Discrete Time Control Systems

Lino Guzzella

Spring 2013

1 Lecture — Introduction

Inherently Discrete-Time Systems, example bank account

Bank account, interest rates $r_+ > 0$ for positive, $r_- > 0$ for negative balances

$$x(k+1) = \begin{cases} (1+r_+)x(k) + u(k), & x(k) > 0 \\ \\ (1+r_-)x(k) + u(k), & x(k) < 0 \end{cases} \tag{1}$$

where $x(k) \in \Re$ is the account's balance at time k and $u(k) \in \Re$ is the amount of money that is deposited to $(u(k) > 0)$ or withdrawn from $(u(k) < 0)$ the account.

In general such systems are described by a difference equation of the form

$$x(k+1) = f(x(k), u(k)), \quad x(k) \in \Re^n, \ u(k) \in \Re^m, \ f : \Re^{n\times m} \rightarrow \Re^n \tag{2}$$

with which an output equation of the form

$$y(k) = g(x(k), u(k)), \quad y(k) \in \Re^p, \ g : \Re^{n\times m} \rightarrow \Re^p \tag{3}$$

is often associated.

You will learn how continuous-time systems can be transformed to a form similar to that form.

Note that there is a fundamental difference between inherently discrete time systems and such approximations: for the former there is no meaningful interpretation of the system behavior in between the discrete time instances $k = \{1, 2, \ldots\}$, while the latter have a clearly defined behavior also between two “sampling” events.

Discrete-Time Control Systems

Most important case: continuous-time systems controlled by a digital computer with interfaces ("Discrete-Time Control" and "Digital Control" synonyms).

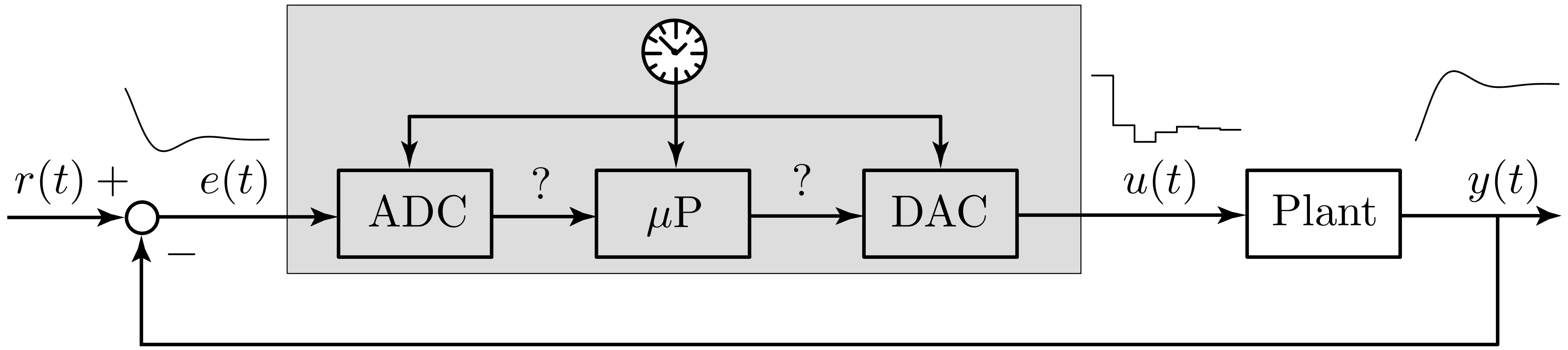
Such a discrete-time control system consists of four major parts:

1 The *Plant* which is a continuous-time dynamic system.

2 The *Analog-to-Digital Converter* (ADC).

3 The *Controller* (μP), a microprocessor with a "real-time" OS.

4 The *Digital-to-Analog Converter* (DAC) .



$r(t)$
$+$
$-$
$e(t)$
ADC
?
μP
?
DAC
$u(t)$
Plant
$y(t)$

The signals y, e, and u are continuous-time variables. The variables entering and exiting the microprocessor (block μP) are *sampled*, i.e., the ADC is an idealized *sampler*

$$e(k) = e(k \cdot T), \tag{4}$$

The constant parameter T is the *sampling time*.

The output of the block μP is again only defined at certain instances. Hold elements transform this variable into a continuous-time signal. *Zero-Order Holds* (ZOH)

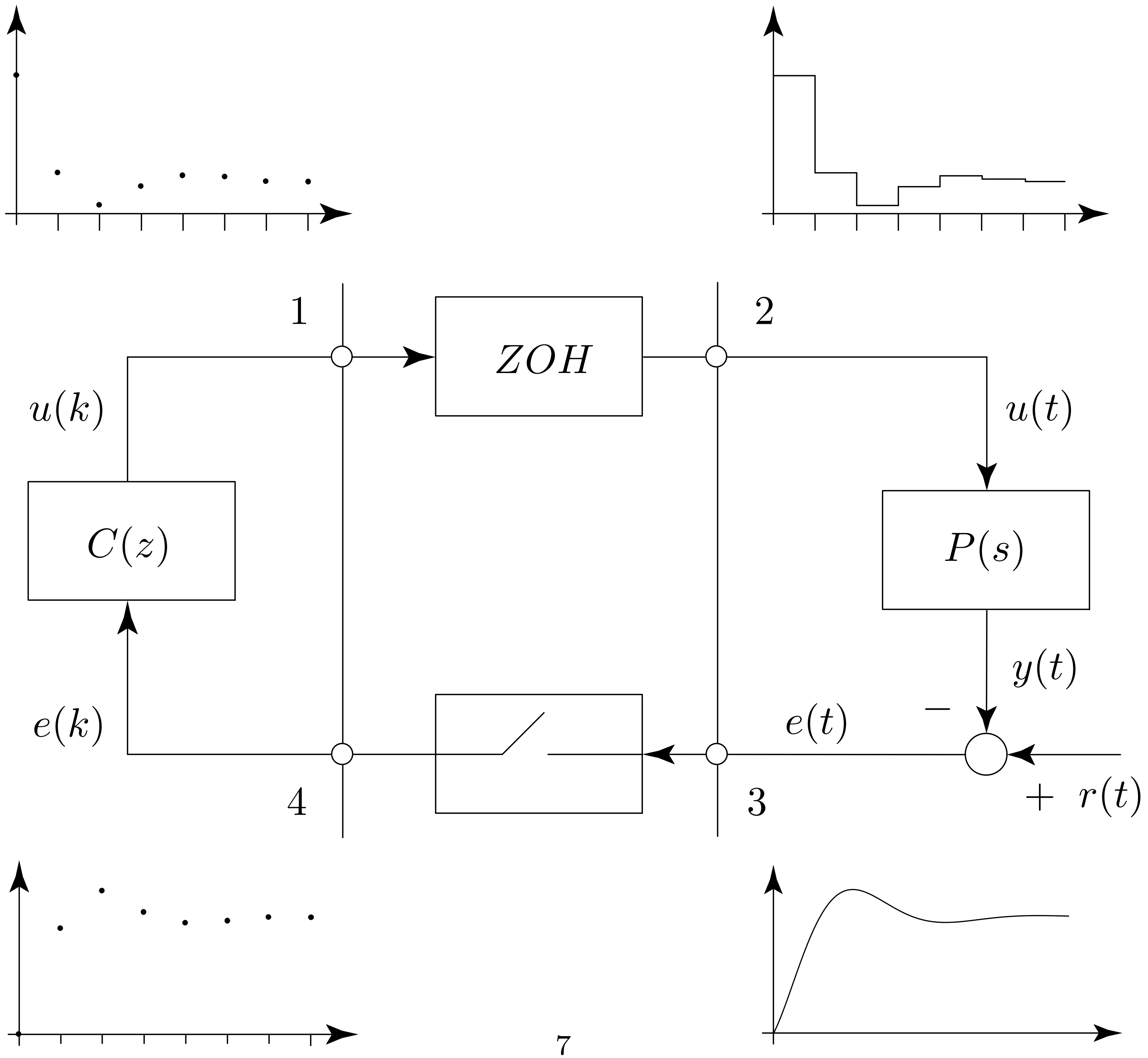
$$u(t) = u(k \cdot T) \qquad \forall t \in [k \cdot T,\ (k+1) \cdot T) \tag{5}$$

Higher-order holds available but seldom used.

Major challenge: loop contains both continuous-time and discrete-time parts. Two analysis approaches possible:

- The analysis is carried out in the continuous-time domain, and the discrete-time part has to be described by a continuous-time system with the input at point 3 and the output at point 2.
- The analysis is carried out in the discrete-time domain, and the continuous-time part has to be described by a discrete-time system with the input at point 1 and the output at point 4.

The two approaches are not equivalent. Obviously, the first one is more powerful since it will provide insights into the closed-loop system behavior for all times t. The second one will only yield information at the discrete time instances $k \cdot T$. Accordingly, it is to be expected that the second approach is easier to follow and this conjecture will be confirmed below.


1
2
3
4
ZOH
C(z)
P(s)
u(k)
u(t)
y(t)
e(k)
e(t)
r(t)
−
+

The output signal of the plant $y(t)$ is a function of the plant's input $u(t)$, a relation which may be described, for instance, by ordinary differential equations (in the finite dimensional case) or by transfer functions $P(s)$ (in the linear case).

Output $u(k)$ of the discrete controller $C(z)$ depends on its input $e(k)$ in a recursive way

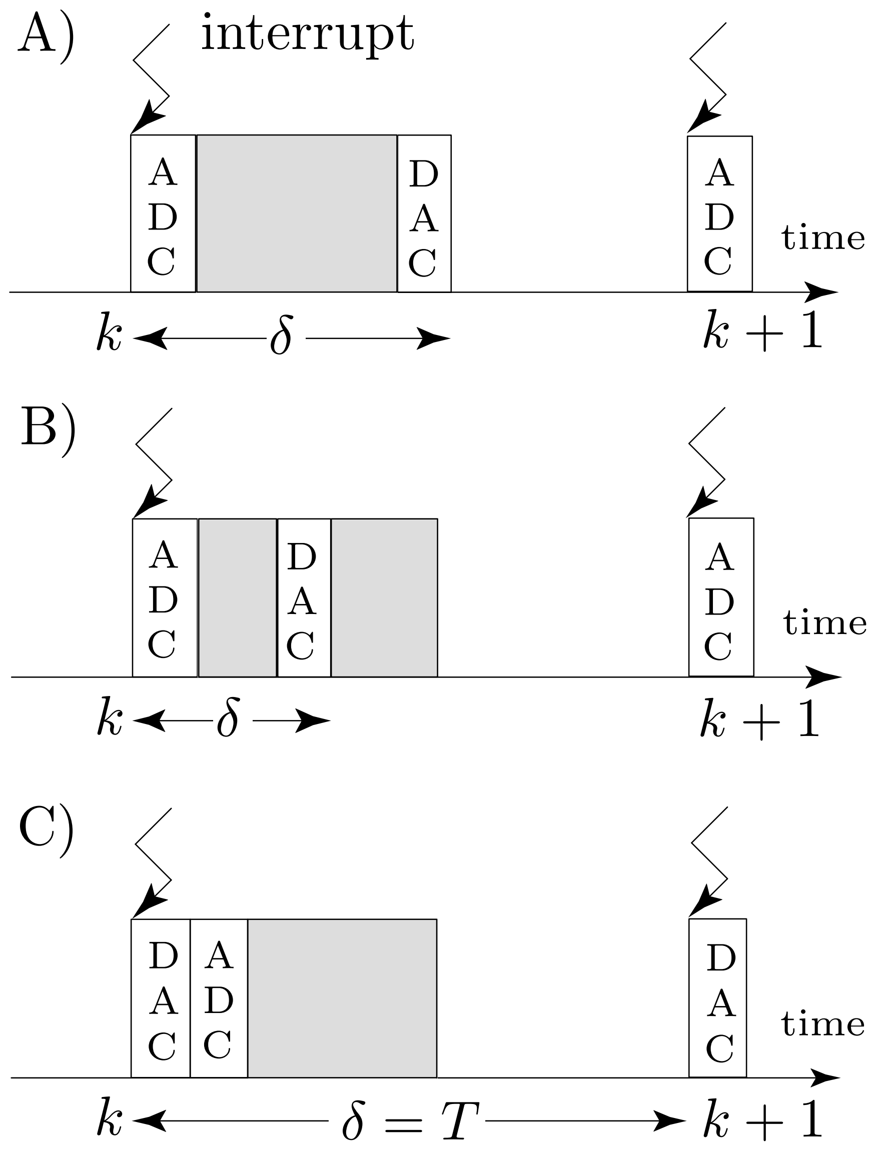
$$u(k) = f(e(k), e(k-1), \ldots, e(k-n), u(k-1), u(k-2), \ldots, u(k-n)) \quad (6)$$

For a linear controller, a frequency domain representation will be derived ($C(z)$).

The microprocessor performs these calculations once every sampling interval. Important aspects of this part are the synchronization and the computation delays (see folllowing Figure). Typically, the programming is done in a “high-level” computer language (“C” is often used). Hardware drivers provided by the manufacturers of the ADC and DAC.

For rapid prototyping control hardware/software systems are used. They are very convenient to test control algorithms because they dovetail with Matlab/Simulink. For series applications such systems are too expensive.

```
initialize system
repeat
    wait for interrupt
    data input
    compute controller output
    update shift registers
    data output
    terminate task
until done
shut down system
```



In all cases additional delays arise. They may be added to the plant during the controller design process, i.e., if $P(s)$ is the true plant, the design process is carried out for the fictitious plant $e^{-\delta \cdot s} \cdot P(s)$.

Unfortunately, the delay δ will not be constant in cases A) and B) (no real-time operating system can guarantee that). In order to cope with that the controller will have to have a sufficient robustness margin (in this case phase margin). Approach C) shows how to avoid varying computation delays by artificially delaying the output until the next interrupt stimulus arrives. This event will be strictly synchronous such that in this case constant delays result. The fictitious plant in this case will be $e^{-T \cdot s} \cdot P(s)$.

$$\begin{tikzcd}
P(s) \arrow[rr, "\text{continuous-time}\\ \text{synthesis}"] \arrow[dd, "ZOH\\ T\ \text{“large”}"'] \arrow[ddr, bend right, "\text{“}A\text{”}"] \arrow[drr, bend left, "\text{“}B\text{”}"] & & C(s) \arrow[d, "\text{emulation}\\ T\ \text{“small”}"] \\
 & & \widetilde{C}(z) \\
P(z) \arrow[r, "\text{discrete-time}\\ \text{synthesis}"'] & C(z) \arrow[ur, phantom, "\diagup"] &
\end{tikzcd}$$

Organization of this Text

Chapter 2: “emulation techniques” (following the path “B”), works well when the sampling times are much smaller than the relevant time constants of the system. No guarantee that stability or robustness properties are invariant to the transformation B.

Chapter 3: main properties of the sample-and-hold procedure, i.e., use continuous-time methods to describe the main effects of the sampling and holding mechanisms.

Chapter 4: mathematical description of discrete time signals and systems. The “$\mathcal{Z}$ transformation (the analog of the Laplace transformation), transformation of continuous-time systems to discrete-time systems and stability analysis.

Chapter 5: synthesis of control systems directly in the discrete-time domain (path “A”), “classical” (loop shaping, root-locus, etc.) and “modern” methods (LQR, LQG, etc.), “dead-beat,” etc.

2 Lecture — Emulation Methods

Introduction

In this chapter the *emulation approach* will be presented. First the key idea is introduced using analogies from the numerical integration of differential equations. As a “by-product” a shift operator z will be introduced using informal arguments. Some first system stability results can be obtained with that. The most useful emulation methods are then introduced and some extensions are shown which improve the closed-loop system behavior. The chapter closes with a discussion of the assumptions under which this approach is recommended and what limitations have to be respected.

Basic Ideas

Starting point: PI controller, in the time domain defined by

$$u(t) = k_p \cdot \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) \cdot d\tau \right] \tag{7}$$

and in the frequency domain by

$$U(s) = k_p \cdot \left[1 + \frac{1}{T_i \cdot s} \right] E(s) \tag{8}$$

If such a controller is to be realized using a digital computer, the proportional part does not pose any problems. The integrator requires more attention. The simplest approach is to approximate it by Euler's forward rule (initial condition $q(0) = 0$ is assumed)

$$q(k \cdot T) = \int_0^{k \cdot T} e(\tau) \cdot d\tau \;\approx\; \sum_{i=0}^{k-1} e(i \cdot T) \cdot T \tag{9}$$

For sufficiently small T this approximation yields a controller which produces a closed-loop behavior similar to the one observed in continuous time.

In a delay-free (the delay can be included in the plant dynamics) and recursive formulation the controller (7) can therefore be approximated in discrete time by

$$\begin{aligned} u(k \cdot T) &= k_p \cdot \left[e(k \cdot T) + \tfrac{1}{T_i} \cdot q(k \cdot T) \right], \\ q(k \cdot T + T) &= q(k \cdot T) + e(k \cdot T) \cdot T \end{aligned} \tag{10}$$

In order to simplify the reasoning below two definitions are useful:

- The notation $x(k)$ is adopted to denote the value of the variable x at time $k \cdot T$.
- The operator z is used to denote a forward shift by one sampling interval, i.e., $z \cdot x(k)$ is equal to $x(k+1)$.

The shift operator z is analogous to the Heavyside operator s ("differentiation"). The analogy can be carried further, i.e., the backward shift operation is denoted by z^{-1} with $z^{-1} \cdot x(k) = x(k-1)$.

The operator z can be used to "solve" linear difference equations

$$y(k+n)+a_{n-1}\cdot y(k+n-1)+\ldots+a_0\cdot y(k) = b_n\cdot u(k+n)+\ldots+b_0\cdot u(k) \tag{11}$$

which are transformed to

$$y(k)+z^{-1}\cdot a_{n-1}\cdot y(k)+\ldots+z^{-n}\cdot a_0\cdot y(k) = b_n\cdot u(k)+\ldots+z^{-n}\cdot b_0\cdot u(k) \tag{12}$$

and therefore

$$y(k) = \frac{b_n + z^{-1}\cdot b_{n-1} + \ldots + z^{-n}\cdot b_0}{1 + z^{-1}\cdot a_{n-1} + \ldots + z^{-n}\cdot a_0}\cdot u(k) \tag{13}$$

Apply this to the problem of approximating the PI controller

$$u(k) = k_p \cdot \left[1 + \frac{T}{T_i\cdot(z-1)}\right]\cdot e(k) \tag{14}$$

Therefore

$$s \approx \frac{z-1}{T} \tag{15}$$

The key idea of all emulation approaches is now to use the substitution (15) (or a similar one) to transform a given continuous-time controller transfer function $C(s)$ into a discrete-time transfer function $C(z)$, i.e.,

$$C(z) \approx C(s)\big|_{s=\frac{z-1}{T}} \tag{16}$$

and to use the shift properties of the variable z to derive a difference recursion for the controller output $u(k)$.

It is important to note that with the correspondence (15) a rational function $C(s)$ is transformed into a rational function $C(z)$. The reasons for that will become clear in a moment.

Example: Emulation of a Lead Controller

Assume that double integrator plant $P(s) = s^{-2}$ is controlled by

$$C(s) = 0.2 \cdot \frac{3 \cdot s + 1}{s + 1} \tag{17}$$

which is to be realized on a digital controller, $T = 1\ s$. Using Euler's forward emulation the result

$$C(z) \approx 0.2 \cdot \frac{3 \cdot (z-1) + 1}{(z-1) + 1} = 0.2 \cdot \frac{3 \cdot z - 2}{z} \tag{18}$$

is obtained. Recalling the definition of $C(s)$ yields

$$u(k) = C(z) \cdot e(k), \ \rightarrow \ u(k) \cdot z = e(k) \cdot (0.6 \cdot z - 0.4) \tag{19}$$

Using the shift properties

$$u(k+1) = 0.6 \cdot e(k+1) - 0.4 \cdot e(k) \tag{20}$$

$$u(k) = 0.6 \cdot e(k) - 0.4 \cdot e(k-1) \tag{21}$$

is obtained (for $T \neq 1$ two shift registers are required).

Euler Backward and Tustin Transformations

Forward (explicit) Euler approach is numerically not efficient (very small integration intervals T required).

Complex algorithms designed for efficient numerical integration not applicable to real-time control systems.

Two intermediate approaches often used for controller emulation: The first one is Euler backward (implicit) approach, which yields integration as

$$q(k+1) = q(k) + e(k+1) \cdot T \tag{22}$$

The second one is the bilinear approach (Heun's rule, trapezoidal or Tustin approximation)

$$q(k+1) = q(k) + \frac{1}{2}\left(e(k) + e(k+1)\right) \cdot T \tag{23}$$

Repeating the steps shown in the last section and using the discrete shift operator, the following two approximation rules are obtained

$$\text{Euler backward}: \quad s \quad \approx \quad \tfrac{z-1}{z \cdot T}; \quad \text{Tustin}: \quad s \quad \approx \quad \tfrac{2}{T} \cdot \tfrac{z-1}{z+1} \tag{24}$$

Especially the Tustin transformation is often used in practice. However, even this approach has its limitations and the emulated discrete-time closed-loop system performance is only comparable to the continuous-time performance if the sampling intervals are sufficiently small. More precisely, as long as the cross-over frequency ω_c and the sampling time T satisfy the inequality

$$T < \frac{\pi}{5 \cdot \omega_c}, \tag{25}$$

the emulated controller is *likely* to produce a satisfactory closed-loop system behavior.

The inverse mappings of the Euler forward, Euler backward, and Tustin are

$$m_1 : z = T \cdot s + 1, \qquad m_2 : z = \frac{1}{1 - T \cdot s}, \qquad m_3 : z = \frac{1 + s \cdot T/2}{1 - s \cdot T/2} \tag{26}$$

For continuous-time systems the asymptotically stable poles are known to be in the open complex left-hand plane. Figure 1 shows how this set is mapped under the transformations m_i.

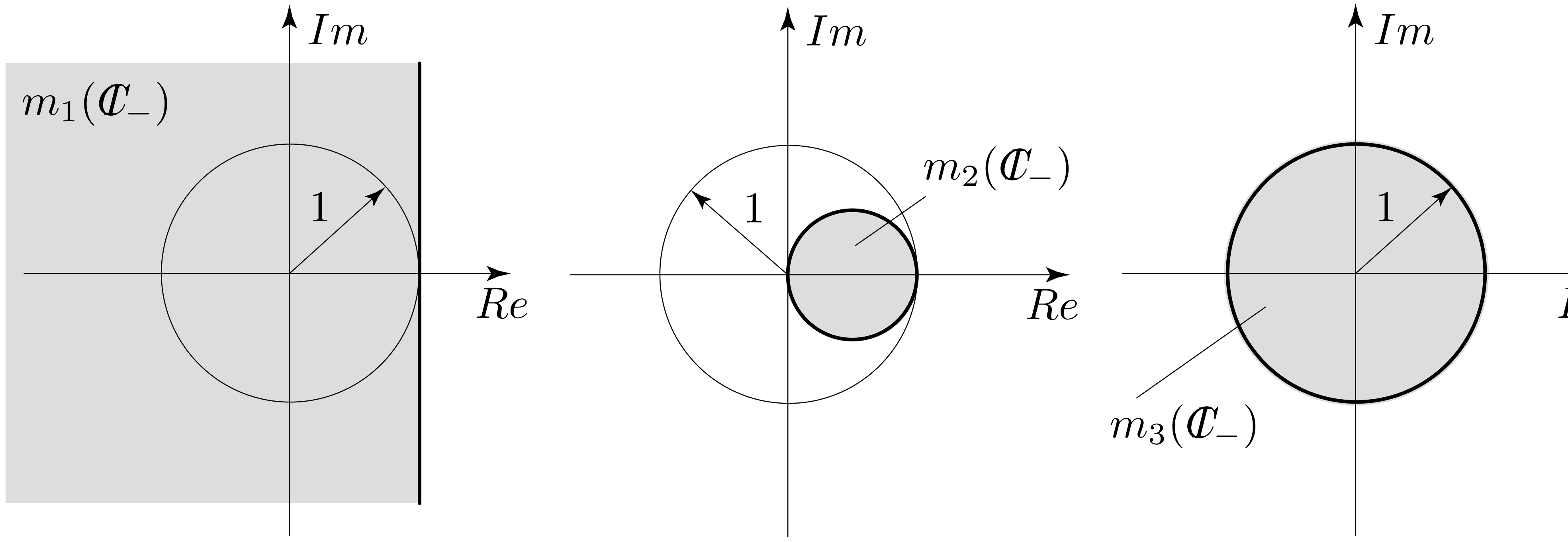


Figure 1: Value set of the mappings m_i

For a first-order system

$$x(k+1) = f \cdot x(k) + g \cdot u(k), \quad y(k) = c \cdot x(k) \tag{27}$$

impulse response $y_i(k)$ ($x(0) = 0$, $u(0) = 1$ and $u(k) = 0$ for all $k \neq 0$) is

$$y(0) = 0, \qquad y(k) = c \cdot f^{k-1} \cdot g \ \text{ for } \ k > 0 \tag{28}$$

Therefore, the system is asymptotically stable iff $|f| < 1$. Using the shift operator formalism

$$y(k) = \frac{c \cdot g}{z - f} \cdot u(k) \tag{29}$$

which shows that the system's pole is given by $z = f$.

Therefore asymptotically stable continuous-time first-order systems are always transformed into asymptotically stable discrete-time systems by m_2 and m_3, but the simpler mapping m_1 can produce unstable systems! Generalizations of these ideas to higher-order systems will be discussed later.

Frequency Domain Analysis of Emulation Methods

The key property of z^{-1} is to shift a time sequence one step backward

$$z^{-1} \cdot x(k) = x(k-1) \tag{30}$$

If $x(k)$ has been obtained by sampling a continuous-time signal $x(t)$, the operator z^{-1} can be defined as follows

$$z^{-1} \cdot x(k) = \left[e^{-s \cdot T} \cdot x(t)\right]_{t=k \cdot T} = e^{-s \cdot T} \cdot [x(t)]_{t=k \cdot T} \tag{31}$$

The shift operator and its inverse can therefore be identified by

$$z = e^{s \cdot T} \quad \text{and therefore} \quad s = \frac{1}{T} \ln(z) \tag{32}$$

The series expansion of the second equation is

$$\frac{1}{T} \ln(z) = \frac{2}{T} \left\{ \frac{z-1}{z+1} + \frac{1}{3} \cdot \frac{(z-1)^3}{(z+1)^3} + \ldots \right\}, \quad |z| < 1 \tag{33}$$

which shows that the Tustin transformation is the first-order rational approximation of (32).

Of course one might be tempted to use the transformation (32) to emulate a given controller $C(s)$ in a discrete-time environment. Unfortunately, the system

$$C(z) = C(s)\big|_{s=\frac{1}{T}\cdot\ln(z)} \tag{34}$$

is no longer rational in z and therefore there is no obvious way to transform the expression (34) into a discrete-time system following the procedure shown in the example on slide 24.

