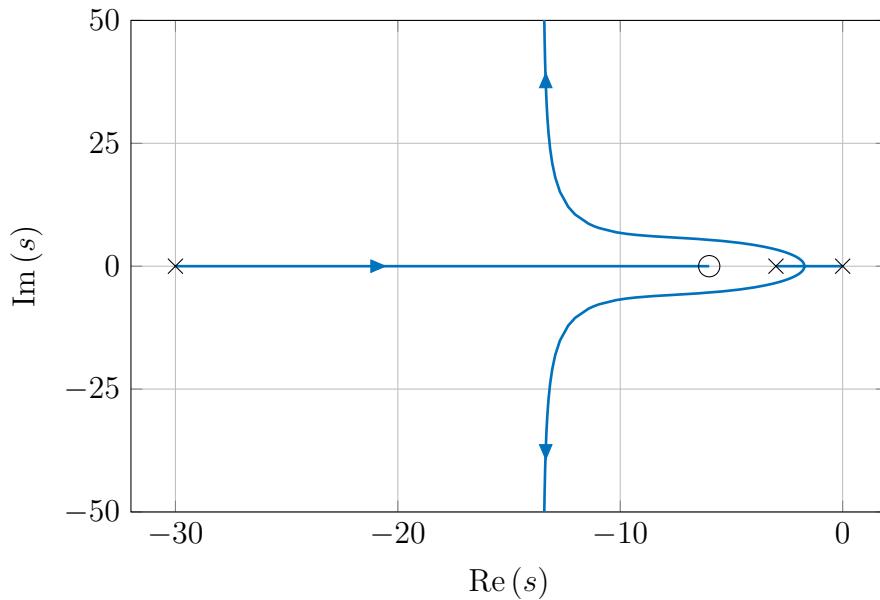
Figure 3.19: Root locus plot for PD control with additional pole at $s = -30$ generated with MATLAB
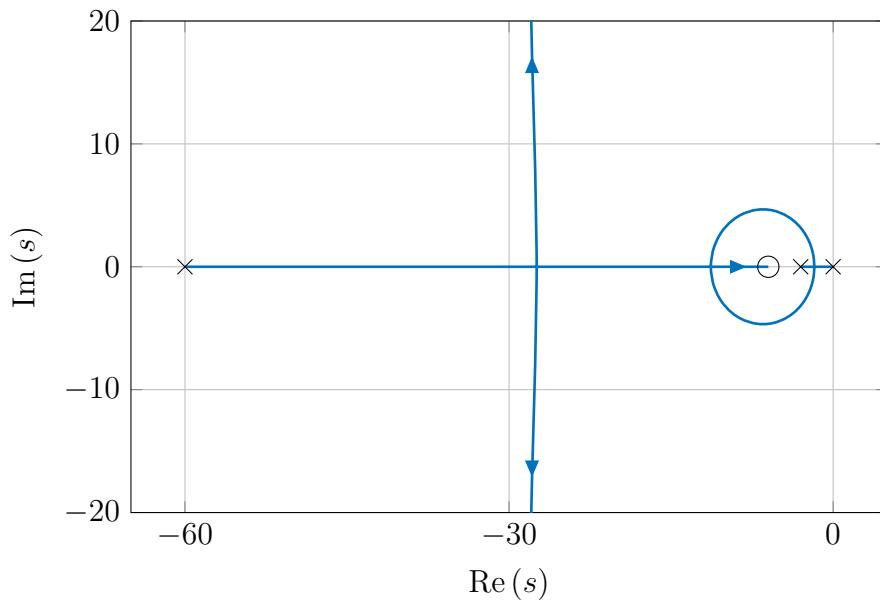
Figure 3.20: Root locus plot for PD control with additional pole at $s = -60$ generated with MATLAB

we have $b_m > 0$. In fact, the loop transfer function can be written as

$$L(s) = \frac{b_m(s - z_1)(s - z_2)\ldots(s - z_m)}{(s - p_1)(s - p_2)\ldots(s - p_n)}$$

and the phase condition for s to be on the root locus is

$$\arg L(s) = \sum_{i=1}^{m} \arg (s - z_i) - \sum_{i=1}^{n} \arg (s - p_i) + \arg b_m = 180^o \pm l360^o, \ l = 0, 1, 2, \ldots$$

If $b_m > 0$ we have $\arg b_m = 0^o$, and this condition is the same as the phase condition used for the construction rules in the previous section. However, if $b_m < 0$ we have $\arg b_m = 180^o$, and the graphical root locus construction must then use the condition

$$\arg \frac{(s - z_1)(s - z_2) \ldots (s - z_m)}{(s - p_1)(s - p_2) \ldots (s - p_n)} = 0^o \pm l360^o, \ l = 0, 1, 2, \ldots$$

The same is true if b_m is positive but $K < 0$. Thus, the rules for constructing a root locus need to be changed in these cases as follows.

Step 2

All points to the left of an *even* number of poles and zeros belong to the root locus (note that 0 is considered even here, i.e. the positive real axis to the right of the rightmost pole or zero is part of the root locus).

Step 3

The angles of the asymptotes for $K \to \infty$ are $\Phi_l^\infty = l\frac{360^o}{n-m}, \ l = 0, 1, \ldots, n - m - 1$.

Step 4

The phase condition for checking the angle of departure is

$$\arg L(s) = \sum_{i=1}^{m} arg(s - z_i) - \sum_{i=1}^{n} arg(s - p_i) = 0^o \pm l360^o, \ l = 0, 1, 2, \ldots$$

These rules must be used when $Kb_m < 0$; the rules presented in the last section apply when $Kb_m > 0$.

3.3 Control of Time Delay Systems

Many practical control systems have a pure time delay in the loop, such cases arise for example in process control applications where the plant may involve the movement of material such as on a conveyer belt or in a pipeline. A typical step response is shown in Figure 3.21: the process is described by a first order system with time constant T_0 and static gain K_0, and by a time delay T_d that passes before any change of the plant output in response to the step input can be observed.

From the properties of the Laplace transform discussed in Section 1.2 we know that the transfer function of this process is

$$G(s) = \frac{K_0}{T_0 s + 1} e^{-T_d s}.$$
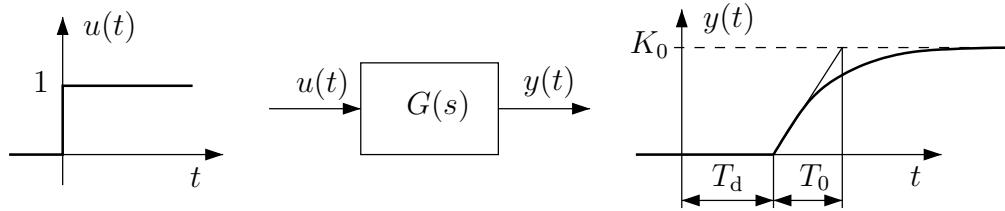
Figure 3.21: Step response of first order system with time delay

We will now investigate the effect of the delay term $e^{-T_d s}$ on the performance of a control loop as shown in Figure 3.22. We will assume that proportional feedback is used, i.e. $C(s) = K_P$, and that $K_0 = T_0 = 1$.
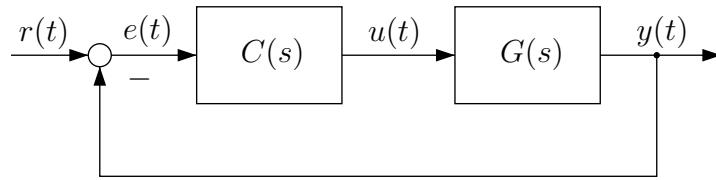
Figure 3.22: Control of a time delay system

To see the effect of the time delay, we will compare the root locus of the feedback system with and without time delay. If $T_d = 0$ the root locus is that of a first order system shown in Figure 3.23.

To construct the root locus plot of the feedback system when $T_d > 0$, we consider the closed-loop characteristic equation

$$1 + K_P \frac{e^{-T_d s}}{s + 1} = 0$$

or

$$\frac{e^{-T_d s}}{s + 1} = -\frac{1}{K_P}$$

where $0 \leq K_P < \infty$. From this we obtain the phase condition

$$\arg e^{-T_d s} - \arg (s + 1) = 180^o \pm l360^o, \quad l = 0, 1, \ldots$$

Substituting $s = \sigma + j\omega$ in the first term on the left hand side yields

$$\arg e^{-T_d(\sigma + j\omega)} = \arg e^{-jT_d\omega} = -T_d\omega \frac{180^o}{\pi}$$

The phase condition for s to be on the root locus is therefore

$$\arg (s + 1) = 180^o \pm l360^o - T_d\omega \frac{180^o}{\pi}, \quad l = 0, 1, \ldots$$

and taking $l = 0$ (the *principal branch* of the root locus) yields

$$\arg (s + 1) = 180^o - T_d\omega \cdot 57^o. \tag{3.4}$$
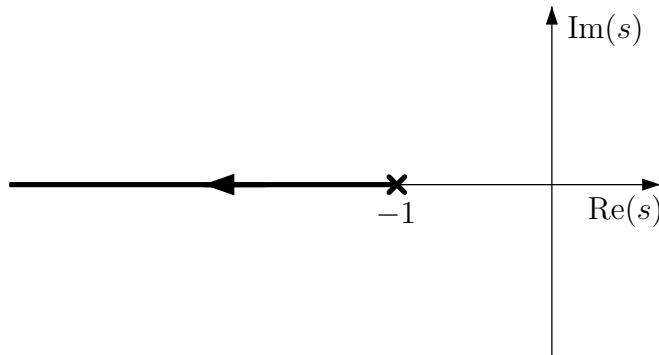
Figure 3.23: Root locus of first order system without delay

To illustrate this result, we construct the root locus for $T_d = 1$. This can be done by applying (3.4) at selected values of ω. First we observe that for $\omega = 0$ we have

$$\arg\left(s + 1\right) = 180^o$$

which is precisely the phase condition of the system without time delay. Thus, the negative real axis ($\omega = 0$) from -1 to $-\infty$ is part of the root locus. A second observation is that if $s \to -\infty$, we have

$$\lim_{s \to -\infty} \frac{e^{-T_d s}}{s + 1} = \lim_{s \to -\infty} \frac{-T_d e^{-T_d s}}{1} = -\infty$$

where l'Hospital's rule has been used. This shows that the delayed system has an open-loop pole at $-\infty$.
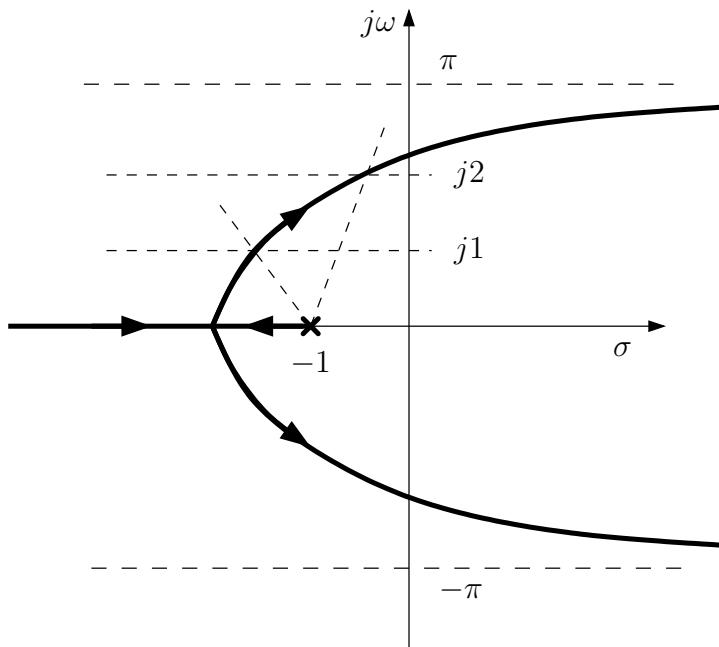
Figure 3.24: Root locus of first order system with delay

For $\omega = 1$ we obtain

$$\arg(s + 1) = 180^o - 57^o = 123^o$$

and for $\omega = 2$

$$\arg(s + 1) = 180^o - 114^o = 66^o$$

Finally, for $\omega \to \pi$ we have

$$\arg(s + 1) = 180^o - \pi \cdot \frac{180^o}{\pi} = 0^o$$

Combining this information, one can sketch the root locus shown in Figure 3.24.

Padé Approximation

For a computer-based generation of root locus plots for time delay systems, the method just presented is not very efficient and a different approach is used. It is based on an approximation of the term $e^{-T_d s}$ by a rational function that was proposed by Padé and is therefore called *Padé approximation*. The idea is to develop both $e^{-T_d s}$ and the rational function into a series and match a suitable number of terms. We have

$$e^{-T_d s} = 1 - T_d s + \frac{(T_d s)^2}{2!} - \frac{(T_d s)^3}{3!} + \ldots$$

and choosing a first order rational function

$$\frac{\beta_1 T_d s + \beta_0}{\alpha_1 T_d s + 1} = \beta_0 + (\beta_1 - \beta_0 \alpha_1) T_d s + (\ldots)(T_d s)^2 + \ldots$$

values α_1, β_0 and β_1 can be obtained by comparing the coefficients of both series. Because the rational function has polynomials of degree one, this is a first order Padé Approximation. If the polynomials have degree n the Padé Approximation is said to be of order n.

Smith Predictor

One way of designing a controller for a time delay system is suggested in Figure 3.25. This control scheme was proposed by O.J.M. Smith in 1958 and is known as a *Smith predictor*.

A controller $C_0(s)$ is first designed for a plant model $G_0(s)$ without time delay. The controller $C(s)$ that is actually implemented is then obtained by feedback around $C_0(s)$ as shown inside the dashed box in Figure 3.25. What makes a Smith predictor useful is the fact that the closed-loop transfer function from r to y is

$$\frac{Y(s)}{R(s)} = \frac{C_0(s)G_0(s)}{1 + C_0(s)G_0(s)} e^{-T_d s}.$$

This is shown in Exercise 3.8. The right hand side is the closed-loop transfer function from r to y of the system shown in Figure 3.26, which is therefore an equivalent representation of the closed-loop system shown in Figure 3.25. The controller $C_0(s)$ can be designed as if there was no time delay, using standard methods.
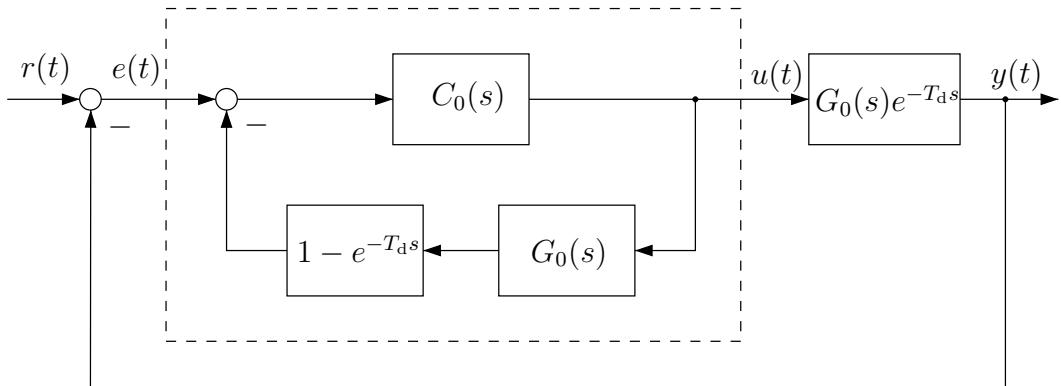
Figure 3.25: Block diagram of a feedback loop with Smith predictor
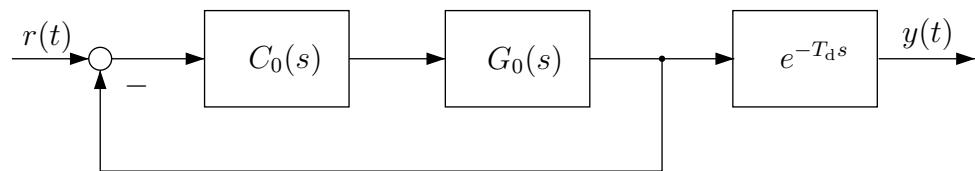
Figure 3.26: Equivalent representation of the feedback system in Figure 3.25

Note that the inner feedback loop in Figure 3.25 is feeding back the simulated response of the delayed system in order to cancel the actual delayed plant response. The controller $C_0(s)$ sees only the simulated response of the plant without time delay, which is also fed back in the inner loop. It is clear that this strategy depends heavily on the accuracy of the plant model. Implementing a Smith predictor requires the simulation of the time delay. This can be easily done if the controller is implemented digitally; if it is implemented on analog hardware a Padé approximation has to be used.

Exercises for Chapter 3

Exercises

Problem 3.1

Sketch the root locus for $K_P > 0$ of a feedback loop with open loop transfer function

$$L(s) = K_P \frac{s+4}{s^3 + 8s^2 + 7s}$$

Problem 3.2

The transfer functions of the plant and the controller in a feedback loop are given by

$$G(s) = \frac{1}{(s+1)(s^2 + 4s + 5)}, \quad C(s) = K_P \frac{s+2}{s+3}$$

a) Sketch the root locus for $K_P > 0$. Calculate the angles of departure.

b) Using your result from (a), show that the closed loop is not stable for all $K_P > 0$.

c) Where can a real zero be placed to stabilize the closed loop system for all $K_P > 0$?

d) Sketch the root locus for $K_P < 0$. Calculate the angles of departure.

Problem 3.3

Consider the control loop shown in Figure 3.27. A valve with transfer function

$$G(s) = \frac{3}{s(s+1)}$$

is to be controlled using a controller with transfer function

$$C(s) = K_P \left(1 + \frac{T_D s}{1 + \gamma T_D s} \right) .$$

a) What kind of controller is $C(s)$?
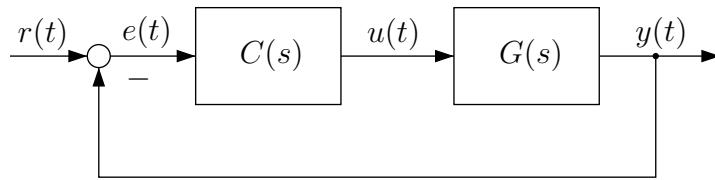
Figure 3.27: Closed loop

b) The given form of the controller $C(s)$ is not practical for the root locus design. Therefore bring it in the form

$$C(s) = K \left(\frac{s - z}{s - p} \right) .$$

c) Sketch the root locus for $\gamma = 0.5$ and $T_D = 0.5$
Assume that you want to design a controller that fulfills certain constraints. Can this easily be done by the use of the sketch of the root locus?

Problem 3.4

This is the computer-based extension of Problem 3.3. In the following you will use the MATLAB function `rltool` to design a controller $C(s)$.

a) By tuning the parameters K, z and p, try to achieve the fastest possible rise time, subject to the constraints that in response to a unit step reference input the control signal satisfies $|u(t)| < 1$ at all times, and that the peak overshoot is less than 5%.

b) What is the steady state error with your controller designed in (a), when the reference input to be tracked is a unit ramp $r(t) = t\sigma(t)$? Use the prefilter block F to realize a ramp input from the given step reference.

Hints for the use of the `rltool`:

- Enter the transfer function G of the plant in MATLAB.

- Call `rltool(G)`.

- Right-click the root locus plot → *Edit Compensator* to enter a controller.

- Use *New Plot* → *New Step* in the toolstrip in order to generate a step response.

MATLAB file: `Problem3_4_PDControlledValve.mlx`

Problem 3.5

Given are the system $G(s)$ from problem 3.3 and a possible controller $C_1(s)$:

$$G(s) = \frac{3}{s(s+1)} \qquad\qquad C_1(s) = \frac{s+1.5}{s+4.5}$$

In the following a controller $C_2(s)$ shall be designed which has the form

$$C_2(s) = \frac{s-z}{s-p}$$

and is used in series with the controller $C_1(s)$. The steady state error should reduce to less than or equal to 0.1 in response to a unit ramp input. However, the compensator should be designed such that the shape of the step response achieved with the controller $C_1(s)$ is not changed significantly.

a) Calculate the minimal static gain of the controller $C_2(s)$.

b) Where should the pole and zero of $C_2(s)$ be placed in order to not effect the rest of the root locus plot?

c) Simulate the response to both step and ramp inputs with the help of the provided MATLAB file.

MATLAB file: `Problem3_5_RlocusLeadLag.mlx`

Problem 3.6

This is a complement to Problem 2.8 part (d). Consider the control system shown in Figure 3.28, where disturbances are acting on plant input and output. Assume initially that feedback is used and the transfer function of the plant is given by

$$G(s) = \frac{1}{s(s+1)} \, .$$

In the following you will use the MATLAB function *rltool* to design a controller

$$C(s) = K\frac{s^2 + as + b}{s^2 + p^2}$$

with a sinusoidal disturbance $d_u(t) = sin(\omega_d t)\sigma(t)$ where $r(t) = d_y(t) = 0$ and $\omega_d = 1$.

a) Where should the zeros of $C(s)$ be placed in order to have a stable closed loop system?

   Hint: Sketch the root locus of the system. Note that there are two possibilities for choosing the zeros:
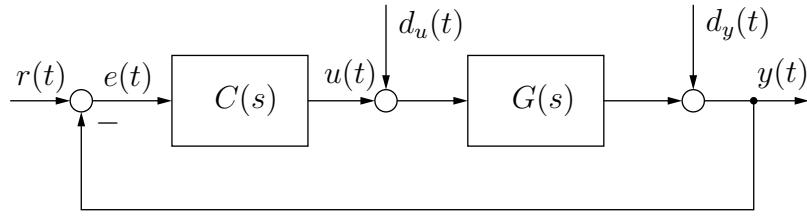
Figure 3.28: Control loop with disturbances

- *The complex pole pair moves to the zeros or*

- *The real pole pair moves to the zeros.*

b) For which values of K are the two cases stable?

c) Now design the controller $C(s)$ by tuning the parameters K, a, b and p. Try to meet the following constraints:

- $\lim_{t \to \infty} e(t) = 0$

- $|e(t)| < 0.05$ for $t \geq 10s$

- $\max |e(t)| < 0.25$.

Hint: Use a SIMULINK model to plot the answer to the sinus disturbance.

SIMULINK file: `Model_Problem3_6_InternalModelPrincipleSin.slx`

Problem 3.7

a) Compute the first order Padé approximation of a pure time delay $e^{-T_d s}$, by comparing the first three coefficients of the series expansions

$$e^{-x} = 1 - \frac{x}{1!} + \frac{x^2}{2!} - \dots$$

$$f(x) = \frac{\beta_1 x + \beta_0}{\alpha_1 x + 1} = f(0) + \frac{f'(0)}{1!} x + \frac{f''(0)}{2!} x^2 + \dots$$

b) Consider a plant that can be modelled as a first order system with time delay

$$G(s) = \frac{K_0}{T_0 s + 1} e^{-T_d s}.$$

For $K_0 = T_0 = 1$, compare the step responses of the system when the following approximations are used:

- $e^{-T_d s} \approx 1 - T_d s$

- $e^{-T_d s} \approx 1/(1 + T_d s)$

      –   first order Padé approximation

      –   second order Padé approximation

      –   third order Padé approximation

For $T_d = 0.1$, Use MATLAB to plot all step responses obtained with these approximations together with the exact step response in one diagram. Repeat this for $T_d = 0.2,\ 0.5,\ 1$.

c*)   For the time delays $T_d = 0.1$ and $T_d = 1$, compare the root locus plots resulting from the different approximations in (b) when proportional feedback is used to control the system. There is no MATLAB command for plotting the exact root locus; use the phase condition for $s = \sigma + j\omega$, $\omega = 0.5,\ 1.0,\ 1.5,\ 2.0$ instead to check the accuracy of the approximated root locus plots.

MATLAB file: `Problem3_7_TimeDelayPadeCompare.mlx`

Problem 3.8

a)   Given the block diagramm of a feedback loop with Smith predictor in Figure 3.29.

    i)   Calculate the transfer function $G_{yr}(s)$ from r to y. The dashed box can be taken as the controller $C(s)$.

    ii)   Calculate the transfer function $C(s) = G_{ue}(s)$ from e to u.

    iii)   Simplify the transfer function $G_{yr}(s)$ from r to y with your result from part ii).

b)   Simplify the given block diagramm in Figure 3.29 with your results from part iii).
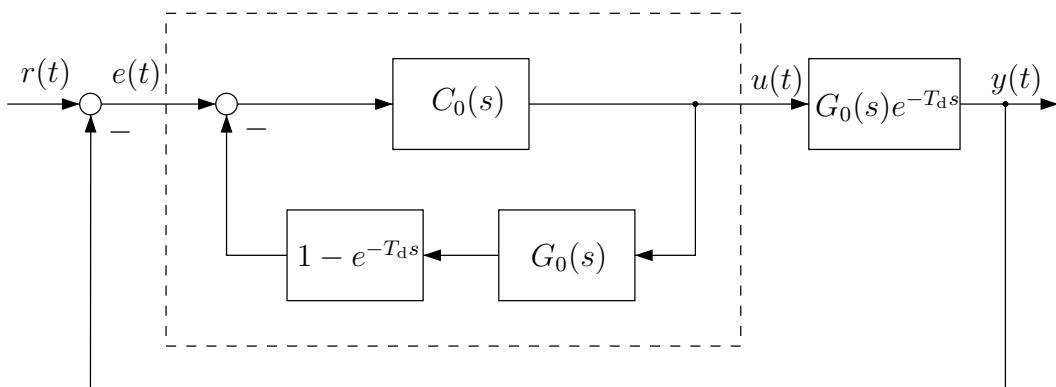
Figure 3.29: Block diagram of a feedback loop with Smith predictor

Problem 3.9

A plant with transfer function

$$G(s) = \frac{1}{s+1} e^{-s}$$

is to be controlled. The design objectives are

- fast tracking of step changes of the reference input with zero steady state error

- peak overshoot less than 5%

- $|u(t)| < 5$ at all times.

a) Use `rltool` to design a controller of the form

$$C_0(s) = K_P \frac{s-z}{s}$$

that meets the design specifications for the plant model *without time delay.*

b) Simulate the response of the system *with time delay* when the controller obtained in (a) is used. For that purpose, use a second order Padé approximation. Try to modify the controller with the help of `rltool` such that the specifications are met. Repeat your design with a 4th order Padé approximation.

   Hint: See Exercise 3.7 for further detail on the Padé approximation.

c) Use the controller obtained in (a) together with a Smith predictor to control the plant, and simulate the step response. Compare the results with the one of (b).

MATLAB file: `Problem3_9_RLToolSmithPredictor.mlx`

Problem 3.10

Consider again the control problem of Exercise 3.9. Assume now that a step disturbance is acting on the plant input, as shown in Figure 3.30.

a) Simulate the response of the closed-loop system with the controller of Exercise 3.9.c, when a step disturbance $d(t) = \sigma(t-5)$ and a unit step reference input $r(t) = \sigma(t)$ are applied. Compare the speed with which the disturbance is suppressed, with that of tracking the reference step input.

b) Simulate the closed-loop response to a reference unit step $r(t) = \sigma(t)$ (assume $d(t) = 0$) when the actual time delay of the plant is $T_d = 0.8$, but the Smith predictor is designed for $T_d = 1$.

c*) Compute the closed-loop transfer function $G_d(s) = Y(s)/D(s)$ from disturbance input to controlled output. Explain the effect observed in a).

MATLAB file: `Problem3.10_RLToolSmithPredictorDist.mlx`
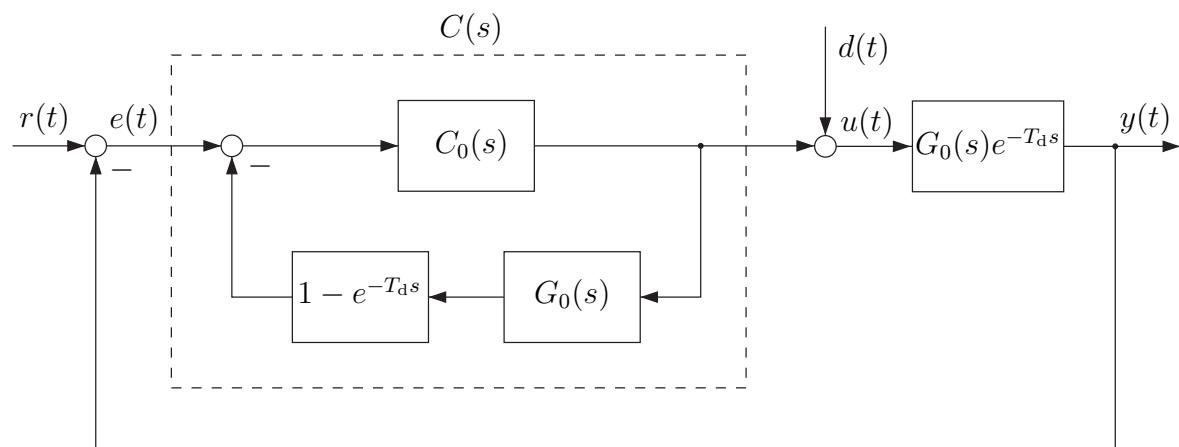
Figure 3.30: Control loop with Smith predictor

Chapter 4

Frequency Response Design

4.1 The Frequency Response

The methods most widely used for the design of feedback systems are based on the frequency response of the plant. The frequency response is the steady state response to sinusoidal inputs over a range of frequencies. It can be obtained experimentally, and it can be used for controller design without explicit knowledge of the transfer function of the plant.

Consider a plant with transfer function $G(s) = b(s)/a(s)$. We assume that $G(s)$ has no poles in the right half plane, and that we are interested in the steady state response to sinusoidal inputs

$$u(t) = \sin \omega t \sigma(t).$$

The Laplace transform of this input is

$$U(s) = \frac{\omega}{s^2 + \omega^2} = \frac{\omega}{(s + j\omega)(s - j\omega)}.$$

Therefore

$$Y(s) = G(s)U(s) = \frac{b(s)}{(s - p_1)(s - p_2)\ldots(s - p_n)} \cdot \frac{\omega}{(s + j\omega)(s - j\omega)}$$

where p_i are the plant poles. Expanding the right hand side yields

$$Y(s) = \frac{A}{s + j\omega} + \frac{\bar{A}}{s - j\omega} + \frac{B_1}{s - p_1} + \ldots + \frac{B_n}{s - p_n}$$

and in time domain we have

$$y(t) = Ae^{-j\omega t} + \bar{A}e^{j\omega t} + B_1 e^{p_1 t} + \ldots + B_n e^{p_n t}.$$

Because $G(s)$ is stable, the transient response $B_1 e^{p_1 t} + \ldots + B_n e^{p_n t}$ approaches zero as $t \to \infty$, thus the steady state response is

$$y(t) = Ae^{-j\omega t} + \bar{A}e^{j\omega t}. \tag{4.1}$$

From the partial fraction expansion we can compute A and $\bar{A}$ as

$$A = G(s) \left.\frac{\omega}{s - j\omega}\right|_{s=-j\omega} = -\frac{1}{2j}G(-j\omega)$$

and

$$\bar{A} = G(s) \left.\frac{\omega}{s + j\omega}\right|_{s=j\omega} = \frac{1}{2j}G(j\omega).$$

We can express $G(j\omega)$ in polar form

$$G(j\omega) = |G(j\omega)|e^{j\phi(j\omega)}$$

where

$$\phi(j\omega) = \arg G(j\omega) = \tan^{-1}\frac{\operatorname{Im} G}{\operatorname{Re} G}$$

and we also have

$$G(-j\omega) = |G(j\omega)|e^{-j\phi(j\omega)}.$$

Substituting these in the expressions for A and $\bar{A}$ gives

$$A = -\frac{1}{2j}|G|e^{-j\phi}, \qquad \bar{A} = \frac{1}{2j}|G|e^{j\phi}.$$

From (4.1), the steady state response to a sinusoidal is therefore

$$y(t) = -\frac{1}{2j}|G|e^{-j\phi}e^{-j\omega t} + \frac{1}{2j}|G|e^{j\phi}e^{j\omega t}$$

$$= |G|\frac{e^{j(\omega t+\phi)} - e^{-j(\omega t+\phi)}}{2j}$$

or

$$y(t) = |G(j\omega)|\sin(\omega t + \phi(j\omega)). \tag{4.2}$$

This shows that the steady state response to a sinusoidal input is again a sinusoid of the same frequency, but with the amplitude scaled by a gain factor $|G(j\omega)|$, and with a phase shift of $\phi(j\omega)$. If the phase shift is positive, it is called a *phase lead*, and if it is negative a *phase lag*. The significance of the magnitude $|G(j\omega)|$ and the phase $\phi(j\omega)$ of the frequency response at a given frequency ω is shown in Figure 4.1.

Example 4.1

Consider the plant shown in Figure 4.2. Substituting $j\omega$ for s we have

$$G(j\omega) = \frac{1}{j\omega + 1}.$$

Rewriting $G(j\omega)$ as

$$G(j\omega) = \frac{1 - j\omega}{1 + \omega^2} \quad \Rightarrow \quad \operatorname{Re} G = \frac{1}{1 + \omega^2}, \quad \operatorname{Im} G = -\frac{\omega}{1 + \omega^2}$$
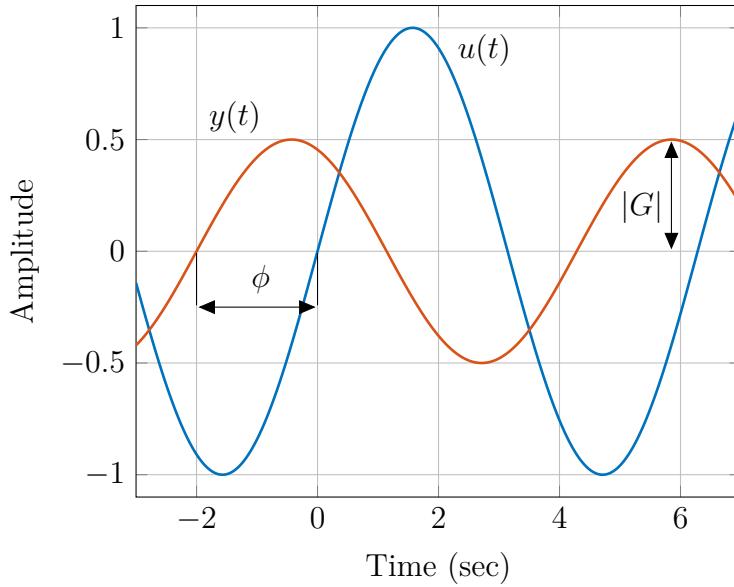
Figure 4.1: Magnitude and phase of the frequency response
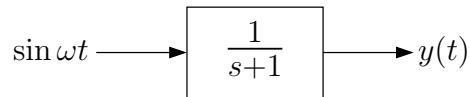
Figure 4.2: Response to sinusoidal inputs, Example 4.1

we find magnitude and phase

$$|G(j\omega)| = \frac{1}{\sqrt{1 + \omega^2}}, \qquad \phi(j\omega) = -\tan^{-1}\omega.$$

The steady state response is therefore

$$y(t) = \frac{1}{\sqrt{1 + \omega^2}} \sin(\omega t - \tan^{-1}\omega). \tag{4.3}$$

This equation gives amplitude and phase of the output signal for each frequency ω. For very high and very low frequencies, the following approximations can be made. If $\omega \ll 1$, the above expressions for magnitude and phase simplify to

$$|G(j\omega)| \approx 1, \qquad \phi(j\omega) \approx 0^o$$

and if $\omega \gg 1$, we have

$$|G(j\omega)| \approx \frac{1}{\omega}, \qquad \phi(j\omega) \approx -90^0.$$

For $\omega = 1$, the exact values are

$$|G(j\omega)| = \frac{1}{\sqrt{2}} = 0.71, \qquad \phi(j\omega) = -45^o.$$

For low frequencies, sinusoidal inputs pass the system almost unchanged, whereas at high frequencies the amplitude decreases proportional to $1/\omega$ and the output has a phase shift of -90°. This is typical first order low pass behaviour .

Bode Diagram

The information about the frequency response contained in (4.3) can be displayed graphically. The most widely used graphical representation of a frequency response is the *Bode diagram*, developed in the Bell laboratories in the 1930s by H. Bode. Magnitude and phase are plotted versus frequency in two separate plots, where a log scale is used for magnitude and frequency and a linear scale for the phase. The log scale is useful because the transfer function is composed of pole and zero factors which can be added graphically: assume for example

$$G(j\omega) = \frac{g_1(j\omega)g_2(j\omega)}{g_3(j\omega)}.$$

Using the notation

$$g_i = |g_i|e^{j\varphi_i}$$

this can be written as

$$G = |G|e^{j\phi} = \frac{|g_1||g_2|}{|g_3|}\, e^{j(\varphi_1 + \varphi_2 - \varphi_3)}$$

thus

$$|G| = \frac{|g_1||g_2|}{|g_3|}, \qquad \phi = \varphi_1 + \varphi_2 - \varphi_3.$$
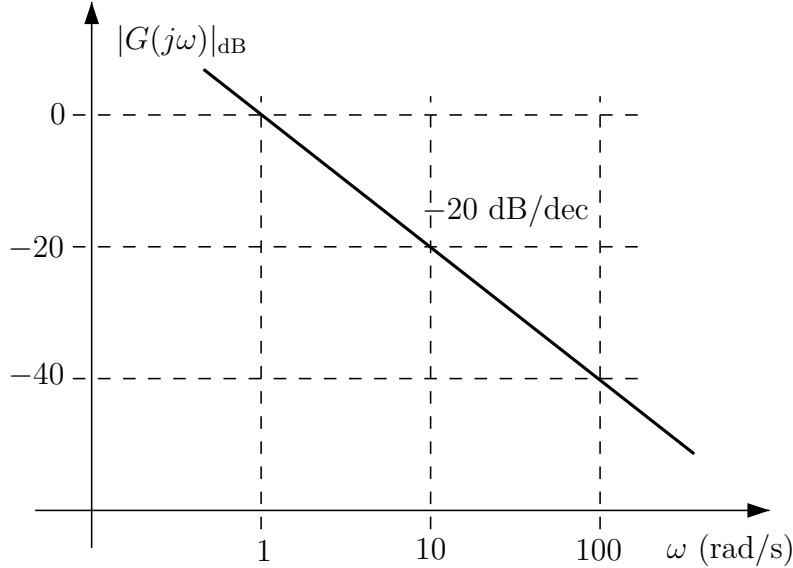
Figure 4.3: Slope of -20dB

The phase of G is the sum of the phase angles of the factors (with a negative sign for pole factors), and on a log scale we also have

$$\log|G| = \log|g_1| + \log|g_2| - \log|g_3|.$$

It is standard to measure the logarithmic gain $\log |G|$ in dB, the definition is

$$|G|_{dB} = 20 \log |G|.$$

For the above example, the gain approximations expressed in dB are

Frequency	$\omega \ll 1$	$\omega = 1$	$\omega \gg 1$
Gain	0 dB	-3 dB	-20 dB/dec

The advantage of plotting the frequency in a log scale is that the decreasing gain at high frequencies appears as a straight line - in the example above with a slope of -20dB/dec; this follows from

$$|G| = \frac{1}{\omega} \quad \Rightarrow \quad \log |G| = -\log \omega.$$

See Figure 4.3. A Bode plot of this system is shown in Figure 4.4.
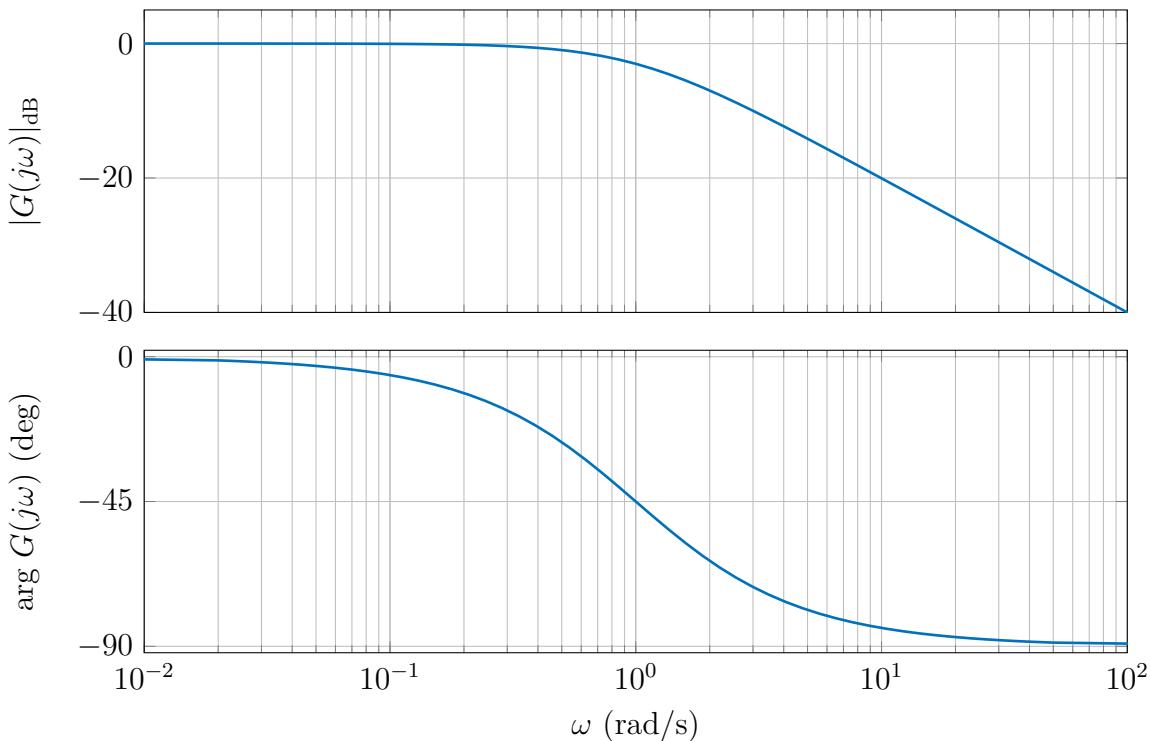
Figure 4.4: Frequency response

The approximation outlined above can be used to sketch the bode plot by hand. The frequency at which the real part and the imaginary part of the pole factor $j\omega + 1$ are equal (i.e. $\omega = 1$), is called the *corner frequency* . Below the corner frequency, the magnitude is approximately 0dB and the phase 0°; above the corner frequency the magnitude falls off at a rate of -20dB/dec and the phase is approximately -90°. Close to the corner frequency, this approximation is not accurate, but for the magnitude the inaccuracy is

at most 3dB. For the phase plot one can draw a tangent from 0° at a fifth of the corner frequency to -90° at five times the corner frequency to achieve a more accurate sketch (see Figure 4.5).
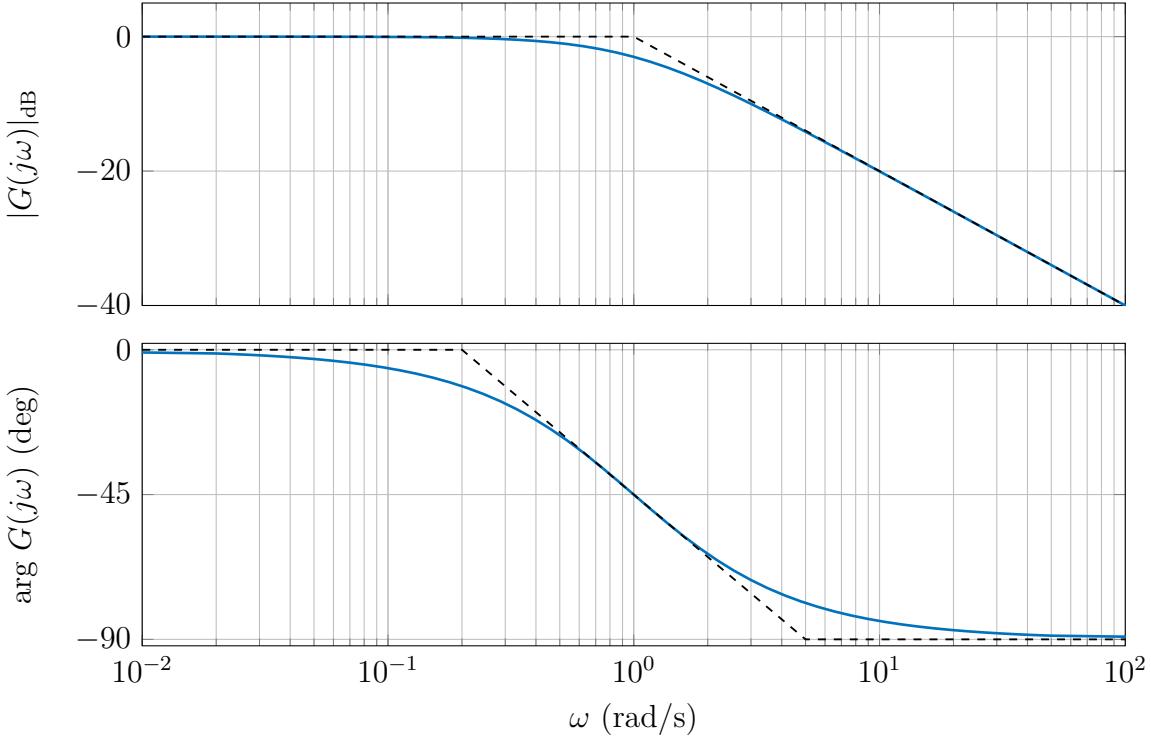
Figure 4.5: Asymptotic approximation of the Bode plot

The log scale for magnitude and frequency makes it easy to sketch a Bode plot for transfer functions with several poles and zeros: one can use the above method to sketch magnitude and phase for each pole and zero, and simply add the curves.

Example 4.2

Consider the transfer function

$$G(s) = \frac{s + 100}{s + 1}.$$

Substituting $j\omega$ for s and normalizing the real part of the zero factor to 1 gives

$$G(j\omega) = \frac{j\omega + 100}{j\omega + 1} = 100 \, \frac{j\frac{\omega}{100} + 1}{j\omega + 1}.$$

The magnitude in dB is

$$|G|_{dB} = 20 \log 100 + 20 \log \left| j\frac{\omega}{100} + 1 \right| - 20 \log |j\omega + 1|.$$

The first term - the constant gain - is 40dB and independent of the frequency. For the second and the third term, the corner frequencies are $\omega = 100$ and $\omega = 1$, respectively. For

the above approximation, the frequency ranges below, above and between the two corner frequencies have to be considered. The approximate values in these frequency ranges are given in the following table:

Frequency	$20\log 100$	$20\log\left\|j\frac{\omega}{100}+1\right\|$	$-20\log\left\|j\omega+1\right\|$
$\omega \ll 1$	40dB	0dB	0dB
$100 \gg \omega \gg 1$	40dB	0dB	[-20dB/dec]
$\omega \gg 100$	40dB	[+20dB/dec]	[-20dB/dec]

Here values in square brackets indicate the slope of the magnitude curve. For the phase, we have

$$\arg G = \arg\left(j\frac{\omega}{100}+1\right) - \arg(j\omega+1)$$

Note that the phase of the constant gain is zero. The approximate values are given in the following table:

Frequency	$\arg\left(j\frac{\omega}{100}+1\right)$	$-\arg(j\omega+1)$
$\omega \ll 1$	0°	0°
$100 \gg \omega \gg 1$	0°	-90°
$\omega \gg 100$	90°	-90°

The asymptotic approximations of the three factors of $G(j\omega)$ are shown in Figure 4.6 together with the exact Bode plot (solid curve).

A pole at the origin indicates the presence of an integrator in the loop. The following example illustrates this case.

Example 4.3

Consider the transfer function
$$G(s) = \frac{10}{s(s+10)}.$$

With $s = j\omega$ we have

$$G(j\omega) = \frac{10}{j\omega(j\omega+10)} = \frac{1}{j\omega\left(j\frac{\omega}{10}+1\right)}.$$

The magnitude is

$$|G|_{dB} = -20\log\omega - 20\log\left|j\frac{\omega}{10}+1\right|,$$
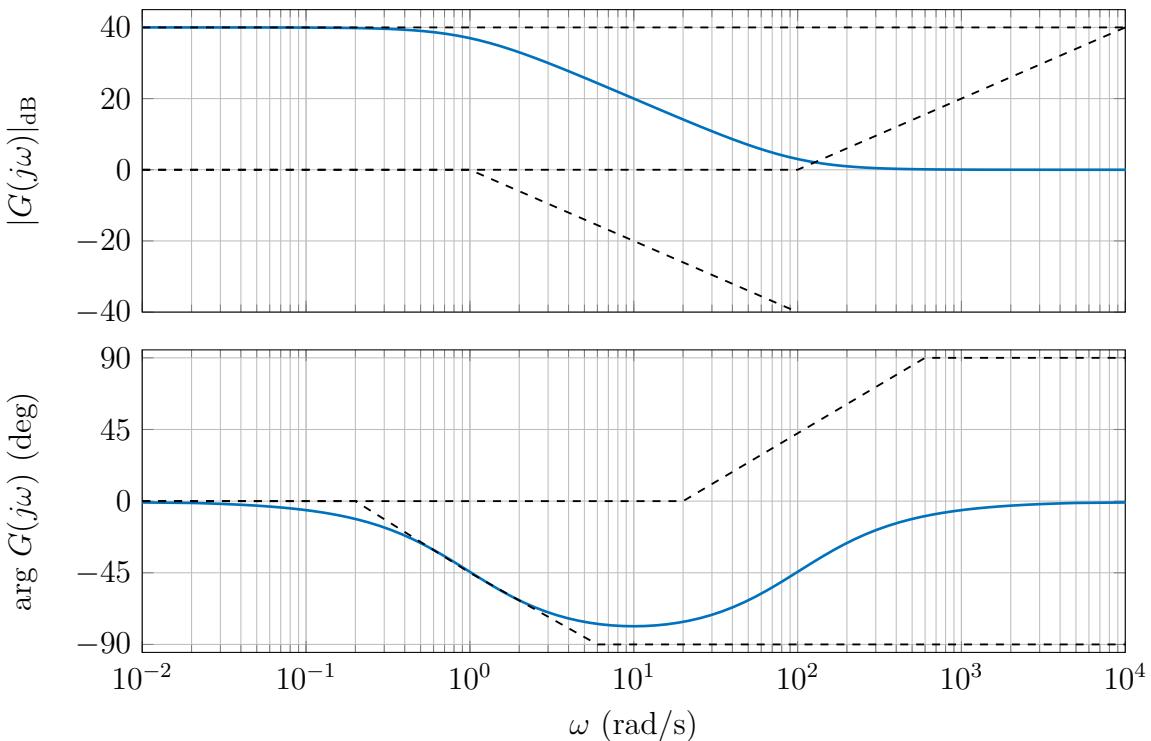
Figure 4.6: Bode plot and asymptotic approximation for Example 4.2

and the phase is

$$\arg G = -\arg(j\omega) - \arg\left(j\frac{\omega}{10} + 1\right).$$

To draw the straight line representing the term $-20\log\omega$ in the magnitude plot, we need a single point on that line: for $\omega = 1$ we obtain 0 dB. The Bode plot of the system together with the asymptotic approximation is shown in Figure 4.7.

System Type

If the transfer function in the above example is the open loop transfer function in a feedback system, then the feedback system is a type 1 system, i.e. it can track step changes in the reference input with zero steady state error. The example illustrates that the system type can be read off the magnitude plot of the open loop transfer function: the presence of a factor $1/s$ is reflected by the slope of -20dB/decade at low frequencies (this factor may be thought of as having its corner frequency at $\omega = 0$). A type 2 system has a factor $1/s^2$ or $1/(j\omega)^2$ with a slope of -40dB/decade at low frequencies, because

$$20\log\frac{1}{\omega^2} = -40\log\omega.$$
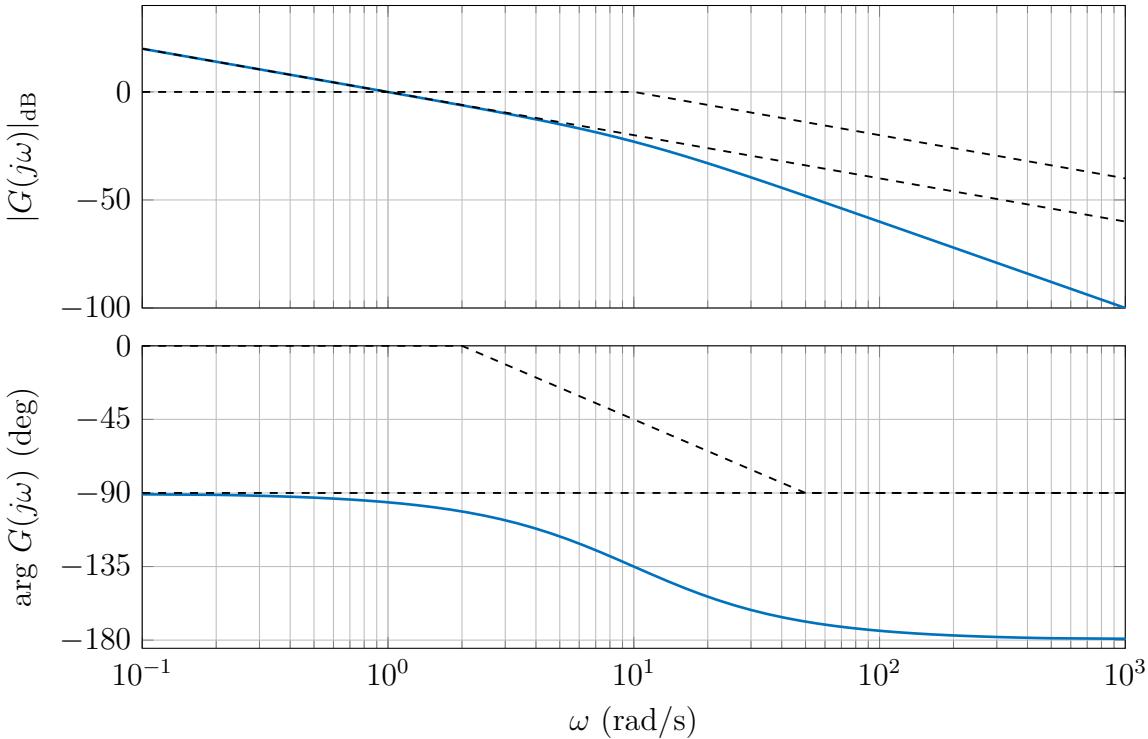
Figure 4.7: Bode plot and asymptotic approximation for Example 4.3

The fact that the slope is negative means that the steady state gain is infinite, in contrast to type 0 systems where the gain is constant at low frequencies. In Figure 4.8 magnitude plots are sketched for the type 0 system

$$L(j\omega) = \frac{K}{\left(j\frac{\omega}{\omega_1} + 1\right)\left(j\frac{\omega}{\omega_2} + 1\right)},$$

the type 1 system

$$L(j\omega) = \frac{K}{j\omega\left(j\frac{\omega}{\omega_1} + 1\right)},$$

and the type 2 system

$$L(j\omega) = \frac{K}{(j\omega)^2}.$$

Rules for Sketching a Bode Plot

In all previous examples, the first step of sketching a Bode plot is to rewrite a given transfer function

$$G(s) = \frac{b(s)}{a(s)}$$

as

$$G(s) = K\frac{\left(j\frac{\omega}{\omega_{z1}} + 1\right)\left(j\frac{\omega}{\omega_{z2}} + 1\right)\dots\left(j\frac{\omega}{\omega_{zm}} + 1\right)}{\left(j\frac{\omega}{\omega_{p1}} + 1\right)\left(j\frac{\omega}{\omega_{p2}} + 1\right)\dots\left(j\frac{\omega}{\omega_{pn}} + 1\right)}. \tag{4.4}$$
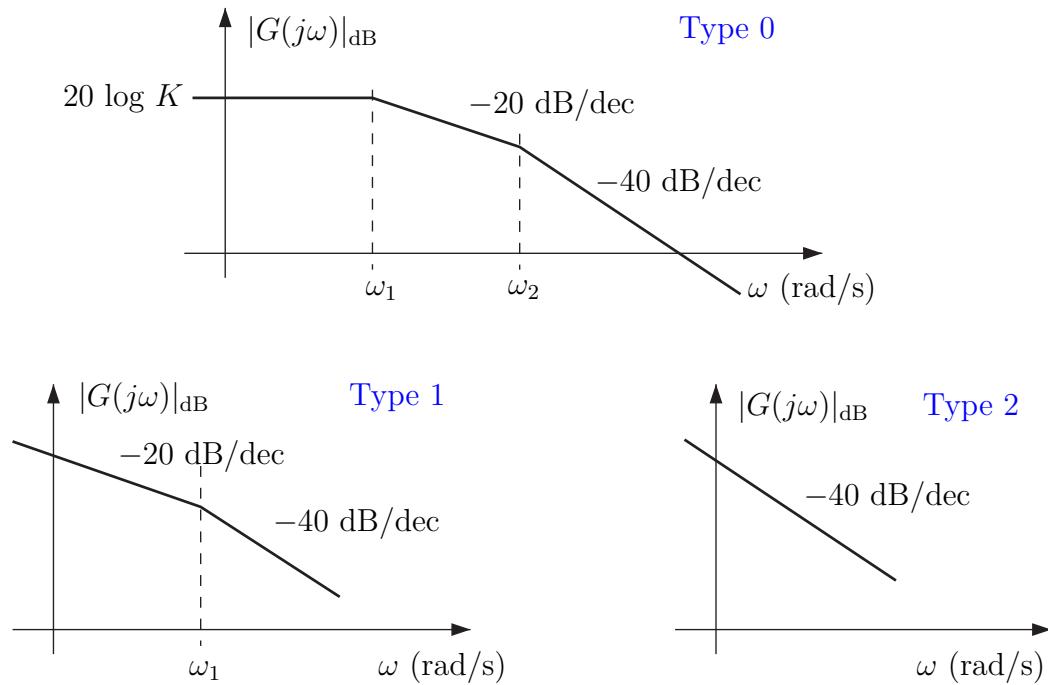
Figure 4.8: Magnitude plots for different system types

If the system has a pole or zero at the origin, a factor $j\omega$ needs to be included in the denominator or numerator, respectively. This form is sometimes referred to as *Bode form*. Note that not every transfer function can be written in the form (4.4); the following assumptions have been made:

- all poles and zeros are in the left half plane or at the origin,

- all poles and zeros are real.

Before we discuss systems with complex or right half plane poles and zeros, we summarize the rules for sketching a Bode plot developed so far. "Sketching" here (and in the exercises) means asymptotic approximations of the magnitude plot, and a rough indication of phase increase or decrease, respectively, at corner frequencies.

1. Rewrite the transfer function in the form (4.4); include factors $j\omega$ if necessary.

2. Sketch magnitude and phase for the low-frequency range. For systems with no pole at the origin, the magnitude at low frequencies is constant and given by K; the low-frequency phase is 0° (assuming $K > 0$, or -180° if K is negative). If the system has k poles at the origin, the magnitude at low frequencies has slope $-k \cdot 20$ dB/dec, and the low-frequency phase is $-k \cdot 90^o$.

3. Mark all corner frequencies ω_{zi} and ω_{pi} on the frequency axis. Draw the magnitude and phase curves moving from low to high frequencies. At a corner frequency

associated with a pole, the magnitude slope drops by 20 dB/dec and the phase by 90°, at a corner frequency of a zero, magnitude and phase increase by the same amounts.

Complex Conjugate Pole Pairs
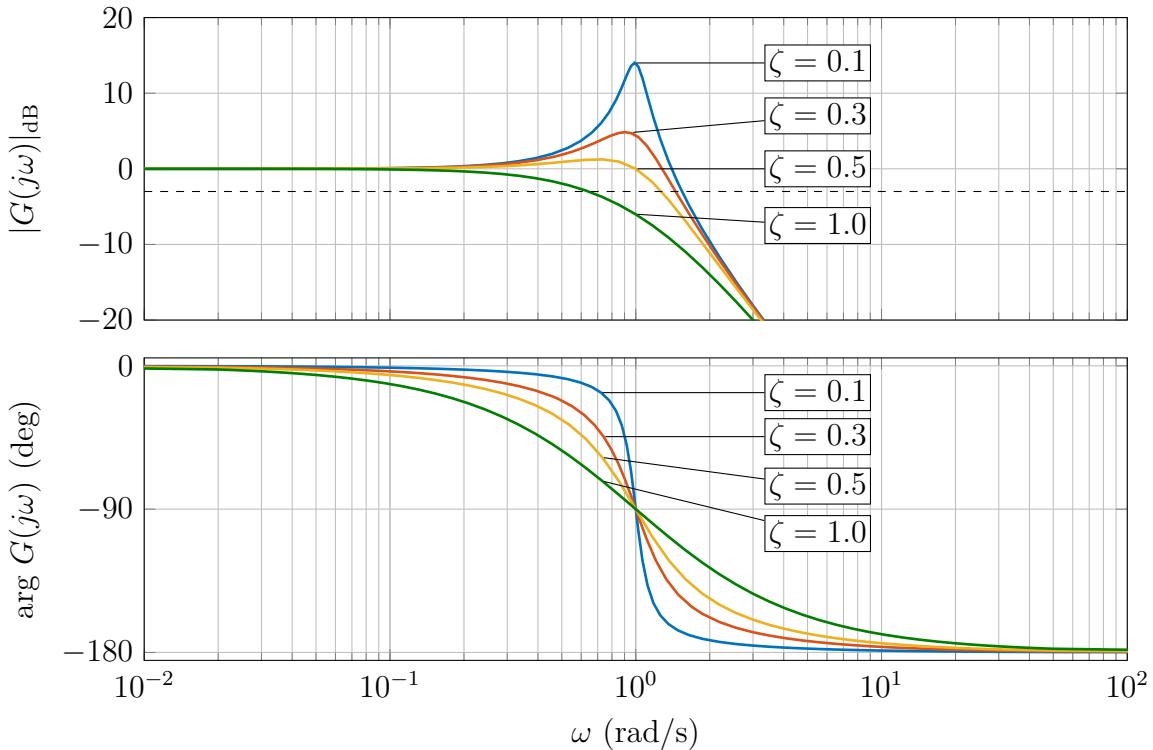
Figure 4.9: Frequency response of second order system with different damping ratios

The previous examples show how to sketch a Bode plot for a transfer function by adding together asymptotic approximations of the pole and zero factors. However, it was assumed so far that the poles and zeros are real. To investigate the frequency response of a system with a complex conjugate pole pair, consider the second order system

$$G(s) = \frac{K}{(s+p)(s+\bar{p})} = \frac{K}{s^2 + (p+\bar{p})s + p\bar{p}}.$$

In the notation introduced in Section 1.3, this can be written as

$$G(s) = \frac{K'\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

where $\zeta\omega_n = \text{Re}\, p$ and $\omega_n = |p|$. For simplicity, assume $K' = 1$ and $\omega_n = 1$ (see remarks on the general case $\omega_n \neq 1$ in Section 1.3). Substituting $j\omega$ for s we have

$$G(j\omega) = \frac{1}{(j\omega)^2 + j2\zeta\omega + 1}.$$

We can again approximate this function for frequencies below and above the corner frequency $\omega = 1$. For low frequencies ($\omega \ll 1$) we obtain

$$G \approx 1 \quad \Rightarrow \quad |G| \approx 1, \quad \phi \approx 0^o$$

and for high frequencies ($\omega \gg 1$)

$$G \approx -\frac{1}{\omega^2} \quad \Rightarrow \quad |G| \approx \frac{1}{\omega^2}, \quad \phi \approx -180^o.$$

At the corner frequency $\omega = 1$ we have

$$G = \frac{1}{-1 + j2\zeta + 1} = \frac{1}{j2\zeta}$$

therefore

$$|G| = \frac{1}{2\zeta}, \quad \phi = -90^o.$$

This shows that for low damping ratios the frequency response can have large *resonant peaks* at the corner frequency, e.g. a damping ratio of $\zeta = 0.1$ leads to a magnitude of $|G| = 5$. Bode plots of $G(s)$ for different damping ratios are given in Figure 4.9; this plot should be compared with Figure 1.15 which shows the effect of the damping ratio on the step response.

