Academic year
2025-2026

Faculty of Applied Engineering

Linear Control Theory

Text book

Walter Daems

Master of Science in Electronics and ICT Engineering Technology
Master of Science in de industriële wetenschappen: elektronica-ICT

2210FTIESY I-Electronic Systems
2212FTIESY I-Elektronische Systemen

This document has been typeset using LaTeX and the `uantwerpendocs` package.
Calculations have been performed using Matlab/Octave.
Graphics have been composed using PGF, TiKZ and InkScape.
All this material has been prepared on a GNU/Linux workstation.

Typesetting of this document was enabled by:

CT-2025-2.5

Contents

Preface

The field of the classical linear control theory, based on linear systems theory is well established for many years. Therefore, this text book has some great foundations in the work of my predecessor, Mark Vanpaemel, who relied heavily on the work of Richard Dorf and Robert Bishop on Modern Control Systems [DB11]. I started from Vanpaemel's text [Van18], translated it, updated some sections and added some exercises.

The advantage of this text book is that it is compact and focuses on the bare necessities. If you whish to study the matter in more detail, please consult the abundant literature.

I would like to thank many people who contributed to this text.
First, of course, Mark Vanpaemel, in whom I found a great teacher. I was one of his first pupils, back in 1993. Thanks for convincing me to go first into analog electronics instead of going head-first in digital electronics. I still benefit from the profound experience this journey gave me.
Secondly, a special thanks to my editor, Paul Levrie, for helping me by reviewing this text and supporting me with his profound experience, joyful humor and music.
Finally, my deepest gratitude goes to my beloved wife and children for enduring me devoting my time to writing this text, instead of spending my time with them.

Any contribution to this work is welcome. You won't get any money for it. I can only offer you my deepest appreciation. You can contact me by e-mail to `walter.daems@uantwerpen.be`.

I hope you enjoy discovering linear control theory!

Walter Daems
Summer 2025
Jordan Green, Norfolk (UK)

Symbol Table

Symbol	Meaning
Number sets	
$\mathbb{N}$	set of natural numbers (positive integer numbers)
$\mathbb{Z}$	set of integer numbers
$\mathbb{R}$	set of real numbers
$\mathbb{Q}$	set of rational numbers
$\mathbb{I}$	set of irrational numbers (real numbers that are not rational)
$\mathbb{C}$	set of complex real numbers
$\mathbb{X}^+$	set $\mathbb{X}$ restricted to positive numbers (not for $\mathbb{C}$)
$\mathbb{X}^-$	set $\mathbb{X}$ restricted to negative numbers (not for $\mathbb{C}$)
$\mathbb{X}_0$	set $\mathbb{X}$ with 0 excluded
Transform symbols - Forward	
$X[k] = \text{FS}(x(t))$	$X[k]$ are the harmonic numbers of the Fourier Series of $x(t)$
$X(\omega) = \mathcal{F}(x(t))$	$X(\omega)$ is the Fourier Transform of $x(t)$
$X(\omega) = \text{DtFT}(x[n])$	$X(\omega)$ is the Discrete-time Fourier Transform of $x[n]$
$X[k] = \text{DFT}(x[n])$	$X[k]$ is the Discrete Fourier Transform of $x[n]$
$X(s) = \mathcal{L}(x(t))$	$X(s)$ is the Laplace Transform of $x(t)$
$X(z) = \mathcal{Z}(x[n])$	$X(z)$ is the Z-Transform of $x[n]$
Transform symbols - Inverse	
$x(t) = \text{FS}^{-1}(X[k])$	C_k are the harmonic numbers of the Fourier Series of $x(t)$
$x(t) = \mathcal{F}^{-1}(X(\omega))$	$X(\omega)$ is the Fourier Transform of $x(t)$
$x[n] = \text{DtFT}^{-1}(X(\omega))$	$X(\omega)$ is the Discrete-time Fourier Transform of $x[n]$
$x[n] = \text{DFT}^{-1}(X[k])$	$X[k]$ is the Discrete Fourier Transform of $x[n]$
$x(t) = \mathcal{L}^{-1}(X(s))$	$X(s)$ is the Laplace Transform of $x(t)$
$x[n] = \mathcal{Z}^{-1}(X(z))$	$X(z)$ is the Z-Transform of $x[n]$
Mapping symbols	
$x(t) \xrightarrow{\text{FS}} X[k]$	$X[k]$ are the harmonic numbers of the Fourier Series of $x(t)$

continued on next page

continued from previous page

Symbol	Meaning
$x(t) \xrightarrow{\mathcal{F}} X(\omega)$	$X(\omega)$ is the Fourier Transform of $x(t)$
$x[n] \xrightarrow{\text{DtFT}} X(\omega)$	$X(\omega)$ is the Discrete-time Fourier Transform of $x[n]$
$x[n] \xrightarrow{\text{DFT}} X[k]$	$X[k]$ is the Discrete Fourier Transform of $x[n]$
$x(t) \xrightarrow{\mathcal{L}} X(s)$	$X(s)$ is the Laplace Transform of $x(t)$
$x[n] \xrightarrow{\mathcal{Z}} X(z)$	$X(z)$ is the Z-Transform of $x[n]$
Linear (time-domain) operations	
$x[n] \star y[n]$	convolution of $x[n]$ and $y[n]$
$x[n] \star y[n]$	correlation of $x[n]$ and $y[n]$
Domain conversion blocks	
DAC	Digital to Analog Converter
ADC	Analog to Digital Converter

Chapter 1

Linear Models

In this chapter, we will illustrate how to compose a linear mathematical model for a system in the Laplace domain, starting from its governing physics laws. By analyzing the full block diagram we will be able to generate an overall transfer function for the entire system. Mason's rule will prove to be a great asset in doing so.

If you're not familiar (anymore) with linear systems theory (and how these systems can be described in the Laplace domain), you'd better take some time first to review that subject.

After having read/studied this chapter, you are expected to be able to

- compose a block diagram of a system, given the physics laws that govern it.
- analyze such a system, either by block diagram transformations or by converting it to a signal flow graph (SFG) and using Mason's rule.

1.1 Composing a block diagram for a linear system

1.1.1 Principle

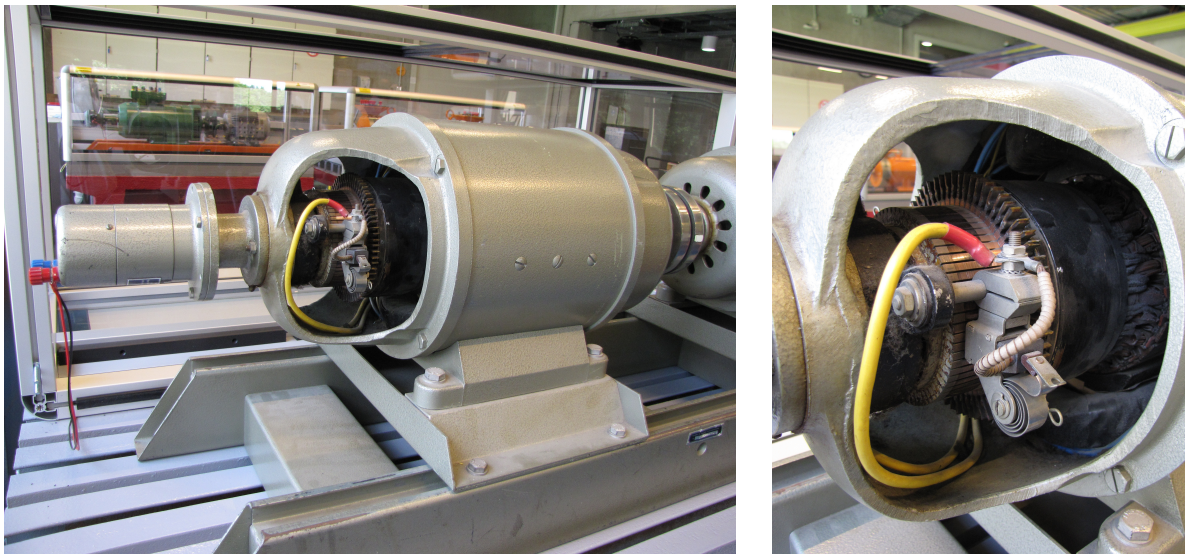
It is easy to compose block diagrams in the Laplace domain starting from basic physics equations that are formulated in terms of differential equations. Remember, as we are working in the Laplace domain, we assume all signals to be causal (i.e. zero for negative time points). Given the fact that we want to compose block diagrams for linear systems, we only need to be able to model linear operations: summation, scaling and derivation/integration.

Therefore, composing a block diagram corresponds to the following mapping:

$$y(t) = Kx(t) \longrightarrow x(t) \longrightarrow \boxed{K} \longrightarrow y(t)$$

$$z(t) = x(t) \pm y(t) \longrightarrow \begin{array}{l} x(t) \\ y(t) \end{array} \longrightarrow \oplus \longrightarrow y(t), \quad \pm \text{ on } y(t)$$

$$\frac{\mathrm{d}y(t)}{\mathrm{d}t} = x(t) \quad \text{or} \quad y(t) = \int_0^t x(t)\,\mathrm{d}t \longrightarrow x(t) \longrightarrow \boxed{\frac{1}{s}} \longrightarrow y(t)$$



(a) Overview (b) The collector in detail

Figure 1.1: Direct current (DC) motor with the housing opened, such that one can see the collector (images courtesy of G. De Winter). In the overview image, on the left the machine is hooked up to a tachometer and on the right to a load. The detailed images shows what is a true open collector (and not the one of a bipolar transistor).

1.1.2 Example: a DC motor

We will use an example to guide us through the process of composing a block diagram. Consider the direct current (DC) motor of Figure 1.1. It is not our goal to understand the complex physics of this motor (with commutation and magnetic saturation effects). Instead, we just want to introduce a simple linear behavioral model that grasps the basic motor operation. This linear model will be our vehicle throughout this chapter and also will be used later on in the course.

A simple electric schematic of this motor can be found in Figure 1.2.

The DC motor converts electrical energy into rotational mechanical energy. The motor consists of a non moving part, the stator, that is fixed to the yoke of the machine (the body) and a rotating part, the rotor (or anchor).

On the stator, field coils generate a magnetic field. The generate magnetic flux is proportional to the field current. In the Laplace domain, we can write:

$$\phi(s) = K_f I_f(s)$$

Obviously, taking into account the series (wire) resistance R_f of the field coil with inductance L_f, we can write:

$$I_f(s) = \frac{V_f(s)}{L_f s + R_f} \tag{1.1}$$

On the rotor, anchor windings allow for an anchor current that will result in a Lorentz force yielding a motor torque $T_m(s)$ according to

$$T_m(s) = K_a \phi(s) I_a(s) = K_a K_f I_f(s) I_a(s) \tag{1.2}$$

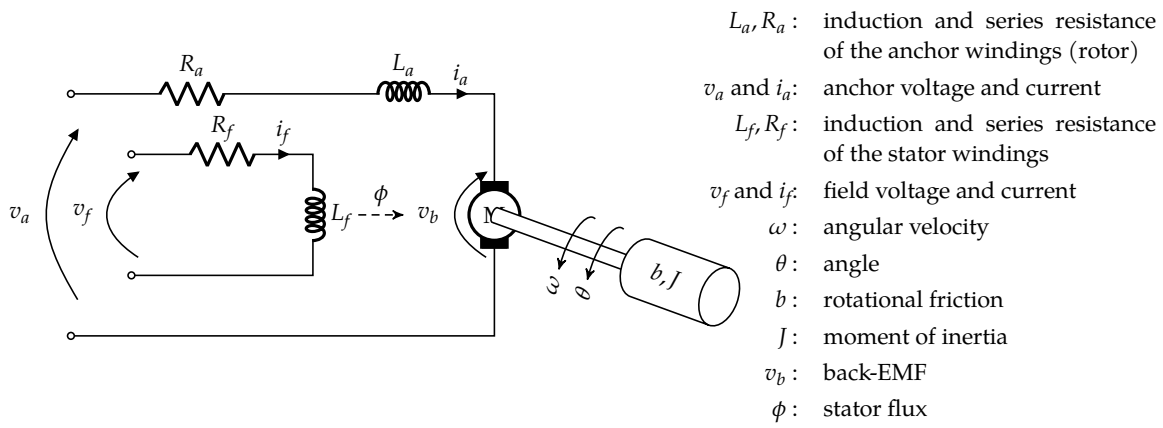


Figure 1.2: Simple electric schematic of a DC motor

The anchor loop looks similar to the field loop, and also consists of a coil with inductance L_a and a series (wire) resistance R_a. However, when the motor starts rotating, it will act as a DC generator and produce a *back electromagnetic force* (*back-EMF*), proportional to its angular velocity:

$$V_b(s) = K_b\omega$$

and therefore:

$$I_a(s) = \frac{V_a(s) - V_b(s)}{L_a s + R_a} = \frac{V_a(s) - K_b\omega}{L_a s + R_a} \tag{1.3}$$

The generated motor torque $T_m(s)$ will attempt to rotate the shaft that is hooked up to the anchor. Of course, the construction of anchor, shaft and whatever is attached to the shaft, will have a certain moment of inertia J and a particular friction b (proportional to the angular velocity of the motor). According to Newton's second law for rotational movements, this represents a net load torque

$$T_l(s) = Js\omega(s) + b\omega(s) \tag{1.4}$$

Note that in the Laplace domain (assuming zero initial conditions) $s\omega(s)$ represents the angular acceleration (derivative of the angular velocity).

In addition to the nominal load (that is incorporated in J and b), the load and the friction may also vary. This effect is represented by a *dynamic torque* term $T_d(s)$, such that

$$T_m(s) = T_l(s) + T_d(s) \tag{1.5}$$

In the remainder of this text, we will use the following values for our DC motor example:

Parameter	Value	Parameter	Value
K_f	2 Wb/A	K_a	2 Nm/Wb
L_f	1 H	L_a	1 H
R_f	10 Ω	R_a	10 Ω
J	10 Nm/(rad/s^2)	b	10 Nm/(rad/s)

1.1.3 Composing block diagrams for the DC motor

Basically, there are two modes to control a DC motor:

- *field control mode*, keeping the anchor current constant
- *anchor control mode*, keeping the field current constant

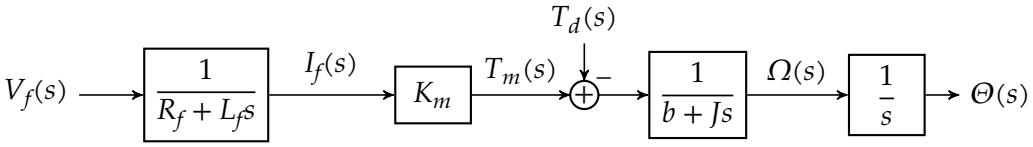
We will compose block diagrams for both of them.

Field control mode

Assuming the anchor current to be constant (equal to I_a), we can rewrite (1.2) as

$$T_m(s) = K_m I_f(s) \qquad \text{with } K_m = K_f K_a I_a$$

The combination of this equation with equations (1.1), (1.4) and (1.5) leads to the block diagram below:



in which we assumed $\omega(t) = \frac{\mathrm{d}\theta(t)}{\mathrm{d}t}$.

Given this block diagram it is easy to determine the transfer function relating $\Omega(s)$ to $V_f(s)$. One can easily find that:

$$H(s) = \frac{\Omega(s)}{V_f(s)} = \frac{K_m}{(L_f s + R_f)(Js + b)} = \frac{\frac{K_m}{bR_f}}{(\tau_f s + 1)(\tau_l s + 1)}$$

with

$$\tau_f = \frac{L_f}{R_f} \qquad \tau_l = \frac{J}{b}$$

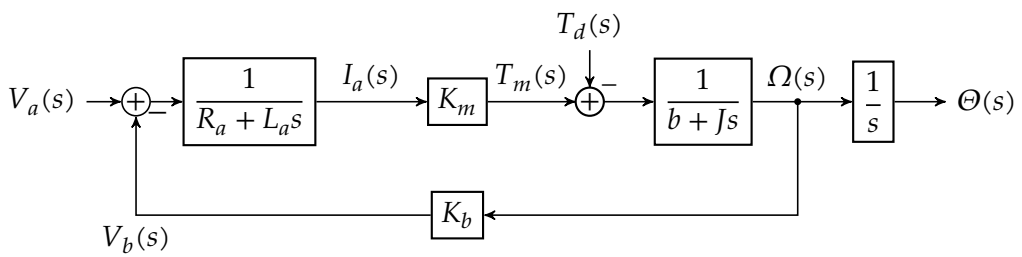
In the remainder of this text we will assume $I_a = 5\,\mathrm{A}$ such that $K_m = 20\,\mathrm{Nm/A}$.

Anchor control mode

Assuming the field current to be constant (equal to I_f), we can rewrite (1.2) as

$$T_m(s) = K_m I_a(s) \qquad \text{with } K_m = K_a K_f I_f$$

The combination of this equation with equations (1.1), (1.4), (1.5) and (1.3) leads to the block diagram below:



in which we again assumed $\omega(t) = \frac{\mathrm{d}\theta(t)}{\mathrm{d}t}$.

Given this block diagram it is easy to determine the transfer function relating $\Omega(s)$ to $V_a(s)$. In the next section we will see how to do this by block diagram manipulation or using Mason's rule. For now, take for granted that:

$$H(s) = \frac{\Omega(s)}{V_a(s)} = \frac{K_m}{(L_a s + R_r)(Js + b) + K_b K_m} = \frac{\frac{K_m}{bR_a}}{(\tau_a s + 1)(\tau_l s + 1) + \frac{K_b K_m}{bR_a}}$$

with

$$\tau_a = \frac{L_a}{R_a} \qquad \tau_l = \frac{J}{b}$$

In the remainder of this text we will assume $I_f = 5\,\mathrm{A}$ such that $K_m = 20\,\mathrm{Nm/A}$.

Remarks

1. We can prove that $K_b = K_m$, based on the fact that the electric power should be fully converted into mechanical power:

$$\begin{aligned} P_m &= P_e \\ T_m \cdot \omega &= v_b \cdot i_a \\ &\downarrow v_b = K_b \cdot \omega \text{ and } T_m = K_m \cdot i_a \\ K_b \cdot \omega \cdot i_a &= K_m \cdot i_a \cdot \omega \\ K_b &= K_m \end{aligned}$$

2. Note how in anchor control mode, the back-emf introduces a feedback loop in the block diagram. Of course, in field control mode, the back-emf is also present, but its effect is suppressed by the fact that we assume the anchor current to be constant (independent of the back-emf).

1.2 Calculating the transfer function of a block diagram

We can derive the transfer function of a block diagram in two major ways:

- using block diagram transformations,
- using Mason's rule.

For the latter, we will also study the technique of converting a block diagram into a signal flow graph (SFG).

As a test case, we will use the example of Figure 1.3.