A frequency domain interpretation of the two equations (32) and (33) is derived now. For the sake of simplicity, the following example of $C(s)$ will be used

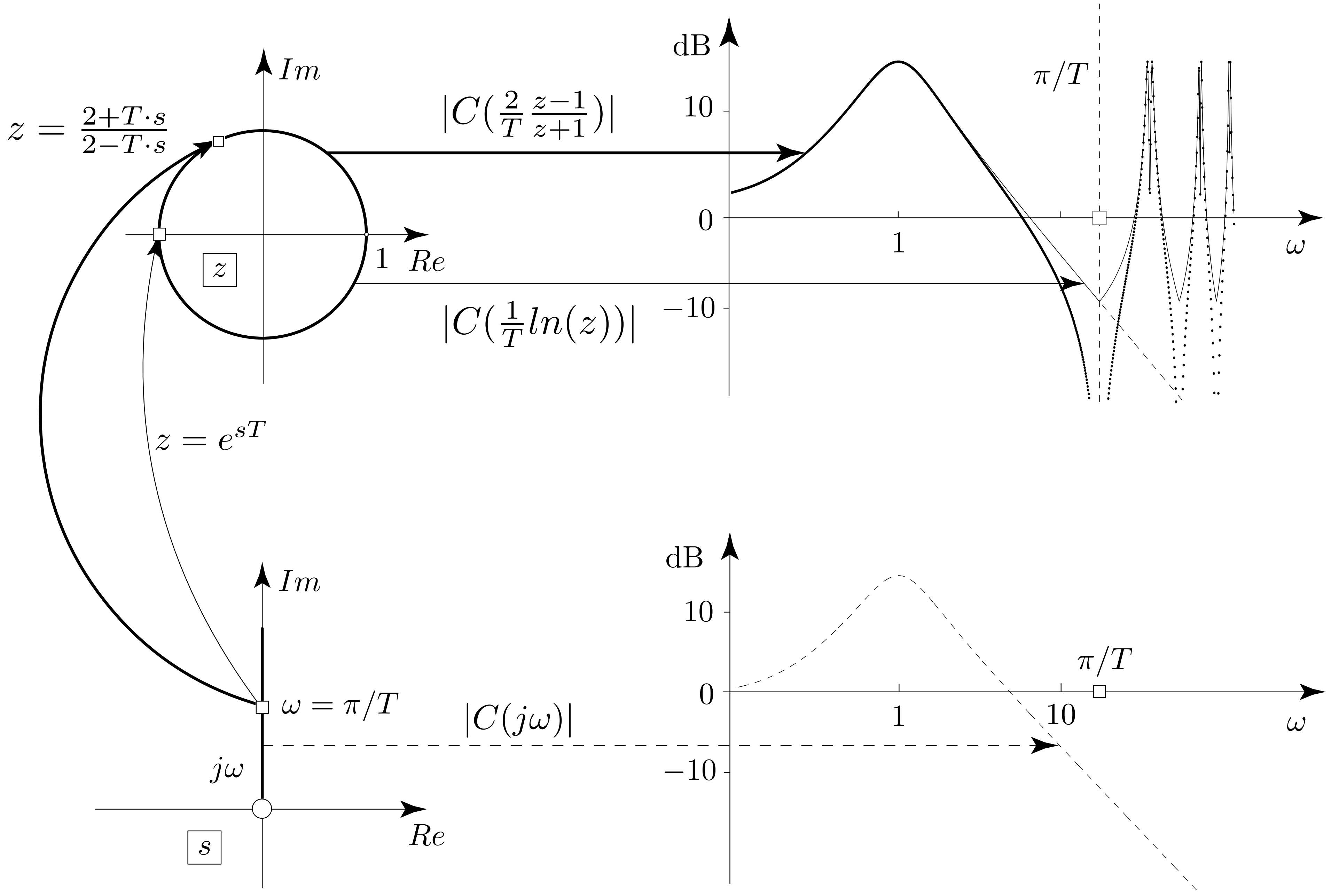
$$C(s) = \frac{5 \cdot s + 1}{s^2 + s + 1} \tag{35}$$

Upper left-hand plot: imaginary axis mapped by $e^{s \cdot T}$ ($T = 0.2\ s$ in this example) or Tustin. Value set is in both cases the unit circle. However, in the exact emulation the circle is transversed (infinitely) many times, while Tustin maps the positive imaginary axis only on the upper semi-arc.

The Nyquist frequency defined by

$$\omega_N = \frac{\pi}{T} \tag{36}$$

is indicated by a small rectangle; $e^{s \cdot T}$ maps $j\,\omega_N$ to $z = -1$ while Tustin maps it to $z = -0.423\ldots + j\,0.906\ldots$.



Im
$z = \frac{2+T\cdot s}{2-T\cdot s}$
$|C(\frac{2}{T}\frac{z-1}{z+1})|$
dB
π/T
10
0
1
Re
z
1
ω
$|C(\frac{1}{T}ln(z))|$
-10
$z = e^{sT}$
dB
Im
10
π/T
0
$\omega = \pi/T$
1
10
ω
$|C(j\omega)|$
-10
$j\omega$
Re
s

Upper right-hand plot: mapping the points on the unit circle again to a Bode magnitude plot for the same transfer function (35). The dashed curve is the original Bode plot as shown in the lower right-hand figure.

The thin black curve is the Bode plot obtained using the exact emulation. Up to the Nyquist frequency this curve coincides with the original Bode plot, after that the two curves separate. This is due to an "aliasing" effect which is caused by the ambiguity introduced by the mapping $z = e^{s \cdot T}$.

The bold black curve is the Bode magnitude plot obtained using the Tustin transformation. Due to the frequency distortions introduced by the Tustin approximation this curve deviates from the original one even for frequencies below ω_N.

Intuitively, the results shown in the last figure seem to indicate that the sampling times T must be chosen such that “the interesting part of $C(s)$” is in a frequency range well below the Nyquist limit (36). This confirms the rule of thumb (25) which was given earlier without an explicit justification.

The frequency domain interpretation shown in the last figure is also useful to understand an emulation method that is known as the “prewarped Tustin” approach. In this method the Bode plots of the original design $C(s)$ and of its emulation $\widetilde{C}(z)$ are exactly matched both in magnitude and phase at one desired frequency ω^* by prewarping the frequency s.

Prewarped Tustin Approximation of a First-Order System

Example: find emulation $\widetilde{C}(z)$ of $C(s) = \frac{k}{\tau \cdot s+1}$. The two filters must coincide in their frequency response at ω^*, i.e., $C(j\omega^*) = \widetilde{C}(e^{j\omega^* T})$. Key idea: use Tustin emulation with an additional parameter $\sigma > 1$

$$s = \frac{2}{\sigma \cdot T} \cdot \frac{z-1}{z+1} \tag{37}$$

Therefore

$$\frac{k}{j \cdot \tau \cdot \omega^* + 1} = \frac{k}{\dfrac{2\tau}{\sigma T} \cdot \dfrac{e^{j\omega^* T} - 1}{e^{j\omega^* T} + 1} + 1} \tag{38}$$

or

$$\begin{aligned}
j \cdot \sigma \cdot \frac{\omega^* T}{2} &= \frac{e^{j\omega^* T} - 1}{e^{j\omega^* T} + 1} && (39) \\
&= \frac{e^{j\omega^* T/2} \cdot e^{j\omega^* T/2} - e^{j\omega^* T/2} \cdot e^{-j\omega^* T/2}}{e^{j\omega^* T/2} \cdot e^{j\omega^* T/2} + e^{j\omega^* T/2} \cdot e^{-j\omega^* T/2}} && (40) \\
&= j \cdot \tan(\frac{\omega^* T}{2}) && (41)
\end{aligned}$$

The resulting equation

$$\sigma \cdot \frac{\omega^* T}{2} = \tan(\frac{\omega^* T}{2}) \tag{42}$$

has always a nontrivial solution for $\sigma > 1$ and $0 < \omega^* < \pi/T$.
Therefore:

- Choose the matching frequency ω^*.
- Compute the prewarping factor σ as follows

$$\sigma = \tan(\frac{\omega^* T}{2}) \cdot \frac{2}{\omega^* T} \tag{43}$$

- Apply prewarped Tustin (37) and compute $\widetilde{C}(z)$.

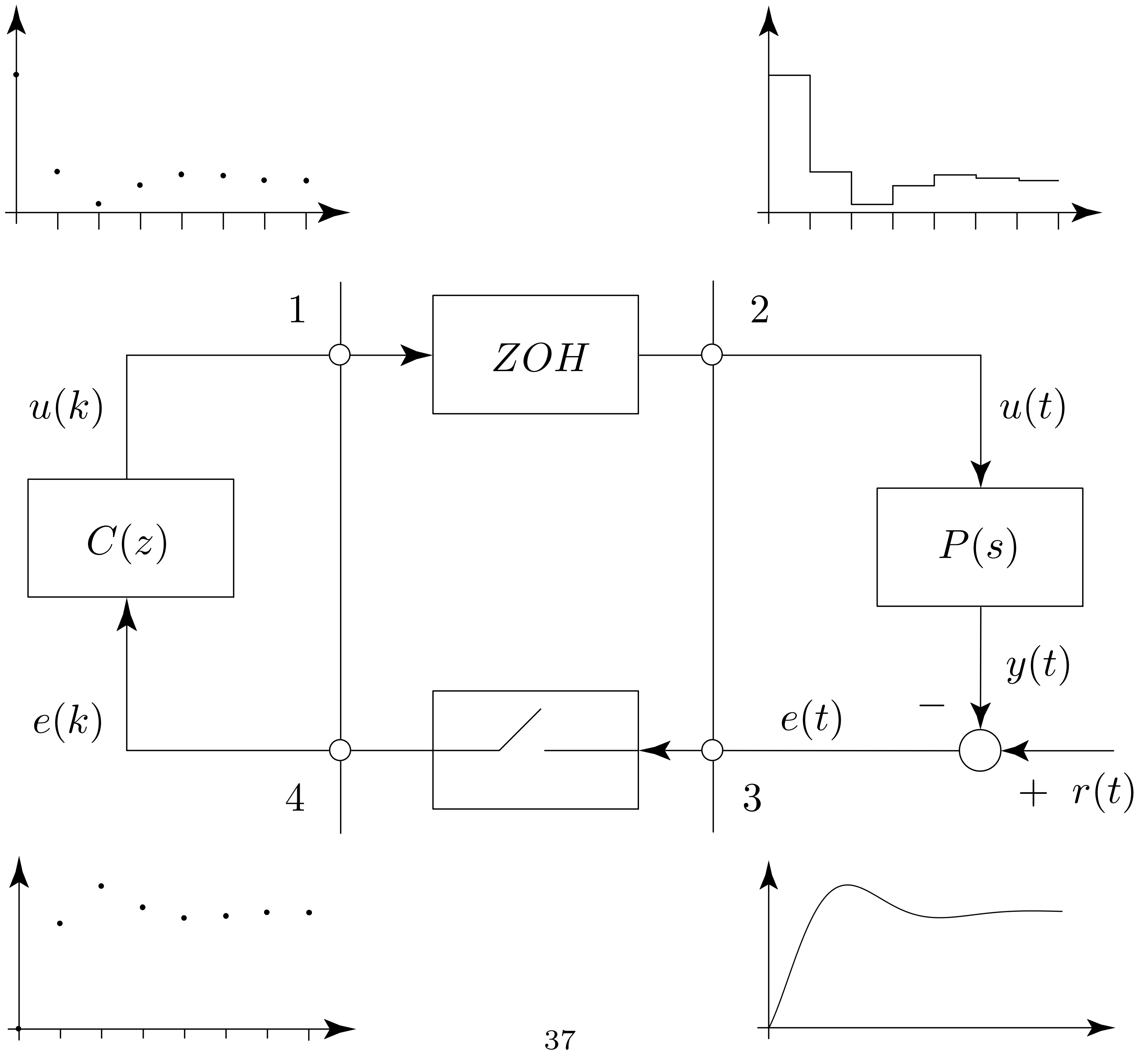
Notice that since the equation (42) does not depend on any system parameters (k or τ), the prewarping algorithm developed for a special case is applicable to arbitrary systems. The interpretation is that the frequency s of $C(s)$ is “prewarped” to $s \cdot \sigma$ prior to the application of the regular Tustin transformation (24).

The actual computation of the discrete-time emulation of a non-trivial controller $C(s)$ can be quite cumbersome. To facilitate that step, all modern CACSD contain commands that automatically compute the corresponding expressions. In Matlab this is the command `c2dm` for which, as usual, the `help` command provides much more information.

3 Lecture — Continuous-Time Analysis of Sample-And-Hold Elements

Discrete-time sampled-data systems will be analyzed using continuous-time methods. A full analysis including arbitrary controllers $C(z)$ and closed-loop systems is beyond the scope of this text. Here only unity-gain controllers and open-loop systems will be analyzed. This will permit to understand general sampled-data systems.

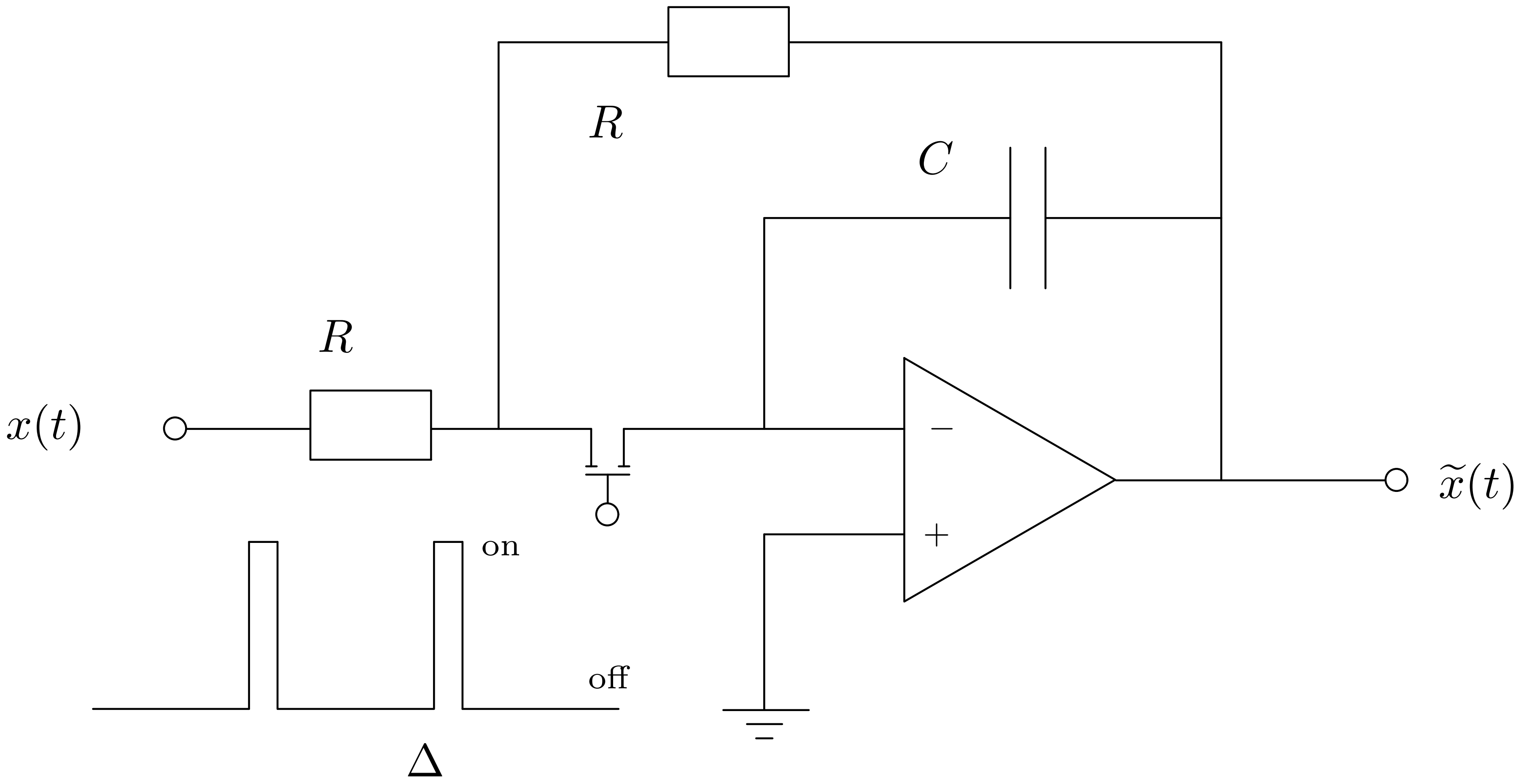
One possible way to approach sampled-data systems is to use continuous-time methods by analyzing the system shown in the figure on the next slide with the point 3 as input and point 2 as output signal. For the sake of simplicity $C(z) = 1$ will be assumed and the system is assumed to be in an open-loop configuration.


1
2
ZOH
u(k)
u(t)
C(z)
P(s)
y(t)
e(k)
e(t)
−
+
r(t)
4
3

Under the mentioned assumption, the “digital parts” behave similar to a ideal sample-and-hold element. A sample-and-hold device can easily be built using operational amplifiers and fast switches (for instance FET transistors, see figure below). When the transistor is “on”, the output voltage is defined by the differential equation

$$\frac{d}{dt}\widetilde{x}(t) = \frac{-1}{R \cdot C}\left[x(t) + \widetilde{x}(t)\right] \tag{44}$$

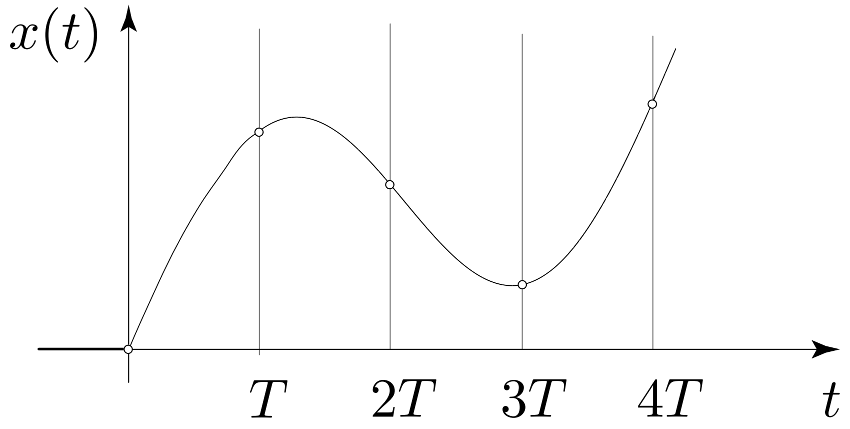
while in the “off” state, the output voltage is kept constant. This device works well if the time constant $\tau = R \cdot C$ is at least ten times smaller than the “on” interval Δ which has to be at least ten times smaller than any relevant time constant of the signal to be sampled.


R
C
R
$x(t)$
$-$
$+$
$\widetilde{x}(t)$
on
off
Δ

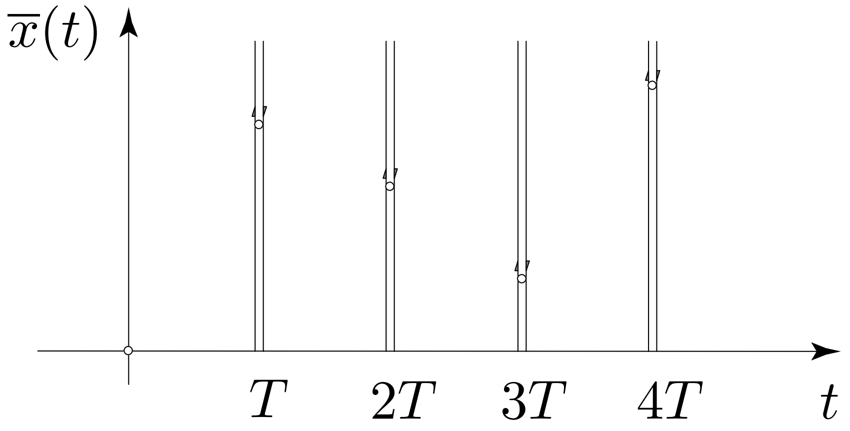
Sample-and-hold devices are easy to build but not as easy to analyze. A mathematical abstraction is therefore preferred here. The key idea is the concept of "impulse sampling." This mechanism is a mathematical concept and not what occurs in reality.

The impulse sampling approach is an approximation of the behavior of a real sample-and-hold element. For decreasing Δ and τ, the behavior of the real sample-and-hold element approaches the ideal impulse sampling characteristics.

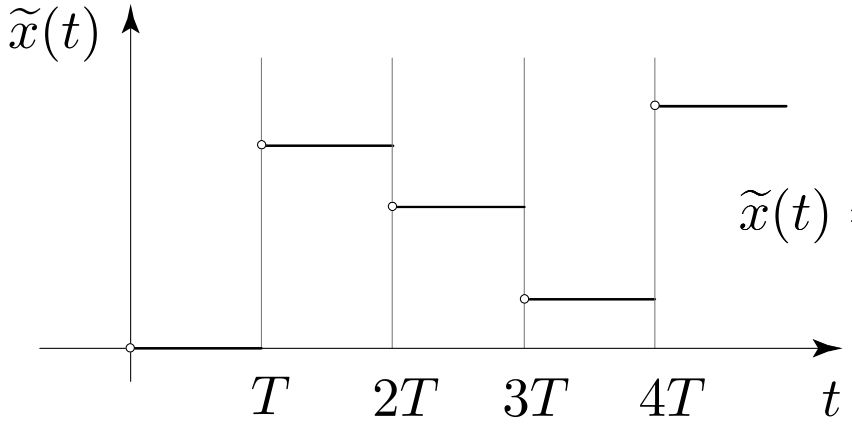
The variables analyzed below are the spectra of the input signal $x(t)$ and of the output $\widetilde{x}(t)$. PM: the spectrum of a signal indicates what frequencies are present in it when a Fourier decomposition is used. The spectrum quantifies the magnitude. If the signal is periodic, its spectrum is discrete; otherwise it is continuous.



$$x(t) = 0,\ \forall\ t < 0$$



$$\bar{x}(t) = \sum_{k=0}^{\infty} x(kT) \cdot \delta(t - kT)$$



$$\widetilde{x}(t) = \sum_{k=0}^{\infty} x(kT) \cdot [h(t - kT) - h(t - (k+1)T)]$$

Analyze spectral properties of $\widetilde{x}$.

Start with

$$\bar{x}(t) = \sum_{k=0}^{\infty} x(k \cdot T) \cdot \delta(t - k \cdot T) = x(t) \cdot \sum_{k=-\infty}^{\infty} \delta(t - k \cdot T) \tag{45}$$

Linearity! Notice summation is now from $-\infty$ to $+\infty$, no problem since $x(t) = 0$ for all times $t < 0$.

Second part periodic, therefore Fourier series

$$\sum_{k=-\infty}^{\infty} \delta(t - k \cdot T) = \sum_{n=-\infty}^{\infty} c_n \cdot e^{j \cdot \frac{2\pi n}{T} \cdot t} \tag{46}$$

The coefficients c_n are found using the usual integral transformation

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} \sum_{k=-\infty}^{\infty} \delta(t - k \cdot T) \cdot e^{-j \cdot \frac{2\pi n}{T} \cdot t} \cdot dt \tag{47}$$

In $-T/2 \leq t \leq T/2$ $\delta(t - k \cdot T)$ not zero only for $k = 0$ therefore

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} \delta(t) \cdot e^{-j \cdot \frac{2\pi n}{T} \cdot t} \cdot dt = \frac{1}{T} \tag{48}$$

Rewrite equation (45)

$$\bar{x}(t) = x(t) \cdot \frac{1}{T} \cdot \sum_{n=-\infty}^{\infty} e^{j \cdot \frac{2\pi n}{T} \cdot t} \tag{49}$$

$x(t)$ not periodic, therefore use Laplace transformation

$$\bar{X}(s) = \int_{-\infty}^{\infty} \bar{x}(t) \cdot e^{-s \cdot t} \cdot dt \tag{50}$$

Inserting equation (49) yields

$$\bar{X}(s) = \int_{-\infty}^{\infty} x(t) \cdot \frac{1}{T} \cdot \sum_{n=-\infty}^{\infty} e^{j \cdot \frac{2\pi n}{T} \cdot t} \cdot e^{-s \cdot t} \cdot dt \tag{51}$$

Interchange order of integration and summation and rearrange terms

$$\bar{X}(s) = \frac{1}{T} \cdot \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} x(t) \cdot e^{-(s-j\cdot\frac{2\pi n}{T})\cdot t} \cdot dt = \frac{1}{T} \cdot \sum_{n=-\infty}^{\infty} X(s-j\cdot\frac{2\pi n}{T}) \tag{52}$$

is obtained.

Result: The integral is simply the Laplace transformation of the signal $x(t)$ with the "shifted" Laplace variable $\sigma = s - j \cdot \frac{2\pi n}{T}$. Accordingly, the spectrum of $\bar{x}(t)$ will be obtained by taking the spectrum of the input signal $x(t)$ and adding this to the infinitely many copies obtained by shifting the original spectrum by the frequencies $2\pi n/T$.

Example: First-Order System Impulse Response

The signal $x(t)$ is assumed to be the impulse response of a first-order low-pass element

$$P(s) = \frac{1}{\tau \cdot s + 1} \tag{53}$$

The Laplace transformation of $x(t)$ is of course equal to the system's transfer function. For the sake of simplicity, only the original and the first two copies of the signal are analyzed below

$$\bar{X}(s) \approx \frac{1}{T} \cdot \left[\frac{1}{\tau \cdot s + 1} + \frac{1}{\tau \cdot (s - j \cdot \frac{2\pi}{T}) + 1} + \frac{1}{\tau \cdot (s + j \cdot \frac{2\pi}{T}) + 1} \right] \tag{54}$$

The amplitude spectra of these signals for $T = \tau/3$ are shown in Figure 2 using a linear scale for the frequency. The spectrum of $\bar{X}(s)$ is distorted when compared to the spectrum of $X(s)$. This "aliasing" is caused by the sampling process.

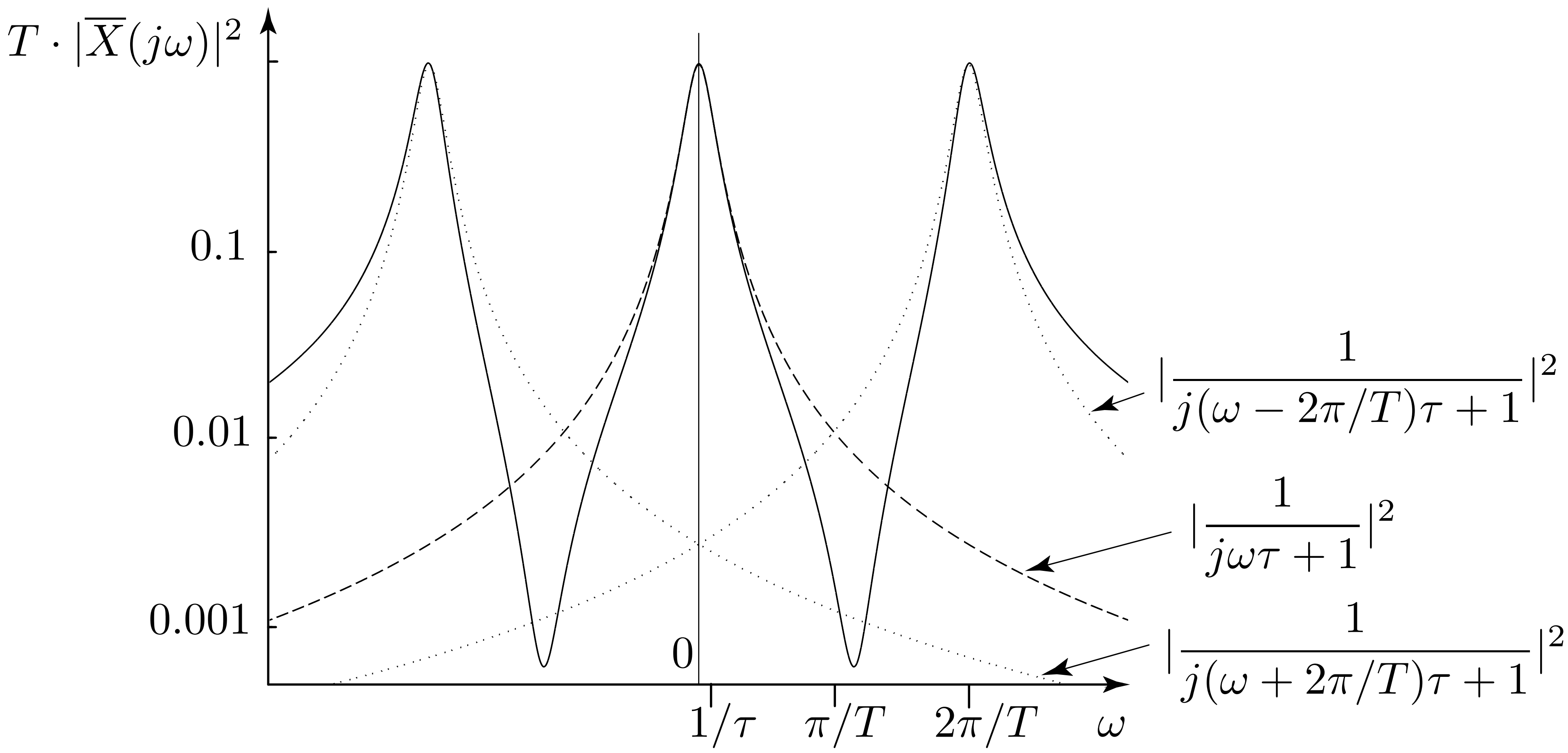


Figure 2: Spectrum of the signals $x(t)$ and $\widetilde{x}(t)$ (approximated).