Bandwidth

The first order system in Example 4.1 has a static gain of 0 dB, and at the corner frequency the gain is reduced to -3 dB. For first order systems

$$G(s) = \frac{b_0}{s + a_0}$$
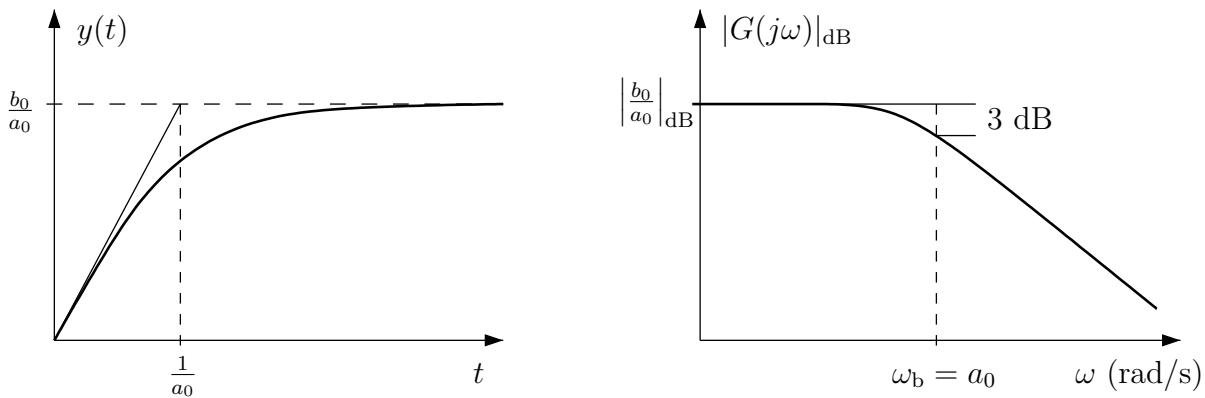
Figure 4.10: Bandwidth of a first order system

it is generally true that the gain at the corner frequency is reduced by 3 dB, compared with the static gain, see Figure 4.10. For systems with finite static gain, the lowest frequency

at which the gain drops 3 dB from its static value is referred to as the *bandwidth ω_b* of the system. The bandwidth is a measure of the systems ability to reproduce input signals without significant distortion. If we let $\tau = 1/a_0$ denote the time constant of the above first order system, we see that the bandwidth is the inverse of the time constant, i.e.

$$\omega_b = \frac{1}{\tau} \tag{4.5}$$

For the second order system shown in Figure 4.9, we see that the bandwidth ω_b (the frequency at which the magnitude curve intersects the -3 dB line) depends on the damping ratio, but lies for all damping ratios shown between 0.5 and 1.5. For a damping ratio of 0.5, the bandwidth is approximately equal to 1. Considering that the natural frequency ω_n of this system is 1, we conclude that for a general second order system

$$G(s) = \frac{K}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

we can take

$$\omega_b \approx \omega_n \tag{4.6}$$

as a reasonable approximation of the bandwidth. Using the estimate (1.29) of the rise time t_r of a second order system, we obtain the approximate relationship

$$\omega_b \approx \frac{1.7}{t_r} \tag{4.7}$$

between rise time and bandwidth.

The above discussion of first and second order systems suggests that the bandwidth is roughly proportional to the speed of response of a system. This is also true for higher order systems; in particular for systems that are dominated by a complex pole pair the approximate relationship (4.7) will still be valid.

Unstable and Non-Minimum-Phase Systems

All examples discussed so far in this chapter have in common that both the poles and the zeros are located in the left half plane (or at the origin in the case of an integrator). This is a property shared by many, but not all systems that arise in practical control problems. We shall now consider the frequency response of systems that have poles and/or zeros in the right half plane, and we begin with the following definition.

Definition 4.1

A system with transfer function $G(s)$ is called minimum phase if it has no pole or zero in the right half plane.

This definition implies that a minimum-phase system is stable; in addition, there must be no zeros in the right half plane. Conversely, a system is *non-minimum phase* if it has

right half plane zeros or poles or both. Note that for unstable systems, the frequency response is only a theoretical - but very useful - concept; it is not possible in practice to measure it by applying sinusoidal inputs. Moreover, when we evaluate $G(j\omega)$ for an unstable system, we encounter a problem concering the region of convergence of the Laplace transform as discussed in Section 1.2: The transfer function $G(s)$ of a system is the Laplace transform of its impulse response $g(t)$, and if the system has right half plane poles, the region of convergence of the Laplace transform does not include the imaginary axis. It is however possible to extend the region of convergence to include the whole complex plane by a technique known as *analytic continuation*, as shown in Exercise 4.5, thus making it possible to evaluate $G(j\omega)$ for unstable transfer functions.

Two examples will illustrate the difference it makes for the frequency response whether a system is minimum or non-minimum phase.

Example 4.4

As an example of a minimum phase system, we take the system from Example 4.1

$$G(s) = \frac{1}{s+1}, \qquad G(j\omega) = \frac{1}{j\omega+1}.$$

The asymptotic approximation for low and high frequencies was discussed earlier and is repeated here for convenience

$$\omega \ll 1: \quad G \approx 1 \quad \Rightarrow \quad |G| \approx 1, \quad \phi \approx 0^o$$

and

$$\omega \gg 1: \quad G \approx \frac{1}{j\omega} = -j\frac{1}{\omega} \quad \Rightarrow \quad |G| \approx \frac{1}{\omega}, \quad \phi \approx -90^o.$$

Example 4.5

As an example of a system that does not have the minimum-phase property, we take the system with transfer function

$$G(s) = \frac{1}{s-1}, \qquad G(j\omega) = \frac{1}{j\omega-1}$$

At low frequencies we have

$$\omega \ll 1: \quad G \approx -1 \quad \Rightarrow \quad |G| \approx 1, \quad \phi \approx -180^o$$

and at high frequencies

$$\omega \gg 1: \quad G \approx \frac{1}{j\omega} = -j\frac{1}{\omega} \quad \Rightarrow \quad |G| \approx \frac{1}{\omega}, \quad \phi \approx -90^o.$$
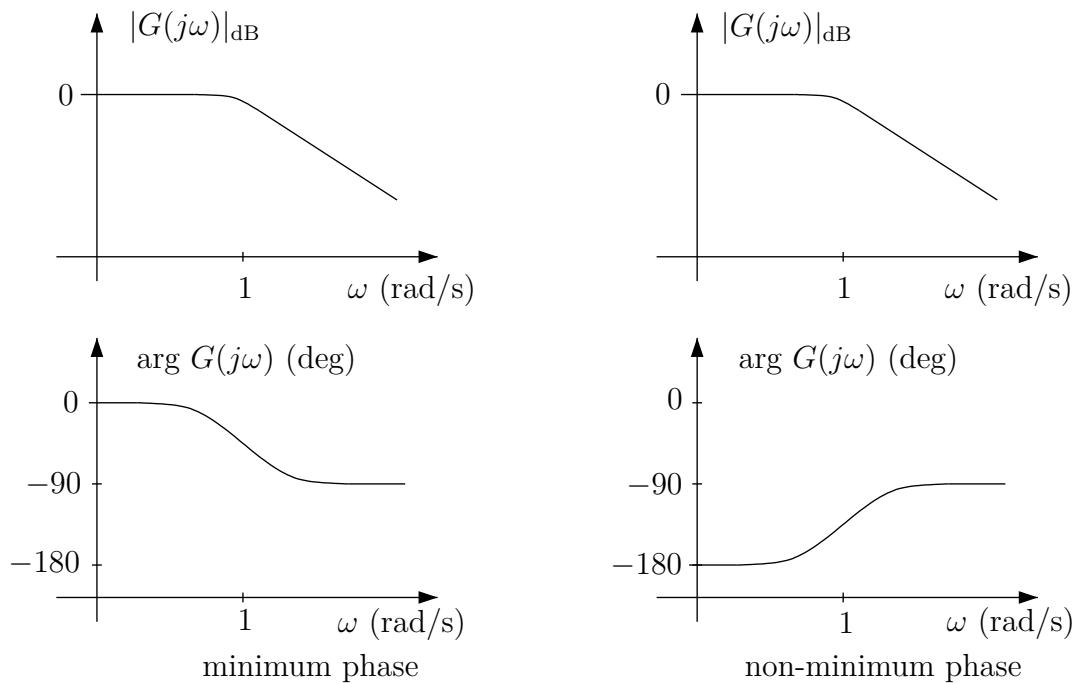
Figure 4.11: Minimum and non-minimum phase system

Both frequency responses are shown in Figure 4.11. The magnitude plots are the same, but the phase of the non-minimum phase system is below that of the minimum phase system at low frequencies.

Bode's Gain-Phase Theorem

For minimum phase systems, Bode showed that the phase of the frequency response is uniquely determined by its magnitude. A simplified version of this result - the form in which it is used for practical applications - is this:

> If the magnitude plot has a constant slope of $n \cdot 20$ dB/dec in a given frequency range (constant means for practical purposes more than a decade away from a corner frequency), then the phase angle in that frequency range is $\phi = n \cdot 90^o$.

Minimum phase systems derive their name from the fact that the phase lag given by the above rule is the smallest possible phase lag. The phase of a non-minimum phase system is in some frequency range below that of the minimum phase system with the same magnitude plot (as in Figure 4.11 at low frequencies).

Figure 4.12 shows an example of the phase gain relationship of a minimum phase system.

The above discussion shows that when sketching the Bode plot of a non-minimum phase system, the rules for sketching the magnitude can be applied as before, but the rules for sketching the phase must be modified: at a corner frequency associated with a right half
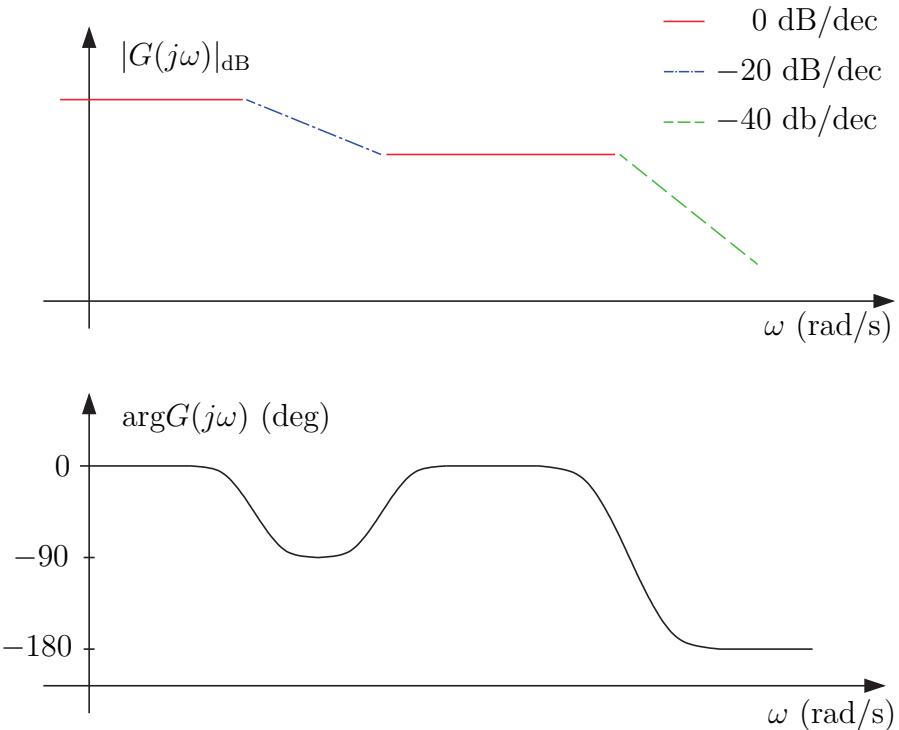
Figure 4.12: Gain and phase of minimum phase system

plane pole, the phase increases by 90°, whereas at the corner frequency of a right half plane zero the phase drops by the same amount. Conversely, when given a Bode plot, the gain phase relationship can be used to detect the presence of right half plane poles or zeros. While a drop of the magnitude slope by 20 dB/dec indicates the presence of a pole, an increase in phase at the same frequency reveals that this pole is in the right half plane. Similarly, while an increase in magnitude slope indicates the presence of a zero, a phase decrease at the same frequency shows that the zero is in the right half plane.

4.2   The Nyquist Stability Criterion

In this section we introduce an important test for closed-loop stability, based on the frequency response of the open-loop system. The method was developed in 1932 by H. Nyquist, who was studying stability problems of feedback amplifiers in the Bell Laboratories. We first present an alternative method of displaying the frequency response of a system, known as *Nyquist plot*.

Nyquist Plots

In a Bode plot, magnitude and phase of the response are shown in two separate diagrams. In a Nyquist plot, the frequency response is displayed in one diagram: the imaginary part is plotted versus the real part of $G(j\omega)$, and the frequency ω is taken as a parameter

that varies from zero to infinity. Even though negative frequencies do not have a physical meaning, we will find it useful to extend the frequency axis and let the parameter ω vary from $-\infty$ to $+\infty$.
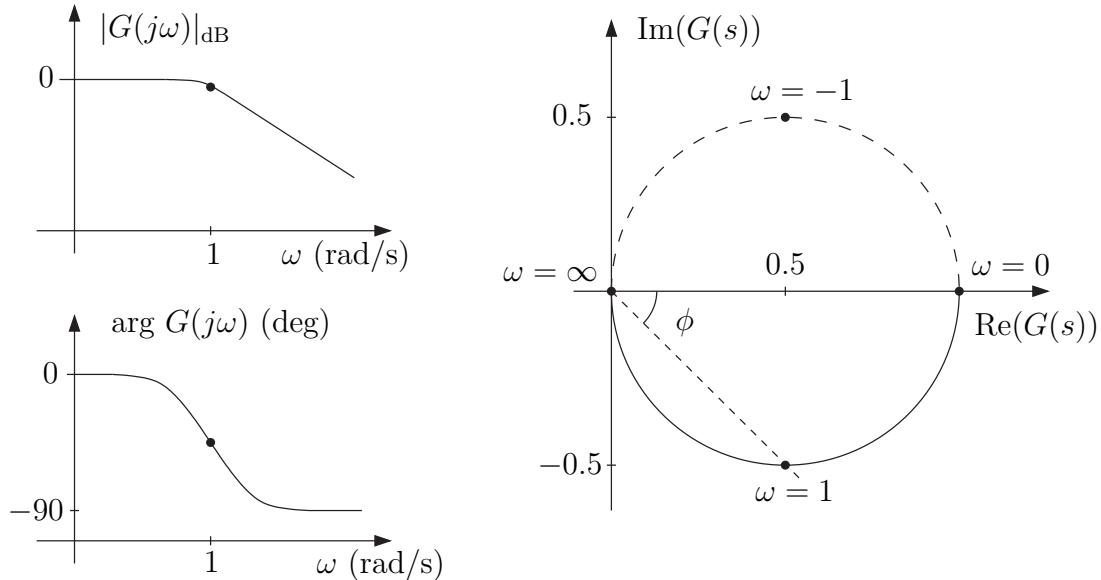
Figure 4.13: Bode and Nyquist plot for Example 4.1

To illustrate the idea of a Nyquist plot, we consider again the transfer function

$$G(s) = \frac{1}{s+1} \quad \text{or} \quad G(j\omega) = \frac{1}{j\omega + 1}.$$

For $\omega = 0$ we have $G = 1$, and for $\omega = \infty$ we have $G = 0$. At the corner frequency $\omega = 1$

$$G = \frac{1}{1+j} = \frac{1-j}{2} = \frac{1}{2} - j\frac{1}{2}.$$

A sketch of the Bode plot of $G(s)$ is shown in the left hand side of Figure 4.13. The magnitude and phase taken by $G(j\omega)$ at the frequencies zero and infinity can be read off the Bode plot as the limits of the magnitude and phase curves infinitely far to the left and right, respectively. The values of the magnitude are 1 (or 0dB) and 0 (or $-\infty$ dB) and the phase angles are 0° and -90°. At the corner frequency, the magnitude is $1/\sqrt{2}$ (-3dB) and the phase is -45°. The complex values at the frequencies 0, 1 and ∞ are marked in the $G(s)$-plane in the right half of Figure 4.13. In this way, magnitude and phase for any frequency can be read off the Bode diagram and plotted as a point in the $G(s)$-plane. Doing this for sufficiently many frequencies reveals that the points are located on a semicircle centered at $1/2$; this semicircular plot is the Nyquist plot of $G(s)$ for positive frequencies.

Once the Nyquist plot for positive frequencies has been constructed, it can be extended to include negative frequencies by adding the mirror image across the real axis to the plot; this follows from

$$G(-j\omega) = \bar{G}(j\omega).$$

The negative frequency part is shown as a dashed curve in Figure 4.13; the complete Nyquist plot is a circle of radius 0.5 centered at $1/2$.
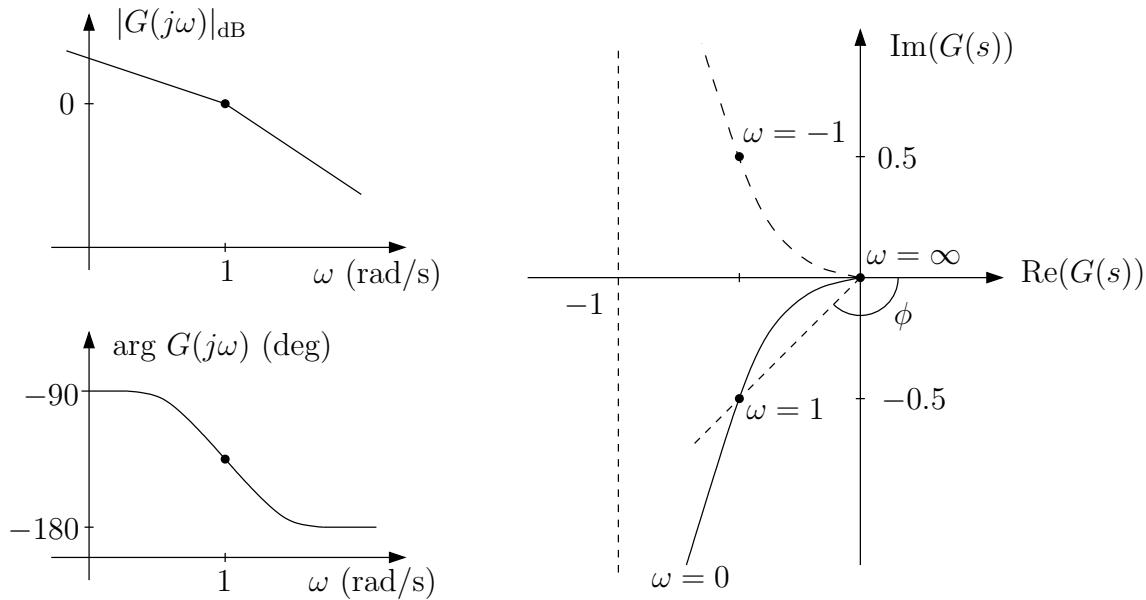
Figure 4.14: Bode and Nyquist plot for Example 4.2

As a second example, consider

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

Evaluated on the imaginary axis

$$G(j\omega) = \frac{1}{j\omega(j\omega+1)} = \frac{1}{-\omega^2 + j\omega}$$

$$= \frac{-\omega^2 - j\omega}{\omega^4 + \omega^2} = -\frac{1}{\omega^2 + 1} - j\frac{1}{\omega(\omega^2 + 1)}$$

We have

$$\omega = 0: \quad G = -1 - j\infty$$
$$\omega = 1: \quad G = -\frac{1}{2} - j\frac{1}{2}$$
$$\omega = \infty: \quad G = 0$$

Bode plot and Nyquist plot are shown in Figure 4.14.

The Nyquist Stability Criterion

The Nyquist stability criterion is based on Cauchy's principle; the idea is to use the contour evaluation of an open-loop transfer function to determine the presence of unstable closed-loop poles.

Assume we are given a transfer function $L(s)$, say

$$L(s) = \frac{s - z}{(s - p_1)(s - p_2)}.$$

The left side of Figure 4.15 shows the s-plane, and the right side the $L(s)$-plane, into which the function $L(s)$ maps its arguments (points in the s-plane). In the s-plane, two poles and a zero are marked, together with a test point s_0 which is moved clockwise along a closed contour C. We are interested in the change of the phase angle of $L(s_0)$ when s_0 is moved around the closed contour C. The phase angle of $L(s_0)$ is the sum of the zero angle and the negative pole angles indicated in the plot. It is clear that after a full traverse, the change of the phase angle is zero. The function $L(s)$ maps each point on C into the
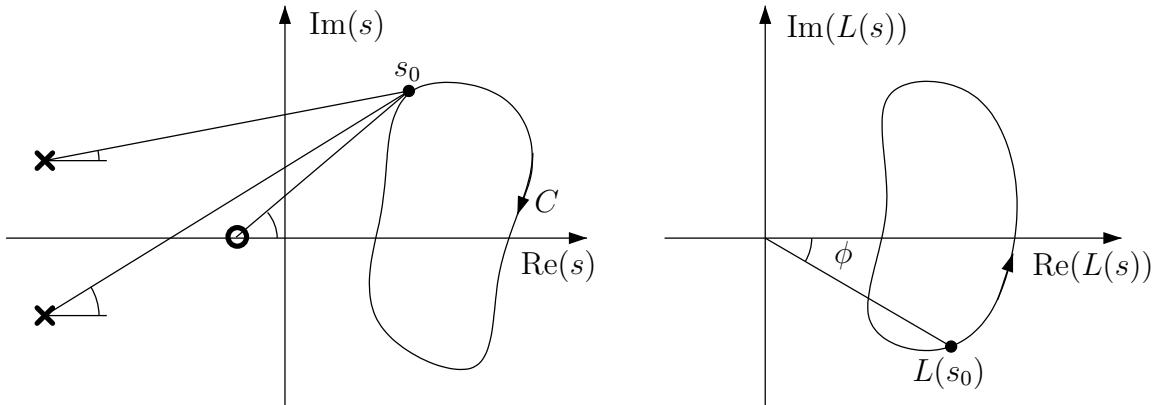
Figure 4.15: Contour evaluation

$L(s)$-plane, and the point $L(s_0)$ moves along a closed contour which is the mapping of C as s_0 moves along C. Such a contour is shown in the right half of Figure 4.15, where ϕ denotes the phase angle of $L(s_0)$. This plot also shows that the change of the phase angle of $L(s_0)$ after a full traverse is zero, i.e. $\Delta\phi = 0^o$.

Figure 4.16 shows a different example, where one of the poles of $L(s)$ is enclosed by C. In this case the phase change of $L(s_0)$ after a full traverse is not zero: the angle of the enclosed pole undergoes a change of 360°. If s_0 moves clockwise, then $L(s_0)$ moves counterclockwise and we have $\Delta\phi = 360^o$. The closed contour into which C is mapped is shown in the right half of Figure 4.16; note that this contour must encircle the origin of the $L(s)$-plane counterclockwise. If instead of a pole a zero is enclosed, $\Delta\phi = -360^o$ and $L(s_0)$ encircles the origin clockwise.

In Figure 4.17 two poles are enclosed, and we have $\Delta\phi = 720^o$ after one traverse along C. Therefore the mapping of C must encircle the origin of the $L(s)$-plane two times counterclockwise.
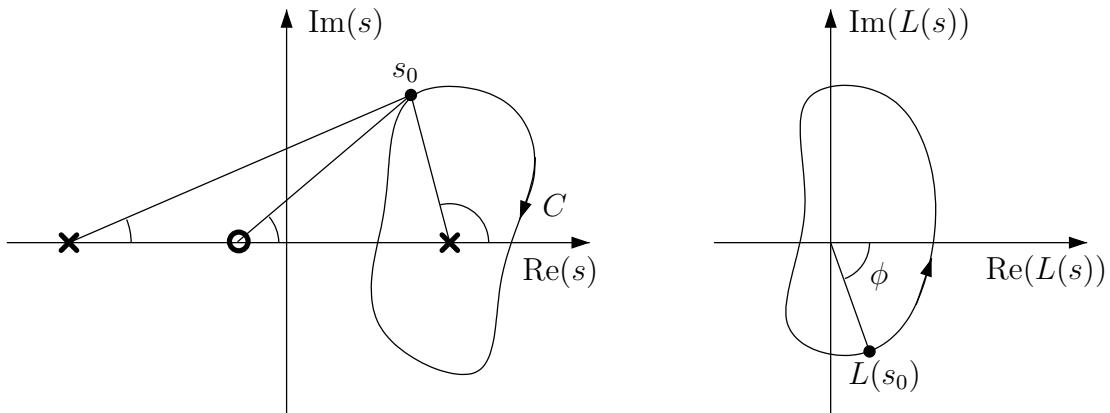
Figure 4.16: Contour evaluation with one pole enclosed
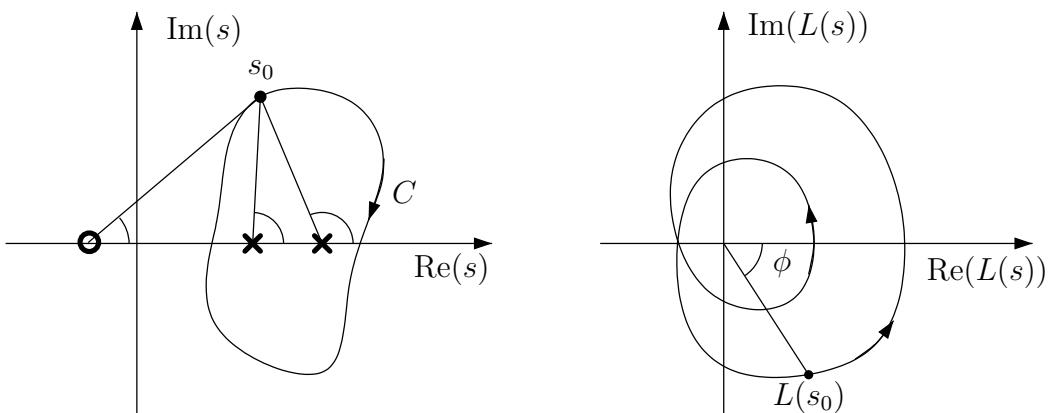
Figure 4.17: Contour evaluation with two poles enclosed

These examples illustrate Cauchy's principle: the evaluation of $L(s)$ along a closed contour C encircles the origin only if C encircles poles or zeros of $L(s)$. Because C is oriented clockwise, an enclosed zero causes a clockwise encirclement of the origin, and an enclosed pole a counterclockwise encirclement.

Nyquist's idea was to use this fact to detect the presence of closed-loop poles in the right half plane. For this purpose, the contour C must be chosen such that it encloses the whole right half plane. Such a contour, known as the *Nyquist path* , is shown in Figure 4.18. It consists of the imaginary axis and an infinitely large semicircle that encloses the right half plane. Now consider the feedback system shown in Figure 4.19. The above idea will now be applied to check the closed-loop stability of this system.

The closed-loop characteristic equation is

$$1 + KL(s) = 0$$

and an unstable zero of $1 + KL(s)$ indicates an unstable closed-loop pole. We could evaluate this function along the Nyquist path; but before we proceed we make the following simplification. Instead of evaluating $1 + KL(s)$ and checking encirclements of the origin, we
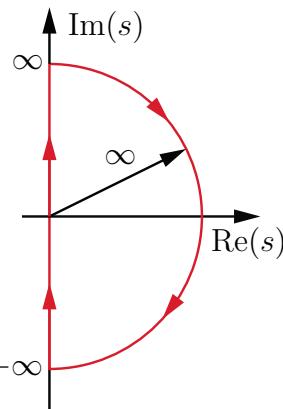
Figure 4.18: Nyquist path

can equivalently evaluate the open-loop transfer function $KL(s)$ and check encirclements of the point -1, see Figure 4.20.
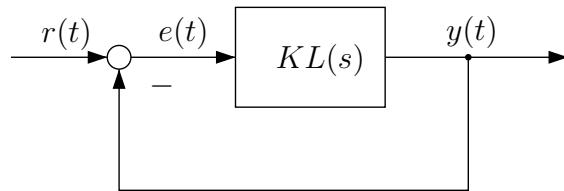
Figure 4.19: Closed-loop system

To evaluate $KL(s)$ along the Nyquist path means to evaluate along the $j\omega$-axis from $\omega = -\infty$ to $\omega = \infty$, and along the infinite arc. However, the transfer functions of physical systems are zero at infinite frequency, and the infinite arc in the s-plane is mapped into the origin of the $L(s)$-plane (as can be seen in Figure 4.13 and 4.14). Therefore, $KL(s)$ needs to be evaluated only from $-j\infty$ to $j\infty$, and this yields precisely the Nyquist plot of $KL(s)$ for positive and negative frequencies, as for example shown in Figure 4.13 and 4.14.

Note that the number of encirclements of the point -1 is not only influenced by the *zeros* but also by *poles* of $1 + KL(s)$ in the right half plane. So far, we have established the following:

> If the right half plane contains a zero or a pole of $1 + KL(s)$, the Nyquist plot of $KL(s)$ encircles the point -1 (clockwise or counterclockwise, respectively).

To distinguish between poles and zeros of the function $1 + KL(s)$, we write $L(s) = b(s)/a(s)$ to obtain

$$1 + KL(s) = \frac{a(s) + Kb(s)}{a(s)}.$$

This shows that
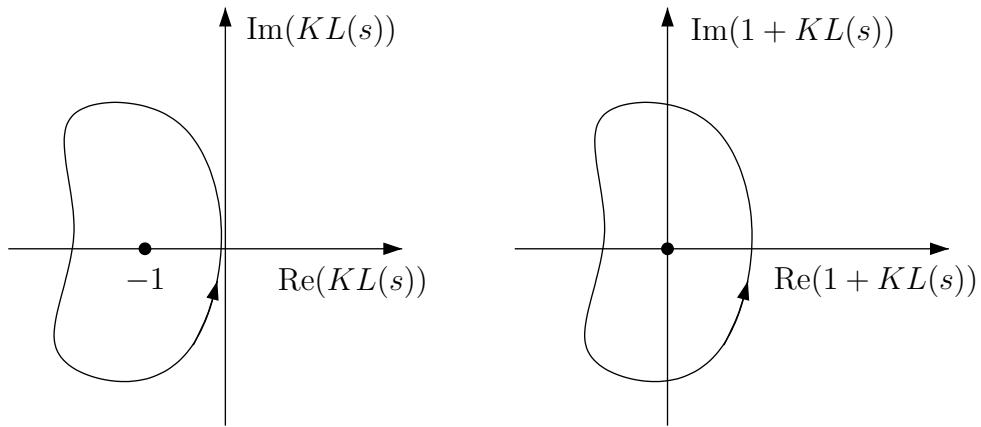
Figure 4.20: Shifting the evaluation of $1 + KL(s)$ to the left

- the *zeros* of the function $1 + KL(s)$ are the *closed-loop poles*, and

- the *poles* of the function $1 + KL(s)$ are the *open-loop poles* (the poles of $L(s)$).

We assume that if the open loop transfer function is unstable, the number of unstable poles is known, so that these can be taken into account when checking closed-loop stability. If we let Z denote the number of unstable closed-loop poles, and P the number of unstable open-loop poles, then the above considerations show that the Nyquist plot of $KL(s)$ encircles $N = Z - P$ times clockwise the point -1 (a negative value of N indicates counterclockwise encirclements). Closed-loop stability can now be checked as follows.

Nyquist Stability Test

1) Draw the Nyquist plot of $KL(s)$

2) Determine the number N of clockwise encirclements of the point -1 and the number P of unstable open-loop poles

3) The closed-loop system has $Z = N + P$ unstable poles

If the open loop transfer function is stable, then the closed-loop system is stable if the Nyquist plot does not encircle the point -1. If $KL(s)$ has one unstable pole, then closed-loop stability requires one counterclockwise encirclement of -1, etc.

Example 4.6

Consider

$$KL(s) = \frac{K}{s + 1}.$$

The Nyquist plot is shown in Figure 4.21, it is a circle centered at $K/2$ with radius $K/2$ (the Nyquist plot of Figure 4.13 scaled by the factor K). Because $P = 0$ (no unstable

poles of $L(s)$) and $N = 0$ (no encirclements of -1) for all $K > 0$, the closed-loop system is stable for all positive K.
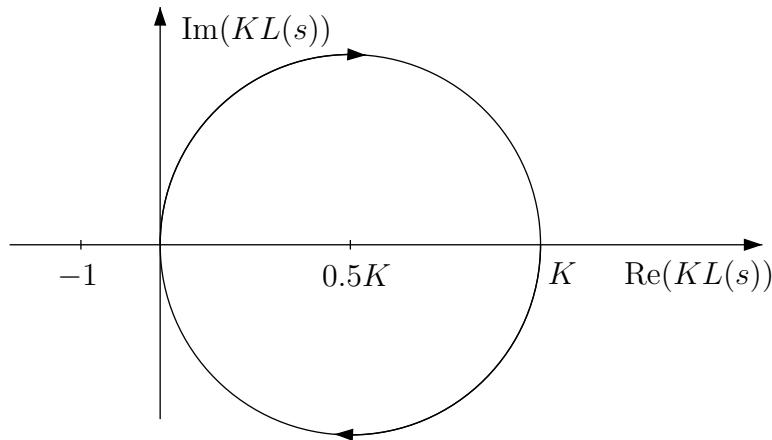
Figure 4.21: Nyquist stability test for Example 4.1

Example 4.7

The Nyquist plot of the open-loop transfer function

$$KL(s) = \frac{K}{s(s+1)}$$

is obtained by scaling the Nyquist plot of Figure 4.14 by the factor K. Apparently, that Nyquist plot is not a closed contour even though it is the mapping of the Nyquist path, which is a closed contour. This is due to the fact that the Nyquist path as shown in Figure 4.18 passes through a pole of $L(s)$ at the origin. At a pole, the value of the transfer function is infinite, and the Nyqist plot is in fact closed by an arc at infinity. To see why this is so, consider the path shown in Figure 4.22. The Nyquist path has been modified to avoid the pole at the origin by making an infinitely small detour to the right. It is this small arc that is mapped into an arc at infinity, and to determine whether or not the Nyquist plot encircles the point -1, it is important to know whether this infinite arc encloses the right half plane or the left half plane. On the small arc in Figure 4.22, three test points s_i, $i = 1, 2, 3$, are marked. Because the radius of the semicircle around the origin is infinitely small, the magnitude of $KL(s_i)$ is infinite, and the phase angles - determined by the pole angles - are

$$\arg KL(s_1) = +90^o, \quad \arg KL(s_2) = 0^o, \quad \arg KL(s_3) = -90^o$$

This shows that the Nyquist plot is completed by an infinite arc to the right, as shown in Figure 4.23. Because the modified Nyquist path does not encircle the pole at the origin,
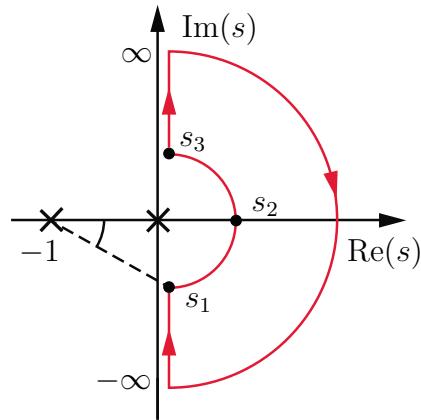
Figure 4.22: Modified Nyquist path

we have $P = 0$, and from Figure 4.23 we conclude that there is no encirclement of -1 for any positive value of K (i.e. $N = 0$), and therefore the closed-loop system is stable for any positive gain.
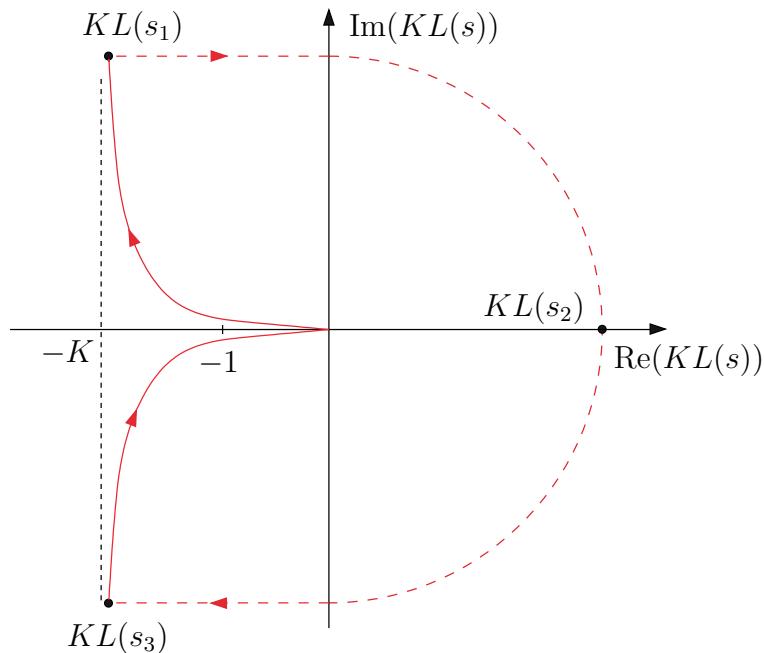
Figure 4.23: Nyquist stability test for Example 4.7

Example 4.8

Consider the transfer function

$$L(s) = \frac{10}{(s+1)^3} = \frac{10}{s^3 + 3s^2 + 3s + 1}.$$

Substituting $j\omega$ for s gives

$$L(j\omega) = \frac{10}{(j\omega)^3 + 3(j\omega)^2 + 3j\omega + 1}.$$

At low frequencies $\omega \ll 1$ we have

$$L \approx 10 \quad \Rightarrow \quad |L| \approx 10, \quad \phi \approx 0^o$$

and at high frequencies

$$L \approx \frac{10}{(j\omega)^3} = \frac{10}{-j\omega^3} = j\frac{10}{\omega^3} \quad \Rightarrow \quad |L| \approx \frac{10}{\omega^3}, \quad \phi \approx -270^o.$$

The Bode plot of this function is shown in Figure 4.24. The points where the phase plot intersects the -90° line and the -180° line have been marked as A and B, respectively. The Nyquist plot of the same function is shown in Figure 4.25. The points marked A
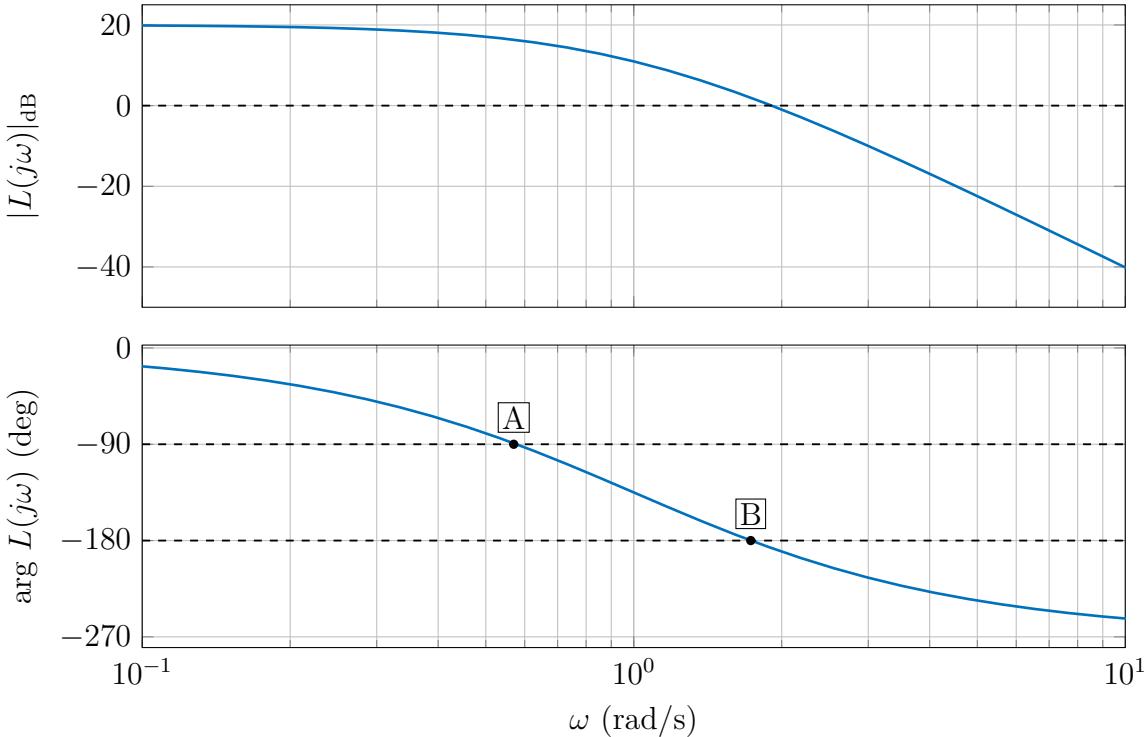
Figure 4.24: Bode plot of $L(s)$ for Example 4.8

and B in the Bode plot correspond to the points where the Nyquist plot intersects the negative imaginary and the negative real axis, respectively. The phase angle of -270° that can be read off the Bode plot at high frequencies (where the magnitude approaches zero) indicates that the Nyquist plot approaches the origin from above. The Nyquist plot of $L(s)$ (which is the same as the Nyquist plot of $KL(s)$ when $K = 1$) encircles the critical point -1 two times clockwise ($N = 2$). Because $L(s)$ has no unstable poles ($P = 0$), this implies that the closed-loop system has two unstable poles when $K = 1$.
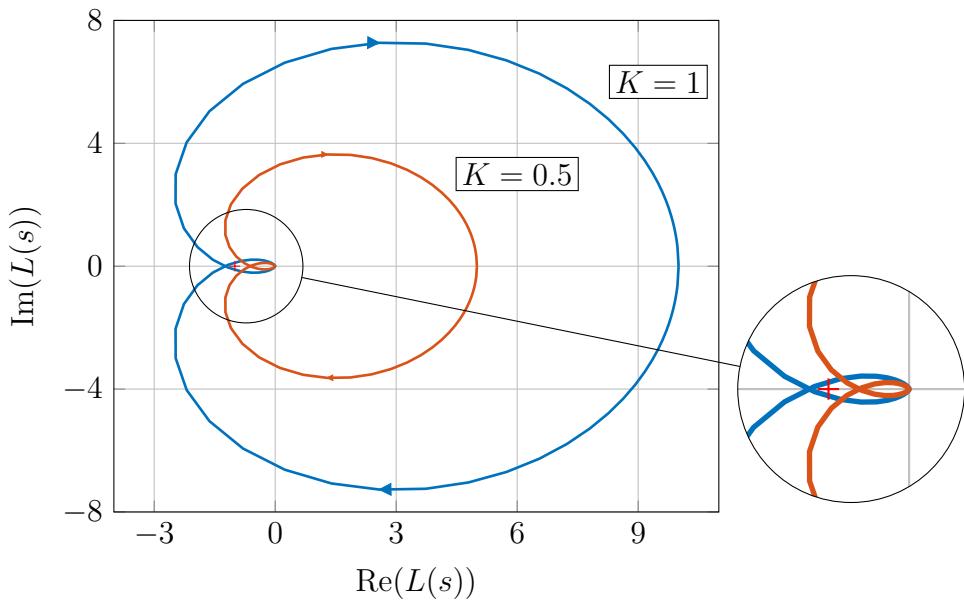
Figure 4.25: Nyquist plot of $L(s)$ for Example 4.8

Figure 4.25 shows the Nyquist plot with $K = 0.5$. The Nyquist plot of $L(s)$ has been "scaled down" to an extent that the critical point -1 is not encircled anymore. With this gain, the closed-loop system is stable.

To determine closed-loop stability for different values of K, it is convenient to use the Nyquist plot of $L(s)$ instead of $KL(s)$: the Nyquist plot of $KL(s)$ encircles the point -1 precisely if the Nyquist plot of $L(s)$ encircles the point $-1/K$. Therefore closed-loop stability can be assessed by counting the encirclements of the point $-1/K$ by the Nyquist plot of $L(s)$. Figure 4.25 shows that the Nyquist plot of $L(s)$ crosses the negative real axis at -1.25, therefore the Nyquist plot of $KL(s)$ does not encircle the critical point -1 if the gain is less than 0.8, and the closed-loop system is stable for all $0 < K < 0.8$.

4.3   Gain and Phase Margin

The last example shows that the Nyquist stability test does not only indicate whether a closed-loop system is stable, it also gives information about the "distance" from the stability boundary. Such a measure is called a *stability margin*, and in this section we introduce two stability margins which play an important role in frequency response analysis and design: the *gain margin* and the *phase margin*. Gain margin and phase margin are useful when studying systems that behave like the system in the last example, i.e. they are stable for small gain values and become unstable when the gain is increased beyond a certain value. Such systems are typical for many practical applications.

For an illustration of the stability margins, consider Figure 4.26. It shows part of the Nyquist plot of the open-loop transfer function $L(s)$ considered in the last example in the previous section. Because the intersection with the negative real axis (marked B) is to
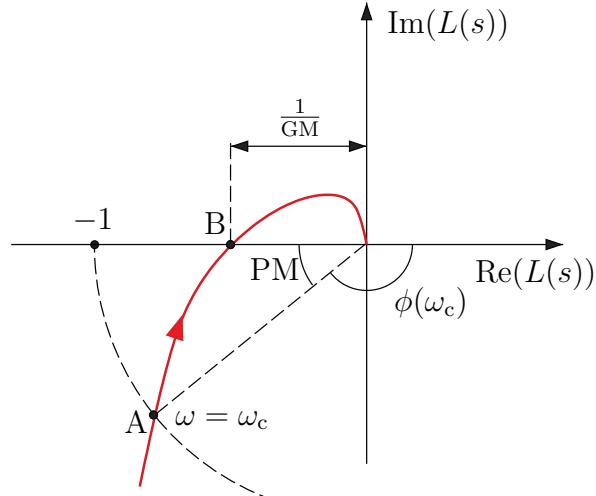
Figure 4.26: Definition of gain and phase margin

the right of the critical point -1, the closed-loop system in Figure 4.19 is stable if $K = 1$. Moreover, the gain K can take values up to the inverse of the magnitude of L in point B before the closed-loop system becomes unstable. This upper bound on K, above which the closed-loop system becomes unstable, is called the *gain margin* (GM) of the system.

In a similiar way, the *phase margin* (PM) is defined as the maximum phase change of $L(s)$ that the closed-loop system can tolerate before it becomes unstable. A phase shift means to rotate the Nyquist plot about the origin. The point on the Nyquist plot that will hit the point -1 can be found by drawing a circle with radius 1 around the origin; in Figure 4.26 the intersection is marked as point A. The maximum phase shift the system can tolerate before becoming unstable is precisely the angle of rotation required to move point A to the point -1, and it is this angle (marked PM) that is referred to as phase margin.

The values of GM and PM can often be read off the Bode diagram, as shown in Figure 4.27. A more detailed discussion of when this is not possible can be found at the end of this section. The points A and B are the intersection of the magnitude plot with the 0dB line and the intersection of the phase plot with the -180° line, respectively. The frequency at which the magnitude curve intersects the 0dB-line is called the *crossover frequency ω_c*. A condition for closed-loop stability is that at the crossover frequency the phase angle is above -180°, or - equivalently - that at the frequency where the phase is -180° the magnitude is below 0 dB.

Returning to Example 4.8, we recall that the closed-loop system is stable when $K = 0.5$. The Bode plot in Figure 4.28 gives a frequency domain interpretation of the effect of
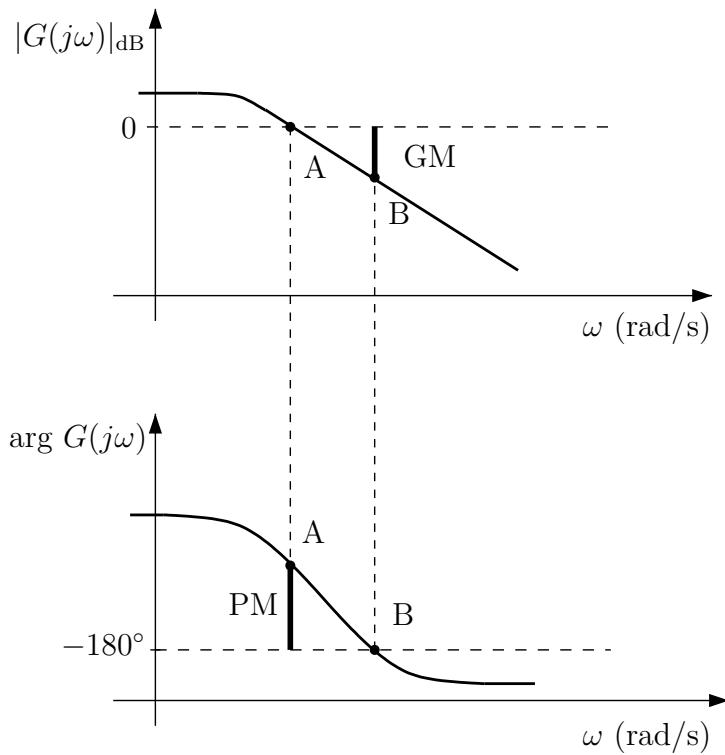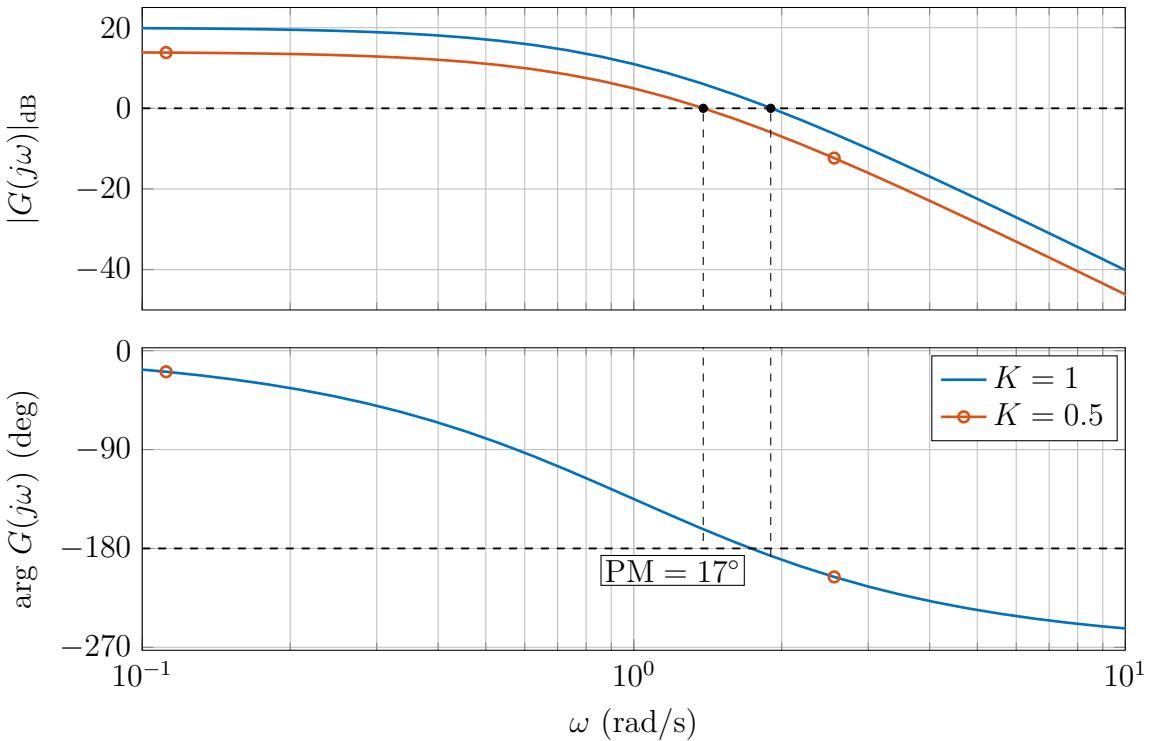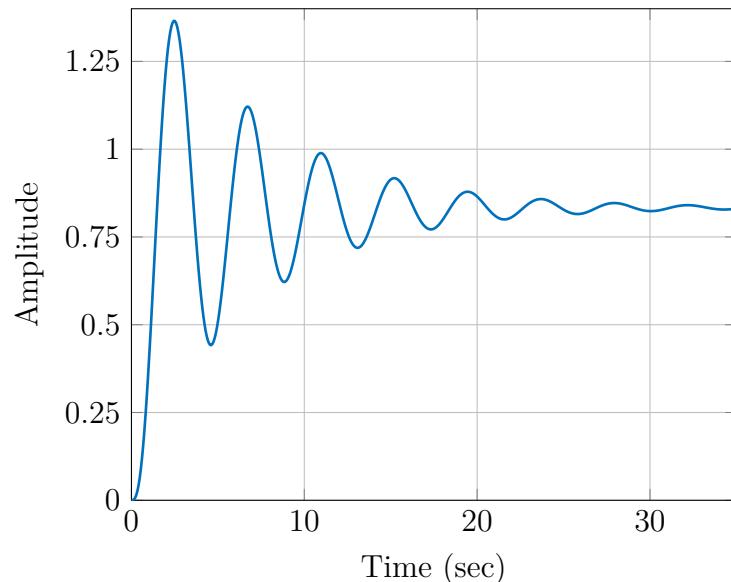
Figure 4.27: Phase and gain margin in the Bode plot

reducing the gain: it leads to a lower crossover frequency and - because the phase remains unchanged - to a positive phase margin and thus to a stable closed-loop system. The phase margin of 17° is however rather small, the system is still close to instability. Figure 4.29 shows the closed-loop step response for this case; with considerable overshoot and oscillation such a response would be unacceptable for most applications. A phase margin of 30° is usually considered to be the minimum required for acceptable transient behaviour.

The root locus for this feedback system is shown in Figure 4.30, where the closed-loop pole locations for $K = 0.5$ are marked. The poles are close to the imaginary axis, which explains the oscillatory step response.

Figure 4.28: Phase margin with $K = 0.5$

Figure 4.29: Closed-loop step response with $K = 0.5$

Second Order Approximation

For a second order system, peak overshoot and oscillation in the transient response are determined by the damping ratio ζ. A rule of thumb that relates the damping ratio to the phase margin is

$$\zeta \approx \frac{\text{PM}}{100}$$

In practice, this relation is often used as an approximation even for higher order systems. This is partly justified when the response is dominated by a pole pair close to the imaginary axis - as in Figure 4.30, where the "fast" pole to the left has only little effect on the transient response.
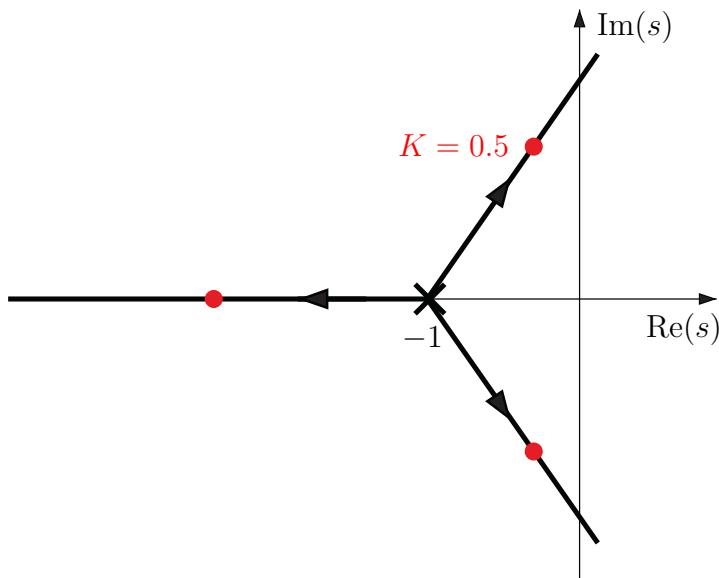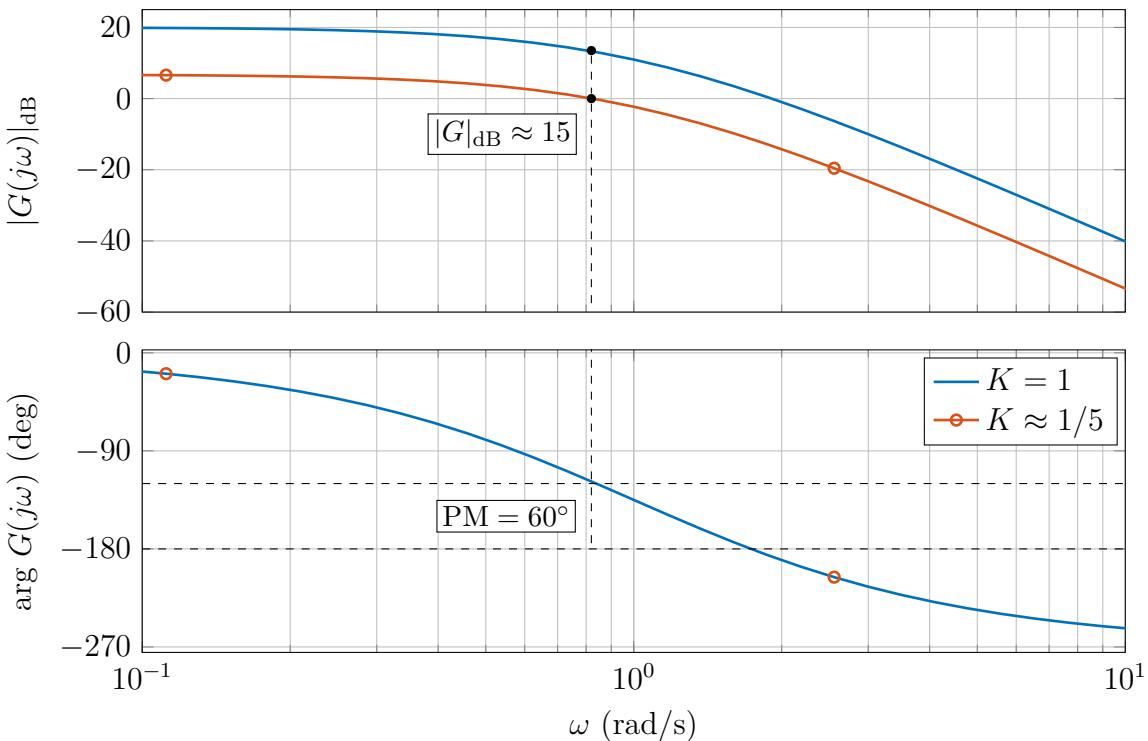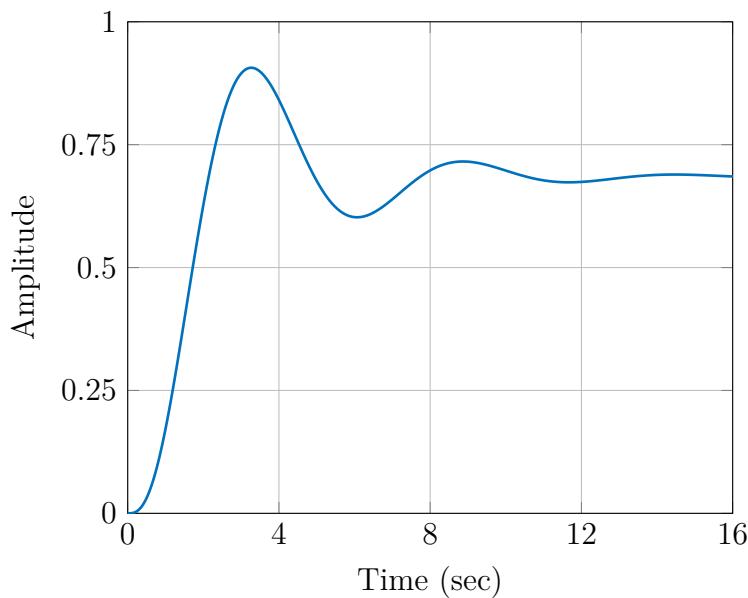
Figure 4.30: Closed-loop pole locations for $K = 0.5$ (marked in red)

To illustrate how this rule of thumb can be used for a simple design, assume that the design objective is to reduce the peak overshoot in the above example to 10%, i.e. $M_p = 0.1$. This requires a damping ratio of about 0.6, so we design the system to have a phase margin of

$$\text{PM} \approx 60^o$$

The crossover frequency and gain required for a phase margin of 60° can be directly read off the Bode plot, as shown in Figure 4.31. The crossover frequency required for a 60° phase margin is approximately 0.8. The magnitude at this frequency is 15 dB (≈ 5.6 on a linear scale), so a gain of $K \approx 1/5$ makes this frequency the crossover frequency of $KG(s)$. Figure 4.32 shows the closed-loop step response with $K \approx 1/5$.

Figure 4.31: Design for a phase margin of 60°

Figure 4.32: Step response with 60° phase margin ($K \approx 1/5$)

Phase and Gain Margin can not Always be Obtained from Bode Plots

Obtaining phase and gain margins from Bode plots was discussed above for systems that are stable for small gains and become unstable for large gains. It should be noted that there are situations were these margins cannot be obtained from the Bode plot: for systems

that are unstable for small gains, or for non-minimum phase systems, the rules discussed above do not apply. Higher order systems may have several crossings of the 0 dB or -180 degree line; this requires a re-definition of the margins. A practical example of such a case is studied in Exercises 4.15-4.17, where a system with a resonant peak is to be controlled. The rules for asserting stability from gain and phase margins presented here are valid for minimum phase systems where the magnitude has only a single (decreasing) crossover of the 0-dB line. If this is not the case, a rigorous assessment of the stability range is always possible by applying the Nyquist criterion directly.

4.4 Dynamic Compensation

Even though peak overshoot and oscillation have been reduced in the previous example, the response in Figure 4.32 cannot be considered satisfactory: the steady state value in the response to a unit step is 0.625, showing a steady state error of almost 40%. The large steady state error is due to the low gain, which in turn is required for an acceptable transient response. This problem has been discussed before; a satisfactory solution cannot be obtained by proportional feedback alone. Additional dynamics - derivative and integral feedback - have to be introduced. In terms of the root locus, additional zeros and poles are required to change the shape of the root locus plot. In this section, we review the approach discussed earlier and introduce a systematic design procedure based on the open-loop frequency response. We consider feedback systems of the form shown in Figure 4.33, with plant transfer function $G(s)$ and controller transfer function $C(s)$. The frequency response on which the design is based is that of the open-loop transfer function $L(s) = G(s)C(s)$.
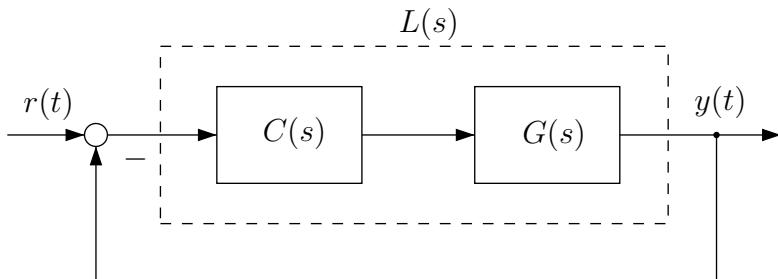
Figure 4.33: Closed-loop system

Closed-Loop Frequency Response

In the previous two sections, properties of the *closed-loop system* have been inferred from the *open-loop* response. Now we consider the desired shape of the closed-loop frequency response and the relationship between *closed-loop* and *open-loop* frequency response. For tracking reference inputs, the ideal closed-loop transfer function would be $G_{cl}(s) = 1$, which implies perfect tracking $Y(s) = R(s)$ or $y(t) = r(t)$. Such a closed-loop behaviour

is not physically realizable. This is obvious for high-frequency inputs: no physical system can - due to inertia - respond arbitrarily fast to changes in the command input. On the other hand, at low frequencies and in particular in steady state operation, a closed-loop transfer function equal to or close to 1 would be a reasonable design objective. The magnitude of a typical closed-loop frequency response is shown in Figure 4.34.
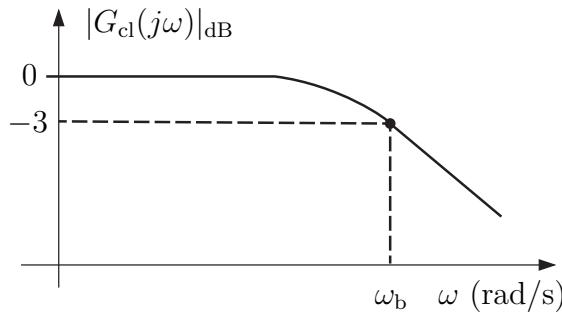
Figure 4.34: Desired closed-loop frequency response

In the low frequency region, the magnitude is approximately 1. At higher frequencies it starts to roll off at some point; the frequency at which the magnitude is -3 dB is called the *closed-loop bandwidth* ω_b. The bandwidth represents the frequency range in which the system is to be operated and where the closed-loop transfer function should be close to 1. At high frequencies it will be mainly disturbances and measurement noise that will act on the system. For this reason, the magnitude of the closed-loop response should be close to 1 within the bandwidth, but roll off rapidly beyond the bandwidth such that noise and high frequency disturbances are suppressed.