One can verify that this block diagram corresponds to the following equations:

$$\begin{aligned} Q(s) &= X(s) + G_8 Y(s) \\ R(s) &= G_1 Q(s) - G_2 G_6 R(s) \\ U(s) &= G_2 R(s) + G_7 Y(s) \\ V(s) &= G_3 U(s) + G_5 X(s) \\ Y(s) &= G_4 T(s) \end{aligned}$$

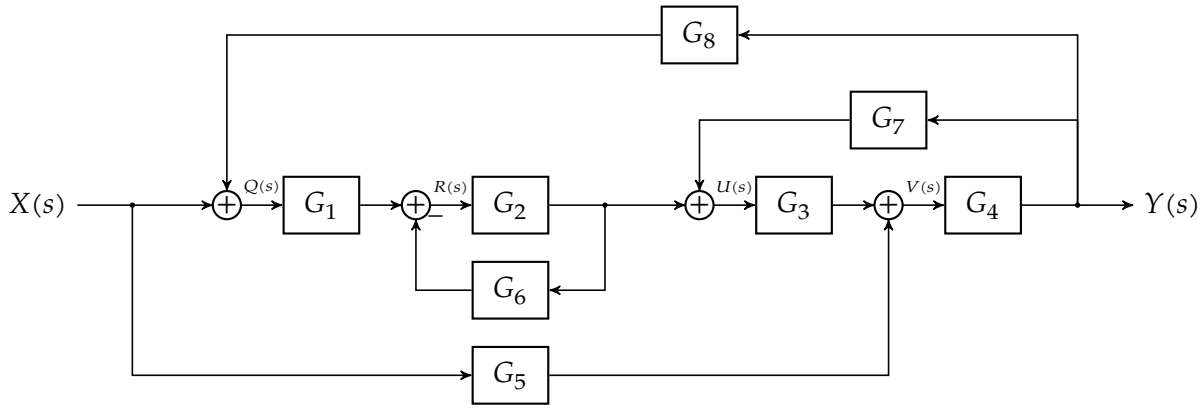


Figure 1.3: Example block diagram

One could find the transfer from $X(s)$ to $Y(s)$ by solving this system of equations for $H(s) = Y(s)/X(s)$. However, it is easier using block diagram transformations.

1.2.1 Using block diagram transformations

Basic principles

The basic idea is that we will try to simplify the block diagram using the equivalencies of Table 1.1 on page 8. You can easily prove these equivalencies by manipulating the equations they represent.

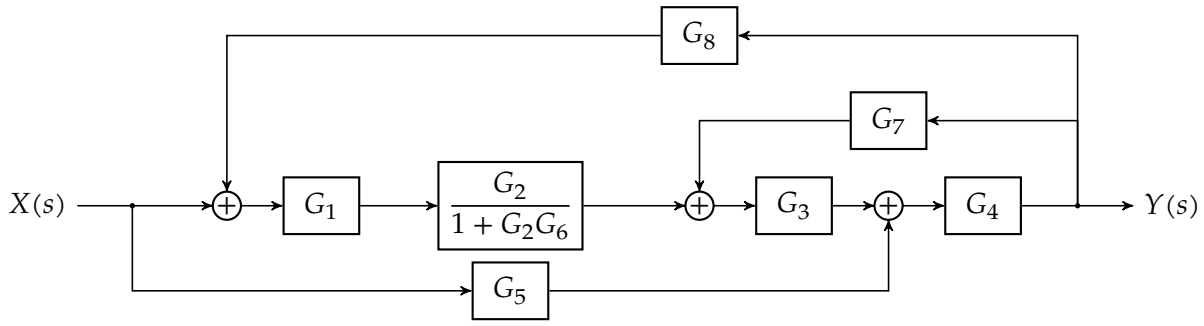
Using these equivalencies, we can simplify a diagram by joining cascades, collapsing parallel paths and removing loops by shifting summation and tapping points. An example will make things clear.

In rare cases one also needs two less common rules, that allow interchanging a summation and a tapping point. These have been listed in Table 1.2 on page 9.

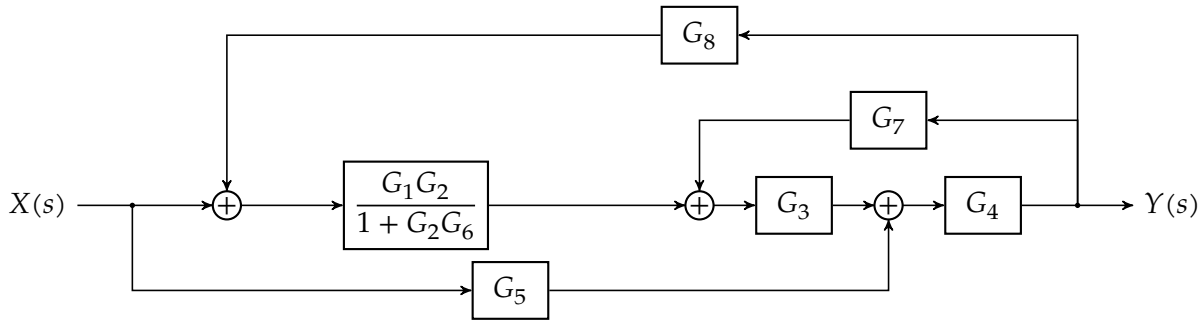
Example

Let's apply these transformations on the example of Figure 1.3. It contains multiple forward paths and many loops. The issue is that these paths and loops are entangled. Our goal is to untangle them.

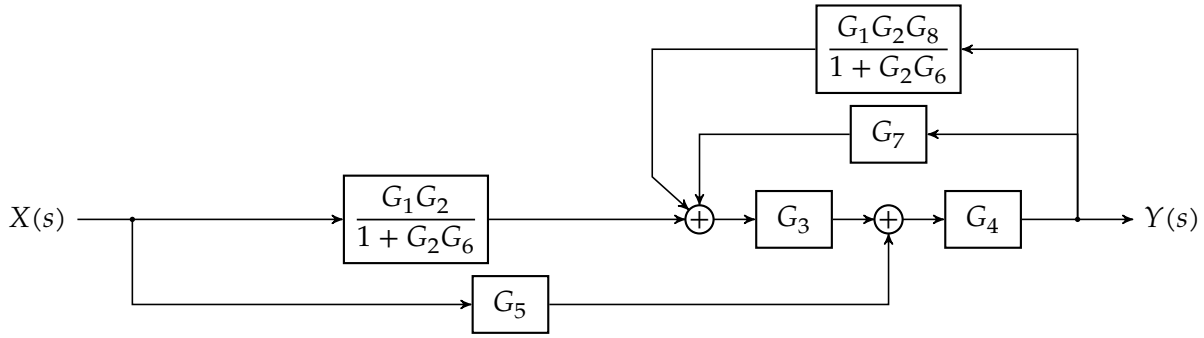
To start, there is one feedback loop that can be solved straightaway, i.e. the loop containing G_6:



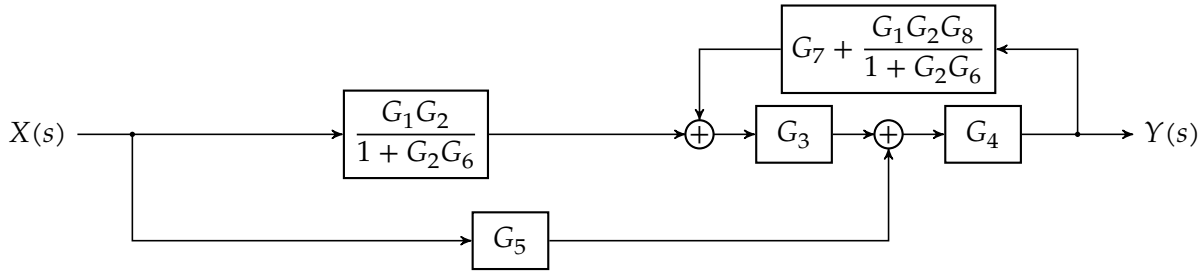
As a logical next step, we can join the two cascaded blocks in the main forward path:



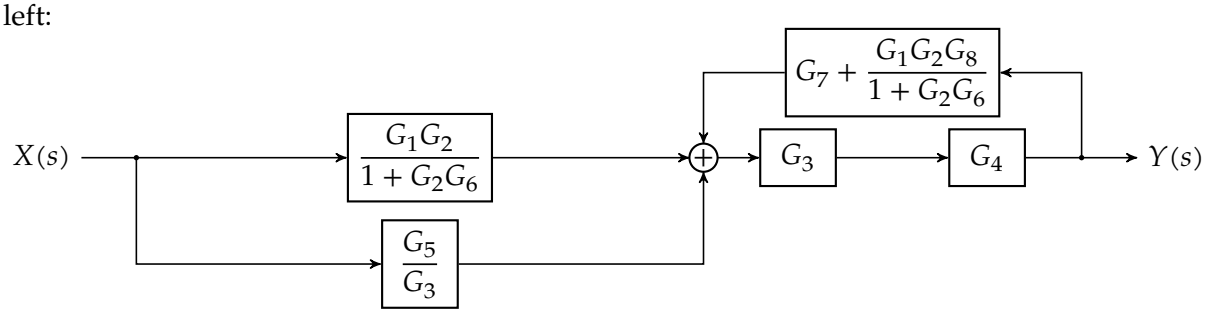
Next, we can shift the leftmost summation to the right:



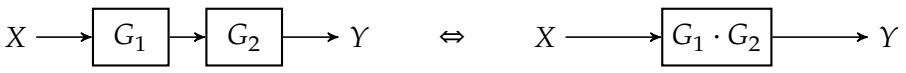
This allows for collapsing the two top feedback loops as they are parallel paths:



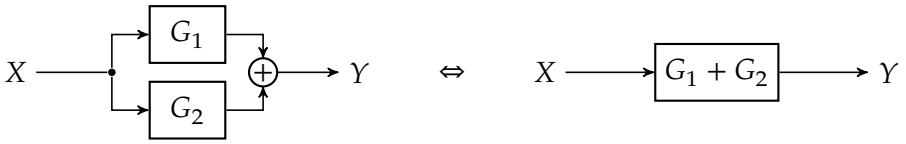
In a next step, we can avoid the forward path through G_5 from interfering with the feedback loop, by shifting the summation that joins that forward path to the main forward path to the left:



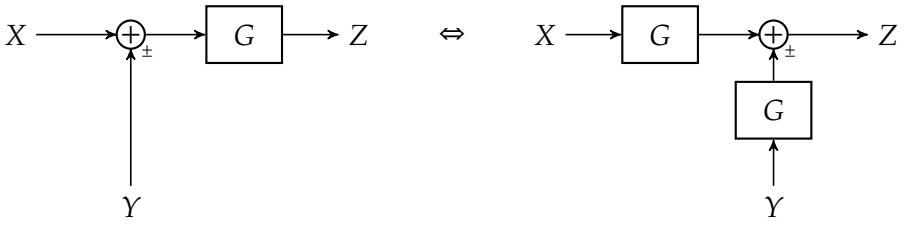
Cascade:



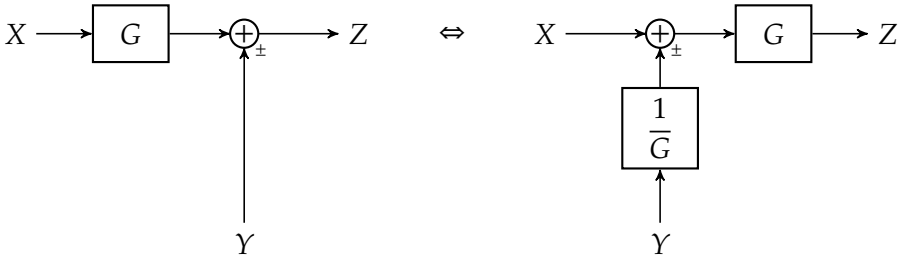
Parallel paths:



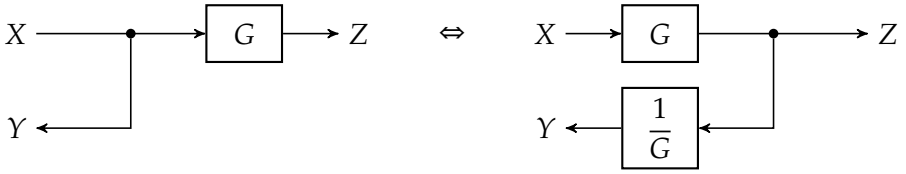
Moving summation point ahead:



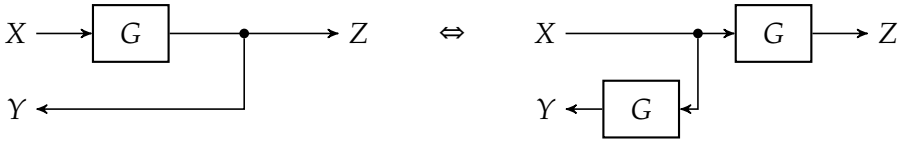
Moving summation point back:



Moving tapping point ahead:



Moving tapping point back:



Eliminating a feedback loop:

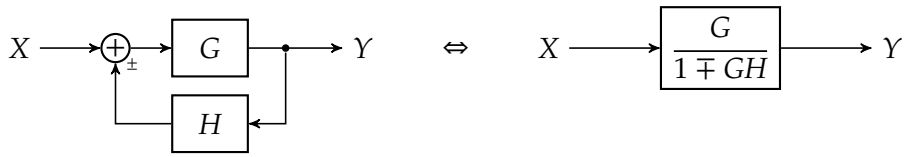
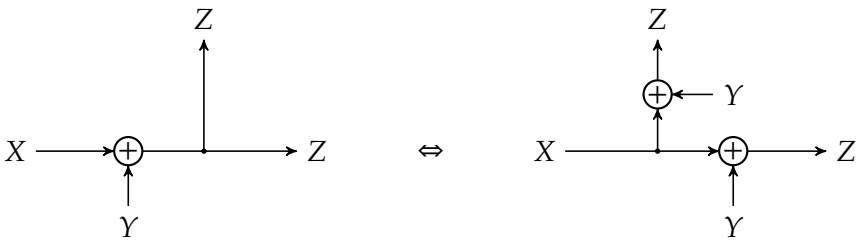


Table 1.1: Basic block diagram equivalency rules to simplify block diagrams

Summation/Tapping point interchange



Tapping point/Summation interchange

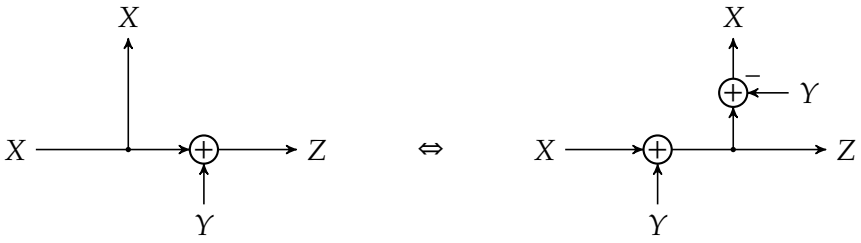
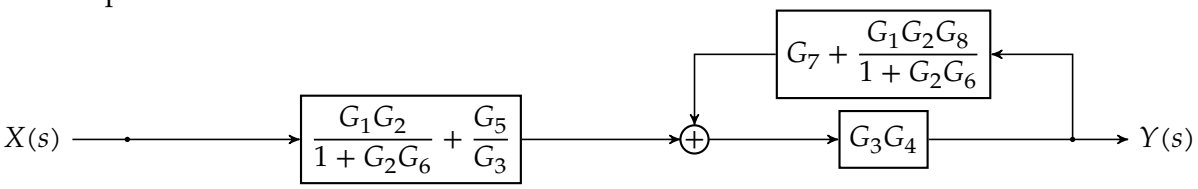
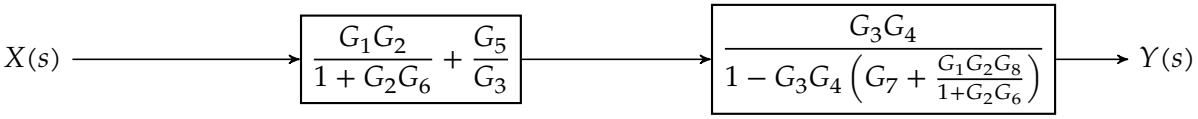


Table 1.2: Advanced block diagram equivalencies rules to simplify block diagrams

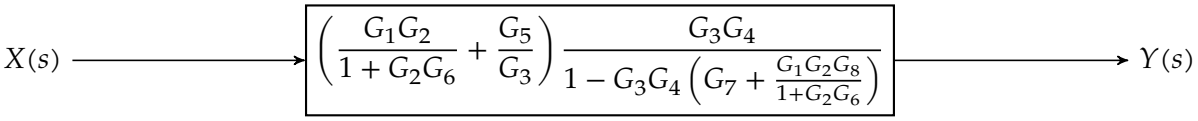
Now, we can join the two parallel forward paths and join the cascade in the second half of the forward path:



And, nearly reaching the end, we can eliminate the feedback loop:



Finally, we can join the remaining cascade:



Of course, this result can be further simplified:

$$\begin{aligned} H(s)=\frac{Y(s)}{X(s)} &= \left(\frac{G_1G_2}{1+G_2G_6}+\frac{G_5}{G_3}\right)\frac{G_3G_4}{1-G_3G_4\left(G_7+\frac{G_1G_2G_8}{1+G_2G_6}\right)} \\ &= \frac{G_1G_2G_3+G_5(1+G_2G_6)}{(1+G_2G_6)\cancel{G_3}}\,\frac{\cancel{G_3}G_4}{1-G_3G_4\left(G_7+\frac{G_1G_2G_8}{1+G_2G_6}\right)} \\ &= \frac{G_1G_2G_3G_4+G_4G_5+G_2G_4G_5G_6}{1+G_2G_6-G_3G_4G_7-G_2G_3G_4G_6G_7-G_1G_2G_3G_4G_8} \end{aligned} \tag{1.6}$$

Though simpler than solving the equations straight away[1], it still took a considerable amount of time. Mason's rule will solve this.

1.2.2 Mason's rule

1.2.2.1 Transforming the block diagram to a signal flow graph

We could formulate Mason's rule based on block diagrams. However, we're less prone to errors if we redraw the system as a *signal flow graph*, and you are highly recommended to always take this intermediate step.

How do we transform a block diagram in to a signal flow graph?

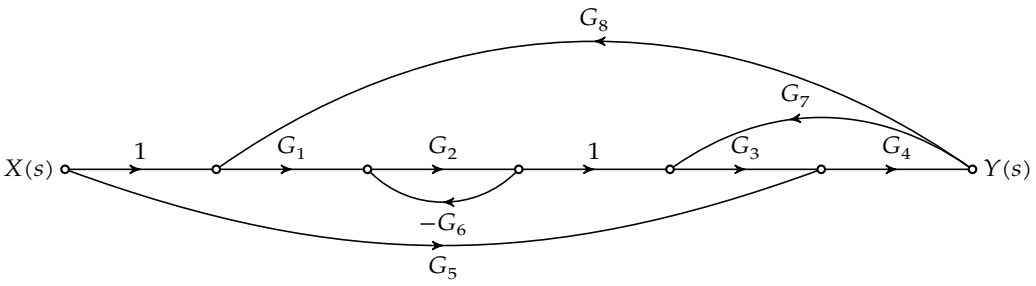
- Signals become nodes.
- Blocks become directed edges with a corresponding gain factor.
- Summations become nodes in which multiple edges arrive (in fact all edges entering a node are assumed to be summed).

[1] You are encouraged to do this on a rainy day.

- If there is a tapping point before a summation, this tapping point gets its own node followed by an edge with gain 1 to the summation node.
- The input of the graph is called *the source*, and the output *the sink*.

A node is usually drawn as a bullet, and a directed edge as an arrow in between two nodes.

If we do this for our example, then this is the result:



1.2.2.2 Graph terminology

Before we can state Mason's rule it is important that you master a number of terms/concepts related to graphs:

- source and sink
- path, forward path, non-touching paths
- loop, non-touching loops

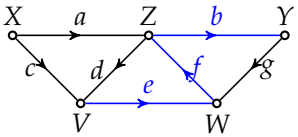
The source and sink

The input variable of a graph is the source. The output variable is the sink.
As the input variable is imposed on the graph, any arriving edge to the input node is to be discarded as it is overpowered by the set input variable.

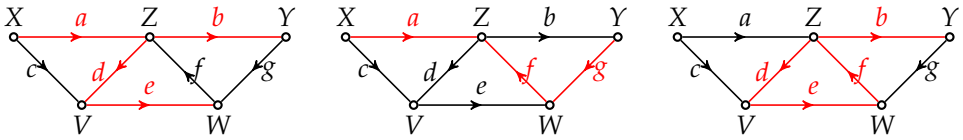
A path

A path is a sequence of edges starting from a chosen node and ending in a (different) chosen node that never visits a node twice.