The output signal of the sample-and-hold device is

$$\widetilde{X}(s) = \frac{1 - e^{-T \cdot s}}{s} \cdot \bar{X}(s) = \frac{1 - e^{-T \cdot s}}{s} \cdot \frac{1}{T} \cdot \sum_{n=-\infty}^{\infty} X(s - j \cdot \frac{2\pi n}{T}) \quad (55)$$

The $ZOH(s)$ element is analyzed now, its DC gain is equal to T

$$\lim_{s \to 0} ZOH(s) = \lim_{s \to 0} \frac{T \cdot e^{-s \cdot T}}{1} = T \quad (56)$$

The Bode diagram is shown in Figure 3.

Notice: in Simulink the complete sample-and-hold element – which is not realized as discussed here by impulse sampling – is designated as “ZOH,” its gain is then of course equal to one.

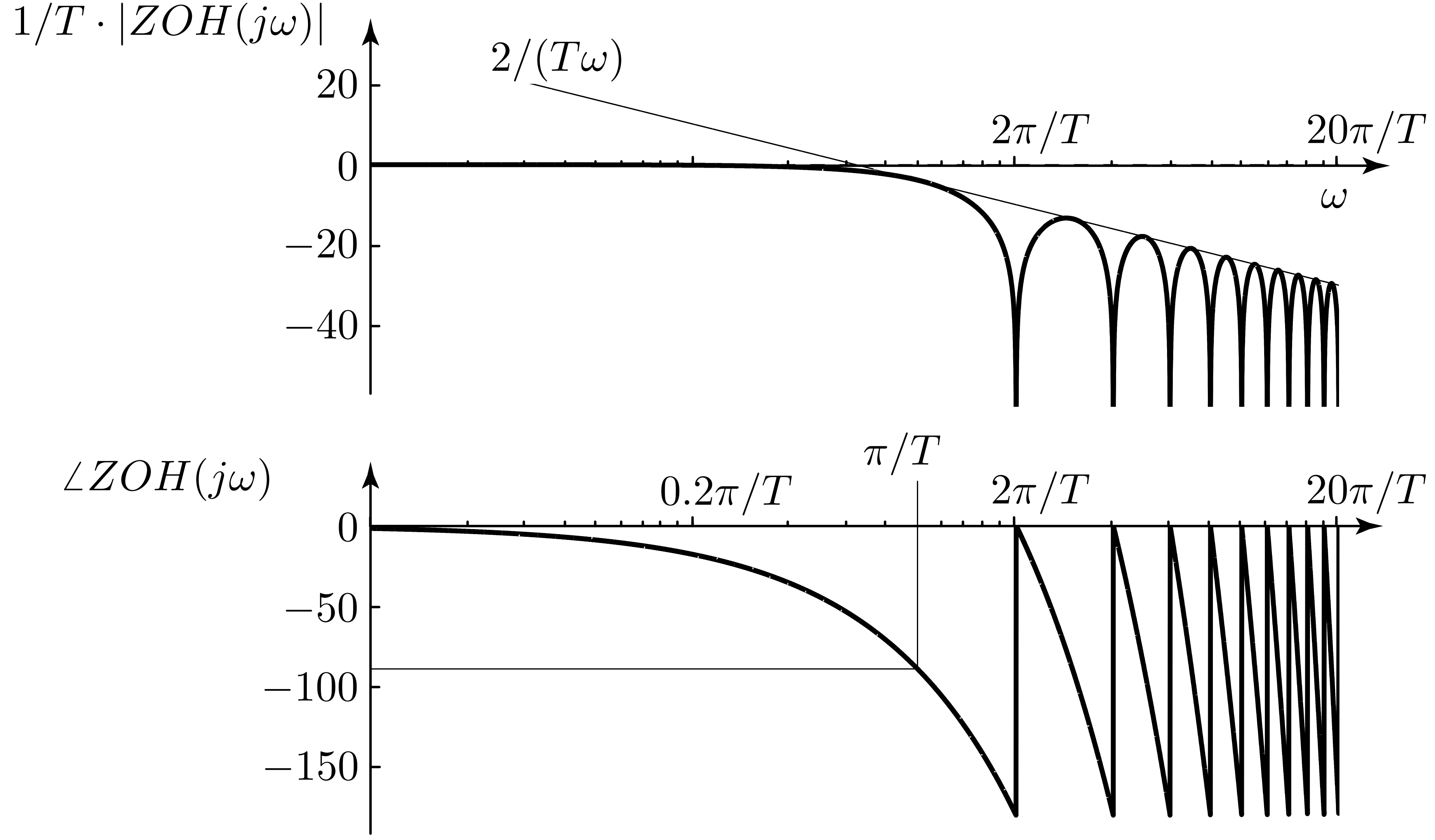


Figure 3: Bode diagram of $ZOH(s)$

$ZOH(s)$ is a low-pass with complex high-frequency behavior: it has infinitely many zeros at $\omega = n \cdot 2 \cdot \pi/T$, its local maxima are bounded

In summary, the following interpretation of the sample-and-hold operation is obtained:

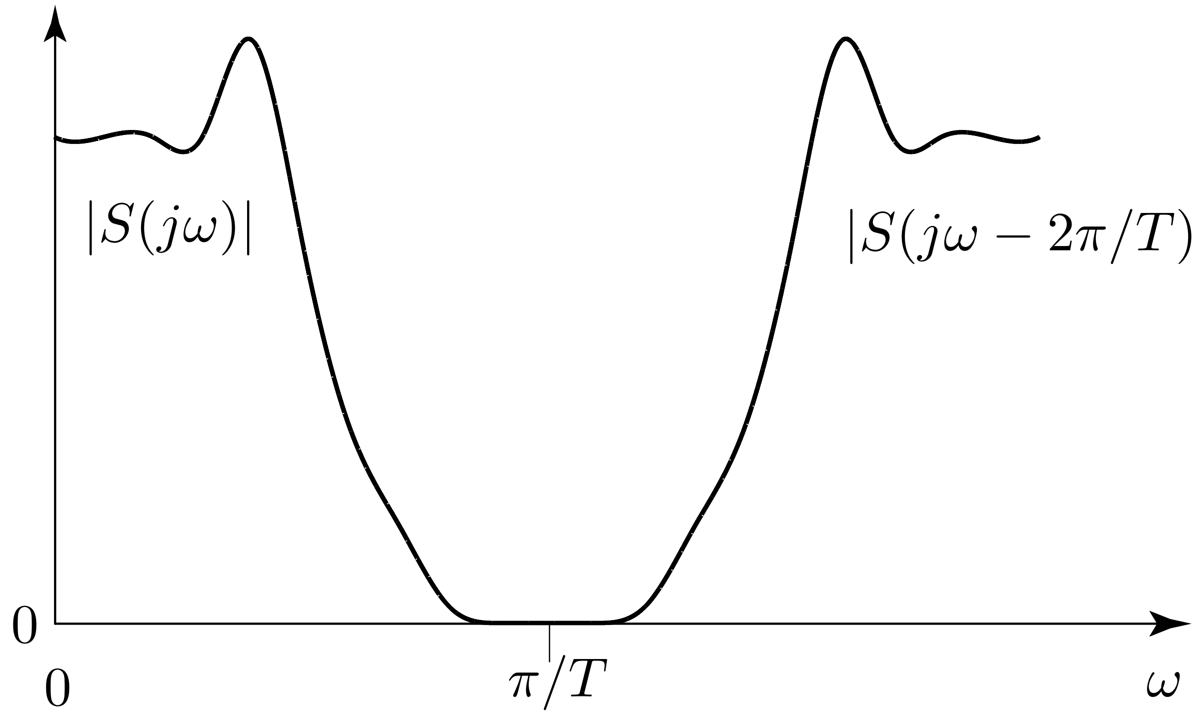
- The sampling can be interpreted as the multiplication with a sequence of Dirac functions, each weighted by the value of the signal $x(t)$ at the sampling times $t = k \cdot T$.
- The zero-order hold can be interpreted as an integration of the difference between that signal and a version delayed by one sampling interval, i.e.

$$ZOH(s) = \frac{1 - e^{-s \cdot T}}{s} \tag{57}$$

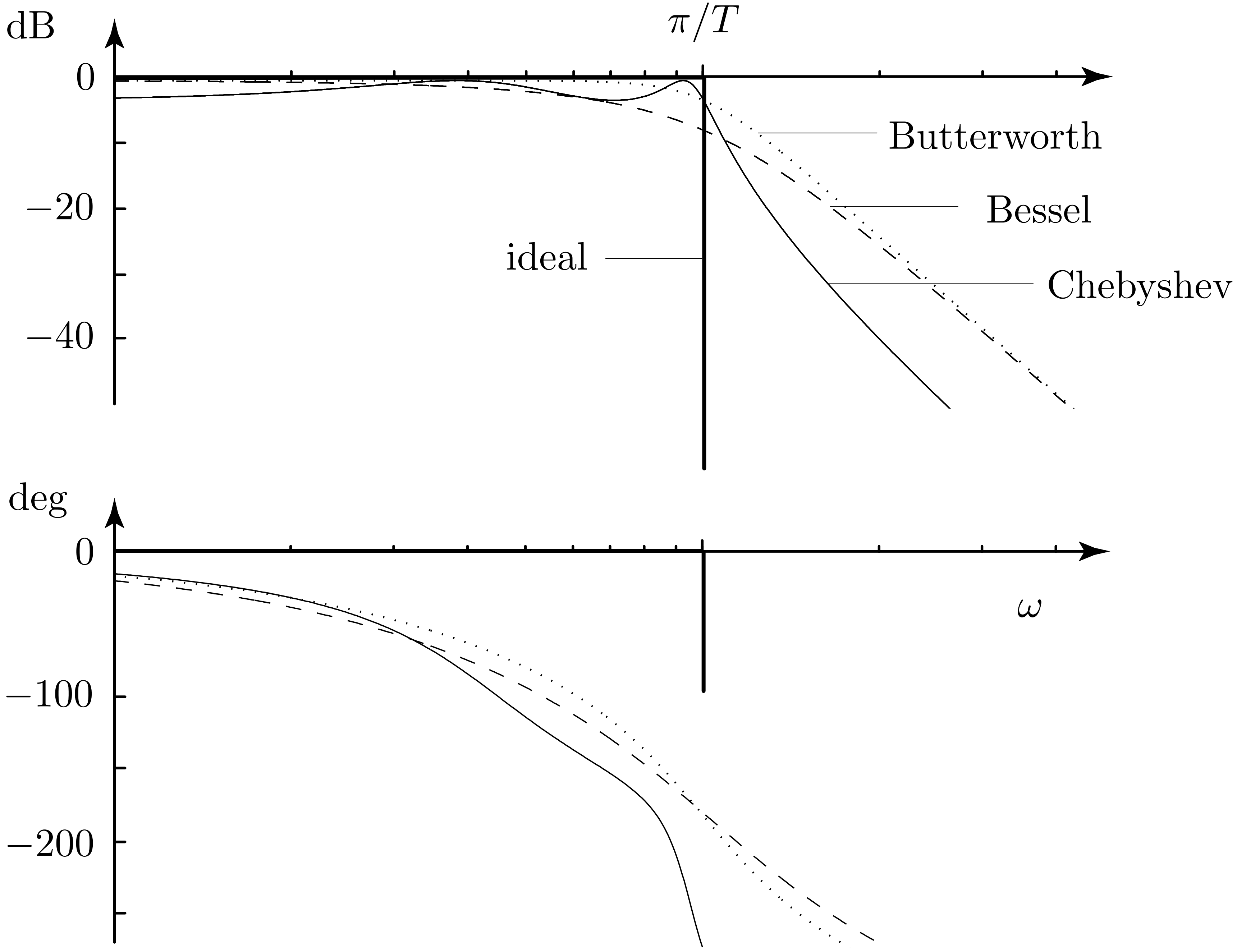
The spectrum of the signal $\widetilde{x}(t)$ is influenced by both effects, i.e., the spectrum of the original input signal $x(t)$ is aliased by the sampling process yielding a distorted spectrum to which the effect of the zero-order hold element is added.

Anti-Aliasing Filters and “Perfect” Reconstruction

For sampled signals, the aliasing effect is unavoidable, i.e., for frequencies $\omega > \omega_N = \pi/T$ fictitious contributions always appear in the spectrum of the sampled signal. However, if the original signal spectrum has no contributions for $\omega \geq \omega_N$, such a distortion may be avoided for frequencies $\omega < \omega_N$.



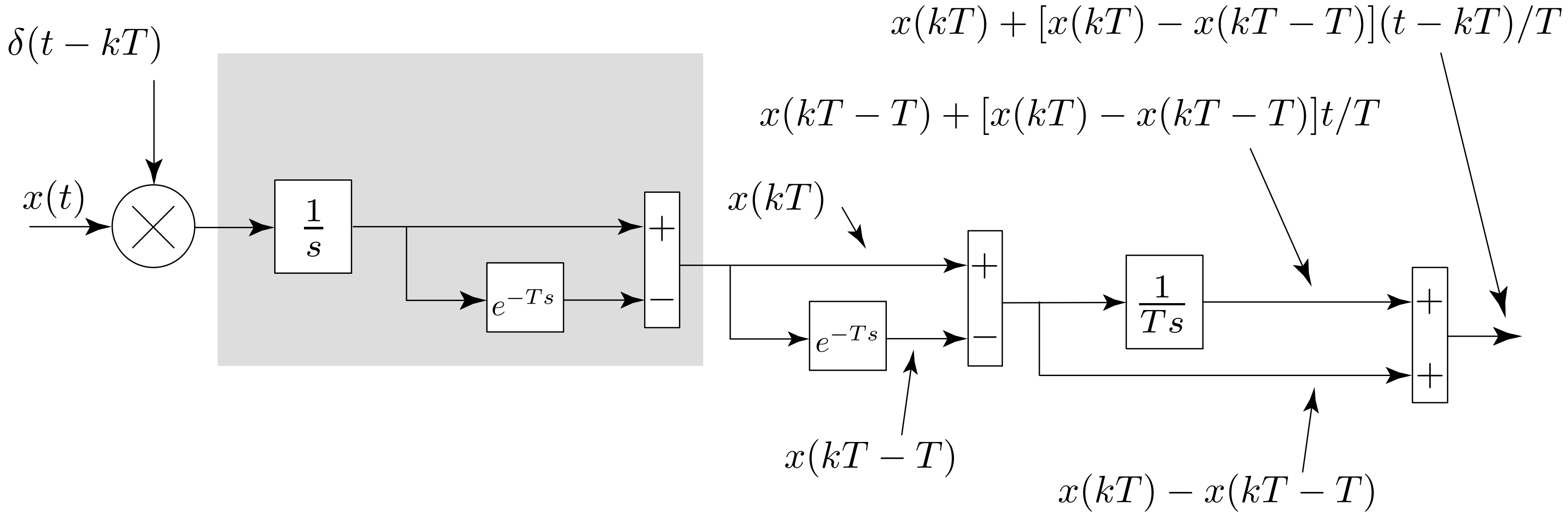
Of course, real signals never satisfy this condition. Therefore, "Anti-Aliasing Filters" (AAF) are usually employed, which have to be placed *before* the sampling element. The next figure shows the Bode diagram of an ideal and of three real AAF. Real AAF cannot cut off all frequency contributions for $\omega > \omega_N$ and introduce unwanted phase lags at frequencies well below the Nyquist frequency. For these reasons, they have to be included in the controller design step by adding their dynamics to the plant dynamics.


dB
π/T
0
−20
−40
Butterworth
Bessel
ideal
Chebyshev
deg
0
ω
−100
−200

Only zero-order hold elements have been discussed here so far.
First-order hold element

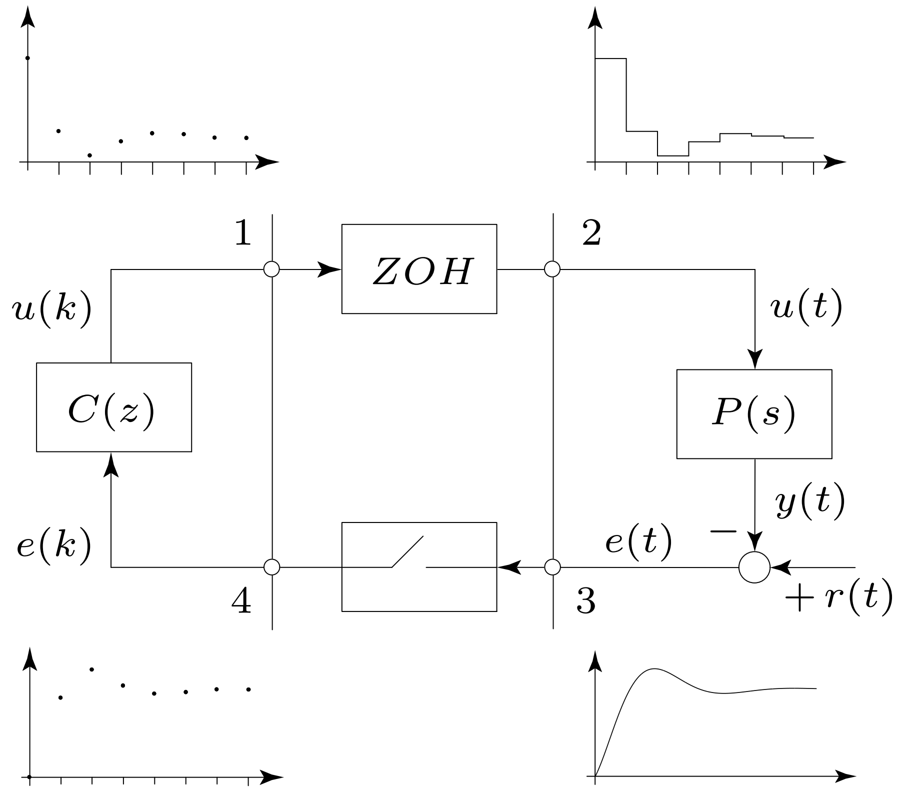
$$FOH(s) = \frac{\left(1 - e^{-T\cdot s}\right)^2}{T \cdot s^2} \cdot (T \cdot s + 1) \tag{58}$$

The next figure shows the values of the signals inside the FOH during the time interval $t \in [k \cdot T, k \cdot T + T)$.



4 Lecture — Discrete-Time System Analysis

In this chapter the second approach to discrete-time system analysis is presented, i.e., an inherently discrete-time approach is shown which starts by discretizing the continuous-time part of the system shown below.



First a time domain approach is shown, which relies on the solution of a linear system to piecewise constant inputs. Then a frequency domain analysis is outlined. The cornerstone of that part is the $\mathcal{Z}$ transformation.With these tools the main analysis problems can be tackled.

Plant Discretization

The starting point of this section is the prediction of the state x and of the output y of a linear system $\{A, B, C, D\}$ when its initial condition $x(0)$ and the input signal u are known. Matrix exponentials

lie at the core of this problem

$$e^{A\cdot t} = I + \frac{1}{1!}A\cdot t + \frac{1}{2!}(A\cdot t)^2 + \frac{1}{3!}(A\cdot t)^3 + \cdots + \frac{1}{n!}(A\cdot t)^n + \cdots \quad (59)$$

where $A \in \mathbb{R}^{n\times n}$ is a real square matrix. The matrix exponential satisfies

$$\frac{d}{dt}e^{A\cdot t} = A\cdot e^{A\cdot t} = e^{A\cdot t}\cdot A \quad (60)$$

where eq. (60) emphasizes that e^{At} and A commute. Since $A\cdot t$ and $A\cdot\tau$ commute $\forall\ t, \tau \in \mathbb{R}$ the following equation holds true

$$(e^{A\cdot t})^{-1} = e^{-A\cdot t} \quad (61)$$

Recursive System Equations

Solution to the initial value problem:

$$x(t) = e^{A\cdot t} \cdot x(0) + \int_0^t e^{A\cdot(t-\tau)} \cdot B \cdot u(\tau)\, d\tau \tag{62}$$

and

$$y(t) = C \cdot e^{A\cdot t} \cdot x(0) + \int_0^t C \cdot e^{A\cdot(t-\tau)} \cdot B \cdot u(\tau)\, d\tau + D \cdot u(t) \tag{63}$$

If an intermediate solution $x(t_1)$ has been computed, this point can be used to restart the solution procedure

$$x(t) = e^{A\cdot(t-t_1)} \cdot x(t_1) + \int_{t_1}^t e^{A\cdot(t-\tau)} \cdot B \cdot u(\tau)\, d\tau \tag{64}$$

In the discrete-time setting discussed here, the input $u(t)$ is piecewise constant, i.e.,

$$u(t) = u(k \cdot T), \qquad \forall t \in [k \cdot T, k \cdot T + T) \tag{65}$$

For that reason, the equation (64) can be further simplified to be

$$x(t) = e^{A\cdot(t-k\cdot T)}\cdot x(k\cdot T)+\int_{k\cdot T}^{t} e^{A\cdot(t-\tau)}\,d\tau\cdot B\cdot u(k\cdot T), \quad \forall t \in [k\cdot T, k\cdot T+T) \tag{66}$$

Choosing t to be equal to the next sampling point $k \cdot T + T$ yields

$$x(k\cdot T+T) = e^{A\cdot T}\cdot x(k\cdot T)+\int_{k\cdot T}^{k\cdot T+T} e^{A\cdot(k\cdot T+T-\tau)}\,d\tau\cdot B\cdot u(k\cdot T) \tag{67}$$

Finally, using the substitution $\sigma = k \cdot T + T - \tau$ in the integral part, this expression is simplified to be

$$x(k \cdot T + T) = e^{A\cdot T} \cdot x(k \cdot T) + \int_0^T e^{A\cdot\sigma}\,d\sigma \cdot B \cdot u(k \cdot T) \tag{68}$$

This equation permits to adopt a purely discrete-time description of the behavior of the plant

$$x(k+1) = F \cdot x(k) + G \cdot u(k), \quad y(k) = C \cdot x(k) + D \cdot u(k) \tag{69}$$

with the obvious definitions

$$F = e^{A \cdot T}, \qquad G = \int_0^T e^{A \cdot \sigma} \, d\sigma \cdot B \tag{70}$$

For the case $\det(A) \neq 0$, i.e., when A has no eigenvalues in the origin, the integral part can be solved explicitly using equation (60)

$$G = A^{-1} \cdot \left(e^{A \cdot T} - I\right) \cdot B \tag{71}$$

System Stability

The system

$$x(k+1) = F \cdot x(k), \quad x(0) \neq 0 \tag{72}$$

will be classified here as:

- Asymptotically stable if $\lim_{k \to \infty} ||x_k|| = 0$
- Stable if $\lim_{k \to \infty} ||x_k|| < \infty$
- Unstable if neither of the two former conditions are satisfied.

Since the system

$$x(k+1) = F \cdot x(k) + G \cdot u(k), \quad y(k) = C \cdot x(k) + D \cdot u(k) \tag{73}$$

is linear and time-invariant, the two concepts of *BIBO stability* (bounded input bounded output) and *asymptotic* stability, as defined above, coincide if the system is completely observable and controllable.

Spectral Methods

The stability of the system is determined by the properties of the matrix F only. In general, F is similar to a Jordan matrix, i.e., a coordinate transformation $x = Q \cdot z$ exists which transforms F into an “almost diagonal matrix” J_F

$$J_F = Q^{-1} \cdot F \cdot Q \tag{74}$$

where some elements immediately above the main diagonal may be equal to one.

If the eigenvalues λ_i satisfy the condition $\lambda_i \neq \lambda_j$ for all $i \neq j$, then the matrix J_F is guaranteed to be diagonal. If multiple eigenvalues occur, then the situation is more complex and the exact form of J_F depends on the rank loss associated to each multiple eigenvalue.

The main point is, however, that arbitrary powers of J_F will remain triangular, the elements of J_F^k being powers of the scalars λ_i up to λ_i^k.

Example: J_F with one eigenvalue λ_1 with multiplicity 4 and rank loss 2

$$J_F = \begin{bmatrix} \lambda_1 & 1 & 0 & 0 & 0 \\ 0 & \lambda_1 & 1 & 0 & 0 \\ 0 & 0 & \lambda_1 & 0 & 0 \\ 0 & 0 & 0 & \lambda_1 & 0 \\ 0 & 0 & 0 & 0 & \lambda_2 \end{bmatrix} \tag{75}$$

and hence

$$J_F^k = \begin{bmatrix} \lambda_1^k & k \cdot \lambda_1^{k-1} & \frac{k\cdot(k-1)}{2} \cdot \lambda_1^{k-2} & 0 & 0 \\ 0 & \lambda_1^k & k \cdot \lambda_1^{k-1} & 0 & 0 \\ 0 & 0 & \lambda_1^k & 0 & 0 \\ 0 & 0 & 0 & \lambda_1^k & 0 \\ 0 & 0 & 0 & 0 & \lambda_2^k \end{bmatrix} \tag{76}$$

The general solution of the homogeneous ($u(k) = 0 \ \forall k$) system (73) is given by

$$x(k) = F^k \cdot x(0), \quad k > 0 \tag{77}$$

The matrix

$$\begin{aligned} F^k &= (Q \cdot J_F \cdot Q^{-1})^k \\ &= Q \cdot J_F \cdot Q^{-1} \cdot Q \cdot J_F \cdot Q^{-1} \ldots Q \cdot J_F \cdot Q^{-1} \\ &= Q \cdot J_F^k \cdot Q^{-1} \end{aligned} \tag{78}$$

converges to the zero matrix if and only if all its eigenvalues λ_i satisfy the condition $|\lambda_i| < 1$ ($\lim_{k\to\infty} k^m$ "grows slower than" $\lim_{k\to\infty} \lambda^{k-n}$ decreases for any finite m, n).

The system is unstable if one of the eigenvalues of F has a modulus larger than one. The intermediate case, where some $|\lambda_j| = 1$, may be stable or unstable, depending on the details of the Jordan structure. If one of the multiple eigenvalues with $|\lambda_i| = 1$ has a rank loss smaller than its multiplicity (i.e., the Jordan block contains some elements 1), then the system will be unstable. If the rank loss of all multiple eigenvalues with $|\lambda_i| = 1$ is equal to the corresponding multiplicity (i.e., J_F is diagonal) then the system is stable.