The closed-loop transfer function is

$$G_{cl}(s) = \frac{L(s)}{1 + L(s)}$$

and its magnitude is

$$|G_{cl}| = \left| \frac{L}{1 + L} \right|.$$

Conditions for the open-loop transfer function $L(s)$ to achieve the desired shape of the closed-loop response are

$$\omega \ll \omega_b : \quad |L| \gg 1 \quad \Rightarrow \quad |G_{cl}| \approx 1$$

and

$$\omega \gg \omega_b : \quad |L| \ll 1 \quad \Rightarrow \quad |G_{cl}| \approx |L|.$$

The requirement that the open-loop gain be large at low frequencies is a generalization of the result discussed in Section 2.3: that the steady state gain should be large in order to
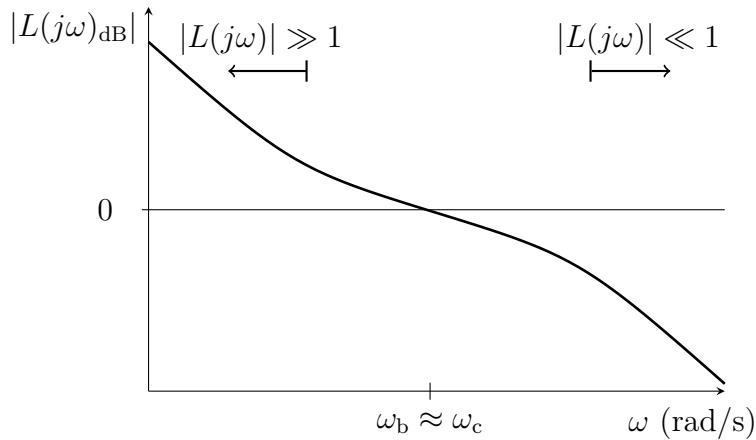
Figure 4.35: Typical open-loop frequency response

achieve a small steady state error. For the steady state error to be zero, the low frequency gain should have a negative slope, i.e. the system should be of type 1 or higher.

A typical open-loop frequency response is shown in Figure 4.35. It is the response of $L(s) = G(s)C(s)$, where the plant transfer function $G(s)$ is given, and the controller transfer function $C(s)$ is the design parameter that can be used to achieve the desired shape of the closed-loop response.

It is clear from the above requirements (and from Figure 4.35) that the closed-loop bandwidth ω_b is in general close to the open-loop crossover frequency ω_c. In order to achieve the shape of the closed-loop response in Figure 4.34, one might conclude that the open-loop response should have a steep negative slope (of say -40 dB/dec or -60 dB/dec) over all frequencies. This is however impossible because of the phase lag associated with such a slope, as discussed below.

Roll-Off and Phase Margin

We now return to the open-loop magnitude plot in Figure 4.35: for a small steady state error and good rejection of high frequency noise, the open loop magnitude should have a steep roll-off at low as well as at high frequencies. At the crossover frequency however, the slope is limited by the requirement that for a reasonable phase margin, the phase must be well above -180°. A constant slope of -40 dB/dec in the region of the crossover frequency results in a phase margin of zero: the closed-loop system is on the boundary to instability. For acceptable transient behaviour, the phase margin should be above 30°; a desirable phase margin is 90° which requires a constant slope at the crossover frequency of -20 dB.

Phase-Lead Compensation

If the phase of the plant transfer function $G(s)$ at the crossover frequency ω_c does not provide a sufficient phase margin, the controller transfer function $C(s)$ can be used to compensate for this phase lag. An additional zero with corner frequency below the crossover frequency increases the phase angle by 90°. In fact, introducing an additional zero is equivalent to introducing derivative feedback (see Section 2.2). The transfer function of a PD controller is

$$C(s) = K_P(1 + T_D s).$$

Defining the corner frequency $\omega_D = 1/T_D$ and substituting $j\omega$ for s this can be written as

$$C(j\omega) = K_P \left(1 + j \, \frac{\omega}{\omega_D} \right).$$
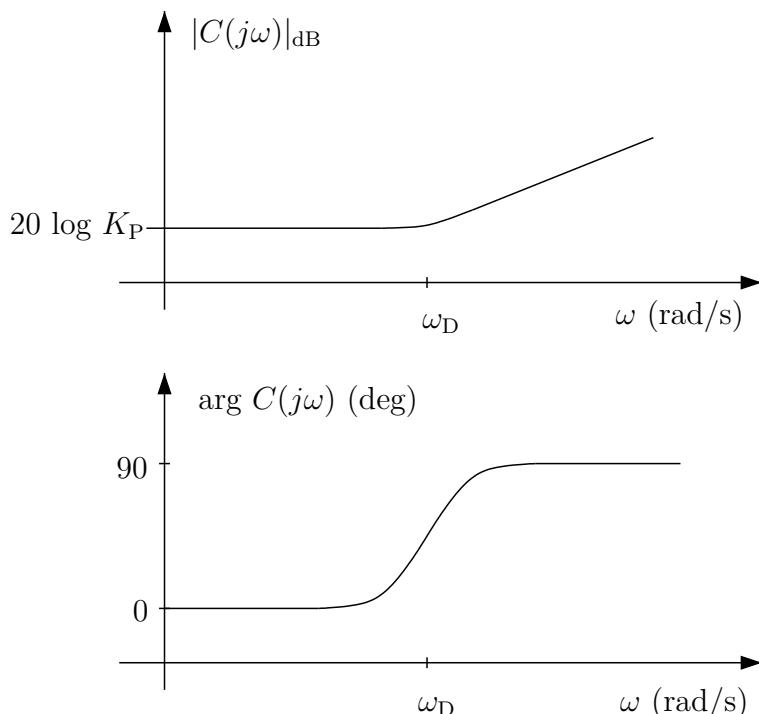
Figure 4.36: Frequency response of a PD controller

Figure 4.36 shows the frequency response of a PD controller: it achieves the desired phase increase. However, as discussed in Section 2.2, such a controller is neither physically realizable, nor desirable in this form: the 20 dB/dec gain increase above the corner frequency would make the system extremely sensitive to high frequency noise. An additional pole at a higher frequency can be introduced to bring the slope of the magnitude curve back

to zero; this also reduces the phase to 0° above the corner frequency associated with the pole. The resulting controller is called a *phase-lead compensator*, its transfer function is

$$C(s) = K_P \, \frac{1 + T_L s}{1 + \alpha T_L s} \tag{4.8}$$

where $\alpha < 1$. Introducing the corner frequency $\omega_L = 1/T_L$, this can be written as

$$C(j\omega) = K_P \, \frac{1 + j\,\frac{\omega}{\omega_L}}{1 + j\,\frac{\omega}{\omega_L/\alpha}}.$$
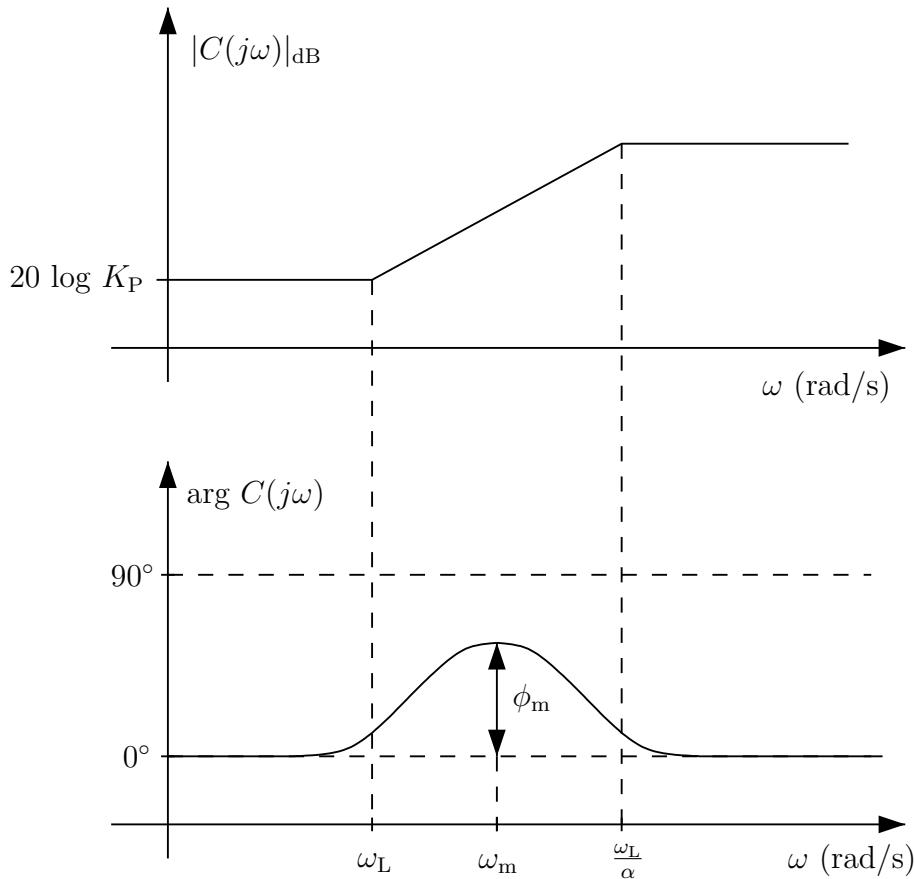
Figure 4.37: Frequency response of a phase-lead compensator

Figure 4.37 shows the frequency response of the compensator. The phase maximum is located at the frequency ω_m, the phase angle at ω_m depends on the "distance" between the pole and the zero of the compensator. If the distance is large, say three decades (obtained by taking $\alpha = 0.001$), a phase increase of almost 90° can be achieved, however at the expense of a 60 dB gain increase at high frequencies. The compensator design involves a trade-off between phase increase and low noise sensitivity, and the distance between ω_L and ω_L/α should just be large enough to obtain the desired phase margin, without unnecessarily increasing the high frequency gain.

When designing a phase-lead compensator, there are three design parameters to be chosen: the proportional gain K_P, the corner frequency ω_L and the parameter α. An exact relationship between the maximum phase angle ϕ_m and the parameter α can be derived from the Nyquist plot of the compensator. For simplicity, we assume $K_P = 1$ (without loss of generality, because K_P has no effect on the phase).
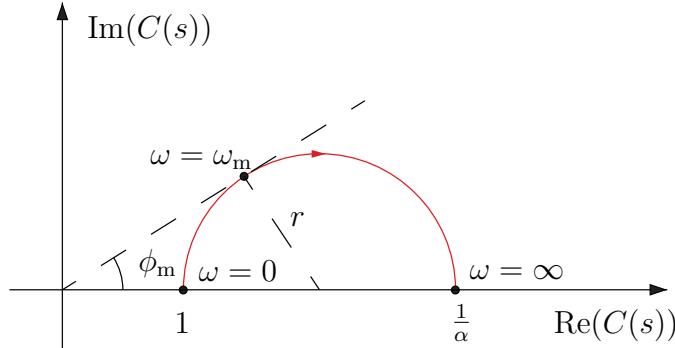
Figure 4.38: Maximum phase lead at ω_m

The Nyquist plot for this case is shown in Figure 4.38; it is a semicircle with its position determined by

$$\omega = 0 \quad \Rightarrow \quad C = 1$$
$$\omega = \infty \quad \Rightarrow \quad C = \frac{1}{\alpha} > 1$$

and the fact that the phase for frequencies $0 < \omega < \infty$ is positive. The point where the phase attains its maximum ϕ_m is marked. The radius of the semicircle is

$$r = \frac{1}{2}\left(\frac{1}{\alpha} - 1\right)$$

and we have

$$\sin \phi_m = \frac{r}{1 + r} = \frac{\alpha r}{\alpha + \alpha r}$$

or

$$\sin \phi_m = \frac{1 - \alpha}{1 + \alpha}.$$

Solving for α yields

$$\alpha = \frac{1 - \sin \phi_m}{1 + \sin \phi_m}. \tag{4.9}$$

When the parameter α has been determined to achieve the desired phase lead ϕ_m, the corner frequency ω_L must be chosen such that the phase maximum is located at the crossover frequency, i.e. ω_m should be equal to the crossover frequency. Taking the log scale of the frequency axis into account, we have

$$\log \omega_m = \frac{1}{2}\left(\log \omega_L + \log \frac{\omega_L}{\alpha}\right)$$

which can be solved for

$$\omega_L = \sqrt{\alpha}\,\omega_m. \tag{4.10}$$

Design Procedure for Phase Lead Compensation

We now outline a systematic design procedure for a lead compensator, *i.e.* a procedure for choosing the values of K_L, α and ω_L to meet given design specifications. The procedure is illustrated with an example: control design for a plant with double integral behaviour

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

such that a specified bandwidth ω_b is achieved whith a reasonable phase margin.

Initial Gain K'_L
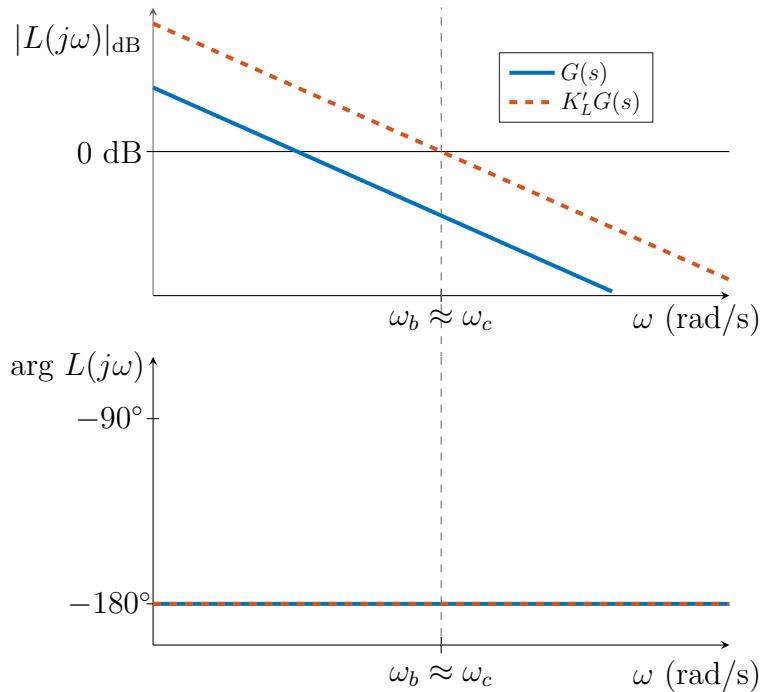
The first step is to choose an initial gain K'_L, based on design specifications for the bandwidth (speed of response) and/or the steady state accuracy. The gain K'_L is chosen such that it achieves the desired bandwidth or the desired steady state accuracy in a purely proportional feedback design, *i.e.* assuming $C(s) = K'_L$. If design specifications for both bandwidth *and* steady state accuracy are given, one needs to determine the required gain for either condition and select the larger one. Figure 4.39 shows the Bode plot of the open-loop plant $G(s)$ (solid) and the magnitude of $K'_L G(s)$ (dashed) required for achieving a desired bandwidth, assuming $\omega_b \approx \omega_c$. Note that in this example it is assumed that the gain needs to be increased to achieve the specification.

Crossover Frequency ω_c and the Parameters α and ω_L

Choosing the initial gain K'_L determines the crossover frequency ω_c. One can then obtain the phase margin of the uncompensated loop (by checking the open-loop phase at that frequency in the Bode plot), and determine the desired phase lead ϕ_m. With that, we can from (4.9) calculate the parameter α, and by selecting $\omega_m = \omega_c$ we find the frequency ω_L from (4.10). Now we can form an initial controller

$$C'(s) = K'_L \frac{1 + j\frac{\omega}{\omega_L}}{1 + j\frac{\omega}{\omega_L/\alpha}}$$

that places the designed phase lead at ω_c. This is illustrated in Figure 4.40. That figure however also shows an undesired effect of the phase lead compensation: the compensator zero leads to an increase in the gain above ω_L, which in turn results in an increase of the crossover frequency. In Figure 4.40 the new crossover frequency is denoted by ω'_c. The phase margin in this example would thus be less than what the compensator was designed to achieve.

Figure 4.39: Bode plot of $G(s)$ and $K'_L G(s)$

Calculate the Gain K_L

To prevent the shift of the crossover frequency, one needs to calculate the gain K_L from K'_L. The gain K'_L was initially chosen to satisfy the condition

$$K'_L \, |G(j\omega_c)| = 1 \tag{4.11}$$

in order to make the frequency ω_c the crossover frequency. Using a contoller

$$C(s) = K_L \frac{1 + j\frac{\omega_c}{\omega_L}}{1 + j\frac{\omega_c}{\omega_L/\alpha}}$$

with an adjusted gain K_L, the gain condition at the cross-over frequency becomes

$$K_L \left| \frac{1 + j\frac{\omega_c}{\omega_L}}{1 + j\frac{\omega_c}{\omega_L/\alpha}} \right| \, |G(j\omega_c)| = 1. \tag{4.12}$$

Comparing (4.11) with (4.12), we thus need

$$K'_L = K_L \, \beta,$$

where

$$\beta = \left| \frac{1 + j\frac{\omega_c}{\omega_L}}{1 + j\frac{\omega_c}{\omega_L/\alpha}} \right|.$$
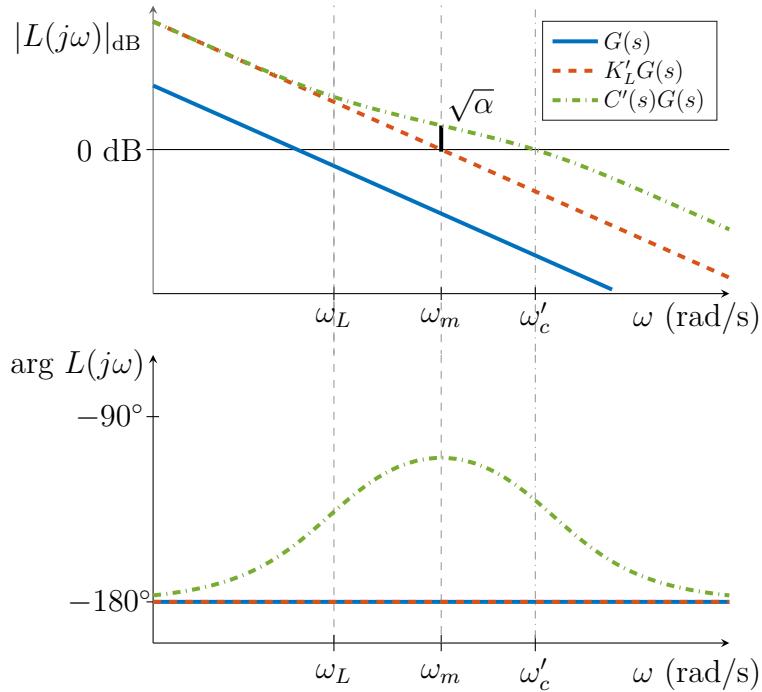
Figure 4.40: Increasing the phase margin

From (4.10) we have $\omega_L = \sqrt{\alpha}\,\omega_c$ and thus

$$\beta = \frac{1 + j/\sqrt{\alpha}}{1 + j\sqrt{\alpha}},$$

from which we obtain

$$|\beta| = \frac{\sqrt{2 + \alpha + \frac{1}{\alpha}}}{1 + \alpha} = \sqrt{\frac{\alpha + 1}{\alpha^2 + \alpha}} = \sqrt{\frac{1}{\alpha}}.$$

For the feedback loop to have the initially designed crossover frequency ω_c, we therefore need to reduce the initial gain K'_L to

$$K_L = K'_L \sqrt{\alpha}. \tag{4.13}$$

The Bode plot of the open-loop with the final controller $C(s)$ using the adjusted gain K_L is shown in Figure 4.41.

The design procedure can be summarized as follows.

1. Fix an initial controller gain K'_L based on specifications for bandwidth and/or steady state accuracy; this determines the crossover frequency ω_c.

2. From the uncompensated phase at ω_c and the desired phase margin, find the required phase lead ϕ_m; calculate α and ω_L from (4.9) and (4.10), respectively.

3. Implement the compensator $C(s)$ with gain $K_L = K'_L \sqrt{\alpha}$.
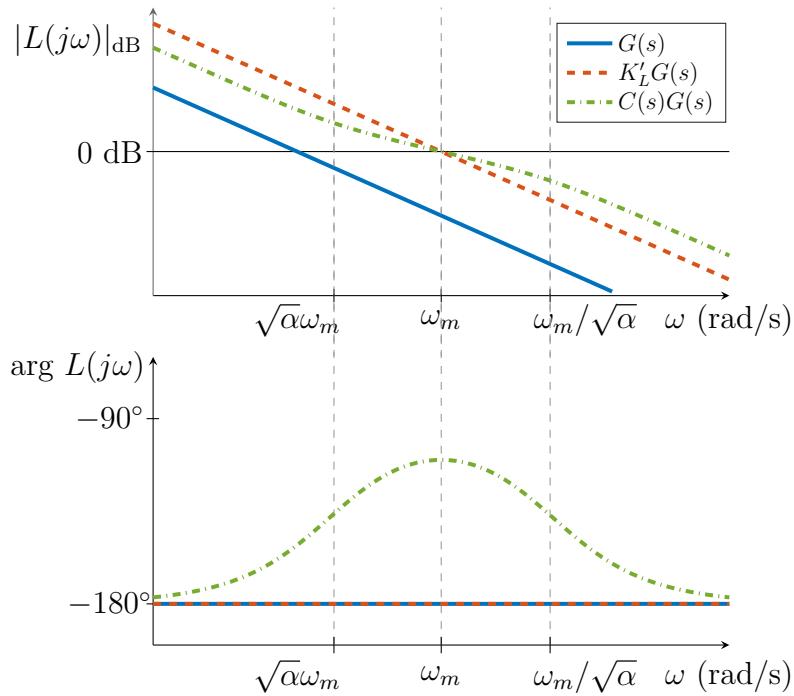
Figure 4.41: Adjusted gain

Note that if the initial gain K'_L was chosen to achieve a given steady state accuracy, this design will end up slightly below the specification because the implemented gain K_L is reduced in step 3. To account for this, one can select K'_L in step 1 to achieve a slightly higher accuracy than actually required.

Phase-Lag Compensation

Figure 4.42 shows the frequency response of a controller with transfer function

$$C(s) = K_P \frac{1 + T_L s}{1 + \alpha T_L s} \tag{4.14}$$

where $\alpha > 1$. The only difference to a phase-lead compensator is that α is greater, not less than 1. As a result, the corner frequency associated with the pole is lower than that of the zero, and the effect on gain and phase is reversed. Because of the decrease in phase, such a compensator is called a *phase-lag compensator*. Phase-lag compensation can be used to increase the loop gain at low frequencies, or to reduce the gain at high frequencies. Note that the desired effect of a phase-lag compensator is the change in the loop gain, whereas the change in phase is an undesired side effect. This is in contrast to phase-lead compensation, where the phase change is the desired effect and the gain change is an undesired side-effect.

When designing a phase-lag compensator, care must thus be taken to locate the corner frequencies well below (or well above) the crossover frequency such that the phase lag does not reduce the phase margin (see Figure 4.43). It is possible to combine phase-lag compensation with phase-lead compensation. This can be useful for example when a
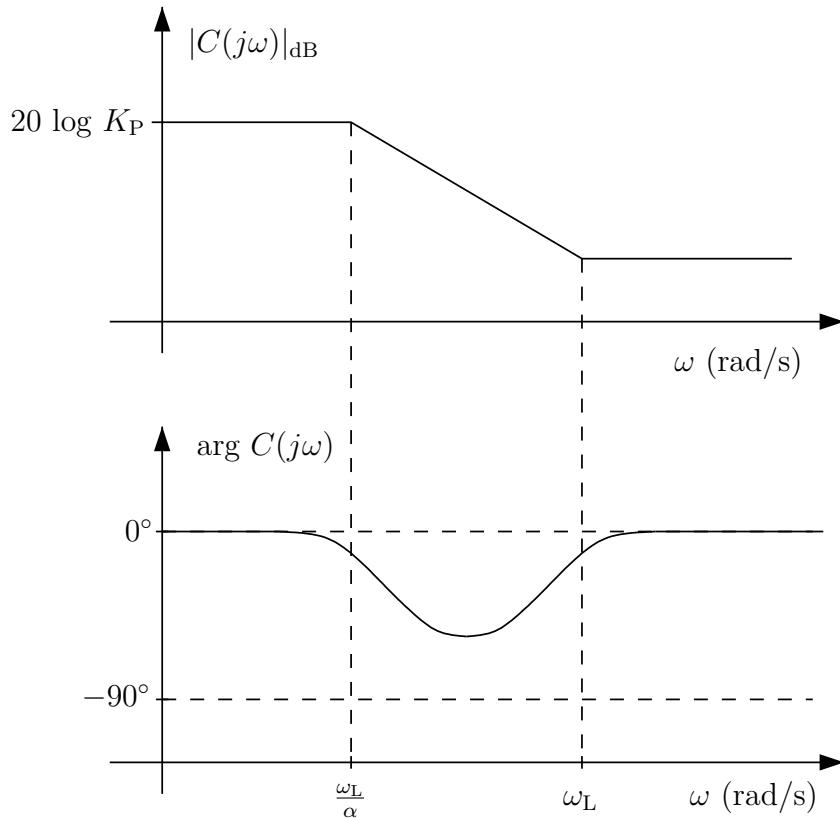
Figure 4.42: Frequency response of a phase-lag compensator

phase-lead design has already been carried out and an acceptable transient behaviour of the closed loop has been achieved, but the steady state accuracy is still not satisfactory. A lag compensator can then be added to increase low-frequency gain without destroying the phase-lead design, see Exercise 4.11.

A Frequency Response Interpretation of PID Control

The review of PD control in this chapter revealed that - seen from a frequency response point of view - the beneficial effect of PD control that leads to improved transient behaviour is the associated phase lead. Improving steady state accuracy, on the other hand, requires an increase in the low-frequency gain, which will entail a phase lag, as is the case with phase-lag compensation but also with integral control. It should therefore be of interest to review the behaviour of PID control loops - as discussed in Chapter 2 - in terms of their open-loop frequency response.

Thus, consider an ideal PID controller

$$C(s) = K_P \left(1 + T_D s + \frac{1}{T_I s} \right) \tag{4.15}$$

which can be rewritten as

$$C(s) = K_P \frac{T_I s + T_I T_D s^2 + 1}{T_I s} = K_P T_D \frac{s^2 + \frac{1}{T_D} s + \frac{1}{T_D T_I}}{s}.$$
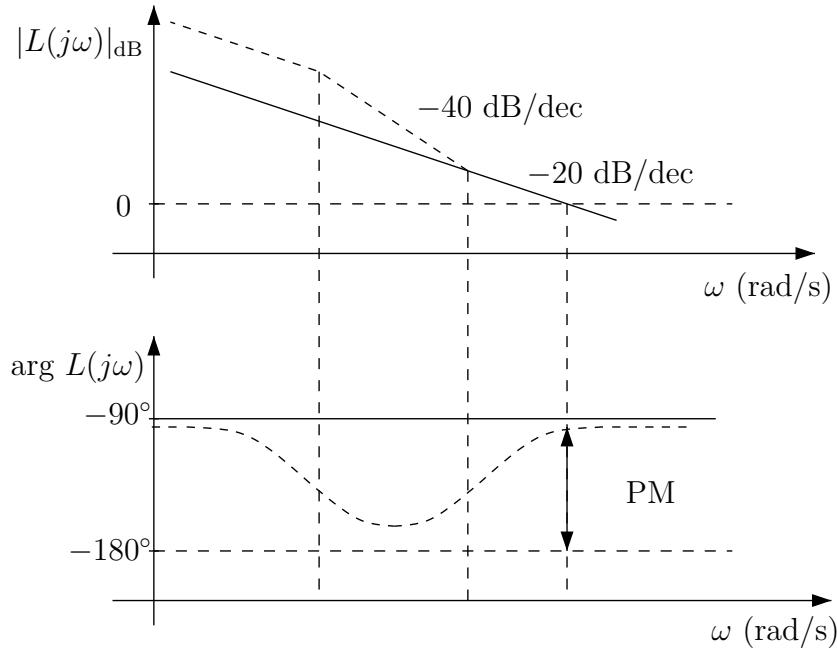
Figure 4.43: Increasing the low frequency gain by phase-lag compensation

This controller has two zeros at

$$s_{1,2} \ = \ -\frac{1}{2T_D} \pm \sqrt{\frac{1}{4T_d^2} - \frac{1}{T_D T_I}} \ = \ -\frac{1}{2T_D} \pm \sqrt{\frac{T_I - 4T_D}{4T_D^2 T_I}}.$$

Note that the controller zeros will be real if $T_I > 4T_D$.

To gain further insight, we introduce the following approximation. Using the fact that if $T_D \ll T_I$ (which is often the case) we have $1 + T_D/T_I \approx 1$, we can write the controller transfer function (4.15) as

$$
\begin{aligned}
C(s) \ &\approx \ K_P \left(1 + \frac{T_D}{T_I} + T_D s + \frac{1}{T_I s} \right) \\
&= \ \frac{K_P}{s} \left(s + \frac{T_D s}{T_I} + T_D s^2 + \frac{1}{T_I} \right) \\
&= \ \frac{K_P}{s} \left(T_D s + 1 \right) \left(s + \frac{1}{T_I} \right).
\end{aligned}
$$

To sketch the frequency response of the controller, we note that

$$C(j\omega) \ = \ \frac{K_P}{s} \left(j\frac{\omega}{1/T_D} + 1 \right) \frac{1}{T_I} \left(j\frac{\omega}{1/T_I} + 1 \right).$$

A sketch of the frequency response is shown in Figure 4.44. It illustrates the different ways in which a PID controller improves transient behaviour *and* the steady state accuracy: the former is achieved by the phase lead resulting from the zero at $1/T_D$, whereas the latter comes from the -20 dB/dec slope at low frequencies due to the pole at the origin. Shown in Figure 4.44 by a dashed line is the frequency response that results when the

physically not realizable term $T_D s$ is replaced by $T_D s/(1 + \gamma T_D s)$ with a small positive constant γ as discussed in Chapter 2.
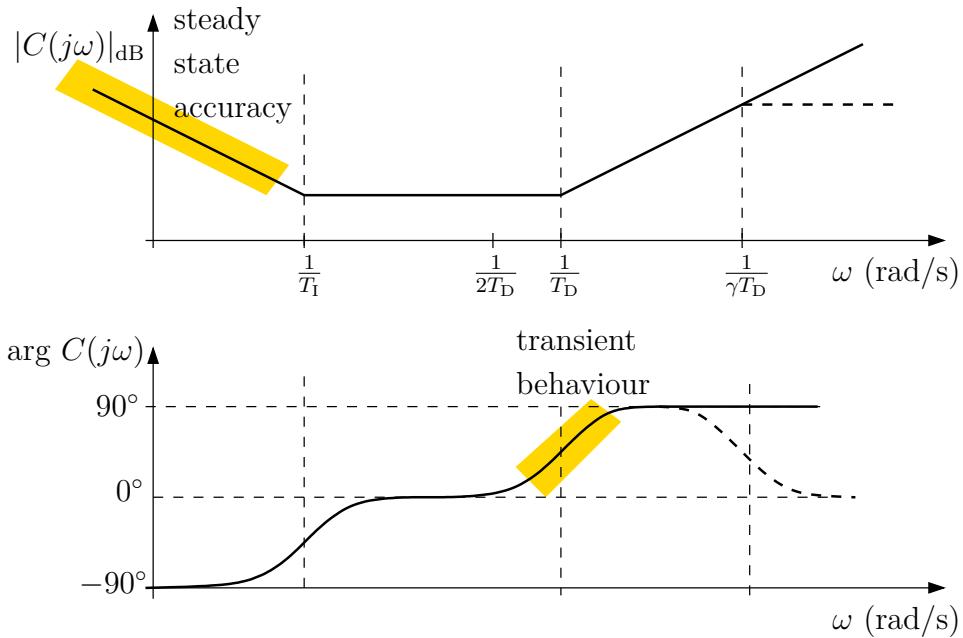
Figure 4.44: Frequency response of a PID controller; solid: ideal PID control, dashed: with extra pole at $-1/\gamma T_D$

The frequency response interpretation of PID control is explored in Exercise 4.13.

Exercises for Chapter 4

Exercises

Problem 4.1

Sketch the Bode plot and the step response of the transfer functions in a) and b). Use the bandwidth to estimate the speed of response.

a)
$$G(s) = \frac{100}{s + 10}$$

b)
$$G(s) = \frac{100}{(s^2 + 0.6s + 1)(s + 10)}$$

c) Sketch the Bode plot of the transfer function

$$G(s) = 10 \frac{(10 + 100s)(s - 10)}{s^2 + 101s + 100}$$

Use MATLAB to verify your sketches.

Problem 4.2

a) Sketch the Bode diagram of a pure time delay, and also the Bode diagram of its first order Padé approximation. Sketch magnitude and phase over the scaled frequency ωT_d, and mark on the frequency axis the points $\omega T_d = 1$ and $\omega T_d = 10$.

b) For $T_d = 1$, use MATLAB to plot the Bode diagram of the time delay together with its 1st and 4th order Padé approximations in the frequency range $0.1 \leq \omega \leq 10$.

MATLAB file: `Problem4_4_BodePlotTimeDelay.mlx`

Problem 4.3

Figure 4.45 shows the frequency response of a plant. Determine its transfer function from the Bode plot.
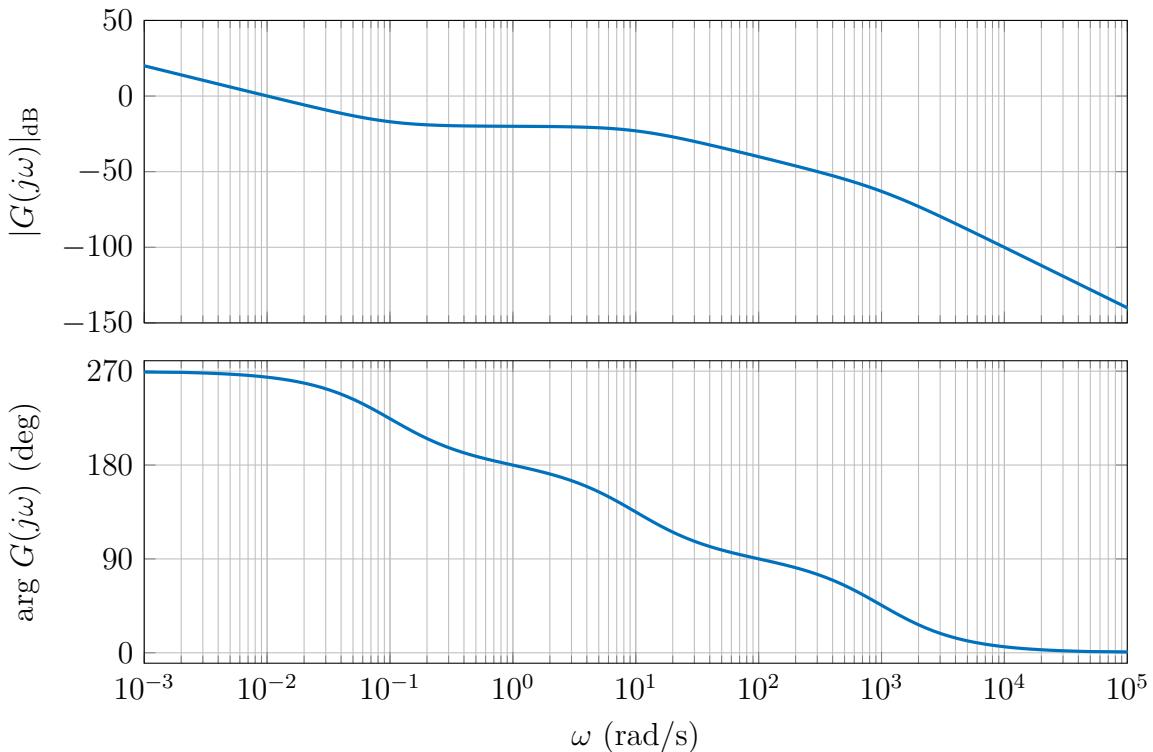
Figure 4.45: Frequency response

Problem 4.4

Assume that the frequency response shown in Figure 4.45 represents the open-loop transfer function $L(s)$ of the feedback loop in Figure 4.46. What is the steady state error when $r(t) = t\sigma(t)$? How can you determine the velocity error constant K_{vel} directly from the Bode diagram?
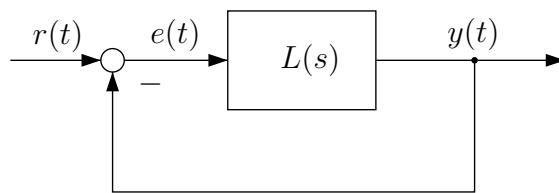
Figure 4.46: Feedback loop

Problem 4.5   *

This problem illustrates how the technique of *analytic continuation* can be used to extend the domain of definition of the Laplace transform. Consider the signal

$$x(t) = Ke^{at}\sigma(t)$$

and assume that $a > 0$.

a)   What is the region of convergence of the Laplace transform

$$X(s) = \int_0^\infty x(t)e^{-st}dt = \frac{K}{s-a}$$

in the sense discussed in Section 1.2? Does it include $X(j\omega)$ on the imaginary axis?

b)   For $K = 1$, show that for a given point s_0 in the complex plane $X(s)$ can be written as

$$X(s) = -\left(\frac{1}{a-s_0} + \frac{s-s_0}{(a-s_0)^2} + \frac{(s-s_0)^2}{(a-s_0)^3} + \cdots\right)$$

by expanding $X(s)$ in a Taylor series about s_0.

c)   Use the ratio test to show that the series converges if

$$|s - s_0| < |a - s_0|$$

What does this imply about the region of convergence, and in particular about convergence of $X(j\omega)$ along the imaginary axis?

Problem 4.6

An integral feedback controller $C(s) = K_P/s$ is to be designed for a plant with transfer function

$$G(s) = \frac{1000}{(s+1)(s+100)}$$

a)   Sketch the Bode diagram of the open-loop transfer function for $K_P = 1$.

b)   With the help of a Nyquist sketch, find the range of values of K_P for which the closed-loop system is stable.

Hint: Sketch only the part of the Nyquist diagram that is required for applying the Nyquist stability test.

MATLAB file: `Problem4_6_NyquistCriterion.mlx`

Problem 4.7

How would a time delay of $T_d = 1$ change the Bode and Nyquist diagrams of Problem 4.6? What about the phase margin and the gain margin?

MATLAB file: `Problem4_7_NyquistCriterionTdelay.mlx`

Problem 4.8

Use MATLAB to plot the Bode and the Nyquist diagram of the system with transfer function

$$G(s) = 2\frac{(s+1)^2}{s^3}$$

Discuss the stability margins when proportional feedback is used to control this plant.

MATLAB file: `Problem4_8_BodeNyquistInfGainMargin.mlx`

Problem 4.9

Consider a plant with transfer function

$$G(s) = \frac{10}{s(s+3)}$$

a) Design a lead compensator to achieve a closed-loop bandwidth $\omega_b \approx 5$ and a phase margin $PM = 70^o$ using the procedure given in the lecture notes.

b) Retune the controller with SISO tool so that the magnitude of the control input in response to a unit step reference input is less than 5 at all times. Find a good trade off between all design requirements.

MATLAB file: `Problem4_9_LeadComp.mlx`

Problem 4.10

In this problem the frequency response of the DC motor will be obtained experimentally and compared with that of the motor model.

a) Consider the transfer function of the motor from input voltage to angular velocity obtained in Problem 1.7.f

$$G(s) = \frac{21}{1.1s + 1}.$$

Use MATLAB to create the Bode plot for this model. What is the corner frequency?

b) In the following the frequency response will be obtained experimentally by applying sinusoidal input signals in a wide frequency range to the motor.

   – Use the SIMULINK template `Model_Problem4_10_BodeExp.slx` to measure data for the frequency response experimentally. The input to the motor is a sinusoidal signal with an amplitude of 2 V and an offset of 3 V, such that the maximum input of 6 V is not exceeded and the motor does not switch direction of rotation (to avoid friction effects).

Table 4.1: Experimental bode plot

	ω $\left(\frac{\text{rad}}{\text{s}}\right)$	T_{Sim} (s)	$\lvert G \rvert$	$\lvert G \rvert_{\text{dB}}$	Φ (°)
	10	20			
	5	20			
	2	25			
Measurements	1	25			
	0.5	50	≈ 20.6	≈ 26.3	≈ -29.5
	0.2	100	≈ 22.9	≈ 27.2	≈ -13.5
	0.1	200	≈ 23.2	≈ 27.3	≈ -6.1

- The frequency ω of the sinusoidal input is set to the variable *frequency*, that you have to assign in the command window.

- For each frequency ω determine the magnitude and the phase of the system. To facilitate that procedure you can use the function `Problem4_10_PlotSinusComparison.m`. Enter `help Problem4_10_PlotSinusComparison` to use the MATLAB help and get more information about this function.

- Use a table like 4.1 to record you measurements. Approximations of some measurements are already given to save time.

c) 🖳 Now the frequency response obtained experimentally will be compared with that of the model. The focus will be first on the magnitude and then on the phase plots.

   i) Display the magnitude and phase information you obtained experimentally in the form of a Bode plot and compare it to the Bode plot of the motor model. For this you can use the function `Problem4_10_BodeComparison.m`.

   ii) Concentrate on the magnitude plot first: If the corner frequency of the experimentally obtained Bode plot does not match that of the model, adjust the model to achieve a better match.

   iii) The motor model in the form (1.34) used here was derived from physical principles, and the numerical values of the model parameters were obtained experimentally in Problem 1.7. Relate the adjustment that was required in (ii) to match the corner frequencies to the physical parameters involved. For which physical parameters is an adjustment meaningful and for which ones not?

d) 🖳 Now compare the phase plots of the experimental and the model frequency response. It was already observed in Problem 1.8 and Problem 2.4 that a time delay is present in the physical setup; this time delay is about 10 ms. When using the function `Problem4_10_BodeComparison.m`, a time delay can be included in the motor model when comparing it with the experimental results. Check if including a time delay leads to a better match of the phase drop in simulation and experiment.

Simulink file: `Model_Problem4_10_BodeExp.slx` Matlab files:

– `Problem4_10_BodeComparison.m`
– `Problem4_10_PlotSinusComparison.m`

Problem 4.11   📆

a) Design a lead compensator to control the angle of the DC motor. The control objectives are

– a rise time of $t_r = 0.45$ s (or faster)

– and a phase margin of (at least) $PM = 50\,°$.

Use the transfer function of the motor from input voltage to angular velocity derived in Problem 1.7.f

$$G(s) = \frac{21}{1.1s + 1}.$$

and add a pole at the origin because the angle rather than the speed is to be controlled. Use the Matlab command '`sisotool(G)`'.

i) Calculate the controller parameters required to meet the objectives. What peak overshoot can you expect? Check the closed-loop step response of your design using the appropriate functions of the '`sisotool`'.

ii) 📆 Validate your controller from (i) experimentally; use the Simulink "`Model_Problem4_11_Lead.slx`" template for this purpose. Apply the reference input $r(t) = 10\sigma(t)$ rad. Choose 10 ms as sampling time.

iii) Comment on the steady state error that can be observed in the experiments, and compare it with that expected from the model. What would be the effect of a step disturbance at plant input, and which physical effects could here be interpreted as input disturbances?

iv) 📆 Now use the Simulink template "`Model_Problem4_11_LeadDist.slx`" and perform an experiment to measure the steady state error that results from a step input disturbance. Set the final value of the disturbance step input to 1 rad.

b) In order to reduce the steady state errors resulting from input disturbances, an additional lag compensator is to be designed.

i) Design an additional lag compensator that can be used in series with the lead compensator, such that the steady state error is reduced by a factor of 10. Make sure that the lag compensator does not significantly change the dynamic behaviour achieved by the lead compensator. Use the Matlab command '`sisotool(G,C)`'. Check the closed-loop reference and disturbance step responses of your design using the appropriate functions of the '`sisotool`'.

ii) 🖳 Use the SIMULINK template `Model_Problem4_11_LeadLag.slx` with and without an additional input disturbance of 1 rad and perform experiments to validate the effect of the lead-lag compensator.

iii) How quickly does the lead-lag compensator suppress a step input disturbance? By tuning your lag compensator, how fast can you make the disturbance rejection without significantly changing the tracking behaviour?

SIMULINK files:

- `Model_Problem4_11_Lead.slx`
- `Model_Problem4_11_LeadDist.slx`
- `Model_Problem4_11_LeadLag.slx`

Problem 4.12

Assume that for a given plant a controller is designed as

$$C_0(s) = K_p \frac{1 + 0.2s}{1 + 0.05s}$$

a) Sketch the Bode and the Nyquist plots of the controller $C_0(s)$. Mark the values of magnitude and phase of $C_0(j\omega)$ for $\omega = 0$ and $\omega \to \infty$ on the Nyquist plot. What is such a controller called?

b) The controller from (a) is not physically realisable. To implement the controller an additional pole was added, resulting in a controller with transfer function

$$C(s) = C_0(s) \frac{1}{T_f s + 1}$$

What should T_f be? Sketch the Bode and Nyquist plot of $C(s)$. Mark the values of magnitude and phase of $C(j\omega)$ for $\omega = 0$ and $\omega \to \infty$.

Problem 4.13

In this problem the heuristically tuned PID controller from Exercise 2.5 is compared with the lead-lag compensator designed in Exercise 4.11.

a) Plot a Bode diagram of the PID controller that was designed in Exercise 2.5, and also a Bode diagram of the open-loop transfer function of the corresponding angle control loop. What effect do changes in T_I and T_D have on these frequency responses?

b) Compare the Bode diagram of the nominal PID controller with that of the lead-lag compensator designed in Exercise 4.11. Do the same for the corresponding open-loop transfer functions.

c) Compare the simulated closed-loop step responses obtained with the lead-lag compensator and the PID controller, respectively, and comment on your observations.

Problem 4.14

Assume that a lead-lag compensator was designed in order to control the DC motor's angle. The motor is modeled by the transfer function

$$G(s) = \frac{21}{1.1s^2 + s}$$

and controlled by the lead-lag compensator

$$C(s) = C_{\text{lead}}(s)C_{\text{lag}}(s) = K_p\frac{1 + T_L s}{1 + \alpha T_L s}\frac{s + T_{\text{Llag}}}{s + T_{\text{Llag}}\alpha_{\text{lag}}} = 0.56\frac{1 + 0.29s}{1 + 0.061s}\frac{s + 0.3125}{s + 0.03125} \ .$$

The design goals are a rise time $t_r = 0.45$ s and phase margin $PM = 50°$. The computed parameters are $K_p = 0.56$, $T_L = 0.29$, $\alpha = 0.21$, $T_{\text{Llag}} = 0.3125$ and $\alpha_{\text{lag}} = 0.1$.

The MATLAB tool Systune is used in order to tune a similar lead-lag compensator automatically.

a) At first, the given system is to be analyzed.

   i) Open the MATLAB script "Problem_Systune.m". Complete the lead-lag compensator by adding the lag part with tunable parameters and with initial values according to the manually computed compensator.

   ii) Evaluate section 1) and 2). Plot the system's response for a step input at r and the bode diagram of the system's open-loop response. Determine the phase margin PM and cross-over frequency ω_c. Does the controller reject step disturbances entering the system at d?
   Hints: Access a single transfer function by `T('out','in')`. Cut the system T open by adjusting `T.Blocks.X.Open`.

b) The system $T0$ is now supposed to be tuned automatically for better input disturbance rejection while maintaining its original performance goals. The tuning goal for rejection is already defined.

   i) Familiarize yourself with the goals `TuningGoal.Tracking` and `TuningGoal.Margins` class in the MATLAB help. Add a tuning goal of each kind to section 3) with a target phase margin $PM = 50°$ and an arbitrary small gain margin $GM \approx 5$. Chose appropriate values for the response time and steady state error. The peak error can be left at its default value.

   b.ii) Evaluate section 4). Familiarize with the function Systune in the MATLAB help. What is the effect of the option `RandomStart`? Try to explain why the option seems to be necessary with respect to the information returned by Systune in the command line.

c) The tuned system $T1$ returned by Systune is to be compared to the manually tuned lead-lag compensator.

  i) Plot the step responses for the inputs r and d for both systems $T0$ and $T1$. What is the tuned system's rise time?

  ii) Plot the bode diagram of the tuned system. Are the requirements for phase margin and response time met?

Problem 4.15

Figure 4.47 shows the frequency response of a plant with the transfer function

$$G(s) = \frac{100}{s(s^2 + s + 1000)} \quad .$$

a) Is the open-loop plant $G(s)$ stable?

b) Sketch the Nyquist plot for both

  i) positive frequencies; mark the points A, B and the point C where $\omega \to \infty$ and

  ii) negative frequencies.

c) How is the Nyquist plot closed at infinity?
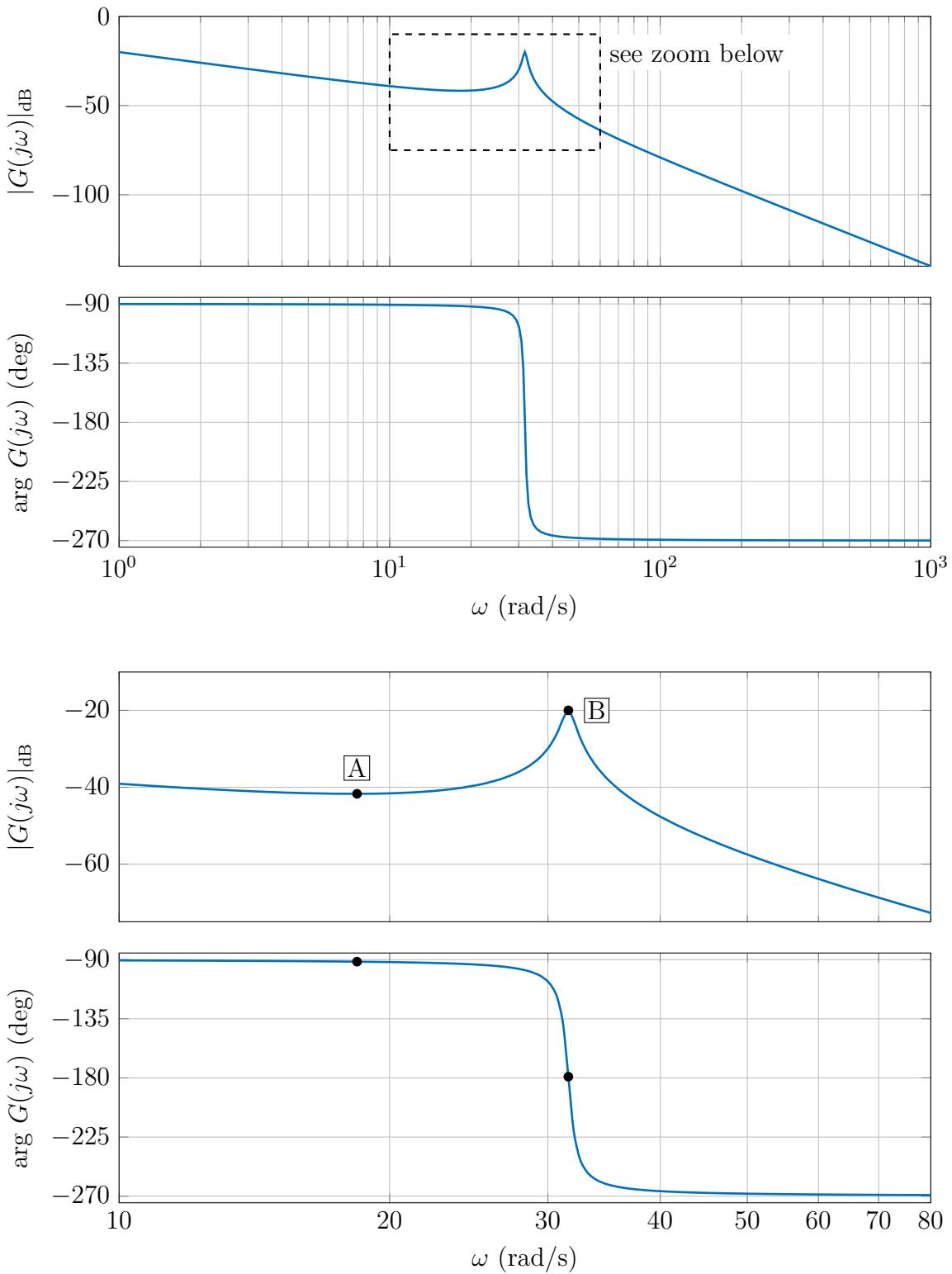
Figure 4.47: Frequency response of $G(s)$

Problem 4.16

Consider the closed-loop system in Figure 4.48 with the transfer function $G(s)$ from Problem 4.15 and $C(s) = K_P$.
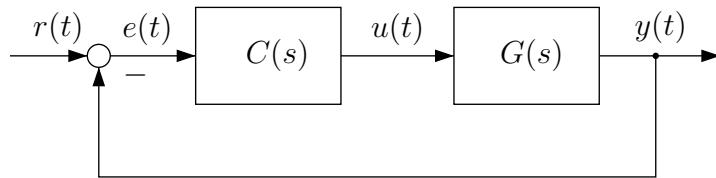
Figure 4.48: Feedback loop

a) Considering your answer from Problem 4.15 (b) and (c), find the range of values of K_P for which the closed-loop system is stable.

b) How can one determine from the Bode diagram in Problem 4.15 whether the loop is stable?

c) Sketch the root locus plot and explain your answer to part (a).

Problem 4.17

Consider the closed-loop system in Figure 4.49 with the transfer function

$$G(s) = 10000 \cdot \frac{s + 10}{s^2(s^2 + 0.2s + 10000)} \quad .$$
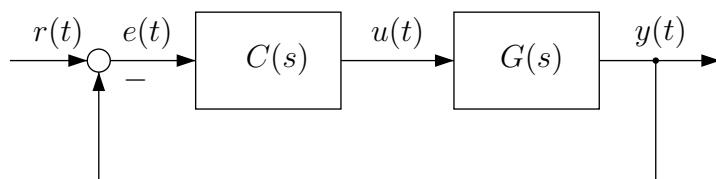
Figure 4.49: Feedback loop

a) Sketch the Bode diagram of the transfer function G(s).

b) Determine a gain K_P with $C(s) = K_P$ such that a closed-loop bandwidth of $\omega_b = 1\text{rad/s}$ is achieved. Is the closed-loop system stable for $C(s) = K_P$?

c) Determine a phase lead compensator $C(s) = C_1(s)$ such that a closed-loop phase margin of $PM = 60°$ is achieved without affecting the intended closed-loop bandwidth from b) significantly.

d) Determine approximately the gain of the controller $C_1(s)$ at $\omega = 100 rad/s$. What follows for the stability of the closed loop? Give reasons.

e) With the help of a compensator $C_2(s) = \frac{K_2}{s+a}$ the gain of $L(s) = G(s)C_1(s)C_2(s)$ is to be decreased by $20dB$ at $\omega = 100 rad/s$ without affecting significantly the intended closed-loop bandwidth from b. Determine the parameters K_2 and a of C_2. What is the consequence regarding the closed-loop system?

Chapter 5

A Brief Look at Digital Control

5.1 Digital Control

All controllers discussed so far were modelled by differential equations or transfer functions. These representations imply that the input and output signals of such controllers are *continuous-time signals*, i.e. they are defined on the whole time axis. From an implementation point of view, this assumes that controllers are implemented using analog electronic devices. Today, most controllers are implemented digitally - the control law is realized as code on a microprocessor. Microprocessors cannot handle continuous-time signals directly, they can only process sequences of numbers - these are called *discrete-time signals*. In this chapter we will give a brief introduction to the topic of digital control.
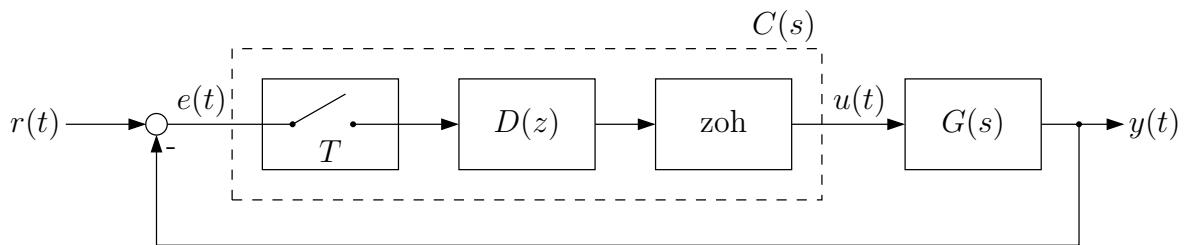
Figure 5.1: Continuous-time controller design

The block diagram in Figure 5.1 shows the structure of a control loop with a digital controller. The plant - represented by the transfer function $G(s)$ - has continuous-time input and output signals. An analog-to-digital (A/D) converter takes samples of the control error $e(t)$ repeatedly every T seconds, where T is the *sampling period*, and converts the sampled values into binary numbers. A control algorithm - denoted by $D(z)$ - takes the sampled control error as input and generates a sequence of values, which is then converted into a continuous-time control signal $u(t)$ by a digital-to-analog (D/A) converter and a zero order hold (zoh) element. We assume that the operation of the A/D and D/A

converters and of the sample and hold devices are synchronized. The components inside the dashed box in Figure 5.1 together form a digital controller.

The operation of a digital controller involves two types of discretization: a discretization of time, and a discretization of signal values due to the conversion into binary numbers with finite word length. In this chapter we will concentrate on the consequences of the discretization of time. We will assume that the resolution is sufficiently high such that quantization effects can be neglected.
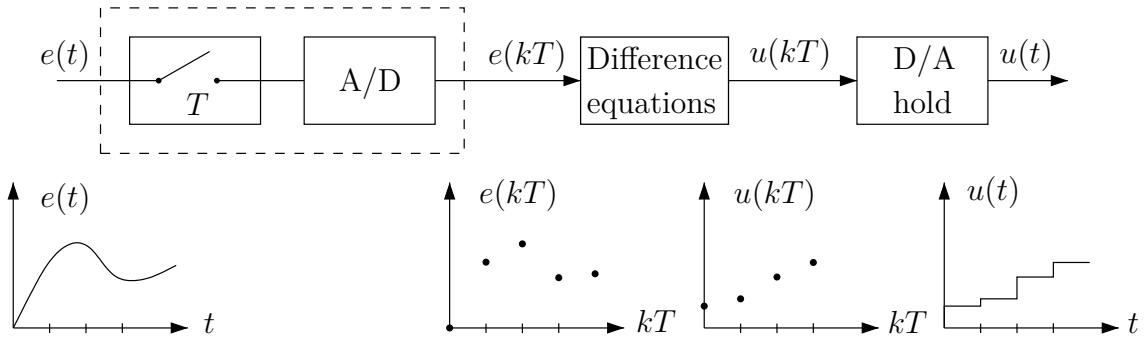
Figure 5.2: Continuous-time and discrete-time signals in the loop

The conversion between continuous-time and discrete-time signals is shown again in more detail in Figure 5.2. The signals entering and leaving the controller are continuous-time signals, they represent plant input $u(t)$ and output $y(t)$. The control algorithm itself operates on discrete-time signals, i.e. sequences of numbers $u(kT)$ and $e(kT)$, which are the values of the continuous-time plant input and output sampled at time instants $t = kT$, where $k = 0, 1, \ldots$

Difference Equations

The block labelled "difference equations" in Figure 5.2 represents the dynamic behaviour of the controller. The input to this block is the sequence of numbers $e(kT)$, and the output is the sequence of numbers $u(kT)$, where $k = 0, 1, \ldots$ The factor T in the time argument reflects the fact that these sequences are generated by sampling continuous-time signals every T seconds. The processor that generates the control action is only sequentially processing incoming numbers - the sampling period itself has no effect on the sequence of numbers that is produced as output. (We assume however that the sampling period is longer than the computation time required for generating the next output value). We will sometimes use a simplified notation and write $y(k)$, $e(k)$ and $u(k)$ instead of $y(kT)$, $e(kT)$ and $u(kT)$, but we will return to the latter notation when we are interested in the interaction between discrete-time signals and continuous-time systems. Formally we let

the integer variable k run from $-\infty$ to $+\infty$, but - similar to our treatment of continuous-time systems - we will assume that all signals $x(k)$ satisfy

$$x(k) = 0, \quad k < 0.$$

A linear difference equation has the form

$$y(k) + a_1 y(k-1) + \ldots + a_n y(k-n) = b_0 u(k) + b_1 u(k-1) + \ldots + b_n u(k-n) \quad (5.1)$$

If $b_0 \neq 0$ the output $y(k)$ at time k depends on the input $u(k)$ at the same time - the system responds instantaneously. Physical systems cannot do this, and when a discrete-time model is used to represent a physical plant we will usually assume $b_0 = 0$. However, models with $b_0 \neq 0$ are used as an approximation - e.g. for controllers - when the response is fast compared with the sampling period.

A comment is also in order about the number of samples involved in the above difference equation. The integer n is used to denote the number of past input and output samples that have an effect on the present value of the output - equation (5.1) is called a nth order difference equation. In general, the number of past input samples and output samples that determine the present output need not be the same, in that case we let n denote the larger one of these values and set the coefficients that are not needed to zero.

Example 5.1

The use of difference equations for describing discrete-time dynamic behaviour is now illustrated with an example. Thus, consider a system governed by

$$y(k) + 0.5y(k-1) = u(k-1) \quad (5.2)$$

and assume that the input signal is $u(k) = \sigma(k)$, where $\sigma(k)$ is the *discrete-time unit step function*

$$\sigma(k) = \begin{cases} 1, & k \geq 0 \\ 0, & k < 0 \end{cases} \quad (5.3)$$

Because input and output are zero for negative times we have $y(0) = 0$. The solution for $k > 0$ is obtained by computing

$$y(k) = -0.5y(k-1) + u(k-1)$$

successively, this yields

k	$u(k)$	$y(k)$
0	1	0
1	1	1
2	1	0.5
3	1	0.75
4	1	0.625
5	1	0.6875
$\dots$	$\dots$	$\dots$

The output is oscillating and appears to converge towards a value between 0.625 and 0.6875. If the difference equation (5.2) is changed to

$$y(k) - 0.5y(k-1) = u(k-1) \tag{5.4}$$

we have to solve

$$y(k) = 0.5y(k-1) + u(k-1)$$

yielding

k	$u(k)$	$y(k)$
0	1	0
1	1	1
2	1	1.5
3	1	1.75
4	1	1.875
5	1	1.9375
$\dots$	$\dots$	$\dots$

Now the solution is monotonically increasing and appears to converge to a value around 2.