E.g. the blue path below from V to Y:



but not the red paths below:

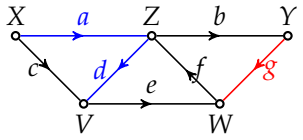


The product of all gains in a path is called the path gain (or transmission). In the (blue) case above, this is $T = efb$.

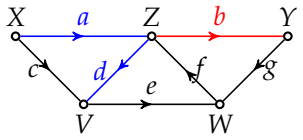
Non-touching paths

Two paths are non-touching if they have no nodes in common.

E.g. the blue and red path below:



but not:

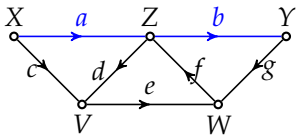


Automatically, non-touching paths also cannot share any edges.

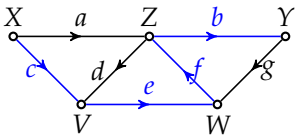
A forward path

This is a path that takes you from the source to the sink.

E.g. The blue paths below (with indicated path gain or transmission):

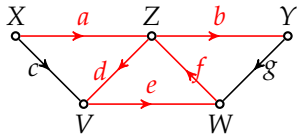


$T_{f1} = ab$



$T_{f2} = cefb$

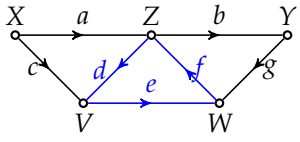
but not the red path below (because it is no path):



A loop

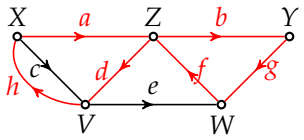
This is a path with identical start and end-point.

E.g. the blue path below (with indicated path gain or transmission):



$T_l = cde$

but not the red path below (because it is no path):



1.2.2.3 Mason's rule

Now that we know the basic terminology related to signal flow graphs, we can formulate Mason's rule (without proof):

Mason's rule

1. Remove all edges arriving at the source.
2. The path gain / transmission from source to sink can be calculated as:

$$T = \frac{\sum_k T_k \Delta_k}{\Delta}$$

with:

- k a counter covering all forward paths with transmission T_k
- Δ the graph's determinant
- Δ_k the graph's determinant with the k^{th} forward Δ path fully removed (including its nodes!)

Δ_k is also called the co-factor of the k^{th} path.

Then what is the graph's determinant?

Graph's determinant

$$\Delta = 1 - \sum_n L_n + \sum_{p,q} L_p L_q - \sum_{r,s,t} L_r L_s L_t + \sum_{a,b,c,d} L_a L_b L_c L_d - \cdots$$

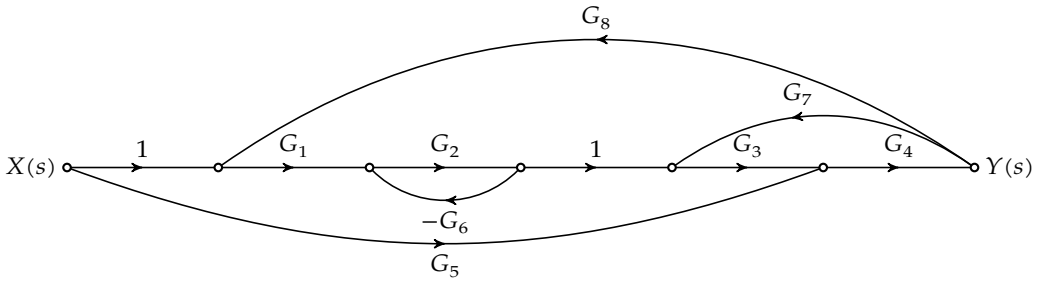
with

- n covering all loops
- p and q covering all pairs of non-touching loops
- r, s and t covering all triplets of non-touching loops
- a, b, c and d covering all quadruplets of non-touching loops
- ...

1.2.2.4 Applications of signal flow graphs and Mason's rule

Determining transfer functions of signal flow graphs

Let's again consider the signal flow graph corresponding to Figure 1.3 and let's try to calculate the transmission between $X(s)$ and $Y(s)$, i.e. let's try to calculate the transfer function $H(s) = Y(s)/X(s)$.



This signal flow graph contains two forward paths between source and sink. Let's analyze them and also calculate the graph's determinant.

First forward path

This is the dead-straight path from X to Y. The gain of this path is $T_1 = G_1G_2G_3G_4$.

To calculate the corresponding cofactor, we first remove all nodes and edges that compose the forward path, from the graph. If you remove a node from the graph, you also need to remove all edges that are attached to this node.

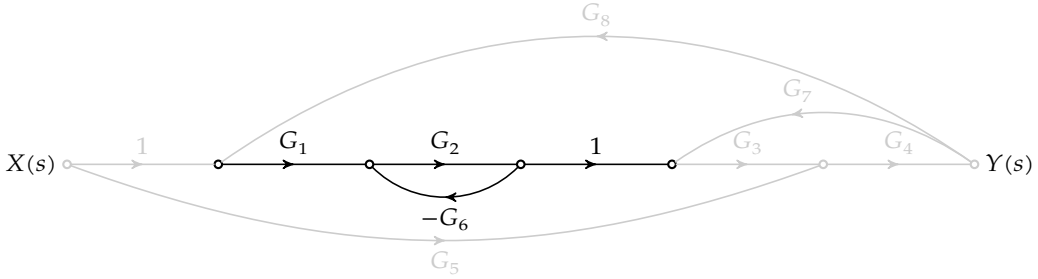
Given the fact that the considered forward path covers all nodes, after removing them, there is nothing of the graph left. Hence there are no loops and therefore:

$$\Delta_1 = 1$$

Second forward path

This part starts at X, then through G_5 and G_4 to Y. The gain of this path is $T_2 = G_4G_5$.

To calculate the cofactor, we again remove all nodes and edges from the graph that compose the considered forward path. The result can be found below. All removed elements have been grayed out.



Given the fact that there is only one loop with transmission $-G_2G_6$ remaining, the cofactor equals:

$$\Delta_2 = 1 + G_2G_6$$

Graph's determinant

The graph's determinant equals:

$$\Delta = 1 - \sum_{n} L_n + \sum_{p,q} L_pL_q - \sum_{r,s,t} L_rL_sL_t + \sum_{a,b,c,d} L_aL_bL_cL_d - \cdots$$

Let's start by identifying the individual loops and their transmission loop gains:

- $L_1 = -G_2G_6$,
- $L_2 = G_1G_2G_3G_4G_8$, and
- $L_3 = G_3G_4G_7$.

Considering them pairwise, only L_1 and L_3 are non-touching. The single triplet of loops is touching.

Therefore:

$$\begin{aligned}\Delta &= 1 - (L_1 + L_2 + L_3) + L_1L_3 \\ &= 1 + G_2G_6 - G_1G_2G_3G_4G_8 - G_3G_4G_7 - G_2G_3G_4G_6G_7\end{aligned}$$

Applying Mason's rule

Having gathered all the individual parts, applying Mason's rule is easy:

$$\begin{aligned}T = \frac{T_1\Delta_1 + T_2\Delta_2}{\Delta} &= \frac{G_1G_2G_3G_4 \cdot 1 + G_4G_5 \cdot (1 + G_2G_6)}{1 + G_2G_6 - G_1G_2G_3G_4G_8 - G_3G_4G_7 - G_2G_3G_4G_6G_7} \\ &= \frac{G_1G_2G_3G_4 + G_4G_5 + G_2G_4G_5G_6}{1 + G_2G_6 - G_1G_2G_3G_4G_8 - G_3G_4G_7 - G_2G_3G_4G_6G_7}\end{aligned}$$

You can check that this is the same result as (1.6).

You should make a judgment yourself, but overall my experience is that applying Mason's rule is quicker than applying the block diagram transformations. However, you need to make sure you understand the rule in detail and therefore you need excellent knowledge about the graph concepts involved. Exercise!

Solving systems of linear equations

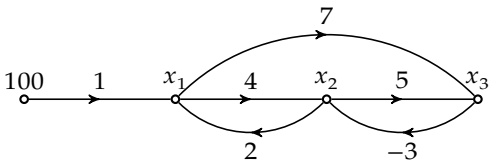
Another nice application of Mason's rule is solving systems of linear equations. We will introduce this using a simple 3×3 example:

$$\begin{cases} x_1 - 2x_2 = 100 \\ 4x_1 - x_2 - 3x_3 = 0 \\ 7x_1 + 5x_2 - x_3 = 0 \end{cases}$$

The first step, is to write this system such that you have each of the unknowns as the single term on the left hand side of the equations, e.g.

$$\begin{cases} x_1 = 100 + 2x_2 \\ x_2 = 4x_1 - 3x_3 \\ x_3 = 7x_1 + 5x_2 \end{cases}$$

Now, represent every variable using a node and introduce extra nodes for the constants in the right-hand sides of the equations. This allows drawing the following signal flow graph:



Solving the equation set for a particular variable, corresponds to calculating the path gain (or transmission) from a constant node[2] to the variable, i.e. determining:

$$G_1 = \frac{x_1}{100} \qquad G_2 = \frac{x_2}{100} \qquad G_3 = \frac{x_3}{100}$$

And then realizing that after calculation of G_1, G_2 and G_3, the solution is as simple as:

$$x_1 = 100 \cdot G_1 \qquad x_2 = 100 \cdot G_2 \qquad x_3 = 100 \cdot G_3$$

So, let's start by calculating these gains. Note that for any of the gains the graph determinant is the same (as the graph is the same). The graph contains three loops:

- $L_1 = 4 \cdot 2$
- $L_2 = 5 \cdot (-3)$
- $L_3 = 7 \cdot (-3) \cdot 2$

There is no pair of non-touching loops and therefore also no likewise triplet.

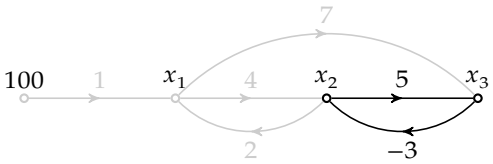
Therefore:

$$\Delta = 1 - (L_1 + L_2 + L_3) = 1 - (4 \cdot 2 + 5 \cdot (-3) + 7 \cdot (-3) \cdot 2) = 50$$

Determining G_1 to solve for x_1

As the node '100' is the source and x_1 the sink, there is only one path from source to sink, i.e. $T_1 = 1$.

The corresponding cofactor can be determined by calculating the graph's determinant after removing all forward path nodes and edges from the graph. They have been grayed below:



Therefore, there is only one remaining loop and the cofactor is $\Delta_1 = 1 - 5 \cdot (-3) = 16$.

Applying Mason's rule results in:

$$G_1 = \frac{T_1 \Delta_1}{\Delta} = \frac{1 \cdot 16}{50} = \frac{8}{25}$$

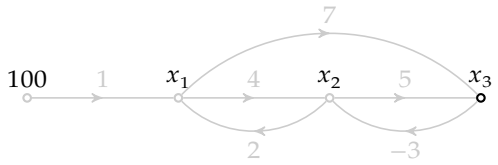
Conclusion: $x_1 = 100 \cdot G_1 = 32$

Determining G_2 to solve for x_2

As the node '100' is the source and x_2 the sink, there are two forward paths: $T_1 = 1 \cdot 4 = 4$ and $T_2 = 1 \cdot 7 \cdot (-3) = -21$.

Removing the forward path T_1 from the graph results in:

[2] If there is more than one constant (input) node, then you have to consider these input nodes in superposition.



Therefore, $\Delta_1 = 1$.

Removing the forward path T_2 from the graph voids the entire graph. Therefore, $\Delta_2 = 1$.

Applying Mason's rule results in:

$$G_2 = \frac{T_1\Delta_1 + T_2\Delta_2}{\Delta} = \frac{4 \cdot 1 - 21 \cdot 1}{50} = \frac{-17}{50}$$

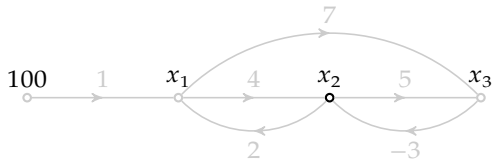
Conclusion: $x_2 = 100 \cdot G_1 = -34$

Determining G_3 to solve for x_3

As the node '100' is the source and x_3 the sink, there are two forward paths from source to sink, i.e. $T_1 = 1 \cdot 4 \cdot 5 = 20$ and $T_2 = 1 \cdot 7 = 7$.

Removing T_1 from the graph again voids the entire graph, and therefore $\Delta_1 = 1$.

Removing T_2 from the graph reduces the graph to a single node:



Therefore, $\Delta_2 = 1$.

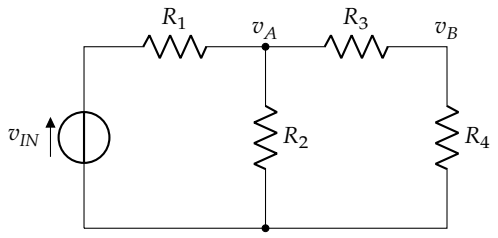
Applying Mason's rule results in:

$$G_3 = \frac{T_1\Delta_1 + T_2\Delta_2}{\Delta} = \frac{20 \cdot 1 + 7 \cdot 1}{50} = \frac{27}{50}$$

Conclusion: $x_3 = 100 \cdot G_3 = 54$

Solving linear electrical networks

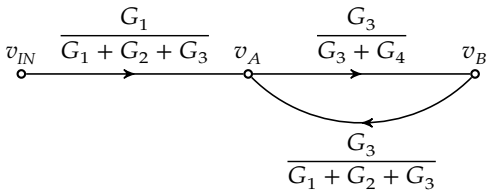
Finally, Mason can also help to analyze linear electrical networks. We will demonstrate this using the following example:



Using Millman's theorem [Mil40] (with $G = 1/R$), we can compose the following equations:

$$v_A = \frac{G_1 v_{IN} + G_3 v_B}{G_1 + G_2 + G_3}$$
$$v_B = \frac{G_3 v_A}{G_3 + G_4}$$

The corresponding graph is:



This graph has a single forward path and only a single loop that touches this forward path. Hence:

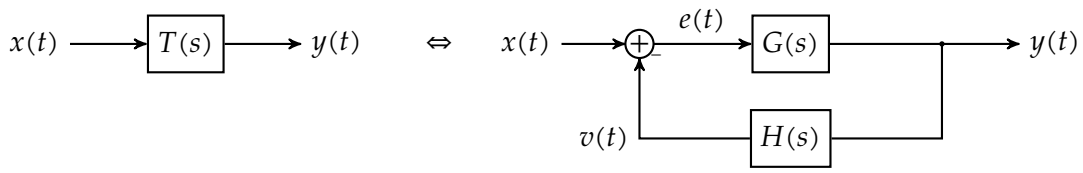
$$\frac{v_B}{v_{IN}} = \frac{\frac{G_1}{G_1+G_2+G_3}\frac{G_3}{G_3+G_4}}{1 - \frac{G_3}{G_3+G_4}\frac{G_3}{G_1+G_2+G_3}} = \frac{G_1 G_3}{G_1 G_3 + G_2 G_3 + G_1 G_4 + G_2 G_4 + G_3 G_4}$$
$$= \frac{R_2 R_4}{R_2 R_4 + R_1 R_4 + R_2 R_3 + R_1 R_3 + R_1 R_2}$$

1.3 Terminology

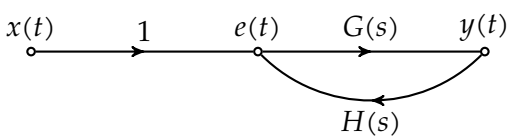
Terms like loop gain, open-loop gain, closed-loop gain and overall gain are very frequently used in conjunction with feedback systems. Though this text tries to be consistent only using the terms *overall gain* and *loop gain*, the terms open-loop and closed-loop gain are also very common but are also often confusing.

The good terminology

Consider the block diagram below of a system $T(s)$ (on the left) that is actually implemented as a feedback system (on the right).



The signal flow graph of the same system has been drawn below:



The *overall gain* of this schematic is $T(s)$:

$$T(s) = \frac{G(s)}{1 + \underbrace{G(s)H(s)}_{\equiv L(s)}}$$

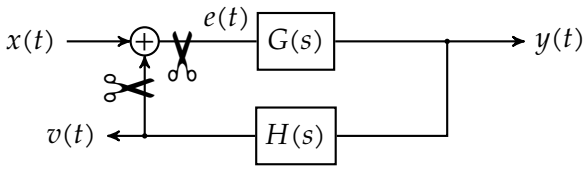
with $L(s)$ the *loop gain*. Note that this corresponds to the definition of the loop gain (or transmission) in Mason's sense: the total gain of the path that forms the loop.

The bad terminology

Very commonly the terms open-loop gain and closed-loop gain are also used. They stem from the attempt to describe the loop gain as follows:

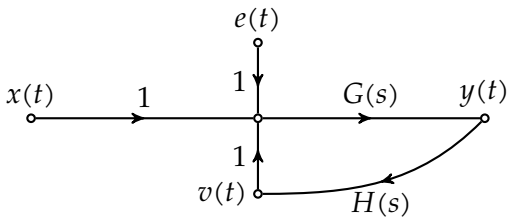
$$L(s) = \frac{V(s)}{E(s)}$$

However, this requires determining the gain from E to V. However E is not an input signal, so to make it available, one would have to open the block diagram (e.g. using the scissors as indicated below), leading to the concept of loop gain after opening the loop, the so-called *open-loop gain*.



Though this way of viewing things makes sense, the trouble arises when defining the counterpart term: the closed-loop gain.

Most people define it as the overall gain $T(s) = G(s)/(1 + G(s)H(s))$. However, then it is no longer the gain of the loop. A more consistent definition would be the true gain of the loop, augmenting the signal flow graph with an extra input for $e(t)$, and an extra output node for $v(t)$, leading to



and applying Mason's rule:

$$\left.\frac{V(s)}{E(s)}\right|_{X(s)=0} = \frac{G(s)H(s)}{1 + G(s)H(s)}$$

However, no one uses this definition and therefore let's agree not to use the terms open-loop and closed-loop gain. They are confusing.

1.4 Conclusion

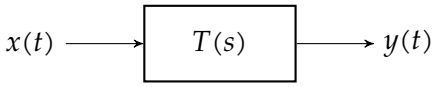
Composing and analyzing a block diagram for a linear system and analyzing its transfer functions using block diagram manipulation or Mason's rule is a basic skill in analyzing linear control systems. Make sure you master it. It will also prove to be useful in other domains dealing with linear systems, e.g. digital signal processing.

Chapter 2

Specifications

When designing a system, we should think about what we want to achieve. What are our design goals? These are usually fixed in a set of specifications. With respect to control theory, we should think about what specifications are relevant.

Consider the system below, with input $x(t)$ and output $y(t)$:



Our goal is to use linear systems in a *controlled manner*. What does that mean? It means that we provide the system with an appropriate excitation $x(t)$ such that $y(t)$ exhibits a desired response.

The desired response might be of different gradations of importance (in increasing order):

- the output should remain *stable* at all time (i.e. don't run away to infinity)
- the output should *track* the input
  - changing to a new desired 'constant' value
  - following a 'varying' input signal
- the reaction should be *swiftly*

To this end, it makes sense to define some quality metrics that specify:

1. how *stable* the output remains,
2. how *prone* the system is to parameter variations and disturbance signals, and
3. how *quick and accurate* the input is tracked.

We will focus on the former two in the subsequent chapters. In this chapter, we will focus on the latter.

We can classify these speed-related specifications into three categories:

- Time-domain specifications
  - speed

  - overshoot
- Frequency-domain specifications
  - Bandwidth
  - Gain-bandwidth
  - resonance in the frequency domain
- Steady-state specifications

Of course, our overview is incomplete. One can define many more performance metrics like ISE, IAE, ITSE and ITAE. We will not go into detail. Check the literature for more information on these [DB11].