Note that in general the system

$$x(k+1) = F \cdot x(k) + G \cdot u(k), \quad y(k) = C \cdot x(k) + D \cdot u(k) \tag{79}$$

represents the linearization around an equilibrium of a nonlinear system. In this case, the "critical" situation (some $|\lambda_j| = 1$) yields no information about the stability properties of that equilibrium. The "higher-order" terms of the original nonlinear system description will be decisive in this case.

Remarks

For the explicit computation of the matrix exponential F dedicated algorithms are known such that even large-scale problems can be solved efficiently and reliably.

Some caution is necessary when interconnected systems have to be analyzed. A typical situation is the series connection of two dynamic systems

$$\begin{aligned} \dot{x}(t) &= A \cdot x(t) + B \cdot u(t) \quad & y(t) &= C \cdot x(t) \\ \dot{z}(t) &= K \cdot z(t) + L \cdot y(t) \quad & v(t) &= M \cdot z(t) \end{aligned} \tag{80}$$

For this example the two operations "interconnection" and "discretization" do not commute, i.e. the matrices R and S defined by

$$R = e^{Q\cdot T}, \quad Q = \begin{bmatrix} A & 0 \\ L\cdot C & K \end{bmatrix} \tag{81}$$

and

$$S = \begin{bmatrix} e^{A\cdot T} & 0 \\ K^{-1}\cdot(e^{K\cdot T} - I)\cdot L\cdot C & e^{K\cdot T} \end{bmatrix} \tag{82}$$

are not equal in their the lower left-hand block.

Therefore, the specific location of the various ZOH and sampling elements has to be taken into account explicitly.

The plant has been characterized using an *internal* description, and matrices $\{F, G, C, D\}$ are unique, provided that the associated state variables have a physical interpretation.

If only the input/output behavior of the plant is to be analyzed, this choice of matrices is no longer unique. Any set of matrices that satisfy the similarity conditions

$$\widetilde{F} = T^{-1} \cdot F \cdot T, \qquad \widetilde{G} = T^{-1} \cdot G, \qquad \widetilde{C} = C \cdot T, \qquad \widetilde{D} = D \tag{83}$$

with $T \in \mathbb{R}^{n \times n}$ and $\det(T) \neq 0$ will have the same output y for a specific input u.

If the system is SISO, its input/output behavior can be described using an n-th order difference equation

$$y(k+n)+a_{n-1}\cdot y(k+n-1)\ldots+a_0\cdot y(k) = b_n\cdot u(k+n)+\ldots+b_0\cdot u(k) \tag{84}$$

It will be shown in the next section that the coefficients a_i and b_i are the coefficients of the denominator and numerator polynomials of the rational function

$$P(z) = c \cdot (z \cdot I - F)^{-1} \cdot g + d \tag{85}$$

This input/output description is the starting point of all system identification methods which, by exciting an unknown system with a signal u and recording the system's response y, identify the input/output behavior of the plant. Of course, equations (84) and (85) represent only the controllable *and* the observable parts of the corresponding system.

$\mathcal{Z}$ Transformation

Definition of the $\mathcal{Z}$ Transformation

The $\mathcal{Z}$ transformation acts on discrete-time signals x_k with

$$x_k \in \mathbb{R}, \qquad k = -\infty, \ldots, -1, 0, 1, \ldots, \infty \tag{86}$$

Below it will be always assumed that $x_k = 0$ for all $k < 0$.

The $\mathcal{Z}$ transformation is defined by

$$X(z) = \mathcal{Z}\{x_k\} = \sum_{k=0}^{k=\infty} x_k \cdot z^{-k} \tag{87}$$

where $z \in \mathbb{C}$ is the new independent variable.

- The $\mathcal{Z}$ transformation is a linear operator, i.e., for the signals x and y

$$\mathcal{Z}\{a \cdot x + b \cdot y\} = a \cdot \mathcal{Z}\{x\} + b \cdot \mathcal{Z}\{y\}, \qquad a, b \in \mathbb{R} \tag{88}$$

- The $\mathcal{Z}$ transformation of a signal x satisfies the shift property

$$\mathcal{Z}\{x_{.+1}\} = z \cdot \mathcal{Z}\{x\} - z \cdot x_0 \tag{89}$$

  (the signal $x_{.+1}$ is obtained from x by shifting each value x_k one step forward $x_k \to x_{k+1}$)

- The limit behavior of a signal x satisfies the equality

$$\lim_{k\to\infty} x_k = \lim_{z\to 1} (z-1) \cdot X(z) \tag{90}$$

- Convolution in the time domain is equivalent to multiplication in the frequency domain

$$\mathcal{Z}\{\sum_{j=0}^{k} x_j \cdot y_{k-j}\} = X(z) \cdot Y(z) \tag{91}$$

$\mathcal{Z}$ Transformation Table

$X(s)$	$\leftarrow x(t) \quad \vert \quad x(k \cdot T) \rightarrow$	$X(z)$
$\frac{1}{s}$	$h(t)$	$\frac{z}{z-1}$
$\frac{1}{s^2}$	$t\,h(t)$	$\frac{Tz}{(z-1)^2}$
$\frac{1}{s^3}$	$\frac{1}{2}t^2h(t)$	$\frac{T^2z(z+1)}{2(z-1)^3}$
$\frac{1}{s+a}$	$e^{-at}h(t)$	$\frac{z}{z-e^{-aT}}$
$\frac{1}{(s+a)^2}$	$te^{-at}h(t)$	$\frac{Tze^{-aT}}{(z-e^{-aT})^2}$
$\frac{a}{s(s+a)}$	$[1-e^{-at}]\,h(t)$	$\frac{z(1-e^{-aT})}{(z-1)(z-e^{-aT})}$

$X(s)$	$\leftarrow x(t) \quad \vert \quad x(k\cdot T) \rightarrow$	$X(z)$
$\frac{a}{s^2+a^2}$	$\sin(at)h(t$	$\frac{z\sin(aT)}{z^2-2\cos(aT)z+1}$
$\frac{s}{s^2+a^2}$	$\cos(at)h(t)$	$\frac{z(z-\cos(aT))}{z^2-2\cos(aT)z+1}$
$\frac{b}{(s+a)^2+b^2}$	$e^{-at}\sin(bt)h(t)$	$\frac{zr\sin(bT))}{z^2-z2r\cos(bT)+r^2},\ r=e^{-a\cdot T}$
$\frac{s+a}{(s+a)^2+b^2}$	$e^{-at}\cos(bt)h(t)$	$\frac{z^2-zr\cos(b\cdot T)}{z^2-z2r\cos(bT)+r^2},\ r=e^{-a\cdot T}$

4.1 Transfer Functions

Using the shift property introduced above, the $\mathcal{Z}$ transformation can be used to "solve" the difference equation (69)

$$x(k+1) = F \cdot x(k) + G \cdot u(k) \qquad \Rightarrow [z \cdot I - F] \cdot X(z) = G \cdot U(z) + z \cdot x_0 \tag{92}$$

and from that the discrete-time transfer function of the plant is found to be

$$Y(z) = P(z) \cdot U(z) = \left[C \cdot (z \cdot I - F)^{-1} \cdot G + D\right] \cdot U(z) \tag{93}$$

If the system has only one input and one output (SISO system), the transfer function (93) will be a scalar rational function of z. This can be seen if Cramer's rule is applied, i.e.,

$$M^{-1} = \mathrm{Adj}(M) / \det(M) \tag{94}$$

where $\det(M)$ is the determinant of M and $\mathrm{Adj}(M)$ is its adjoint matrix. Recall that the elements of $\mathrm{Adj}(M)$ are the subdeterminants of M (the principal minors).

Using this rule the transfer function (93) can be written as

$$P(z) = c \cdot (z \cdot I - F)^{-1} \cdot g + d = \frac{c \cdot \mathrm{Adj}(z \cdot I - F) \cdot g + d \cdot \det(z \cdot I - F)}{\det(z \cdot I - F)} \tag{95}$$

Since the determinant and the subdeterminants of M are polynomial functions of z, the transfer function $P(z)$ is indeed a rational function of z. If the system has multiple inputs and outputs (MIMO), one transfer function (95) will be obtained for each IO channel. The discussion of MIMO systems using IO techniques is mathematically more complicated and beyond the scope of this text. MIMO systems will therefore be treated here using the time domain approaches only.

Another connection to a result introduced above is obtained by transforming the rational function (95) back into the time domain using the usual correspondence

$$z \cdot X(z) \to x(k+1) \tag{96}$$

This yields directly the IO description (84) introduced above.

The transfer function (95) has been derived using a state-space "detour," which is actually what is preferred in numerical solutions. However, for theoretical considerations or for simple systems, for which the solutions have been precomputed and tabulated, there exists an alternative, direct approach. The starting point is a continuous-time plant $P(s)$ which is inserted in the usual sample and zero-order hold configuration. The resulting discrete-time transfer function can be found using an inverse Laplace transformation as follows

$$P(z) = (1 - z^{-1}) \cdot \mathcal{Z}\left\{ \mathcal{L}^{-1}\left[\frac{1}{s} \cdot P(s)\right]_{t=k\cdot T} \right\} \tag{97}$$

In order to derive this expression, the transfer function of the ZOH element

$$ZOH(s) = \frac{1 - e^{-s \cdot T}}{s} \tag{98}$$

and the shift property of the $\mathcal{Z}$ transformation

$$e^{-s \cdot T} \cdot x(t) \rightarrow z \cdot X(z) \tag{99}$$

have been used. Similar expressions can be found for higher-order holds.

The general form of a discrete-time SISO transfer function is given by

$$P(z) = Y(z)/U(z) = \frac{b_n \cdot z^n + b_{n-1} \cdot z^{n-1} \ldots + b_1 \cdot z + b_0}{z^n + a_{n-1} \cdot z^{n-1} \ldots + a_1 \cdot z + a_0} \tag{100}$$

It is easy to show that $b_n = d$, i.e., that the transfer function (100) is strictly proper iff the plant has no feed-through term.

The inverse $\mathcal{Z}$ transformation yields the difference equation (84). In practice, only the backward shift operator z^{-1} may be used. Multiplying both the numerator and the denominator of (100) by z^{-n} yields

$$P(z) = \frac{b_n + b_{n-1} \cdot z^{-1} \ldots + b_1 \cdot z^{n-1} + b_0 \cdot z^{-n}}{1 + a_{n-1} \cdot z^{-1} \ldots + a_1 \cdot z^{n-1} + a_0 \cdot z^{-n}} \tag{101}$$

Transforming that expression back into a discrete-time equation, a realizable plant model is obtained

$$y(k) = -a_{n-1} \cdot y(k-1) \ldots - a_0 \cdot y(k-n) + b_n \cdot u(k) + b_{n-1} \cdot u(n-1) \ldots + b_0 \cdot u(k-n) \tag{102}$$

As in continuous-time systems, the transfer function (100) may be factorized into linear terms

$$P(z) = k \cdot \frac{(z-\zeta_1)\cdot(z-\zeta_2)\ldots(z-\zeta_n)}{(z-\pi_1)\cdot(z-\pi_2)\ldots(z-\pi_n)} \tag{103}$$

where the complex numbers ζ_i are defined as the zeros and the π_i as the poles of the plant $P(z)$.

In signal processing applications such a system is identified as an Infinite Impulse Response (IIR) filter. The other popular systems are Finite Impulse Response filters (FIR) which are not defined recursively and, therefore, have no non-zero poles.

Notice that it is easier to identify the plant's zeros using the description (100) than (101).

Transfer Function

The general form of a discrete-time SISO transfer function is given by

$$P(z) = Y(z)/U(z) = \frac{b_n \cdot z^n + b_{n-1} \cdot z^{n-1} \ldots + b_1 \cdot z + b_0}{z^n + a_{n-1} \cdot z^{n-1} \ldots + a_1 \cdot z + a_0} \tag{104}$$

The transfer function is strictly proper with $b_n = 0$ iff the continuous-time plant has no feed-through term.
Even if the plant has a feed-through term, the computation delays added to the plant introduce one delay and therefore the discrete-time plant is strictly proper.

Delays $q \cdot T$ are often separated from the remaining part

$$P(z) = z^{-q} \cdot \frac{b_{n-1} \cdot z^{n-1} \ldots + b_1 \cdot z + b_0}{z^n + a_{n-1} \cdot z^{n-1} \ldots + a_1 \cdot z + a_0} \tag{105}$$

In general, $b_{n-1} \neq 0$ even if the continuous-time plant has less than $n-1$ finite zeros.

Discrete-Time Frequency Response

Same idea as in continuous time

$$u(k) = \cos(k\,\omega\,T) \tag{106}$$

will produce an output signal

$$y(k) = y_{tr}(k) + y_{st}(k) \tag{107}$$

$$\begin{aligned} y_{st}(k) &= m(\omega)\cdot\cos(k\,\omega\,T + \varphi(\omega)) \\ &= m(\omega)\cdot\cos(\varphi(\omega))\cdot\cos(k\,\omega\,T) - m(\omega)\cdot\sin(\varphi(\omega))\cdot\sin(k\,\omega\,T) \\ &= A(\omega)\cdot\cos(k\,\omega\,T) - B(\omega)\cdot\sin(k\,\omega\,T) \end{aligned} \tag{108}$$

$\mathcal{Z}$-transform of input

$$U(z) = \frac{z \cdot (z - \cos(\omega T))}{z^2 - 2 \cdot \cos(\omega T) \cdot z + 1} \tag{109}$$

Therefore

$$Y(z) = P(z) \cdot U(z) = P(z) \cdot \frac{z \cdot (z - \cos(\omega T))}{z^2 - 2 \cdot \cos(\omega T) \cdot z + 1} \tag{110}$$

Using a partial fraction expansion

$$Y(z) = \sum_{i=1}^{n} \frac{C_i}{z - z_i} + \frac{A(\omega) \cdot z \cdot (z - \cos(\omega T)) - B(\omega) \cdot z \cdot \sin(\omega T)}{z^2 - 2 \cdot \cos(\omega T) \cdot z + 1} \tag{111}$$

If the system is asymptotically stable, all of the terms $\mathcal{Z}^{-1}\{\frac{C_i}{z - z_i}\}$ will go to zero for $k \to \infty$.

To compute $A(.)$ and $B(.)$, evaluate expression at the poles $z = e^{\pm j\omega T}$

$$\begin{gathered}\lim_{z \to e^{j\omega T}} P(z) \cdot U(z) = \\ \lim_{z \to e^{j\omega T}} \tfrac{A(\omega)\cdot e^{j\omega T}\cdot(e^{j\omega T}-\cos(\omega T))-B(\omega)\cdot e^{j\omega T}\cdot\sin(\omega T)}{z^2-2\cdot\cos(\omega T)\cdot z+1}\end{gathered} \tag{112}$$

After elimination of the common denominator and using

$$\sin(\omega T) = -j \cdot (e^{j\omega T} - \cos(\omega T))) \tag{113}$$

the following equation is obtained

$$P(e^{j\omega T})\cdot e^{j\omega T}\cdot(e^{j\omega T}-\cos(\omega T)) = (A(\omega)+jB(\omega))\cdot e^{j\omega T}\cdot(e^{j\omega T}-\cos(\omega T)) \tag{114}$$

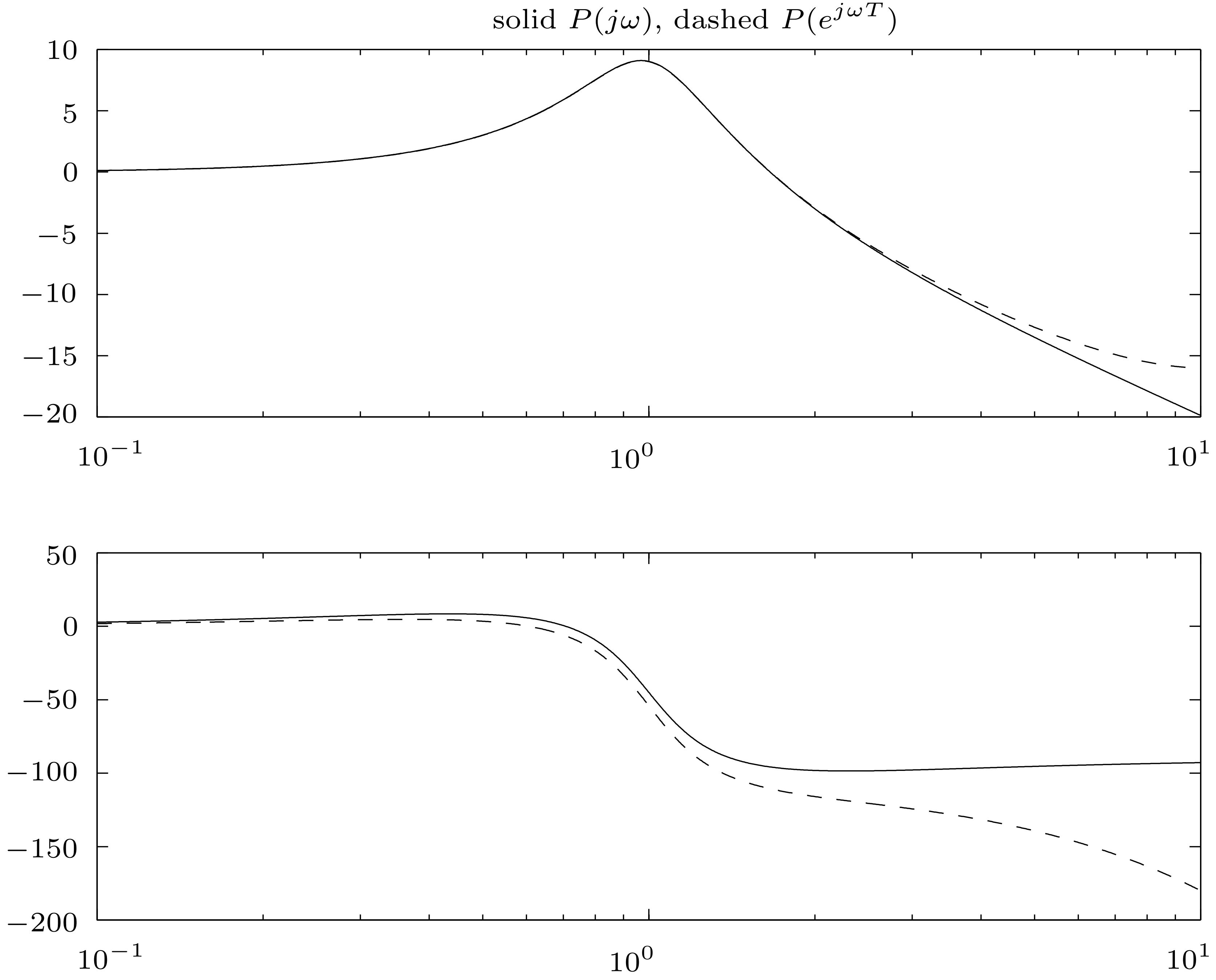
Therefore, the following relations are true

$$A(\omega) = Re\{P(e^{j\omega T})\}, \qquad B(\omega) = Im\{P(e^{j\omega T})\} \tag{115}$$

and

$$m(\omega) = |P(e^{j\omega T})|, \qquad \varphi(\omega) = \angle\{P(e^{j\omega T})\} \tag{116}$$

```
T=pi/10;
w=logspace(-1,1,1000);
nc=[1,1];
dc=[1,0.5,1];
Pc=polyval(nc,sqrt(-1)*w)./polyval(dc,sqrt(-1)*w);
magc=abs(Pc);
phic=180*unwrap(angle(Pc))/pi;
[nd,dd]=c2dm(nc,dc,T,'zoh');
Pd=polyval(nd,exp(sqrt(-1)*w*T))./polyval(dd,exp(sqrt(-1)*w*T));
magd=abs(Pd);
phid=180*unwrap(angle(Pd))/pi;
subplot(211)
semilogx(w,20*log10(magc),w,20*log10(magd));
subplot(212)
semilogx(w,phic,w,phid)
subplot(111)
```


solid $P(j\omega)$, dashed $P(e^{j\omega T})$
10
5
0
−5
−10
−15
−20
10^{-1}
10^{0}
10^{1}
50
0
−50
−100
−150
−200
10^{-1}
10^{0}
10^{1}

Numerical Solution

The solution of

$$x(k+1) = F \cdot x(k) + G \cdot u(k), \quad y(k) = C \cdot x(k) + D \cdot u(k) \tag{117}$$

can be found by inspection

$$\begin{aligned} x(1) &= F \cdot x(0) + G \cdot u(0) \\ x(2) &= F \cdot x(1) + G \cdot u(1) = F^2 \cdot x(0) + F \cdot G \cdot u(0) + G \cdot u(1) \\ x(3) &= \ldots \end{aligned} \tag{118}$$

In general

$$x(k) = F^k \cdot x(0) + \sum_{l=0}^{k-1} F^{k-1-l} \cdot G \cdot u(l) \tag{119}$$

$$y(k) = C \cdot \left[F^k \cdot x(0) + \sum_{l=0}^{k-1} F^{k-1-l} \cdot G \cdot u(l) \right] + D \cdot u(k) \tag{120}$$

The explicit solution can also be found using $\mathcal{Z}$ transformation techniques. Since the $\mathcal{Z}$ transformation of the discrete-time impulse

$$d(k) = \begin{cases} 1, & \text{for } k = 0 \\ \\ 0, & \text{else} \end{cases} \tag{121}$$

is simply $D(z) = 1$ the $\mathcal{Z}$ transform of the impulse response of the system (95) is given by

$$\Gamma(z) = c \cdot (z \cdot I - F)^{-1} \cdot g + d \tag{122}$$

The solution to arbitrary inputs u is then found using either inverse $\mathcal{Z}$ transformation techniques or convolution operations

$$y(k) = \sum_{l=0}^{\infty} \gamma(l) \cdot u(k-l) \tag{123}$$

where $\gamma(k)$ is the inverse $\mathcal{Z}$ transform of $\Gamma(z)$. As mentioned above, this approach is interesting for theoretical considerations involving simple input signals.

Transformation Rules for System Poles

A correspondence between the continuous-time original system

$$\dot{x}(t) = A \cdot x(t), \quad x(0) \neq 0 \tag{124}$$

and its sampled and ZOH discretized counterpart

$$x(k+1) = F \cdot x(k), \quad F = e^{A \cdot T} \tag{125}$$

can be derived if A is transformed into a Jordan matrix $A = P \cdot J_A \cdot P^{-1}$ by $x = P \cdot v$. Combining this with the definition of the matrix exponential one obtains

$$\begin{aligned} e^{A \cdot T} &= I + \tfrac{1}{1!} P J_A P^{-1} T + \tfrac{1}{2!} P J_A P^{-1} P J_A P^{-1} T^2 \\ &\quad + \tfrac{1}{3!} P J_A P^{-1} P J_A P^{-1} P J_A P^{-1} T^3 + \cdots \end{aligned} \tag{126}$$

and therefore

$$\begin{aligned} e^{A \cdot T} &= P P^{-1} + P J_A P^{-1} T + \tfrac{1}{2!} P J_A^2 P^{-1} T^2 + \tfrac{1}{3!} P J_A^3 P^{-1} T^3 + \cdots \\ &= P e^{J_A \cdot T} P^{-1} \end{aligned} \tag{127}$$

Since the eigenvalues are invariant under similarity transformations this proves that the eigenvalues λ_i^z of F and the eigenvalues λ_i^s of A satisfy the important relation

$$\lambda_i^z = e^{\lambda_i^s \cdot T} \tag{128}$$

which is sometimes written as $z = e^{s \cdot T}$ to emphasize the fact that this equation can be interpreted as a mapping between two complex planes (more details of this argument will be given below). Notice that the inverse mapping $s = \ln(z)/T$ is *not* unique. Only the region

$$\mathcal{C} = \{s \in \mathbb{C} \mid -\pi/T \leq \text{imag}(s) \leq \pi/T\} \tag{129}$$

is mapped one-to-one by $z = e^{s \cdot T}$ and its inverse function $s = \ln(z)/T$. This is of course simply a different formulation of the Shannon theorem.