Digital Implementation of PID Control - Tustin approximation

We will now consider a digital implementation of a PID control law

$$U(s) = C(s)E(s) = K_P \left(1 + \frac{1}{T_I s} + T_D s \right) E(s)$$

or in time domain

$$u(t) = K_P \left(e(t) + \frac{1}{T_I} \int_0^t e(\tau)d\tau + T_D \dot{e}(t) \right).$$

We assume that a continuous-time PID controller $C(s)$ has been designed that meets the design specifications. The task is then to find a difference equation such that the system

in Figure 5.2 - which is a continuous-time system due to the presence of the sampler and the hold - approximates $C(s)$ sufficiently well so that the design specifications are also met with a digital controller. There are various ways of constructing discrete-time approximations of continuous-time transfer functions; here we will use a method known as *Tustin approximation.*

We begin with the discrete-time approximation of one of the basic building blocks of the PID controller, an integrator. Thus, consider a pure integral controller

$$u(t) = \int_0^t e(\tau)d\tau. \tag{5.5}$$

At time $t = kT + T$ we have

$$u(kT + T) = \int_0^{kT+T} e(\tau)d\tau = \int_0^{kT} e(\tau)d\tau + \int_{kT}^{kT+T} e(\tau)d\tau$$

or

$$u(kT + T) = u(kT) + \int_{kT}^{kT+T} e(\tau)d\tau.$$
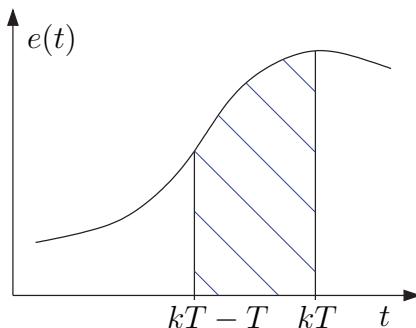
Figure 5.3: Sampling an integrator

The second term on the right hand side represents the area under $e(t)$ marked in Figure 5.3. The Tustin method approximates the function $e(t)$ between the last two samples by a straight line. This yields the difference equation

$$u(kT + T) = u(kT) + \frac{T}{2}\left(e(kT) + e(kT + T)\right) \tag{5.6}$$

which approximates the relationship (5.5).

Forward Shift Operator and Discrete-Time Transfer Functions

Working with discrete-time systems can be considerably simplified when difference equations are transformed into frequency domain - just as the Laplace transform simplifies working with continuous-time systems. Here we will introduce a discrete-time equivalent

to the Laplace transform simply by defining a *forward shift operator z* as its complex variable: if $X(z)$ is the transform of the discrete-time signal $x(kT)$, then $zX(z)$ is the transform of $x(kT+T)$. Similarly, $z^{-1}X(z)$ is the transform of $x(kT-T)$. The underlying idea is that of the *z-transform*, which will be studied in the course *Control Systems 2*. Here we will use the shift operator and discrete-time transfer functions only as computational tools when working with difference equations; the results will then be transformed back into the discrete time domain.

Applying the transformation defined above to the difference equation (5.6) for $u(kT)$ yields

$$zU(z) = U(z) + \frac{T}{2}(E(z) + zE(z))$$

or

$$(z - 1)U(z) = \frac{T}{2}(z + 1)E(z).$$

We can then define the discrete-time transfer function from e to u as

$$\frac{U(z)}{E(z)} = \frac{T}{2} \cdot \frac{z + 1}{z - 1}$$

This is a discrete-time approximation of the continuous-time transfer function $C(s) = 1/s$.

Next we consider the building block of derivative action, a differentiator

$$u(t) = \dot{e}(t).$$

Here the role of u and e are reversed compared with integration. Exchanging u and e in (5.6) yields

$$e(kT + T) = e(kT) + \frac{T}{2}(u(kT) + u(kT + T))$$

which in frequency domain leads to

$$\frac{U(z)}{E(z)} = \frac{2}{T} \frac{z - 1}{z + 1}$$

This is a discrete-time approximation of the continuous-time transfer function $C(s) = s$.

The proportional term $u(t) = K_P e(t)$ becomes in discrete time

$$u(kT) = K_P e(kT)$$

or

$$U(z) = K_P E(z).$$

Combining these terms, we obtain the discrete-time approximation of a PID control law

$$D(z) = \frac{U(z)}{E(z)} = K_P \left(1 + \frac{T}{2T_I} \cdot \frac{z + 1}{z - 1} + \frac{2T_d}{T} \cdot \frac{z - 1}{z + 1} \right)$$

which can be transformed back into a difference equation in $e(kT)$ and $u(kT)$.

The construction of a discrete-time PID controller shows that, more generally, making the substitution

$$s = \frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} \tag{5.7}$$

in every term of a controller transfer function $C(s)$ that contains s, yields a discrete-time transfer function $D(z)$ that is based on the above trapezoidal approximation of an integral.

Static Gain

To see how the static gain of a discrete-time system with transfer function $D(z)$ can be determined, consider the difference equation

$$y(k) + a_1 y(k - 1) + \ldots + a_n y(k - n) = b_0 u(k) + b_1 u(k - 1) + \ldots + b_n u(k - n).$$

In steady state, the signals $u(k)$ and $y(k)$ are constant, i.e.

$$u(k) = u(k - 1) = u(k - 2) = \ldots = \bar{u}, \qquad y(k) = y(k - 1) = y(k - 2) = \ldots = \bar{y}.$$

Thus, in steady state we have

$$\bar{y} + a_1 \bar{y} + \ldots + a_n \bar{y} = b_0 \bar{u} + b_1 \bar{u} + \ldots + b_n \bar{u}$$

or

$$(1 + a_1 + \ldots + a_n)\bar{y} = (b_0 + b_1 + \ldots + b_n)\bar{u}.$$

The static gain of this system is therefore

$$\frac{\bar{y}}{\bar{u}} = \frac{b_0 + b_1 + \ldots + b_n}{1 + a_1 \ldots + a_n}.$$

In terms of the discrete-time transfer function

$$D(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_n z^{-n}}{1 + a_1 z^{-1} + \ldots + a_n z^{-n}} = \frac{b_0 z^n + b_1 z^{n-1} + \ldots + b_n}{z^n + a_1 z^{n-1} + \ldots + a_n}$$

the static gain is given by $D(1)$.

Example 5.1 (continued)

Returning to Example 5.1, we have

$$G(z) = \frac{z^{-1}}{1 + 0.5 z^{-1}} = \frac{1}{z + 0.5}$$

and thus $G(1) = 0.67$. For the system in (5.4) we obtain

$$G(z) = \frac{1}{z - 0.5}$$
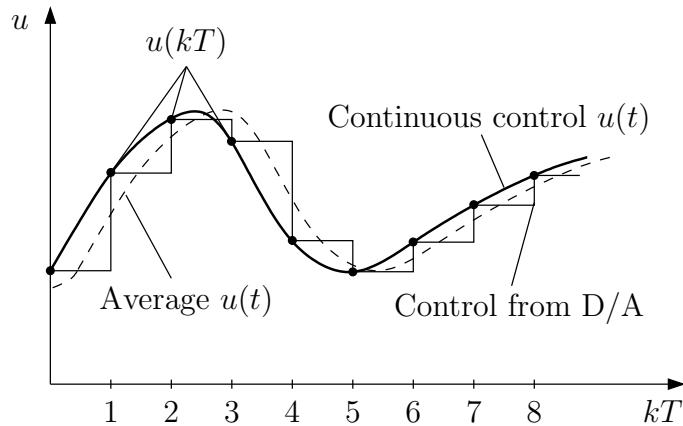
Figure 5.4: Introduction of a time delay through the hold operation

and a static gain $G(1) = 2$, confirming our guess from the recursive solution.

Choice of Sampling Frequency

If a discrete-time controller $D(z)$ is implemented in the control system in Figure 5.1, the value of the control signal $u(t)$ is held constant by the zero order hold block connected to the controller output until the next value is available, so that the continuous-time control input consists of steps as shown in Figure 5.2. Due to the fact that physical systems have a limited bandwidth, the effect of this control input on the plant will have the same effect as a low-pass filtered version of this step-shaped signal. This is illustrated in Figure 5.4, where it can be seen that the hold operation introduces an effective time-delay of approximately $T/2$. Thus, while the idea was to approximate the behaviour of the continuous-time controller $C(s)$ by a discrete-time controller, the discretized controller $D(z)$ is actually emulating the effect of a controller

$$\tilde{C}(s) \approx C(s)e^{-\frac{T}{2}s} \approx C(s)\frac{2/T}{s + 2/T}.$$

Taking this time delay into account when designing the continuous-time controller gives a reasonable prediction of the effect of the zero-order hold when the sampling rate is slower than $20\omega_b$. To keep this effect small, the sampling frequency $\omega_s = 2\pi/T$ should be much higher than the system bandwidth ω_b - experience suggests that the sampling frequency should be at least $20 \sim 30\omega_b$. This point is illustrated in Exercise 5.2.

Exercises for Chapter 5

Exercises

Problem 5.1

The lead compensator designed in Exercise 4.9 is to be implemented digitally.

$$C(s) = 4\,\frac{s+1}{s+6}$$

a) Find a difference equation that approximates the dynamic behaviour of the controller when input and output are sampled with sampling period T. For this purpose, use the Tustin approximation to determine a discrete-time transfer function $D(z)$.

b) Compare the static gain of the controller $C(s)$ with that of its discrete-time approximation $D(z)$.

c) Use your result from (a) to compute a discrete-time approximation of the controller designed in Exercise 4.9 with sampling period $T = 0.1$. Check your result with the MATLAB function `c2d` (use the option 'tustin').

MATLAB file: `Problem5_1_TustinApprox.mlx`

Problem 5.2

Simulate the step response of the closed-loop system in Exercise 4.9 both with a continuous-time and a sampled-data controller.

a) Compare the step responses obtained with continuous-time and sampled-data controller when $T = 0.1$.

b) Repeat the comparison with $T = 0.01$. Explain the effect of the sampling rate on the dynamic behaviour of the closed-loop system.

c) Repeat the comparison with $T = 0.1$, but replace the plant model $G(s)$ with $G(s)e^{-T/2}$ when using the continuous-time controller.
(*Hint: Note that you cannot simulate the time delay directly.*)

MATLAB file: `Problem5_2_TustinSimulation.mlx` SIMULINK files:

- `Model_Problem5_2_TustinSimulation_C.slx`
- `Model_Problem5_2_TustinSimulation_D.slx`

Problem 5.3 ☑

In this problem the effect of undersampling on the performance of a digital control loop is investigated experimentally. Consider again the transfer function of the DC motor

$$G(s) = \frac{21}{1.1s^2 + s}.$$

from input voltage to shaft angle, that was obtained in Problem 2.3.a.

a) Convert the transfer function of the lead compensator designed in Problem 4.11.a into a transfer function of a discrete-time system with the help of the MATLAB function `c2d` (use the option `'tustin'`). Use three different sampling times $T_0 = 10$ ms, $T_1 = 30$ ms and $T_2 = 100$ ms.

b) Build SIMULINK models and simulate four closed-loop step responses: one with the continuous-time controller and three using discrete-time controllers with sampling times T_0, T_1 and T_2, respectively. Use in each case the continuous-time model of the motor given above. Set the final value of the reference step to 2π rad, and compare the responses.

Hint: You can place all control loops into a single SIMULINK diagram and just replace the continous-time controller with the respective discrete-time counterpart (use the `'discrete transfer function'` block and also set the correct sampling time in its properties).

c) Simulate the closed-loop step response with the continuous controller as in (b), but replace the plant model $G(s)$ with

$$G(s)e^{-\frac{T_i}{2}s}$$

where T_i should correspond to the sampling time. Compare the result with the step responses obtained by using the discretized controllers in (b). Use the SIMULINK-Template for comparing the different sampling times.

d) ☑ Use the given SIMULINK templates to verify the results from (b) and (c) experimentally.

Simulink files:

- – Model_Problem5_3_Sampling10ms.slx
- – Model_Problem5_3_Sampling30ms.slx
- – Model_Problem5_3_Sampling100ms.slx
- – Model_Problem5_3_SimLeadContinuousDiscreteComparison.slx

Appendix A

Solutions to Exercises

A.1 Solutions for Chapter 1

Problem 1.1 (Laplace transformation)

We have for a unit ramp signal $x(t) = t\sigma(t)$

$$\mathcal{L}[t\sigma(t)] = \int_0^\infty t\sigma(t)e^{-st}dt = \int_0^\infty te^{-st}dt$$

and using integration by parts

$$\mathcal{L}[t\sigma(t)] = -\frac{1}{s}e^{-st} \ t\Big|_0^\infty - \int_0^\infty -\frac{1}{s}e^{-st} \ 1 \ dt$$

$$= 0 - 0 + -\frac{1}{s^2}e^{-st}\Big|_0^\infty = 0 - (-\frac{1}{s^2}) = \frac{1}{s^2}.$$

This result can also be found in the Laplace Table 1.1.

Problem 1.2 (Convolution Theorem)

From

$$x_1(t) * x_2(t) = \int_0^t x_1(\tau)x_2(t-\tau)d\tau$$

we obtain

$$\mathcal{L}[x_1(t) * x_2(t)] = \int_0^\infty \left[\int_0^t x_1(\tau)x_2(t-\tau)d\tau\right]e^{-st}dt$$

$$= \int_0^\infty \int_0^t x_1(\tau)x_2(t-\tau)e^{-st}d\tau dt.$$

Now the order of integration is changed. The effect of this change is shown in Figure A.1. The left hand side of the figure shows the integration order as given above: the inner integral is with constant t over τ, and the outer integral over t from 0 to ∞. Integration over the same area can be obtained with exchanged order of integration: first integrate with constant τ over t from τ to ∞ and then over τ from 0 to ∞.

Thus we obtain

$$\mathcal{L}[x_1(t) * x_2(t)] = \int_0^\infty \int_\tau^\infty x_1(\tau)x_2(t-\tau)e^{-st}dt d\tau$$
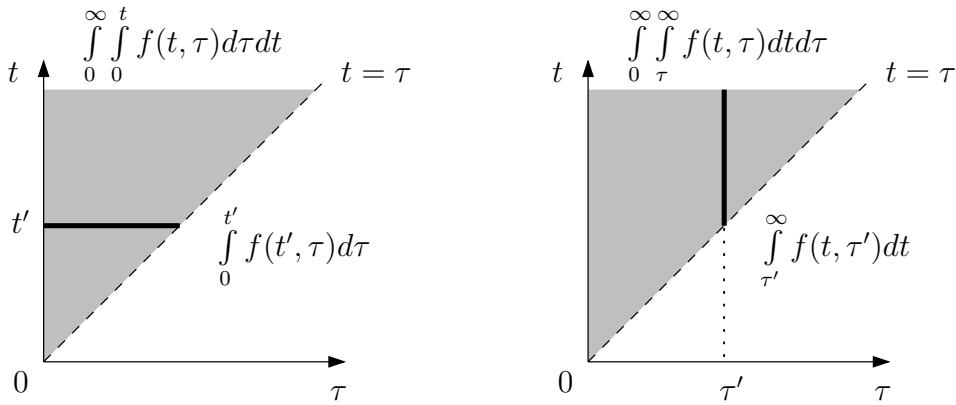
Figure A.1: Bode plot and step response of the system in (b)

By including the factor $e^{-s\tau}e^{s\tau}$ and substituting $\lambda = t - \tau$ the convolution integral can be expressed as

$$
\begin{aligned}
\mathcal{L}[x_1(t) * x_2(t)] &= \int_0^\infty \int_\tau^\infty x_1(\tau)x_2(t-\tau)e^{-st}e^{-s\tau}e^{s\tau}dt\,d\tau \\
&= \int_0^\infty \int_\tau^\infty x_1(\tau)e^{-s\tau}x_2(t-\tau)e^{-s(t-\tau)}dt\,d\tau \\
&= \int_0^\infty \int_0^\infty x_1(\tau)e^{-s\tau}x_2(\lambda)e^{-s\lambda}d\lambda\,d\tau \\
&= \int_0^\infty x_1(\tau)e^{-s\tau}d\tau \int_0^\infty x_2(\lambda)e^{-s\lambda}d\lambda \\
&= X_1(s)X_2(s).
\end{aligned}
$$

Note that from $t = \lambda + \tau$ follows $dt = d\lambda$, and for $t = \tau$ the lower bound of the integral becomes $\lambda = 0$.

Problem 1.3 (Step response 1st order differential equation)

a)   Apply the Laplace transformation to obtain

$$
sY(s) + a_0 Y(s) = b_0 U(s) \quad \Rightarrow \quad G(s) = \frac{Y(s)}{U(s)} = \frac{b_0}{s + a_0}
$$

With $U(s) = 1/s$ the transformed step response is then

$$
Y(s) = G(s)U(s) = \frac{b_0}{s(s + a_0)},
$$

and can be transformed into time domain by expansion into partial fractions

$$
Y(s) = \frac{A}{s + a_0} + \frac{B}{s} \quad \text{with the residues}
$$

$$A = \frac{b_0}{s(s+a_0)} \cdot (s+a_0) \bigg|_{s=-a_0} = \frac{-b_0}{a_0}; \quad B = \frac{b_0}{s(s+a_0)} \cdot s \bigg|_{s=0} = \frac{b_0}{a_0}$$

The partial fractions are then transformed back

$$Y(s) = \frac{-b_0/a_0}{s+a_0} + \frac{b_0/a_0}{s} \quad \Rightarrow \quad y(t) = \left(-\frac{b_0}{a_0}e^{-a_0t} + \frac{b_0}{a_0}\right)\sigma(t) = \frac{b_0}{a_0}\left(1 - e^{-a_0t}\right)\sigma(t)$$

b) The sketch can be obtained by the superposition of the two terms. See Figure A.2.
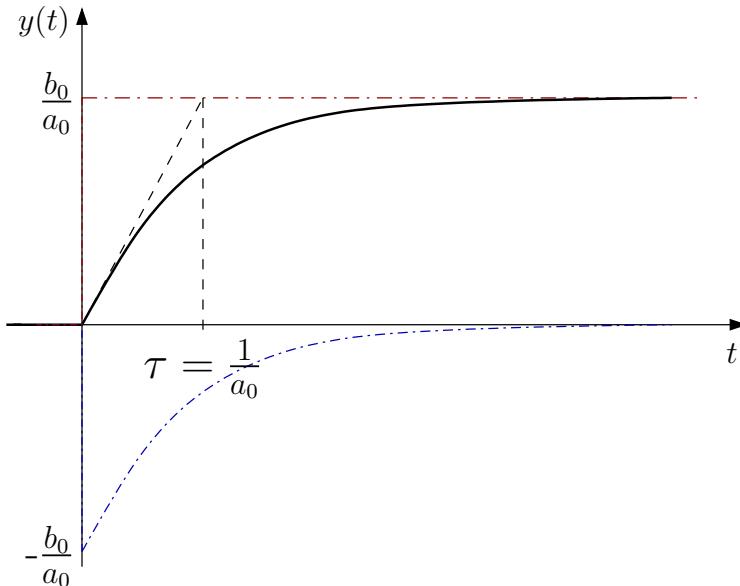
Figure A.2: Step response of first order system

c) The slope is calculated as $\dot{y}(t) = b_0 e^{-a_0t} \quad \Rightarrow \quad \dot{y}(0) = b_0$.

d) From the tangent with slope b_0 and the static gain $k_s = b_0/a_0$ one obtains the time constant $\tau = 1/a_0$.

e) See (b)

MATLAB file: `Sol_Problem1_3_1stOrderStepResponse.mlx`

Problem 1.4 (Step response 2nd order differential equation)

a) From

$$\ddot{y}(t) + 3\dot{y}(t) + 2y(t) = 2u(t) \quad \Longleftrightarrow \quad s^2 Y(s) + 3sY(s) + 2Y(s) = 2U(s)$$

we have

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

The transformed step response $Y(s)$ is

$$Y(s) = G(s)U(s) = \frac{2}{s(s+1)(s+2)} = \frac{A}{s} + \frac{B}{s+1} + \frac{C}{s+2} \quad \text{with residues}$$

$$A = \left.\frac{2}{(s+1)(s+2)}\right|_{s=0} = 1; \quad B = \left.\frac{2}{s(s+2)}\right|_{s=-1} = -2; \quad C = \left.\frac{2}{s(s+1)}\right|_{s=-2} = 1$$

Thus

$$Y(s) = \frac{1}{s} - \frac{2}{s+1} + \frac{1}{s+2} \quad \Rightarrow \quad (1 - 2e^{-t} + e^{-2t})\sigma(t)$$

b) The sketch can be obtained by the superposition of the three signals. See Figure
    A.3.
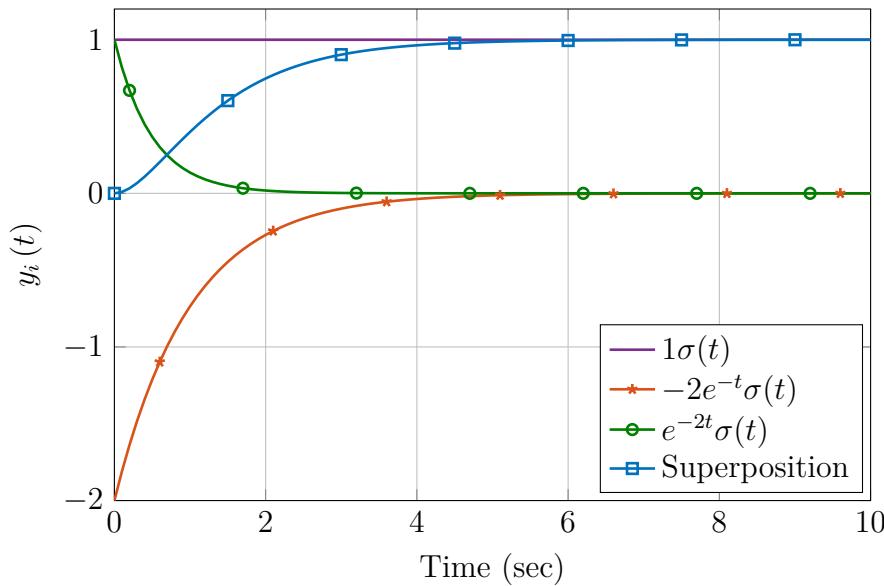
Figure A.3: Step response of second order system

c) Both, the static gain of the system and the speed of the response can be changed
    by varying the parameters. There can even be an oscillation if the poles become
    complex. The relevant MATLAB commands and the step response can be found in
    `"Sol_Problem1_4_2ndOrderStepResponse.mlx"`.

Problem 1.5 (Modelling the DC motor)

a) Choosing the state vector $x = \begin{bmatrix} \dot\theta & i \end{bmatrix}^T$, one obtains

$$\frac{d\dot\theta}{dt} = -\frac{b_v}{J}\dot\theta + \frac{K_m}{J}i$$
$$\frac{di}{dt} = -\frac{K_g}{L}\dot\theta - \frac{R}{L}i + \frac{1}{L}v$$

and thus

$$A = \begin{bmatrix} -\frac{b_v}{J} & \frac{K_m}{J} \\ -\frac{K_g}{L} & -\frac{R}{L} \end{bmatrix}; \qquad b = \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix}; \qquad c = \begin{bmatrix} 1 & 0 \end{bmatrix}; \qquad d = 0$$

b) Substituting the state-space model from Task 1.5.a) in (1.17) yields

$$G(s) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} s + \frac{b_v}{J} & -\frac{K_m}{J} \\ \frac{K_g}{L} & s + \frac{R}{L} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} = \frac{\frac{K_m}{JL}}{s^2 + \left(\frac{b_v}{J} + \frac{R}{L}\right)s + \frac{K_m K_g + b_v R}{JL}}$$

c) If the angle shaft and not the angular velocity is the output of the system, the transfer function is modifed to

$$G_\theta(s) = G(s)\frac{1}{s} = \frac{\frac{K_M}{JL}}{s^3 + \left(\frac{b_v}{J} + \frac{R}{L}\right)s^2 + \frac{K_m K_g + b_v R}{JL}s}$$

because the shaft angle $\theta(t)$ can be obtained by simply integrating the angular velocity $\dot\theta(t)$, which in Laplace domain is done by multiplying by $\frac{1}{s}$. A possible new state vector is $\begin{bmatrix} \theta & \dot\theta & i \end{bmatrix}$ and the corresponding matrices of the state-space model are

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{b_v}{J} & \frac{K_m}{J} \\ 0 & -\frac{K_g}{L} & -\frac{R}{L} \end{bmatrix}; \qquad b = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix}; \qquad c = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}; \qquad d = 0$$

which implies $\dot x_1 = \dot\theta = x_2$

Problem 1.6 (Simplify the System of the DC motor)

a) Follows directly from substituting $L = 0$ and $b_v = 0$.

b) The equations

$$T = K_m i \quad \text{and} \quad e_b = K_g\dot\theta$$

lead then to

$$K_g = K_m =: K.$$

Problem 1.7 (Identifying the real coefficients of the DC motor)

a) By repeating step input experiments using different final values for the input step and reading out the final angular velocity values, Table A.1 can be obtained.

Table A.1: Experimental data for identifying the motor constant K_m

Quantity	Values					
v (V)	1	2	3	4	5	6
$\dot{\theta} = \Omega$ (rad/s)	14	36.5	58	80	102	124

A plot of the angular velocity $\dot{\theta}$ over the voltage v is shown in Figure A.4. When using the linear basic fitting tool (Tools $\rightarrow$ Basic Fitting) (see Figure A.4), one obtains the linear approximation

$$y = 22x - 7.8 \qquad \text{for the equation} \qquad \dot{\theta} = \frac{1}{K_m} \cdot v - \frac{R}{K_m} \cdot i,$$

which means that

$$\frac{1}{K_m} = 22 \frac{\text{rad}}{\text{V s}} \quad \Rightarrow \quad K_m \approx 0.0455 \frac{\text{V s}}{\text{rad}}.$$
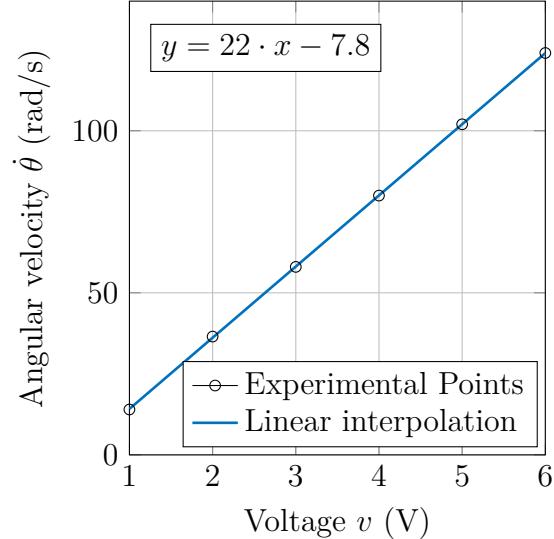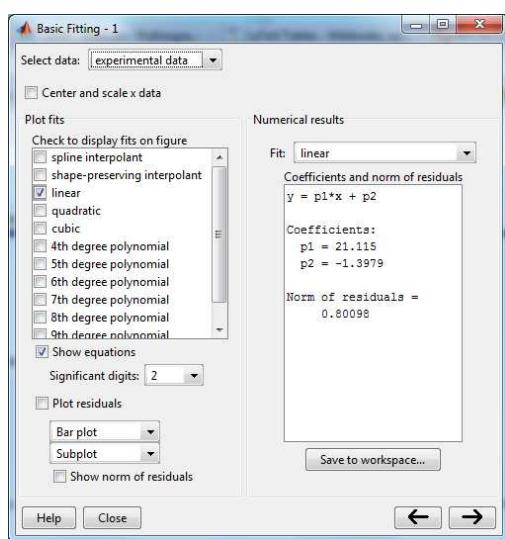
Figure A.4: left: Basic Fitting Tool, right: linear curve fitting on the behaviour between v and $\dot{\theta}$

Note that the constant term is negative due to the friction. If the applied voltage is too small to overcome the friction, the motor does not move. An idealised plot

of velocity over voltage is shown in Figure A.5. The offset that can be seen in this plot has been compensated in the SIMULINK motor block. Since the friction depends on various parameters like the motor temperature, this offset does however not always compensate the friction perfectly.
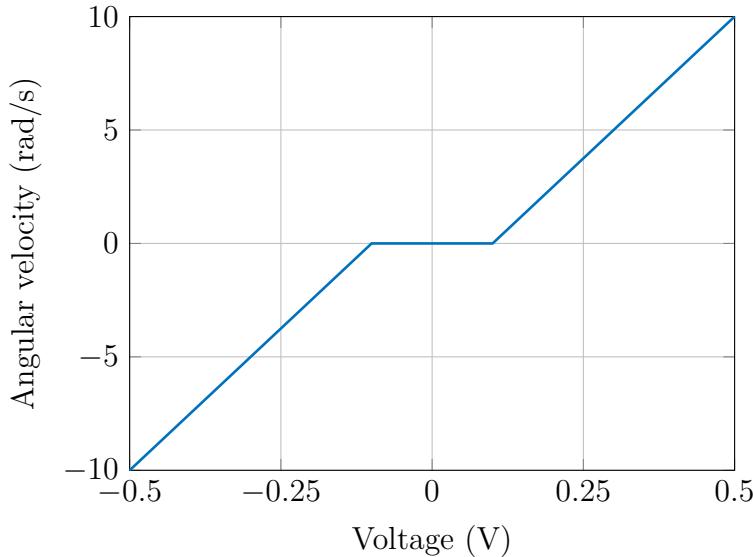
Figure A.5: Idealised plot of angular velocity over voltage

b) The unit of the static gain is $\frac{\text{rad}}{\text{V s}}$. To obtain the static gain experimentally, one can apply an input step signal to the plant and determine the final value of the step response. Because of friction one obtains better results when applying large steps. The response to a step from 0 to 6 V is shown in Figure A.6.

The final value is about 124 rad/s, therefore the static gain is

$$K_0 = \frac{124}{6} \frac{rad}{Vs} \approx 21 \frac{rad}{Vs}.$$

c) From (1.33) we have $K_0 = 1/K$ where $K = K_g = K_m$. One thus obtains

$$K_m = \frac{1}{K_0} \approx 0.0476 \frac{\text{V s}}{\text{rad}}.$$

This confirms the result obtained in (a).

d) Equation (1.35) can be rewritten as

$$v = R \cdot i + K_m \cdot \dot{\theta}.$$

For $\dot{\theta} = 0$ this describes a linear relationship between i and v with the unknown resistance R as the slope. By manually stopping the motor such that $\dot{\theta} = 0$, and
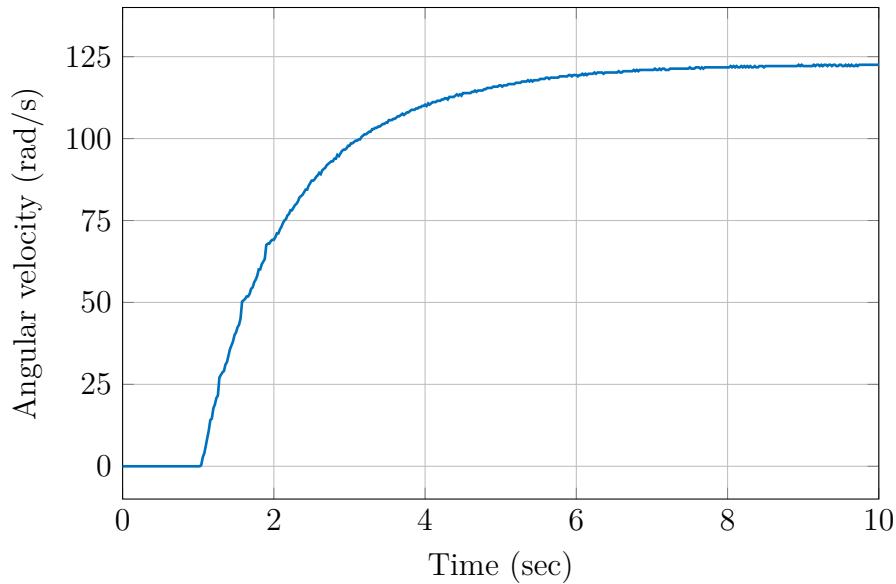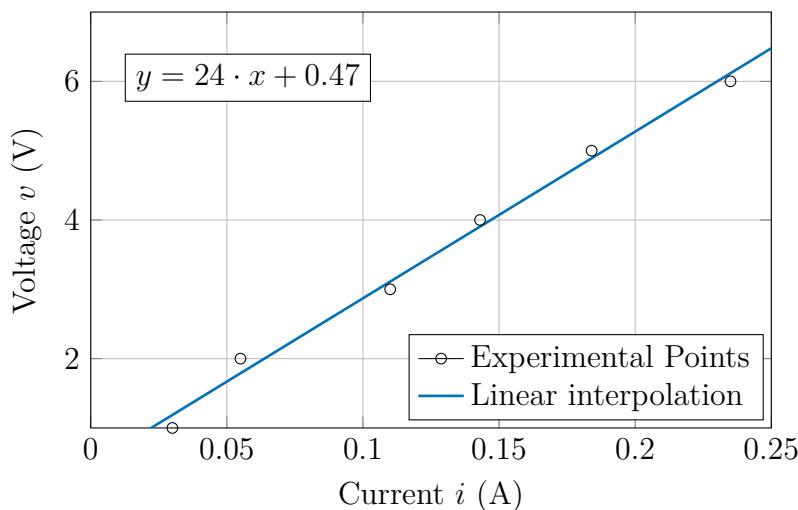
Figure A.6: Step response of the DC motor

Table A.2: Experimental data for identifying the unknown resistance R as the slope

Quantity	Values					
v (V)	1	2	3	4	5	6
i (A)	0.03	0.055	0.11	0.143	0.184	0.235

by changing the output from "Velocity" to "Electric Current" this linear relationship can be experimentally verified. Experimental data showing v over i can be seen in Figure A.7 together with a linear approximation. The slope and thus the resistance is given by $R \approx 24\Omega$. See Table A.2.

Figure A.7: Linear curve fitting for the plot of v over i

e)   From Figure 1.13 the time constant of a first order system can be determined as the time at which 63.2% of its steady state value k are reached in response to a step input. Measurements should give a value $\tau \approx 1.1$ s for the time constant. It is advisable to conduct several experiments and take the mean value of the time constants.

f)   From $\tau = JR/K^2$ and using the previously obtained parameter values

$$R \approx 24 \ \Omega \quad \text{and} \quad K = \frac{1}{21} \ \frac{\text{V}}{\text{rad}}$$

the inertia can be calculated as $J = K^2\tau/R = 1.0417 \cdot 10^{-4} \ \text{kg m}^2$. The resulting transfer function is

$$G(s) = \frac{21}{1.1 \cdot s + 1}.$$

SIMULINK files:

- `Model_Problem1_7_OpenLoopRampInput.slx`
- `Model_Problem1_7_OpenLoopStepInput.slx`

Problem 1.8 (Validation of the transfer function)

Figure A.8 shows a SIMULINK model that generates the response to input steps from 0 to 6 volts, assuming that the model parameters `K0` and `tau` are properly defined in the MATLAB workspace with the values determined in Problem 1.7. You can either use the 'To Workspace' block as a sink to for plotting in MATLAB or you can use the given scope to compare the results of the simulation and the experiment.
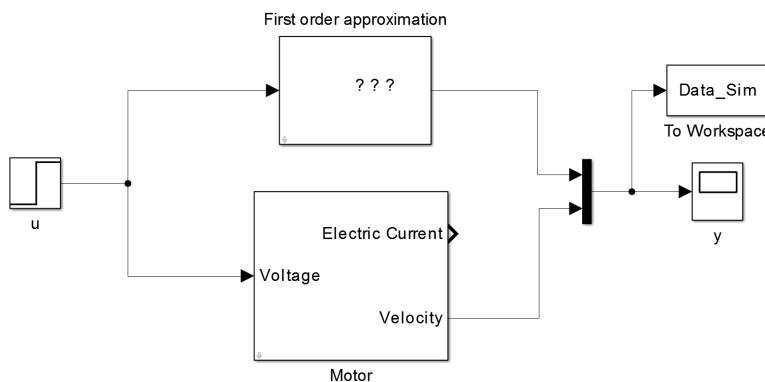
Figure A.8: Simulink model of the DC motor

Figure A.9 shows a plot of the step response of the simulation together with the step response of the real DC motor. In this case we conclude that our simplifications were reasonable.

SIMULINK file: `Model_Problem1_8_ParameterTest.slx`
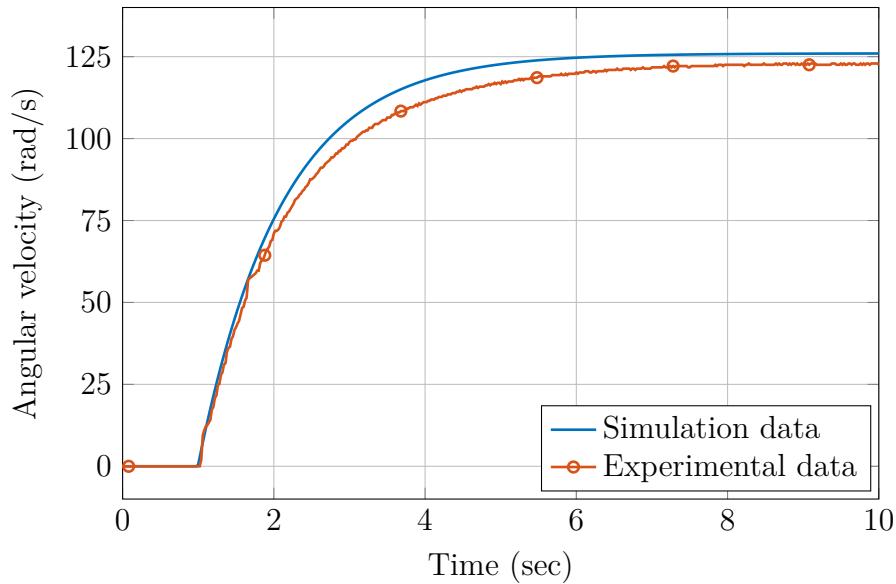
Figure A.9: Simulated and experimental step response of the DC motor

Problem 1.9 (State-space form)

a) Defining $x_1 = y$, $x_2 = \dot{y}$, from $\ddot{y}(t) = -3\dot{y}(t) - 2y(t) + 2u(t)$ follows

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \dot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 2 \end{bmatrix} u$$

$$y = x_1 = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + 0\, u$$

b) In MATLAB a ss model can be generated using the `ss` function and simulated using `step` or `lsim` as shown in the MATLAB script:

```
G2 = ss([0 1; -2 -3], [0; 2], [1 0], 0);
step(G2)
```
The step response is shown in A.10

MATLAB file: `Sol_Problem1_9_DifferentialEquationWithoutInitialConditions.mlx`
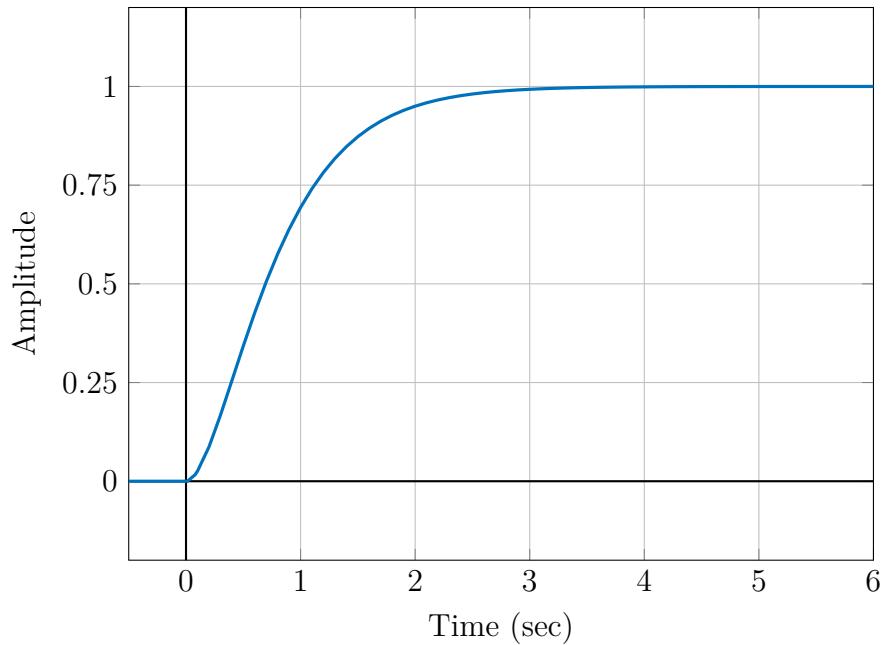
Figure A.10: Step response of second order system

Problem 1.10 (2nd order differential equation with non-zero initial values)

The solution of the differential equation with non-zero initial values can be obtained as the superposition of the free response $y_{free}(t)$ and the forced response $y_{forced}(t)$

$$y(t) = y_{free}(t) + y_{forced}(t).$$

The free response describes the system behaviour, resulting from the initial values. The forced response is the system response to external excitation. For the calculation of the solution $y(t)$ first the homogeneous solution will be derived and then, by variation of the constants, the particular solution.

The solution can be obtained using the formulae for Laplace transformation of derivatives

$$\mathcal{L}[\dot{y}(t)] = sY(s) - y(0)$$
$$\mathcal{L}[\ddot{y}(t)] = s^2Y(s) - sy(0) - \dot{y}(0) .$$

From

$$\ddot{y}(t) + 3\dot{y}(t) + 2y(t) = 0 \iff s^2Y(s) - 3s + 3(sY(s) - 3) + 2Y(s) = 0$$

we have

$$Y(s) = \frac{3s + 9}{(s + 1)(s + 2)} = \frac{A}{s + 2} + \frac{B}{s + 1}$$

with residues

$$A = \left.\frac{3s+9}{(s+1)}\right|_{s=-2} = -3\,; \quad B = \left.\frac{3s+9}{(s+2)}\right|_{s=-1} = 6\,.$$

Thus

$$Y(s) = -\frac{3}{s+2} + \frac{6}{s+1} \quad \Rightarrow \quad y_{free} = \left(-3e^{-2t} + 6e^{-t}\right)\sigma(t)$$

Which will give the same result as above

$$y(t) = y_{free}(t) + y_{forced}(t) = \left(2 + 2e^{-t} - e^{-2t}\right)\sigma(t)\,.$$

The responses are shown in Figure A.11.
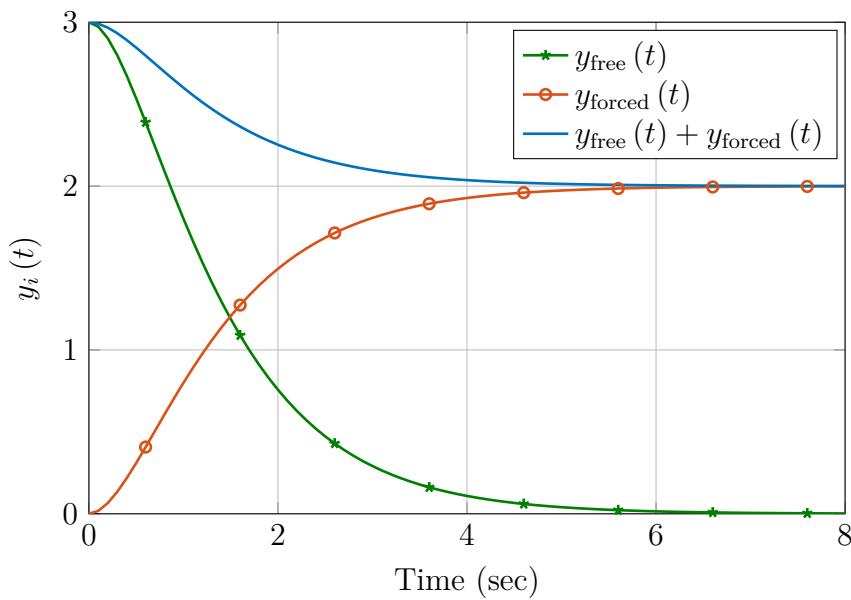
Figure A.11: Free and forced response

The relevant MATLAB commands and the signals are in:

"Sol_Problem1_10_DifferentialEquationWithInitialConditions.mlx".

Problem 1.11 (Steady state and transient response of 2nd order system)

a) The step response $y(t)$ can be divided into a stationary and a transient component

$$y(t) = y_{st}(t) + y_{tr}(t); \qquad y_{st}(t) = A\sigma(t); \qquad y_{tr}(t) = Be^{p_1 t} + Ce^{p_2 t}$$

From the partial fraction expansion

$$A = \frac{b_0}{p_1 p_2}; \qquad B = \frac{b_0}{p_1(p_1 - p_2)}; \qquad C = \frac{b_0}{p_2(p_2 - p_1)}$$

b) For $t = 0^+$ the exponential terms as well as the step function $\sigma(t)$ take the value 1; thus we obtain

$$y(0^+) = A + B + C = 0$$

c) From the initial value theorem follows

$$y(0^+) = \lim_{s \to \infty} s \cdot \frac{1}{s} \cdot \frac{b_0}{s^2 + a_1 s + a_0} = 0$$

d) The derivation in the time domain corresponds to a multiplication with s in the frequency domain. With the initial value theorem we obtain

$$\dot{y}(0^+) = \lim_{s \to \infty} s \cdot s \cdot \frac{1}{s} \cdot \frac{b_0}{s^2 + a_1 s + a_0} = 0$$

This shows that the step resposne of a second order system without any zeros contains a horizontal slope at $y(0^+)$.

e) Again, the derivation in the time domain corresponds to a multiplication with s in the frequency domain.

$$\dot{y}(0^+) = \lim_{s \to \infty} s \cdot s \cdot \frac{1}{s} \cdot \frac{b_1 s + b_0}{s^2 + a_1 s + a_0} = b_1$$

This shows that the steepness at $y(0^+)$ of the step resposne of a second order system depends on the location of the zero.

Problem 1.12 (RC model)

a) Using the complex impedance of the capacitor

$$Z_C(s) = \frac{U_C(s)}{I_C(s)} = \frac{1}{sC}$$

one can compute the transfer function

$$G(s) = \frac{V_o(s)}{V_i(s)} = \frac{Z_C(s)}{R + Z_C(s)} = \frac{\frac{1}{sC}}{R + \frac{1}{sC}} = \frac{1}{RCs + 1}$$

The step response is $h(t) = 1 - e^{-t/RC}$. For the period $0 \le t \le \epsilon$ the input signal is identical to a step with magnitude $1/\epsilon$, thus the output signal is

$$v_o(t) = \frac{1}{\epsilon}\left(1 - e^{-t/RC}\right) \quad \text{for} \quad 0 \le t \le \epsilon$$
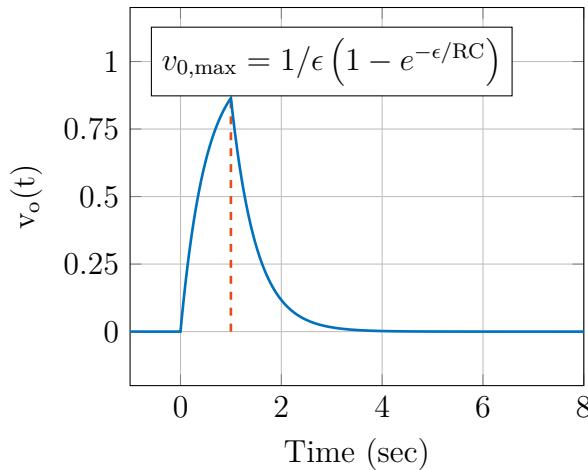
After time $t > \epsilon$ the input signal is again 0 and the output signal corresponds to the free response to the system with initial condition

$$v_o(t) = v_{o,max} e^{-(t-\epsilon)/RC} \quad \text{for} \quad t \ge \epsilon$$

where the initial values are computed as the final values of the first interval

$$v_{o,max} = \frac{1}{\epsilon}(1 - e^{-\epsilon/RC})$$

The output signal is shown in Figure A.12.

Figure A.12: Output voltage $v_{o,\epsilon}$

b)   By further reducing ϵ the rectangular pulse will become shorter and higher, but maintaining a constant area of 1. In the limit as $\epsilon \to 0$ it converges to a unit impulse and one might expect that the output voltage is the impulse response.

To show this, calculate the limit as $\epsilon \to 0$ of the initial values of the second interval

$$v_{o,max} = \lim_{\epsilon \to 0} \frac{1}{\epsilon}(1 - e^{-\frac{\epsilon}{RC}}) = \lim_{\epsilon \to 0} \frac{1}{\epsilon}(1 - 1 + \frac{\epsilon}{RC} + \cdots) = \frac{1}{RC}$$

Thus

$$v_o(t) = \frac{1}{RC}e^{-\frac{t}{RC}} = g(t)$$

which is exactly the impulse response. This is illustrated by the MATLAB file `"Sol_Problem1_12_RCModel.mlx"`. The results are shown in Figure A.13.
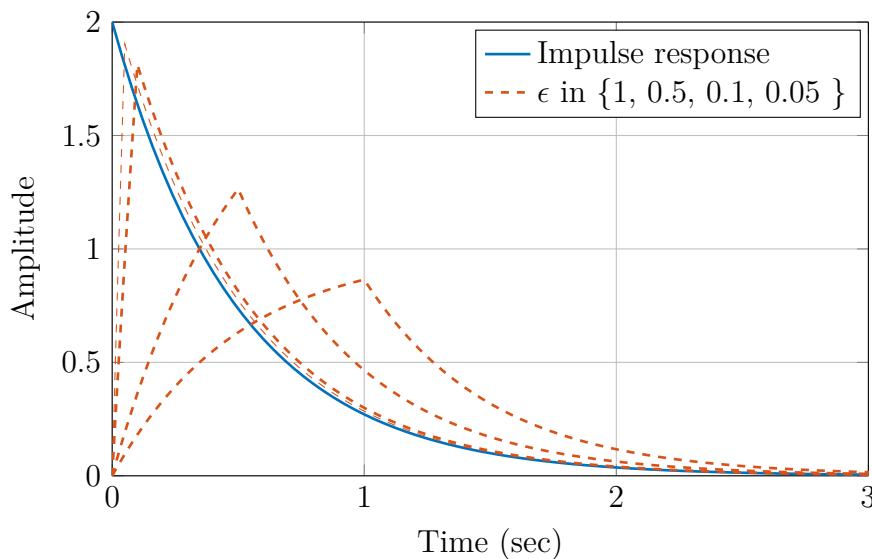
Figure A.13: Responses to rectangular pulses and impulse response

Problem 1.13 (Discrete convolution)

The MATLAB code `Problem1_13_DiscreteConvolution.mlx` produces the plots shown in Figures A.14 and A.15
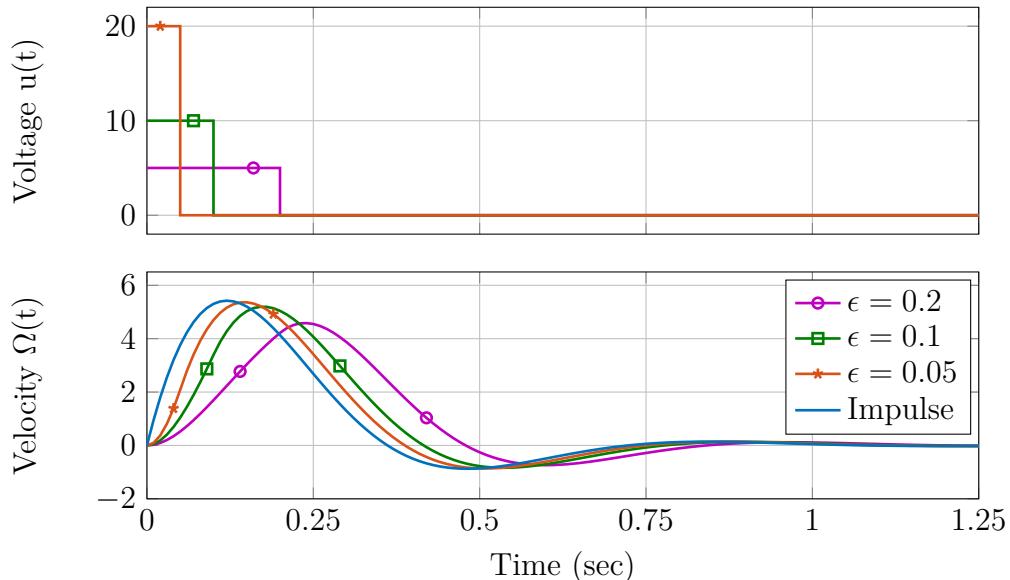
Figure A.14: Block pulse responses and impulse response for part (a)
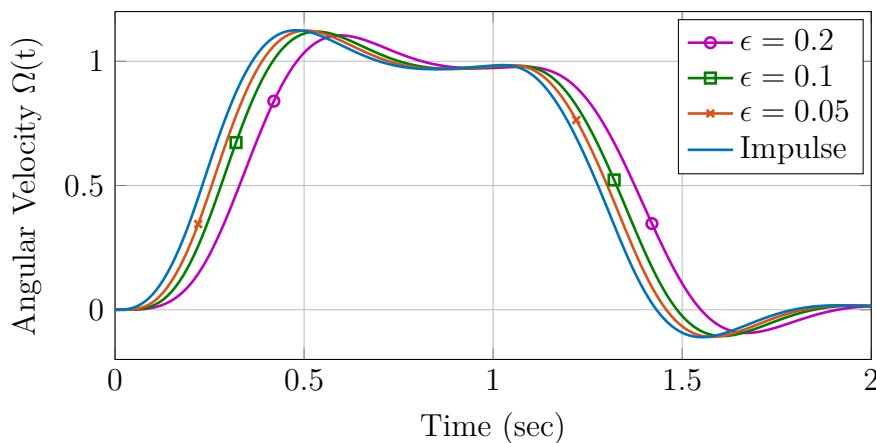
Figure A.15: Response to a trapezoidal input signal for part (b)

Problem 1.14 (Mass-spring-damper system)

a)   The transfer function of the mass-spring-damper system is

$$G_s(s) = \frac{1}{m\,s^2 + b\,s + k} = \frac{\frac{1}{m}}{s^2 + \frac{b}{m}s + \frac{k}{m}}$$

where m is the mass, b is the damping coefficient and k is the spring constant. From the standard form of a second order system one can easily find

$$
\begin{aligned}
\text{natural frequency:} \quad & \omega_n = \sqrt{\tfrac{k}{m}} \\
\text{static gain:} \quad & K = \tfrac{1}{k} \\
\text{damping:} \quad & \zeta = \frac{\frac{b}{m}}{2\omega_n} = \frac{b}{2\sqrt{mk}}
\end{aligned}
$$

b) The MATLAB file `"Sol_Problem1_14_MassSpringDamperSystem.mlx"` can be used to simulate the step response, see Figure A.16.
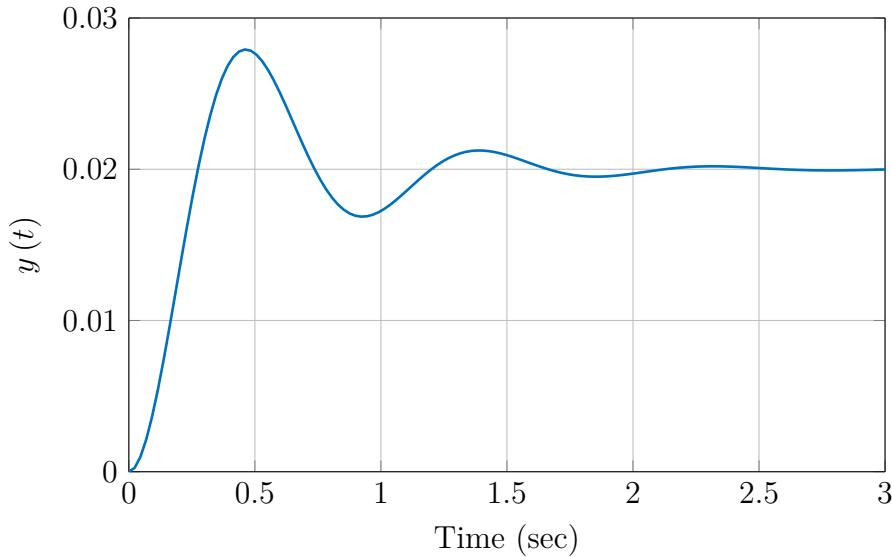
Figure A.16: Responses to rectangular pulses and impulse response

c) From the standard form of a second order system and from the transfer function of an RLC system derived from (1.2), it is straightforward to compute

$$
\omega_n = \sqrt{\frac{1}{LC}}; \qquad K = 1; \qquad \zeta = \frac{R/L}{2\omega_n} = \frac{R}{2}\sqrt{\frac{C}{L}}
$$

Problem 1.15 (Peak overshoot)

a) Rewriting the Laplace transform $H(s)$ of the step response in the shown way and using the Laplace transformation Table 1.1 one derives

$$
\begin{aligned}
h(t) = \mathcal{L}^{-1}\left[H(s)\right] &= \left(1 - e^{-\zeta\omega_n t}\cos\omega_d t - \frac{\zeta\omega_n}{\omega_d}e^{-\zeta\omega_n t}\sin\omega_d t\right)\sigma(t) \\
&= \left(1 - e^{-\zeta\omega_n t}\left(\cos\omega_d t + \frac{\zeta}{\sqrt{1-\zeta^2}}\sin\omega_d t\right)\right)\sigma(t)
\end{aligned}
$$

b*) For the components can be derived

$$\frac{d}{dt}\left(e^{-\zeta\omega_n t}\cos\omega_d t\right) = -\zeta\omega_n e^{-\zeta\omega_n t}\cos\omega_d t - \omega_d e^{-\zeta\omega_n t}\sin\omega_d t$$

$$\frac{d}{dt}\left(\frac{\zeta\omega_n}{\omega_d}e^{-\zeta\omega_n t}\sin\omega_d t\right) = \frac{\zeta\omega_n}{\omega_d}\left[-\zeta\omega_n e^{-\zeta\omega_n t}\sin\omega_d t + \omega_d e^{-\zeta\omega_n t}\cos\omega_d t\right]$$

$$= -\frac{(\zeta\omega_n)^2}{\omega_d}e^{-\zeta\omega_n t}\sin\omega_d t + \zeta\omega_n e^{-\zeta\omega_n t}\cos\omega_d t$$

leading to

$$\frac{d}{dt}h(t) = \omega_d e^{-\zeta\omega_n t}\sin\omega_d t + \frac{(\zeta\omega_n)^2}{\omega_d}e^{-\zeta\omega_n t}\sin\omega_d t$$

$$= \left(\omega_d + \frac{(\zeta\omega_n)^2}{\omega_d}\right)e^{-\zeta\omega_n t}\sin\omega_d t.$$

From the $\dot{h} = 0$ follows that $\sin\omega_d t = 0$, i.e.

$$\omega_d t_p = n\pi, \quad n = 0,1,2,\ldots$$

$$t_p = \frac{n\pi}{\omega_d}.$$

Substituting the overshoot time $t_p = \frac{\pi}{\omega_d}$ in the transfer function

$$M_p = \frac{\hat{h} - h(\infty)}{h(\infty)} = 1 - e^{-\zeta\omega_n\frac{\pi}{\omega_d}}\left(\cos\omega_d\frac{\pi}{\omega_d} + \frac{\zeta\omega_n}{\omega_d}\sin\omega_d\frac{\pi}{\omega_d}\right) - 1$$

$$= e^{\frac{-\zeta\omega_n\pi}{\omega_d}}.$$

Problem 1.16 (Identification of mass-spring-damper system)

The transfer function of the mass-spring-damper system was derived in Problem 1.14. Using the standard form of a second order system (1.25) we can write

$$G(s) = \frac{\frac{1}{m}}{s^2 + \frac{b}{m}s + \frac{k}{m}} = K\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

The parameters are computed from

$$\begin{array}{llll}
\frac{1}{m} &= K\omega_n^2 & m &= \frac{1}{K\omega_n^2} \\
\frac{b}{m} &= 2\zeta\omega_n & \Rightarrow \quad b &= \frac{2\zeta}{K\omega_n} \\
\frac{k}{m} &= \omega_n^2 & k &= \frac{1}{K}
\end{array}$$

The steady state value of the given step response yields the static gain

$$K = \frac{y(\infty)}{u(\infty)} = \frac{2.5}{5} = 0.5$$

From the plot the peak time is $t_p = 7.5$ and the peak overshoot is

$$M_p = \frac{3.25 - 2.5}{2.5} = 0.3$$

From (1.27) it follows that

$$\sqrt{1 - \zeta^2} \ln M_p = -\zeta \pi$$

$$\zeta = \sqrt{\frac{(\ln M_p)^2}{\pi^2 + (\ln M_p)^2}}$$

Substituting the peak overshoot value, the damping ratio can be calculated as $\zeta = 0.3579$. Moreover, from

$$t_p = \frac{\pi}{\omega_d} = \frac{\pi}{\omega_n \sqrt{1 - \zeta^2}}$$

we obtain

$$\omega_n = \frac{\pi}{t_p \sqrt{1 - \zeta^2}}$$

(Alternatively one can use the approximation for $t_r \approx \frac{1.7}{\omega_n}$ or $t_s \approx \frac{4.6}{\zeta \omega_n}$ to compute ω_n; this leads to slightly different values.)

Using the above numerical values yields $\omega_n = 0.4486$, which leads to

$$m \approx 10 ; \qquad b \approx 3 ; \qquad k \approx 2$$

MATLAB solution: `Sol_Problem1_16_MassSpringDamperSystemIdentification.mlx`

A.2 Solutions for Chapter 2

Problem 2.1 (Speed control of a DC motor with proportional feedback)

a) We have

$$K_{pos} = \lim_{s \to 0} L(s) = \lim_{s \to 0} \frac{K_\mathrm{P}K}{JRs + K^2} = \frac{K_\mathrm{P}}{K}$$

and thus

$$e_\infty = \frac{1}{1 + K_{pos}} = \frac{K}{K + K_\mathrm{P}}.$$

A steady state accuracy $e_\infty \leq 0.1$ then requires $K_\mathrm{P} \geq 9K$, where $K \approx 0.0476$ from Problem 1.7.a. The smallest value of K_P that achieves this accuracy is therefore

$$K_\mathrm{P} = 9K \approx 0.43.$$

b) The same calculation as in (a) yields as requirement on the gain $K_\mathrm{P} \approx 0.904$.