After having read/studied this chapter, you are expected to be able to

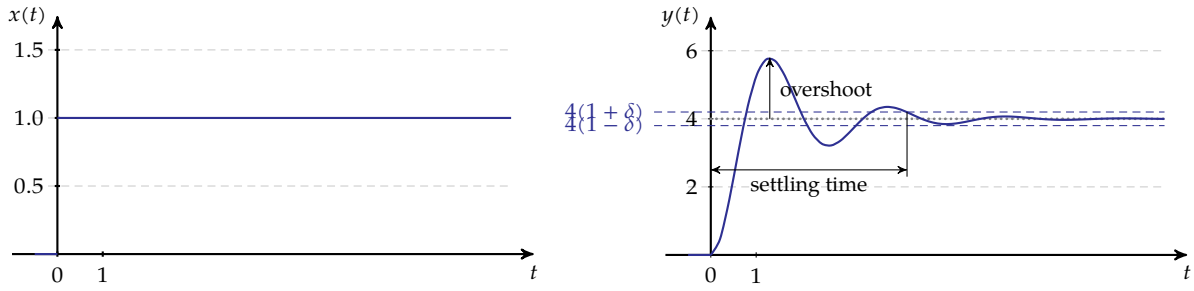
- understand and explain the relevant time-domain and frequency-domain specifications,
- calculate these specifications for basic (first-order and second-order systems), and estimate what happens for higher-order systems, and
- translate given specification in requirement for the location of poles and the zeros of the system.

2.1 Time-domain specifications

When considering time-domain specifications, we often study the system when it is excited by a step input. In this case — if the system is stable (which we will deal with in detail later) — the output should evolve to specific final value. That final value can easily be calculated using the final value theorem from Laplace theory:

$$\lim_{t\to\infty} y(t) = \lim_{s\to 0} sY(s) = \lim_{s\to 0}\left(s\cdot T(s)\cdot \underbrace{\frac{1}{s}}_{\text{step input}}\right) = \lim_{s\to 0} T(s)$$

Consider e.g. a system $T(s)$ that is fed a step input as input (below left) and that reacts as indicated below right.



The δ settling time and the overshoot have been indicated on the graph. We will go into detail in the next subsections.

2.1.1 Settling time

Settling time
The time it takes a system to reach its final value A within $A \cdot (1 \pm \delta)$ when fed a step input, is called the δ-settling time.

First-order system

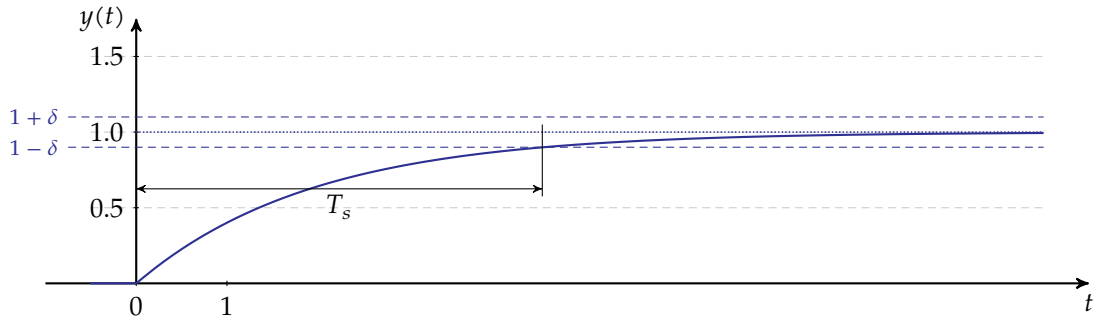
Consider the generic first-order system with transfer function:

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

The output in response to a unit step input can be calculated to be

$$y(t) = 1 - \mathrm{e}^{-\frac{t}{\tau}}$$

This response has been graphed below:



It can be seen that as of $t = T_s$ the response stays within the $1 \pm \delta$ band around the final value. This allows for calculating the δ-settling time by solving $y(t) = 1 - \delta$ for t, i.e.

$$T_{s,\delta} = \tau \ln\left(\frac{1}{\delta}\right)$$

Often δ is taken to be 2%. In that case

$$T_{s,0.02} \approx 4\tau$$

which can serve as rule of thumb.

Second-order system

Consider the generic second-order system with transfer function

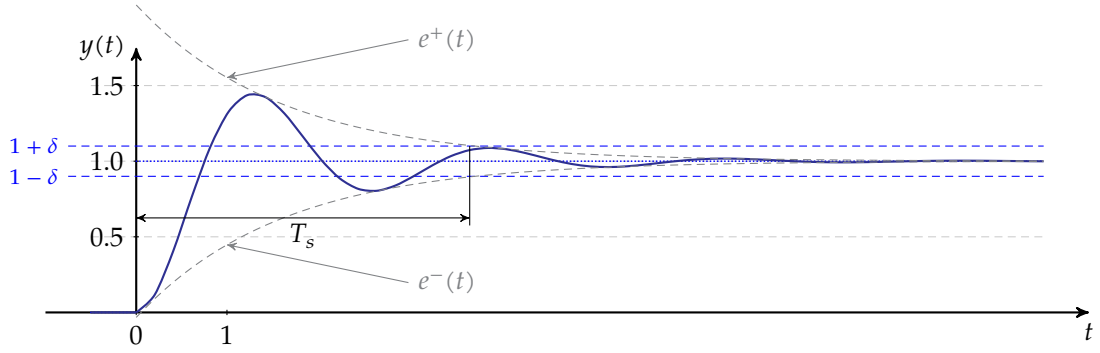
$$T_s = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

The output in response to a unit step function can be calculated to be

$$y(t) = 1 - \frac{1}{\sqrt{1-\zeta^2}} \mathrm{e}^{-\zeta\omega_n t} \sin(\omega_d t + \phi) \tag{2.1}$$

with $\omega_d = \omega_n\sqrt{1-\zeta^2}$ and $\phi = \text{atan}\,\frac{\sqrt{1-\zeta^2}}{\zeta}$.

The resulting waveform has been graphed below:



Calculating at what time instant the oscillating curve enters the target zone $1 \pm \delta$ is difficult. By simplifying the problem a bit, we can obtain a useful result more easily. Let's agree that settling has occurred as soon as the envelope of the oscillation (indicated as $e^+(t)$ and $e^-(t)$ on the graph) enter the target zone.

This allows for a more easy calculation of the settling time. The sine wave in (2.3) on itself oscillates between -1 and 1 (as any sine wave does). The upper envelope corresponds to $y(t)$ assuming the sine is fixed to its maximal value (i.e. 1). The lower envelope corresponds to $y(t)$ assuming the sine wave is fixed to its minimal value (i.e. -1).

Therefore:

$$e^+(t) = 1 + \frac{1}{\sqrt{1-\zeta^2}}\,\mathrm{e}^{-\zeta\omega_n t}$$
$$e^-(t) = 1 - \frac{1}{\sqrt{1-\zeta^2}}\,\mathrm{e}^{-\zeta\omega_n t}$$

The settling time can now be calculated by e.g. solving

$$1 - \frac{1}{\sqrt{1-\zeta^2}}\,\mathrm{e}^{-\zeta\omega_n t} = 1 - \delta$$

for t.

This results in:

$$T_{s,\delta} = \frac{1}{\zeta\omega_n}\ln\frac{1}{\delta\sqrt{1-\zeta^2}}$$

Again, often δ is taken to be 2%. Assuming $\zeta \ll 1$ and therefore $\sqrt{1-\zeta^2} \approx 1$, we can derive a new rule of thumb for second-order systems:

$$T_{s,0.02} \approx \frac{4}{\zeta\omega_n} \tag{2.2}$$

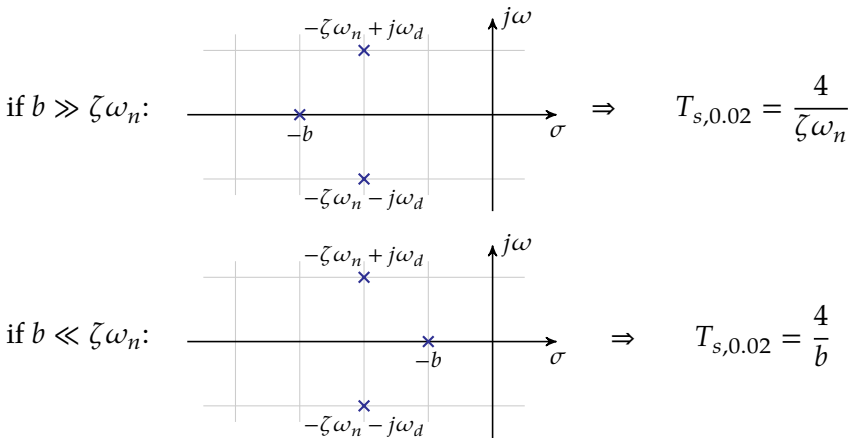
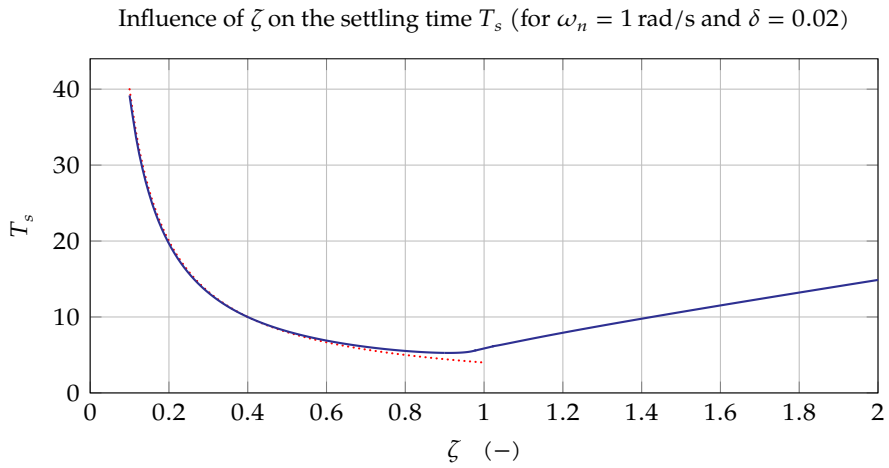


Figure 2.1: In an all-pole third-order system, the dominant pole(s) determine(s) the settling time

Of course, this is a huge simplification as the real settling time depends on ζ. Additionally, the approximation of (2.2) is quite accurate. It has been indicated on the graph below (in red dots), together with the true settling time that depends on ζ.



Note that the shortest settling time occurs for $\zeta = 1$ (i.e. critical damping).

Third-order system

With an extra pole at $s = -b$ to a system with complex conjugate poles, we can write the transfer function as:

$$T(s) = \frac{\omega_n^2 b}{(s^2 + 2\zeta\omega_n s + \omega_n^2)(s + b)}$$

In this case it is mostly a matter of checking whether the complex pole pair is dominant or not. Dominant means 'closer to the imaginary axis'. This has been illustrated in Figure 2.1.

Zeros

Though zero's can have a significant effect on overshoot (as we will see in the next section), in most cases their influence on settling time is limited. However, simulating their effect is recommended to be sure.

2.1.2 Overshoot

Overshoot
If a system surpasses its final value when it responds to a step input, we denote this as overshoot.

We can calculate the maximal amount of overshoot as an absolute value or as a relative value, w.r.t. the step made in the output.

First-order system

From the typical exponential response of a first-order system, we must conclude that overshoot does not occur.

Second-order system

Consider the generic second-order system with transfer function

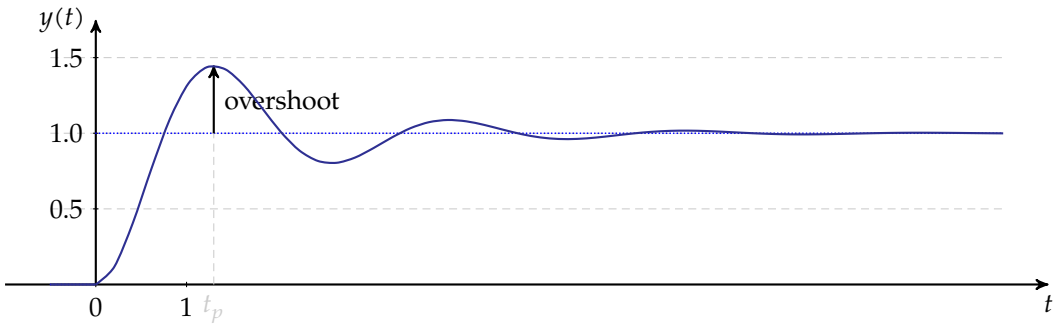
$$T_s = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

The output in response to a unit step function can be calculated to be

$$y(t) = 1 - \frac{1}{\sqrt{1-\zeta^2}} \, \mathrm{e}^{-\zeta\omega_n t} \sin(\omega_d t + \phi) \tag{2.3}$$

with $\omega_d = \omega_n \cdot \sqrt{1-\zeta^2}$ and $\phi = \operatorname{atan} \frac{\sqrt{1-\zeta^2}}{\zeta}$.

The resulting waveform has been graphed below:

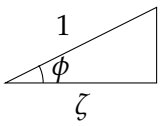


Obviously, the overshoot occurs at one of the maximum of this graph. We can determine this maximum as the smallest nonzero local extremum of the graph. These local extrema occur at

the stationary points of the function, i.e.

$$\frac{\mathrm{d}}{\mathrm{d}t}y(t) = \frac{1}{\sqrt{1-\zeta^2}}\left(\mathrm{e}^{-\zeta\omega_n t}(\zeta\omega_n)\sin\left(\omega_n\sqrt{1-\zeta^2}t+\phi\right) - \mathrm{e}^{-\zeta\omega_n t}\cos\left(\omega_n\sqrt{1-\zeta^2}t+\phi\right)\cdot(\omega_n\sqrt{1-\zeta^2})\right)$$

$$= \frac{1}{\sqrt{1-\zeta^2}}\mathrm{e}^{-\zeta\omega_n t}\omega_n\left(\zeta\sin\left(\omega_n\sqrt{1-\zeta^2}t+\phi\right) - \sqrt{1-\zeta^2}\cos\left(\omega_n\sqrt{1-\zeta^2}t+\phi\right)\right)$$

$\downarrow$ 

$\qquad \phi = \operatorname{atan}\frac{\sqrt{1-\zeta^2}}{\zeta}$

$$= \frac{1}{\sqrt{1-\zeta^2}}\mathrm{e}^{-\zeta\omega_n t}\omega_n\left(\cos\phi\sin\left(\omega_n\sqrt{1-\zeta^2}t+\phi\right) - \sin\phi\cos\left(\omega_n\sqrt{1-\zeta^2}t+\phi\right)\right)$$

$$= \frac{1}{\sqrt{1-\zeta^2}}\mathrm{e}^{-\zeta\omega_n t}\omega_n\sin\left(\omega_n\sqrt{1-\zeta^2}t+\phi-\phi\right)$$

$$= \frac{1}{\sqrt{1-\zeta^2}}\mathrm{e}^{-\zeta\omega_n t}\omega_n\sin\left(\omega_n\sqrt{1-\zeta^2}t\right)$$

The first nonzero zero crossing t_p (i.e. maximum) occurs when the sine wave has its first nonzero root:

$$\omega_n\sqrt{1-\zeta^2}t_p = \pi \qquad \Leftrightarrow \qquad t_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}}$$

This also allows for calculating the corresponding maximum:

$$\begin{aligned} y(t_p) &= 1 - \frac{1}{\sqrt{1-\zeta^2}}\mathrm{e}^{-\zeta\omega_n t_p}\sin\left(\omega_d t_p+\phi\right) \\ &= 1 - \frac{1}{\sqrt{1-\zeta^2}}\mathrm{e}^{-\frac{\zeta}{\sqrt{1-\zeta^2}}\pi}\sin(\pi+\phi) \\ &= 1 - \frac{1}{\sqrt{1-\zeta^2}}\mathrm{e}^{-\frac{\zeta}{\sqrt{1-\zeta^2}}\pi}(\underbrace{\sin\pi}_{=0}\cos\phi + \underbrace{\cos\pi}_{=-1}\sin\phi) \\ &= 1 + \frac{1}{\sqrt{1-\zeta^2}}\mathrm{e}^{-\frac{\zeta}{\sqrt{1-\zeta^2}}\pi}\underbrace{\sin\phi}_{=\sqrt{1-\zeta^2}} = 1 + \mathrm{e}^{-\frac{\zeta}{\sqrt{1-\zeta^2}}\pi} \end{aligned}$$

Note that this maximum does not depend on ω_n!

Conclusion: the relative overshoot O equals:

$$O = \mathrm{e}^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \tag{2.4}$$

To get an idea of typical overshoot values (and how they depend on ζ) you can inspect Figure 2.2.

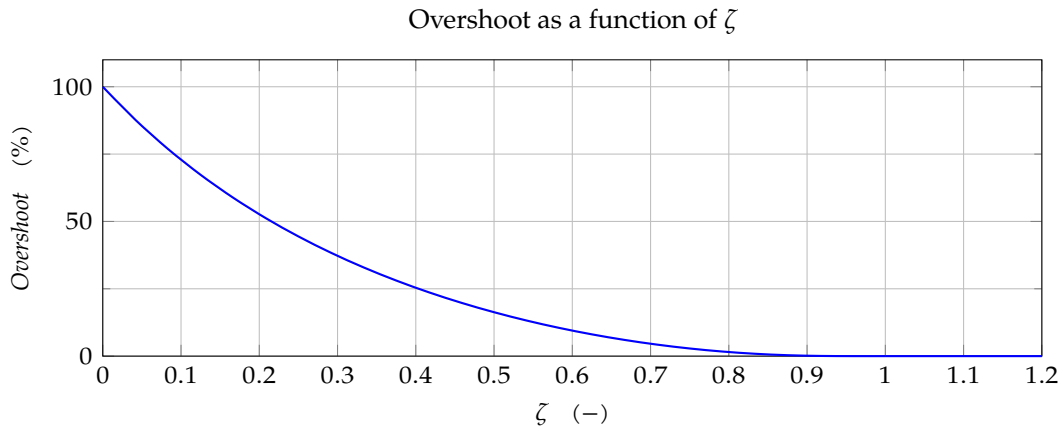


Figure 2.2: Overshoot of a second-order system without zeros

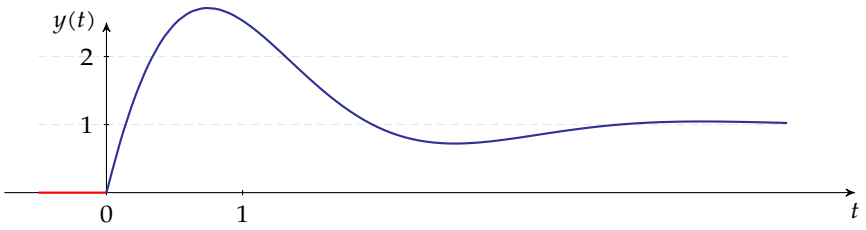


Figure 2.3: Response of a second-order system with negative zero on a step input ($\omega_n = 2$, $\zeta = 0.5$ and $\tau_Z = 2/\zeta\omega_n$)

Third-order systems

As the third (real) pole adds damping to the system, the overshoot will be smaller and smaller as the third pole moves from a non-dominant to a dominant position (i.e. closer to the imaginary axis). The overshoot may even vanish altogether.

Second-order systems with a zero

While the influence of zeros on settling time was minor, the influence of zeros on overshoot can be huge!