With this correspondence at hand, it is now possible to define a "nice pole region" for discrete-time systems. The starting point is the characteristic equation of a second-order continuous-time system

$$p(s) = s^2 + 2 \cdot \delta \cdot \omega_0 \cdot s + \omega_0^2 \tag{130}$$

which is assumed to be representative of the dominant part of a high-order plant. The solutions of the equation $p(s) = 0$ are the poles of the system (124)

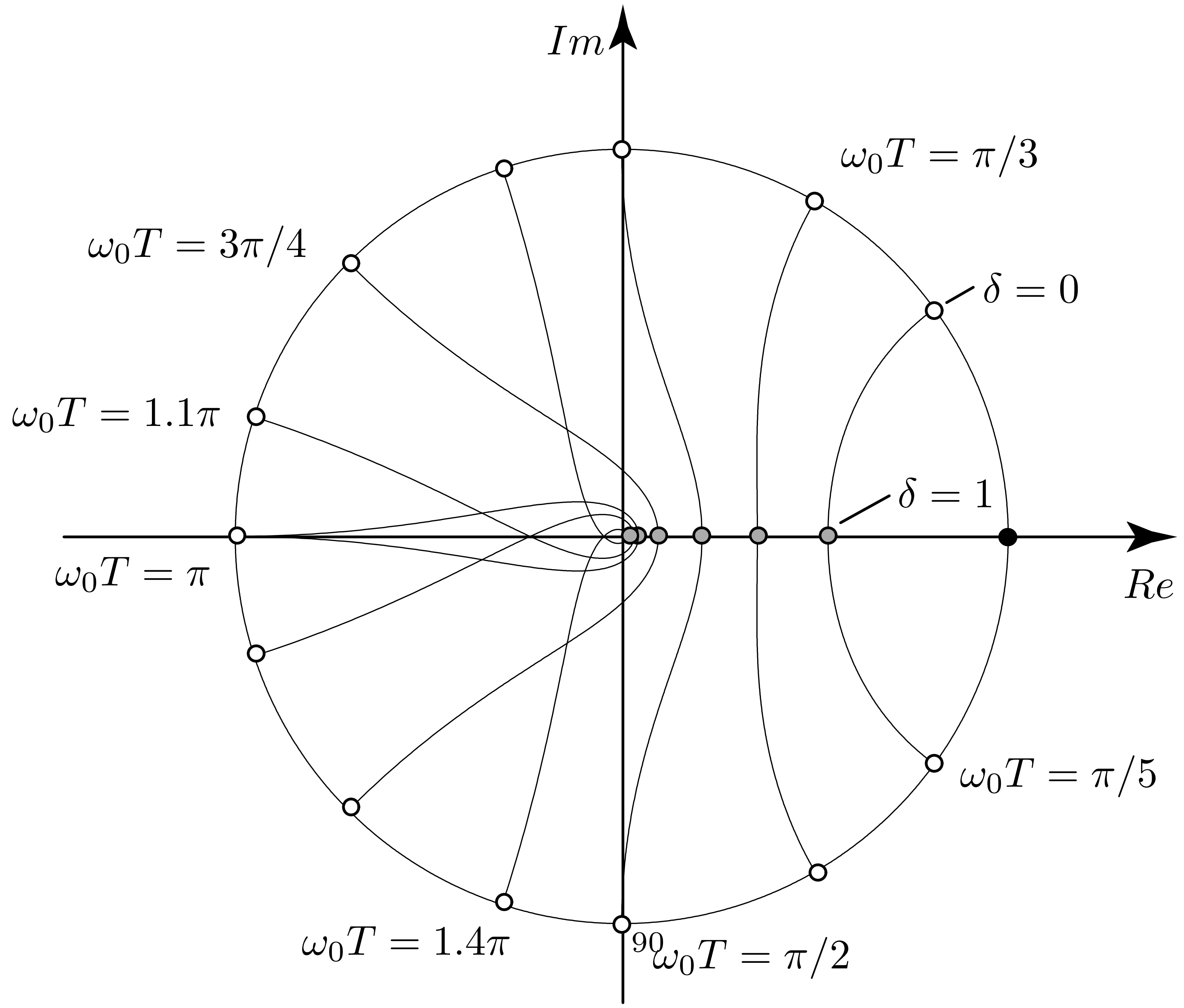
$$\pi^s_{1/2} = -\omega_0 \cdot (\delta \pm j\sqrt{1-\delta^2}), \quad \delta < 1 \tag{131}$$

and, using the mapping (128), the poles of the corresponding discrete-time system are found to be

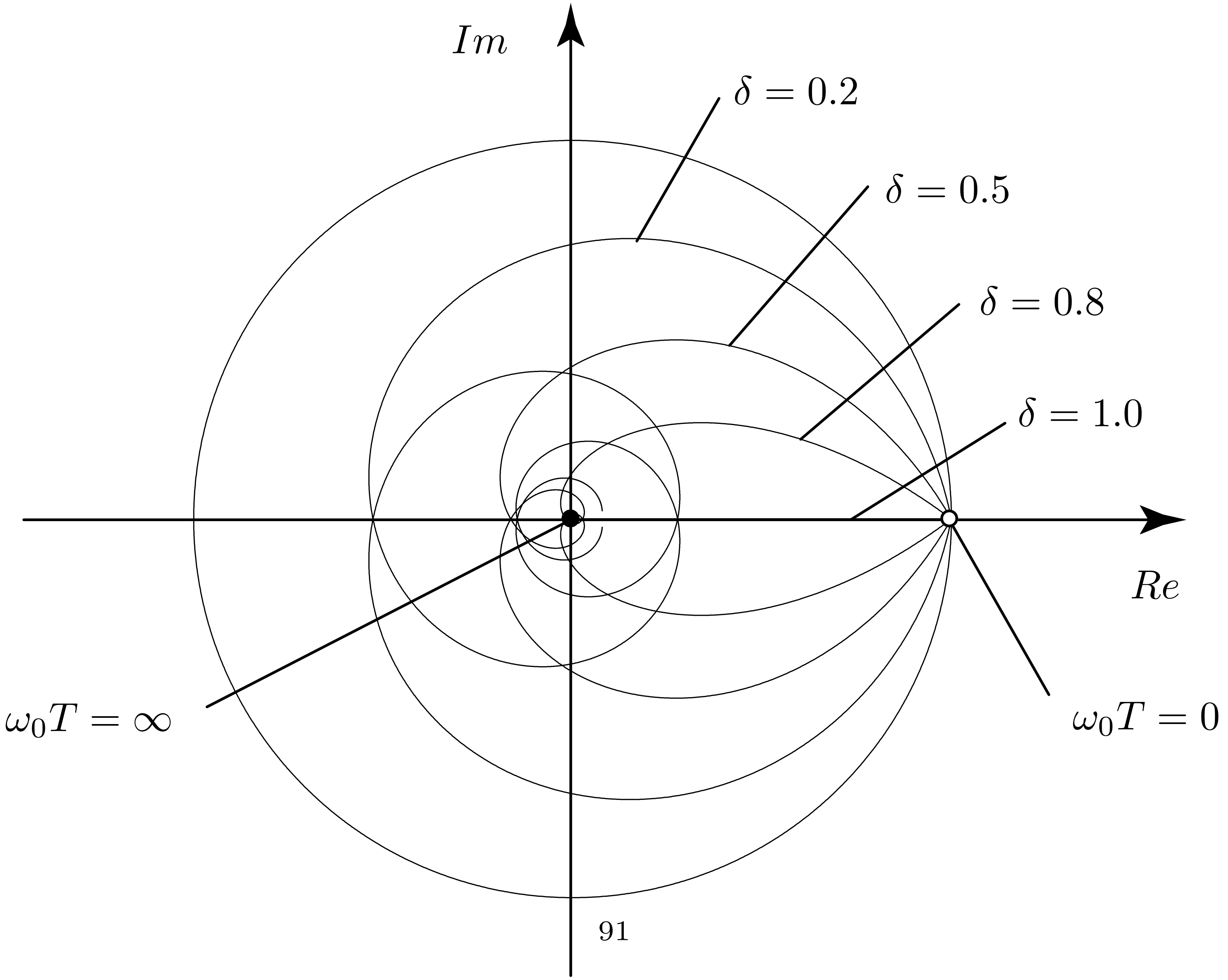
$$\pi^z_{1/2} = e^{-\delta \cdot \omega_0 \cdot T} \cdot \Big(\cos(\omega_0 \cdot \sqrt{1-\delta^2} \cdot T) \pm j \sin(\omega_0 \cdot \sqrt{1-\delta^2} \cdot T)\Big) \tag{132}$$

The next two figures show the level curves of this mapping.

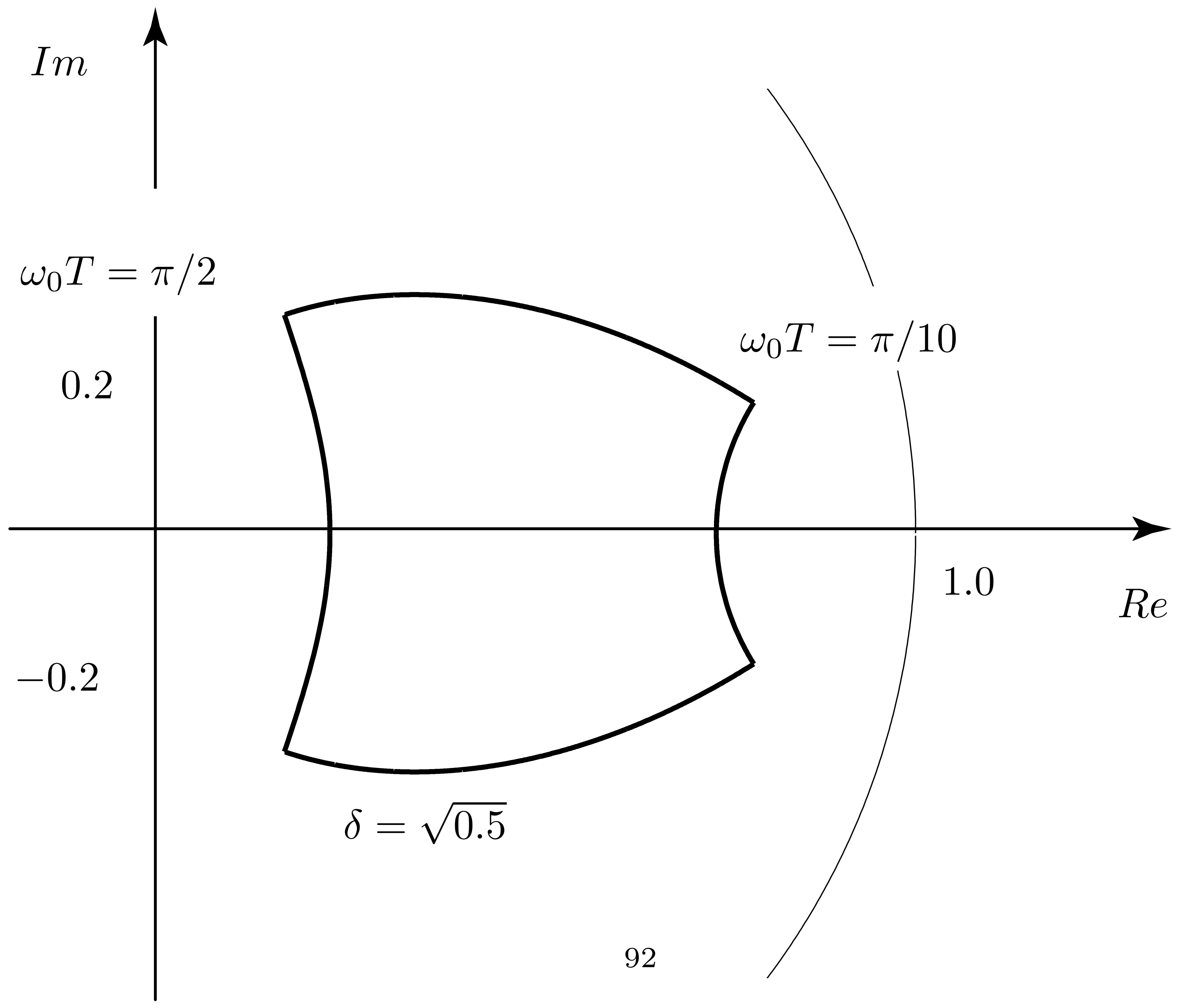
Mapping for constant normalized frequencies $\omega_0 T$



Mapping for constant damping ζ



“Nice pole region” for a second-order discrete-time system.



The “nice pole region” for a discrete-time system has the form shown in the figure. The arguments supporting that specific choice are:

- The damping has to be larger than $\delta = 1/\sqrt{2}$ to avoid a large overshoot.
- The frequency ω_0 has to be sufficiently smaller than the Nyquist frequency π/T. A limit of $\omega_0 < 0.5 \cdot \pi/T$ is reasonable.
- If the system $P(z)$ has to be implemented using a digital computer, the frequency ω_0 may not be several orders of magnitude faster than the Nyquist frequency (“fast sampling”) otherwise even very small numerical errors are likely to have unacceptable effects. The limit $\omega_0 > 0.1 \cdot \pi/T$ is reasonable.

These rules of thumb will be especially useful in root locus controller-design techniques which will be introduced in the next chapter.

Transformation Rules for System Zeros

The connections between the zeros ζ_i^z of the discrete-time system $P(z)$ and the zeros ζ_i^s of the original continuous-time system $P(s)$ are not as simple as it is the case for the poles of these systems. For instance it is not true that $\zeta_i^z = e^{\zeta_i^s\,T}$ and even the number of finite zeros of $P(s)$ and $P(z)$ do not coincide.

In general, a discrete-time system $P(z)$ obtained using a sample-and-hold operation of a strictly proper continuous-time system $P(s)$ will have $n-1$ finite zeros (of course, if the plant $P(s)$ is proper but not strictly proper, the discretized plant $P(z)$ will have n finite zeros as well) where n is the order of $P(s)$.

However, for sufficiently small sampling times T, m of these zeros will be close to $e^{\zeta_i^s\,T}$ with m being the number of finite zeros n_i of the continuous-time plant $P(s)$. The following example illustrates this point.

Zeros of a First-Order System

The continuous-time system

$$P(s) = \frac{s+b}{s+a} \tag{133}$$

may be transformed directly into its discrete-time counterpart

$$P(z) = \frac{z - \left[1 - b/a(1 - e^{-aT})\right]}{z - a^{-aT}} \tag{134}$$

The discrete-time zero

$$\zeta^z = 1 - b/a(1 - e^{-aT}) \tag{135}$$

is of course not equal to $e^{\zeta^s T}$ where $\zeta^s = -b$. However,

$$\zeta^z = 1 - b/a(1 - [1 - aT + \frac{1}{2}a^2T^2 - \ldots]) \tag{136}$$

shows that for small sampling times T the zero of $P(z)$ is at

$$\zeta^z \approx 1 - bT \approx e^{-bT} \tag{137}$$

The remaining $n-m-1$ zeros will converge for $T \to 0$ to the zeros of the following polynomials

$$\begin{array}{c|l} n-m-1=1 & p_1(z) = z+1 \\ n-m-1=2 & p_2(z) = z^2+4z+1 \\ n-m-1=3 & p_3(z) = z^3+11z^2+11z+1 \\ n-m-1=4 & p_4(z) = z^4+26z^3+66z^2+26z+1 \\ \cdots & \cdots \end{array} \tag{138}$$

Polynomials p_i with $i > 1$ have roots outside the unit disc! Therefore, MP $P(s)$ will be transformed always into NMP $P(z)$, provided $r > 2$ of $P(s)$ and $T \to 0$.

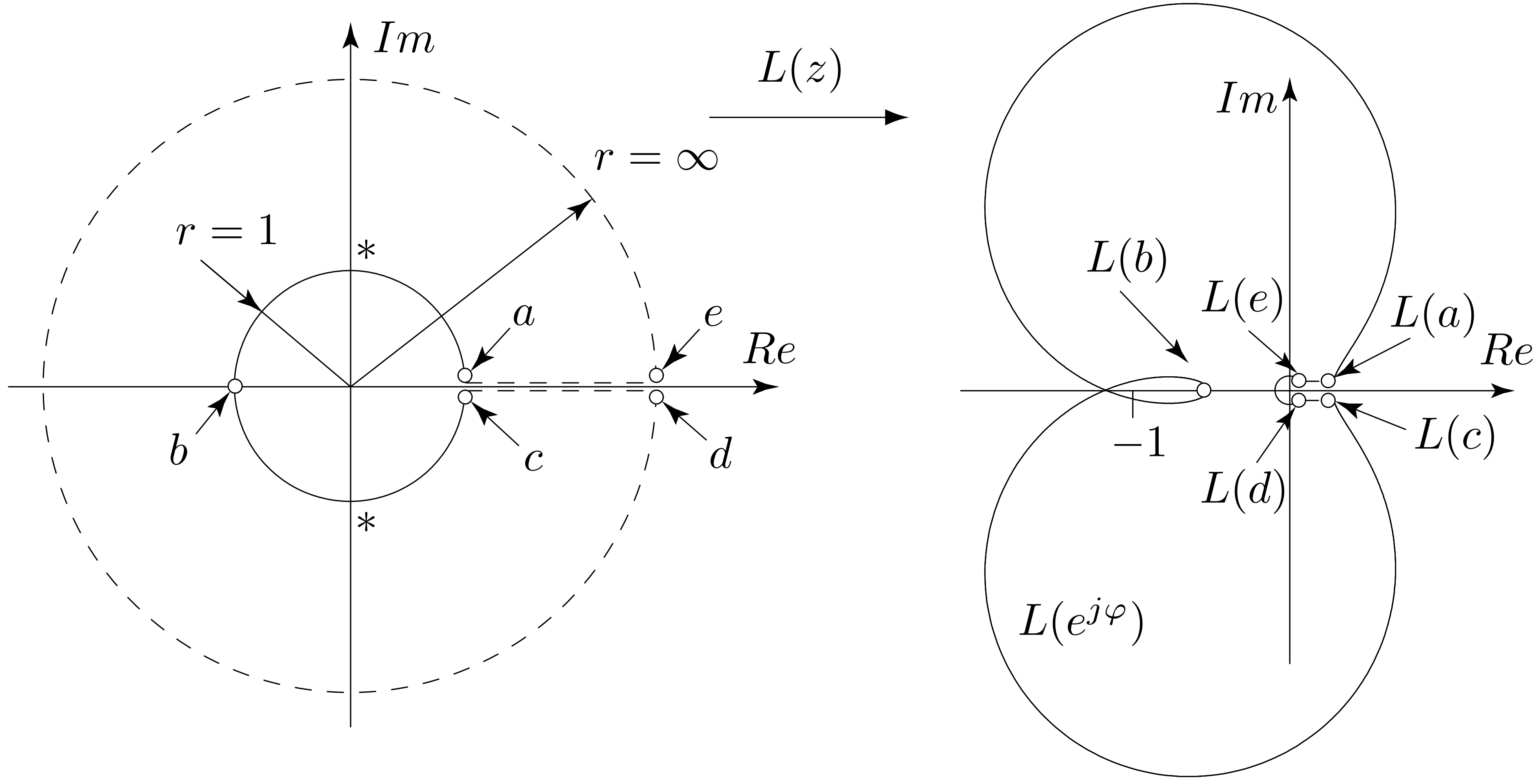
Interestingly, it is not always the case that NMP systems $P(s)$ are transformed into NMP systems $P(z)$.

Nyquist Criterion

The Nyquist criterion is one of the most useful tools for the analysis of continuous-time systems. Its main advantage is that it permits to determine the stability of a closed-loop feedback system by analyzing the corresponding open-loop transfer function. Obviously, such a tool would be useful in a discrete-time setting as well.

The main idea behind the Nyquist criterion is to analyze the phase change of a mapping of points lying on a curve which encloses all unstable poles of that mapping (using the so called "principle of arguments"). A possible choice of such a curve for the discrete-time case is shown below. The important part is the "perforated unit circle" (starting at point a, passing through point b, and ending at point c). The closure of the curve, indicated with the dashed lines and the dashed circle with radius $r \to \infty$, is usually not analyzed. In fact, for strictly proper loop gains $L(s)$, most of these points will be mapped to the origin.

Definition of the discrete-time Nyquist contour (left-hand side) and Nyquist curve of the following example. The unstable poles are marked with an asterisk.



All other arguments in the derivation of the Nyquist criterion remain the same as in the continuous-time case. Accordingly, the discrete-time Nyquist criterion guarantees the stability of the closed-loop system

$$T(z) = \frac{L(z)}{1 + L(z)} \tag{139}$$

iff the curve $L(e^{j\varphi})$ encloses the critical point -1 for $\varphi \in [\epsilon,\ 2\pi - \epsilon]$ as many times as the number of unstable poles of $L(z)$ (counting counter-clockwise encirclements positively).

Notice that the Nyquist criterion is rather powerful because it permits to estimate the stability margin of a certain control system. In fact, a system which has a large minimum distance to the critical point -1 will be “robust” because it can tolerate large model uncertainties.

Stability of a Closed-Loop System

Assume that the open-loop transfer function of a control system is given by

$$L(z) = \frac{z - 0.5}{z^2 - 0.2z + 1.4} \tag{140}$$

The system is unstable with two poles at $0.1 \pm j \cdot 1.179$. Therefore its Nyquist plot $L(e^{j\varphi})$ has to encircle the critical point twice.

As the last figure shows, this is indeed the case. Accordingly, the closed-loop system is stable having its two poles at $-0.4 \pm j \cdot 0.86\ldots$

The control loop, however, is not robust. Small perturbations in the loop gain will destabilize the system.

5 Lecture — Discrete Time Synthesis

Motivating example showing that there is something to be gained ...
Plant to be controlled

$$P(s) = \frac{1}{s+1} \tag{141}$$

embedded in the usual sample-and-hold structure, sampling time $T = 1$ s. The discrete-time representation of $P(s)$ is

$$P(z) = \frac{b}{z-a}, \qquad b = 1 - e^{-T}, \quad a = e^{-T} \tag{142}$$

For these plants three different controllers are compared:

1. A continuous-time PI controller

$$C_1(s) = k_p \cdot \left(1 + \frac{1}{T_i s}\right) \tag{143}$$

with gain $k_p = 1$ and integration time $T_i = 0.5$. This controller will be used to control the continuous-time plant $P(s)$.

2. The discrete-time emulation of this PI controller obtained by applying the Tustin transformation (24)

$$C_2(z) = 2 \cdot \frac{z}{z-1} \tag{144}$$

   C_2 and the subsequent controller C_3 will be used to control the discrete-time plant $P(z)$.

3. A discrete-time controller

$$C_3(z) = \kappa \cdot \frac{z-\beta}{z-\alpha} \tag{145}$$

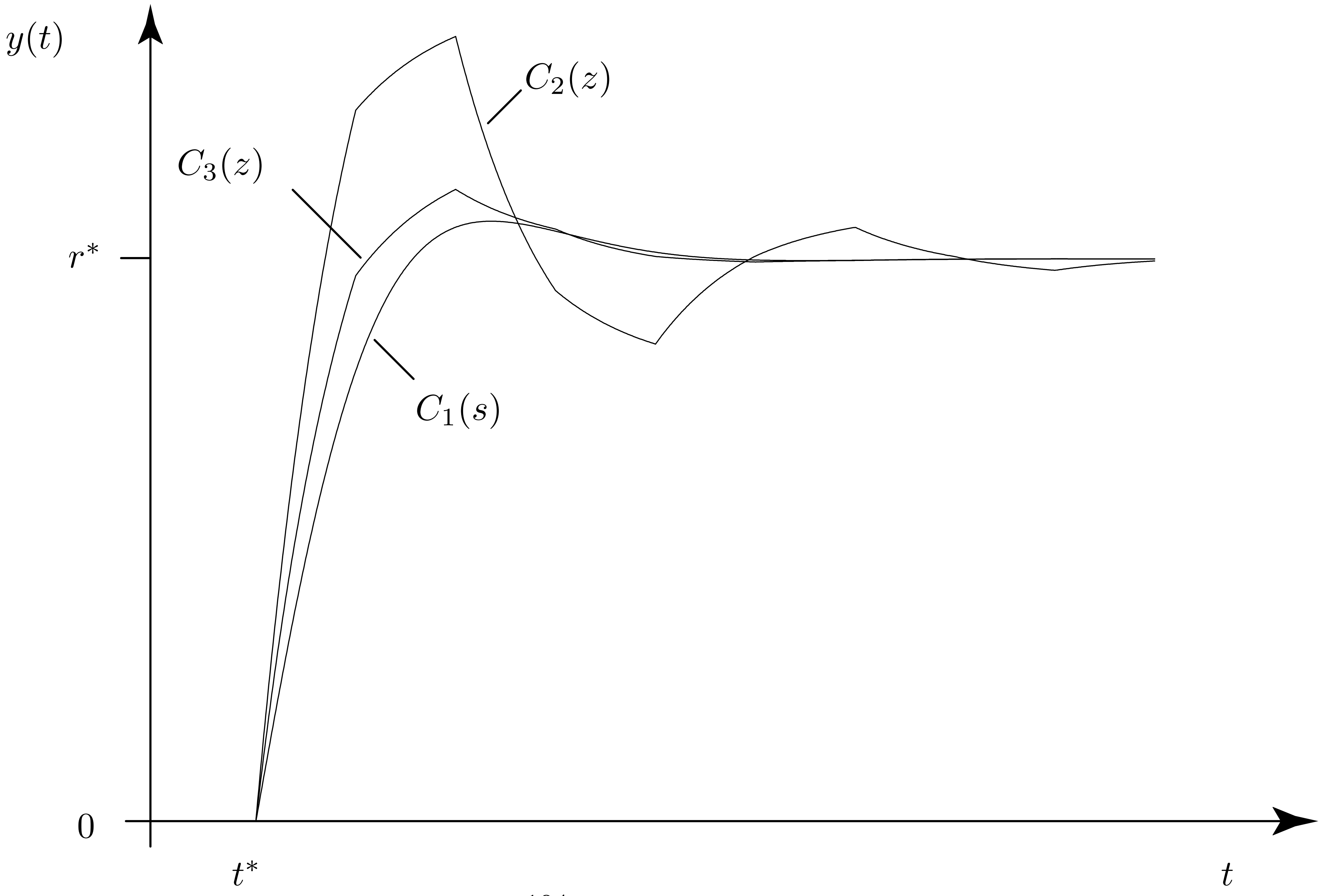
   whose parameters $\kappa,\ \beta,\ \alpha$ will be chosen below in order to produce a desired pole position.

The cross-over frequency ω_c of the loop gain $L(s) = P(s)\,C(s)$ is at $1.4\ldots\ rad/s$. A sampling time of $T = 1\ s$ obviously is a slow sampling situation with a Nyquist frequency $\omega_N = \pi \approx 2 \cdot \omega_c$.

The poles of the closed-loop continuous-time system $T(s)$ are at $s_{1,2} = -1 \pm j$. The poles to be realized by the controller $C_3(z)$ are chosen to be $z_i = e^{s_i},\ i = \{1, 2\}$, i.e., the controller $C_3(z)$ produces the same closed-loop poles as the continuous-time closed-loop system has after the usual sample-and-hold operation. The following values of the controller parameters satisfy this requirement

$$\kappa = \frac{1 + a - z_1 - z_2}{b}, \qquad \beta = \frac{a - z_1 \cdot z_2}{1 + a - z_1 - z_2}, \qquad \alpha = 1 \tag{146}$$

The next figure shows the step response of the system to a change of the reference value r from 0 to r^* at time t^*. As expected, the continuous-time controller $C_1(s)$ performs best. The "correctly" discretized controller $C_3(z)$ doesn't cause substantial deterioration of the closed-loop behavior. The emulation controller $C_2(z)$ performs quite poorly. A direct discrete-time controller design can lead to an acceptable closed-loop performance even in slow-sampling situations where emulation-based controllers fail.


$y(t)$
$C_2(z)$
$C_3(z)$
r^*
$C_1(s)$
0
t^*
t

Goal $\lim_{k\to\infty} y(k) = r_0$, where $r(k) = h(k) \cdot r_0$. Asymptotically stable loop gain $L(z)$ first condition. Second condition derived using the final-value theorem

$$\lim_{k\to\infty} x(k) = \lim_{z\to 1}(z-1)\cdot X(z) \tag{147}$$

Pro memoria

$$\mathcal{Z}\{r(k)\} = \frac{z}{z-1}\cdot r_0 \tag{148}$$

System output

$$y(k),\ \text{with } \mathcal{Z}\{y(k)\} = T(z)\cdot\frac{z}{z-1}\cdot r_0 \tag{149}$$

Applying the final-value theorem yields

$$\lim_{k\to\infty} y(k) = \lim_{z\to 1}(z-1)\cdot T(z)\cdot\frac{z}{z-1}\cdot r_0 = T(1)\cdot 1\cdot r_0 \tag{150}$$

Therefore, $T(1) = 1$ required. Since

$$T(z) = \frac{L(z)}{1+L(z)} \tag{151}$$

$L(1) = \infty$ is required ($L(z)$ has at least one pole at $z = 1$).

Root-Locus Methods

Root-locus methods represent a powerful approach for designing low-order feedback controllers. The starting point is a plant

$$P(z) = \frac{b_{n-1} z^{n-1} + \ldots b_1 z + b_0}{z^n + a_{n-1} z^{n-1} + \ldots a_1 z + a_0} = \frac{n_P(z)}{d_P(z)} \tag{152}$$

and a static controller $C(z) = k_p$ where the free parameter $k_p \in [k_{p,\,min}, k_{p,\,max}] \subset \mathbb{R}$. This free parameter has to be chosen such as to produce a desired pole configuration for the closed-loop system which is given by

$$T(k_p, z) = \frac{k_p \cdot P(z)}{1 + k_p \cdot P(z)} = \frac{k_p \cdot n_P(z)}{d_P + k_p \cdot n_P(z)} \tag{153}$$

The "noce-pole region" is the target zone for the poles ofthe close-loop system's poles. Obviously, for $k_p = 0$ the poles of the closed-loop system correspond to the poles of the uncontrolled plant. For $k_p \to \infty$ the poles of $T(z)$ approach the $n-1$ finite zeros of the open-loop plant. The remaining pole will be real and will converge to $-\infty$ for $k_p\, b_{n-1} > 0$ and to $+\infty$ for $k_p\, b_{n-1} < 0$.

As shown above, when an n-dimensional strictly-proper continuous-time plant $P(s)$ is transformed into its equivalent discrete-time system $P(z)$, the latter in general will have $n-1$ finite zeros!

One degree of freedom is usually not sufficient to attain a specific desired pole configuration. Iterations with more degrees of freedom often lead to better results. For a full pole placement, however, a controller with n degrees of freedom and more advanced controller design approaches are necessary. These approaches which will be presented below.

Example — Root Locus Design

The continuous-time plant to be controlled is assumed to be given by

$$P(s) = \frac{0.5\omega_0^2 s + \omega_0^2}{s(s^2 + 2\zeta\omega_0 s + \omega_0^2)} \tag{154}$$

with $\omega_0 = 2\pi/3$ and $\zeta = 0.9$. This plant is transformed into its discrete-time equivalent using the usual ZOH transformation. The sampling time is assumed to be $T = 0.6\ s$, i.e., only five times smaller than the plant's time constant.