Problem 2.2 (Speed control of a DC motor with proportional feedback)

a) In Figure A.17 the SIMULINK model of the model is shown.
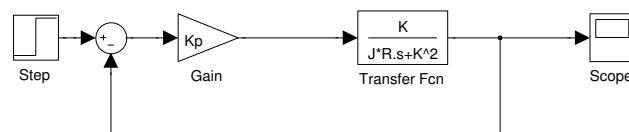
Figure A.17: Feedback loop with proportional controller

Figure A.18 shows the simulated step response for $K_\mathrm{P} = 0.5$. The static gain is about $43.6\,rad/s$, which implies that the steady state error is less than 10%.
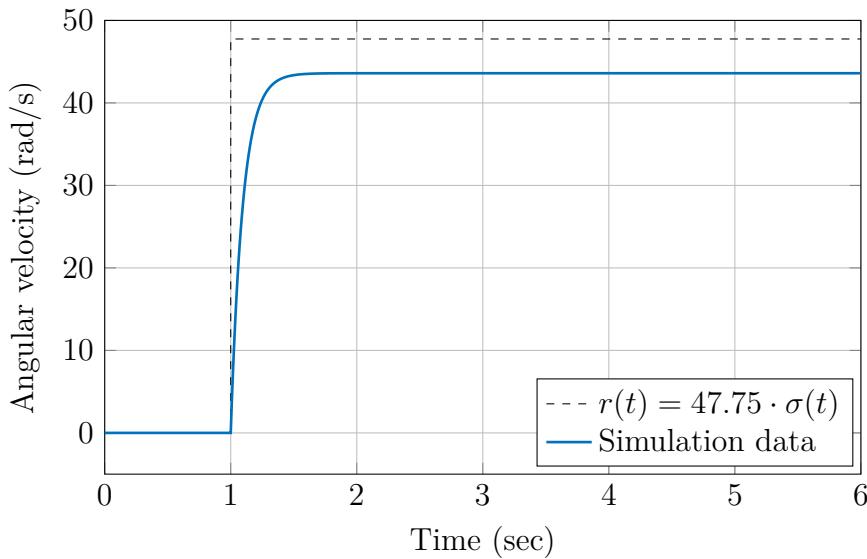
Figure A.18: Simulated step response with P controller and $K_{\mathrm{P}} = 0.5$

b) For $K_{\mathrm{P}} = 0.5$ one obtains the step response shown in Figure A.19.
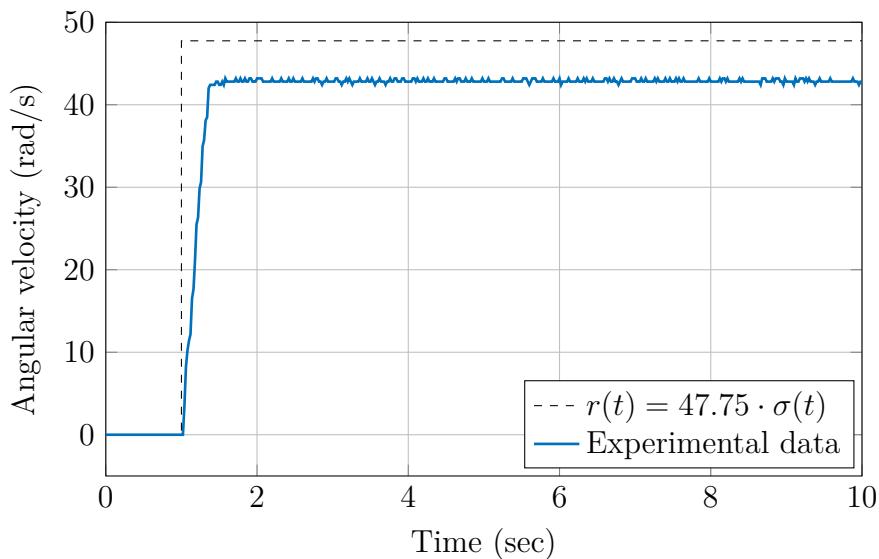
Figure A.19: Experimental step response with P controller and $K_{\mathrm{P}} = 0.5$

A comparison of the simulated step response in Figure A.18 with the experimental results in Figure A.19 shows that the static gain and consequently the steady state error are almost equal. In simulation the system achieves a final value of $43.6\frac{\mathrm{rad}}{\mathrm{s}}$ as expected, whereas the experiments tend to a value of $43.2\frac{\mathrm{rad}}{\mathrm{s}}$. One significant difference of the two responses is the noise in the measured response of the real motor.

c)   The experimental results shown in Figure A.20 confirms the expected steady state error. The step response shows a slight overshoot and oscillatory behaviour. This contradicts the assumption that the behaviour of the motor is represented by a first order system: with a P controller the closed-loop system is also first order, and a first order system does not display overshoot. The observed behaviour is in fact caused by unmodelled effects such as a time delay and hardware low-pass filters, which are implemented in the rotational speed sensor. The effect such filters can have on the closed-loop performance will be discussed in Chapter 4.
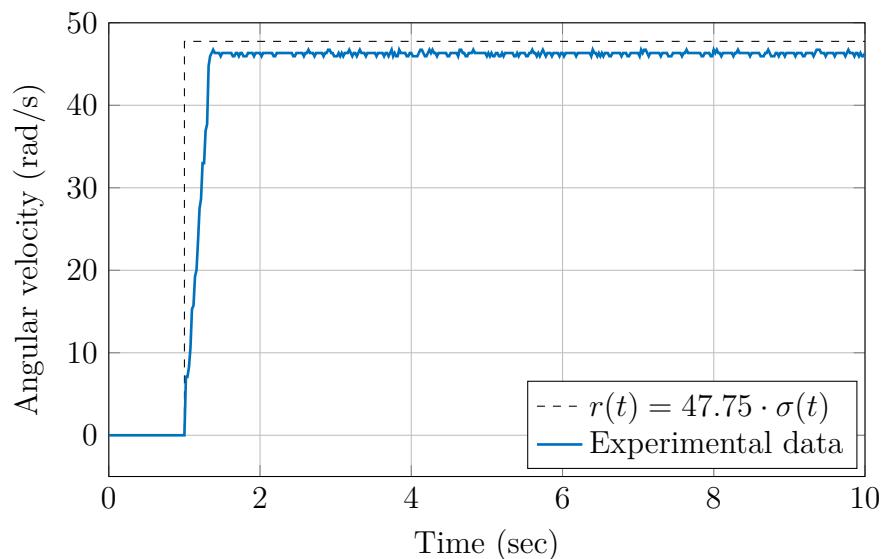
Figure A.20: Speed control, experimental step response with P controller and $K_\mathrm{P} = 1.7$

SIMULINK file: `Model_Problem2_2_PControllerVelocity.slx`

Problem 2.3 (Angle control of a DC motor with proportional feedback)

a) In Problem 1.5 the same task has already been solved for the second order motor
    model: since θ is the integral of $\dot{\theta}$, it is necessary to integrate the output of $G(s)$.
    Thus the transfer function from u to θ is

    $$G(s) = \frac{21}{s(1.1s + 1)}.$$

b) As a starting point for the Ziegler-Nichols Method 2, the critical gain K_{cr} is needed.
    To find this gain, closed loop proportional control is used. For small proportional
    gains K_{P}, the closed loop step response converges. If the gain is increased, the
    closed loop step response starts to oscillate. The gain K_{P} for which the system
    starts oscillating is called K_{cr}. The gain K_{cr} for this problem is usually around 3.8,
    varying slightly with the used motor and operating conditions. Figure A.21 shows
    the experimental closed loop step response for $K_{\mathrm{P}} = K_{\mathrm{cr}} = 3.8$.
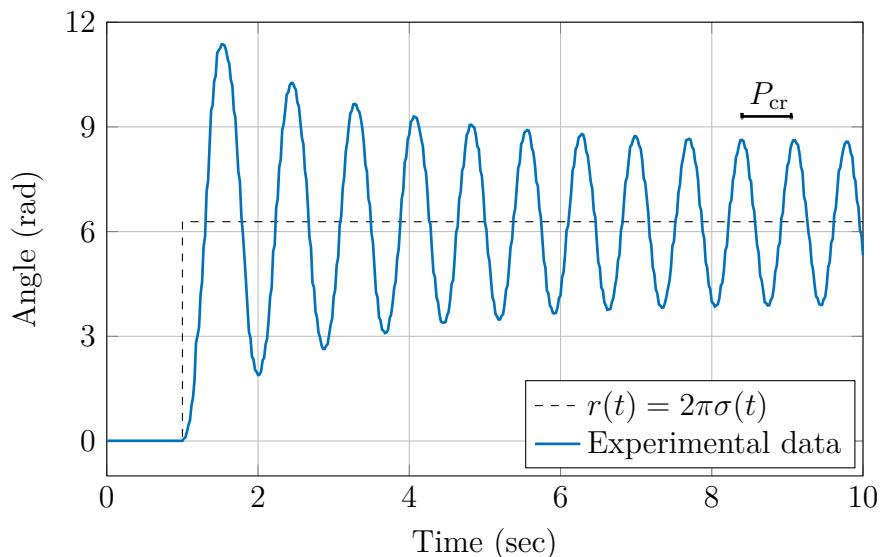
Figure A.21: Oscillating step response for $K_{\mathrm{P}} = K_{\mathrm{cr}} = 3.8$

According to the Ziegler-Nichols tuning rules, this leads to $K_{\mathrm{P}} = \frac{K_{\mathrm{cr}}}{2} = 1.9$. This
gain is a good starting point for further tuning.
*Note: From Figure A.21 one obtains the critical period $P_{\mathrm{cr}} \approx 0.74$ s. This can be
used to design a PI or PID controller.*

c) Figure A.22 shows the step response with a P controller and $K_{\mathrm{P}} = 1.9$. The steady
    state error is almost zero. The reason for this is the integral action in the plant
    (the shaft angle is the integral of the motor speed), resulting in a type 1 system for
    which the steady state error in the step response is zero.

On the other hand, the peak overshoot of approximately 85% and the oscillatory behaviour is undesirable.
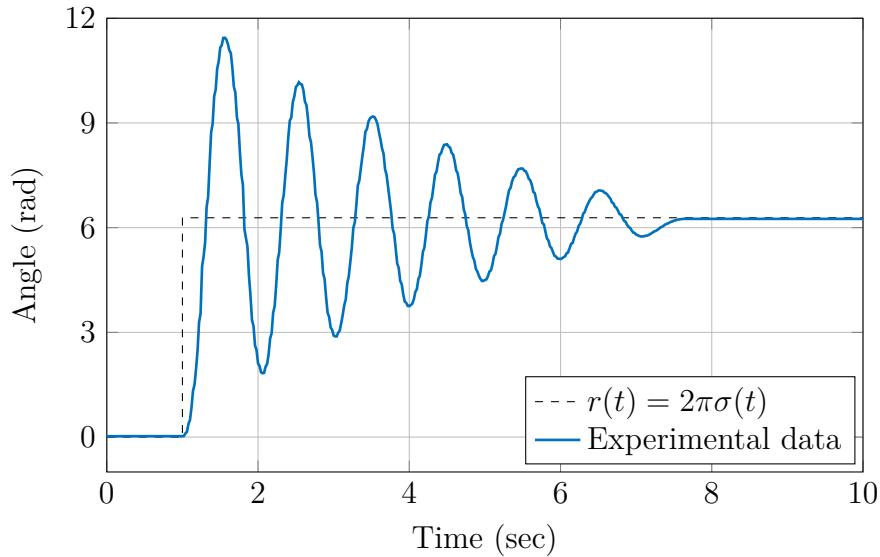
Figure A.22: Angle control, experimental step response with P controller and gain from Ziegler-Nichols tuning

d) The fastest possible response without any overshoot is achieved, if the closed loop system is critical damped ($\zeta = 1$), i.e. if the two closed loop poles are real and identical (see discussion in Section 1.3).

The closed loop transfer function is

$$G_{cl}(s) = \frac{21 K_P}{1.1 s^2 + s + 21 K_P},$$

with poles

$$0 = s^2 + \frac{1}{1.1} s + \frac{21 K_P}{1.1}$$

$$s_{1,2} = -\frac{1}{2 \cdot 1.1} \pm \sqrt{\left(\frac{1}{2 \cdot 1.1}\right)^2 - \frac{21 K_P}{1.1}}.$$

The damping ratio is therefore 1, if $\left(\frac{1}{2 \cdot 1.1}\right)^2 - \frac{21 K_P}{1.1} = 0$, leading to $K_P = 0.011$.

The simulated step response with this value of K_P is shown in Figure A.23. There is no steady state error due to integral action and no overshoot because K_P has been chosen to achieve the critical damping ratio. But in trade-off for the zero overshoot, the rise time is very long.
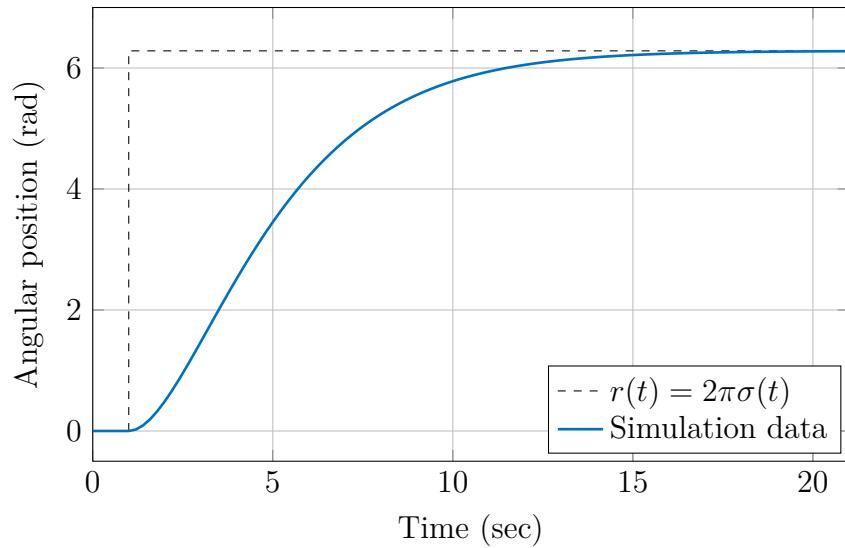
Figure A.23: Angle control, simulated step response with P controller that achieves critical damping

e) The experimental step response of the motor with the controller from (d) is shown in Figure A.24. It does not behave as in the simulation. Although there is no overshoot, the steady state angle is only a fraction of its desired value. This is clearly due to friction effects: the motor moves very slowly because of the low power in the input signal, because K_P is that small, and is not able to overcome friction. This shows that experimentally a P controller is due to friction not able to achieve the desired steady state value without overshoot.
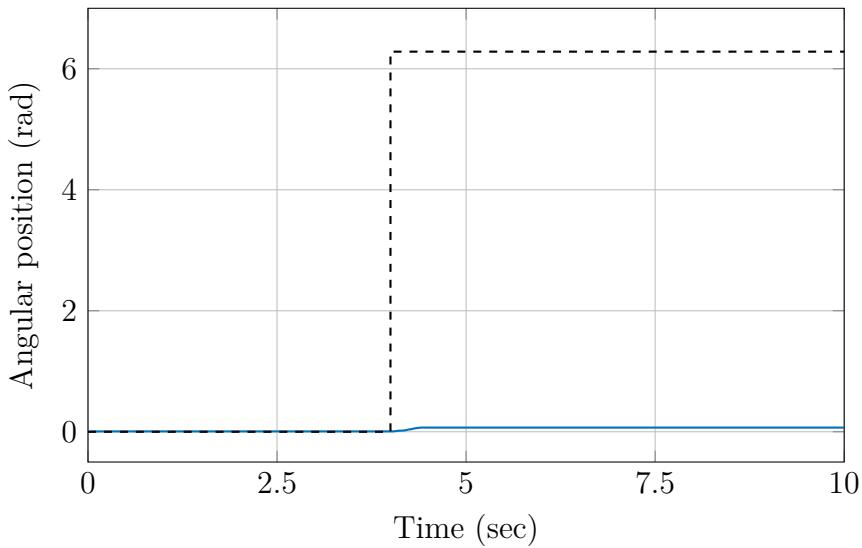
Figure A.24: Angle Control, experimental step response with P controller that achieves critical damping

f) In task (c) and (e) two extreme cases have been seen. In (c) a high value of K_P is used, resulting in very short rise time , but also high overshoot. In (e) K_P has been chosen that small, that no overshoot occurs but on the other hand the rise time is very long. In addition in experiment the small K_P leads to such a low input power that friction cannot be overcome, such that steady state accuracy can not be guaranteed. As a trade-off a rise time of 0.5 s is desired now.

With the closed-loop transfer function

$$G_{cl}(s) = \frac{K_P K}{JRs^2 + K^2 s + K_P K} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

we have

$$\omega_n = \sqrt{\frac{K_P K}{JR}} \approx \frac{1.7}{t_r}$$

and solving for K_P gives

$$K_P = \frac{JR}{K}\left(\frac{1.7}{t_r}\right)^2.$$

The desired rise time of $t_r \leq 0.5$ s leads to $K_P \geq 0.6072$. We choose $K_P = 0.61$. Figure A.25 indicates a good fit between simulation and real-time experiment, as well as a confirmation of the expected rise time, but the overshoot is still too high. In summary, we have seen that a P controller is always a trade-off between shorter rise time or smaller overshoot. To improve both characteristics at once other control strategies are needed.
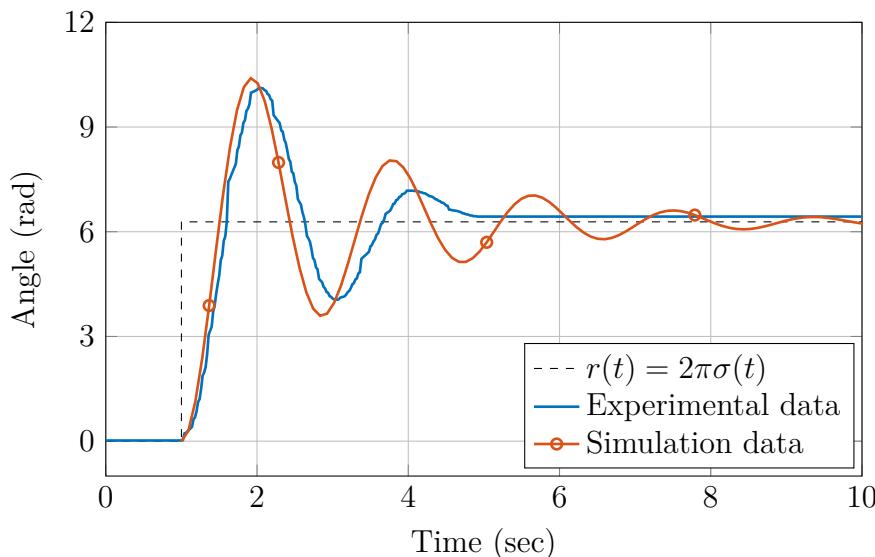
Figure A.25: Simulated (red) and experimental (blue) step response with P controller and $K_P = 0.61$.

Simulink file: `Model_Problem2_3_PControllerAngle.slx`

Problem 2.4 (Angle control of the DC motor with PD controller)

a)   Figure A.26 shows simulated step responses with a PD controller and $K_P = 0.44$ and $T_D \in [0.3, \ldots, 0.8]$. One can see that increasing T_D reduces the overshoot as well as the rise time.
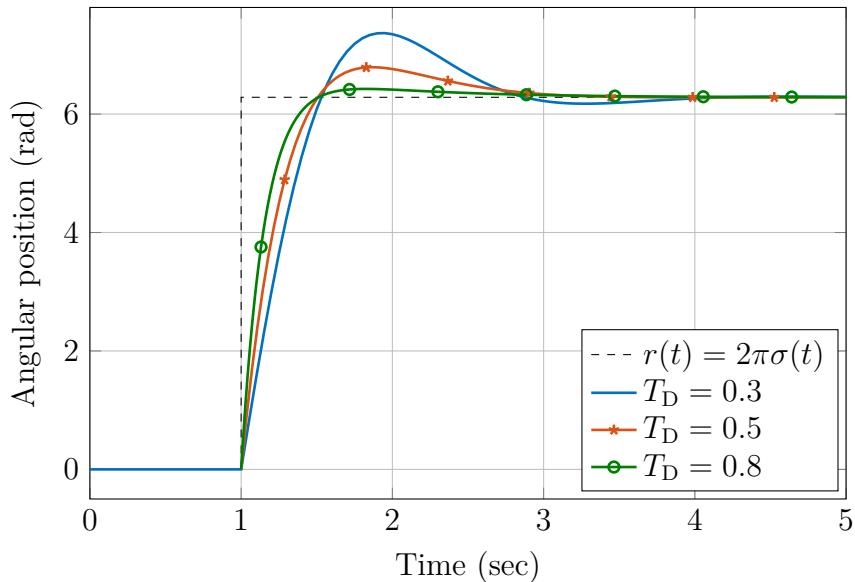
Figure A.26: Angle control, simulated step responses with PD control, $K_P = 0.44$ and $T_D \in [0.3, \ldots, 0.8]$

b)   The simulated and experimental responses in Figure A.27 show that higher values for T_D result in better damped transients, but in contrast to the simulation results in (a), due to saturation the rise time is increased.

Figure A.28 shows an experimental step response (subject to saturation), together with simulated step responses with and without saturation. With the selected parameter values for K_P and T_D and with the model from Problem 2.3.a, the closed-loop poles are real. The overshoot in the simulated response without saturation is due to the presence of a closed-loop zero in the left half plane. This overshoot is reduced when saturation is included in the simulation. That the experimental response also displays a slight overshoot is due to unmodelled effects, including a time delay of about 30 ms in the physical set up.

Simulink files:

•   `Model_Problem2_4_PDControllerAngleut.slx`
•   `Model_Problem2_4_PDControllerAngleWithSaturation.slx`
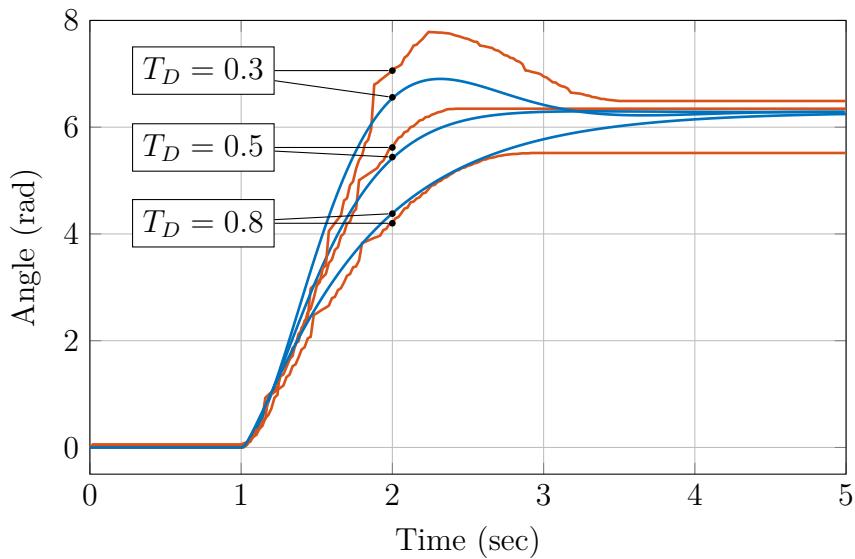
Figure A.27: Angle control, simulated step responses (smooth, blue) with saturation, together with the corresponding experimental responses (serrated, red). The PD controller parameters are $K_P = 0.44$, $T_D \in [0.34, 0.5, 0.8]$.
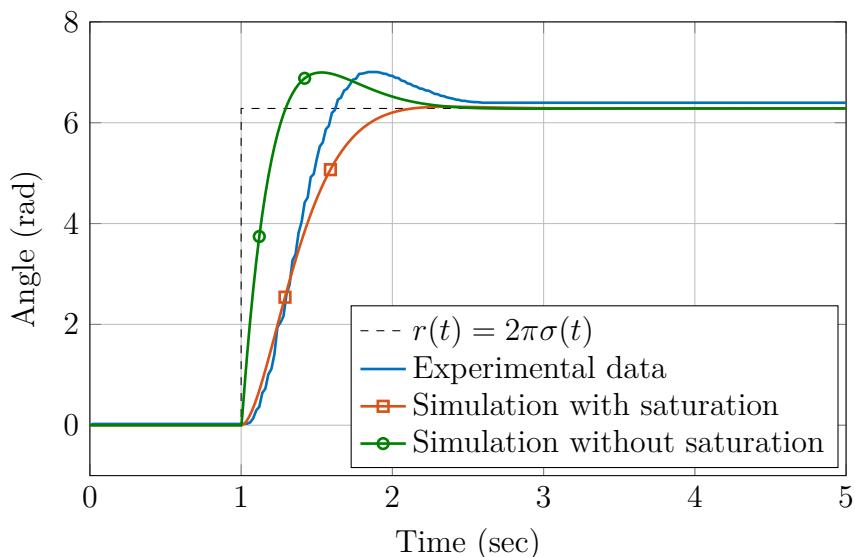
Figure A.28: Angle control, simulated step response with PD controller, together with the experimental response. Controller parameters are $K_P = 1$, $T_D = 0.34$.

Problem 2.5 (Shaft angle control of a DC motor with PID controller)

a)   With these values one obtains $K_P = 2.28$, $T_I = 0.37$ and $T_D \approx 0.1$.

b)   After fine-tuning the controller in simulation, one possible parameter set is

$$K_\mathrm{P} = 1.2, \quad T_\mathrm{I} = 0.44, \quad T_\mathrm{D} = 0.38.$$

With these values a rise-time of about 0.11 s and a peak overshoot of just less than 20% is achieved.

c)   i)   In the experiment, the achieved rise time is $t_r = 0.27$ s and there is a peak overshoot $M_p = 0.4$, which violates the specifications.

ii) Experimental and simulated responses are shown in Figure A.29.
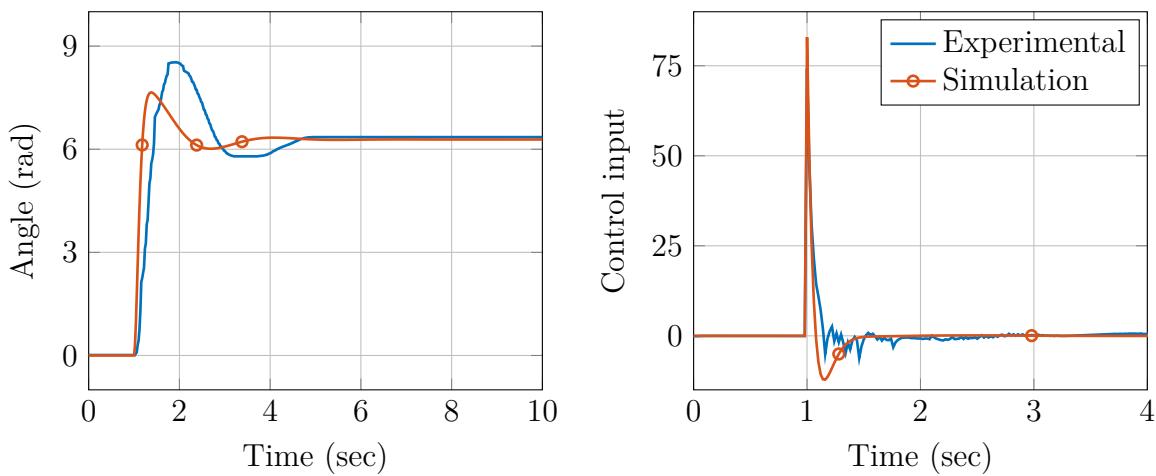
Figure A.29: Left: shaft angle control with PID controller, right: control input; shown are experimental (——) and simulated (——) responses

iii) Trying to reduce the overshoot to 20% by re-tuning the controller is not possible: one attempt might be to reduce the gain K_P and to increase the derivative time T_D, but this does not give the desired closed-loop characteristics.

The reason for this is that the control signal at the step time $t = 1$ s is well above the limit of the control input which can be applied to the plant. Actuator saturation occurs and leads to a windup-effect.

d)   A possible SIMULINK model is shown in Figure A.30. The simulated and experimental step response are presented in Figure A.31. Clearly now the responses match well, thus confirming a windup-effect as main reason for the mismatch observed before.
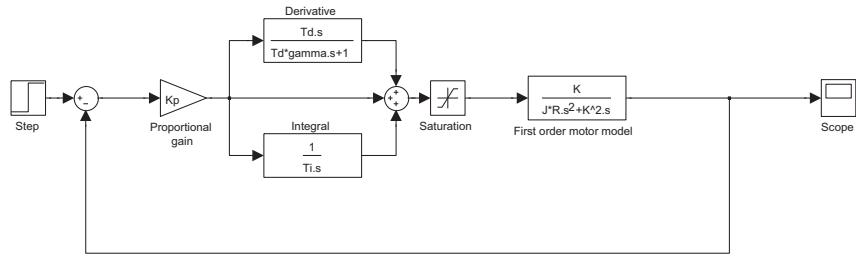
Figure A.30: SIMULINK model of PID controlled motor with saturation
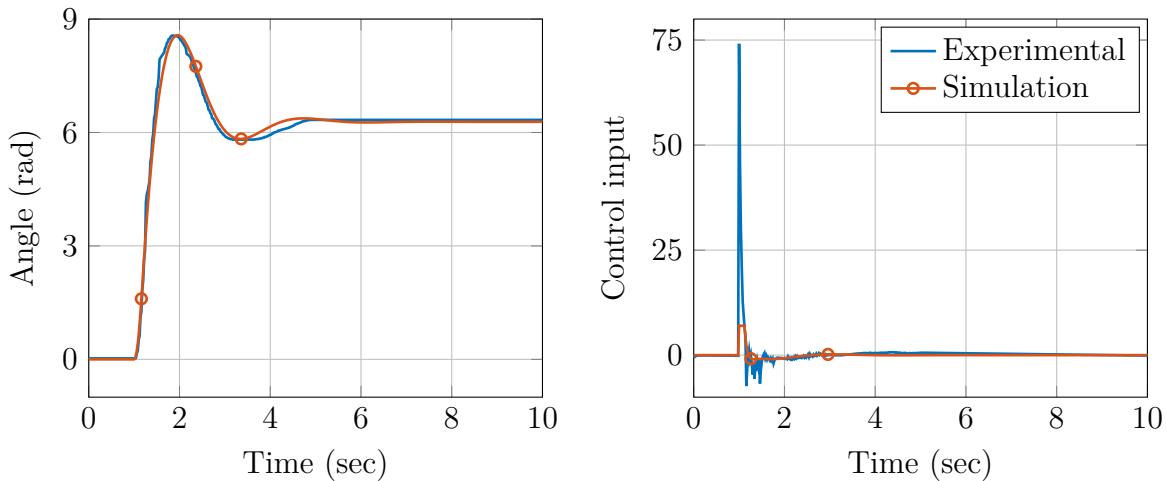
Figure A.31: Left: Angle control with PID controller and saturation, right: control input; shown are experimental (—) and simulated (—) response, the simulation includes actuator saturation

SIMULINK files:

- `Model_Problem2_5_PIDControllerAngle.slx`
- `Model_Problem2_5_PIDControllerAngleWithSaturation.slx`

Problem 2.6 (Integrator windup effect)

a)   Figure A.32 shows the general form of the SIMULINK model.

b)   The effect that saturation has on the step response is illustrated in Figure A.33. After a step change from 0 to 1 of the reference input, the controller generates a control signal $u_c(t)$ that exceeds the actuator range, which is limited by the saturation block to $-1 < u < 1$. The integrator is being charged up. It takes a significant time for the negativ error to wind down the integrator and to bring the control output back to its physical limits. The result is a huge overshoot.
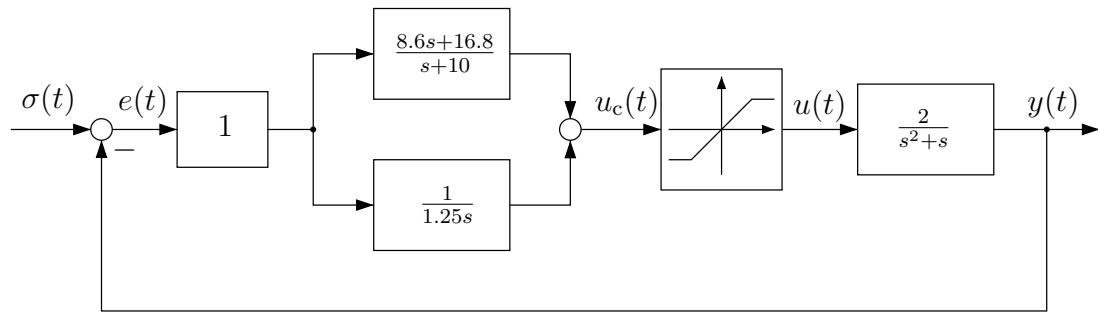
Figure A.32: PID control loop with actuator saturation
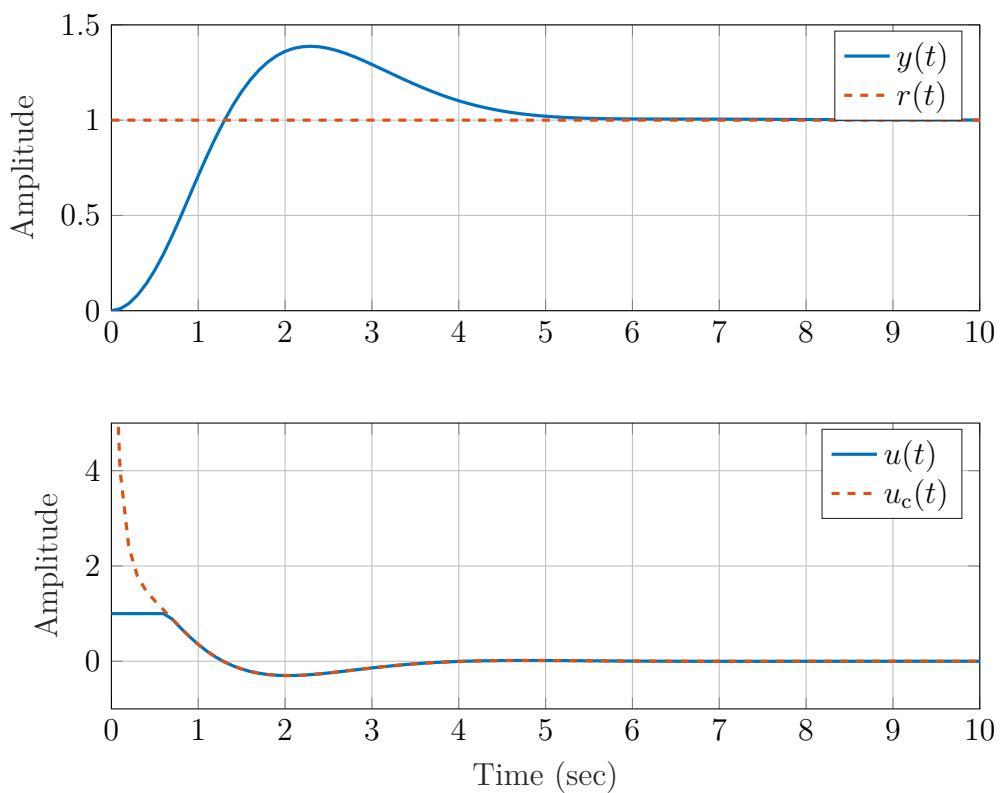
Figure A.33: Step response with windup effect

c) For a value of $K_{aw} = 1.6$ Figure A.34 illustrates the change of the step respone. The huge overshoot is damped and the integrator can wind down faster.
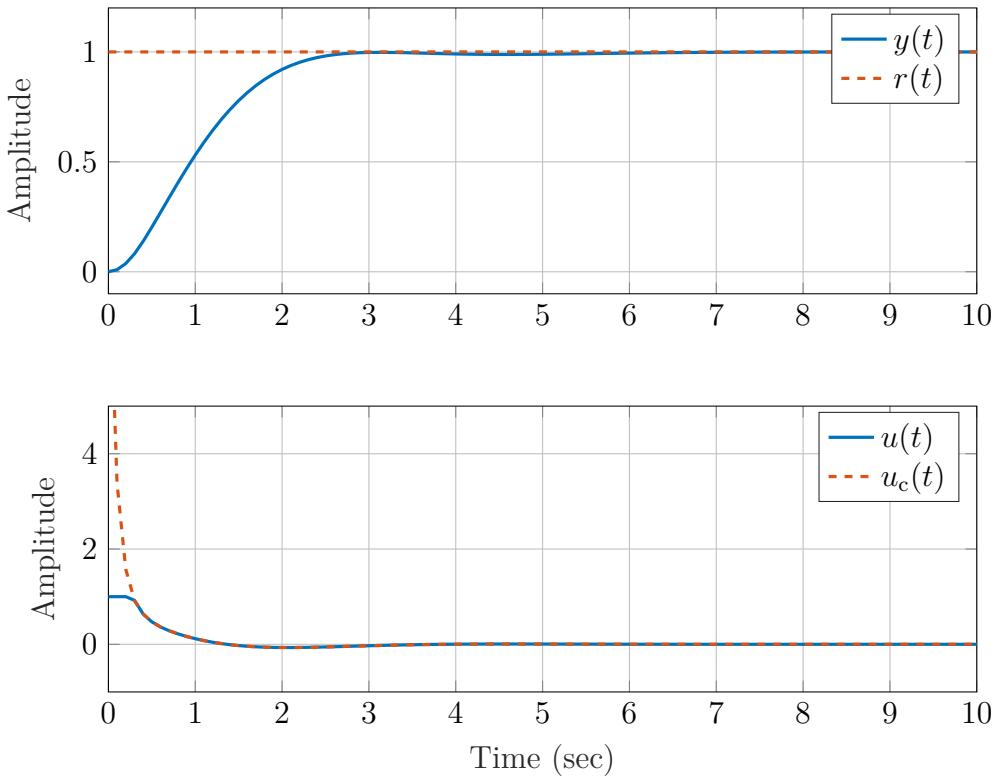
Figure A.34: Step response with anti-windup configuration

SIMULINK file: `Sol_Model_Problem2_6_IntegratorWindup.slx`

Problem 2.7 (PID Controller with anti-windup)

Using the SIMULINK model `"Model_Problem2_7_PIDAntiWindup.slx"` shown in Figure A.35 the values of K_P, T_I, T_D and K_{aw} can be tuned.

Figure A.36 shows the experimental step response with $K_P = 2.8$, $T_I = 1.0$ and $T_D = 0.18$ (top), together with the control input (bottom). One can see that saturation occurs at $t \approx 3$s.

Now the peak overshoot is $M_p = 0.06$, which is significantly less than the overshoot $M_p = 0.4$ obtained in Task 2.5 c). The achieved rise time of $t_r = 0.3$ s, however, is not much different from the one obtained before.

SIMULINK file: `Model_Problem2_7_PIDAntiWindup.slx`
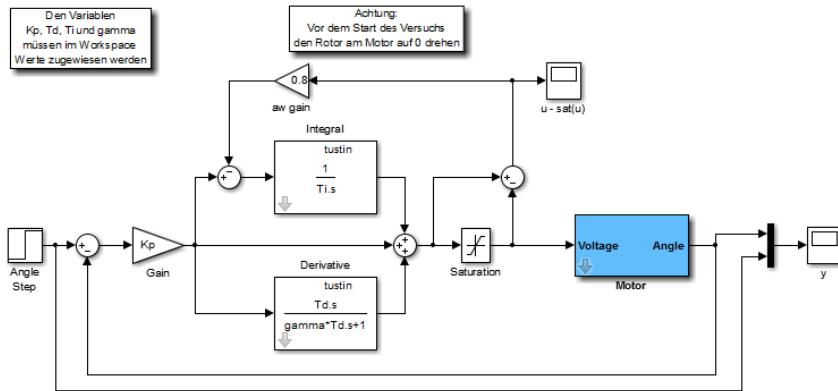
Figure A.35: Simulink model for experimental angle control with PID controller and anti windup function
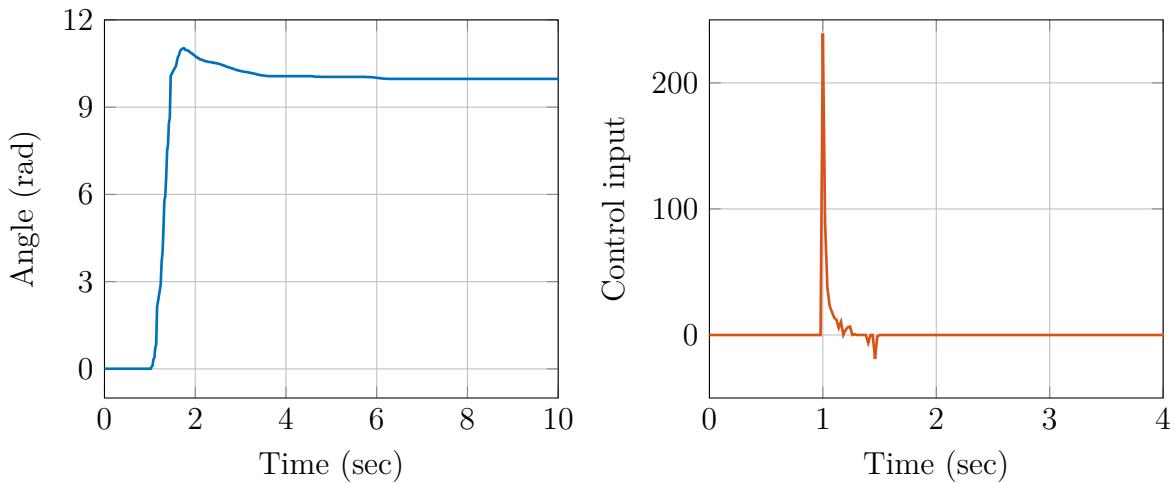
Figure A.36: Experimental step response with a PID Controller and anti-windup device (top), and control input (bottom)

Problem 2.8 (Internal model principle)

a)   Position error is considered and the feedback system is type 1 $\Rightarrow$ $e_\infty = 0$.

b)   The transfer function from the output disturbance $d_y(t)$ to the control error $e(t)$ is

$$G_{ed_y}(s) = \frac{-1}{1 + G(s)C(s)} = -G_{er} \quad \Rightarrow \quad e_\infty = 0$$

The transfer function from the input disturbance $d_u(t)$ to the control error $e(t)$ is

$$G_{ed_u}(s) = \frac{-G(s)}{1 + G(s)C(s)} = \frac{-1}{s(s+1) + K_\mathrm{P}} \quad \Rightarrow \quad e_\infty = \lim_{s \to 0} G_{ed_u}(s) = -\frac{1}{K_\mathrm{P}}$$

c)   The input disturbance transfer function with the controller $C(s)$ is

$$G_{ed_u}(s) = \frac{-1}{s(s+1) + C(s)}$$

Applying the final value theorem with $\mathcal{L}[t\sigma(t)] = 1/s^2$ leads to

$$e_{\infty_{d_u}} = \lim_{s \to 0} \left[s \frac{-1}{s(s+1) + C(s)} \frac{1}{s^2} \right] = \lim_{s \to 0} \frac{-1}{sC(s)} \stackrel{!}{=} 0$$

thus $C(s)$ must have a factor $1/s^2$. Note however, that the numerator of the controller has to be appropriately chosen, to guarantee system stability!

d)   The Laplace transformation of $d_u(t)$ is

$$D_u(s) = \frac{\omega_d}{s^2 + \omega_d^2}.$$

This can be used to obtain the error signal in the Laplace domain

$$E_{d_u}(s) = \frac{-G(s)}{1 + G(s)C(s)} \cdot \frac{\omega_d}{s^2 + \omega_d^2}.$$

Writing $C(s) = n_c(s)/d_c(s)$ and $G(s) = b(s)/a(s)$, where from the task description $b(s) = 1$ and $a(s) = s(s+1)$ are known, leads to

$$E_{d_u}(s) = \frac{-b(s)d_c(s)}{a(s)d_c(s) + b(s)n_c(s)} \cdot \frac{\omega_d}{s^2 + \omega_d^2} = \frac{-d_c(s)}{s(s+1)d_c(s) + n_c(s)} \cdot \frac{\omega_d}{s^2 + \omega_d^2}.$$

The final value theorem to obtain the error $e_{\infty_{d_u}}$ as $t \to \infty$ can only be used when the limit exists, i.e. when the error converges to a static value. If the partial fraction expansion of $E_{d_u}(s)$ is considered, it becomes apparent from

$$E_{d_u}(s) = \underbrace{\frac{C_1}{s + p_1} + \dots + \frac{C_n}{s + p_n}}_{\text{Transient response, that dies out}} + \underbrace{\frac{C_{\omega_d}}{s + j\omega_d} + \frac{\bar{C}_{\omega_d}}{s - j\omega_d}}_{\text{Undamped oscillation}},$$

that $e_{d_u}(t)$ will contain a term $\frac{|C_{\omega_d}|}{\omega_d} \sin(\omega_d t)$ which will not vanish unless $C_{\omega_d} = \bar{C}_{\omega_d} = 0$. This is guaranteed by the choice of the controller denominator $d_c(s) = (s^2 + \omega_d^2)\tilde{d}_c(s)$ as this eliminates the imaginary roots of $(s^2 + \omega_d^2)$. With this choice

$$E_{d_u}(s) = \frac{-b(s)\tilde{d}_c(s)\omega_d}{a(s)\tilde{d}_c(s)(s^2 + \omega_d^2) + b(s)n_c(s)} = \frac{-\tilde{d}_c(s)\omega_d}{s(s+1)\tilde{d}_c(s)(s^2 + \omega_d^2) + n_c(s)},$$

is obtained and it is left to adequately choose $\tilde{d}_c(s)$ and $n_c(s)$, in order to obtain a stable closed-loop. In fact, in practice the procedure would usually be the other way round: First a stabilizing controller is designed (by the methods described in Chapters 3 and beyond) and then the controller is eventually augmented by

including the roots of the disturbance signal's Laplace transform denominator into the denominator of the controller.

A controller that stabilises the system and achieves zero steady-state error for $\omega_d = 1$ is:

$$C(s) = 5\frac{s^2 + 0.2s + 1}{s^2 + 1}.$$

The way how to design such a controller is discussed in Problem 3.6.

SIMULINK file: `Model_Problem2_8_InternalModelPrincipleSin.slx`

Problem 2.9 (Ramp tracking)

a)   Since we are interested in the error when tracking a *ramp* signal, we use the velocity error constant

$$K_{\text{vel}} = \lim_{s \to 0} sL(s) = \lim_{s \to 0} sK_{\text{P}}\frac{21}{s(1.1s + 1)} = 21K_{\text{P}}.$$

To achieve the required accuracy we need

$$e_\infty = \frac{1}{K_{\text{vel}}} = K_{\text{P}}\frac{1}{21} < 0.15$$

and thus

$$K_{\text{P}} > \frac{1}{0.15 \cdot 21} = \frac{20}{63}.$$

b)   Obviously, the controller should exhibit integral action, i.e. PI and PID controllers will achieve $e(t) \to 0$.

Simulation results with P, PI and PID controllers and K_{P} as in (a) are shown in Figures A.37, A.38 and A.39, respectively. It is possible to achieve qualitatively similar results with the experimental setup.
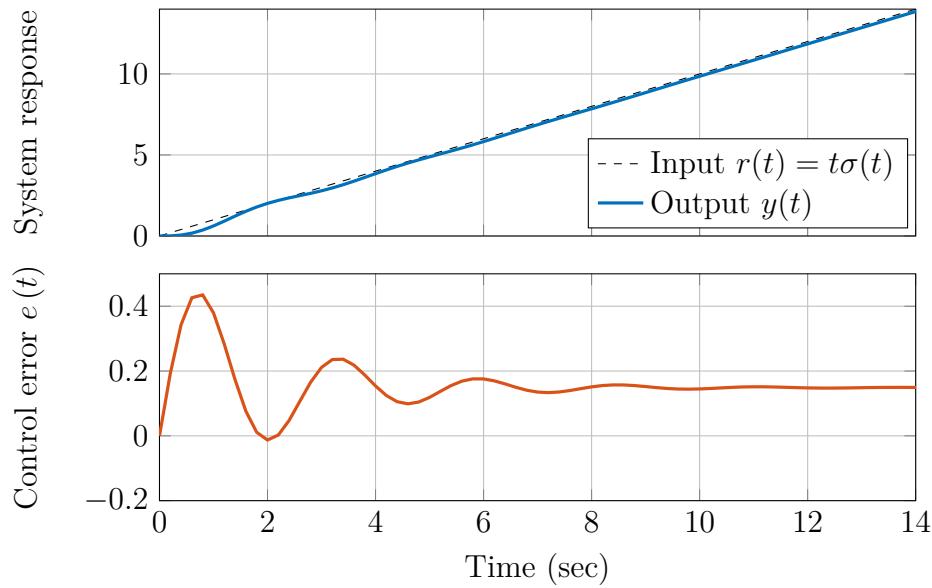
Figure A.37: Response to unit ramp with P controller
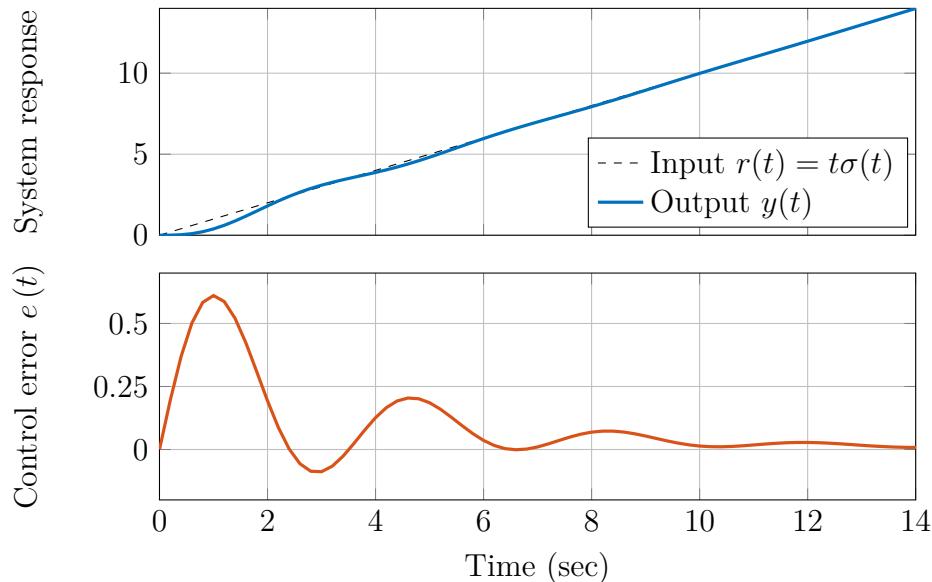
Figure A.38: Response to unit ramp with PI controller
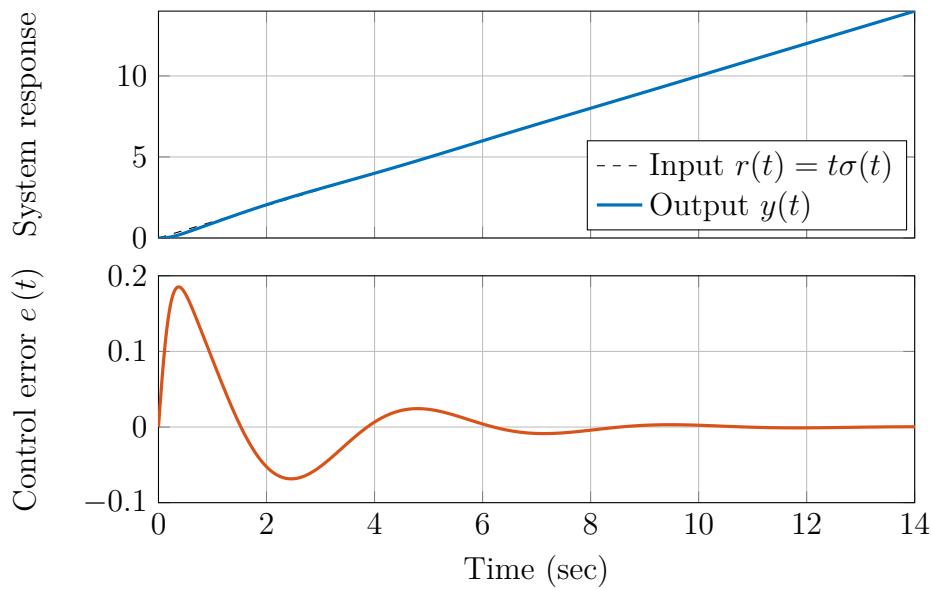
Figure A.39: Response to unit ramp with PID controller

SIMULINK files:

- `Model_Problem2_9_PAngleRamp.slx`
- `Model_Problem2_9_PDAngleRamp.slx`
- `Model_Problem2_9_PIDAngleRamp.slx`

A.3 Solutions for Chapter 3

Problem 3.1 (Root locus)

MATLAB function `rlocus`

Construction of the root-locus plot:

1. Draw all poles and zeros. The transfer function has one zero $z_1 = -4$ and three poles at $p_1 = 0$, $p_2 = -1$, $p_3 = -7$, respectively.

2. Determine the sections on the real axis. Left of an odd number of poles and zeros lie the intervals $[-1, 0]$ and $[-7, -4]$.

3. Determine the asymptotes. The pole excess $n - m$ is 2, therefore there are two asymptotes, with the angles $\pm 90^o$. Calculating the asymptote intersection point gives

$$\alpha = \frac{\sum_{i=1}^n p_i - \sum_{i=1}^m z_i}{n - m} = \frac{0 - 1 - 7 + 4}{2} = -2$$

    Draw the asymptotes.

4. Calculate the departure angles for small gains. From point 2 it is clear that the poles will move first along the real axis. This can be easily checked, for example for pole $p_2 = -1$ we have

$$\arg(L(s)) = \sum_{i=1}^m \arg(s - z_i) - \sum_{i=1}^n \arg(s - p_i)$$
$$= \arg(s + 4) - \arg(s) - \arg(s + 1) - \arg(s + 7)$$

    or

$$0^o - 180^o - \phi_2 - 0^o = 180^o + l360^o$$

    It follows that for $l = -1$ the angle $\phi_2 = 0$, i.e. the pole at $p_2 = -1$ will move towards the right along the real axis.

5. *It is possible - even though not practical - to use the rule of step 5 to compute the break point. With the numerator polynomial $b(s) = s + 4$ and the denominator polynomial $a(s) = s^3 + 8s^2 + 7s$ we obtain*

$$b\frac{da}{ds} - a\frac{db}{ds} = (s + 4)(3s^2 + 16s + 7) - (s^3 + 8s^2 + 7s) \cdot 1$$
$$= 2s^3 + 20s^2 + 64s + 28 = 0$$

    The only real root of the polynomial $s^3 + 10s^2 + 32s + 14$ is at $\beta = -0.5166$ (MATLAB command `roots`), where the two branches break away from the real axis. Note however that for a third order polynomial this is not easily done by hand.

The root-locus plot is shown in Figure A.40.
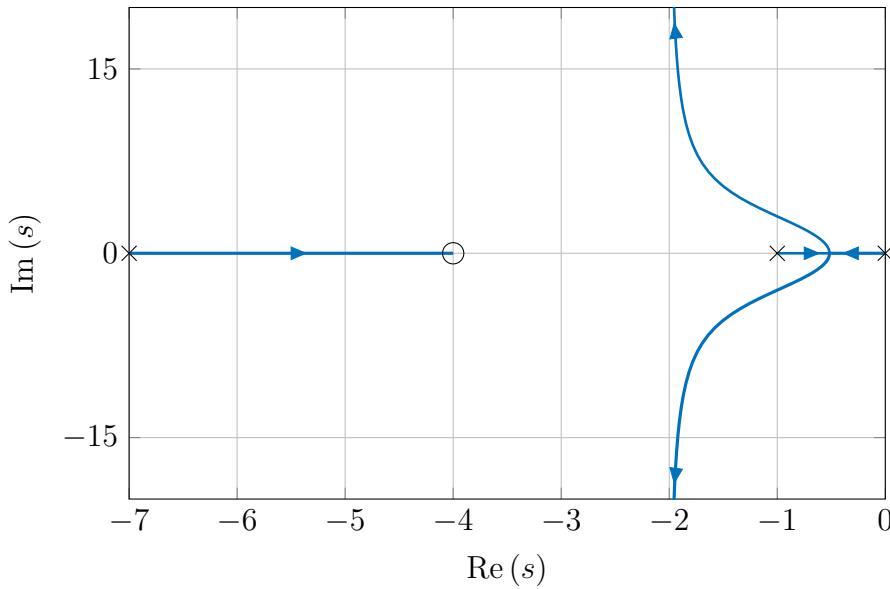
Figure A.40: Root locus

Problem 3.2 (Root locus with complex pole pair)

a)   The open loop transfer function can be simplified as follows

$$L(s) = K_P \frac{s+2}{(s^2+4s+3)(s^2+4s+5)} = K_P \frac{s+2}{(s+3)(s+1)(s+2+i)(s+2-i)} \ .$$

Construction of the root-locus plot:

1.   Draw zeros at $z_1 = -2$ and poles at $p_1 = -3$, $p_2 = -1$, $p_3 = -2 - i$ and $p_4 = -2 + i$.

2.   The sections on the real axis are $(-\infty, -3]$ and $[-2, -1]$.

3.   The pole excess $n - m$ is 3. There will be three asymptotes: one with the angle $+60^o$, one with -60° and one with 180°. The asymptote intersection point is

$$\alpha = \frac{-3 - 1 - 2 - 2 + 2}{3} = -2 \ .$$

4.   The angles of departure reveal to:

$$p_1 = -3 : \quad \Phi_1 = -180^o - (+135^o) - 180^o - (-135^o) + 180^o + l \cdot 360^o = 180^o$$
$$p_2 = -1 : \quad \Phi_2 = -180^o - 0^o - 45 - (-45^o) + 0^o + l \cdot 360^o = 180^o$$
$$p_3 = -2 - i : \quad \Phi_4 = -180^o - (-45^o) - (-90^o) - (-135^o) + (-90^o) + l \cdot 360^o = 0^o$$
$$p_4 = -2 + i : \quad \Phi_3 = -180^o - 45^o - 90^o - 135^o + 90^o + l \cdot 360^o = 0^o$$

5.   Calculating the intersection points is not applicable.

The root-locus plot is shown in Figure A.41.

Figure A.41: Root locus

b)   Since there are 3 asymptotes, as K_P becomes larger two closed loop poles will move into the right half plane. This can be seen in Figure A.41. Therefore, the system will become unstable for large K_p.

c)   The pole excess will be 2 and there will be two asymptotes: One with $+90^o$ and one with -90°. Let the location of the new zero be x, then the asymptote intersection is

$$\alpha = \frac{-3 - 1 - 2 - 2 + 2 - x}{2} = -3 - \frac{x}{2} \ .$$

The asymptote intersection and the location of the zero must not be in the right half plane. Therefore the location of the zero reveals to $-6 < x < 0$.

d)   Construction of the root-locus plot:

1.   Draw zeros at $z_1 = -2$ and poles at $p_1 = -3$, $p_2 = -1$, $p_3 = -2 - i$ and $p_4 = -2 + i$.

2.   The sections on the real axis are $[-3, -2]$ and $[-2, \infty]$.

3.   The pole excess $n - m$ is 3. There will be three asymptotes: one with the angle $+120^o$, one with -120° and one with 0°. The asymptote intersection point is

$$\alpha = \frac{-3 - 1 - 2 - 2 + 2}{3} = -2 \ .$$

4. The angles of departure reveal to:

$$p_1 = -3: \quad \Phi_1 = -(+135^o) - 180^o - (-135^o) + 180^o + l \cdot 360^o = 0^o$$

$$p_2 = -1: \quad \Phi_2 = -0^o - 45 - (-45^o) + 0^o + l \cdot 360^o = 0^o$$

$$p_3 = -2 - i: \quad \Phi_4 = -(-45^o) - (-90^o) - (-135^o) + (-90^o) + l \cdot 360^o = 180^o$$

$$p_4 = -2 + i: \quad \Phi_3 = -45^o - 90^o - 135^o + 90^o + l \cdot 360^o = 180^o$$

5. Calculating the intersection points is not applicable.

The root-locus plot is shown in Figure A.42.

Figure A.42: Root locus

Problem 3.3 (PD controller for a valve)

a) This is a real PD controller.

b)

$$C(s) = K_P \left(1 + \frac{T_D s}{1 + \gamma T_D s} \right)$$

$$= K_P \left(\frac{1 + \gamma T_D s + T_D s}{1 + \gamma T_D s} \right)$$

$$= K_P \frac{\gamma T_D + T_D}{\gamma T_D} \left(\frac{s + \frac{1}{T_D(\gamma+1)}}{s + \frac{1}{\gamma T_D}} \right)$$

$$= K_P \frac{\gamma + 1}{\gamma} \left(\frac{s - \left(-\frac{1}{T_D(\gamma+1)} \right)}{s - \left(-\frac{1}{\gamma T_D} \right)} \right)$$

Thus

$$K = K_P \frac{\gamma + 1}{\gamma}; \qquad z = -\frac{1}{T_D(\gamma + 1)}; \qquad p = -\frac{1}{\gamma T_D}$$

c) The root locus is shown in Figure A.43.

It is rather easy to see whether the closed loop system is stable. Also, it is possible to design controllers with the root locus by hand. However, it is much easier with MATLAB.
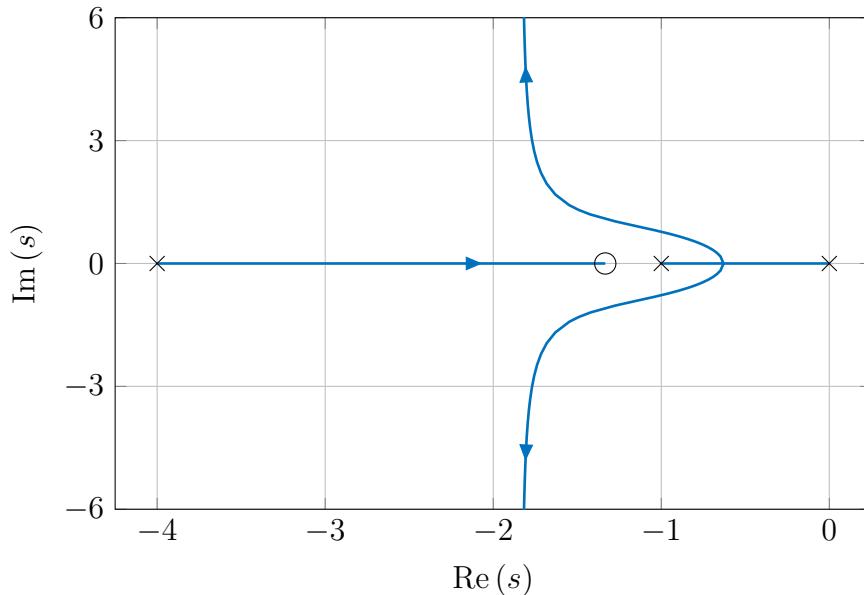
Figure A.43: Root locus of the valve with a PD controller

Problem 3.4 (PD controller for a valve, computer-based)

a) A possible controller - obtained with `rltool` - is given by $K = 0.9889$, $z = -1.57$ and $p = -4.48$, corresponding to parameter values

$$K_P = 0.3466; \qquad T_D = 0.4137; \qquad \gamma = 0.5396$$

The step response with this controller displays overshoot $= 4.99\%$, $u_{max} = 0.989$ and rise time $= 1.64$s.

MATLAB file: `Sol_Problem3_4_PDControlledValve.mlx`

b) A ramp input is not available in `rltool`; however, with the help of the prefilter block `F` a step input r can be integrated to generate a ramp ($F(s) = 1/s$). This can be done in `rltool` by clicking the block "F". From *Analysis–Response to step command* the desired ramp signal can be obtained. The steady state error can be obtained with the parameters chosen in (a) and with the help of the velocity error constant

$$K_{vel} = K_P K_0 = 1.04 \quad \Rightarrow \quad e_\infty = \frac{1}{K_{vel}} = 0.96$$

Problem 3.5 (Controller design)

a) The static gain of the additional compensator is determined by the required steady state error of less than or equal to 0.1:

$$K_{vel} = \frac{1}{e_\infty} = \frac{1}{0.1} = 10 \Rightarrow 10 \overset{!}{\leq} \lim_{s \to 0} s \cdot \frac{3}{s(s+1)} \frac{s+1.5}{s+4.5} \cdot C_2(s) \Rightarrow \lim_{s \to 0} C_2(s) \geq 10$$

Because the static gain is asked to be minimal,

$$\lim_{s \to 0} C_2(s) = 10.$$

b) The pole-zero pair is then placed close to zero, in order to not affect the transient behaviour achieved in Problem 3.3, for example by choosing

$$C_2(s) = \frac{s+0.1}{s+0.01}.$$

c) The responses can be seen in Figure A.44 .