Consider a second-order system with a negative zero:

$$T_s = \frac{\omega_n^2(\tau_z s + 1)}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

Subjecting this system to a unit step response results in a time-domain response equaling:

$$y(t) = 1 - \mathrm{e}^{-\zeta\omega_n t}\left(\cos(\omega_d t) + \frac{\zeta - \tau_z\omega_n}{\sqrt{1-\zeta^2}}\sin(\omega_d t)\right)$$

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

An example of such a response can be found in Figure 2.3. Note how the overshoot in this case can be larger than 100%!

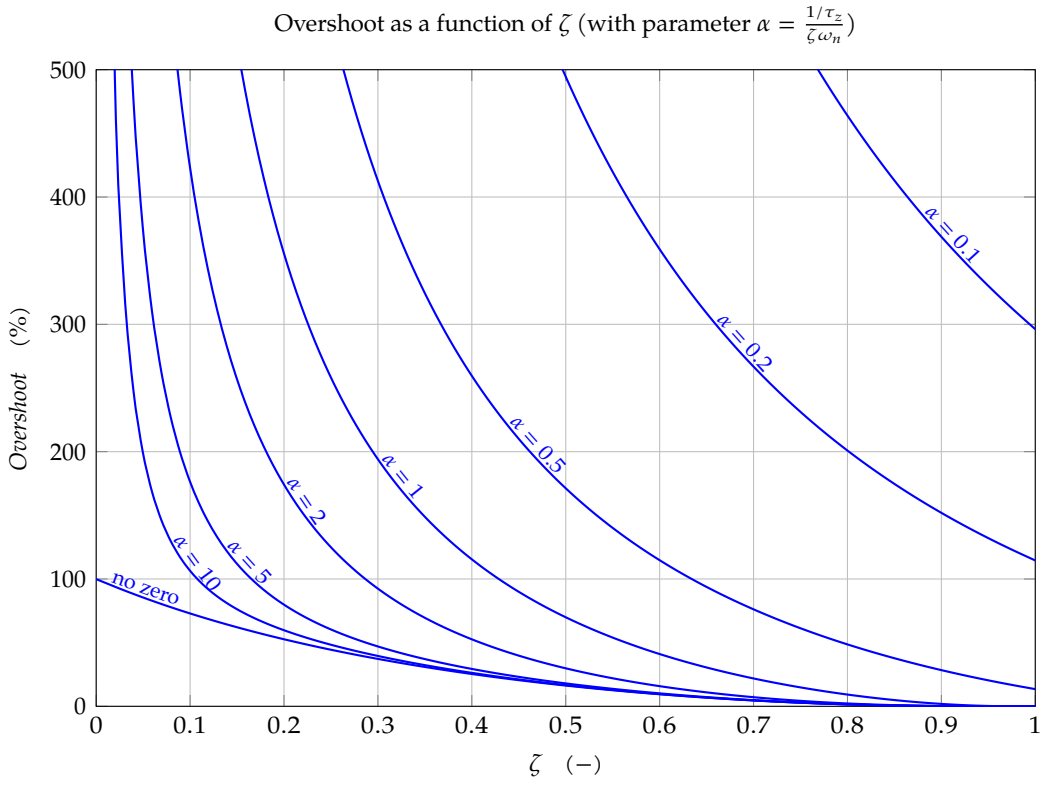


Figure 2.4: The influence of an additional zero in a second-order system on overshoot (w.r.t. ζ)

Calculating an expression for the overshoot is not so simple in this case[1]. However, the numerical approach that resulted in Figure 2.4 gives a fairly good idea of the huge impact of zeros. Another interesting view can be found in Figure 2.5.

2.2 Frequency domain specifications

2.2.1 Bandwidth

The bandwidth of a system with transfer function $T(s)$ is well defined if it has a frequency range in which the gain is almost constant and maximal.

Bandwidth The bandwidth of a system with transfer function $T(s)$ is defined as the frequency range for which:

$$|T(j\omega)| \geq \frac{\max_\omega |T(j\omega)|}{\sqrt{2}}$$

Therefore, the boundaries ω_B of this range can be obtained by solving

$$|T(j\omega_B)| = \frac{\max_\omega |T(j\omega)|}{\sqrt{2}}$$

[1]Even Belgium is not rainy enough to allow for enough consecutive rainy days to perform this analysis to the end.

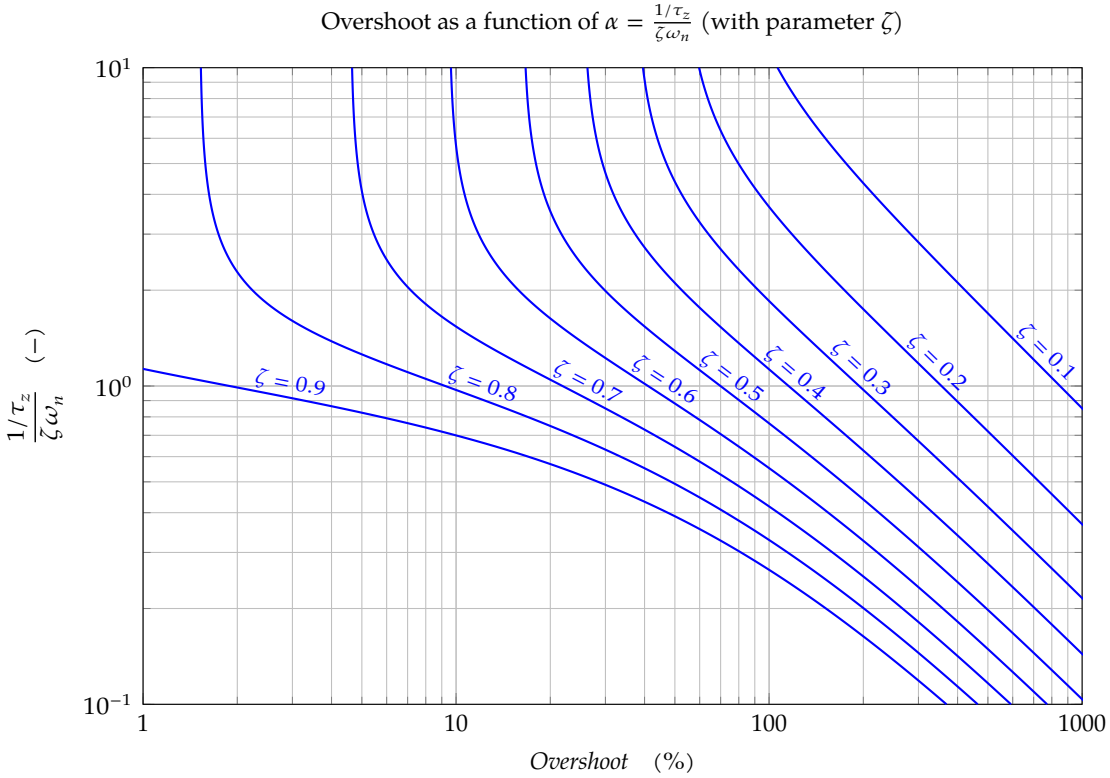


Figure 2.5: The influence of an additional zero in a second-order system on overshoot (w.r.t. α)

for ω_B.

First-order systems

Considering the generic first-order system with transfer function

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

The maximal gain equals 1 and therefore

$$\omega_B = \frac{1}{\tau}$$

Second-order systems

Considering the generic second-order system with transfer function

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

The maximal gain equals 1 and solving

$$\left| \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \right|_{s=j\omega_B} = \frac{1}{\sqrt{2}}$$

for ω_B (try this yourself!) yields:

$$\omega_B = \omega_n \sqrt{1 - 2\zeta^2 + \sqrt{(1 - 2\zeta^2)^2 + 1}}$$

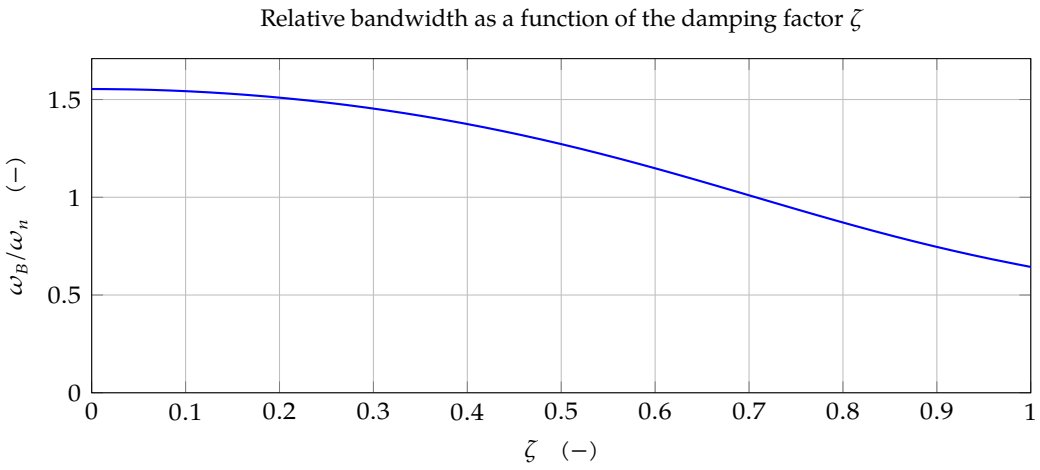


Figure 2.6: Relative bandwidth of a second-order systems as a function of damping

Taking a look at Figure 2.6, we may conclude as a rough rule of thumb that in a wide range $\omega_B \approx \omega_n$.

In general

As the settling time in the first-order case and the second-order case is proportional to $\frac{1}{\tau}$ and $\frac{1}{\zeta\omega_n}$ we may conclude that in these cases $T_s \sim \frac{1}{\omega_B}$. It turns out that in general fast settling requires a sufficient bandwidth.

2.2.2 Gain-bandwidth

A more useful concept is *gain-bandwidth*. In the next chapter, we will see that feedback loops only operate well as long as the magnitude of their loop gain $|L(s)|_{s=j\omega}$ exceeds 1. Given the fact that we are building physical systems (and that the gain will drop to zero for very high frequencies), one can investigate for which frequency range the gain is still larger than one.

The frequency ω_{GBW} for which $|L(j\omega_{GBW})| = 1$ is often called the *gain-bandwidth* of the loop. Above that frequency the loop will stop working and only the forward paths remain in action.

As such it often defines the useful frequency operation range of a control loop. You may assume that almost always $\omega_B < \omega_{GBW}$.

2.2.3 Resonance

Frequency resonance means that there is a (sharp) peak in the spectrum of gain around a specific frequency, the resonance frequency.

First-order systems

In first-order systems no resonance occurs.

Second-order systems

From studying second-order systems, we know that frequency resonance of a system with gain

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

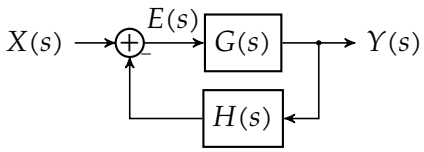
occurs for $0 < \zeta < \frac{1}{\sqrt{2}}$ at frequency ω_r and peak T_r with

$$\omega_r = \omega_n\sqrt{1-2\zeta^2} \qquad T_r = \frac{1}{2\zeta\sqrt{1-\zeta^2}}$$

2.3 Steady-state error

2.3.1 Definition

The steady-state error is a specification that only makes sense for systems with feedback. To this end, consider the feedback system below.



We know that for this system the gain from input to output amounts to:

$$T(s) = \frac{Y(s)}{X(s)} = \frac{G(s)}{1 + G(s)H(s)}$$

However, we can also calculate the output of the summation at the input, which we will call the error signal $e(t) \xrightarrow{\mathcal{L}} E(s)$. The error signal is easily related to the output as indicated below:

$$E(s) = \frac{Y(s)}{G(s)} = \frac{1}{1 + \underbrace{G(s)H(s)}_{\mathrm{L(s)}}}X(s) = \frac{1}{1+L(s)}X(s)$$

We call $L(s)$ the *loop gain*.

Steady-state error The steady-state error e_{ss} is defined as the final value of $e(t)$, i.e.

$$e_{ss} = \lim_{t\to+\infty} e(t)$$

Because of the final value theorem:

$$e_{ss} = \lim_{s\to 0} s\frac{1}{1+L(s)}X(s)$$

Note that e_{ss} depends on the chosen input signal! This can be any signal, however, two common ones are:

- a step function: $x(t) = A\,u(t) \xrightarrow{\mathcal{L}} X(s) = A/s$
- a ramp function: $x(t) = B\,t\,u(t) \xrightarrow{\mathcal{L}} X(s) = B/s^2$

2.3.2 System classification

Before we can continue analyzing the steady-state error, we need to dive into the generic nature of loop gains.

In general, they look like:

$$L(s) = K\frac{(s-z_1)(s-z_2)\cdots(s-z_M)}{s^Q(s-p_1)(s-p_2)\cdots(s-p_N)}$$

with

- M the number of zeros,
- $N+Q$ the number of poles, and
- Q the number of poles at the origin (i.e. the number of perfect integrators in the loop).

Based on this generic formulation, we can define

- the *order* of the system to be the number of poles $N+Q$, and
- the type of the system to be the number of poles at the origin Q.

Remarks

- If Q would be negative, it can represent zeros at the origin. However, this is a rare situation, as in this case the loop gain would be close to zero for low frequencies. As we will see in the next chapter, this would make the feedback system very sensitive to parameter variations and therefore long-term unstable.
- Often it makes sense to consider the DC-gain of the loop *without the integrators*. We will denote that as K'_{DC}, i.e.
$$K'_{DC} = \lim_{s\to 0} s^Q L(s) = K\frac{(-z_1)(-z_2)\cdots(-z_M)}{(-p_1)(-p_2)\cdots(-p_N)}$$
Given the fact that $L(s)$ is dimensionless (symbolically: $[L(s)]=1$) and due to the multiplication with s^Q in the definition of K'_{DC}, we must conclude that $\left[K'_{DC}\right] = (\text{rad/s})^Q$.

2.3.3 Steady-state error for a step input

When the input is a step function $x(t) = A\,(u)t$ then $X(s) = A/s$ and therefore in this case:

$$e_{ss} = \lim_{s\to 0}\left(s\cdot\frac{1}{1+L(s)}\cdot\frac{A}{s}\right) = \frac{A}{1+\lim_{s\to 0}L(s)}$$

Note: $\lim_{s\to 0}L(s)$ is the DC gain of the loop *with the integrators*.

Depending on the type Q, we can distinguish the following cases:

- $Q=0 \quad\Rightarrow\quad \lim_{s\to 0}L(s) = K'_{DC} \quad\Rightarrow\quad e_{ss} = \dfrac{A}{1+K'_{DC}}$
- $Q\geq 1 \quad\Rightarrow\quad \lim_{s\to 0}L(s) = \infty \quad\Rightarrow\quad e_{ss} = 0$

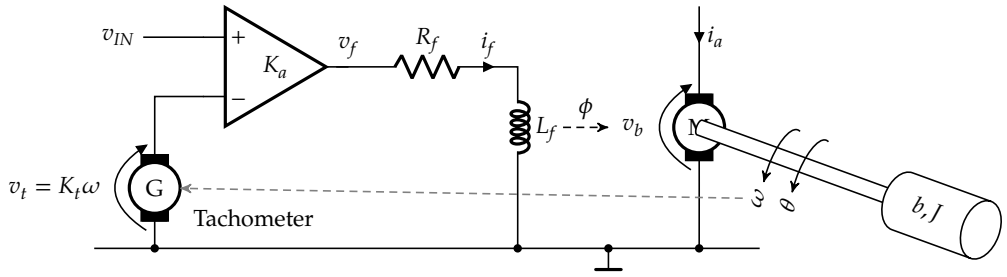


Figure 2.7: A DC motor in field control mode whose rotational speed is measured using a tachometer. The output voltage of the tachometer is compared with a set-point v_{IN} using a differential amplifier that controls the field current i_f.

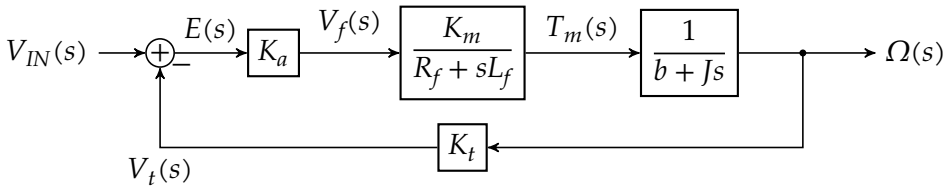


Figure 2.8: Block diagram of the motor setup of Figure 2.7; K_a is the (finite) amplifier gain and K_t the gain of the tachometer (units: V/(rad/s)).

2.3.4 Steady-state error for a ramp input

When the input is a ramp function $x(t) = B\,t\,u(t)$, then $X(s) = B/s^2$ and therefore in this case:

$$e_{ss} = \lim_{s\to 0}\left(s \cdot \frac{1}{1+L(s)} \cdot \frac{B}{s^2}\right) = \lim_{s\to 0} \frac{B}{s(1+L(s))} = \frac{B}{\lim\limits_{s\to 0} sL(s)}$$

Depending on the type Q, we can distinguish the following cases:

- $Q = 0 \quad \Rightarrow \quad \lim\limits_{s\to 0} L(s) = K'_{DC} \quad \Rightarrow \quad e_{ss} = \infty$
- $Q = 1 \quad \Rightarrow \quad \lim\limits_{s\to 0} L(s) = \dfrac{K'_{DC}}{s} \quad \Rightarrow \quad e_{ss} = \dfrac{B}{K'_{DC}}$
- $Q \geq 2 \quad \Rightarrow \quad \lim\limits_{s\to 0} L(s) = \dfrac{K'_{DC}}{s^Q} \quad \Rightarrow \quad e_{ss} = 0$

2.3.5 Application: DC motor with feedback

Consider the motor setup of Figure 2.7. A DC motor in field-control mode is equipped with a tachometer (i.e. a device that produces an output voltage that is proportional with the rotational speed), such that we can use an differential amplifier with gain K_a to regulate the field current i_f. The block diagram that corresponds to this setup can be found in Figure 2.8.

Steady-state error for a step input

The loop gain of the system is

$$L(s) = \frac{K_a K_m K_t}{(Js+b)(L_f s + R_f)} = \frac{\frac{K_a K_m K_t}{bR_f}}{(\tau_L s + 1)(\tau_f s + 1)} \qquad \text{with } \tau_L = \frac{J}{b} \text{ and } \tau_f = \frac{L_f}{R_f}$$

This is clearly a *type zero system* (i.e. no integrators in the loop).

We can easily calculate the *step steady-state error* as:

$$e_{ss} = \lim_{s\to 0}\left(s \cdot \frac{1}{1+L(s)} \cdot \frac{A}{s}\right) = \frac{A}{1+\underbrace{L(0)}_{=K'_{DC}}} \qquad \text{with } K'_{DC} = \frac{K_a K_m K_t}{R_f b}$$

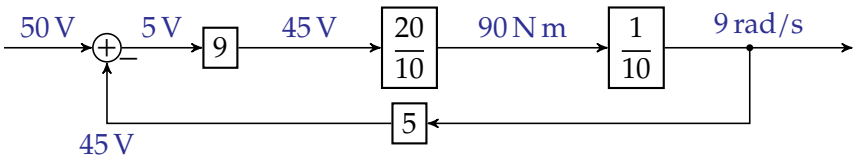
Assuming the following values for the constants in this application

Parameter	Value	Parameter	Value
K_a	9	K_m	20 Nm/A
K_t	5 V/(rad/s)		
R_f	10 Ω	b	10 Nm/(rad/s)

we can calculate $K'_{DC} = 9$ and therefore:

$$e_{ss} = \frac{A}{1+9} = A/10$$

This means that feeding the system with a step input of 50 V will result in an $e_{ss} = 5$ V. The steady-state values of the other variables in the block diagram can easily be determined to be:



Note that we omitted the units from the individual gain factors in order not to overload the diagram.

The fact that the steady-state error is nonzero can be solved in two ways:

- by increasing the DC loop gain, or
- by adding an integrator to the loop, rendering it a type 1 system.