The discretized plant

$$P(z) = \frac{0.2802\ldots z^2 + 0.1101\ldots z - 0.0585\ldots}{z^3 - 1.5510\ldots z^2 + 0.6552\ldots z - 0.1041\ldots} \tag{155}$$

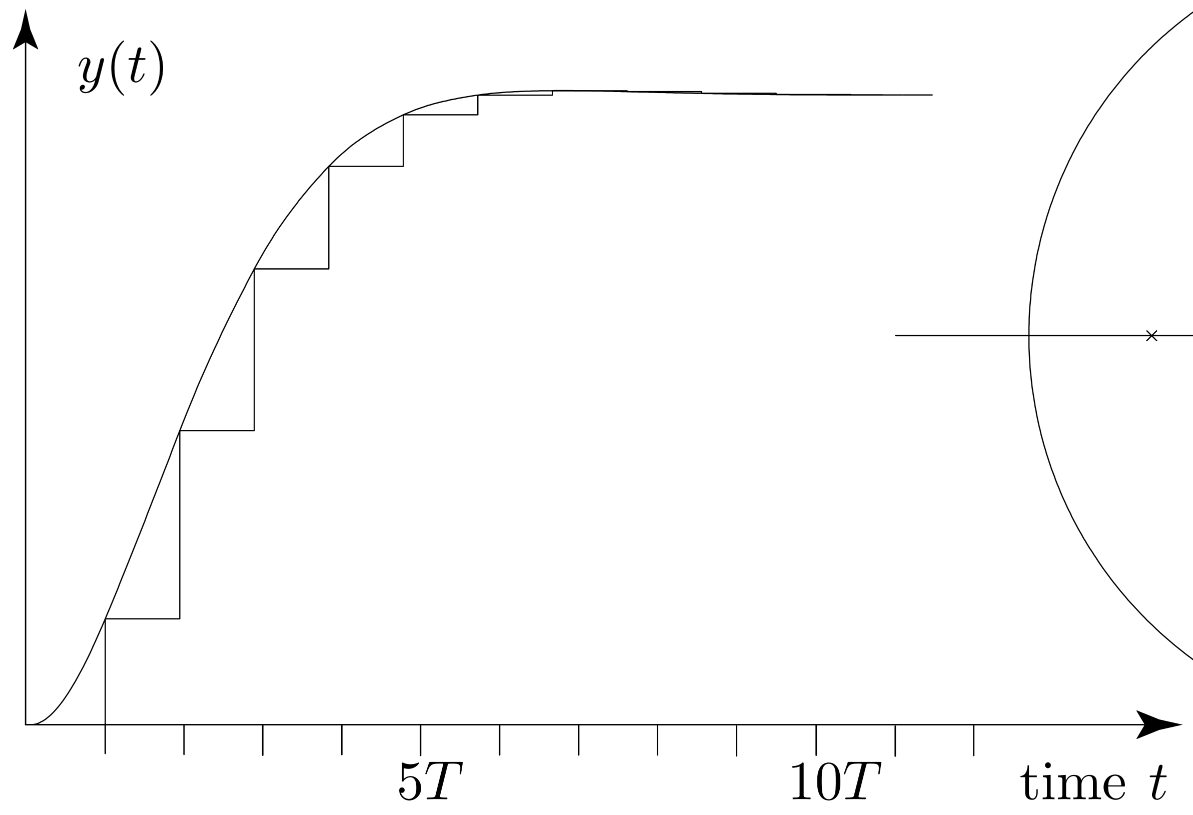
has two finite zeros at $n_1 = -0.6938\ldots$ and $n_2 = 0.3008\ldots$ and three poles at $p_1 = 1$ and $p_{2,3} = 0.2755\ldots \pm j\,0.168\ldots$

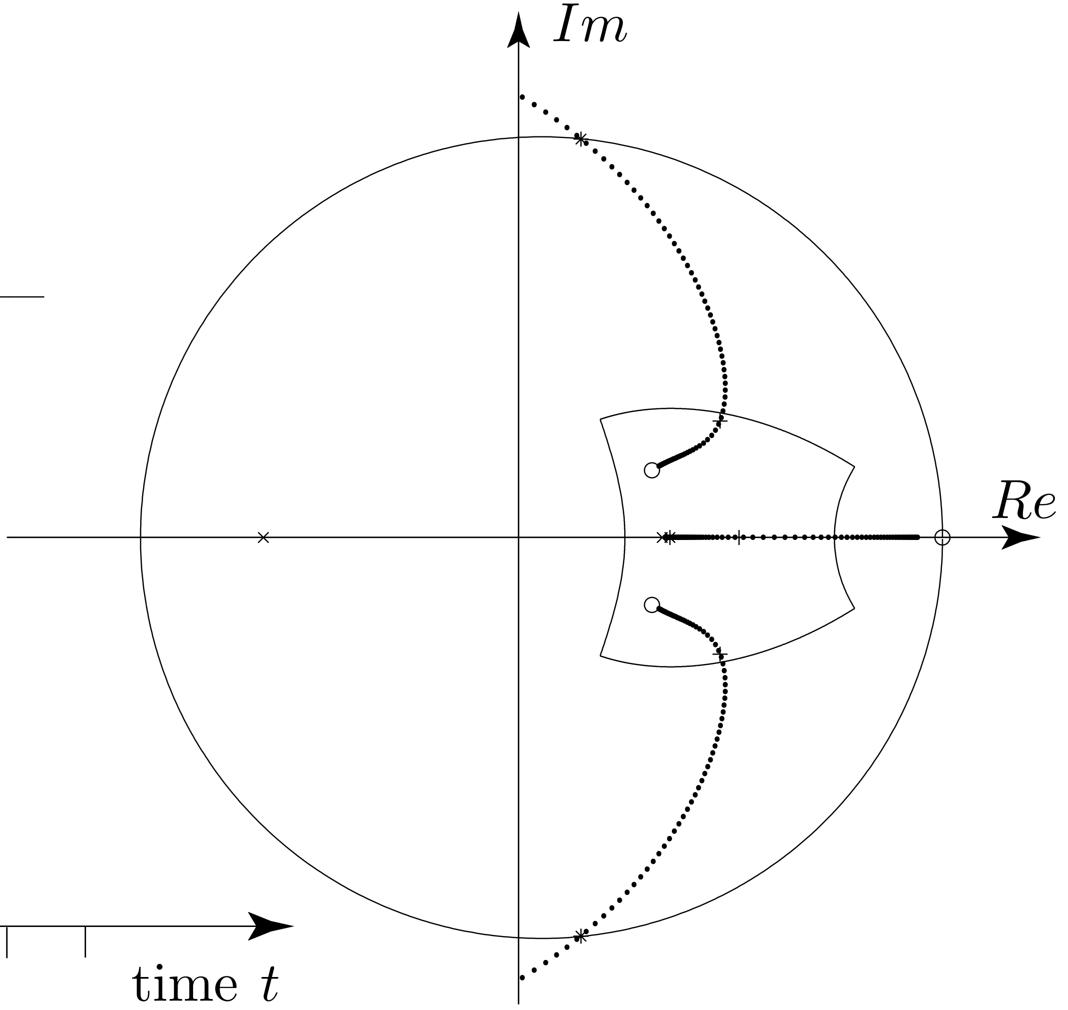
The controller $C(z)$ is a delay-free proportional controller

$$C(z) = k_p \tag{156}$$

The poles of the closed-loop system as a function of the gain k_p are shown in Figure 5.2. The critical gain k_p^*, i.e., the gain at which the system is on the limit between being asymptotically stable and unstable, has a numerical value of $3.69\ldots$ The gain $k_p^+ = 0.6$ is chosen such that the system poles are all in the "nice pole region."

Root locus (left), $\circ$ = open-loop poles, $\times$ = system zeros, $+$ = poles at chosen gain $k_p^+ = 0.6$, $*$ = poles at critical gain $k_p^* = 3.69$. Step response (right) of the closed-loop system with gain $k_p = 0.6$.





Loop-Shaping Design in the Frequency Domain

The main ideas of this design technique are the same as in the continuous-time case. Typically, a controller structure is chosen at the outset (for instance a PID or a lead/lag controller) and the parameters of this system are "tuned" such that the open-loop frequency response attains the desired properties.

According to the Nyquist criterion, the stability of a perturbed closed-loop system is guaranteed as long as the number of encirclements of the critical point remains unchanged from the original design.

Accordingly, the loop-shaping method consists of choosing those controller parameters which produce a large minimum return difference and possibly satisfy other requirements like cross-over frequency limits or gain and phase margins. The following example shows such a design procedure for a simple problem setup.

Example — Controller Design Using Loop-Shaping Techniques

The system to be controlled is assumed to be given by

$$P(s) = \frac{s+1}{s^2+s+1} \tag{157}$$

The controller structure is chosen to be an ideal PID controller

$$C(s) = k_p \left[1 + \frac{1}{T_i\, s} + T_d\, s \right] \tag{158}$$

which is transformed into its discrete-time equivalent using the backward Euler emulation (24)

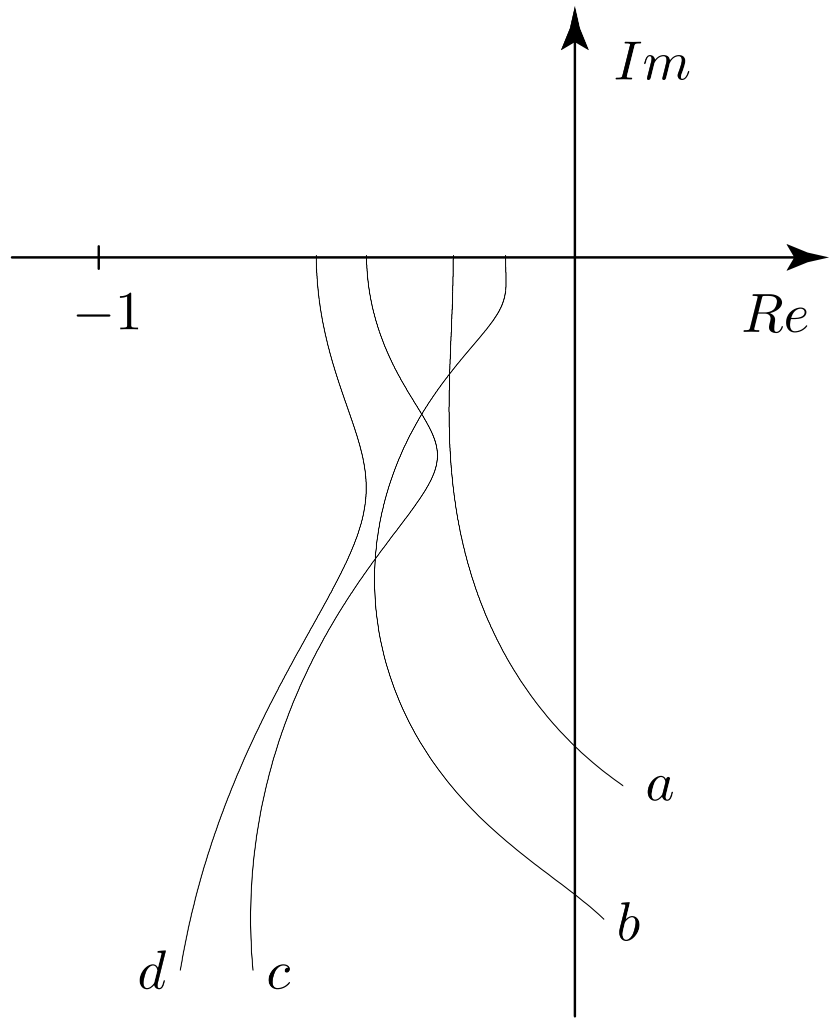
$$C(z) = k_p \left[1 + \frac{T}{T_i} \frac{z}{z-1} + \frac{T_d}{T} \frac{z-1}{z} \right] \tag{159}$$

The three parameters $\{k_p,\, T_i,\, T_d\}$ are chosen such that a “nice” frequency response $L(e^{j\,\omega\, T})$ is obtained. Of course, only frequencies $\omega < \pi/T$ have to be considered.

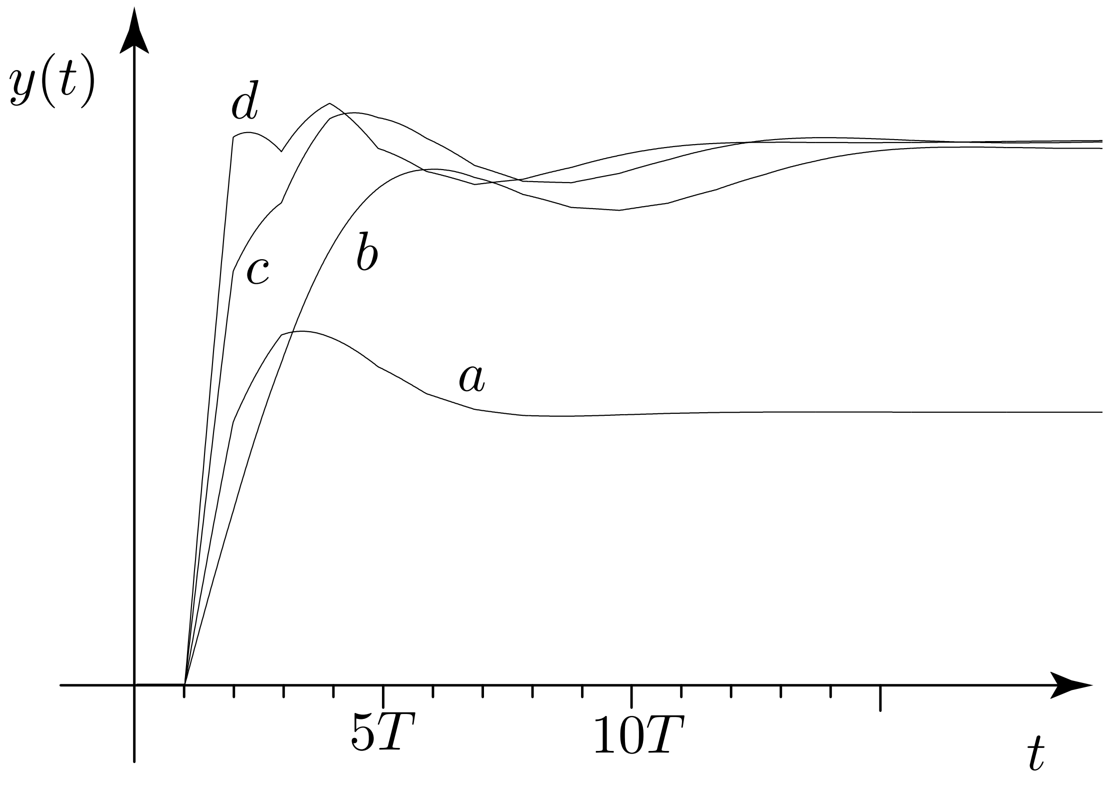
The next figure shows the frequency responses of the open-loop gain and the step responses of the closed-loop system for the following four choices of the controller parameters

- Case a: $\{k_p, T_i, T_d\} = \{1, \infty, 0\}$
- Case b: $\{k_p, T_i, T_d\} = \{0.3, 0.5, 0.1\}$
- Case c: $\{k_p, T_i, T_d\} = \{0.5, 0.4, 0.45\}$
- Case d: $\{k_p, T_i, T_d\} = \{0.8980, 0.5860, 0.2351\}$

As expected, for an integrator gain of zero (i.e., $T_i = \infty$), the closed-loop system has a non-unity gain. In the other cases, reducing the minimum distance to the critical point reduces settling times, but it reduces system robustness as well.



Im
-1
Re
a
b
c
d



$y(t)$
d
c
b
a
$5T$
$10T$
t

Plant Inversion and Dead-Beat Control

At a first glance plant inversion techniques are elegant and powerful design methods. However, they should be used with caution in order to avoid serious problems, as shown in this section. The starting point of the considerations that follow is a SISO plant

$$P(z) = \frac{b_m z^m + b_{m-1} z^{m-1} + \dots b_1 z + b_0}{z^n + a_{n-1} z^{n-1} + \dots a_1 z + a_0} = \frac{n(z)}{d(z)} \tag{160}$$

The plant is assumed to be asymptotically stable and minimum phase, i.e., all poles and finite zeros are assumed to be inside the unit circle.[1] Without this assumption, unstable pole-zero cancellations occur which – even in the best case of a perfect cancellation – produce unstable unobservable modes and therefore internal instabilities.

1) Remember: This is a restricting assumption. Even if the original continuous-time plant was asymptotically stable and minimum phase, the discrete-time system might have zeros outside the unit disk.

The controller $C(z)$ is chosen in order to realize a desired closed-loop system transfer function

$$T(z) \overset{!}{=} \frac{\nu(z)}{\delta(z)} \tag{161}$$

Based on this equation the desired controller transfer function can be derived directly

$$C(z) = \frac{d(z)}{n(z)} \cdot \frac{\nu(z)}{\delta(z) - \nu(z)} \tag{162}$$

Obviously, the reference transfer function must have (at least) as many pure delays as the original plant had (otherwise the controller will not be causal). In addition, very often the reference system will be chosen to have unity gain at steady state

$$\nu(1) = \delta(1) \tag{163}$$

Many alternatives are possible for the polynomials $\nu(z)$ and $\delta(z)$. Typically the poles of such reference models will be chosen to be in the "nice pole region" as defined above.

An interesting option is to prescribe a "dead-beat" behavior, i.e., a system which has a *finite* settling time to changes in reference inputs. Finite settling times can only be achieved by non-recursive reference systems, i.e.,

$$\delta(z) = z^q, \qquad \nu(z) = \beta_p z^p + \beta_{p-1} z^{p-1} + \ldots + \beta_1 z + \beta_0 \tag{164}$$

for which condition (163) yields the additional requirement

$$\sum_{i=0}^{p} \beta_i = 1 \tag{165}$$

Sometimes such systems are referred to as FIR filters, where FIR stands for finite impulse response as opposed to IIR filters which have an infinite impulse response and work recursively.
The resulting controller will be causal if the inequality $n - m \leq q - p$ is satisfied. The specific choice of the coefficients β_i determines the shape of the transients (step responses). Of course the control effort (energy) is greatly affected by this choice. The sampling time needs to be taken into consideration as well. For the same system $P(s)$, shorter sampling times must be compensated by higher controller orders q and p.

The dead-beat controller cancels the poles of the system with its zeros, and it places all of the poles of the closed-loop system at the origin. Accordingly, the closed-loop system reaches its setpoint in q steps, at most.

Warning: as the net figure shows, the settling time of the *discrete-time* closed-loop system is indeed finite. However, the *continuous-time* response may include poorly damped oscillations which are due to the very high gain that dead-beat controllers usually need to achieve the desired behavior. Thus, very small modeling errors may cause a severe performance deterioration. It is therefore recommended to use dead-beat designs only with great caution.

Dead-Beat Design

In this example a dead-beat controller is designed for the continuous-time system

$$P(s) = \frac{1}{s^2 + s + 1} \tag{166}$$

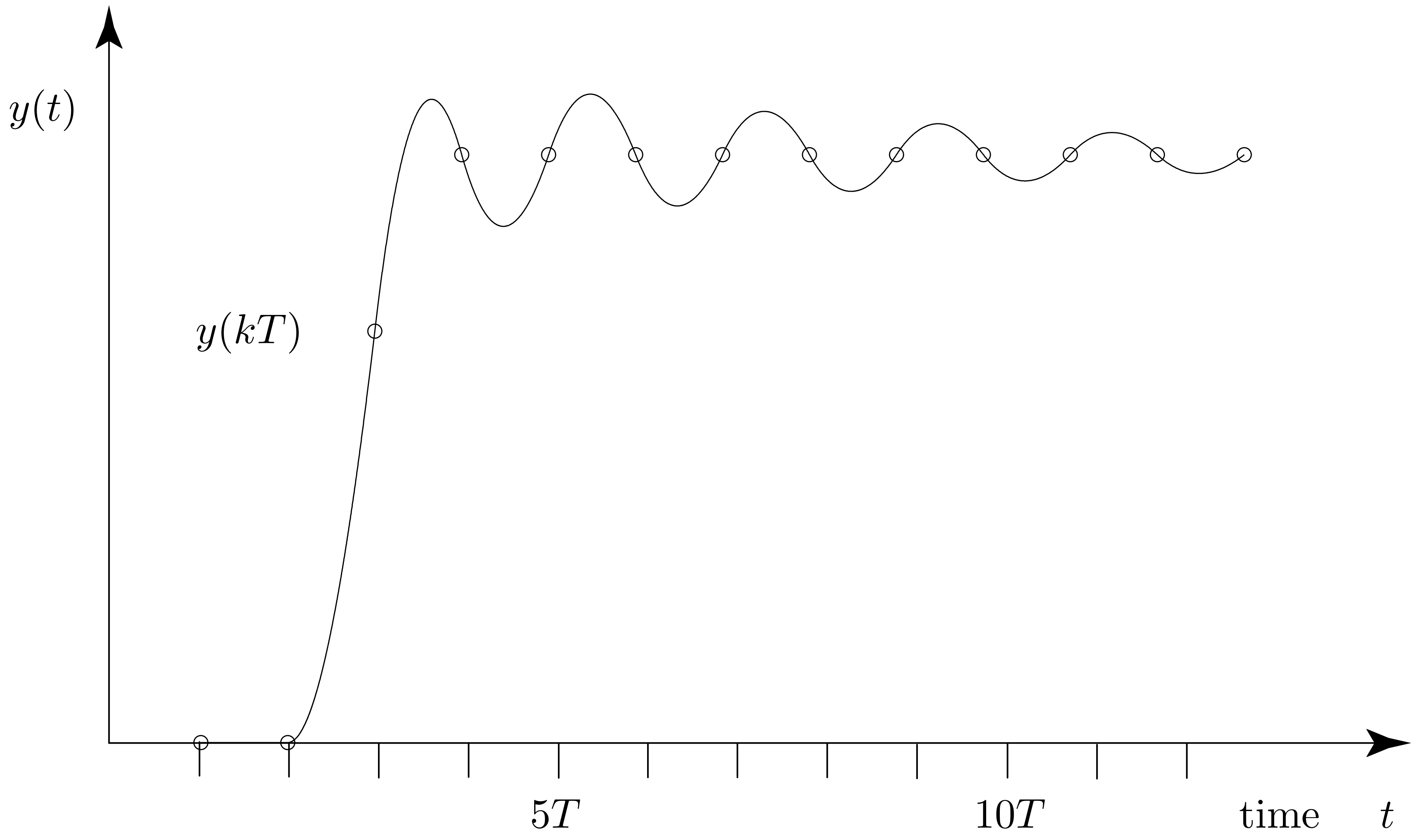
The plant is inserted in the usual ZOH configuration and is discretized with a sampling time of $T = 0.5\,s$. The resulting discrete-time system has the form

$$P(z) = \frac{0.1044\ldots z + 0.0883\ldots}{z^2 - 1.4138\ldots z + 0.6065\ldots} \tag{167}$$

with $n = 2$ and $m = 1$. The reference model is chosen as follows

$$T(z) = \frac{0.7z + 0.3}{z^2} \tag{168}$$

such that the conditions $\beta_1 + \beta_0 = 1$ and $n - m \leq q - p$ are satisfied. The reference step response of the resulting sampled-data closed-loop system is shown in the next figure.


$y(t)$
$y(kT)$
$5T$
$10T$
time t

Controller Synthesis for MIMO systems

Linear Quadratic Regulators

Infinite-horizon linear quadratic regulators are known to have excellent robustness properties in the continuous-time case and to be well applicable in a MIMO setting as well. This section analyzes their discrete-time counterparts.

For discrete-time systems the LQR problem formulation is the same as in the continuous-time case, i.e., the system

$$x_{k+1} = F \cdot x_k + G \cdot u_k, \qquad x_0 \neq 0 \tag{169}$$

has to be brought to the origin by a suitable control signal u such that the criterion

$$J(u) = \sum_{k=0}^{\infty} x_k^T \cdot Q \cdot x_k + u_k^T \cdot R \cdot u_k, \qquad Q = Q^T \geq 0, \quad R = R^T > 0 \tag{170}$$

is minimized.

The solution to this problem is the control signal

$$u_k = -K \cdot x_k \tag{171}$$

where the gain matrix $K \in \mathbb{R}^{m \times n}$ is defined by

$$K = \left[R + G^T\,S\,G\right]^{-1}\,G^T\,S\,F \tag{172}$$

and

$$F^T\,S\,G\left[R + G^T S\,G\right]^{-1} G^T S\,F + S - F^T S\,F - Q = 0 \tag{173}$$

As in the continuous-time case, the optimal controller is linear and time-invariant. The discrete-time algebraic Riccati equation (DARE) (173) is solved using spectral methods (command "“dare” in Matlab).

The formulation as an optimization problem (170) guarantees that the resulting controller will work well in a MIMO setting as well. Its robustness properties are analyzed below for the special case of an SI example (however, the conclusions drawn will be valid in an MI setting as well). In this case the choices $Q_1 = c^T c$ or $Q =$“full” are possible and $R = r \in \mathbb{R}_+$ is a positive scalar.

Fourth-Order SISO System and LQR Controller

The plant to be controlled in this example is described by equation

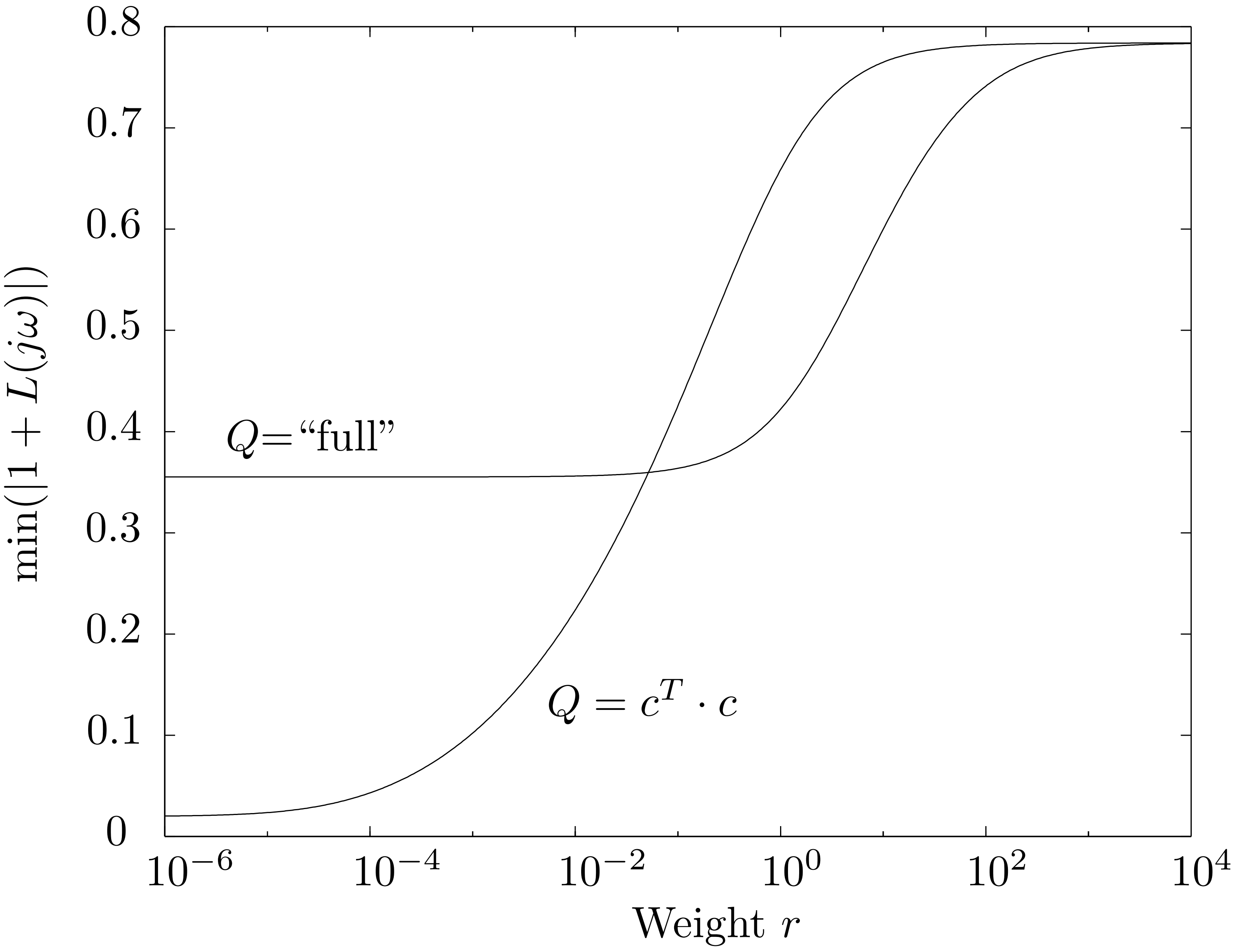
$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & -2 & -2 & -1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u(t), \qquad y(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} x(t) \tag{174}$$

After discretization (with a ZOH element, sampling time $T = 1\ s$) several LQR designs are compared using 2 different Q matrices

$$Q_1 = c^T c, \qquad \text{and} \qquad Q_2 = \begin{bmatrix} 5 & 3 & 1 & 0 \\ 3 & 6 & 4 & 0 \\ 1 & 4 & 8 & 1 \\ 0 & 0 & 1 & 4 \end{bmatrix} \tag{175}$$

and a varying $r \in [10^{-6}, 10^4]$.