MATLAB file: `Sol_Problem3_5_RlocusLeadLag.mlx`
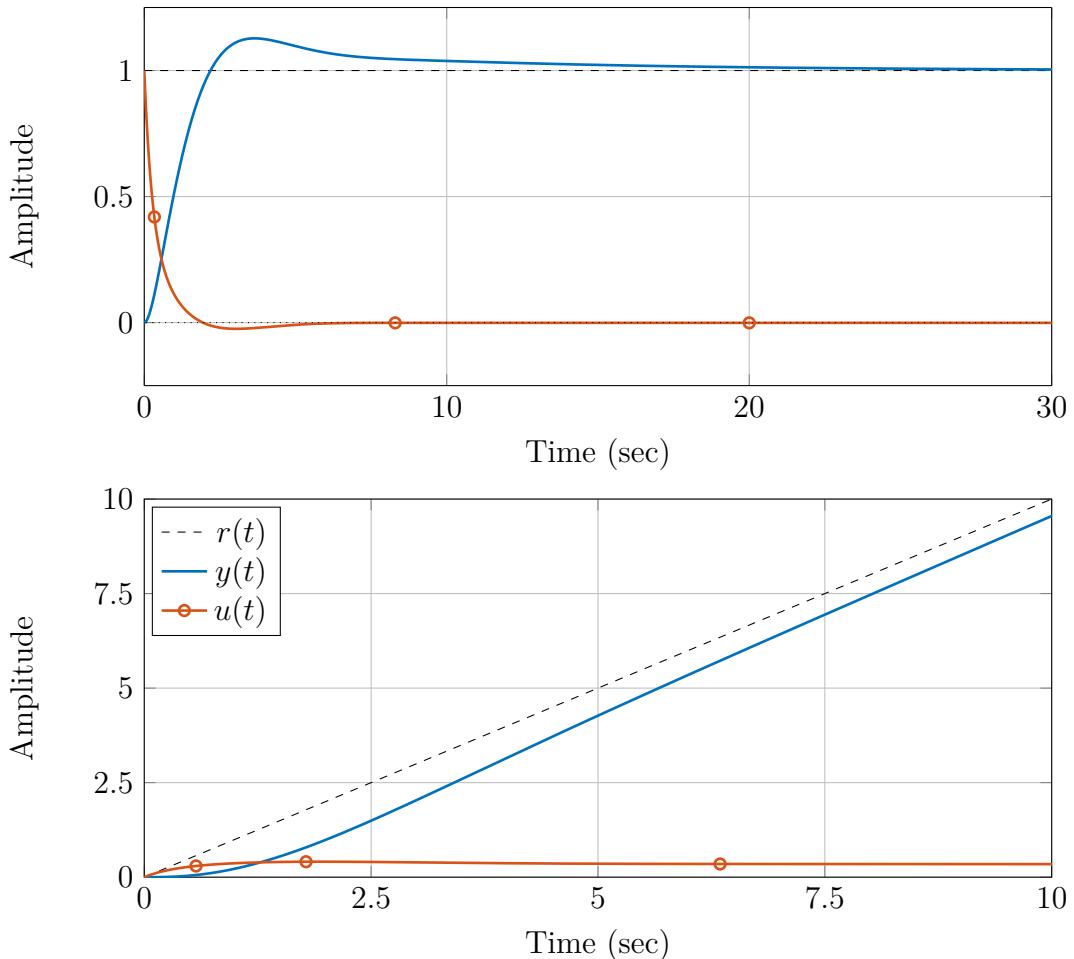
Figure A.44: Simulated response to a step input (top) and simulated response to a ramp input (bottom)

Problem 3.6 (Internal Model Principle)

a)   To have a stable closed loop the zeros have to be placed in the left half plane. With the help of the `rltool` you can see that the zeros can be located in an interval $[0, -0.45]$ so that the closed loop system remains stable.

b)   Figure A.45 and A.46 show the two possibilities for choosing the zeros. Both Figures show that for large values of K the system is stable.

c)   Figure A.47 shows a Simulink model that can be used to find out how the system behaves for different controller values. Figure A.48 shows the response of the loop to the sinus disturbances with the controller C obtained from Figure A.49. This controller meets all constraints.

Simulink file: `Model_Problem3_6_InternalModelPrincipleSin.slx`
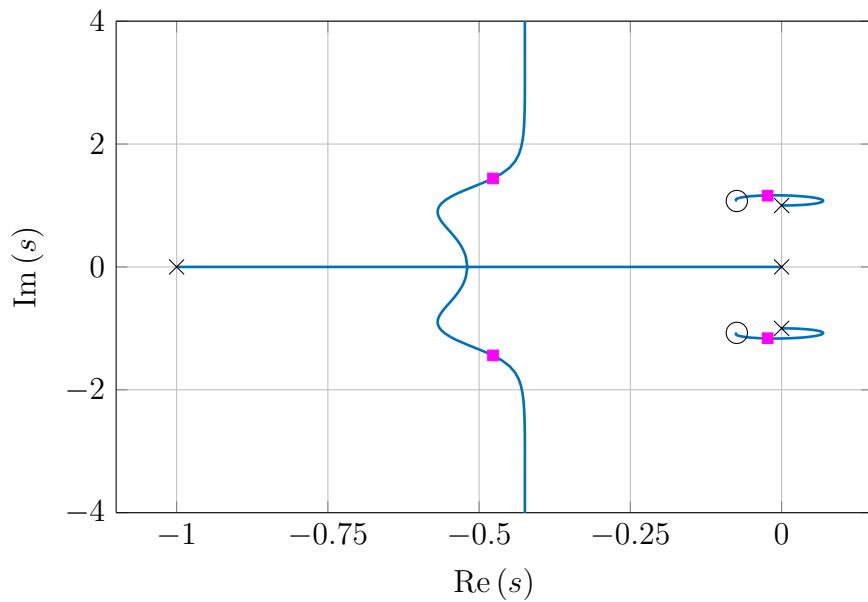
Figure A.45: Complex pole pair moves to the zeros
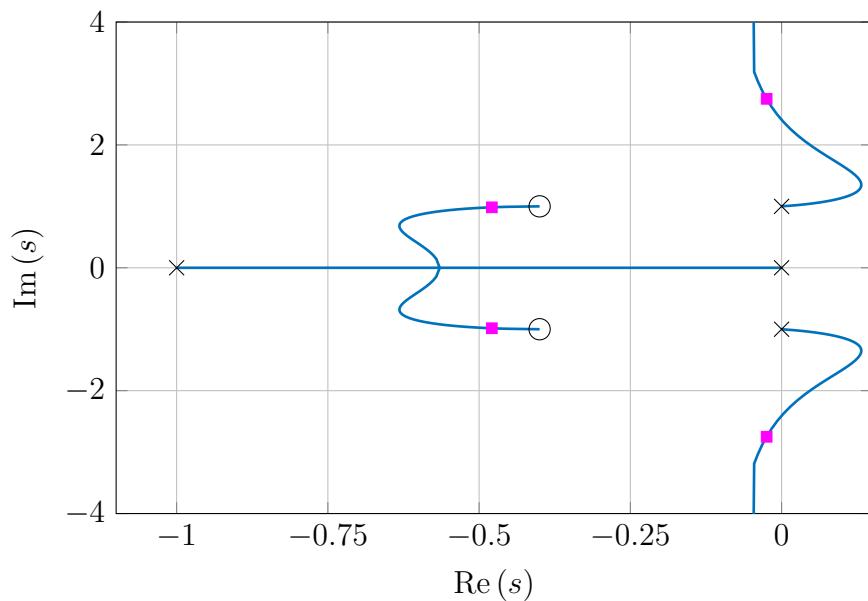
Figure A.46: Real pole pair moves to the zeros
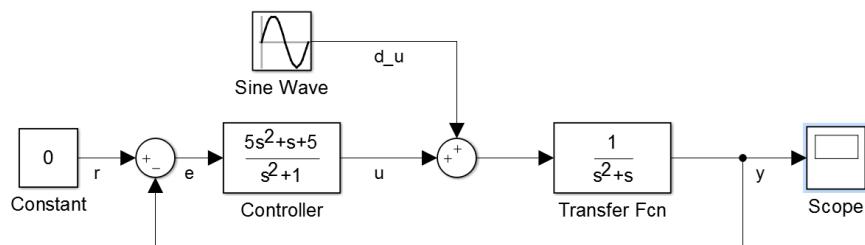
Figure A.47: SIMULINK model with sinus as input
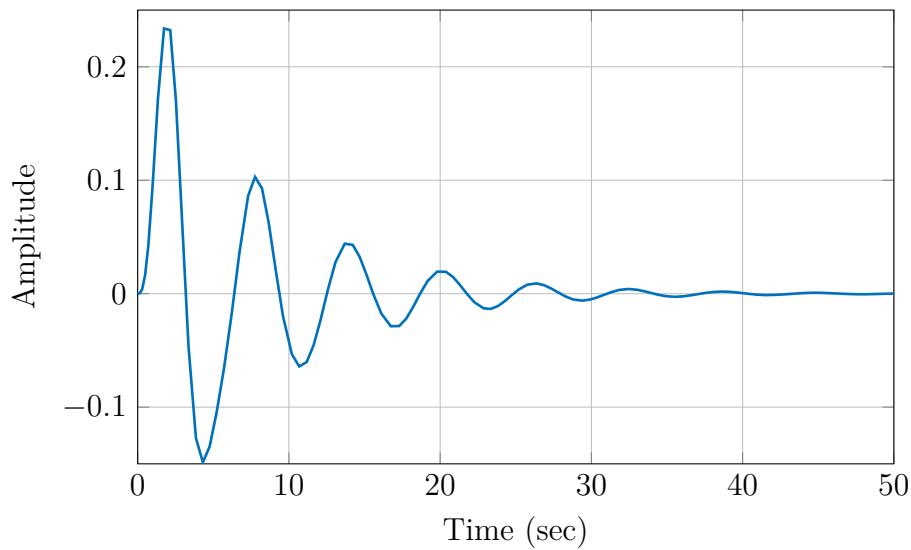
Figure A.48: Closed-loop response to the sinus disturbance
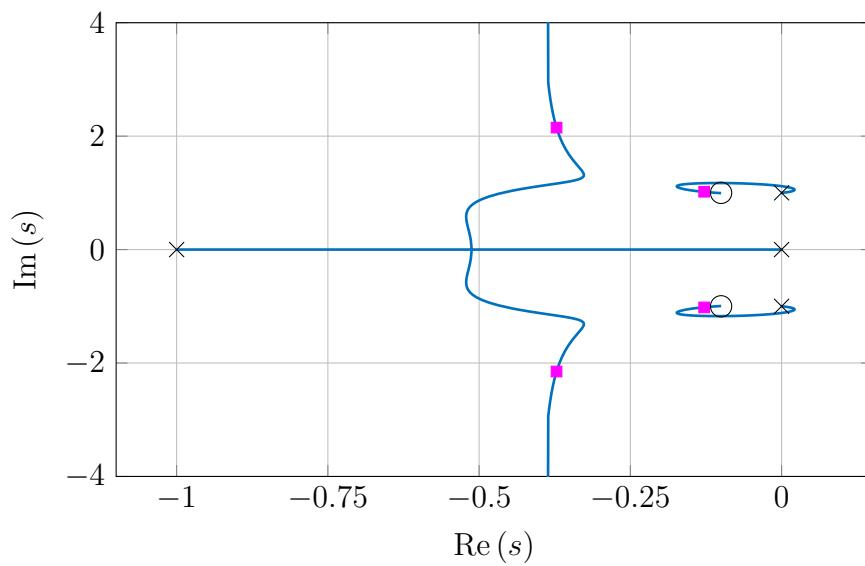
Figure A.49: Root locus design

Problem 3.7 (Time delay systems and approximation)

a) The first three coefficients from the series expansion are

$$e^{-T_d s} = 1 - \frac{T_d s}{1!} + \frac{(T_d s)^2}{2!} - \ldots = 1 - T_d s + \frac{T_d^2}{2!} s^2 - \ldots$$

$$f(s) = \frac{\beta_1 s + \beta_0}{\alpha_1 s + 1} = f(0) + \frac{f'(0)}{1!} s + \frac{f''(0)}{2!} s^2 + \ldots$$

With the derivatives of $f(s)$

$$f'(s) = \frac{\beta_1(\alpha_1 s + 1) - (\beta_1 s + \beta_0)\alpha_1}{(\alpha_1 s + 1)^2} = \frac{\beta_1 - \alpha_1 \beta_0}{(\alpha_1 s + 1)^2}$$

$$f''(s) = \frac{-2\alpha_1(\beta_1 - \alpha_1 \beta_0)}{(\alpha_1 s + 1)^3}$$

comparing coefficients componentwise leads to

$$f(0) = \beta_0 = 1$$
$$f'(0) = \beta_1 - \alpha_1 \beta_0 = -T_d$$
$$f''(0) = 2\alpha_1(\alpha_1 \beta_0 - \beta_1) = 2\alpha_1 T_d = T_d^2$$

The coefficients can be calculated as $\alpha_1 = \frac{T_d}{2}$, $\beta_1 = -\frac{T_d}{2}$. The first order Padé approximation is then

$$e^{-T_d s} = \frac{-\frac{T_d s}{2} + 1}{\frac{T_d s}{2} + 1} = \frac{2 - T_d s}{2 + T_d s}$$

b) MATLAB solution "`Sol_Problem3_7_TimeDelayPadeCompare.mlx`". See Figure A.50.

c) The characteristic equation is

$$1 + K_P \frac{e^{-T_d s}}{s + 1} = 0 \quad \Rightarrow \quad \frac{e^{-T_d s}}{s + 1} = \frac{-1}{K_P}$$

The phase condition gives

$$\arg e^{-T_d s} - \arg (s + 1) = 180^o + l360^o$$

The argument of the time delay is

$$\arg e^{-T_d s} = \arg e^{-T_d j\omega} = \begin{cases} -T_d \omega & \text{(in rad)} \\ -T_d \omega \frac{180^o}{\pi} & \text{(in degrees)} \end{cases}$$

With $l = -1$ we have $\arg(s + 1) = 180^o - T_d \omega \, 57^o$ and thus

$$\begin{aligned} \omega = 1 \quad &: \arg (s + 1) = 180^o - 57^o = 123^o \\ \omega = 2 \quad &: \arg (s + 1) = 180^o - 114^o = 66^o \\ \omega \to \pi \quad &: \arg (s + 1) \to 0^o \end{aligned}$$

See Figure A.51.
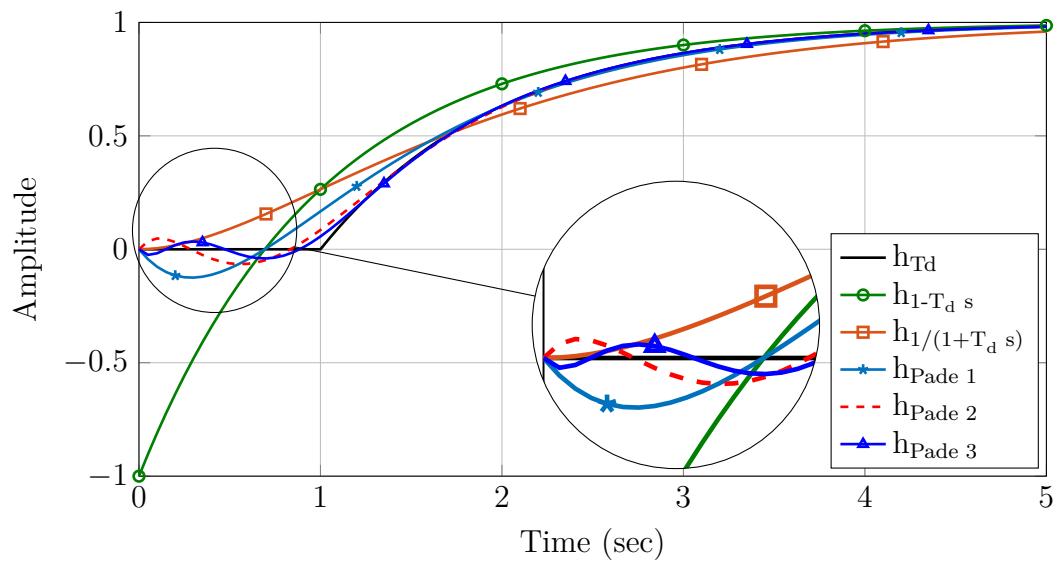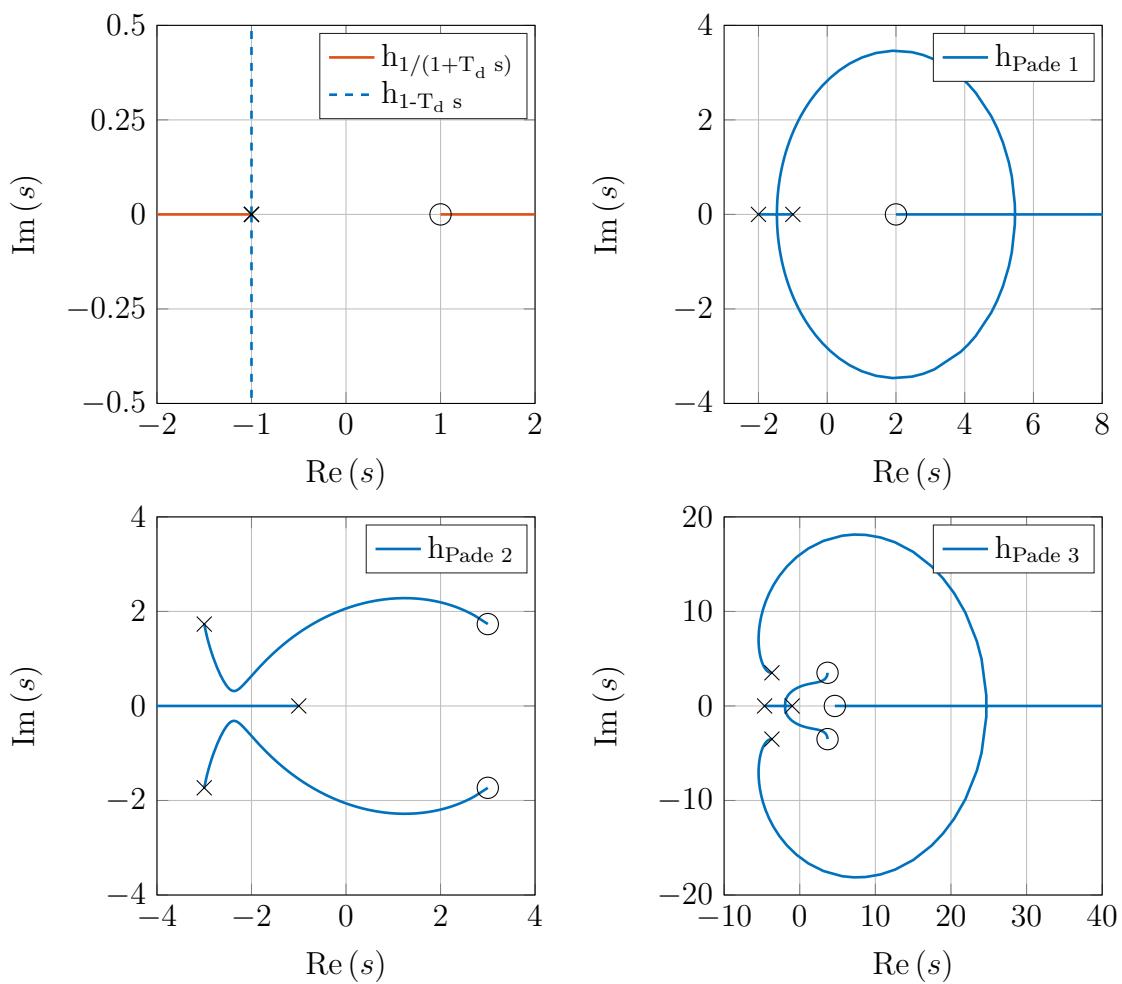
Figure A.50: Time delay approximation for $T_d = 1$s

Figure A.51: Root locus

Problem 3.8 (Smith predictor)

a) i) The transfer function from r input to y output is:

$$G_{yr}(s) = \frac{Y(s)}{R(s)} = \frac{C(s)G_0(s)e^{-T_d s}}{1 + C(s)G_0(s)e^{-T_d s}}$$

   ii) The transfer function $C(s) = G_{ue}(s)$ is:

$$C(s) = G_{ue}(s) = \frac{U(s)}{E(s)} = \frac{C_0(s)}{1 + (1 - e^{-T_d s})G_0(s)C_0(s)}$$

   iii) The transfer function $G_{yr}(s)$ with part ii) is

$$\frac{Y(s)}{R(s)} = \frac{C_0(s)G_0(s)e^{-T_d s}}{1 + (1 - e^{-T_d s})G_0(s)C_0(s) + C_0(s)G_0(s)e^{-T_d s}}$$

which simplifies to

$$\frac{Y(s)}{R(s)} = \frac{C_0(s)G_0(s)}{1 + C_0(s)G_0(s)}e^{-T_d s}.$$

b) With the results from part iii) the controller $C_0(s)$ can be designed as if there was no time delay. The simplified block diagramm is shown in Figure A.52.
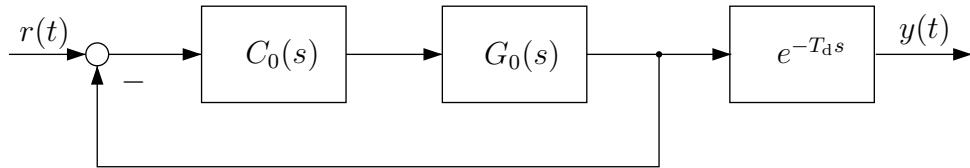
Figure A.52: Equivalent representation of the feedback system in Figure 3.29

Problem 3.9 (Smith predictor)

MATLAB file: `Sol_Problem3_9_RLToolSmithPredictor.mlx`

a) • To define the transfer function without time delay use the matlab command `G=tf(1, [1 1])`

   • Open the tool with `rltool(G)`

   • To design a controller of the form

$$C_0(s) = K_P \frac{s - z}{s}$$

the tool settings must be selected correctly. This can be done under *Preferences* → *Options* in the toolstrip. See Figure A.53 for reference.
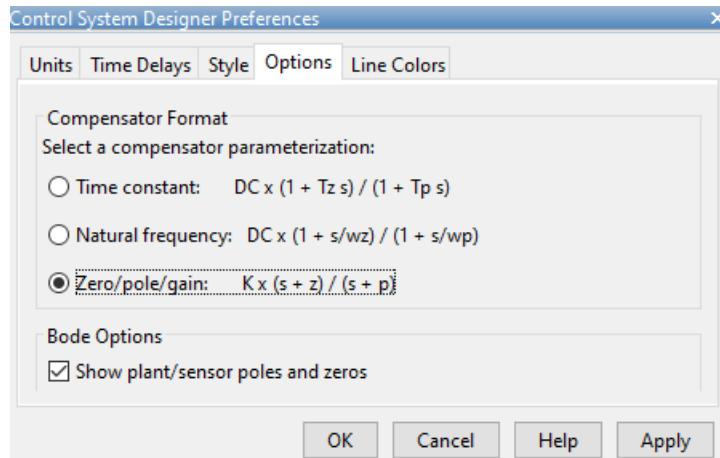
Figure A.53: Tool preferences

- You can now add the integrator and a zero by double clicking the controller `C` in `sisotool`.

- One possible controller which fulfills the design specifications is shown in Figure A.54.
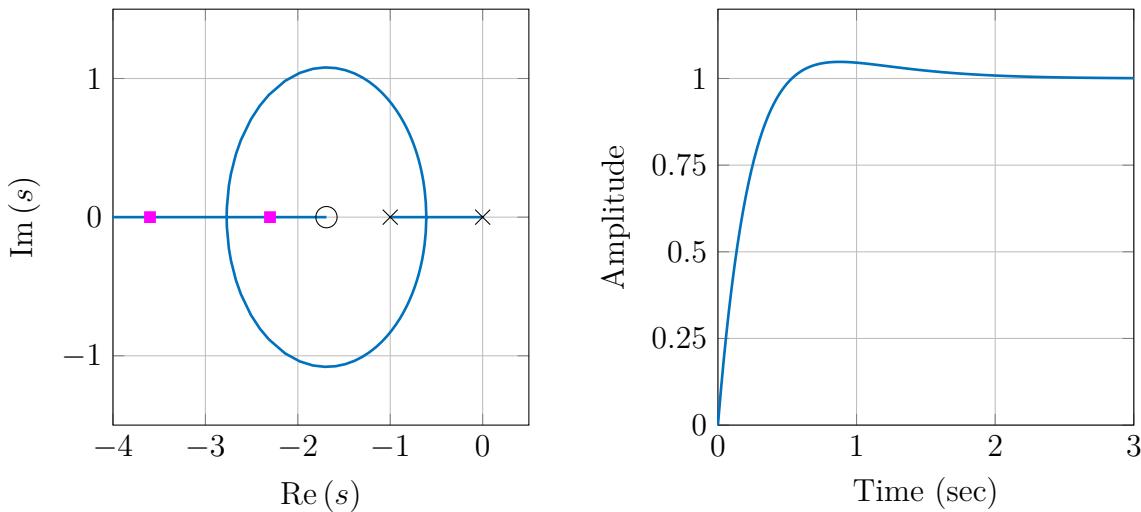
Figure A.54: One possible controller

b) Figure A.55 (i) shows the root locus and step response of the delayed system with the controller from (a). In (ii) the controller is modified to fulfill the the design specifications.

c) Figure A.56 shows a possible composition of a SIMULINK model. You can use the MATLAB solution `"Sol_Problem3_9_RLToolSmithPredictor.mlx"`. Reference tracking with the Smith predictor is shown in Figure A.57.
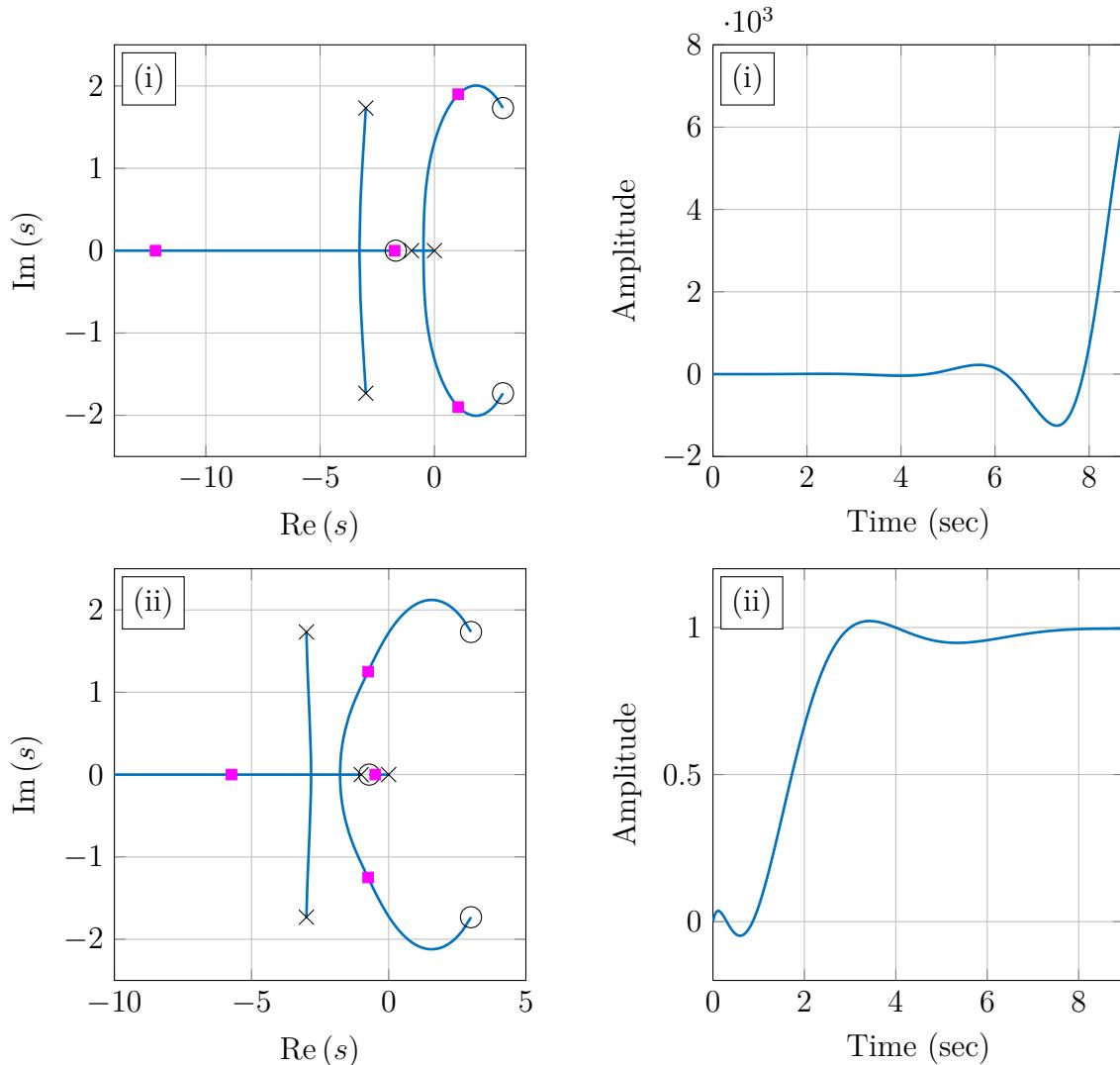
Figure A.55: Root locus and response of the system with time delay
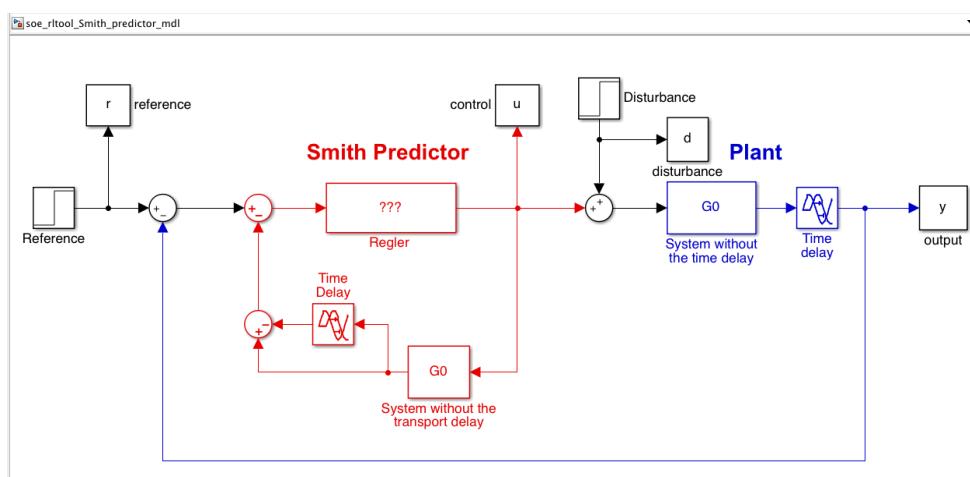
Figure A.56: Control loop with time delay
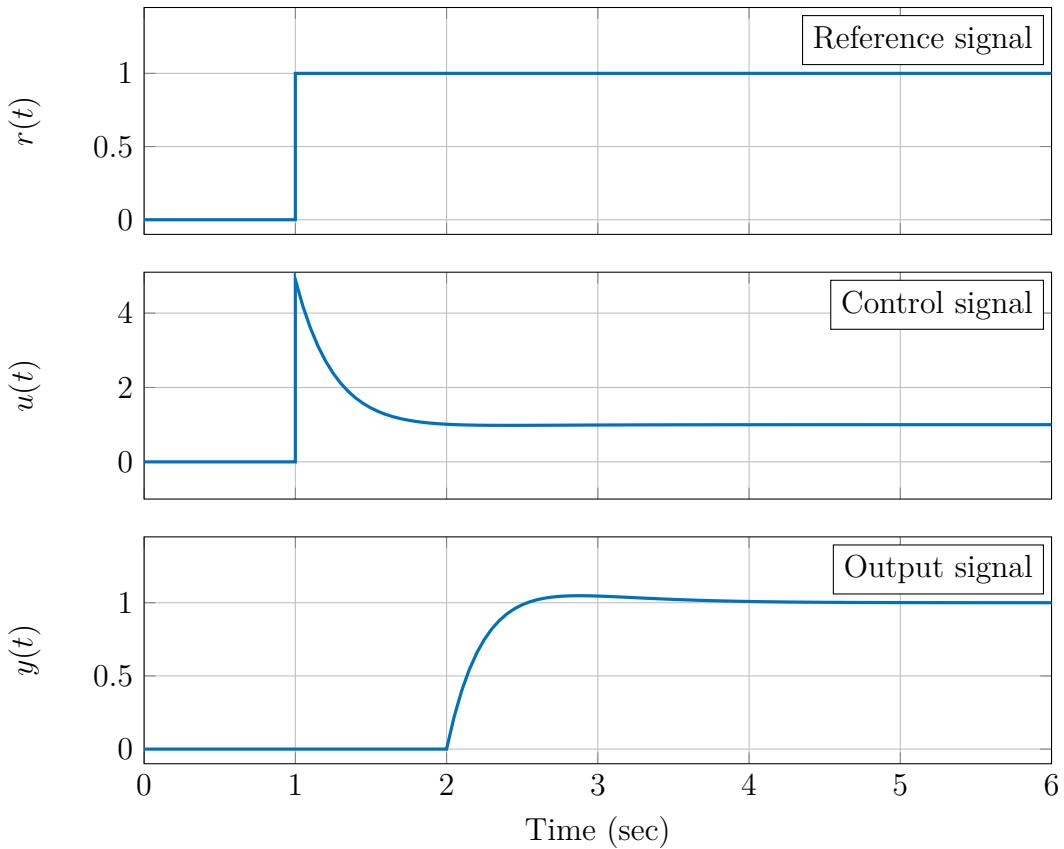
Figure A.57: Reference step response with Smith predictor

Problem 3.10 (Smith predictor by input disturbance)

MATLAB file: `Sol_Problem3_10_RLToolSmithPredictorDist.mlx`

a) Contained in the MATLAB solution, see Problem 3.9. The responses to step disturbance entering the system at time 5 s is shown in Figure A.58.

b) The closed-loop response to a reference unit step when the actual time delay of the plant is $T_d = 0.8$, is shown in Figure A.59.

c) With the plant $G(s) = G_0(s)e^{-T_d s}$ and the controller

$$C(s) = \frac{C_0}{1 + (1 - e^{-T_d s})G_0 C_0}$$

the transfer function from disturbance input to output is

$$G_{yd} = \frac{G(s)}{1 + G(s)C(s)} = \frac{\left(1 + (1 - e^{-T_d s})G_0 C_0\right)G_0 e^{-T_d s}}{\left(1 + (1 - e^{-T_d s})G_0 C_0\right) + G_0 e^{-T_d s}C_0}$$

$$= G_0 e^{-T_d s}\left(1 - \frac{G_0 C_0 e^{-T_d s}}{1 + G_0 C_0}\right)$$
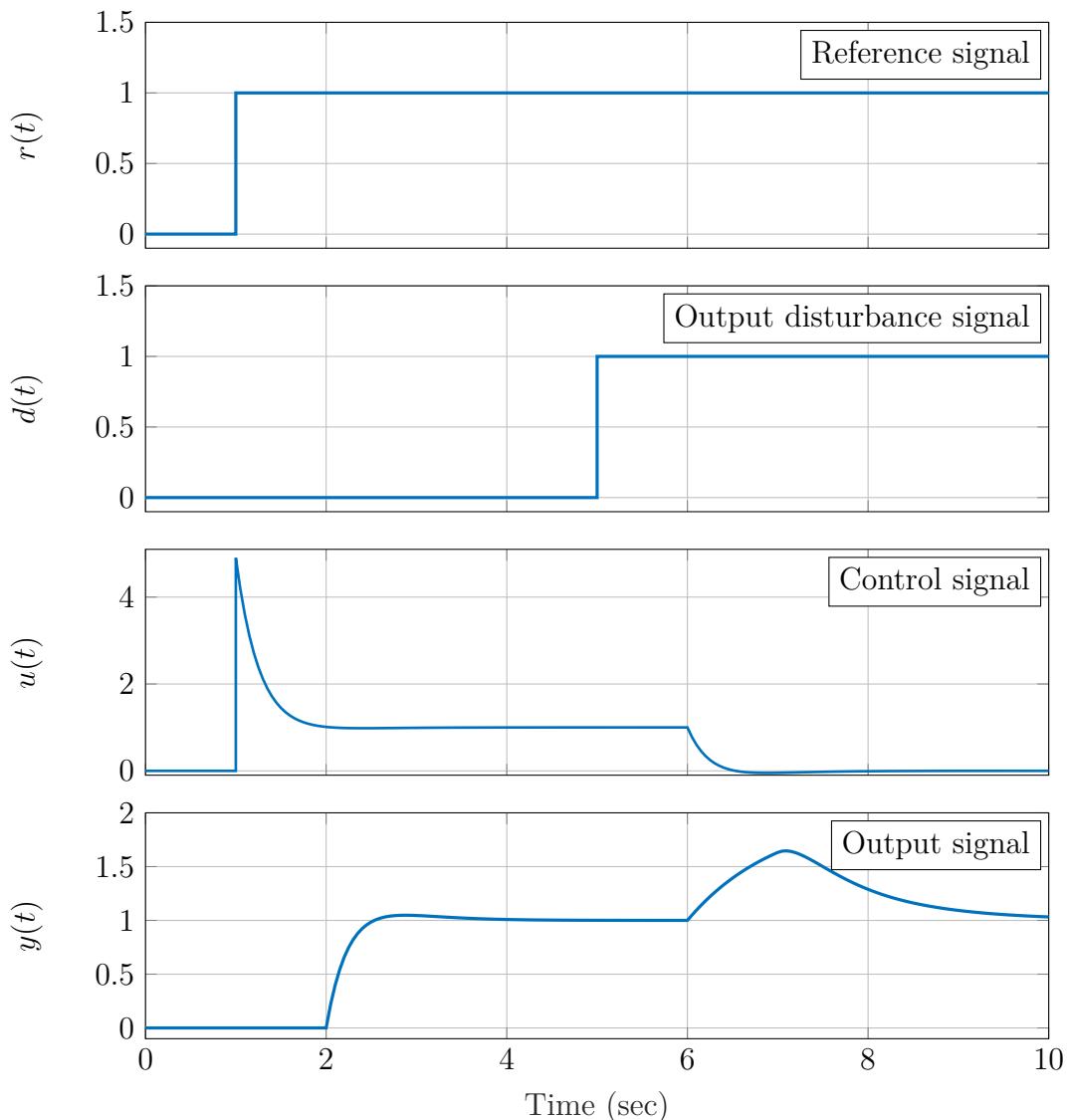
Figure A.58: Disturbance step response with Smith predictor

Thus, the poles of the disturbance transfer function are not only the poles of the closed loop transfer function from reference to output, but include also the *open loop poles*.
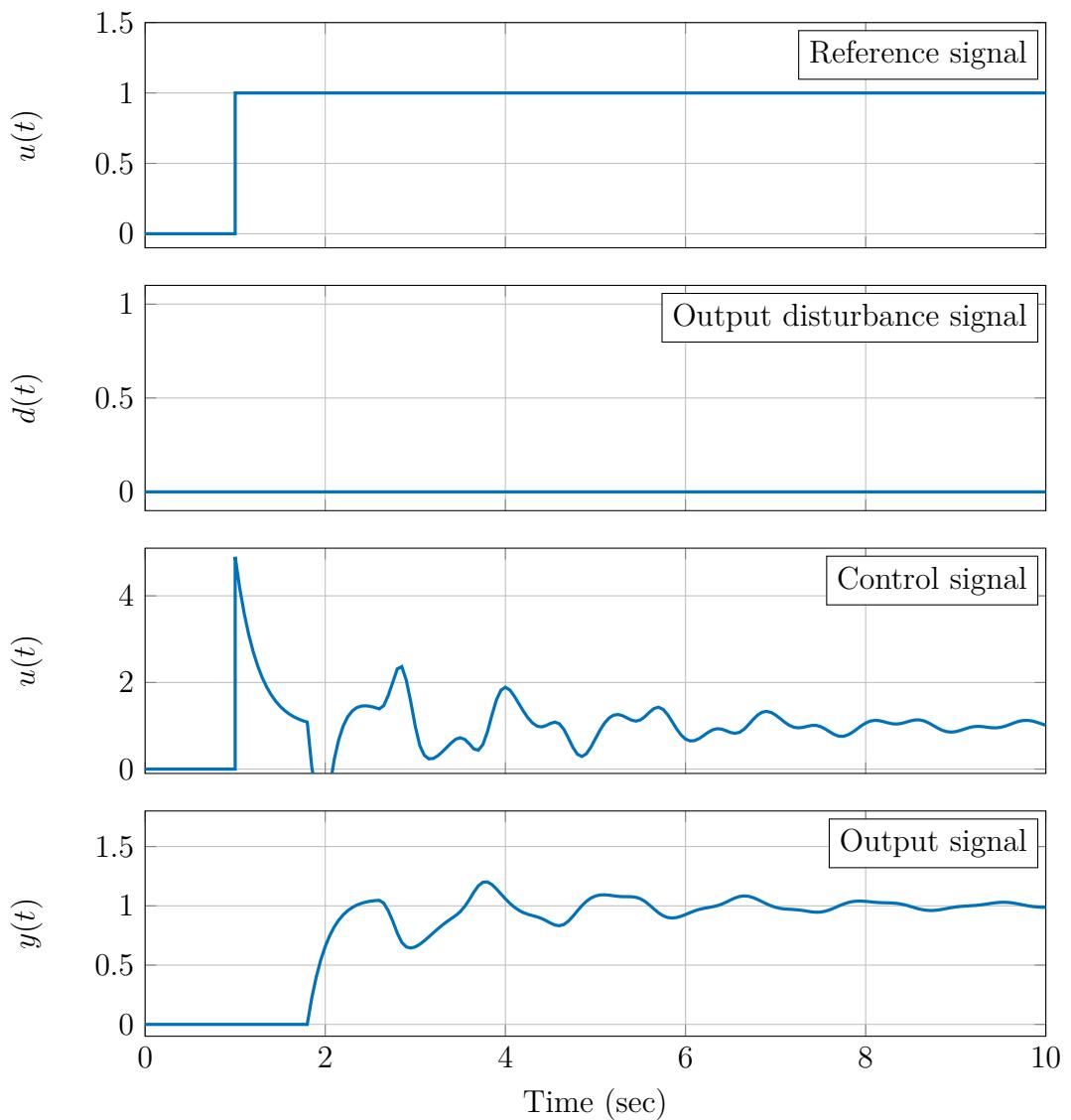
Figure A.59: Step response with Smith predictor

A.4   Solutions for Chapter 4

Problem 4.1 (System bandwidth)

a)   Substituting $j\omega$ for s and normalizing the real part of the pole factor to 1 gives

$$G(j\omega) = 10\frac{1}{1 + \frac{j\omega}{10}}.$$

The corner frequency is $\omega = 10\,\mathrm{rad/s}$. For frequencies below the corner frequency the gain is approximately $20\,\mathrm{dB}$ and for higher frequencies it rolls off with $-20\,\mathrm{dB/dec}$, see Figure A.60. The system bandwidth ω_b is the lowest frequency at which the gain drops $3\,\mathrm{dB}$ from its static value, which in this case is exactly the corner frequency $\omega = 10\,\mathrm{rad/s}$. The speed of response is determined by the time constant $\tau = 1/\omega_\mathrm{b} = 0.1\,\mathrm{s}$, see Figure A.61.
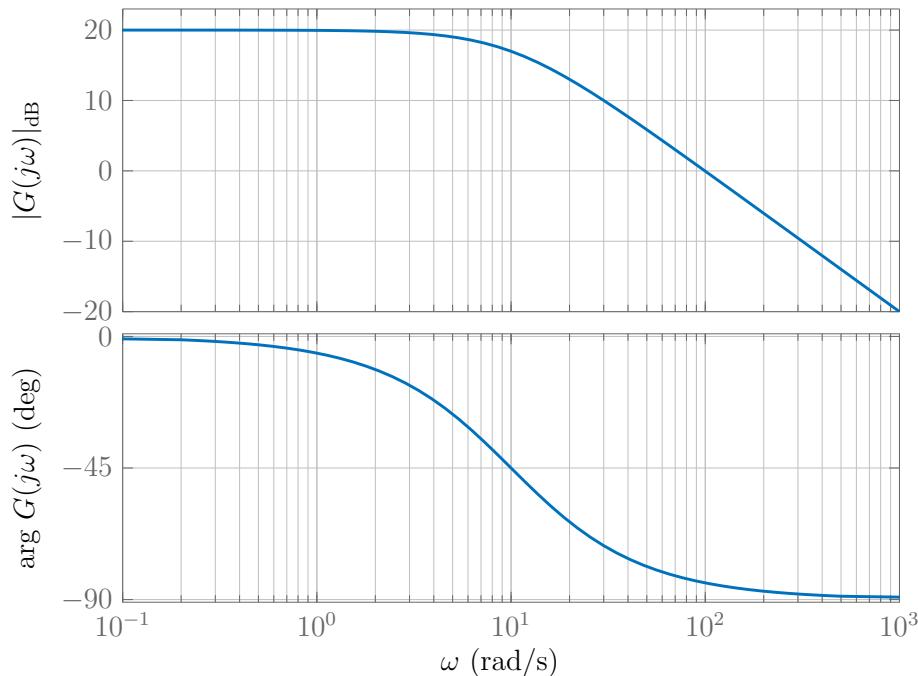
Figure A.60: Bode plot of the system in (a)

b)   The system can be seen as a combination of the above and a complex pole pair. We have a pole at -10 and a pole pair at $-0.3 \pm j0.954$. The complex pole pair is dominant and thus $\omega_\mathrm{b} \approx \omega_\mathrm{n} = 1\,\mathrm{rad/s}$ and $t_\mathrm{r} \approx 1.7/\omega_\mathrm{b} = 1.7\,\mathrm{s}$. The actual rise-time shown in the step response of Figure A.63 is $t_\mathrm{r} = 1.34\,\mathrm{s}$. For a hand-drawn sketch,
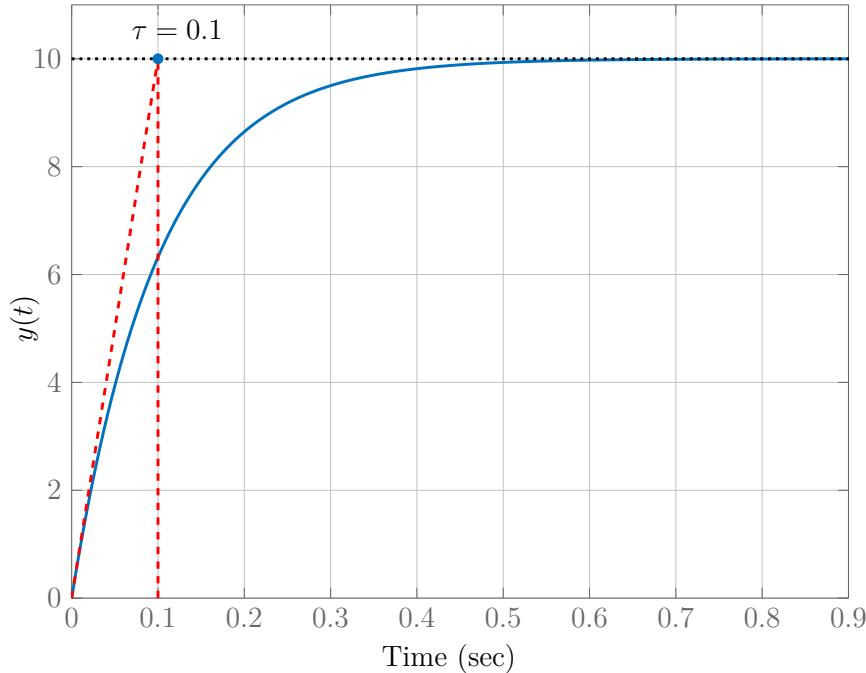
Figure A.61: Step response of the system in (a)

however, $t_\mathrm{r} \approx 1.7\,\mathrm{s}$ is sufficient. Take note of the resonant peak at $\omega_\mathrm{n} = 1\,\mathrm{rad/s}$ in the magnitude plot of Figure A.62.

Hint: Pay attention to the steps on page 123 and following you have to take to find out how large the resonant peak at the corner frequency is.

c) Bringing the system in the normal form one gets

$$G(s) = 10 \, \frac{(10 + 100s) \, (s - 10)}{s^2 + 101s + 100}$$

$$G(jw) = 10 \, \frac{10(1 + \frac{jw}{0.1}) \, (-10)(1 - \frac{jw}{10})}{(1 + jw) \, (100)(1 + \frac{jw}{100})}$$

$$G(jw) = -10 \, \frac{(1 + \frac{jw}{0.1}) \, (1 - \frac{jw}{10})}{(1 + jw) \, (1 + \frac{jw}{100})} \, .$$

From this one can see that the static gain is -10 and there will be:
- a zero in the LHP at $\omega_1 = 0.1\,\mathrm{rad/s}$
- a pole in the LHP at $\omega_2 = 1\,\mathrm{rad/s}$
- a zero in the RHP at $\omega_3 = 10\,\mathrm{rad/s}$
- a pole in the LHP at $\omega_4 = 100\,\mathrm{rad/s}$.
The bode plot can be seen in Figure A.64. The system is proper and therefore physically not realizable since there is no roll-off at high frequencies.
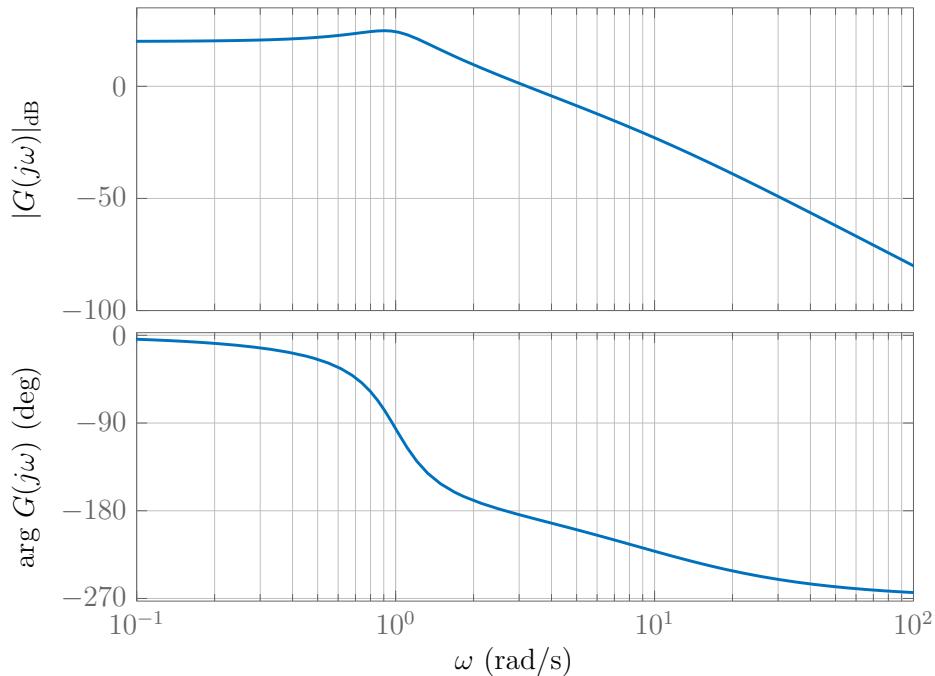
Figure A.62: Bode plot of the system in (b)
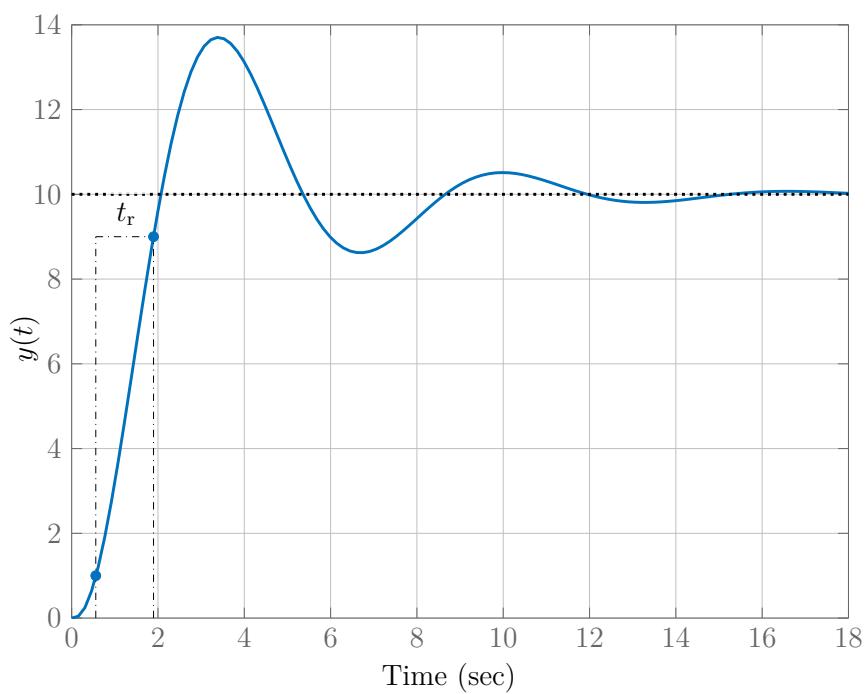
Figure A.63: Step response of the system in (b)
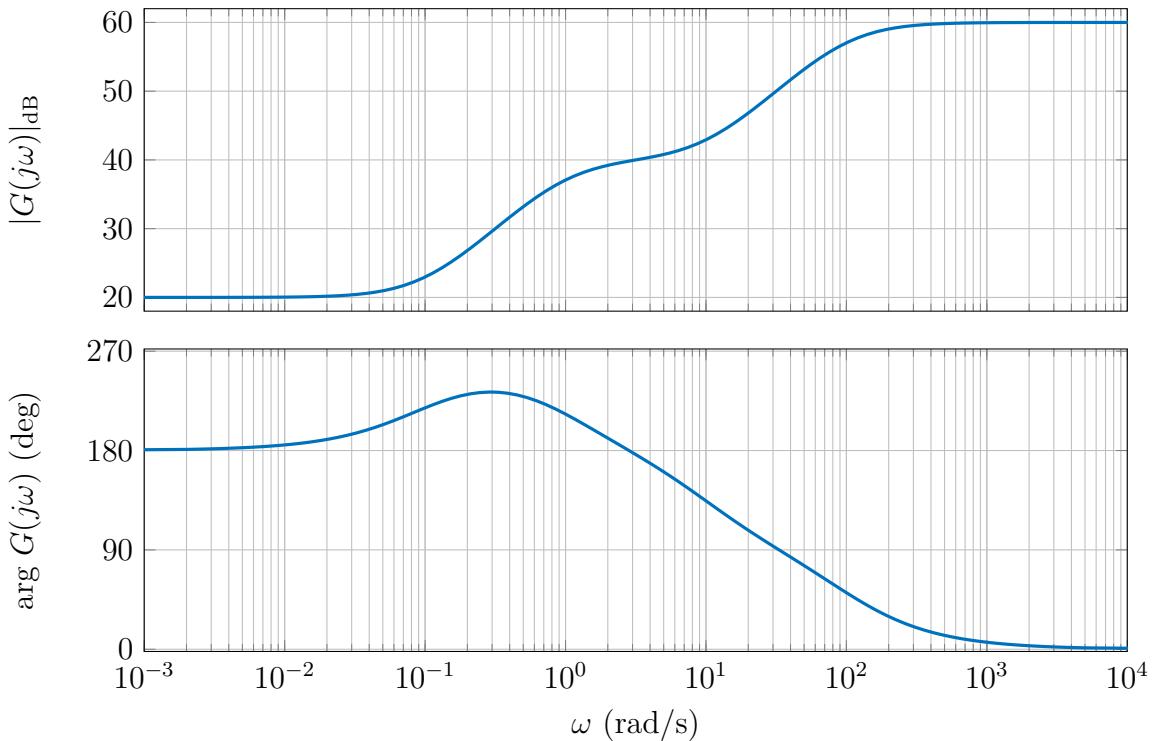
Figure A.64: Bode plot of the system in (c)

Problem 4.2 (Bode plot for time delay)

a) A pure time delay is described by $G(s) = e^{-T_d s}$. The gain and the phase are

$$|e^{-T_d j\omega}| = 0 \text{ dB} \qquad \text{and} \qquad \arg(e^{-T_d j\omega}) = -\omega T_d \frac{180°}{\pi} \approx -\omega T_d \cdot 57°,$$

respectively. The magnitude and phase over scaled frequency are shown in Figure A.65.
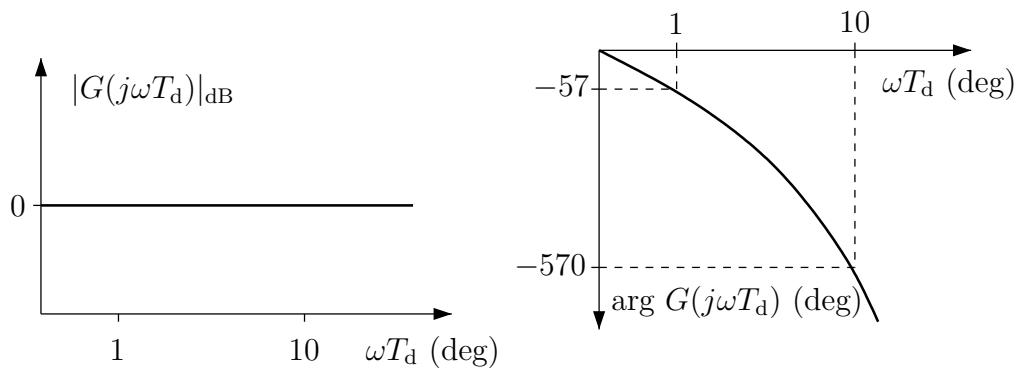
Figure A.65: Bode plot of pure time delay

The transfer function of a first order Padé approximation is

$$G(s) = \frac{1 - \frac{T_d}{2}s}{1 + \frac{T_d}{2}s}$$

This transfer function represents an all-pass filter, thus the gain is 1 for all frequencies

$$|G(j\omega)| = \left| \frac{1 - j\frac{T_d\omega}{2}}{1 + j\frac{T_d\omega}{2}} \right| = 0\text{dB}$$

The pole in the left half plane and the zero in the right half plane lead to a phase drop of $-90°$ at the corner frequency $\omega_k = 2/T_d$. Magnitude and phase are shown in Figure A.66.
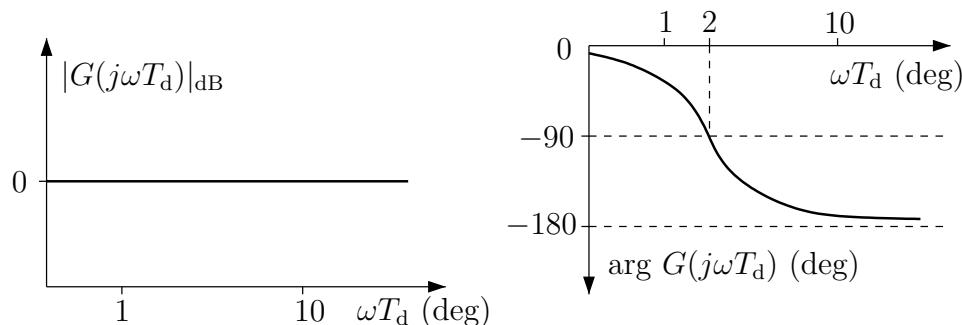
Figure A.66: Bode plot for 1st order Padé approximation of a pure time delay

b)   MATLAB solution: `Sol_Problem4_4_BodePlotTimeDelay.mlx`
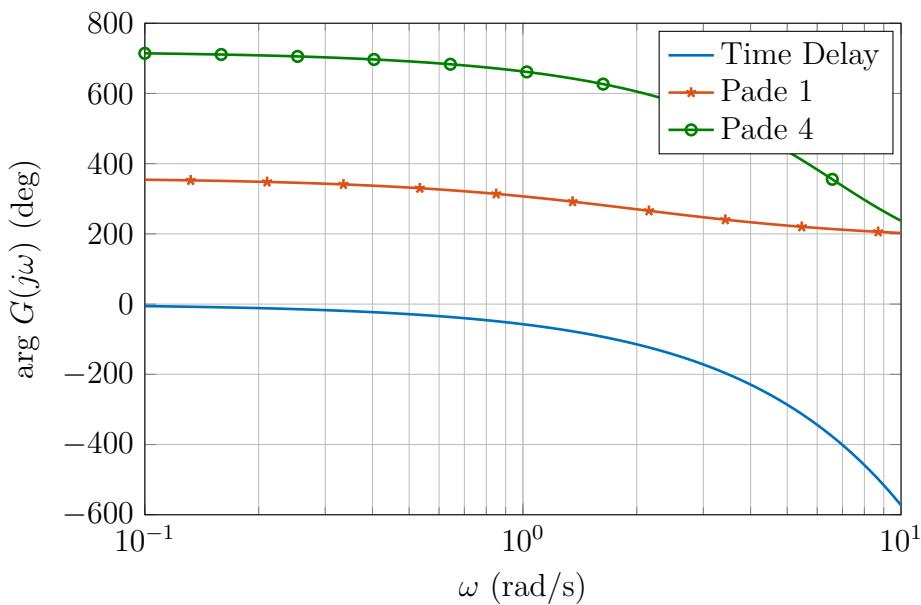
Figure A.67: Comparison of the phase plots of time delay approximations

Problem 4.3 (Transfer function from Bode plot)

The information listed in the table can be obtained from the Bode plot.

Frequency	Slope	Phase
$\omega \ll 0.1$	-20 dB/dec	$-90°$
$0.1 < \omega < 10$	0 dB/dec	$-180°$
$10 < \omega < 1000$	-20 dB/dec	$-270°$
$\omega \gg 1000$	-40 dB/dec	$-360°$

From this the following can be concluded:

- For low frequencies -20 dB/dec $\Rightarrow$ transfer function includes an integrator. The $-90°$ phase means that the gain is positive: $\frac{K}{s}$, $K > 0$;

- Increase in gain slope at $\omega = 0.1$ rad/s means a zero. The simultaneous phase drop means that the zero is in the right half plane: $\frac{1 - \frac{s}{0.1}}{1}$;

- Drop of gain slope by 20 dB/dec and phase shift of $-90°$ at $\omega = 10 \Rightarrow$ left half plane pole: $\frac{1}{1 + \frac{s}{10}}$;

- Drop of gain slope by additional 20 dB/dec and phase shift of another $-90°$ at $\omega = 1000 \Rightarrow$ left half plane pole: $\frac{1}{1 + \frac{s}{1000}}$;

Combining the above leads to the transfer function:

$$G(s) = K \frac{(1 - \frac{s}{0.1})}{s(1 + \frac{s}{10})(1 + \frac{s}{1000})}$$

where the gain K is yet to be determined. There are two ways to do this:

- Use the fact, that for low frequencies only the integrator is important, that is we have only a slope of -20 dB/dec. An integrator $1/s$ has a gain of 1 when $\omega = 1$. An integrator with gain K/s has a gain of 1 when $\omega = K$, that is in the amplitude plot it will cross the 0 dB line at frequency $\omega = K$. Thus to find the gain of $G(j\omega)$ we need to continue the integrator slope until it crosses the 0 dB line, readout the frequency $w_0 = 10^{-2}$ at which this happens an get $K = 10^{-2}$.