The first option will have an important side effect, as increasing the loop gain will in the end render the loop unstable. We will discuss this in detail in the subsequent chapters.

Let's consider the second option. This in fact corresponds to integrating the rotational speed ω obtaining the angle position of the motor, labeled θ.

This has been indicated in the block diagram of Figure 2.9.

The loop gain of this system, can be determined to be

$$L(s) = \frac{K_a K_m K_\theta}{s(Js+b)(L_f s + R_f)} = \frac{\frac{K_a K_m K_t}{b R_f}}{s(\tau_L s + 1)(\tau_f s + 1)} \qquad \text{with } \tau_L = \frac{J}{b} \text{ and } \tau_f = \frac{L_f}{R_f}$$

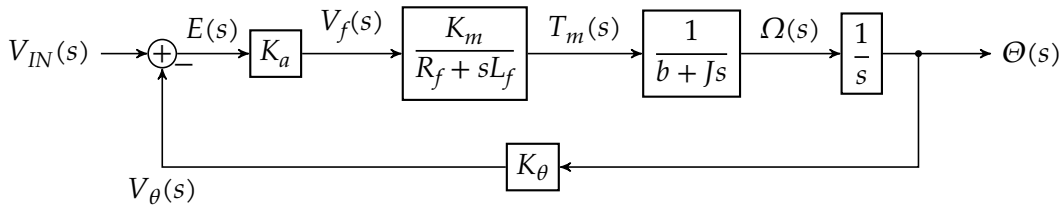


Figure 2.9: Block diagram of the motor setup of Figure 2.7 but this time with a sensor that measures the angular position of the motor shaft); K_a is the (finite) amplifier gain and K_θ the gain of the angular position sensor (units: V/rad).

This is clearly a *type one system* (i.e. one integrators in the loop).

We can easily calculate the *step steady-state error* as:

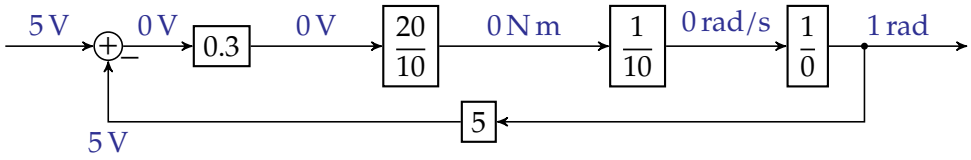
$$e_{ss} = \lim_{s\to 0}\left(s \cdot \frac{1}{1+L(s)} \cdot \frac{A}{s}\right) = \frac{A}{1+\lim\limits_{s\to 0} L(s)} = \frac{A}{1+\lim\limits_{s\to 0}\frac{K'_{DC}}{s}} \qquad \text{with } K'_{DC} = \frac{K_a K_m K_\theta}{R_f b}$$

Assuming K'_{DC} is nonzero this results in $e_{ss} = 0$.

Let's consider feeding a step input of 5 V into the system and calculate the steady-state values, using the following values for the constants in the system:

Parameter	Value	Parameter	Value
K_a	0.3	K_m	20 Nm/A
K_θ	5 V/(rad/s)		
R_f	10 Ω	b	10 Nm/(rad/s)

The result is:



Note that this situation corresponds to the machine standing still with the shaft in a position of 1 rad w.r.t. its reference position.

Steady-state error for a ramp input

It doesn't make sense feeding a ramp input into the type zero system, as the error will grow to infinity for a ramping input.

Instead, let's consider feeding a ramp input $x(t) = B\,t\,u(t)$ to the type one system (corresponding to the setup with the angular position measurement).

If we do so, the steady-state error can be calculated to be:

$$e_{ss} = \lim_{s\to 0}\left(s \cdot \frac{1}{1+L(s)}\frac{B}{s^2}\right) = \frac{B}{\lim\limits_{s\to 0} s(1+L(s))} = \frac{B}{\lim\limits_{s\to 0} s\frac{K'_{DC}}{s}} = \frac{B}{K'_{DC}} \qquad \text{with } K'_{DC} = \frac{K_a K_m K_\theta}{R_f b}$$

When filling out the values for the constants and assume $B = 9\,\mathrm{V/s}$, we obtain:

$$e_{ss} = \frac{B}{K'_{DC}} = \frac{9\,\mathrm{V/s}}{0.3\,\mathrm{rad/s}} = 30\,\mathrm{V}$$

Overview To illustrate the effects of a step input and a ramp input on type zero and type one systems, we have simulated the transients corresponding to the previous examples and plotted the results in Figure 2.10. We simulated long enough to be able to assess the steady-state situation. In the left column one can find the output signals, in the right column the error signals.

2.4 Conclusion

In this chapter, we introduced some important specifications for (feedback) control systems. This will allow us to set clear goals when designing them.

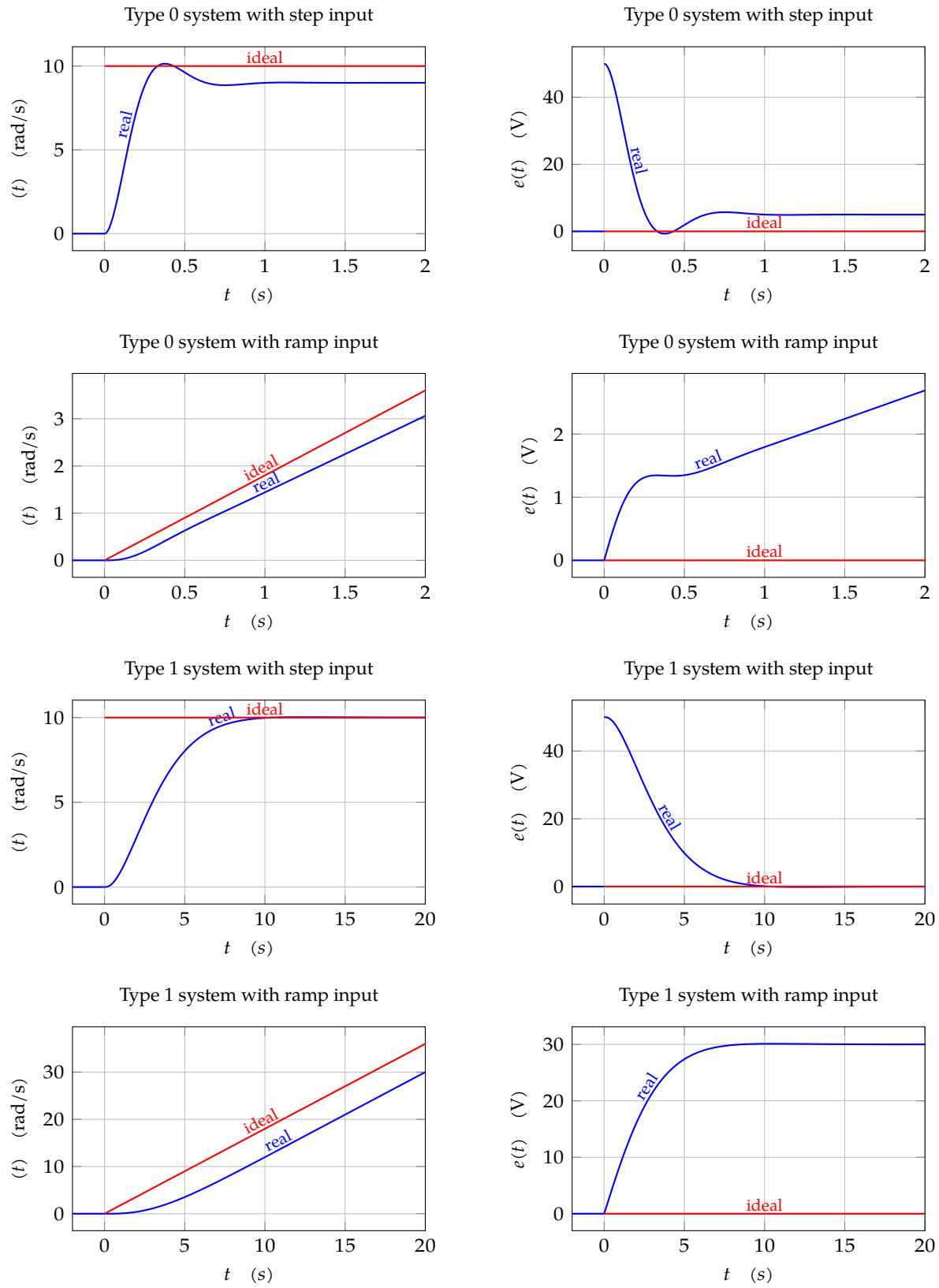


Figure 2.10: In blue, the transient outputs (left column) and error signals (right column) of the DC-motor in Type 0 (top half) and Type 1 (bottom half) configuration, subjected to step inputs (row 1 and 3) and ramp inputs (row 2 and 4). In red, the ideal output curves. They are obtained by assuming that the error signal is zero, i.e. the corresponding output must be equal to the input divided by the feedback gain. They allow to assess the deviation from the ideal situation (if one could make the loop gain infinite).

Chapter 3

The purpose of feedback

Remember, our goal is to use linear systems in a *controlled manner*. That means that we provide the system with an appropriate excitation $x(t)$ such that $y(t)$ exhibits a desired response.

Take for example a system in which we want to control the water tank of a toilet. If the toilet is flushed by opening the flushing valve, the tank needs to be refilled. We could do that by opening the inlet valve for a while and closing it afterwards. It is clear that this manual mode works, but it is not really convenient.

A better way is to measure the level of the water in the tank, and to open the inlet valve automatically. This process of measuring and actuating based on the measurement is called *feedback*.

In this chapter, we will study the advantages of feedback:

1. The system is less sensitive to parameter variations
2. The system is less susceptible to (external) disturbance signals
3. The system can be made faster

After having read and studied the chapter, you are expected to be able to:

- explain the advantages of feedback in detail
- how stability limits the achievable loop gain
- calculate gains of systems without and with feedback
- calculate the effects of parameter variations and disturbances for systems without and with feedback
- calculate the reaction speed of a system without and with feedback
- calculate the effect of (internal) signal saturation

3.1 Sensitivity to parameter variations

3.1.1 Principle

Consider below the system without feedback (on the left) and let's compare it to a system with feedback (on the right) in view of its sensitivity to parameter variations.

Without feedback We know that for the case without feedback

$$Y(s) = F(s)X(s)$$

Our goal is to find out how much $Y(s)$ changes when $F(s)$ changes, i.e. we want to calculate the derivative of $Y(s)$ w.r.t. $F(s)$.

$$\begin{aligned}
\frac{\mathrm{d}Y(s)}{\mathrm{d}F(s)} &= \frac{\mathrm{d}}{\mathrm{d}F(s)}\left(F(s)X(s)\right) = X(s) \\
&\quad \Big\downarrow X(s) = \frac{Y(s)}{F(s)} \\
&= \frac{Y(s)}{F(s)} \\
\frac{\mathrm{d}Y(s)}{Y(s)} &= \frac{\mathrm{d}F(s)}{F(s)}
\end{aligned}$$

This means that a relative change in $F(s)$ will translate itself into an equally large relative change in $Y(s)$.

With feedback We know that for the case with feedback

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

Again, deriving $Y(s)$ w.r.t. $G(s)$ yields:

$$\begin{aligned}
\frac{\mathrm{d}Y(s)}{\mathrm{d}G(s)} &= \frac{1}{(1 + G(s)H(s))^2}X(s) \\
&\quad \Big\downarrow \frac{1}{1 + G(s)H(s)}X(s) = \frac{Y(s)}{G(s)} \\
&= \frac{1}{1 + G(s)H(s)}\frac{Y(s)}{G(s)} \\
\frac{\mathrm{d}Y(s)}{Y(s)} &= \frac{1}{1 + G(s)H(s)}\frac{\mathrm{d}G(s)}{G(s)}
\end{aligned}$$

This means that a relative change in $G(s)$ will be attenuated by $1 + G(s)H(s)$ into a relative change in $Y(s)$.

3.1.2 Interpretation

However, changes (even relative ones) in the Laplace domain aren't easy to interpret, so what does the comparison of the two situations above actually mean?

Parameter	Value	Parameter	Value
K_a	9	K_m	20 Nm/A
K_t	5 V/(rad/s)		
L_f	1 H	R_f	10 Ω
J	100 Nm/(rad/s^2)	b	10 Nm/(rad/s)

Table 3.1: Numerical values used in the DC-motor example

Well an important insight in this matter is that the DC-value (for $s = 0$) of $T(s)$ tells us something about the time domain. The clue lies in the final value theorem. If we subject our system to a unit step, then we know that $Y(s) = T(s)X(s)$ with $X(s) = 1/s$. Therefore:

$$\lim_{t\to+\infty} y(t) = \lim_{s\to 0} s\,T(s) \underbrace{\frac{1}{s}}_{\text{step input}} = \lim_{s\to 0} T(s)$$

This means that the low-frequency response ($\lim_{s\to 0} T(s)$, the so called 'DC-Gain') of the system determines the final steady-state behavior.

This also holds for our sensitivity equations and therefore relative DC parameter changes will appear unaltered in the output signal in the case of a system without feedback, and will be attenuated by a factor of $1 + G(0)(0)$ in the case of a system with feedback.

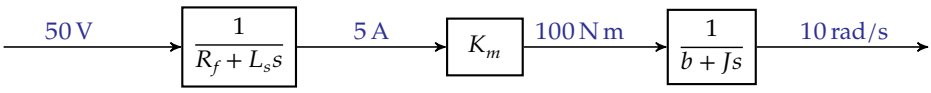
An additional insight lies in the fact that the sensitivity equations also hold when s has been replaced by $j\omega$, yielding similar equations in the Fourier domain. And remember: the Fourier values are the eigenvalues for the sine-wave eigenvectors of our linear system $T(s)$. And therefore, relative changes to $G(j\omega)$ will appear unaltered in the magnitude and phase of the sine wave with frequency ω at the output in the case of a system without feedback, and will appear attenuated in magnitude by $|1 + G(j\omega)H(j\omega)|$ at the output in the case of a system with feedback.

3.1.3 Example

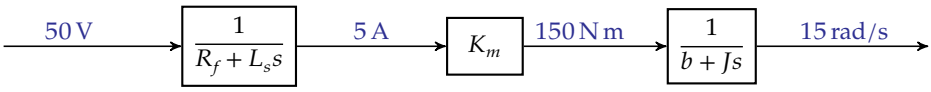
Let's illustrate this with our DC-motor case. We will use the numerical values of Table 3.1.

Without feedback

Let's consider the step steady-state case, i.e. we submit a step of 50 V to the system and we set $s = 0$. The resulting steady-state values have been indicated on the block diagram below:



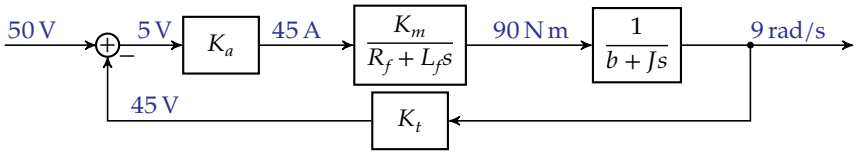
Now let's see what happens, if the value of K_m changes by 50% to 30 Nm/A:



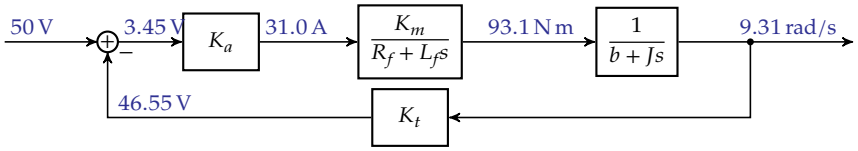
The result is as expected: the output also increases by 50%. In order to keep the output unaltered w.r.t. the situation with the original parameter value, the input should be reduced to 33.3 V. This requires a manual intervention in the system.

With feedback

Let's compare the situation above with the situation with feedback. Again, let's consider the same step-steady case (with a 50 V step at the input). The resulting values have been indicated on the block diagram below:



Now, let's again increase K_m by 50% to 30 Nm/A:



Note how, this time, the output only increases with 3.44%. In fact, what happened is that the feedback loop reduced the effective input to the system (after the summation near the input) itself, based on the increase in output speed.

This illustrates the positive effect of feedback on parameter sensitivity.

3.1.4 Other perspectives

Relationship with steady-state error The factor $1/(1 + G(s)H(s))$ also appeared in the calculation of the step steady-state error. Indeed, for an input step with amplitude A:

$$e_{ss} = \lim_{s \to 0} \frac{A}{1 + G(s)H(s)}$$

The concepts of step steady-state error and parameter desensitization are therefore very related. In general if there are no integrators in the loop, one may expect an (attenuated but real) effect on the output. If there are integrators in the feedback loop, than the step-steady state error will be zero and so will be the effect of parameter changes on the output.

Sensitivity to feedback parameter changes Note that when the parameters of the feedback path change, the effect is not the same. Indeed, calculating the influence of a change in $H(s)$ on

$Y(s)$ reveals:

$$
\begin{aligned}
Y(s) &= \frac{G(s)}{1+G(s)H(s)}X(s) \\
\frac{\mathrm{d}Y(s)}{\mathrm{d}H(s)} &= \frac{-G^2(s)}{(1+G(s)H(s))^2}X(s) \\
&\downarrow X(s) = Y(s)/(G(s)/(1+G(s)H(s))) \\
&= -\frac{G(s)}{1+G(s)H(s)}Y(s) \\
&\downarrow \text{Multiply the right hand side by } H(s)/H(s) \\
&= \frac{G(s)H(s)}{1+G(s)H(s)}\frac{Y(s)}{H(s)} \\
\frac{\mathrm{d}Y(s)}{Y(s)} &= -\frac{G(s)H(s)}{1+G(s)H(s)}\frac{\mathrm{d}H(s)}{H(s)}
\end{aligned}
$$

Given the fact that

$$0 \leq \left|\frac{G(S)H(s)}{1+G(s)H(s)}\right| \leq 1$$

and probably very close to 1 (as we prefer high loop gains to suppress forward parameter influence), this means that changes in feedback parameters appear 100% at the output. Therefore, controlling feedback parameters is of the utmost importance!

The loop gain perspective Another way to understand the low sensitivity to forward parameter changes and the high sensitivity of feedback parameter changes is to reconsider the transfer function of a linear feedback system.

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

Albeit an equation in the complex domain, if the magnitude of $G(s)H(s)$ is large, i.e. if the *loop gain* is high, we can simplify this to:

$$Y(s) = \frac{\cancel{G(s)}}{\cancel{1}+\cancel{G(s)}H(s)}X(s) \approx \frac{1}{H(s)}X(s)$$

This clearly shows that the system is fully dependent on the feedback network ($H(s)$, i.e. the sensor and its amplifier) and not on the forward gain $G(s)$.

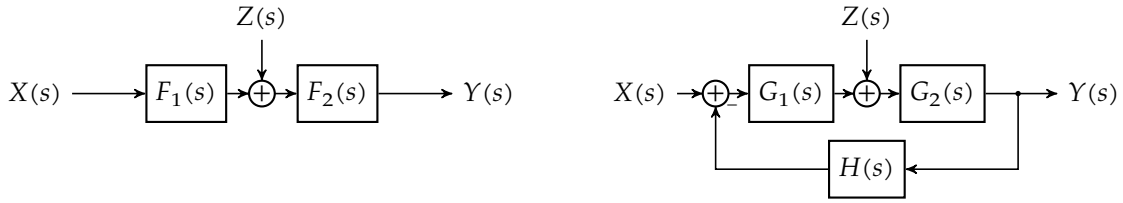
3.1.5 Conclusion

As long as the loop gain of a feedback loop is high enough, the loop's operation will give the system a low sensitivity to parameter changes in the forward path, but not in the feedback path. The steady-state error is a good metric to specify the robustness of the system w.r.t. such a forward parameter changes.