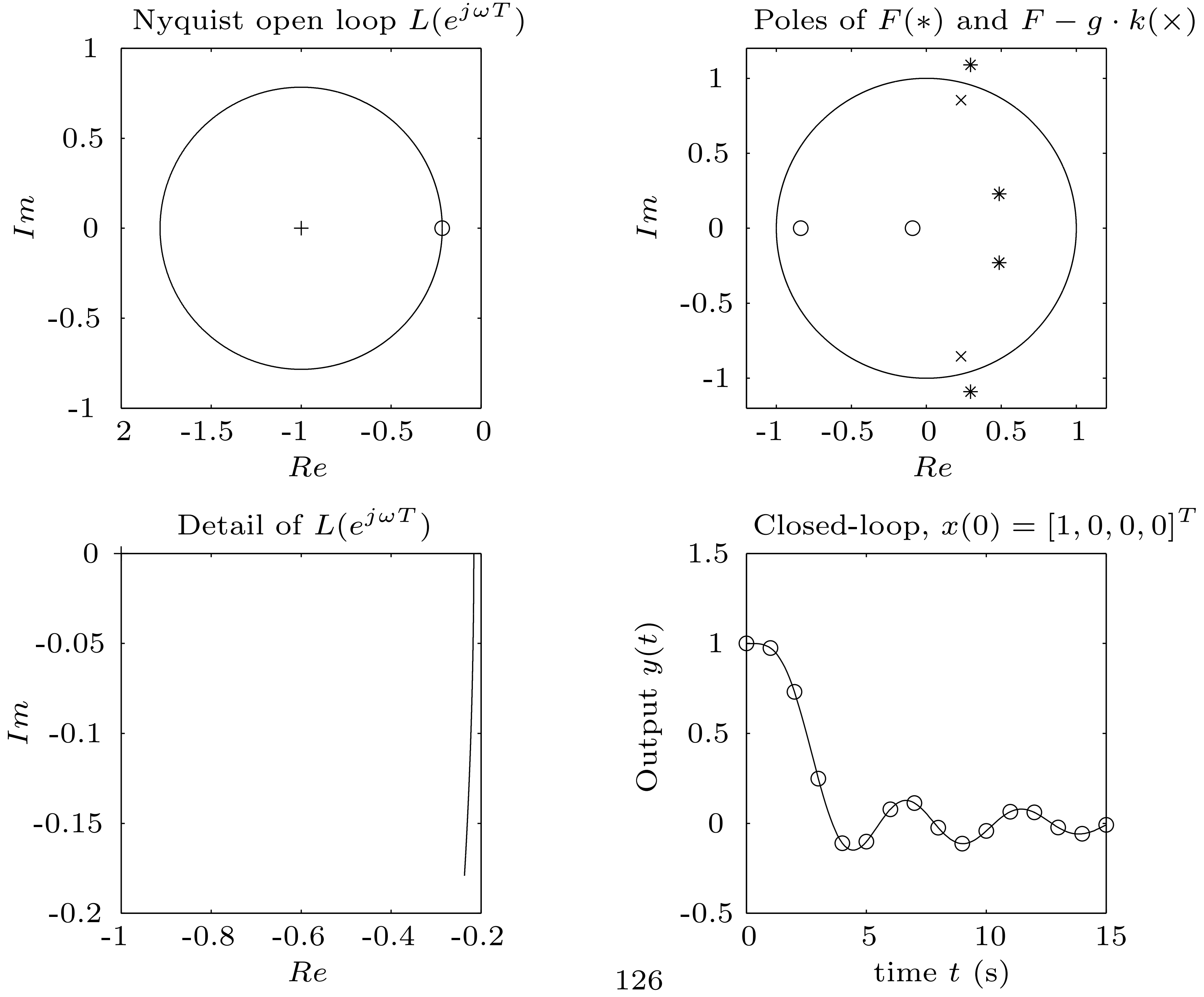
Resulting $|1 + L(e^{j\omega T})|$ for varying weights r



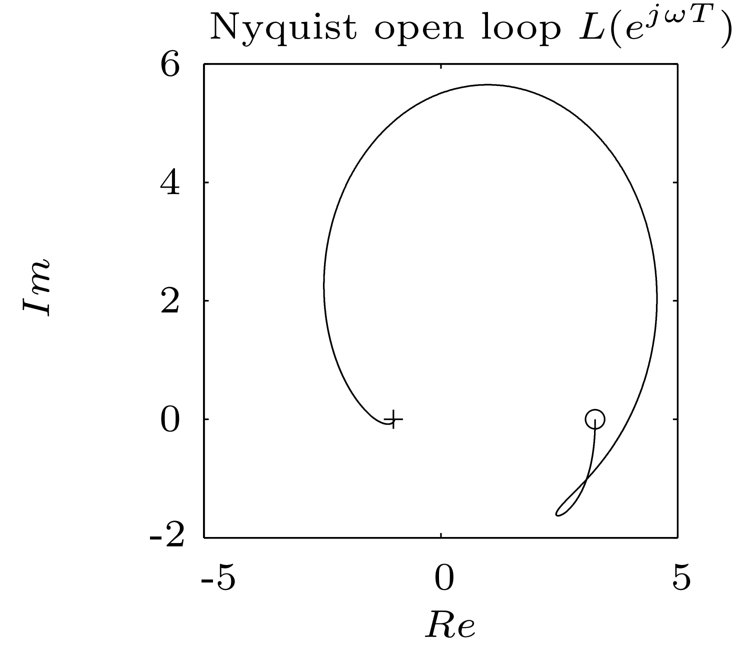
This example shows that the discrete-time LQR controllers do not have the same excellent robustness properties as the continuous-time LQR designs. Especially the minimum return difference $|1 + L(j\omega)|$, even in the best case, is smaller than 1 and may be almost zero.

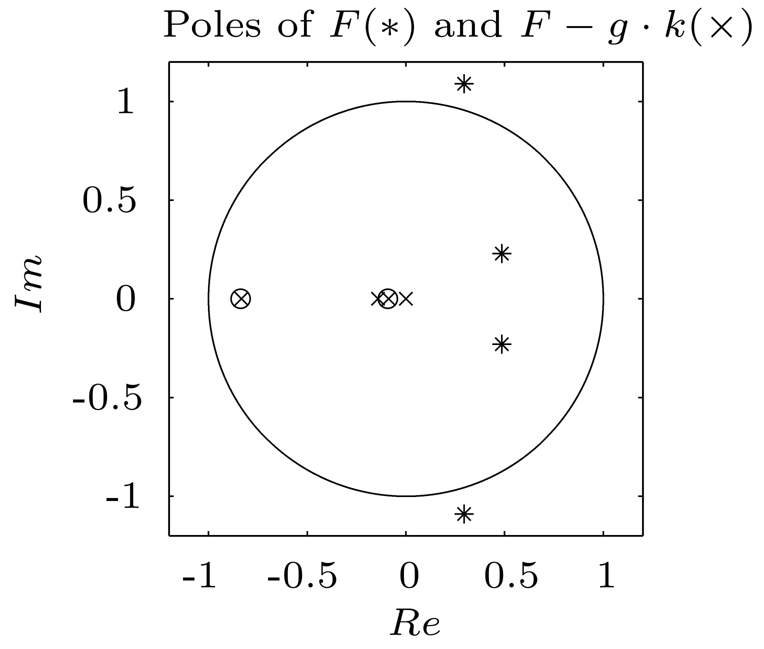
As in the continuous-time case, "expensive-control" feedback (171) minimizes control action and only shifts the unstable poles within the unit circle (reflection at the unit circle in radial direction). The transient behavior is not well damped, but the robustness (expressed by the minimum return difference) is quite good.

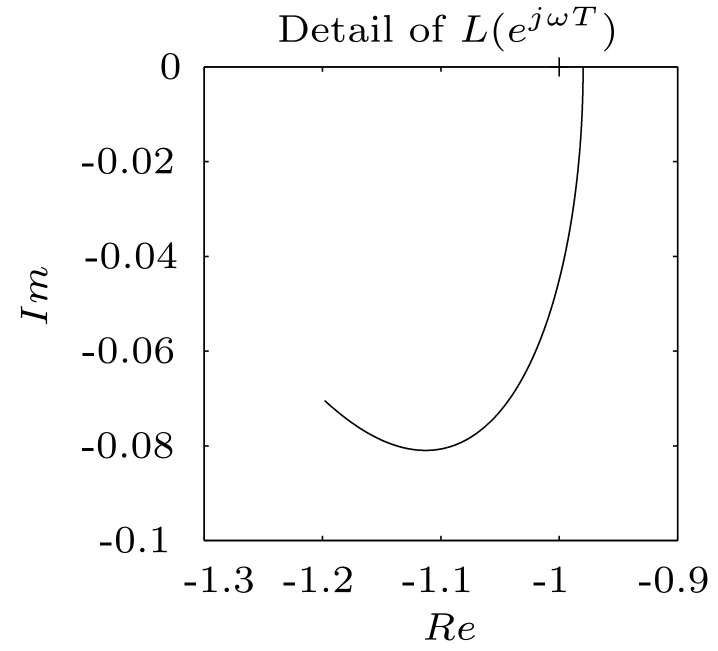
Case 1: “expensive control” case ($r = 10^6$).

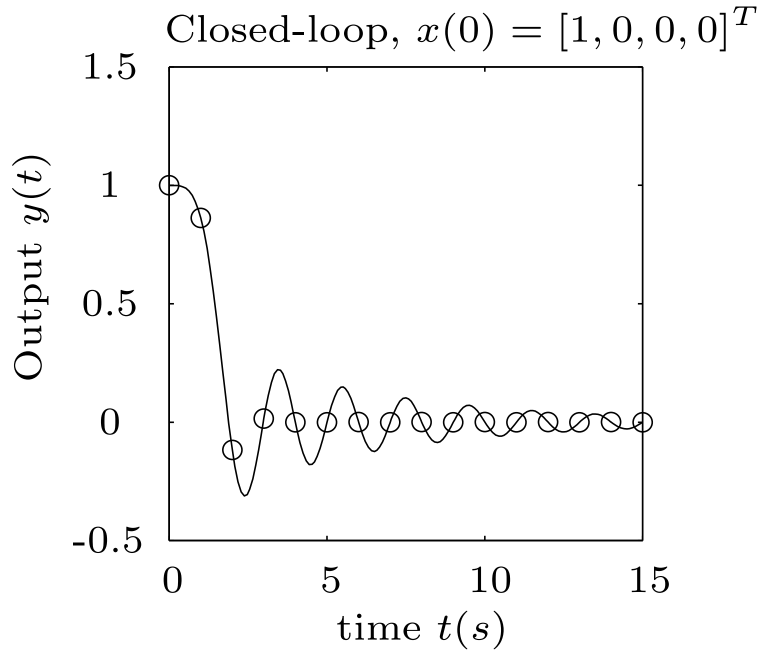


Case 2: Example “cheap control” ($r = 10^{-6}$), weight matrix Q_1.

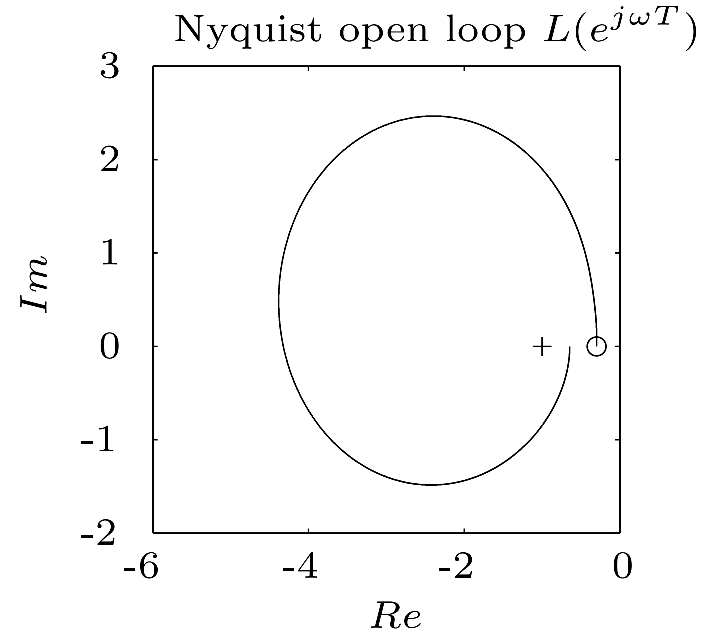


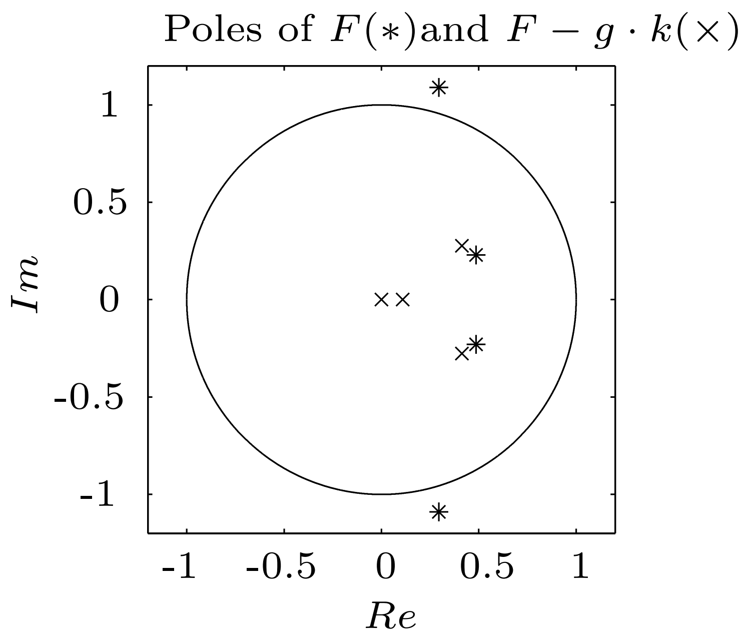


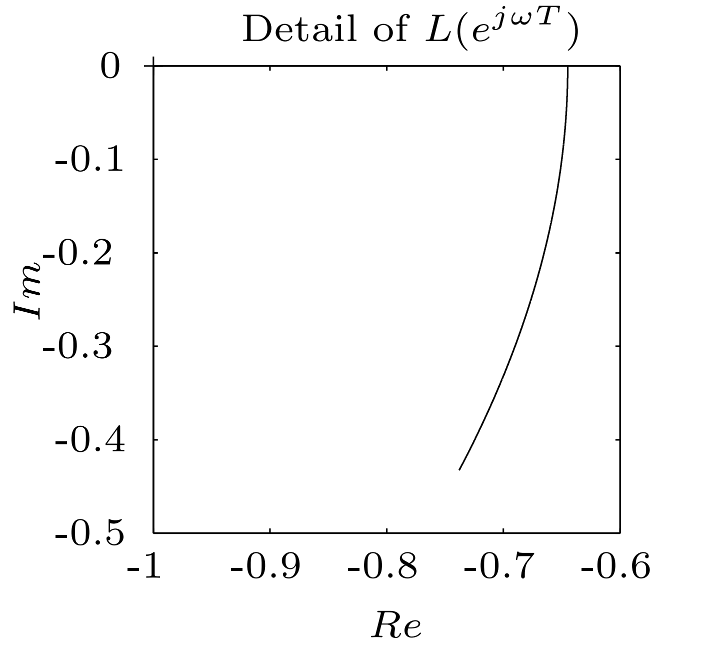


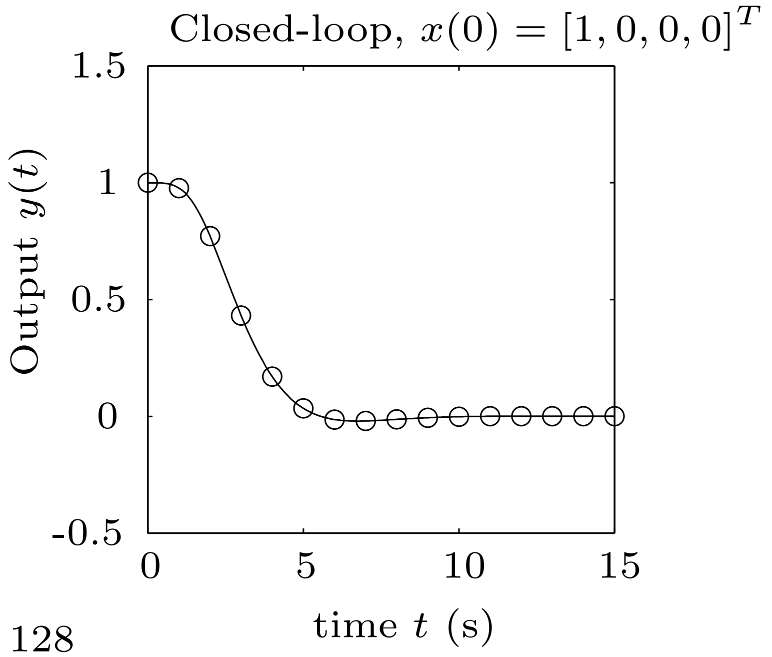


Case 3: "Cheap control" ($r = 10^{-6}$), weight matrix Q_2.









The first “cheap-control” case ($Q = Q_1 = c^T \cdot c$) has 2 finite zeros in the unit circle (the minimum phase transmission zeros of the system $\{F, g, c, 0\}$, indicated by circles in the right top plot). The cheap-control solution moves 2 of the closed-loop poles to these 2 finite zeros (“plant inversion”) and the remaining 2 poles to the origin. This corresponds to a dead-beat control approach. As expected , in this case the robustness properties are extremely poor ($\min\{|1 + L(e^{j\omega T})| \approx 0.02$).

The second “cheap-control” case ($Q = Q_2 =$ “full”) introduces no finite zeros and the pole configuration is determined by the optimization criterion only. The resulting loop has a substantially improved robustness.

A careful choice of the weights Q and R in combination with the sampling time T is therefore necessary. Extreme solutions (for instance “cheap-control”) are not guaranteed to produce the best possible behavior.

In general, LQR controlled systems will have non-zero output for constant disturbances. Therefore, adding p integrators at the system output (adding the integrators at the system input is possible as well and – at least for “square systems” – yields the same closed-loop behavior) is often necessary (see next figure). With these additional integrators the following augmented system description results

$$\widetilde{x}_{k+1} = \widetilde{F} \cdot \widetilde{x}_k + \widetilde{G} \cdot u_k, \qquad \widetilde{x}_k = \begin{bmatrix} v_k \\ x_k \end{bmatrix} \tag{176}$$

where the new system matrices are defined by

$$\widetilde{F} = \begin{bmatrix} I & -C \\ 0 & F \end{bmatrix}, \qquad \widetilde{G} = \begin{bmatrix} 0 \\ G \end{bmatrix}, \tag{177}$$

Introducing a fictitious output equation

$$\widetilde{y}_k = \begin{bmatrix} \gamma \cdot I & C \end{bmatrix} \cdot \widetilde{x}_k \tag{178}$$

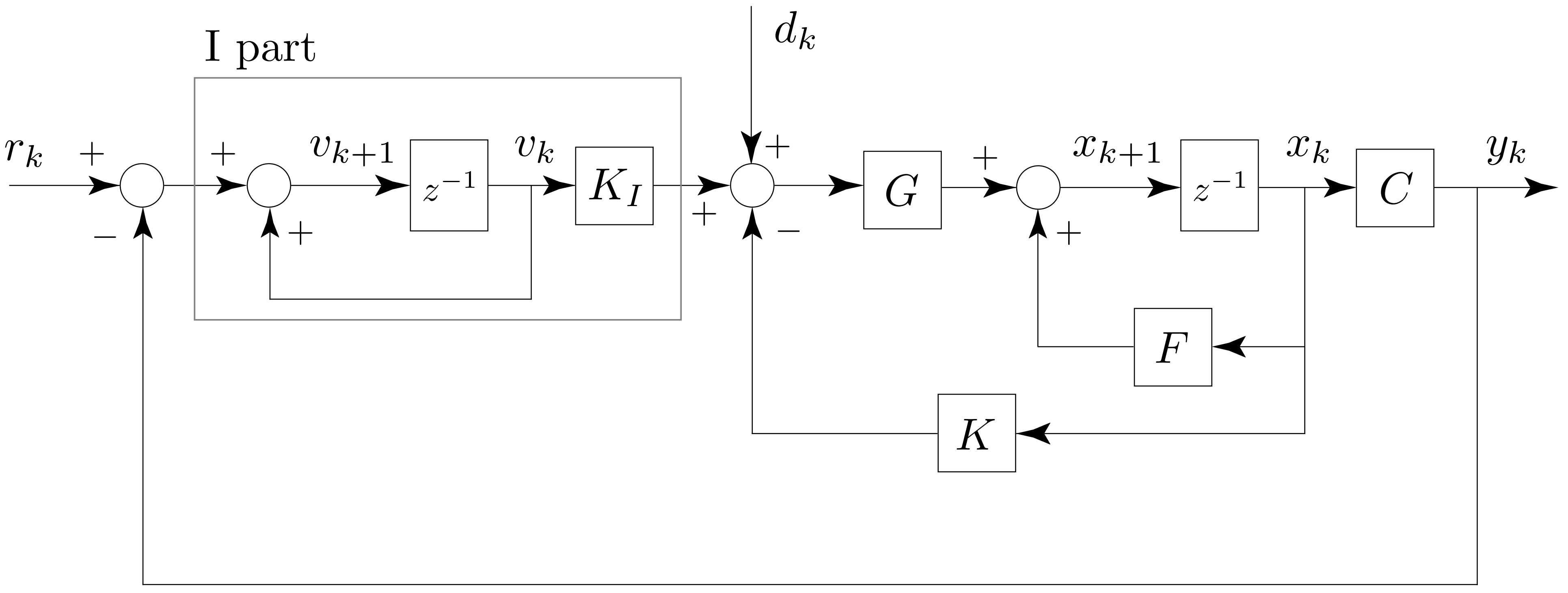
the integrator action can be weighted using the scalar parameter γ.

Using this formulation a solution $\widetilde{K}$ of the LQR problem for the augmented system $\{\widetilde{F}, \widetilde{G}, \widetilde{C}^T \cdot \widetilde{C}, \widetilde{R}\}$ can be found using equations (171) and (173). The resulting feedback matrix $\widetilde{K}$ is then partitioned as follows

$$\widetilde{K} = [-K_I \quad K] \tag{179}$$

and the two matrices $K_I \in \mathbb{R}^{m \times p}$ and $K \in \mathbb{R}^{m \times n}$ may be used as feedback gains as indicated in the next figure

LQR system with p additional integrators at the output.



State Reconstruction and Observer-Based Controllers

In general the state x of a system

$$x_{k+1} = F \cdot x_k + G \cdot u_k, \qquad y_k = C \cdot x_k \tag{180}$$

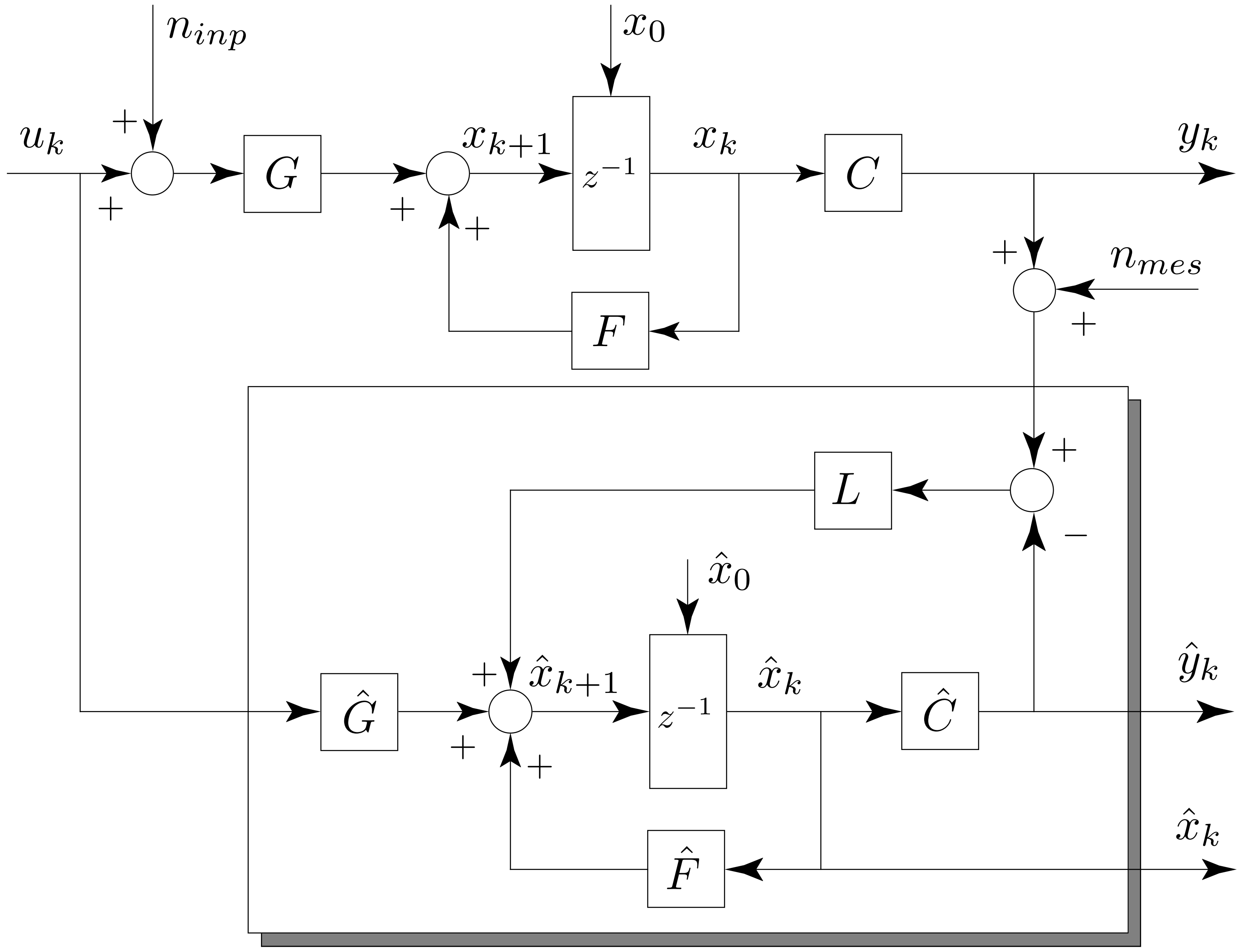
is not accessible and "observers" will have to be used to estimate this information. As in the continuous-time setting, these filters are built using a copy of the system's dynamics and the errors caused by the mismatched initial conditions $x_0 \neq \hat{x}_0$ are reduced by introducing output error feedback

$$\hat{x}_{k+1} = \hat{F} \cdot \hat{x}_k + \hat{G} \cdot u_k + L \cdot (y_k - \hat{y}_k), \qquad \hat{y}_k = \hat{C} \cdot \hat{x}_k \tag{181}$$

Neglecting all noise signals ($n_{inp} = n_{mes} = 0$) and assuming no plant/model mismatch ($\hat{F} = F$, $\hat{G} = G$, and $\hat{C} = C$), the dynamics of the error signal $e_k = x_k - \hat{x}_k$ are given by

$$e_{k+1} = [F - L \cdot C] \cdot e_k, \qquad e_0 = x_0 - \hat{x}_0 \tag{182}$$

Plant and observer block diagram, n_{inp} and n_{mes} are additive input and output noise signals.