- Compute the gain analytically by substituting for a particular ω in $G(j\omega)$, compute the gain and compare it with the gain on the Bode plot. For example, choosing $\omega = 10^{-3}$ we have $|G(j10^{-3})| = 20dB$ and $\arg(G(j10^{-3})) = -90°$, or $G(j10^{-3}) = -j10$.

$$G(j10^{-3}) = -j10 = K \frac{1 - \frac{j10^{-3}}{0.1}}{j10^{-3}(1 + \frac{j10^{-3}}{10})(1 + \frac{j10^{-3}}{1000})} \approx K \frac{(1)}{j10^{-3}(1)(1)} = -jK10^3$$

or $K = 10^{-2}$.

Substituting this gain yields

$$G(s) = 10^{-2} \frac{(1 - \frac{s}{0.1})}{s(1 + \frac{s}{10})(1 + \frac{s}{1000})}$$

Problem 4.4 (Velocity error constant from Bode plot)

It follows from the definition of the velocity error constant (2.9) that for small ω the open-loop function can be approximated as

$$L(j\omega) \approx \frac{K_{vel}}{j\omega}.$$

The Bode plot shows that at low frequencies the open-loop system behaves like an integrator. Combining this observation with the above approximation suggests that the velocity error constant K_{vel} can be seen as the gain of a fictitious integrator. It can be read off the Bode plot at the frequency ω_c at which the straight line representing the low-frequency magnitude crosses the 0 dB line. Thus, since the phase is $-90°$ for low frequencies

$$\left| \frac{K_{vel}}{j\omega_c} \right| = 1 \quad \Rightarrow \quad K_{vel} = \omega_c = 0.01 \quad \Rightarrow \quad e_\infty = 100$$

Problem 4.5 (Analytic continuation)

a)

$$X(s) = \int_0^\infty x(t) e^{-st} dt = \int_0^\infty K e^{at} \sigma(t) e^{-st} dt$$

where, $\sigma(t)$ is unit step, then,

$$X(s) = K \int_0^\infty e^{(a-s)t} dt$$

whose domain of convergence includes the part of s-plane for which $Re(s) > a$. This region will be on the right of a line passing through the point $s = a$ as shown by the shaded region in Figure A.68.
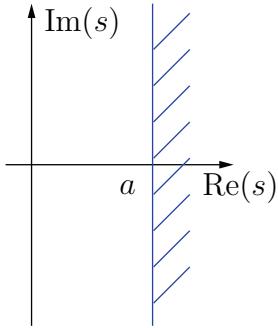
Since, $a > 0$ this region does not include the imaginary axis.

b)   Taylor series expansion of X(s) about the point $s = s_0$ is given by,

$$X(s) = X(s_0) + (s_0 - s)X'(s_0) + (s_0 - s)^2 \frac{X''(s_0)}{2!} + (s_0 - s)^3 \frac{X'''(s_0)}{3!} + \cdots$$

where,

$$X'(s_0) := \left. \frac{dX(s)}{ds} \right|_{s=s_0} = \frac{-1}{(s_0 - a)^2} = \frac{-1}{(a - s_0)^2}$$

$$X''(s_0) := \left. \frac{d^2 X(s)}{ds^2} \right|_{s=s_0} = \frac{2}{(s_0 - a)^3} = \frac{-2}{(a - s_0)^3}$$

$$X'''(s_0) := \left. \frac{d^3 X(s)}{ds^3} \right|_{s=s_0} = \frac{-6}{(s_0 - a)^4} = \frac{-3!}{(a - s_0)^4}$$

Figure A.68: Domain of definition of $x(t)$

The series expansion of $X(s)$ can thus be written as,

$$
\begin{aligned}
X(s) &= -\left(\frac{1}{a-s_0} + \frac{s-s_0}{(a-s_0)^2} + \frac{(s-s_0)^2}{(a-s_0)^3} + \cdots\right) \\
&= -\sum_{n=0}^{\infty} \frac{(s-s_0)^n}{(a-s_0)^{n+1}}
\end{aligned}
$$

c)   If $s = a$, then

$$
X(s) = -\sum_{n=0}^{\infty} \frac{1}{(a-s_0)} = \infty
$$
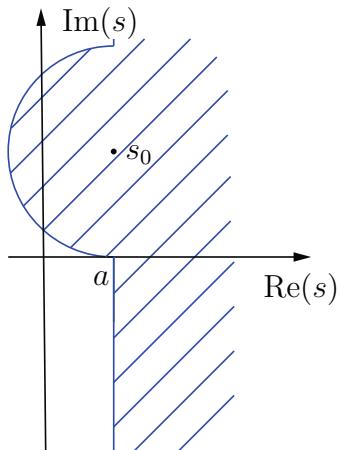
If $s \neq a$, then to prove the convergence of the series we have to use the ratio test. Thus, the series is convergent if

$$
\left|\frac{(s-s_0)^{n+1}}{(a-s_0)^{n+2}} \middle/ \frac{(s-s_0)^n}{(a-s_0)^{n+1}}\right| < 1
$$

$$
\left|\frac{s-s_0}{a-s_0}\right| < 1
$$

$$
|s-s_0| < |a-s_0|
$$

Thus, the region of convergence is extended to a disk centered at s_0, but does not touch, the pole $s = a$ as shown in Figure A.69.

The approach of analytic continuation works by first expanding the function of $s \in C$ about all points in its domain of definition and then extending the domain of definition to all the points for which this series expansion converges.

Above we have shown that the series expansion of an unstable transfer function remains convergent in a disk centered at s_0 and extended up to $s = a$. Using the analytic continuation technique we can extend the domain of definition of an unstable transfer function to the region which is the union of all such disks. It can be seen that this extended region includes all the points in s-plane except the point $s = a$. Hence, the imaginary axis is included in the domain of definition of an unstable transfer function.

Figure A.69: Extended Domain of definition of $x(t)$

Problem 4.6 (Nyquist criterion)

a) The open loop transfer function is

$$L(j\omega) = 10 \, \frac{1}{j\omega \left(1 + j\frac{\omega}{1}\right)\left(1 + j\frac{\omega}{100}\right)}$$

The Bode plot of this transfer function is constructed by superimposing the Bode plots of all factors of the denominator as well as that of the constant gain 10. See Figure A.70.

b) The Nyquist plot can be sketched by determining magnitude and phase at selected frequencies from the Bode plot:

$$
\begin{array}{llll}
\omega \to 0: & |L(j\omega)| \to \infty & \arg(L(j\omega)) = -90° \\
\omega_c: & |L(j\omega)| = 1 & -90° < \arg(L(j\omega_c)) < -180° \\
\omega_B: & |L(j\omega)| \approx -20\text{dB} = 0.1 & \arg(L(j\omega_B)) = -180° \\
\omega \to \infty: & |L(j\omega)| \to 0 & \arg(L(j\omega)) = -270°
\end{array}
$$

The Nyquist diagram generated by MATLAB is shown in Figure A.71. Note that the Nyquist curve intersects the real axis at about -0.1 and approaches the origin at an angle of $-270° = +90°$ (to see this in the Matlab plot one has to zoom in at the origin)

To check whether the infinite arc closes the Nyquist plot on the left or the right hand side, modify the Nyquist path to take a small detour to the right to avoid the open-loop pole at the origin. Checking the phase along the small semicircle reveals that the Nyquist diagram is closed at $+\infty$ (to the right). Thus the critical point -1 is not encircled, i.e. $N = 0$. Since the open loop transfer function does not have any poles in the right half plane, we have $P = 0$. With $Z = P + N = 0$ it
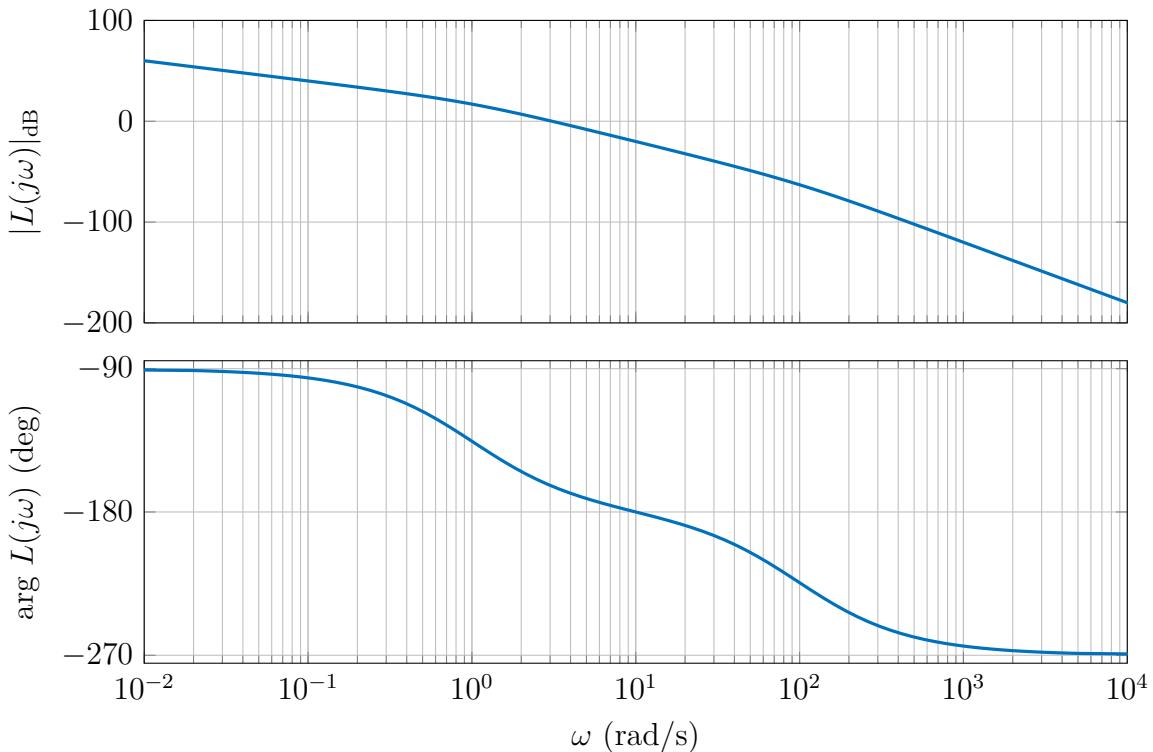
Figure A.70: Bode plot
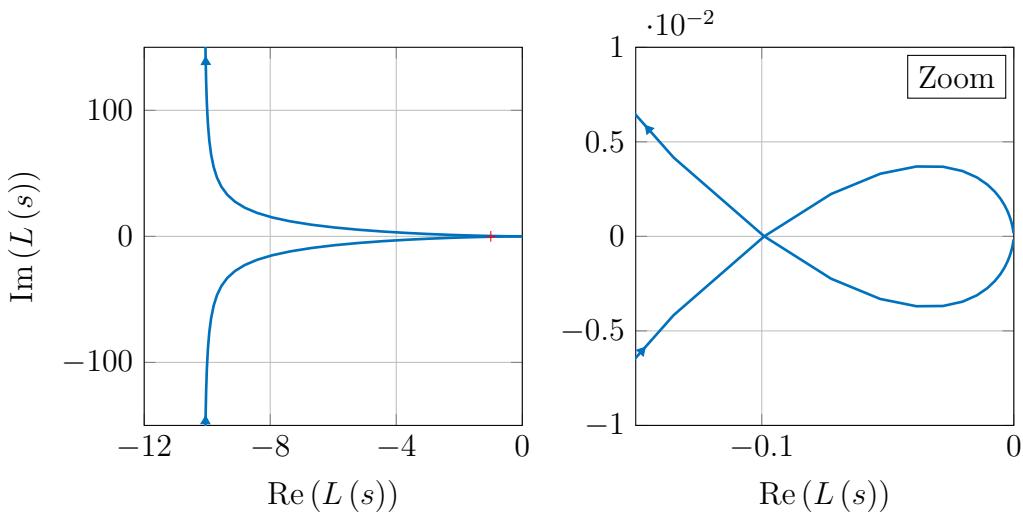
Figure A.71: Nyquist plot

follows that the closed loop system with $K_P = 1$ is stable. It will become unstable as soon as the gain is increased to the point where the intersection with the real axis is moved from $s = -0.1$ to the critical point $s = -1$. This requires a gain of $K_P = 1/0.1 = 10$; thus the closed loop is stable for $0 < K_P < 10$.

MATLAB file: `Sol_Problem4_6_NyquistCriterion.mlx`

Problem 4.7 (Nyquist criterion for systems with time delay)

MATLAB file: `Sol_Problem4_7_NyquistCriterionTdelay.mlx`

If the system has a time delay of $T_d = 1$, an additional phase lag of

$$\arg(e^{-T_d j\omega}) = \arg(e^{-j\omega}) = -\omega\frac{180°}{\pi} \approx -\omega \cdot 57°$$

is introduced. For selected frequencies one can calculate

$$\phi_{Tt}(0.1) = -5.7° \ ; \qquad \phi_{Tt}(1) = -57° \ ; \qquad \phi_{Tt}(10) = -570° \ ;$$

The phase margin without time delay, determined at $w_b \approx 3$, is approximately 16°. Because of the time delay the phase at this frequency is reduced by $\phi_{tt} = 3 \cdot 57° = 171°$. This means that the closed loop system with $K_P = 1$ becomes unstable. Figure A.72 and Figure A.73 show the Bode and Nyquist plots, respectively, generated with MATLAB.
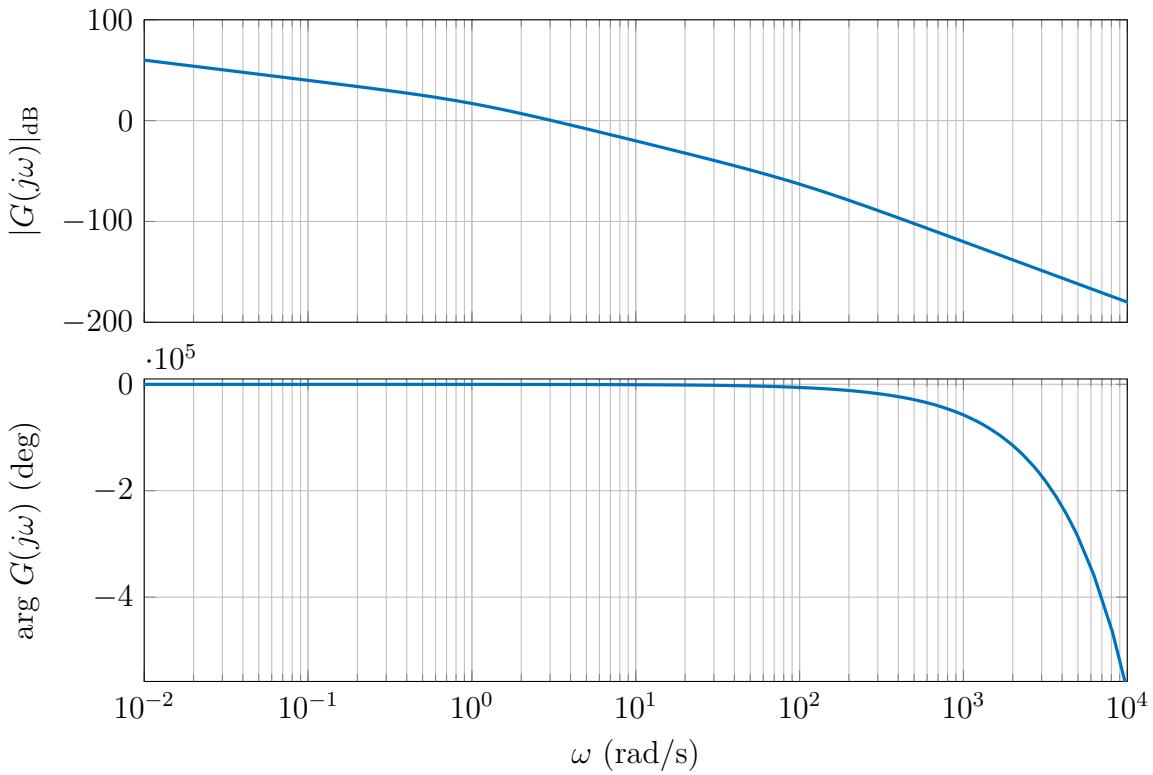
Figure A.72: Bode plot
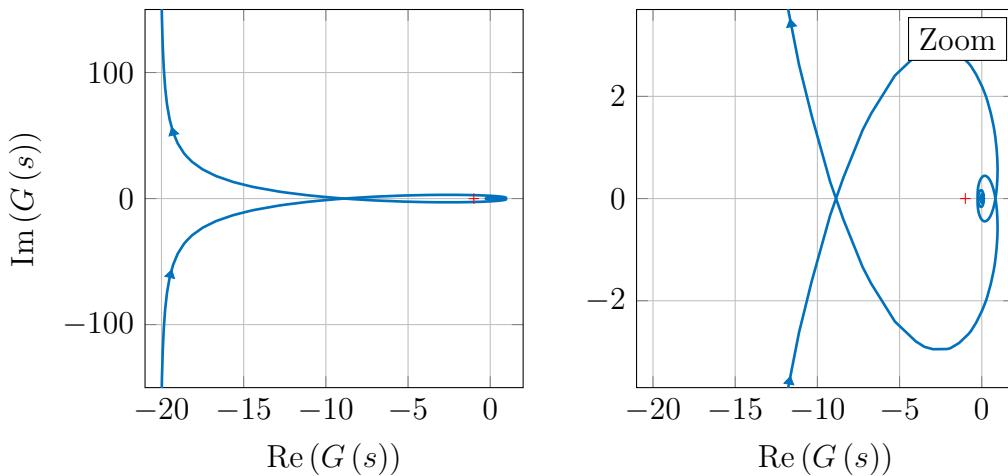
Figure A.73: Nyquist plot

Problem 4.8 (Conditionally stable system)

The open-loop Bode plot of the system is shown in Figure A.74. Trying to obtain the stability margins from the Bode diagram (for $K = 1$) yields a phase margin of 44°, which indicates a stable closed-loop system, whereas the gain margin of -12 dB seems to suggest that the system is unstable. This example illustrates the caveat that was given when stability margins in a Bode diagram were discussed: the rules for inferring closed-loop stability from the Bode diagram do not always apply.

To investigate the stability of the loop for the system given here, one should examine the Nyquist plot, which is shown in Figure A.75 (i) and Figure A.76. This plot displays one counterclockwise encirclement of -1. Checking the phase angle along a small semi-circle around the origin reveals that there is also one clockwise encirclement by an infinite arc; thus there is no net encirclement and the feedback loop is stable when the gain is 1. However, it is clear that making the gain smaller than 0.25 will result in two clockwise encirclements, indicating two unstable closed loop poles. A feedback system that looses stability when the gain is *reduced* is called *conditionally stable*.

Information about closed-loop stability can also be obtained from the root locus plot shown in Figure A.75 (ii): two root locus branches move from the left half plane into the right half plane when the gain becomes smaller than 0.25.

MATLAB file: `Sol_Problem4_8_BodeNyquistInfGainMargin.mlx`
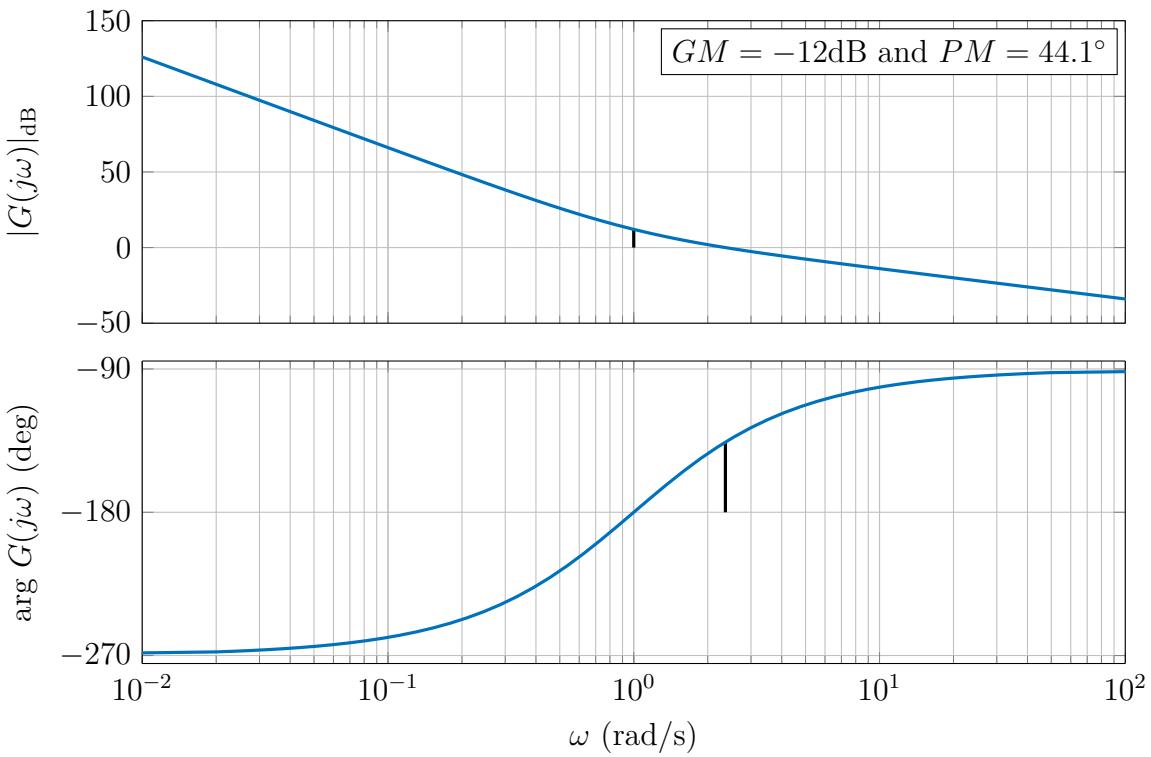
Figure A.74: Bode plot
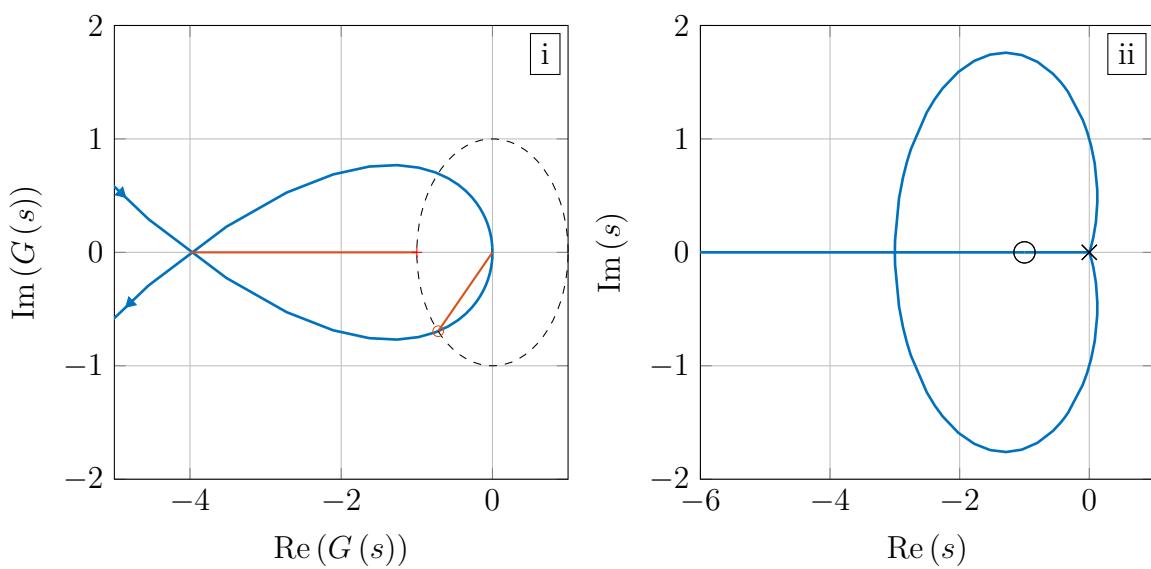
Figure A.75: Nyquist plot (i) and root locus plot (ii)
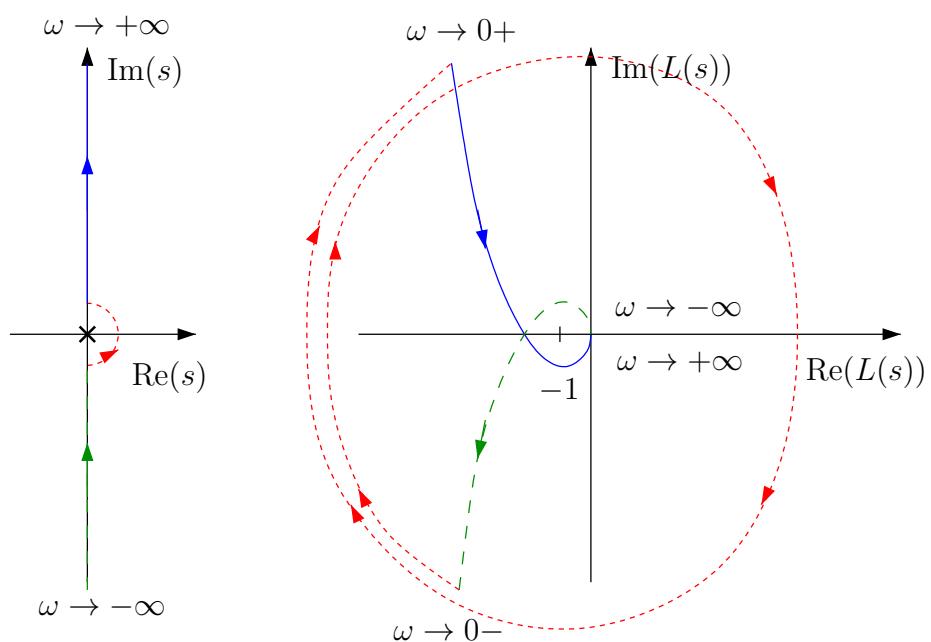
Figure A.76: Nyquist path

Problem 4.9 (Lead compensator)

MATLAB file: `Sol_Problem4_9_LeadComp.mlx`

It is clear from Figures 4.34 and 4.35 that the closed-loop bandwidth ω_b should be in general close to the open-loop crossover frequency ω_c to have a desirable closed-loop response. Using the procedure given in the lecture notes, the controller `C(s)` can be calculated as follows:

1) Determine the gain at the desired bandwidth $|G(\omega_b \approx \omega_c = 5)| = 0.34$ from which $K' = \frac{1}{|G|} = 2.9155$ can be calculated. An increase of magnitude to 0 dB at this frequency will make the closed-loop bandwidth close to 5.

2) Determine $\omega_m = \omega_c$ and $\phi_m = PM - 180 - \phi(\omega_m)$. The phase angle at this frequency is $-149.04°$. To achive PM $= 70°$ the phase has to be raised by $\phi_m = 39.04°$.

3) The controller parameters are therefore

$$\alpha = \frac{1 - \sin \phi_m}{1 + \sin \phi_m} = 0.2271,$$

$$\omega_L = \sqrt{\alpha} \cdot \omega_c = 2.3829.$$

and

$$K = \sqrt{\alpha} \cdot K' = 1.3895$$

The designed controller has this form:

$$C(s) = 6.12 \frac{s + 2.383}{s + 10.49}.$$

In Figure A.77 the different bode plots for each step are shown.

Figure A.78 shows the Matlab `sisotool` view of this controller with three windows showing the root locus, the open loop frequency response and the step-response of the closed-loop system, respectively. One can see that the phase margin requirement and the obtained crossover frequency are satisfied.

To check the condition that the control signal should have a magnitude of less than 5 in response to a unit step reference, use since Matlab 2017a *New Plot – New Step – IOTransfer_r2u*. With the controller above the control input will not exceed 6. One can now try to manually adjust the controller to meets the requirement $|u| < 5$.

By adjusting the zero, pole and gain of the controller (this can be done directly in the Bode diagram or Root Locus with the left mouse button) one can try to find a controller that satisfies all requirements or a good comprise. For example the controller

$$C(s) = 1.12 \frac{1 + 0.46s}{1 + 0.1s},$$
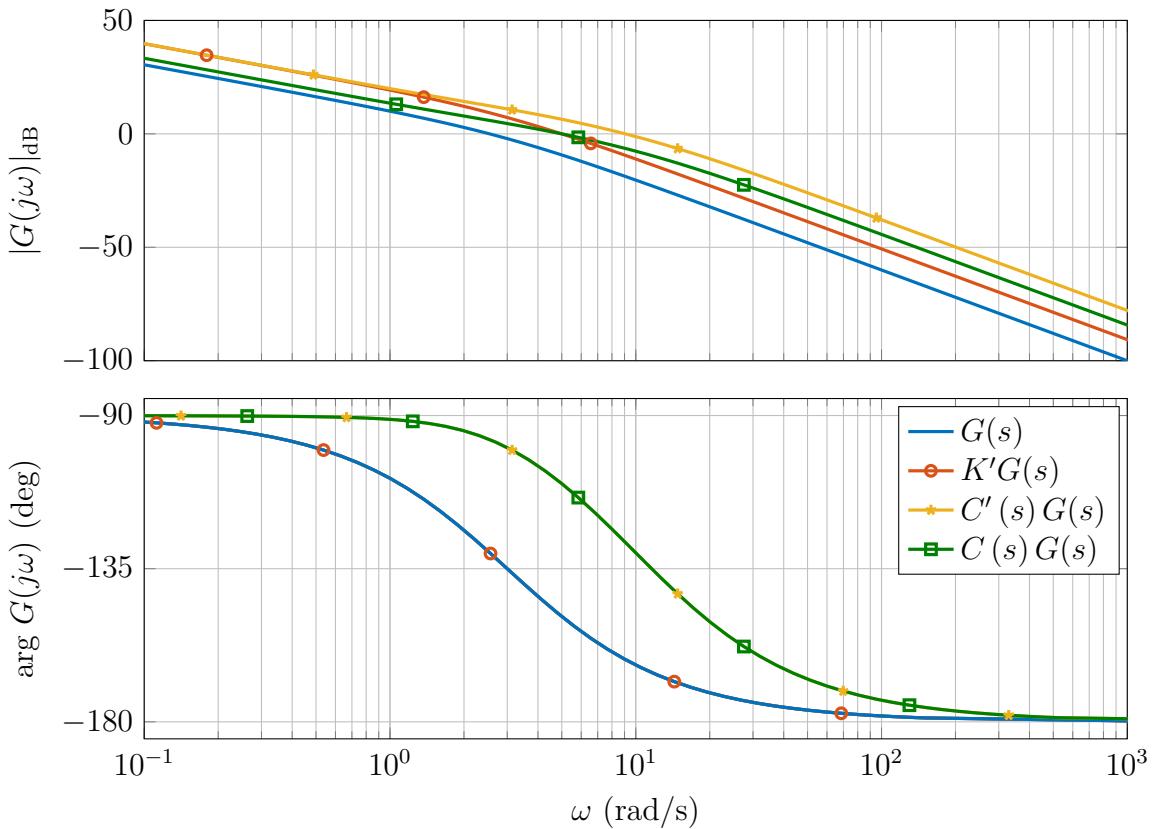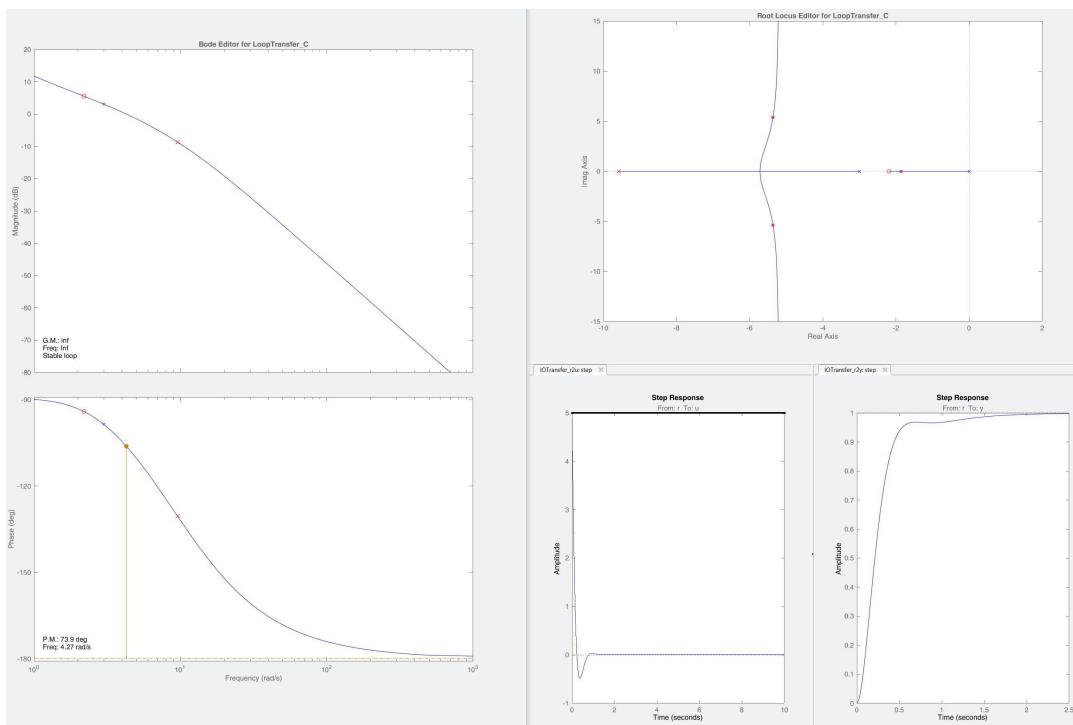
Figure A.77: Bode plot of $G(s)$, $K'G(s)$, $C'(s)G(s)$ and $C(s)G(s)$

Figure A.78: SISOTool windows with plant and lead compensator

leads to the Bode plot and root locus shown in Figure A.78. Note that at *Edit – SISOTool Preferences* under *Options* one can choose between different representations of transfer functions.

Problem 4.10 (Experimental generation of the frequency response)

a) 1) The MATLAB command 'bode' can be used to create the Bode plot shown in Figure A.79.
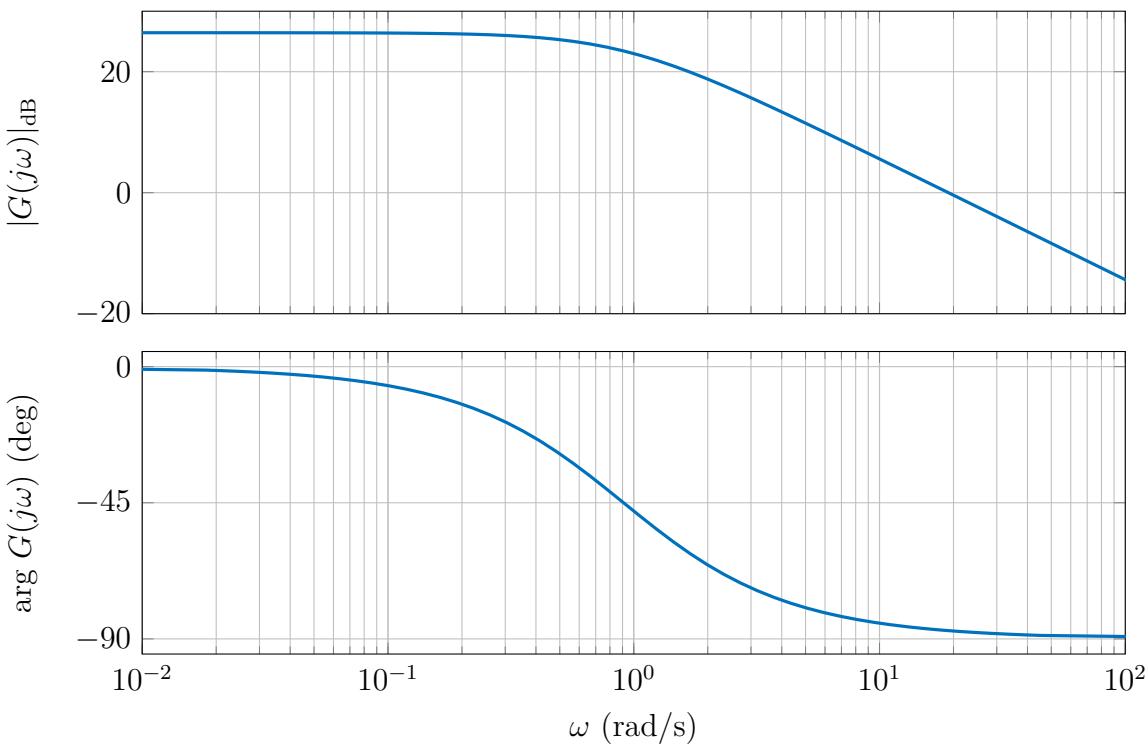
Figure A.79: Bode plot of the motor model

2) The corner frequency for this transfer function is obtained from $|1.1j\omega + 1| = \sqrt{2}$, thus $\omega = 0.9$. This can also be read off the Bode plot, which shows at $\omega = 0.9$ a phase drop of $-45°$.

b) The use of "Problem4_10_PlotSinusComparison.m" is illustrated for $\omega = 1$ in Figure A.80. The left plot shows the determination of the magnitude. The amplitude of the input signal is 2 and that of the output is approximately 24, indicating a magnitude of the motor transfer function at this frequency of $\frac{24}{2} = 12 \approx 21.6$ dB. The right plot illustrates how the phase is obtained, which is $\approx -55°$. This needs to be repeated for all frequencies.

c) i) Create vectors that contain the tested frequencies and the corresponding magnitudes and phases. With "Problem4_10_BodeComparison.m" the experimental Bode plot can be compared to that of the model; this is shown in Figure A.81
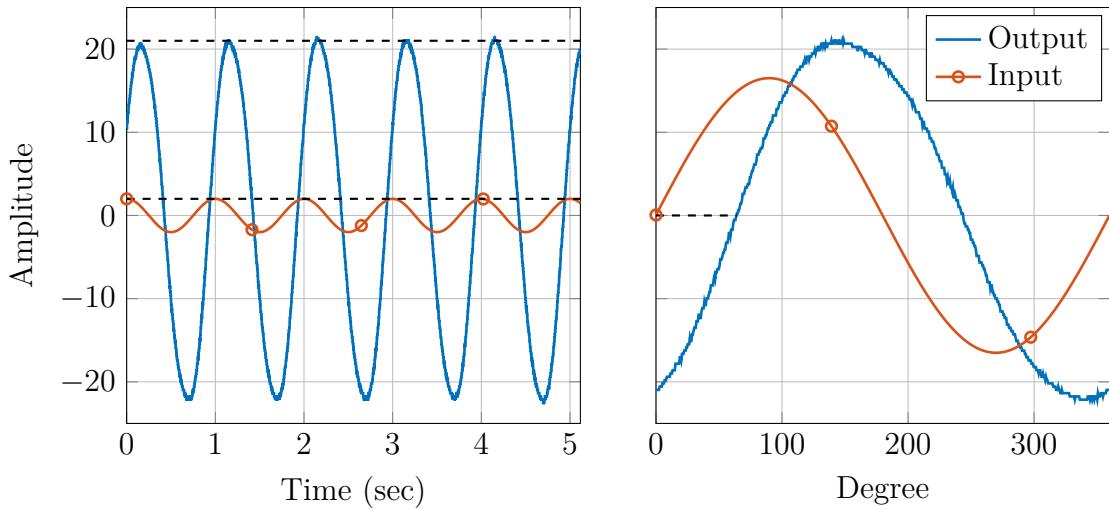
Figure A.80: Magnitude detection (left) and phase detection (right) of the motor transfer function at $\omega = 1$
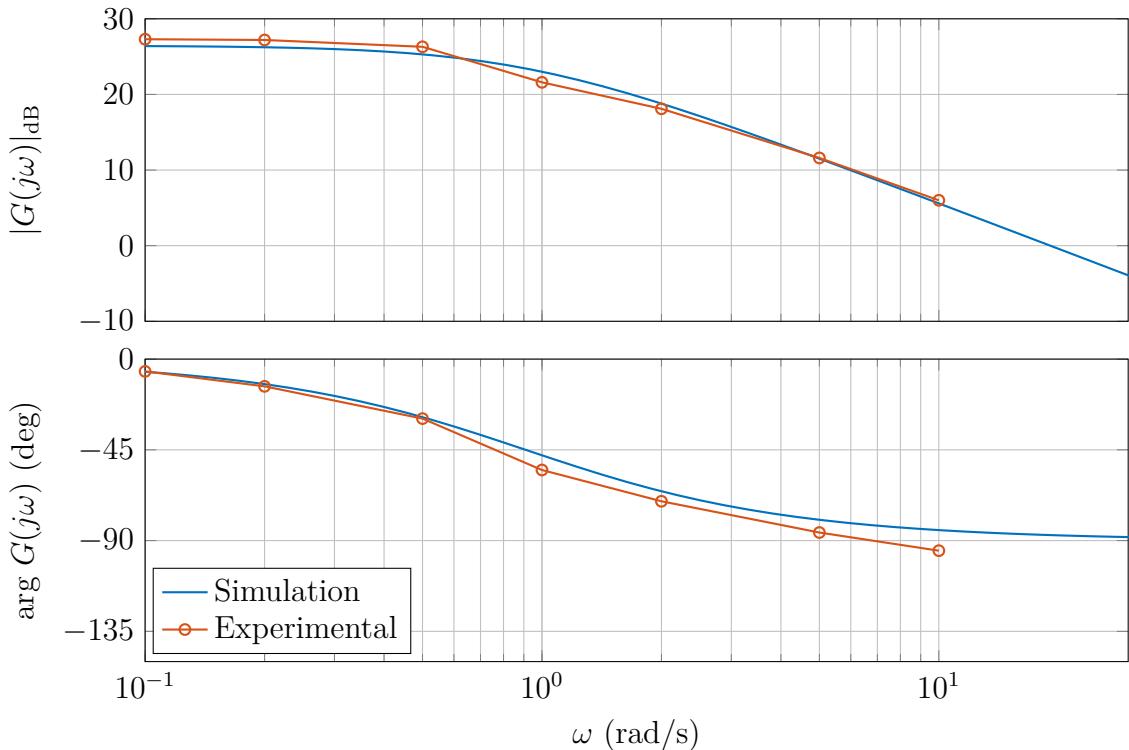
Figure A.81: Comparison of model and experimental Bode plot

ii)   From Figure A.81 it can be seen that the corner frequencies do not match. The experimental frequency response has a corner frequency of ≈ 0.76 rad/s. Adjusting the model such that it has the same corner frequency leads to

$$G(s) = \frac{21}{1.32 \cdot s + 1}.$$

Figure A.82 shows a good match between the experimental and the adjusted model frequency response.
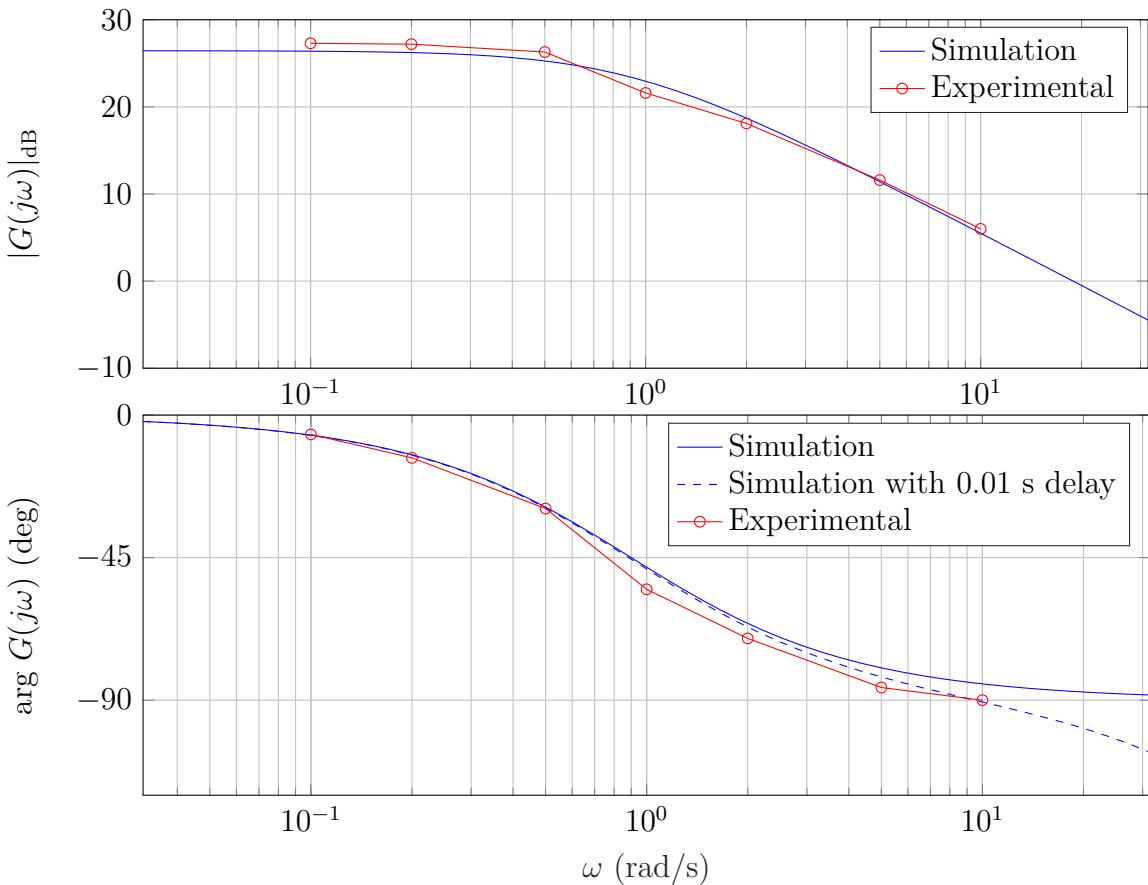
Figure A.82: Comparison of adjusted model and experimental Bode plot

iii)   The motor model in (1.34), which was confirmed in Problem 1.6.b) under the assumption $K = K_g = K_m$ is

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

In Figure A.81 we see that the static gains in both Bode plots match well, whereas the corner frequencies do not. The model transfer function has a static gain of $K_0 = \frac{1}{K}$, thus K should not be changed. The model mismatch is apparently due to inaccurate values of J and/or R; this can have various reasons.

d)   Comparing the simulation and experimental results, it can be seen that the magnitude plot and phase plot already are a good match. By introducing a time delay of 10 ms to the simulation, the magnitude plot does not change, but the phase plot matches better with the experimental results. Keep in mind that the time delay

needed depends on the experimental data. Every motor is slightly different and so are the operating conditions.

SIMULINK file: `Model_Problem4_10_BodeExp.slx`

Problem 4.11 (Lead-lag compensator for shaft control of a DC motor)

MATLAB file: `Sol_Problem4_11_LeadLagComp.mlx`

a) i) According to the rule of thumb, one obtains

$$t_r \approx \frac{1.7}{\omega_n} \approx \frac{1.7}{\omega_b} \quad \Rightarrow \quad \omega_b \approx 3.78.$$

In order to design a lead compensator three parameters have to be determined: the gain K, α and ω_L.

* The gain K' is chosen such that the required bandwidth is met. The Bode plot shows that the magnitude at a frequency of $\omega_c = 3.78$ is approximately $2.3\,\text{dB} \approx 1.3$. Alternatively, one gets exact result by typing the command

$$\texttt{[mag,phase]=bode(G,3.78)}.$$

Hence, the gain K' is taken as $K' = \frac{1}{|G(j3.78)|} \approx \frac{1}{1.3} \approx 0.77$.

* Here ω_m is $\omega_m = \omega_c = 3.78$.

* Now ϕ_m has to be determined such that a phase margin of 50° is obtained. From the Bode plot one finds that the phase is $\phi(\omega_c) \approx -166.47°$, which means that the current phase margin is 13.53°. This results in $\phi_m = 50° - 13.53° = 36.47°$. Using these values yields

$$\alpha = \frac{1 - \sin(\phi_m)}{1 + \sin(\phi_m)} \approx 0.2544$$

$$\omega_L = \sqrt{\alpha} \cdot \omega_m \approx 1.9066$$

$$\text{and} \quad K = \sqrt{\alpha} \cdot K' \approx 0.388.$$

* The designed lead compensator is then

$$C_{lead}(s) = K \frac{1 + \frac{s}{\omega_L}}{1 + \frac{\alpha \cdot s}{\omega_L}} = 0.388 \frac{1 + 0.524s}{1 + 0.133s}. \tag{A.1}$$

With a phase margin PM = 50°, a damping of $\zeta \approx \frac{\text{PM}}{100} = \frac{1}{2}$ is obtained. This leads to an expected overshoot of $M_p = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \approx 0.16 = 16\,\%$.

Now call '`sisotool(G,Clead)`', where G is the transfer function $G(s) = \frac{21}{1.1 \cdot s^2 + s}$ and *Clead* is the lead compensator of equation (A.1). Figure A.83 shows the resulting window. Right click the closed loop step response and choose *Characteristics → Rise Time*. The rise time of $0.317\,s$ and the phase margin of 50.1° which is shown in the phase plot are as requested in the task.
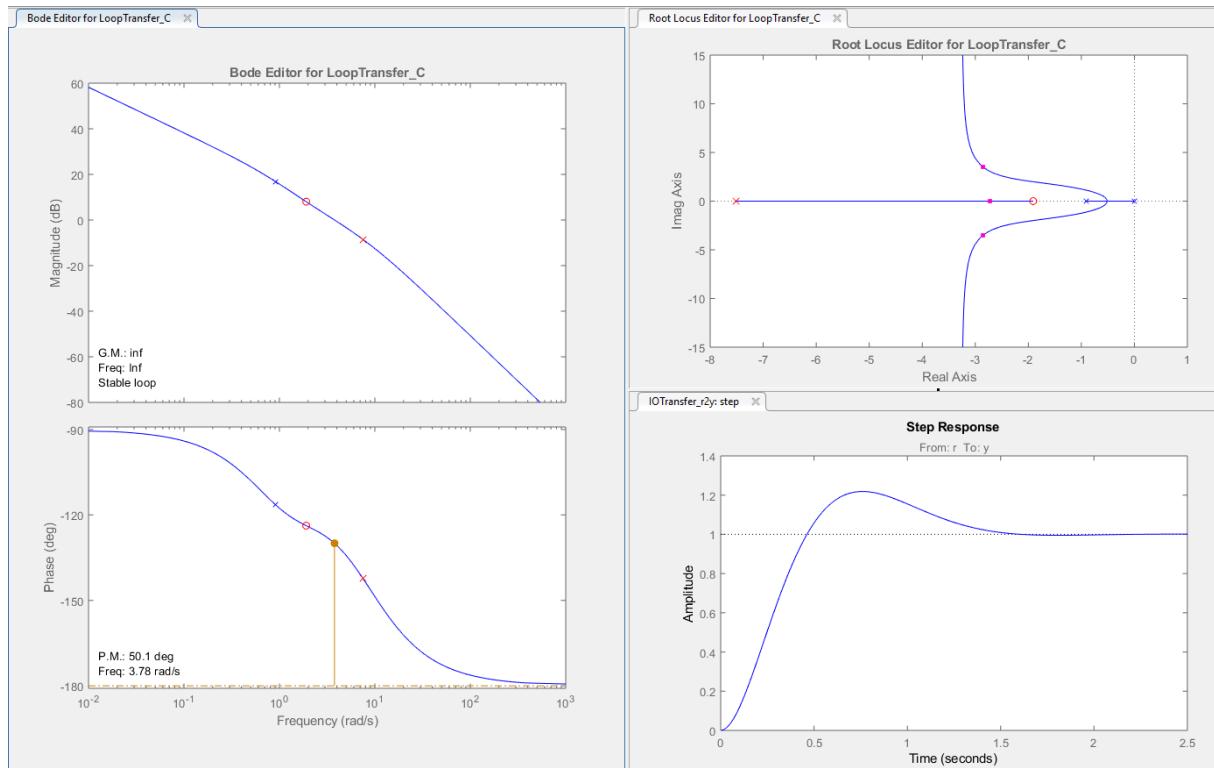
Figure A.83: MATLAB sisotool with plant and designed lead compensator

ii)   An experimental step response with the values $K = 0.388$, $T_L = \frac{1}{\omega_L} = 0.524$ and $\alpha = 0.2544$ for the lead compensator is shown in Figure A.84. The expected overshoot of about 16 % is confirmed.
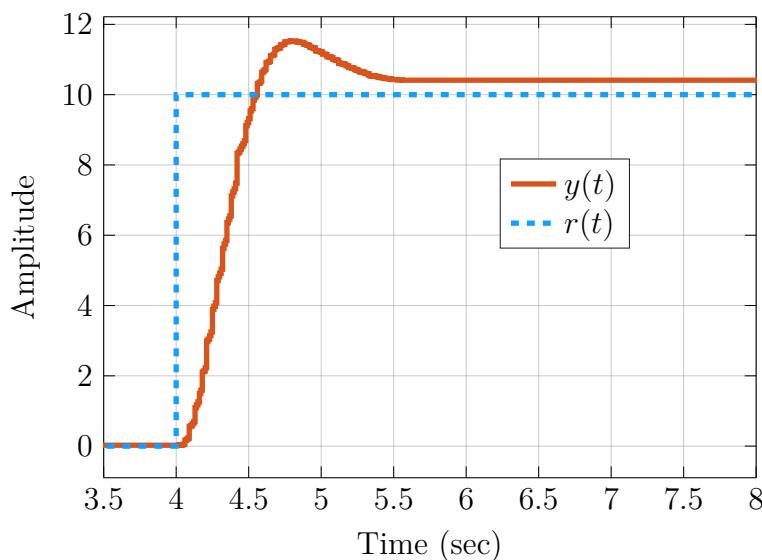
Figure A.84: Angle control, experimental closed-loop step response with lead compensator

iii)   The tracking steady state error should be zero as in the step response of Figure A.83, since there is integral action in the plant. However, one can observe a slight steady state error of about 4% in Figure A.84.

If a step disturbance is added at the plant input, there will be a steady state error, because there is no integral action in the controller but only in the plant. A typical input disturbance is actuator friction, which for the DC motor control problem can be treated similar to the external load torque T_l in model (1.34).

iv)   Figure A.85 shows the response to a input step disturbance while using the lead compensator designed in (a)(i).
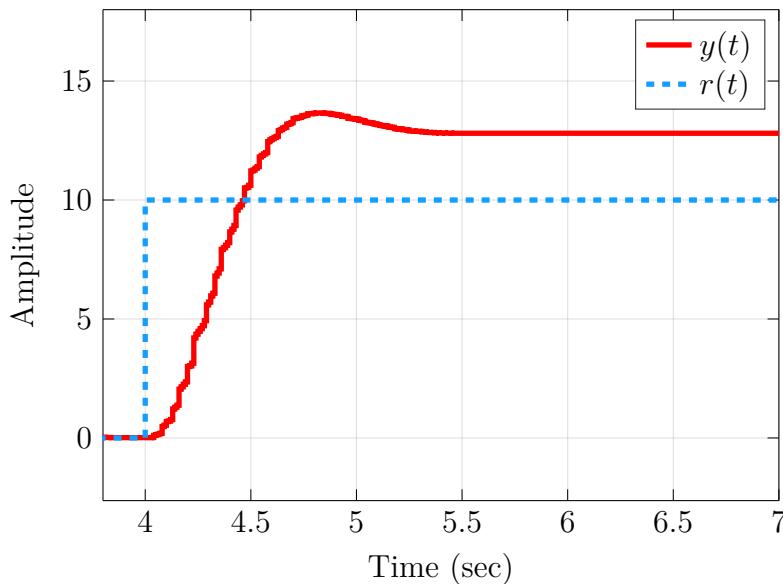
Figure A.85: Angle control, experimental closed-loop response to input step disturbance with lead compensator

As one can see there is a significant steady state error.

b)   i)   The transfer function of a lag compensator (4.14) is the same as that of a lead compensator, except that we have $\alpha > 1$. Using the command 'sisotool(G,C)' a lag compensator can be designed. The static gain of the lag compensator has to be 10, because $e_\infty = \lim\limits_{s \to 0} \frac{R - D_u G}{1 + C_{lag} C_{lead} G}$. The right corner frequency of the lag compensator is chosen approximately one decade below the left corner frequency of the lead compensator in order to not affect the previously tuned phase margin.

The left corner frequency of the lag compensator is then chosen such that the demanded change in magnitude is realized. In this case, the gain at low frequencies shall be increased by 20dB. Hence, the left corner frequency has to be placed one decade below the right corner frequency, resulting in the following lag compensator:

$$C_{lag}(s) = 10\frac{1 + 5.3s}{1 + 53s}. \tag{A.2}$$

Even though the phase margin should not have been changed by the lag compensator, it is reduced to 47.4°. While this will result in a slightly larger overshoot, a similar tracking behavior can be expected.

To add a lag compensator to the controller that has already been designed using the SISOTOOL, right-click a plot, choose *Edit Compensator...* right click the *Dynamics* field and select *Add Pole/Zero* and *Lag* (see Figure A.86).