3.2 Susceptibility to disturbance signals

3.2.1 Principle

Consider a system in which there is an external disturbance signal spoiling the party. Below left you can see the situation without feedback and on the right the situation with feedback.



System without feedback We know that the following holds for the system without feedback:

$$Y(s) = F_2(s)\,(F_1(s)X(s) + Z(s))$$

This means that if $|F_1(s)|$ is large, the influence of $Z(s)$ will be relatively attenuated by that factor with respect to $X(s)$. Therefore, we want as much gain as we can get before the point at which the disturbance signal is injected into the system.

Note that the gain $T(s) = Y(s)/X(s)$ equals $F_1(s)F_2(s)$.

System with feedback By applying Mason's rule, we know that the following holds for the system with feedback:

$$Y(s) = \frac{G_2(s)\,(G_1(s)X(s) + Z(s))}{1 + G_1(s)G_2(s)H(s)}$$

Assuming for starters that $F_1(s) = G_1(s)$ and $F_2(s) = G_2(s)$, the influence of $Z(s)$ relative to $X(s)$ is similar to that of the system without feedback, as the equations are most similar and $X(s)$ and $Z(s)$ appear in the same locations in both equations. So at first sight, feedback is no added value in this case.

3.2.2 Interpretation

However, the secret lies in the fact that the feedback allows us to increase $G_1(s)$ compared to $F_1(s)$ in the case of the system without feedback.

Indeed, if we ensure enough loop gain, then the total gain $T(s) = Y(s)/X(s)$ can be approximated by:

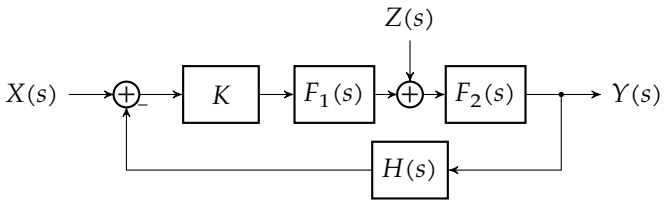
$$T(s) \approx \frac{1}{H(s)}$$

So if we ensure $1/H(s) = F_1(s)F_2(s)$, then the system with feedback has the same overall gain as the system without feedback.

Therefore, we now have the liberty to modify $G_1(s)$ and $G_2(s)$ to our desire. Because of the physics of the system, most likely $G_2(s) = F_2(s)$. However, what we can easily do is add an

amplifier with value K in front of $F_1(s)$, i.e. choose $G_1(s) = K \cdot F_1(s)$, such that it becomes large in magnitude. This corresponds to our earlier statement that we want as much gain as we can get before the point at which the disturbance signal is injected into the system.

This means that we propose our feedback system to look like:



This yields the following equation:

$$Y(s) = \frac{F_2(s)\left(KF_1(s)X(s) + Z(s)\right)}{1 + KF_1(s)F_2(s)H(s)}$$

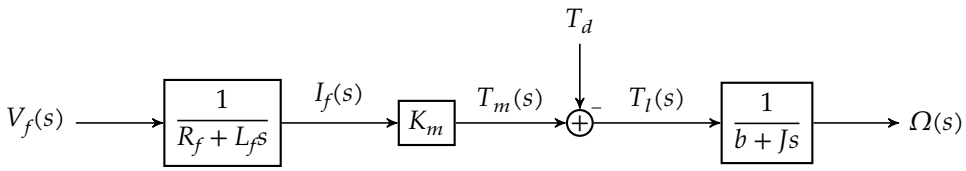
which reduces the influence of $Z(s)$ with another factor K when compared to the system without feedback.

3.2.3 Example

Let's illustrate the effect using our test case, the DC motor in field control mode. We will use the numerical values of Table 3.1.

Without feedback

Assume that our motor setup is suffering from an interfering 'source' disturbing the loading of the motor (i.e. an extra load, we did not anticipate):



We can easily analyze the angular speed $\Omega(s)$ as a function of $V_f(s)$ and $T_d(s)$:

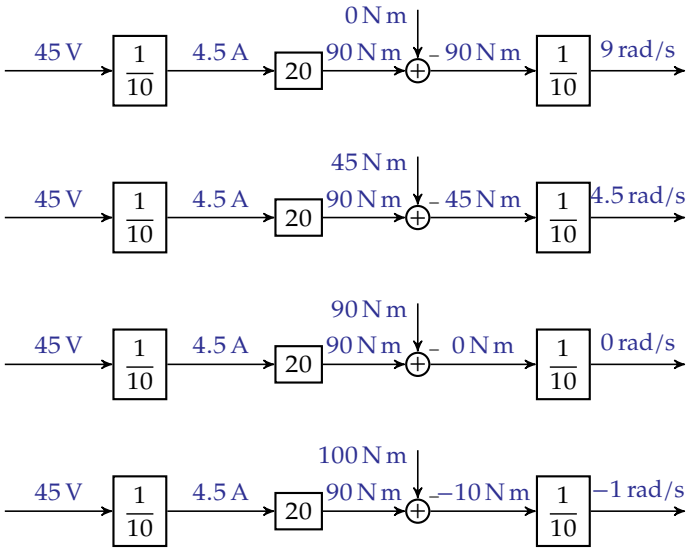
$$\Omega(s) = \frac{K_m}{(R_F + L_f s)(b + Js)} V_F(s) - \frac{1}{b + Js} T_d(s)$$

This allows assessing the step steady-state behavior (if we apply step inputs $v_f = V_F \cdot u(t)$ and $T_d(t) = T_d \cdot u(t)$):

$$\lim_{t \to +\infty} \omega(t) = \frac{K_m}{R_F b} V_F - \frac{1}{b} T_d$$

We therefore see that the disturbance signal is attenuated w.r.t. the input signal by the DC-gain of the part before the injection point.

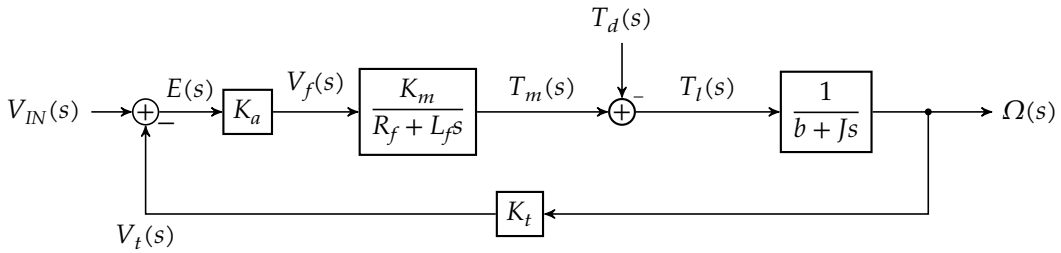
Let's illustrate this using some numerical values. We will increase the load torque T_d in steps from 0 tot 100 N m.



Note how the motor reversed its rotation direction when it is loaded with 100 N m.

With feedback

Now, take a look at the same system with feedback:



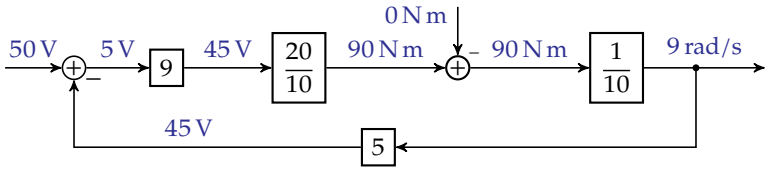
Straightforward analysis yields:

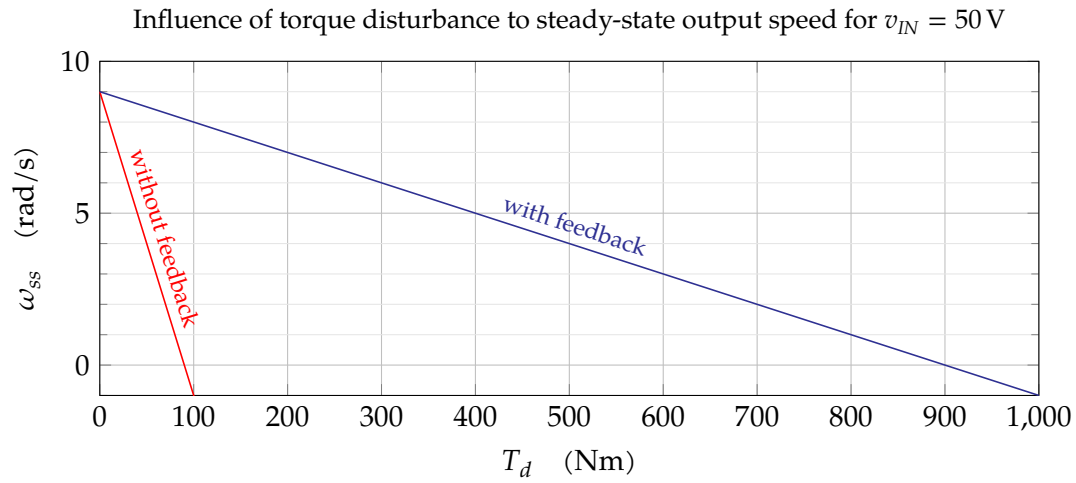
$$\Omega(s) = \frac{\frac{1}{b+Js}\left(\frac{K_a K_m}{R_f+sL_f} v_{IN}(s) - T_d(s)\right)}{1 + \frac{K_a K_m K_t}{(R_f+sL_f)(b+Js)}}$$

This again allows assessing the step steady-state behavior (if we apply step inputs $v_f = V_F \cdot u(t)$ and $T_d(t) = T_d \cdot u(t)$):

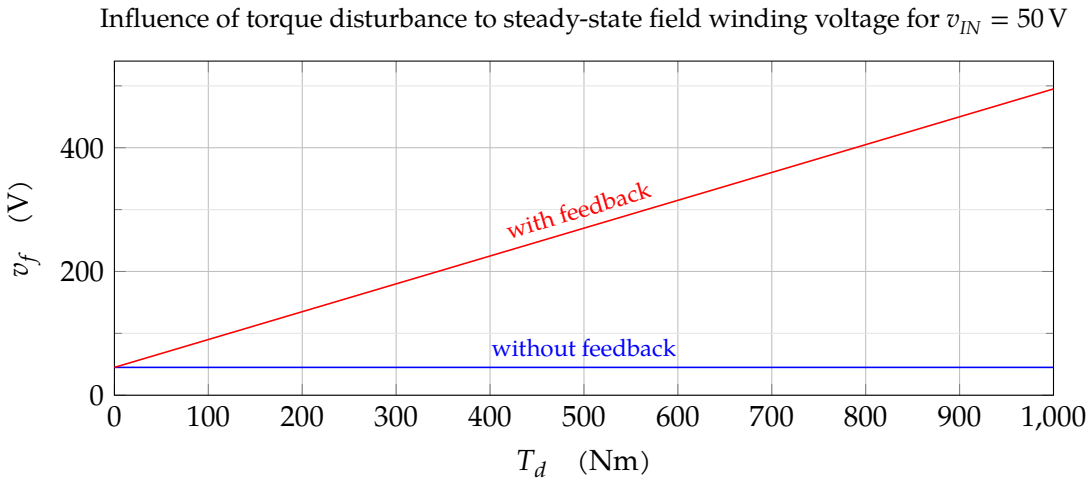
$$\lim_{t\to+\infty} \omega(t) = \frac{\frac{1}{b}\left(\frac{K_a K_m}{R_f} v_{IN} - T_d\right)}{1 + \frac{K_a K_m K_t}{b R_f}}$$

Plugging in the numerical values, allows us to assess the situation when we increase the load torque T_d in steps from 0 to 100 N m.



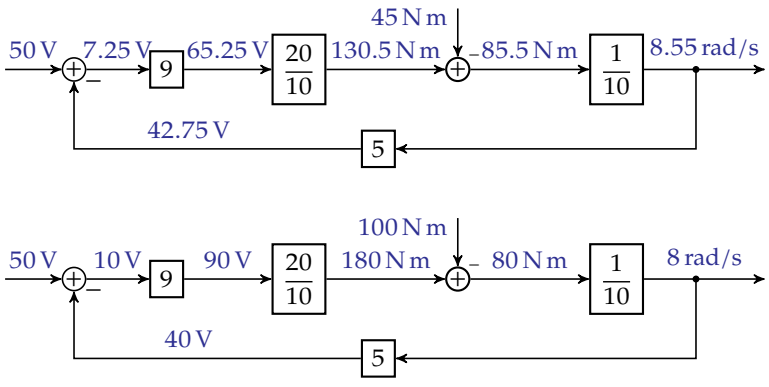


(a) Influence on the steady-state output speed.



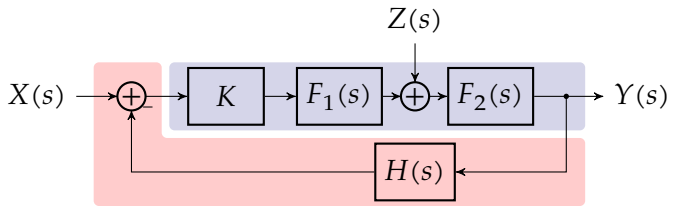
(b) Influence on the steady-state voltage over the field winding

Figure 3.1: DC motor in field-control mode without feedback (red) and with feedback (blue): influence of the load torque.

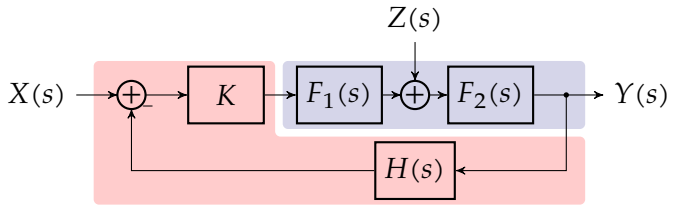


The effect is stunning. Without feedback, the motor reversed when loaded with 100 N m. Now it only decelerates to a speed of 8 rad/s (i.e. only by approximately 11%).

A comparison of the two situations can be found in Figure 3.1.



(a) Part of the topology that is parameter sensitive (red) vs. parameter insensitive (blue)



(b) Part of the topology that is disturbance intolerant (red) vs. disturbance tolerant (blue)

Figure 3.2: Division of the typical control system topology into parameter sensitive/insensitive and disturbance intolerant/tolerant parts

Also note, that the motor gets quite a beating. The field winding in the case without feedback only suffered a mere 45 V, while in the case with feedback that voltage is doubled under full load. We need to make sure that our motor is capable of dealing with these high(er) voltages.

3.2.4 Remarks

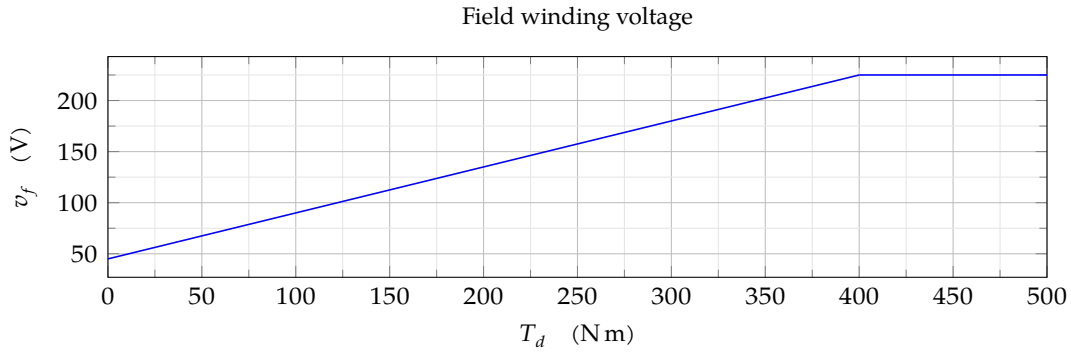
So far, feedback only seems to have advantages. However, it is not all heaven. We need to take into account a number of pitfalls.

Sensitive parts As we have seen so far, feedback reduces the sensitivity of our control system and also reduces the influence of disturbance signals. However the former only held for the forward path and the latter only holds for disturbance signals that have sufficient forward-path gain before the injection point. A situation that is even worse (and that we did not discuss) occurs if the injection happens in the feedback path. Try to analyze that yourself.

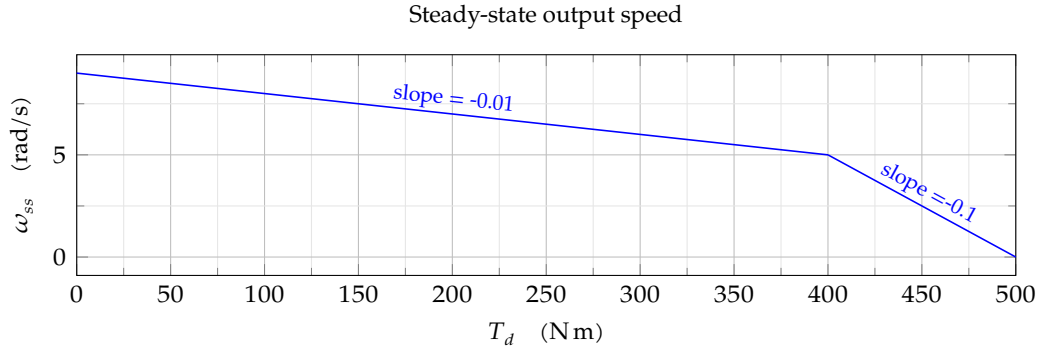
These observations leads to the fact that we can divide our control system into different parts, parameter sensitive vs. parameter insensitive parts and disturbance intolerant vs tolerant parts. These have been indicated in Figure 3.2.

Saturation So far, we assumed no range limitation on the signals that appear in the control system. However, these signals are real voltages/currents/pressures/temperatures/... and therefore they have a limited range. The range limitation is there because of physical constraints (e.g. the limited power supply voltage of the amplifiers used, or input signals of components that should stay within safe limits in order not do damage the component). The (obligatory) clipping of these signals is often referred to as 'saturation'.

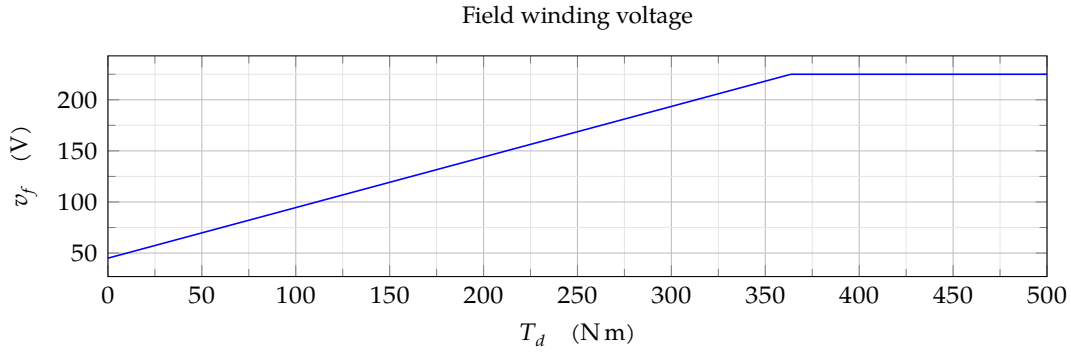
Of course, this also puts a limit on the disturbance signal tolerance. This has been illustrated in Figure 3.3 for the DC-motor case when the input voltage to the field winding of the motor should be limited to 225 V .