The observer gain L must be chosen such that the matrix $F - L \cdot C$ has desired properties (stable, sufficiently fast, no unnecessary noise sensitivity, etc.).

If the measurement and input noises are Gaussian with known covariances, optimal estimators ("Kalman filters") may be designed. If this is not the case, any of the two methods introduced so far(eigenvalue placement and LQR) may be used with some rules of thumb as guidelines. One typical guideline is to choose the poles of the observer three times "faster" than those of the plant to be observed.

Of course, the ability to estimate (with asymptotically zero error) the state of the system (180) is quite an achievement. However, the full power of the observer concept (181) becomes clear only when such an observer is used in an output feedback control loop where the feedback gains K are taken from a fictitious state feedback design.

In this case, the combined system is described by the $2n$-dimensional difference equation

$$\begin{bmatrix} x_{k+1} \\ \\ \hat{x}_{k+1} \end{bmatrix} = \begin{bmatrix} F & -G \cdot K \\ \\ L \cdot C & F - G \cdot K - L \cdot C \end{bmatrix} \cdot \begin{bmatrix} x_k \\ \\ \hat{x}_k \end{bmatrix} + \begin{bmatrix} 0 \\ \\ -L \end{bmatrix} \cdot r_k \tag{183}$$

and by

$$y_k = \begin{bmatrix} C & 0 \end{bmatrix} \cdot \begin{bmatrix} x_k \\ \\ \hat{x}_k \end{bmatrix} \tag{184}$$

where the signal $r_k \in \mathbb{R}^p$ is the reference value for the output y_k.

Using the coordinate transformation

$$\begin{bmatrix} x_{k+1} \\ \\ \hat{x}_{k+1} \end{bmatrix} = T \cdot \begin{bmatrix} z_{k+1} \\ \\ \hat{z}_{k+1} \end{bmatrix}, \qquad \text{with} \qquad T = T^{-1} = \begin{bmatrix} I & 0 \\ \\ I & -I \end{bmatrix} \tag{185}$$

shows that the system matrix of (183) is similar to

$$\begin{bmatrix} F - G \cdot K & -G \cdot K \\ \\ 0 & F - L \cdot C \end{bmatrix} \tag{186}$$

and that therefore the poles of the system (183) are the poles of the original state-feedback design $(F - G \cdot K)$ and the observer design $(F - L \cdot C)$. This is known as the "separation principle." If stability only is to be analyzed, it permits the solution of the two problems of state feedback and observer design separately, without taking into consideration one problem when solving the other. It will be shown below that this is not true anymore when stability is not the only design criterion.

Notice that the error dynamics e_k are not independent of the reference signal r_k in the formulation (183). Of course, this is not desirable since the error should remain zero, once the influence of the mismatched initial conditions has been eliminated. To improve on this, the following feedforward additions may be introduced ("2-DOF controller").

$$\begin{bmatrix} x_{k+1} \\ \hat{x}_{k+1} \end{bmatrix} = \begin{bmatrix} F & -G \cdot K \\ L \cdot C & F - G \cdot K - L \cdot C \end{bmatrix} \cdot \begin{bmatrix} x_k \\ \hat{x}_k \end{bmatrix} + \begin{bmatrix} G \cdot \Gamma \\ \Lambda - L \end{bmatrix} \cdot r_k \tag{187}$$

where the two matrices $\Gamma \in \mathbb{R}^{m \times p}$ and $\Lambda \in \mathbb{R}^{n \times p}$ have to be chosen such that

1. The error dynamics do not depend on the reference signal r_k.
2. The static gain between reference and output is equal to $I_{p \times p}$.

Notice that this formulation is not trivial, i.e., the reference value has to be fed to the plant using the input matrix G while the input to the observer may be chosen completely arbitrarily (hence the matrix Λ). Notice also that only "square systems", i.e., in general only systems with the same number of inputs and outputs ($m = p$) can satisfy these two conditions.

Using these two conditions the following expression can be derived

$$\Lambda = G \cdot \Gamma + L, \qquad \Gamma^{-1} = C\,[I - (F - G \cdot K)]^{-1} \cdot G \tag{188}$$

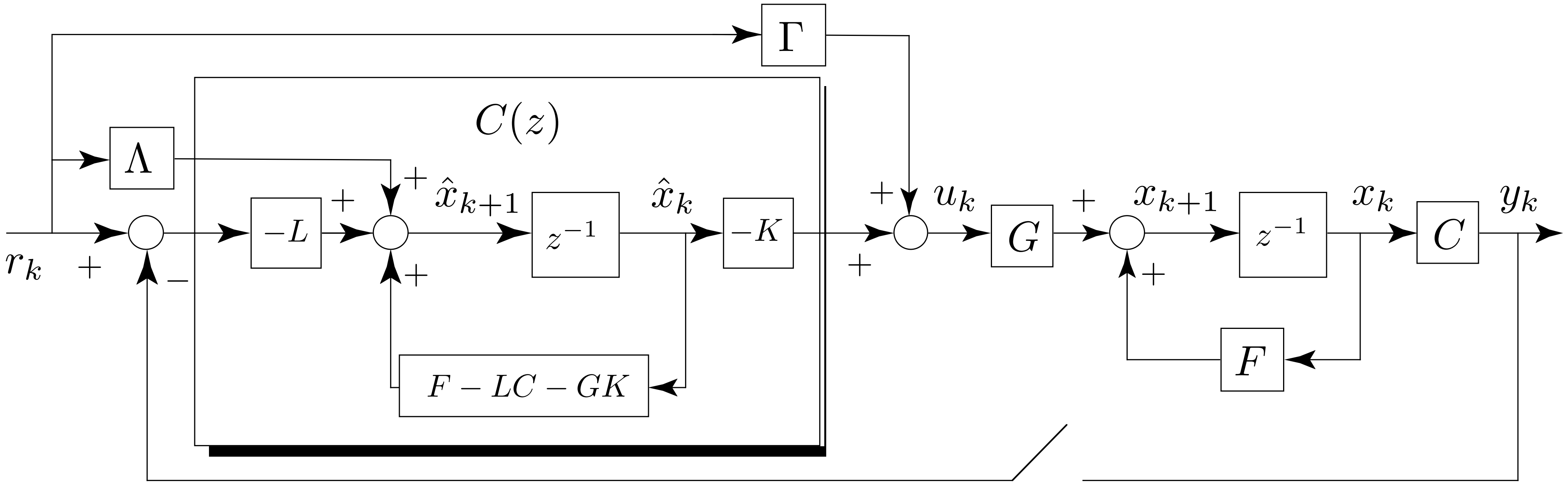
This choice guarantees a *static* gain of $I_{p \times p}$. With dynamic filters more sophisticated feedforward compensation schemes are possible, which improve the system behavior also for non-zero frequencies.

Notice that observer-based controllers are simply n-th order output-feedback controllers. A slight rearrangement of the closed-loop system diagram (as shown in the next figure) reveals that the open-loop gain $L(z)$ is given by

$$L(z) = C \cdot [zI - F]^{-1} \cdot G \cdot K \cdot [zI - (F - G \cdot K - L \cdot C]^{-1} \cdot L \tag{189}$$

This expression will be used below (in its SISO version) to analyze the robustness properties of control system designs.

Observer-based control system interpreted as an output-feedback controller, loop-breaking point indicated as a symbolic switch in the unity-feedback path.



LQG/LTR Methods

In the continuous-time setting the LQG-LTR method (Linear Quadratic Gaussian – Loop Transfer Recovery) is a well-known design approach which permits (at least for minimum-phase systems) the recovery of the excellent robustness properties of LQR designs for observer-based output feedback controllers.

The key idea is to use the optimal control formalism of LQR designs for the state feedback *and* the observer feedback design.

The LQG-LTR procedure comprises – in a nutshell – the following four steps:

1. Design a suitable feedback gain K (171) assuming that state feedback is feasible. Check the time domain (transients) and frequency domain (robustness) properties of this design and iterate on the weights Q and R if the design is not satisfactory. Add integrators (if necessary) according to the approach shown above.

2. Use the state-feedback solution in an observer-based output-feedback structure. Add a feedforward part (if necessary) as shown above.

3. The observer gain L is found by solving an LQR problem for the "dual system", i.e., for the fictitious system described by $\{F^T, C^T, G^T G, q \cdot I\}$.
4. Iterate on the observer gain by reducing the scalar q until sufficient robustness is reached without sacrificing too much of the noise reduction properties.

The following example shows what additional limitations are imposed by the fact that the controller is realized as a discrete-time system.

Fourth-Order SISO System and LQG-LTR Controller

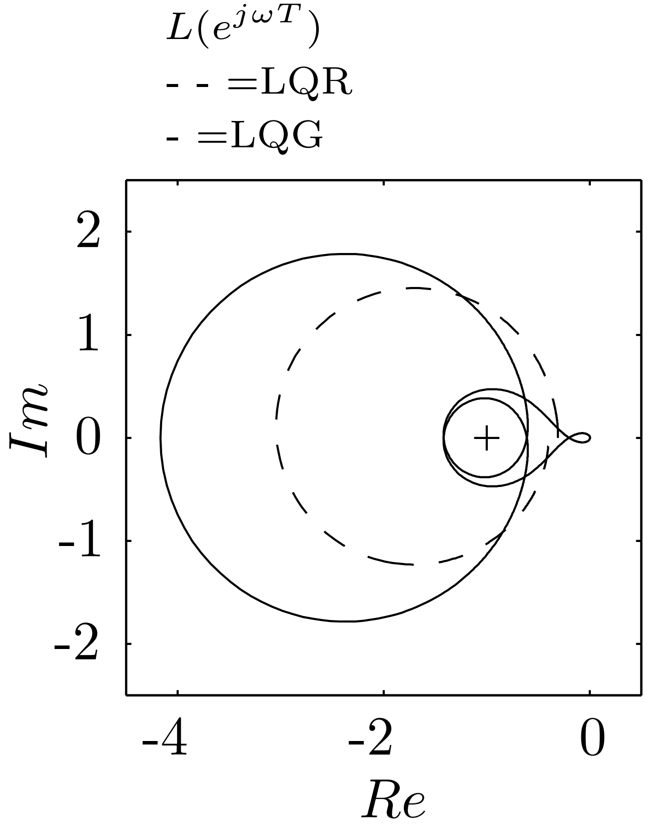
This example is a continuation of the LQR example in which a satisfactory state-feedback gain k was found for the system $\{F, g, c, 0\}$ described by (174). For that system an observer is designed according to the LTR procedure described in this section (a feedforward part is added). The weight Q is chosen as $Q = g \cdot g^T$ and the scalar q is varied. Three typical results are shown.

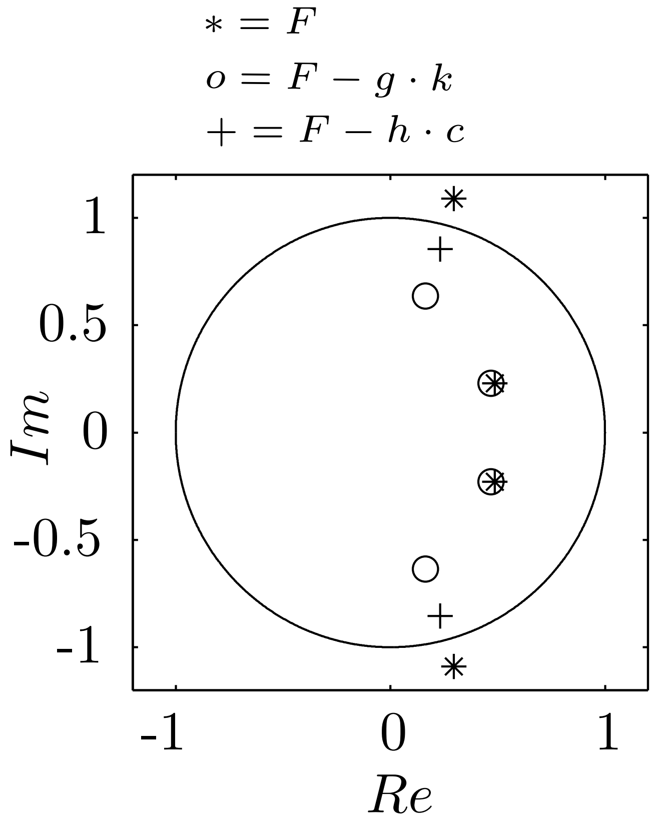
For $q \to \infty$ the transient behavior, i.e., the observation error dynamics $||e_k|| = ||x_k - \hat{x}_k||$ and the reference error dynamics $||r_k - y_k||$ are acceptable. However, the open-loop gain curve $L(e^{j\omega T})$, as introduced in equation (189), passes quite closely to the critical point, i.e., the system is rather "fragile" and small modeling errors are likely to cause severe performance degradations.

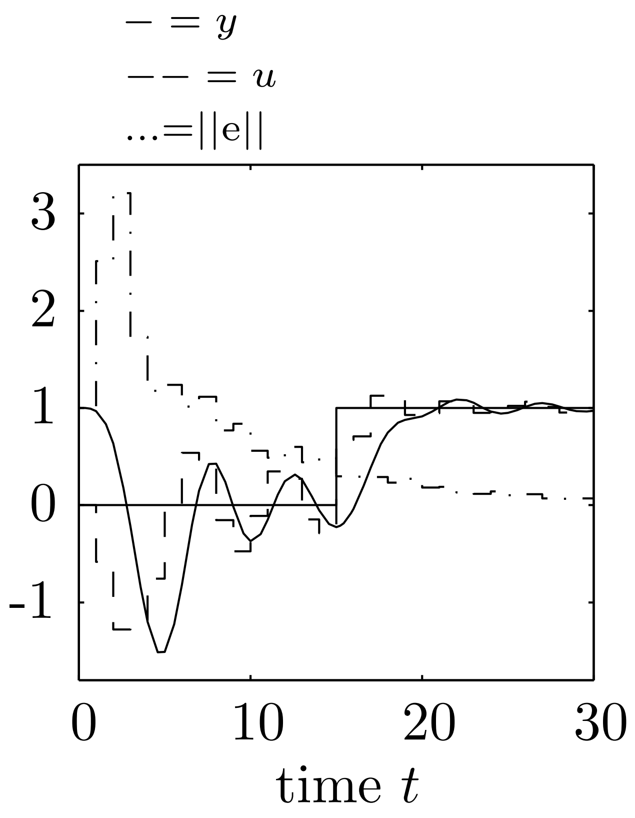
Letting $q \to 0$ improves the open-loop gain $L(e^{j\omega T})$ but produces an unacceptable transient behavior. Notice that even in this limiting case the LQG open-loop gain does not reach the LQR open-loop gain!

After some iterations an LTR gain $q = 0.05$ is chosen as a good compromise between robustness and transient behavior. The resulting open-loop gains, eigenvalues, and step responses are shown in the third figure.

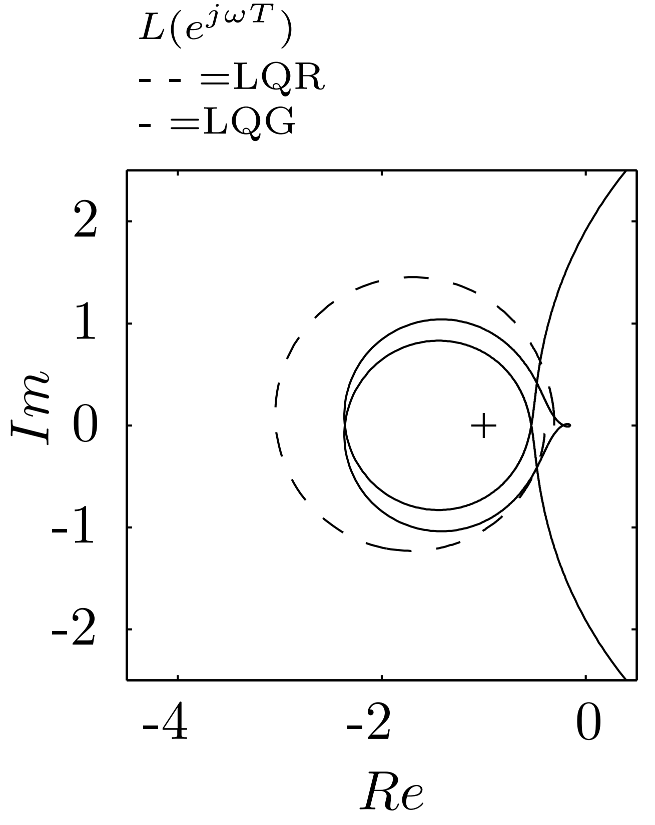
LQG-LTR design with $q \to \infty$

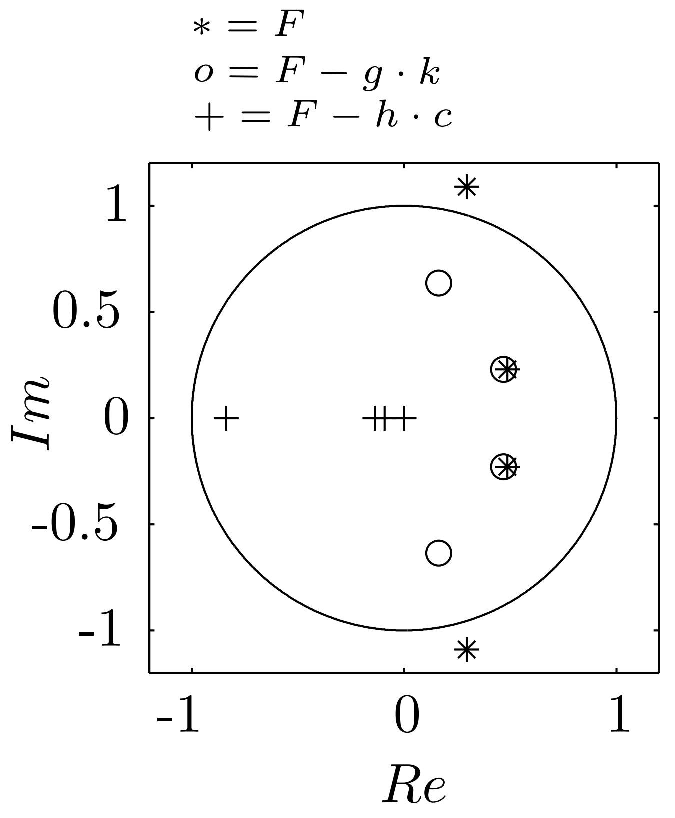


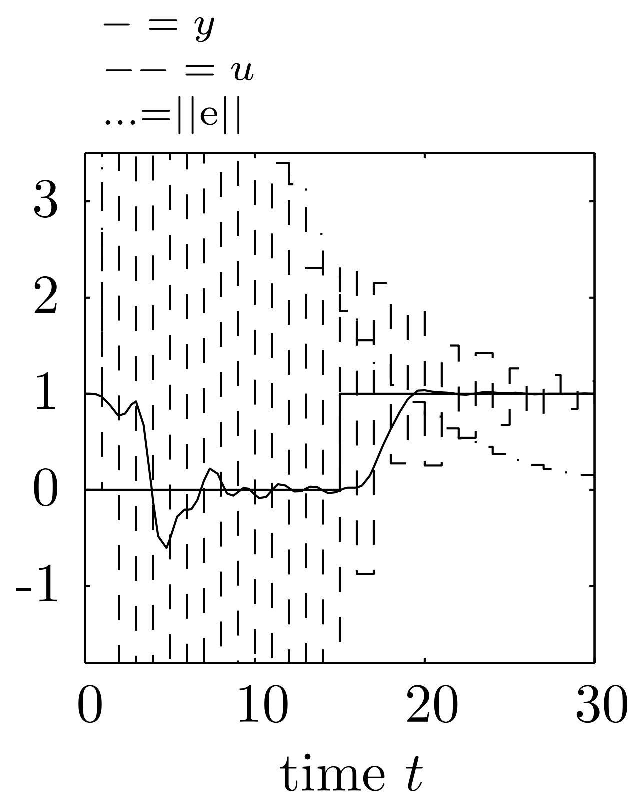




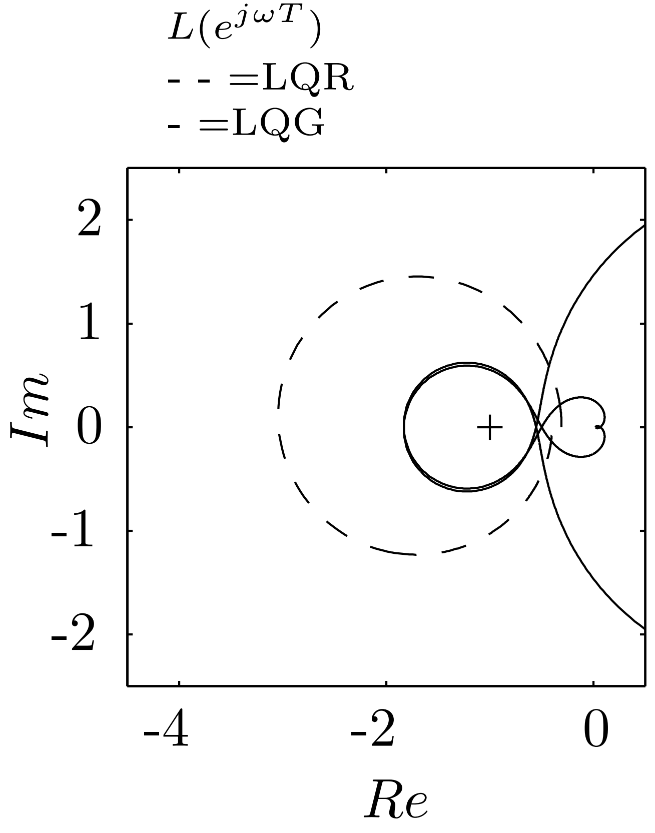
LQG-LTR design with $q \to 0$

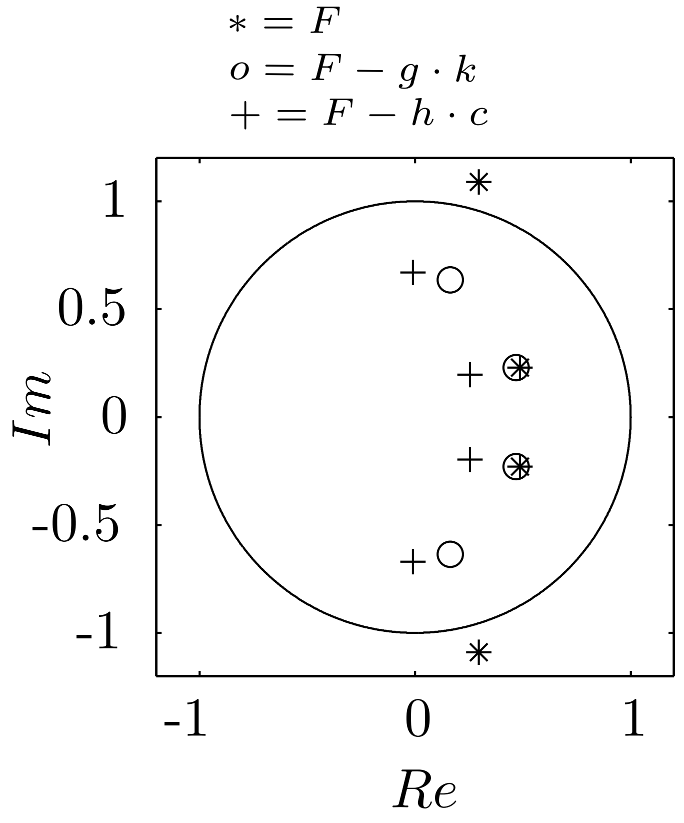


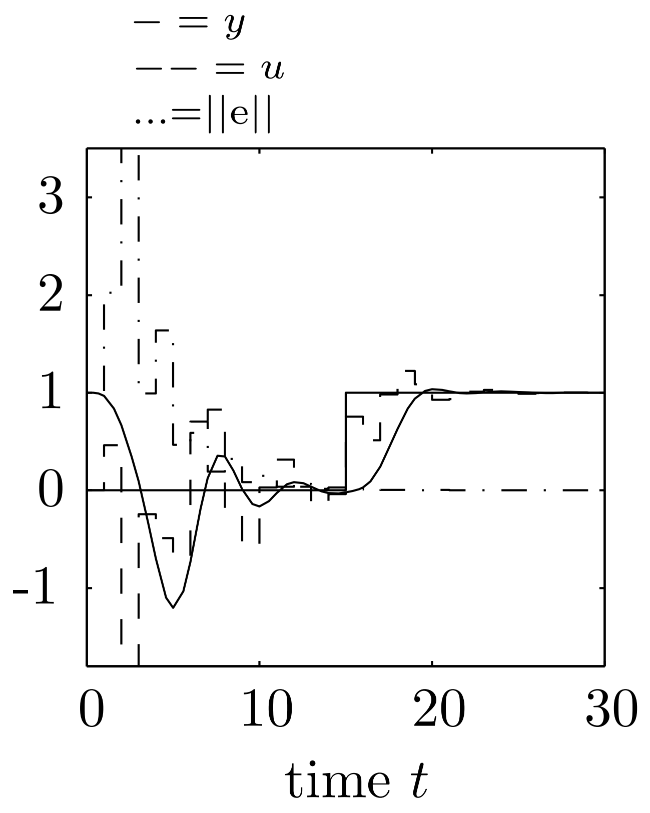




LQG-LTR design with $q = 0.05$


$L(e^{j\omega T})$
- - =LQR
- =LQG
Im
Re


$* = F$
$o = F - g \cdot k$
$+ = F - h \cdot c$
Im
Re


$- = y$
$-- = u$
...=||e||
time t
3
2
1
0
-1
0
10
20
30