To display the effect of an input disturbance step, right-click $IOTransfer\_du2y$ in the *Responses* window and select $Plot \rightarrow step$. Figure A.87 shows that the effect of a step disturbance is reduced by a factor of 10 by using the lag compensator A.2.
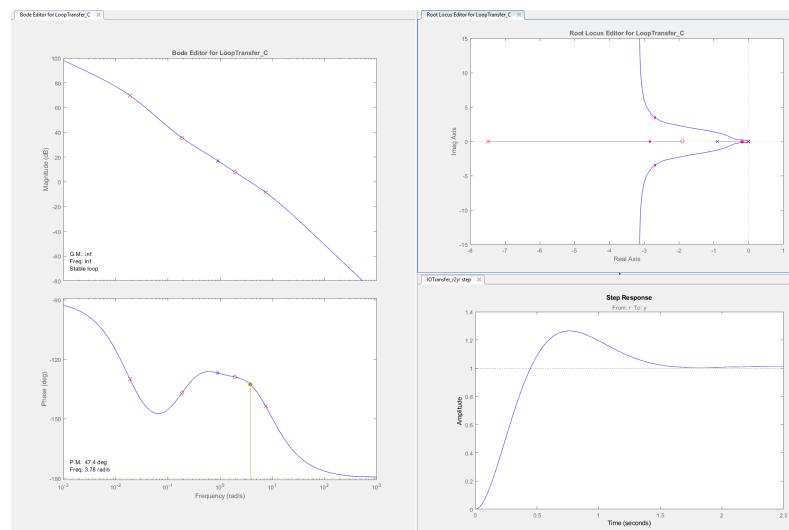
Figure A.86: Lead-lag compensator design (sisotool)

(a) Lead compensator

(b) Lead-lag compensator
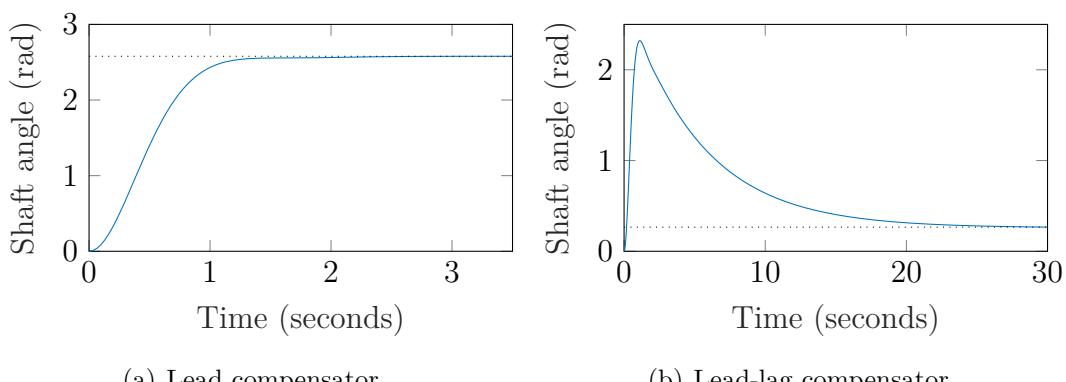
Figure A.87: Input step disturbance response with lead (left) and lead-lag (right) compensator (SISOTOOL)

ii)   By using the given SIMULINK model with the values $K_{lag} = 10$, $T_{L,lag} = 5.3$ and $\alpha_{lag} = 10$ of the lag compensator (A.2) along with the previously designed lead compensator, the effect of the resulting lead-lag compensator can be verified. It can be observed that the steady state performance of the lead-lag compensator with an additional input disturbance (Figure A.88) is better than the lead compensator performance (Figure A.85).
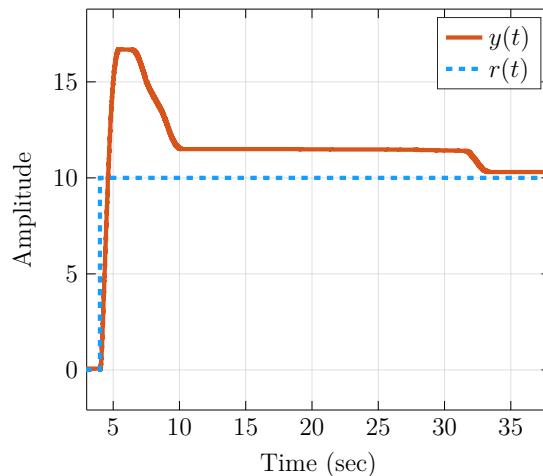
Figure A.88: Experimental angle control with additional input disturbance

Figure A.89 shows the lead-lag compensator performance without an additional input disturbance.
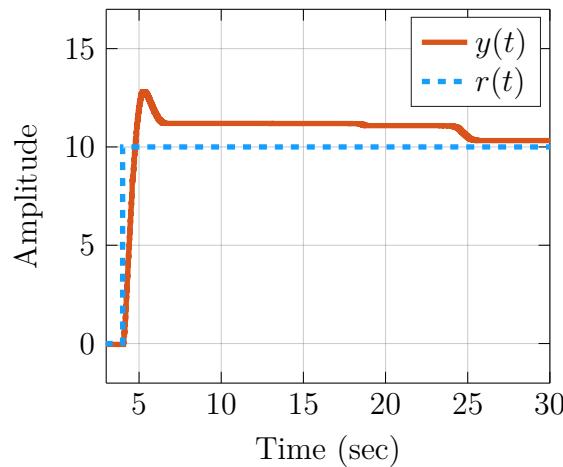
Figure A.89: Experimental angle control

The result depends on the motor and operating conditions, moreover friction is highly nonlinear. It has been assumed, however, that friction is just a constant input disturbance.

iii) If the pole-zero pair of the lag compensator is shifted to the right, the disturbance rejection will be faster. The larger the distance between the pole and the zero is, the smaller will be the steady state error. A possible fast lag controller is:

$$C_{lag}(s) = \frac{s+1}{s+0.01}. \tag{A.3}$$

Because the right corner frequency of this lag compensator is so close to the left corner frequency of the lead compensator, the phase margin is reduced to 35.4°. This results in a greater overshoot. The speed and steady state error are improved, however (see Figure A.90).
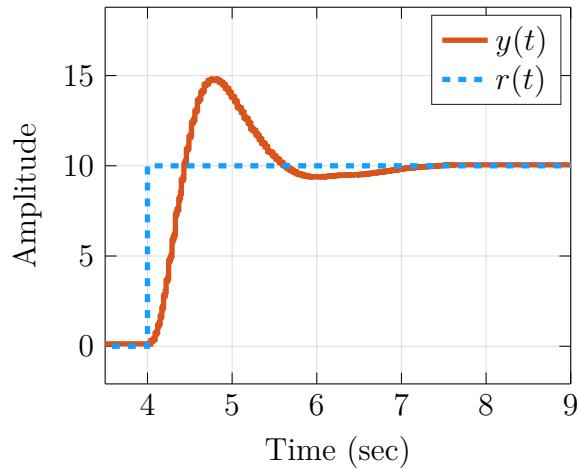
Figure A.90: Experimental angle control without additional input disturbance and faster lag compensator

Problem 4.12 (Realisable lead compensator)

a) The Bode and Nyquist diagrams of $C_0(s)$ are sketched in Figure A.91. For $\omega \to \infty$ the system gain is $K_p \frac{0.2}{0.05} = 4K_p$. This controller is a lead compensator.

b) The extra pole has to be added at high frequency, so that it does not destroy the phase increased achieved by the lead compensator. At the same time such high frequency pole will reduce the sensitivity to noises, that are typically present at high frequencies. Thus one could chose T_f such that the pole is several decades after the pole of the lead compensator, e.g. $T_f = 10^{-4}$. Sketches of the Bode and Nyquist diagrams are shown correspondingly in Figure A.92 and Figure A.93.
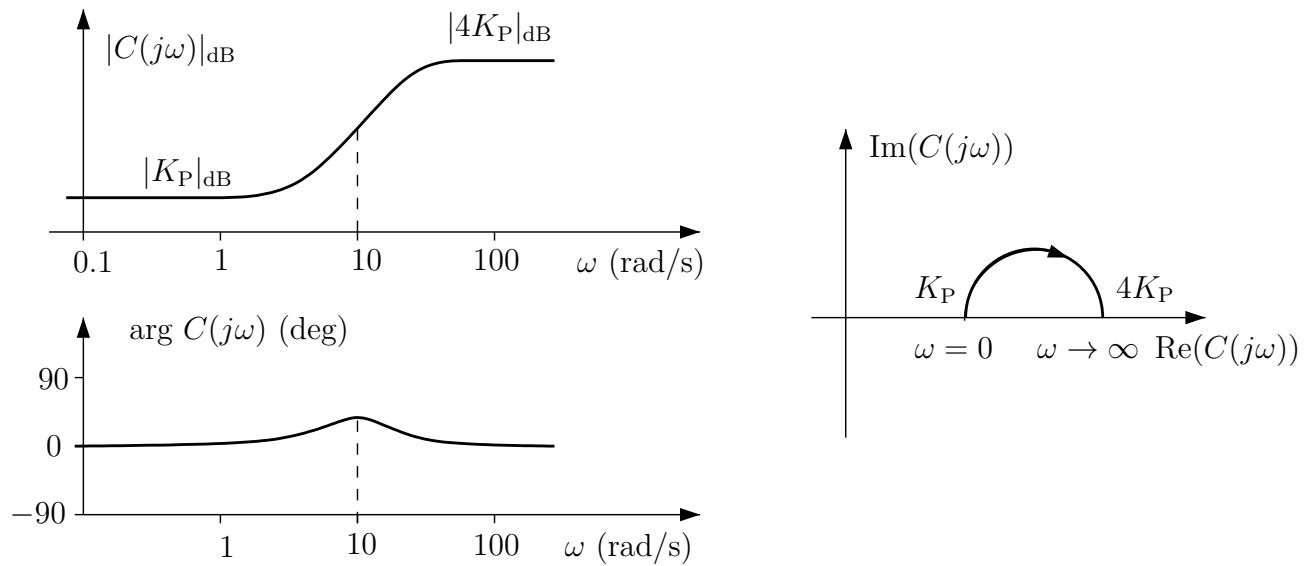
Figure A.91: Bode and Nyquist plot of a lead compensator
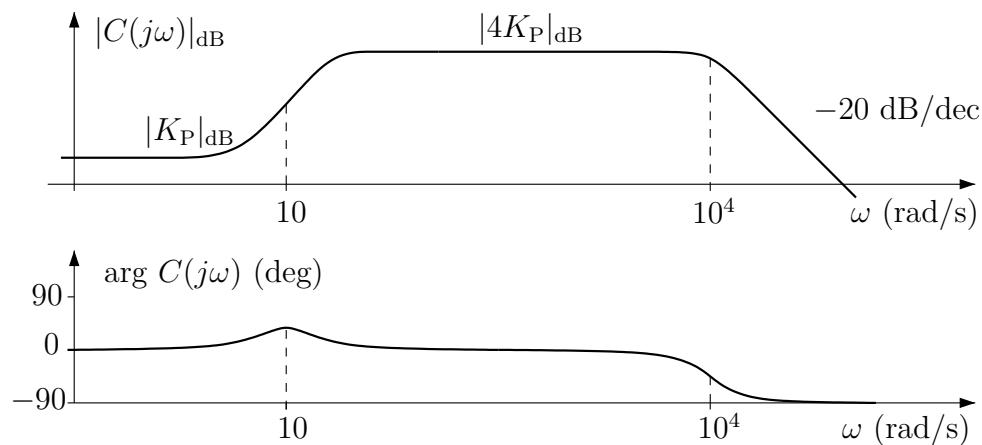
Figure A.92: Bode plot of a lead compensator with roll-off
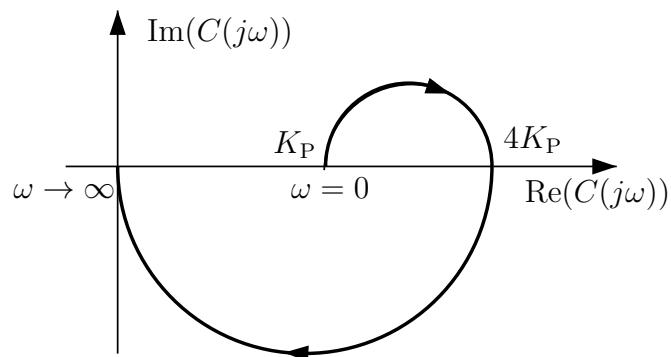
Figure A.93: Nyquist diagram of a lead compensator with roll-off

Problem 4.13 (Design of a PID controller for a DC motor using the Bode plot)

a)   Bode plots of $C(s)$ and $L(s) = G(s)C(s)$ for different values of T_I and T_D are shown in Figure A.94 and A.95. Here the red curve shows the nominal controller as designed in Exercise 2.5. The green curves represent the PID controller where the respective value is decreased by up to 60% and for the blue curves it is increased by up to 60/%, respectively. One can see that the integration time T_I can be used to move the lower corner frequency and the derivative time T_D to move the higher corner frequency of the controller. Starting values from Exercise 2.5 :

$$K_P = 1.2, \qquad T_I = 0.44, \quad \text{and} \quad T_D = 0.38$$
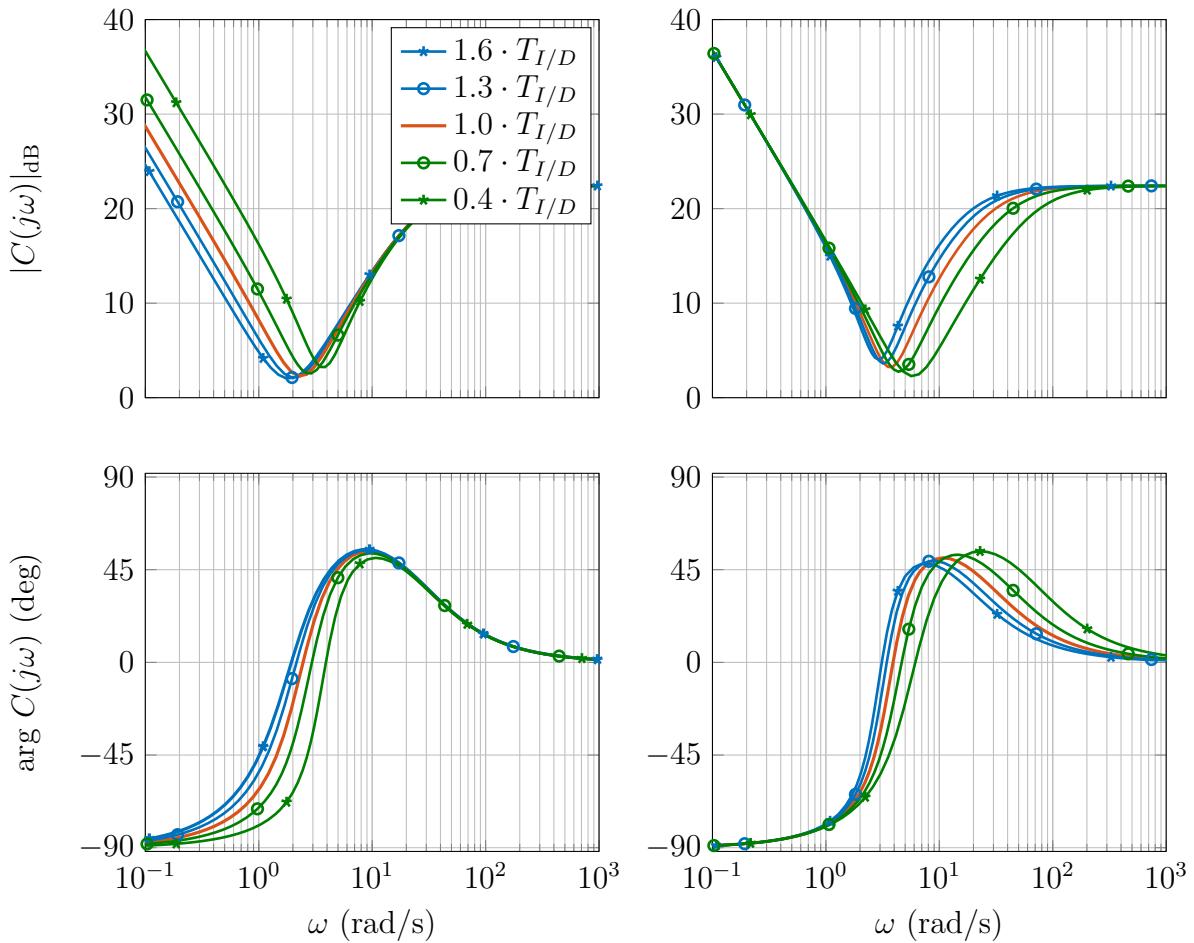
Figure A.94: Bode plot of C, variation of T_I (left) and T_D (right)

b)   The lead-lag compensators $C(s) = C_{lead}(s)C_{lag}(s)$ with

$$C_{lead}(s) = 0.56 \frac{1 + 0.29s}{1 + 0.0609s}$$

and

$$C_{lag,1}(s) = \frac{s + 0.3125}{s + 0.03125} \quad \text{or} \quad C_{lag,2}(s) = \frac{s + 0.5}{s + 0.005},$$
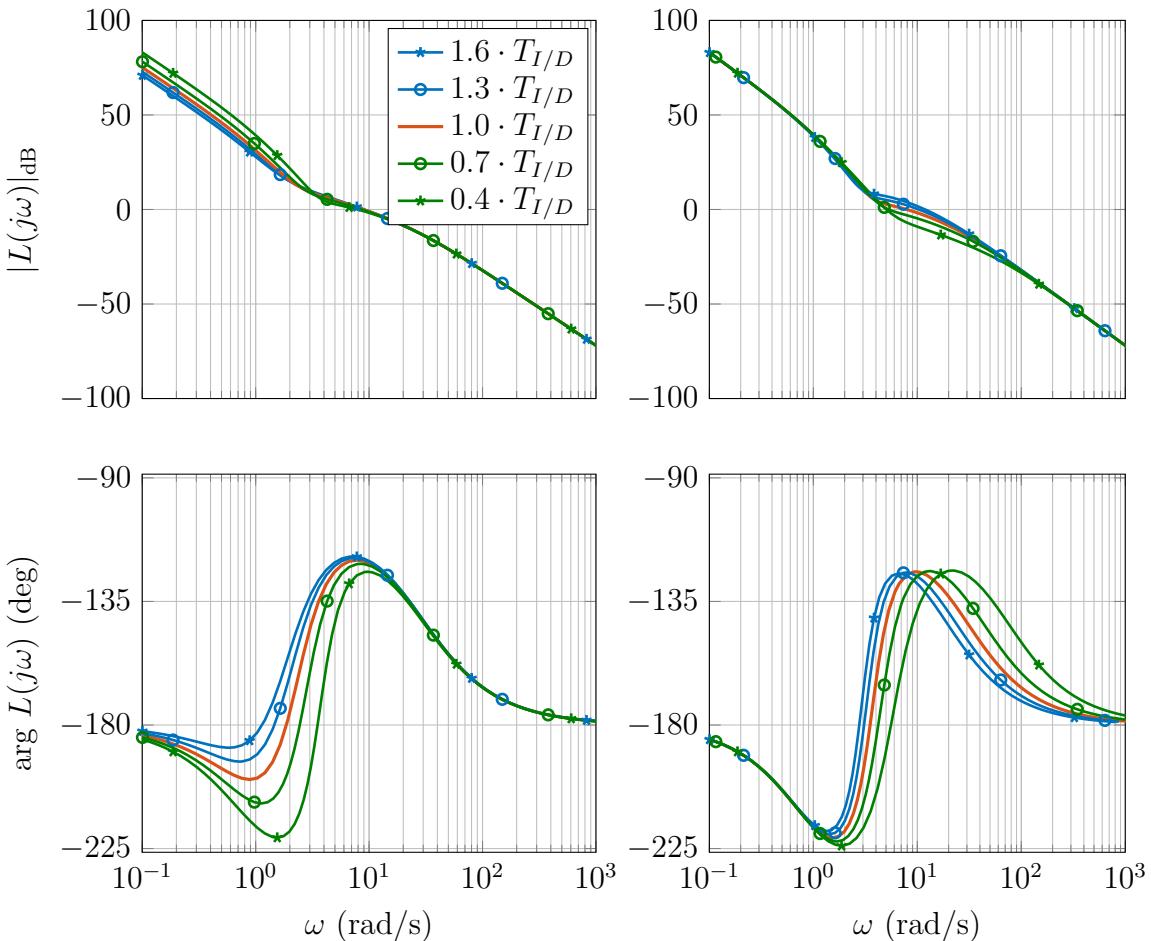
Figure A.95: Bode plot of L, variation of T_I (left) and T_D (right)

both discussed in Exercise 4.11, are compared with the PID controller from Exercise 2.5. In Figure A.96 the Bode plots of both controllers are shown.

While the lead-lag compensator has two real poles in the negative half plane, the PID controller has a single pole at the origin. The latter means that the magnitude plot of the PID controller has a slope of -20dB/dec and the phase starts at $-90°$.

The zeros of a lead-lag compensator can not become complex as can those of a PID controller. For the given values here, the PID controller in fact has a complex zero pair. That explains the fast increase of phase for $1 < \omega < 10$.

In Figure A.97, the open-loop transfer functions $L(s) = G(s)C(s)$ for these controllers are shown. It can be seen that the PID controller has a better phase margin and a larger cross over frequency.

c)   Figure A.98 shows the the step responses of the motor controlled by lead, lead-lag compensator and PID controller. The PID controller shows better performance: it rises faster with less overshoot. The better rise time can be explained by the Bode
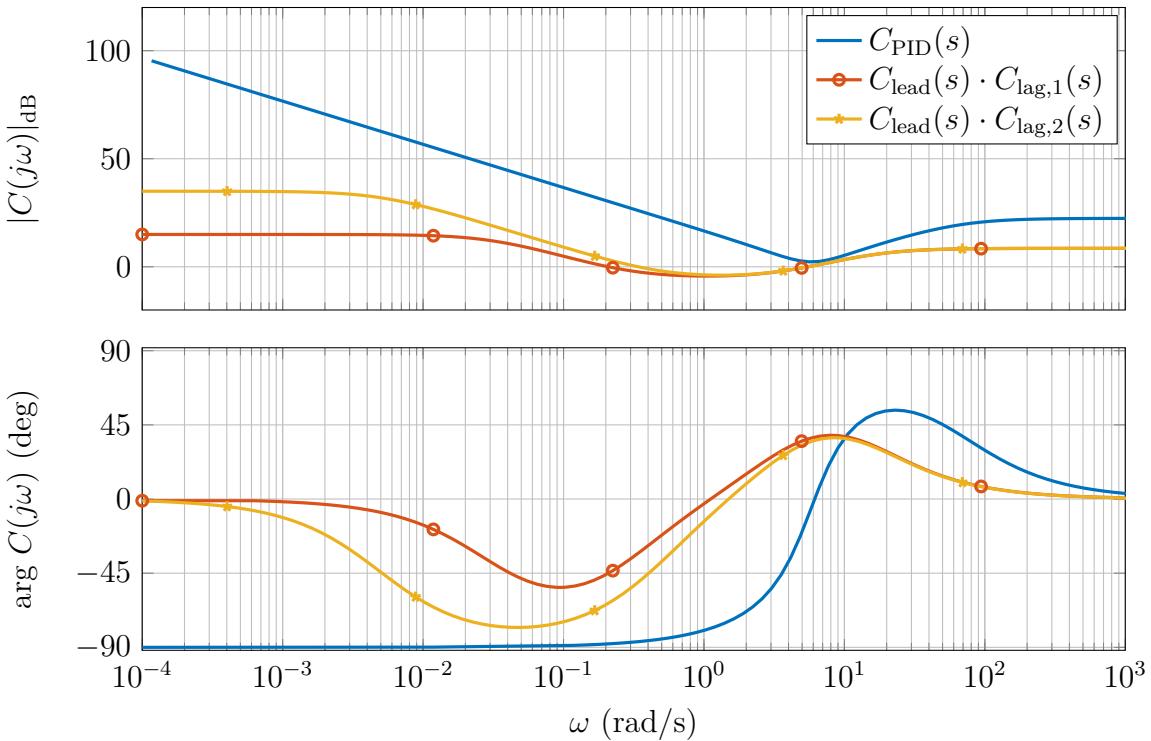
Figure A.96: Bode plot of lead-lag compensator (0.3125/0.03125), lead-lag compensator (0.5/0.005) and PID controller (PID nominal)

plot, since $t_s \approx \frac{1.7}{\omega_b}$. This is reflected in the Bode plot by the larger bandwidth and phase margin of the PID controller.
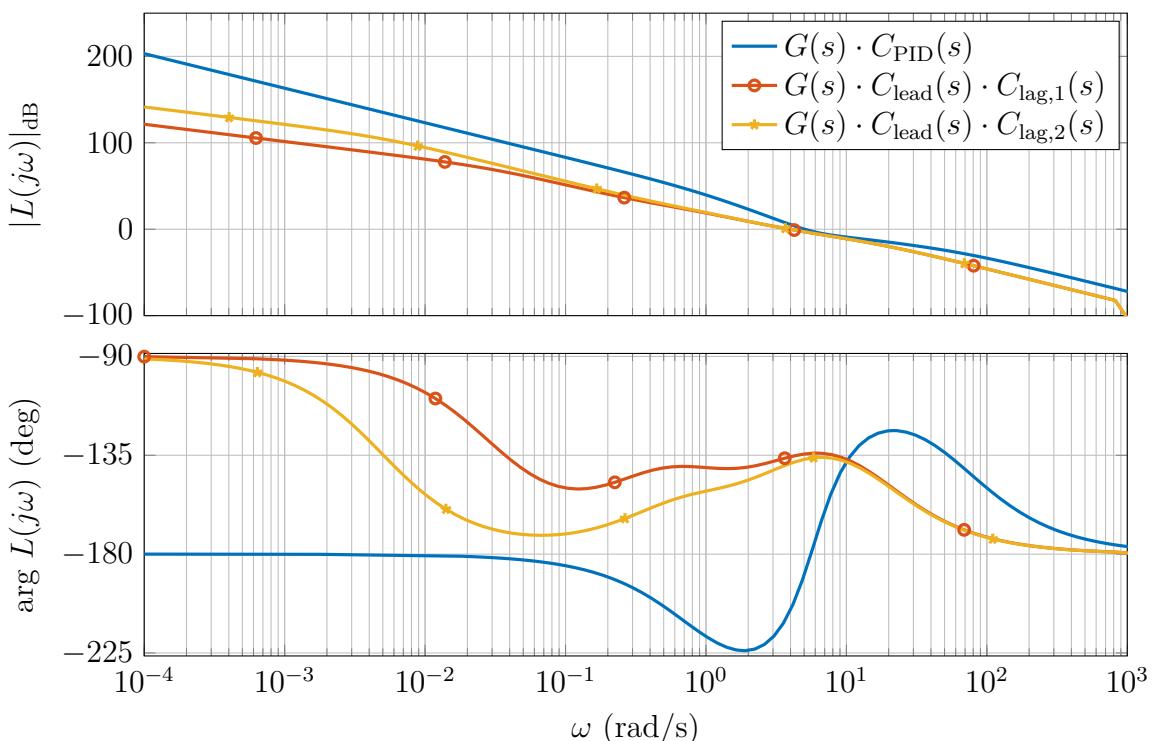
Figure A.97: Bode plot of L with lead-lag compensator (0.3125/0.03125), lead-lag compensator (0.5/0.005) and PID controller (PID nominal)
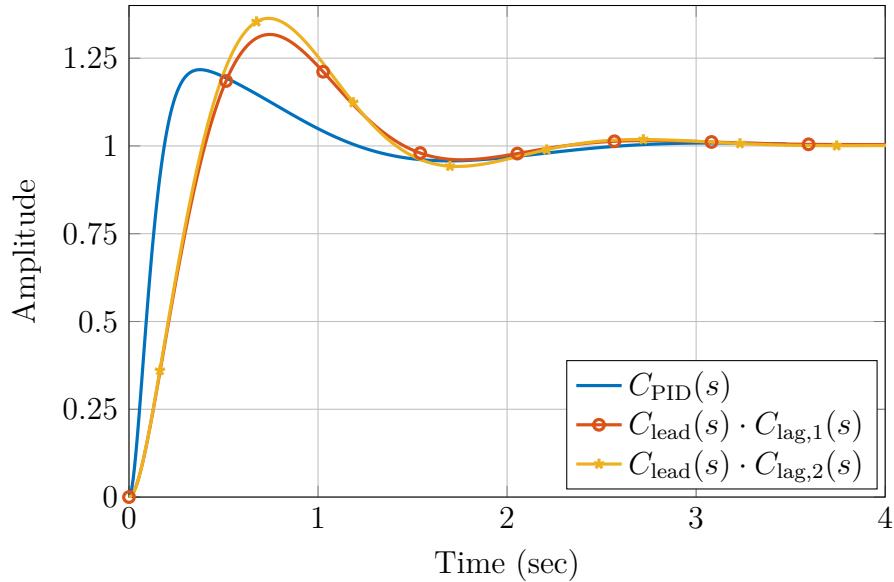
Figure A.98: Simulated step responses with lead-lag compensator (0.3125/0.03125), lead-lag compensator (0.5/0.005) and PID controller (PID nominal)

Problem 4.14 (Systune)

a) i) The lag part of the compensator is defined in the same way as the lead part. Taking the initial values from the given equation, the code to be added yields:

```
TLlag = realp('TLlag',0.3125);
alag = realp('alag',0.1);
Lag = tf([1 TLlag],[1 (TLlag*alag)]);
```

ii) Each section can be evaluated by *rightclick → evaluate current section*. The system's step responses can be directly plotted by `step(T0)` or for a better overview by `step(T0('y','r'))` and `step(T0('y','d'))` in different plots. It is obvious, that disturbances are not fully rejected. In order to plot the open-loop response, the system needs to be cut open. This is achieved by setting the loop switch X open, `T0.Blocks.X.Open = 1`. The phase margin can be directly read of the plot created by `margin(T0('y','r'))`. The phase margin is $PM = 48.9°$.

b) i) The parameters for `TuningGoal.Margins` are already given in the task. Since the goal is an open-loop requirement, the point-to-point measuring location for the response should be X. The desired rise time is $t_r = 0.45$ and thus $\omega_c \approx 1.7/t_r = 3.78$. Therefore, the response time to be set for `TuningGoal.Tracking` should be $t_{resp} = 2/\omega_c \approx 2/4 = 0.5$. The target steady state error does not

have much importance, since it is already implicitly demanded by input disturbance rejection which forces the controller to have integral behavior. The integral behavior leads to larger gains at low frequencies and leads to a low steady state error. The steady state error for the tuning goal should be chosen between $1 - 10$ %. The goal is supposed to be a bound on the closed-loop system, therefore the addressed transfer function must be $r \rightarrow y$. The code for both tuning goals can be defined as followed:

```
MargReq = TuningGoal.Margins('X',5,50);
TrackReq = TuningGoal.Tracking('r','y',0.5,0.1);
```

ii) After running Systune one should notice that ten sets of final values were returned. The option `RandomStart` forces Systune to perform a defined number of optimization runs. Using the MATLAB help, it is clear that a value close to 1 implies that the requirements were met. Moreover, the soft constraints are minimized subject to the hard constraints being below 1. The hard constraint on the pole region is necessary in order to keep the controller in realistic dimensions and maintain a reasonable comparison between the controllers.

Considering the returned values, one can see variations between each run. A value close to 1 implies satisfied tuning goals and some runs return values high above 1. Therefore, it seems reasonable to perform several runs since one might receive a single bad result if only one run is performed. The reason for this behavior is the algorithm used by Systune. It is based on local optimization techniques and global optimality cannot be guaranteed. As seen in the example, sometimes local solutions are found which aren't satisfying. Therefore, one should always perform several runs when using Systune.

c) i) The step responses can be plotted similar to a). E.g., by:

```
figure(1)
step(T0('y','r'),T1('y','r'))
figure(2)
step(T0('y','d'),T1('y','d'))
```

The rise time can be read off the plots by *rightclick → Characteristics → rise time*. For the tuned system, the rise time should be approximately $t_r \approx 0.2$.

ii) The bode diagram can be plotted similar to a). By using `margin()` the phase margin can be directly read off. The response time is calculated by $t_{resp} = 2/\omega_c$. The values should be approximately $PM \approx 70°$ and $\omega_c \approx 7$. Subsequently, the requirements are well met and better disturbance rejection was achieved, as seen in c) i), without worsening the original goals.

Problem 4.15 (Resonance rise)

a) The plant G(s) is stable, because the poles are in the left half plane.

b) i) Figure A.99 shows the Nyquist plot for positive frequencies.
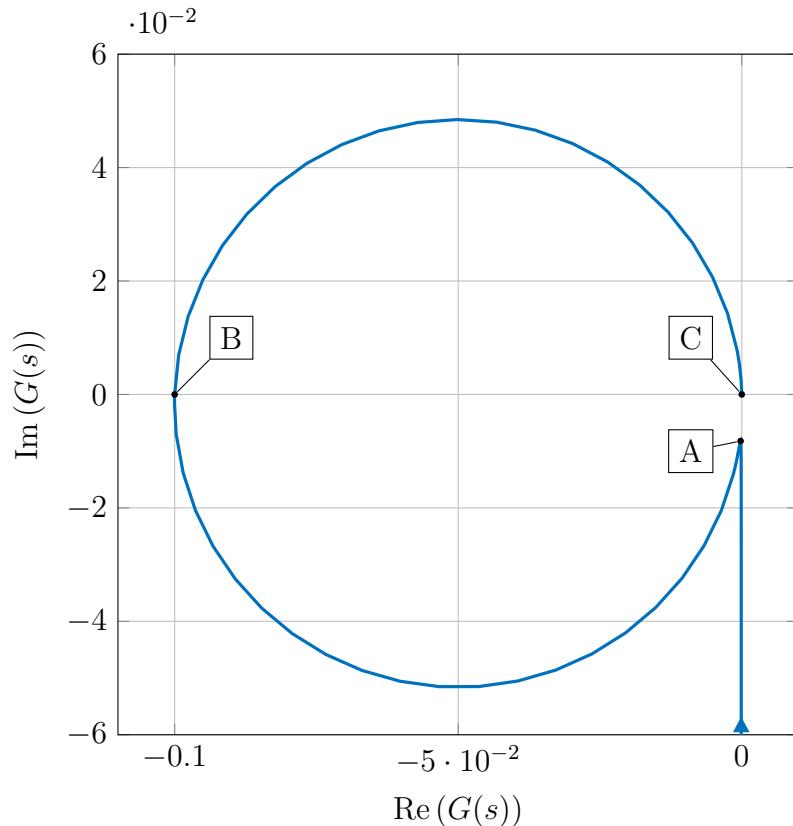
Figure A.99: Nyquist plot for positive frequencies

ii) Figure A.100 shows the Nyquist plot for positive and negative frequencies.

c) Checking the phase angle along a small semi-circle around the origin (*Nyquist path*) reveals that the Nyquist plot is completed by an infinite arc to the right.
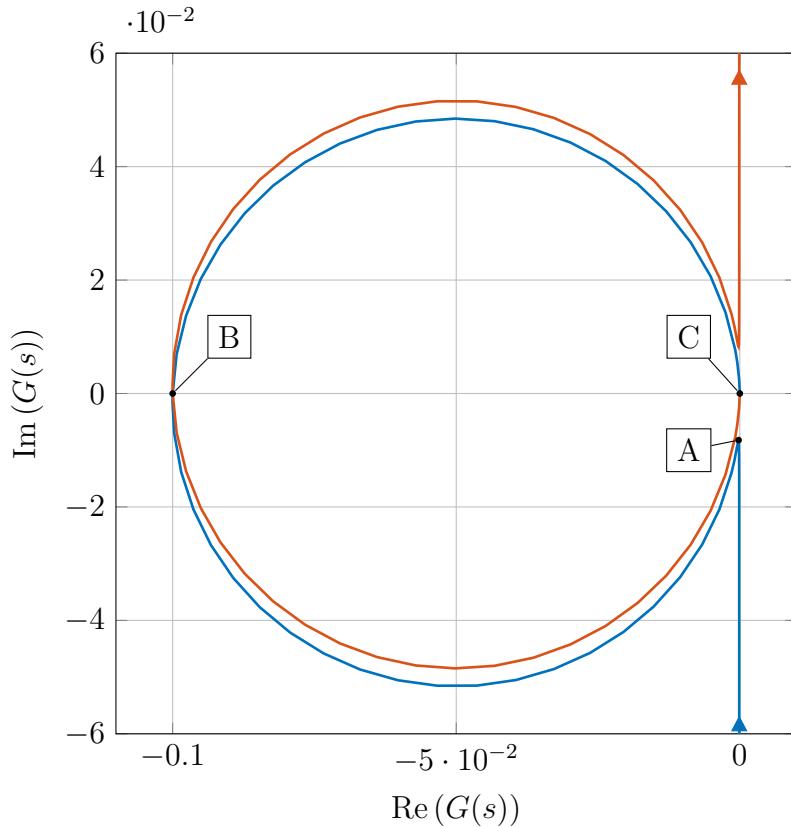
Figure A.100: Nyquist plot for positive (blue) and negative (red) frequencies

Problem 4.16 (Resonance rise)

a)   The range of values of K_P for which the closed-loop system is stable is $0 < K_P < 10$

b)   If the resonance peak crosses the 0dB-line the system becomes unstable.

c)   The transfer function $G(s) = \dfrac{100}{s(s^2 + s + 1000)}$ has three poles. One pole is at the origin and the other poles are complex cojugate at $-0.5 \pm 31.6i$. Figure A.101 shows the root locus; when K_P increases the complex conjugate pole pair leaves the left half plane and the system becomes unstable.
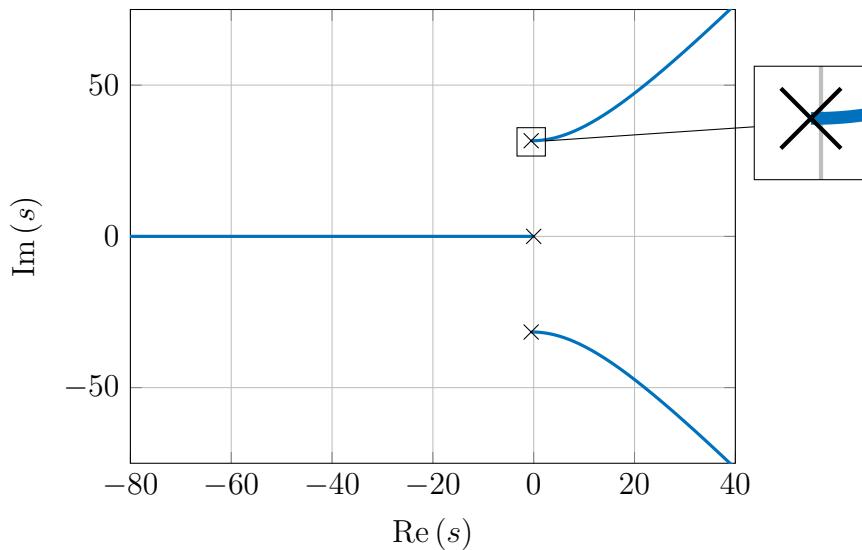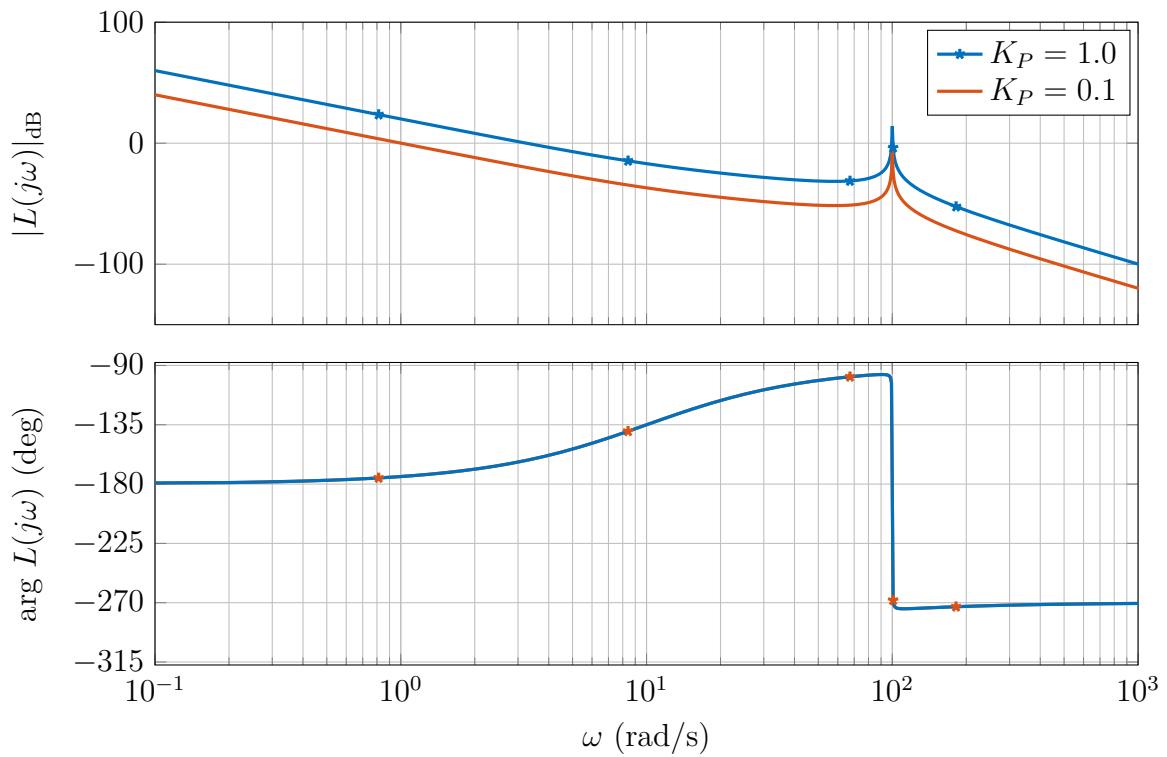
Figure A.101: Root locus

Problem 4.17 (Resonance peak)

a)   The Bode plot of the transfer function G(s) with $K_P = 1$ is shown in Figure A.102.

Figure A.102: Bode plot and $L(s)$ with $\omega_b = 1 rad/s$

b)   With $K_P = 0.1$ the closed-loop bandwidth is $\omega_b = \omega_c = 1 rad/s$, which can be seen
     in Figure A.102 (red curve) . The closed-loop system is now stable, because the
     resonance peak is not crossing the 0dB-line anymore.

c)   From the Bode plot in Figure A.102 we can see that $\phi_m \approx 55°$. Now α can be
     determined as

$$\alpha = \frac{1 - sin(\phi_m)}{1 + sin(\phi_m)} \approx 0.1.$$

     With $\omega_m = 2\omega_c = 2 rad/s$ and $\omega_L = \sqrt{\alpha} \cdot \omega_m = 0.63$ we have

$$C_1 = 0.1 \frac{1 + \frac{s}{0.63}}{1 + \frac{s}{6.3}}.$$

d)   To determine C_1, consider that the frequency $\omega = 100 rad/s$ is more than one decade
     above the pole of C_1. Thus we have

$$C_1(j100) \approx \lim_{\omega \to \infty} C_1(j\omega) = 0.1 \frac{6.3}{0.63} = 1$$

     Figure A.103 shows the new Bode plot of $L(s) = G(s)C_1(s)$. One can see that the
     resonance peak of $L(s) = G(s)C_1(s)$ is crossing the 0dB-line and the closed-loop
     system is again unstable.
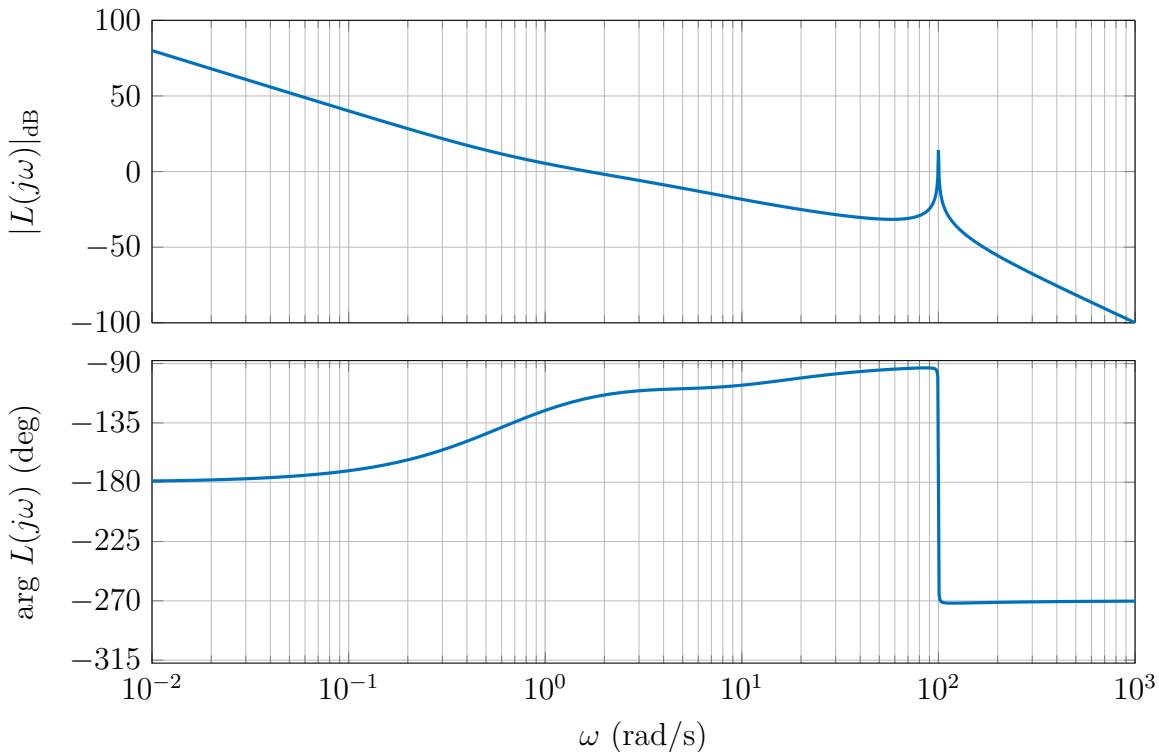
Figure A.103: Bode plot of $L(s) = G(s)C_1(s)$

e)   One possible choise of a compensator is:

$$C_2(s) = \frac{1}{\frac{s}{10} + 1} = \frac{10}{s + 10}$$

To not significantly effect the intended closed-loop bandwidth from (b) the new pole of $C_2(s)$ is one decade above the crossover frequency. Figure A.104 shows the new Bode plot for $L(s) = G(s)C_1(s)C_2(s)$. The resonance peak is not crosssing the 0dB-line anymore and the system is stable.
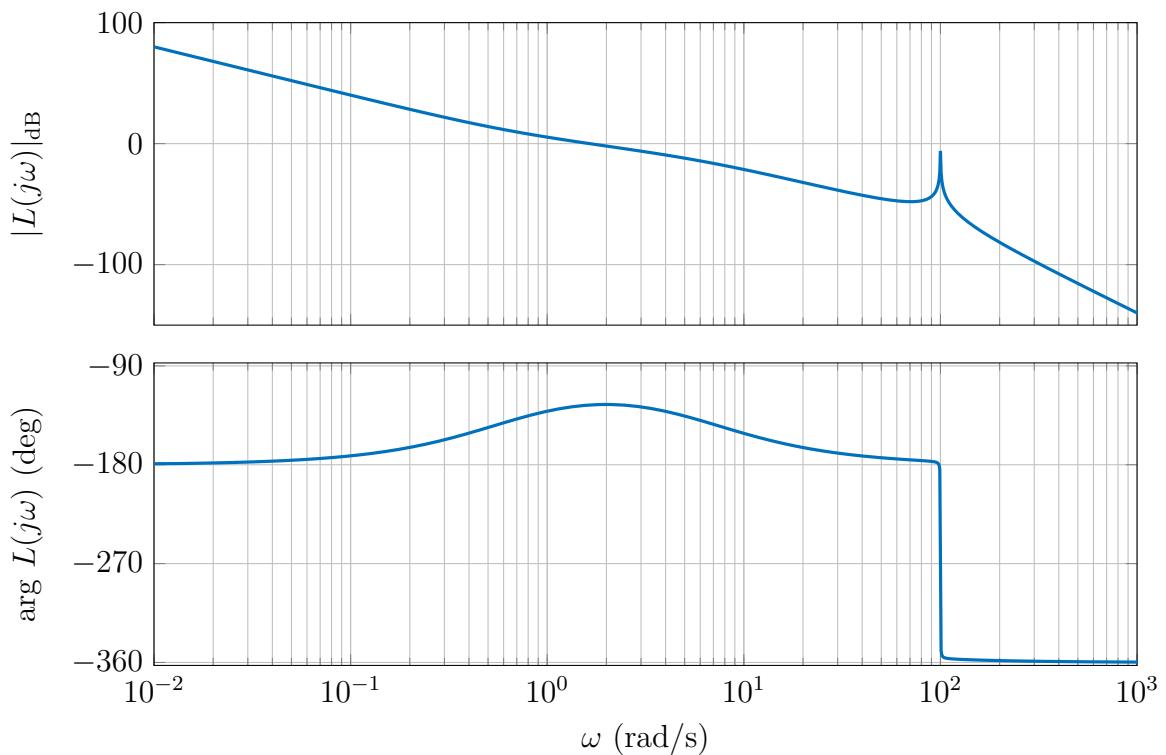
Figure A.104: Bode plot of $L(s) = G(s)C_1(s)C_2(s)$

A.5 Solutions for Chapter 5

Problem 5.1 (Tustin approximation)

a) A possible solution of Problem 4.9 is a lead compensator with transfer function

$$C(s) = 4\,\frac{s+1}{s+6}$$

Using the Tustin approximation (5.7) with

$$s = \frac{2}{T}\cdot\frac{1-z^{-1}}{1+z^{-1}} = \frac{2}{T}\cdot\frac{z-1}{z+1},$$

a discrete-time transfer function that approximates $C(s)$ can be computed as

$$D(z) = 4\,\frac{\frac{2}{T}\frac{z-1}{z+1}+1}{\frac{2}{T}\frac{z-1}{z+1}+6} = 4\,\frac{(T+2)z+(T-2)}{(6T+2)z+(6T-2)} = 4\cdot\frac{T+2}{6T+2}\cdot\frac{z+\frac{T-2}{T+2}}{z+\frac{6T-2}{6T+2}}$$

To finde a difference equation that approximates the controller behaviour we consider

$$D(z) = \frac{U(z)}{E(z)} = 4\cdot\frac{T+2}{6T+2}\cdot\frac{z+\frac{T-2}{T+2}}{z+\frac{6T-2}{6T+2}}.$$

Thus, we have

$$(6T+2)u(kT+T) + (6T-2)u(kT) = 4(T+2)e(kT+T) + 4(T-2)e(kT).$$

b) The static gains are

$$C(0) = \frac{2}{3}$$

$$D(1) = 4\,\frac{2T}{12T} = \frac{2}{3} = C(0)$$

c) For the sampling period $T = 0.1$ one gets

$$D(z) = \frac{8.4z - 7.6}{2.6z - 1.4} = 3.23\,\frac{z - 0.905}{z - 0.538}$$

See MATLAB solution: `Problem5_1_TustinApprox.mlx`

Problem 5.2 (Tustin simulation and sampling time)

MATLAB file: `Problem5_2_TustinSimulation.mlx`

The effect of undersampling is shown in Figures A.105, A.106 and A.107 .
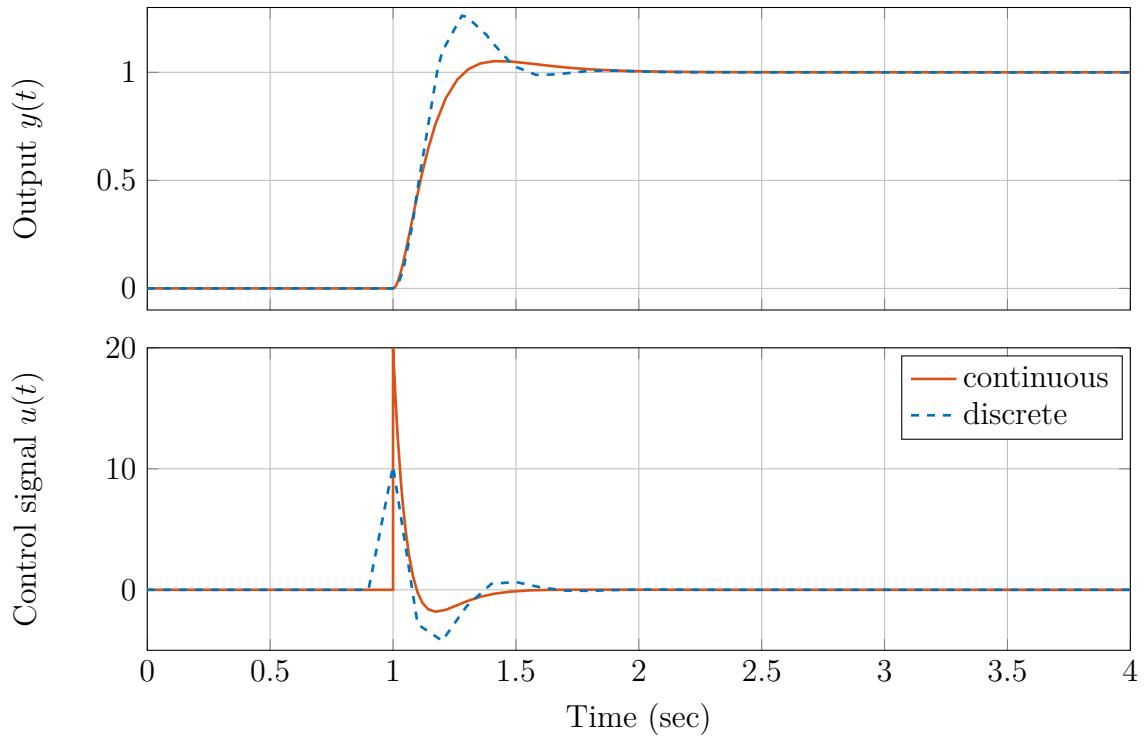
Figure A.105: Comparison between continuous-time and sampled-data control with sampling time T=0.1s
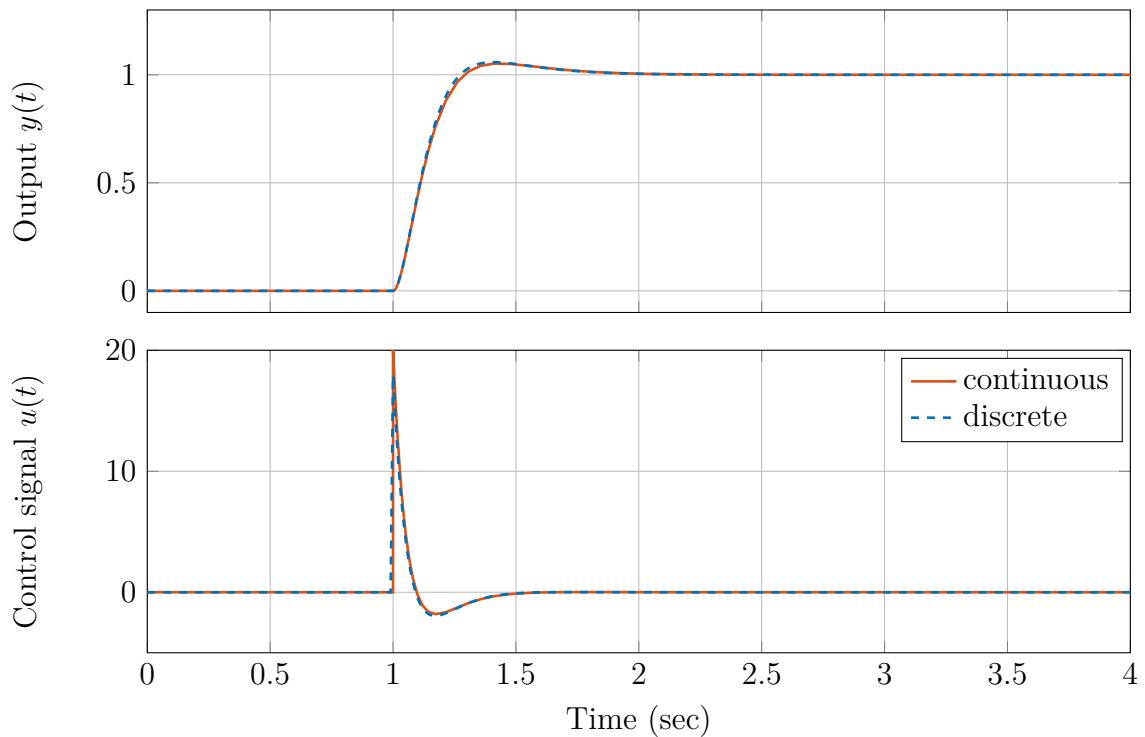
Figure A.106: Comparison between continuous-time and sampled-data control with sampling time T=0.01s
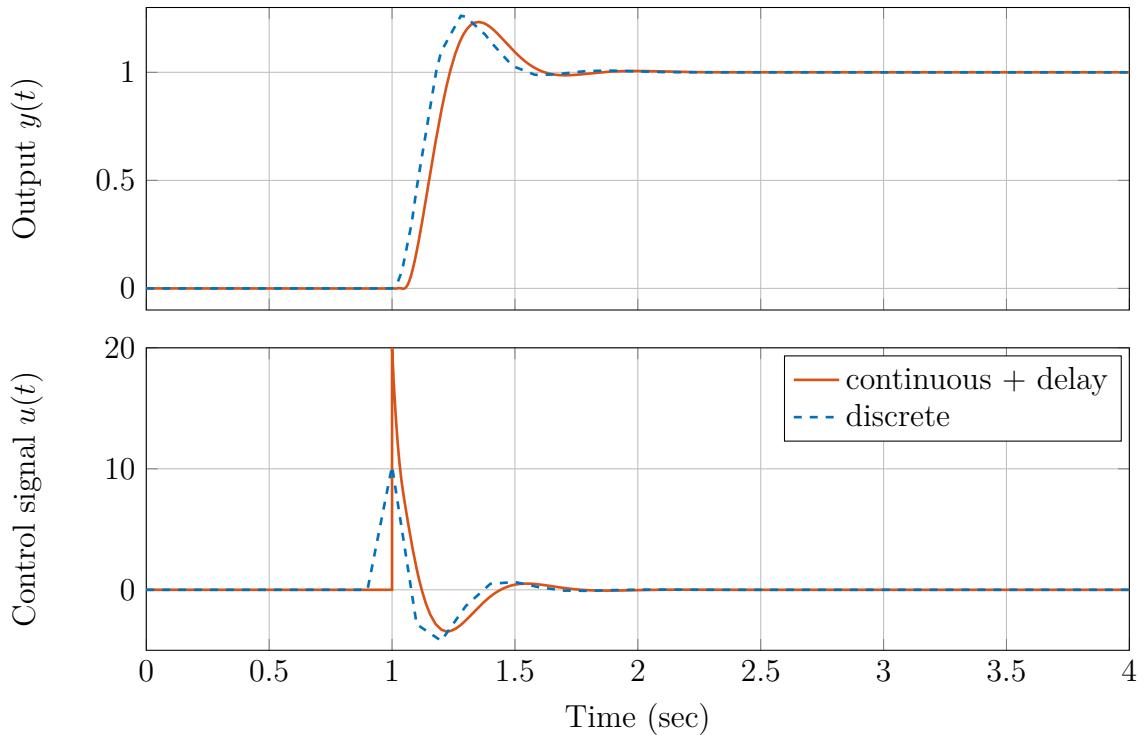
Figure A.107: Comparison between continuous-time and sampled-data control with sampling time time T=0.1s and time delay

Problem 5.3 (Discrete-time control of the DC Motor)

a)   The discretized controllers are:

for $T_0 = 10\,\text{ms}$

$$C_0(s) = \frac{2.662z - 2.578}{z - 0.8488},$$

$T_1 = 30\,\text{ms}$

$$C_1(s) = \frac{2.662z - 2.445}{z - 0.6115},$$

and for $T_2 = 100\,\text{ms}$

$$C_2(s) = \frac{2.662z - 2.211}{z - 0.1941}.$$

b)   A SIMULINK model as shown in Figure A.108 can be used to simulate the closed-loop step responses.
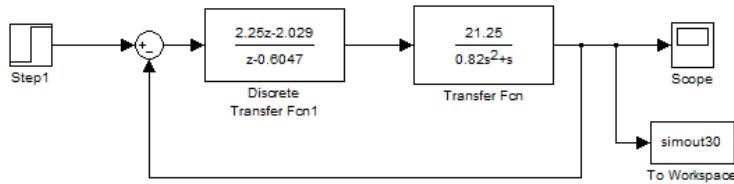
Figure A.108: SIMULINK model to simulate closed-loop step responses

In Figure A.109 the simulated closed-loop step responses with the continuous-time as well as the discretized lead compensators are shown.
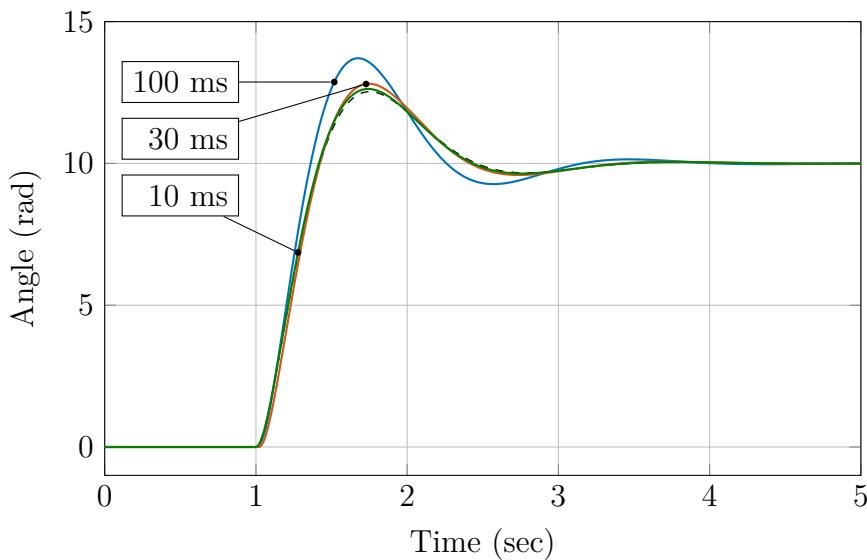
Figure A.109: Simulated closed-loop step responses with continuous-time and discrete-time lead compensators: continuous-time (black,dashed), discrete-time with $T_0 = 10\,\mathrm{ms}$ (green), $T_1 = 30\,\mathrm{ms}$ (red), and with $T_2 = 100\,\mathrm{ms}$ (blue)

The effect of undersampling can be clearly seen. Also, step responses in dotted lines are shown, which are the continuous-time step responses but with a time delay of $T_i/2$, where i corresponds to the given sampling time. One can observe, that $T_0 = 10\,\mathrm{ms}$ is a good choice for the sampling time and that the transient behaviour deteriorates for an undersampled system.

c) When comparing the step responses obtained by using the discretized controllers with the continuous-time responses, in which case a time delay of $T_i/2$ has been added to the model, a rather close agreement can be observed (see Fig. A.109). It might be useful to use the approximation to do stability analysis in frequency domain before implementing a controller on a real plant in its discretized version.

This way, we can use all the tools from continuous-time control theory and still take into account discrete-time implementation!

d) Figure A.110 shows the experimental closed-loop step responses for the sampling times T_0, T_1 and T_2.
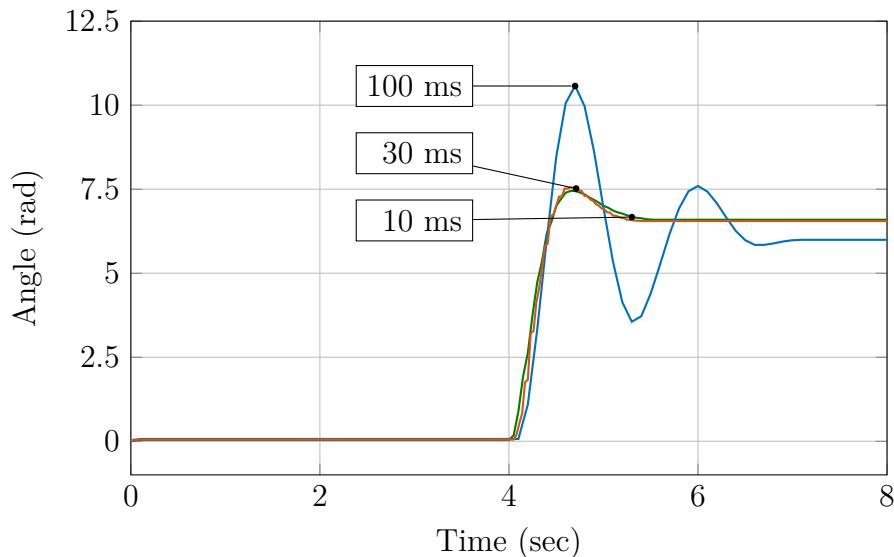
Figure A.110: Experimental closed-loop step response with lead compensators implemented with a sampling time of $T_0 = 10\,\text{ms}$ (green), $T_1 = 30\,\text{ms}$ (red) and of $T_2 = 100\,\text{ms}$ (blue)

As one can see, the effect observed in the simulation also occurs in the experiment: the peak overshoot seen in Figure A.110 is similar to the simulated one with a sampling time of $100\,\text{ms}$.

SIMULINK files:

- Model_Problem5_3_Sampling10ms.slx
- Model_Problem5_3_Sampling30ms.slx
- Model_Problem5_3_Sampling50ms.slx
- Model_Problem5_3_Sampling100ms.slx
- Model_Problem5_3_SimLeadContinuousDiscreteComparison.slx

Appendix B

Translation of Technical Terms

English	German
Actuator	Aktuator
Bandwidth	Bandbreite
Bi-proper	sprungfähig
Bode Diagram	Bode-Diagramm
Characteristic Equation	charakteristische Gleichung
Closed-Loop System	geschlossener (Regel-)Kreis
Continuous-Time	zeitkontinuierlich
Control Error	Regelfehler
Control Input / Controller Output	StellgröSSe
Control Loop	Regelkreis
Controller	Regler
Convolution	Faltung
Corner Frequency	Eckfrequenz
Damping Ratio	DämpfungsmaSS
Denominator	Nenner
Difference Equation	Differenzengleichung
Differential Equation	Differentialgleichung
Direct Feedthrough	direkter Durchgriff
Discrete-Time	zeitdiskret
Disturbance Rejection	Störgrößenunterdrückung
Dominant Pole Pair	dominantes Polpaar
Feedback	Rückführung
Feedback Control	Regelung
Feedforward Control	Steuerung
Final Value Theorem	Endwerttheorem
Frequency Response	Frequenzgang

English	German
Crossover Frequency	Durchtrittsfrequenz
Gain	Verstärkung
Gain Margin	Amplitudenreserve
Impulse Response	Impulsantwort
Initial Value Theorem	Anfangswerttheorem
Input Disturbance	Eingangsstörung
Internal Model Principle	Internes-Modell-Prinzip
Laplace Transform	Laplace Transformation
Linear Time Invariant	linear-zeitinvariant
Magnitude	Betrag
Minimum-Phase	minimalphasig
Natural Frequency	Eigenfrequenz
Numerator	Zähler
Open-Loop System	offener (Regel-)Kreis
Oscillation	Schwingung
Output Disturbance	Ausgangsstörung
Output / Measurement Equation	Ausgangsgleichung / Messgleichung
Partial Fraction Expansion	Partialbruchzerlegung
Peak Overshoot	maximale Überschwingweite
Phase Condition	Phasenbedingung
Phase Margin	Phasenreserve
Phase-Lag Compensation	nacheilende Phasenkorrektur
Phase-Lead Compensation	vorauseilende Phasenkorrektur
Physical Realizability	physikalische Realisierbarkeit
Plant	Regelstrecke
Plant Output	Regelstreckenausgang
Pole	Polstelle
Reference Command	Sollwert
Rise Time	Anstiegszeit
Root Locus	Wurzelortskurve
Sampling Frequency	Abtastfrequenz
Sensitivity	Empfindlichkeit
Settling Time	Ausregelzeit
Stable	stabil
Stability Criterion	Stabilitätskriterium
State Equation	Zustandsgleichung

English	German
State Space Model	Zustandsraummodell
State Variables	Zustandsvariablen
State Vector	Zustandsvektor
Static Gain / DC Gain	statische Verstärkung
Steady State	Stationärer Zustand
Steady State Error	bleibende Regelabweichung
Step Response	Sprungantwort
Strictly Proper	streng proper
System Response	Systemantwort
Time Constant	Zeitkonstante
Time Delay	Totzeit
Time Domain	Zeitbereich
Tracking	Sollwertfolge
Tracking Error	Führungsgrößenabweichung
Transfer Function	Übertragungsfunktion
Transient Response	Einschwingverhalten
Unstable	instabil
Zero	Nullstelle

Further Reading

[1] G. F. Franklin, J. D. Powell, A. Emami-Naeini, and J. D. Powell, *Feedback control of dynamic systems.* Prentice hall Upper Saddle River, 2002, vol. 4.

[2] K. Ogata and Y. Yang, *Modern control engineering.* Prentice hall India, 2010, vol. 5.

[3] R. C. Dorf and R. H. Bishop, *Modern control systems.* Pearson, 2011, vol. 12.

[4] T. M. Inc., *Get started with matlab*, 2020. [Online]. Available: `https://de.mathworks.com/help/matlab/getting-started-with-matlab.html`.

[5] ——, *Get started with simulink*, 2020. [Online]. Available: `https://de.mathworks.com/help/simulink/getting-started-with-simulink.html`.

Index