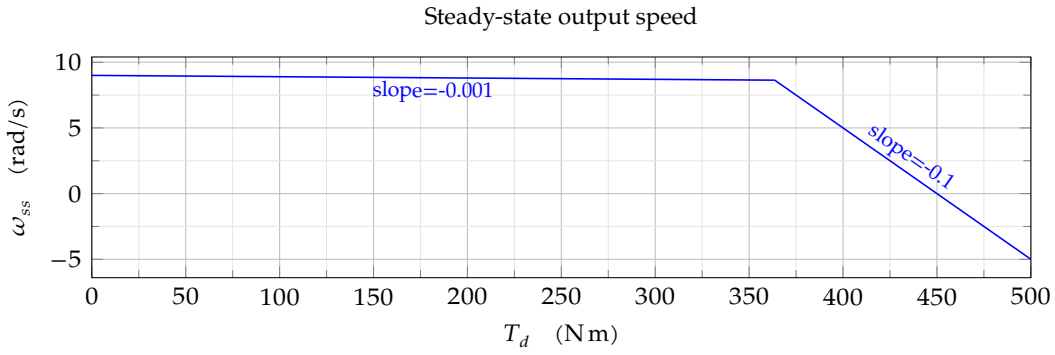
(a) Effect on the field winding voltage for case (1)



(b) Effect on the output speed for case (1)



(c) Effect on the field winding voltage for case (2)



(d) Effect on the output speed for case (2)

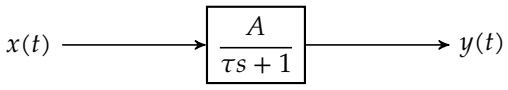
Figure 3.3: Illustration of the steady-state behavioral effect of saturation on the tolerance to disturbance signals (the load torque) for two cases: (1) $K_a = 9$ and $v_{IN} = 50\,\text{V}$, and (2) $K_a = 99$ and $v_{IN} = 45.455\,\text{V}$. The saturation is caused by a clipping circuit that prohibits the voltage on the field winding to go beyond $\pm 225\,\text{V}$.

3.3 Response speed

3.3.1 Principle

The reaction speed of a linear system is largely dependent on the location of its poles and zeros. Feedback influences those locations and therefore feedback also has a significant impact on the reaction speed of a system. We will illustrate this by considering a first-order system.

Without feedback Consider a first-order system with low-frequency gain A:



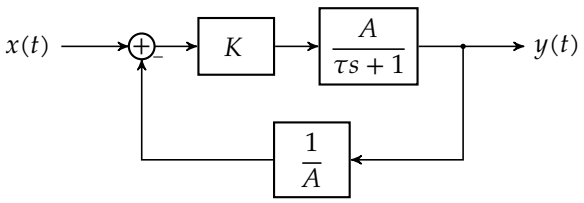
We can readily inspect its overall transfer function to be:

$$T(s) = \frac{A}{\tau s + 1}$$

Obviously, the pole of this system is located at $s = -1/\tau$ and that the 2%-settling time of this system is approximated by:

$$T_{s,0.02} = 4\tau$$

With feedback Let's consider the same first-order system and equip it with significant loop gain and a feedback factor that makes sure the low-frequency gain again equals A:



Solving this block diagram for the transfer function, yields:

$$T(s) = \frac{K\frac{A}{\tau s+1}}{1 + \frac{K}{A}\frac{A}{\tau s+1}} = \frac{KA}{\tau s + 1 + K} \approx \frac{KA}{\tau s + K}$$

The pole of this system is located at $s = -\frac{1+K}{\tau}$. The feedback has pushed it away to a higher frequency. This results in a much lower 2%-settling time:

$$T_{s,0.02} = \frac{4\tau}{K+1} \approx \frac{4\tau}{K}$$

Conclusion: feedback can cause a system to exhibit a much faster response time. Can we increase the response time without limit? No. There are two factors prohibiting this:

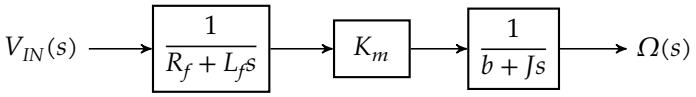
1. The increased response requires higher signal levels. Due to physical constraints these levels are limited and therefore so is the speed up.
2. Increasing the loop gain might render the loop unstable.

Note: we used a first-order system to illustrated the positive effect of feedback on the reaction speed. However, this also holds for higher-order systems. As an exercise, you might want to try to prove that the poles of a second-order system (with complex conjugate poles) also will exhibit a shift of its poles: the imaginary part of the poles will shift to higher values, resulting in the same effect: the system will become faster. However, it will also suffer from a decrease in damping. Try this!

3.3.2 Example

Consider once more our DC-motor in field control mode. Again we use the numerical values of Table 3.1.

Without feedback The block diagram of the system without feedback is drawn below:



Its transfer function is:

$$T(s) = \frac{\Omega(s)}{V_{IN}(s)} = \frac{K_m}{(R_f + L_f s)(b + Js)}$$

If we feed a step input of 45 V to the system, we can calculate the corresponding output in the Laplace domain:

$$\begin{aligned}
\Omega(s) = T(s)\frac{45\,\mathrm{V}}{s} &= 45\,\mathrm{V} \cdot \frac{K_m}{s(R_f + L_f s)(b + Js)} \\
&= 45\,\mathrm{V} \cdot \frac{\frac{K_m}{L_f J}}{s(s + \frac{R_f}{L_f})(s + \frac{b}{J})} \\
&\Big\downarrow \tau_f = \frac{L_f}{R_f} \quad \text{and} \quad \tau_l = \frac{J}{b} \\
&= 45\,\mathrm{V} \cdot \frac{K_m}{R_f b} \cdot \frac{\frac{1}{\tau_f \tau_l}}{s(s + \frac{1}{\tau_f})(s + \frac{1}{\tau_l})} \\
&\downarrow \text{partial fraction expansion} \\
&= 45\,\mathrm{V} \cdot \frac{K_m}{R_f b} \cdot \left(\frac{1}{s} + \frac{\frac{\tau_f}{\tau_l - \tau_f}}{s + \frac{1}{\tau_f}} + \frac{\frac{\tau_l}{\tau_f - \tau_l}}{s + \frac{1}{\tau_l}} \right)
\end{aligned}$$

and by inverse transformation also in the time domain:

$$\omega(t) = 45\,\mathrm{V} \cdot u(t) \cdot \frac{K_m}{R_f b} \cdot \left(1 + \frac{\tau_f}{\tau_l - \tau_f} \mathrm{e}^{-\frac{t}{\tau_f}} + \frac{\tau_l}{\tau_f - \tau_l} \mathrm{e}^{-\frac{t}{\tau_l}} \right)$$

If we compare the time constants of the two exponentials, we clearly see that the inertial time constant is much larger than the electrical time constant:

$$\tau_l = \frac{J}{b} = 10\,\mathrm{s} \gg 0.1\,\mathrm{s} = \frac{L_f}{R_f} = \tau_f$$

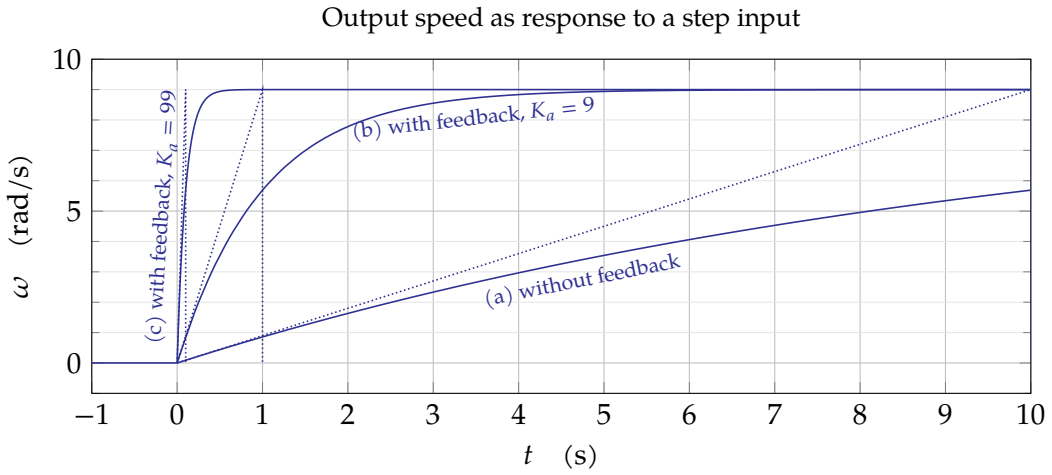


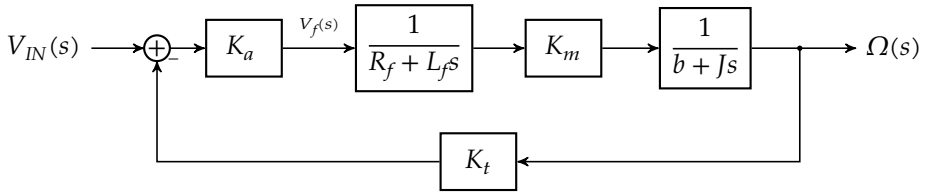
Figure 3.4: Response speed of the DC motor setup in field control mode, neglecting τ_f, (a) without feedback, (b) with feedback, $K'_{DC} = 9$ and (c) with feedback, $K'_{DC} = 99$.

Therefore, for a sufficiently large timescale the time-domain waveform can be approximated as:

$$\omega(t) \approx 45\,\mathrm{V} \cdot u(t) \cdot \frac{K_m}{R_f b} \cdot \left(1 - \mathrm{e}^{-\frac{t}{\tau_l}}\right)$$

Given the fact that $\frac{K_m}{R_f b} = 0.2\,\mathrm{Vs}$, the output will exponentially rise to $\omega = 9\,\mathrm{rad/s}$. This has been depicted in Figure 3.4.

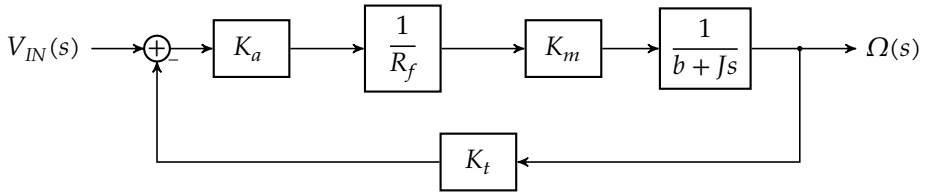
With feedback Now, let's close the loop and see what happens:



To get a similar overall gain, we set:

$$\frac{1}{K_t} \approx \frac{K_m}{R_f b} = \frac{20\,\mathrm{Nm/A}}{10\,\Omega \cdot 10\,\mathrm{Nm/(rad/s)}} = 0.2\,\mathrm{Vs}$$

Let's start with the assumption that $\tau_l \gg \tau_f$ and therefore, the system can be approximated by:



Using for example Mason's rule, we can determine the overall gain of this system. To this end,

we again use the *DC loop gain without integrators*[1]

$$K'_{DC} = \frac{K_a K_m K_t}{b R_f}$$

In this way, we can write:

$$T(s) = \frac{\Omega(s)}{V_{IN}(s)} = \frac{\frac{K'_{DC}}{K_t}\frac{1}{\tau_l s+1}}{1 + K'_{DC}\frac{1}{\tau_l s+1}} = \frac{1}{K_t}\frac{K'_{DC}}{\tau_l s + 1 + K'_{DC}}$$

with $\tau_l = \frac{J}{b}$

If we set $K'_{DC} = 9$ by setting $K_a = 9$, we must apply a step input of 50 V to have the output evolve to $\omega = 9$ rad/s.

The output then becomes:

$$\begin{aligned}
\Omega(s) &= 50\,\mathrm{V} \cdot \frac{1}{s} \cdot T(s) = 50\,\mathrm{V} \cdot \frac{1}{s} \cdot \frac{K'_{DC}}{K_t \tau_l} \frac{1}{s + \frac{1+K'_{DC}}{\tau_l}} \\
&= 50\,\mathrm{V} \cdot \frac{K'_{DC}}{K_t \tau_l} \cdot \frac{1}{s} \cdot \frac{1}{s + \frac{1+K'_{DC}}{\tau_l}} \\
&\downarrow \text{Partial fraction expansion} \\
&= 50\,\mathrm{V} \cdot \frac{K'_{DC}}{K_t \tau_l} \cdot \left(\frac{\frac{\tau_l}{1+K'_{DC}}}{s} - \frac{\frac{\tau_l}{1+K'_{DC}}}{s + \frac{1+K'_{DC}}{\tau_l}} \right) \\
&= 50\,\mathrm{V} \cdot \frac{1}{K_t} \cdot \frac{K'_{DC}}{1 + K'_{DC}} \cdot \left(\frac{1}{s} - \frac{1}{s + \frac{1}{\tau_c}} \right) \qquad \text{with } \tau_c = \frac{\tau_l}{1 + K'_{DC}}
\end{aligned}$$

In the time domain, this becomes:

$$\omega(t) = 50\,\mathrm{V} \cdot \frac{1}{K_t} \cdot \frac{K'_{DC}}{1 + K'_{DC}} \cdot u(t) \left(1 - \mathrm{e}^{-\frac{t}{\tau_c}}\right)$$

The corresponding waveform has been plotted in Figure 3.4. As one can see the speed up is considerable.

If we increase K_a to 99 and apply a corrected input step of 45.455 V, we get an even speedier ramp up to $\omega = 9$ rad/s.

Limiting factors Finally, let's investigate the limiting factors for this feedback design.

Limited signal range Given the fact that we added a considerable gain block to provide the input voltage for the field winding of the motor, we may expect problems with the signal range at that point. Let's calculated the relationship between $V_{IN}(s)$ and $V_f(s)$ under the

[1] Of course, in this case, there are no integrators in the loop.

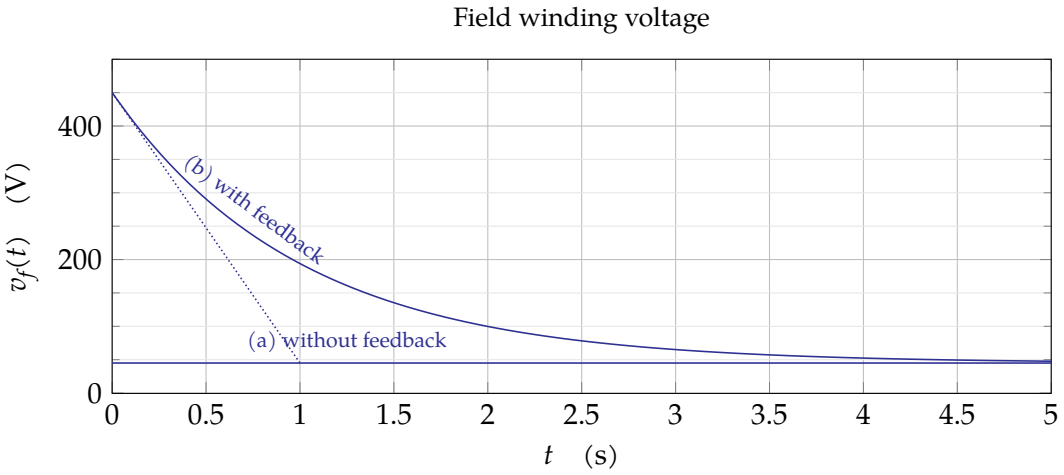


Figure 3.5: The field winding voltage of a field-controlled motor (a) without feedback, and (b) with feedback with $K_a = 9$, when fed with a 50 V input step.

assumption that $\tau_f \ll \tau_l$:

$$\begin{aligned}
\frac{V_f(s)}{V_{IN}(s)} &= \frac{K_a}{1 + \frac{K_a K_m K_t}{R_f} \frac{1}{(Js+b)}} \\
&= \frac{K_a}{1 + \frac{K_a K_m K_t}{R_f b} \frac{1}{(\tau_l s+1)}} && \text{with } \tau_l = \frac{J}{b} \\
&= \frac{K_a (\tau_l s + 1)}{\tau_l s + 1 + K'_{DC}} && \text{with } K'_{DC} = \frac{K_a K_m K_t}{R_f b} \\
&= K_a \frac{s + \frac{1}{\tau_l}}{s + \frac{1}{\tau_c}} && \text{with } \tau_c = \frac{\tau_l}{1 + K'_{DC}}
\end{aligned}$$

The effect of feeding the system with a step input of 50 V can be calculated to result in the following field winding voltage in the time-domain (try this yourself!):

$$v_f(t) = 50\,\text{V} \cdot K_a \cdot \frac{\tau_c}{\tau_l} \left(1 - \left(1 - \frac{\tau_l}{\tau_c}\right) \mathrm{e}^{-\frac{t}{\tau_c}}\right)$$

Consequently:

$$v_f(0) = 50\,\text{V} \cdot K_a \qquad \lim_{t\to\infty} v_f(t) = 50\,\text{V} \cdot K_a \cdot \frac{\tau_c}{\tau_l} = 50\,\text{V} \cdot \frac{K_a}{1 + K'_{DC}}$$

The full field winding voltage profile has been drawn in Figure 3.5. As we can see, the modest input voltage of 50 V is is amplified to a full 450 V. We must make sure that the motor is capable of dealing with this voltage. If not, we have to limit the loop gain (by limiting K_a. This will also limit the achievable speed up. E.g., if the motor can only take 200 V, K_a will be limited to 4. In that case the speed up due to the feedback will be limited to $1 + K'_{DC} = 5$.

Stability Earlier in this example, we neglected the inductive current delay in the field winding because its time constant was two orders of magnitude smaller than the inertial time constant of the motor. However, with feedback τ_l is virtually lowered to τ_c, whose values are more

comparable to τ_f. Indeed for $K_a = 9$ and $K_a = 99$, the value of τ_c is only $10\tau_f$ or equal to τ_f respectively. Therefore, neglecting τ_f was not such a good idea.

Calculating the gain of the system with feedback yields:

$$\begin{aligned}
\frac{\Omega(s)}{V_f(s)} &= \frac{\dfrac{K_a K_m}{(R_f + L_f s)(b + Js)}}{1 + \dfrac{K_a K_m K_t}{(R_f + L_f s)(b + Js)}} \\
&= \frac{\dfrac{K'_{DC}/K_t}{(\tau_f s + 1)(\tau_l s + 1)}}{1 + \dfrac{K'_{DC}}{(\tau_f s + 1)(\tau_l s + 1)}} \qquad \text{with } \tau_f = \frac{L_f}{R_f} \text{ and } \tau_l = \frac{J}{b} \\
&= \frac{K'_{DC}/K_t}{(\tau_f s + 1)(\tau_l s + 1) + K'_{DC}} \\
&= \frac{K'_{DC}/K_t}{\tau_f \tau_l s^2 + (\tau_f + \tau_l)s + 1 + K'_{DC}}
\end{aligned}$$

Let's rework this towards the standard form for second-order systems:

$$\begin{aligned}
\frac{\Omega(s)}{V_f(s)} &= \frac{K'_{DC}/K_t}{1 + K'_{DC}} \cdot \frac{\dfrac{1 + K'_{DC}}{\tau_f \tau_l}}{s^2 + \dfrac{\tau_f + \tau_l}{\tau_f \tau_l} s + \dfrac{1 + K'_{DC}}{\tau_f \tau_l}} \\
&= \frac{K'_{DC}/K_t}{1 + K'_{DC}} \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}
\end{aligned}$$

with

$$\begin{aligned}
\omega_n &= \sqrt{\frac{1 + K'_{DC}}{\tau_f \tau_l}} \\
\zeta &= \frac{\tau_f + \tau_l}{2\tau_f \tau_l \omega_n}
\end{aligned}$$

Considering gains factors $K_a = 9$ and $K_a = 99$, we obtain the following values:

$K_a(-)$	$\omega_n(\text{rad/s})$	$\zeta(-)$
9	3.1623	1.597
99	10	0.505

For these two cases, the rotational speed over time as a response to a 50 V unit step (in the case $K_a = 9$) and a 45.45 V unit step (in the case $K_a = 99$) is graphed in Figure 3.6. This clearly shows that with increasing K_a the feedback loop improves its behavior regarding reaction speed, but also gets more and more undamped, reaching borderline stability (i.e. pure oscillation) for an infinite gain K_a. We cannot but conclude: this system evolves to an unstable

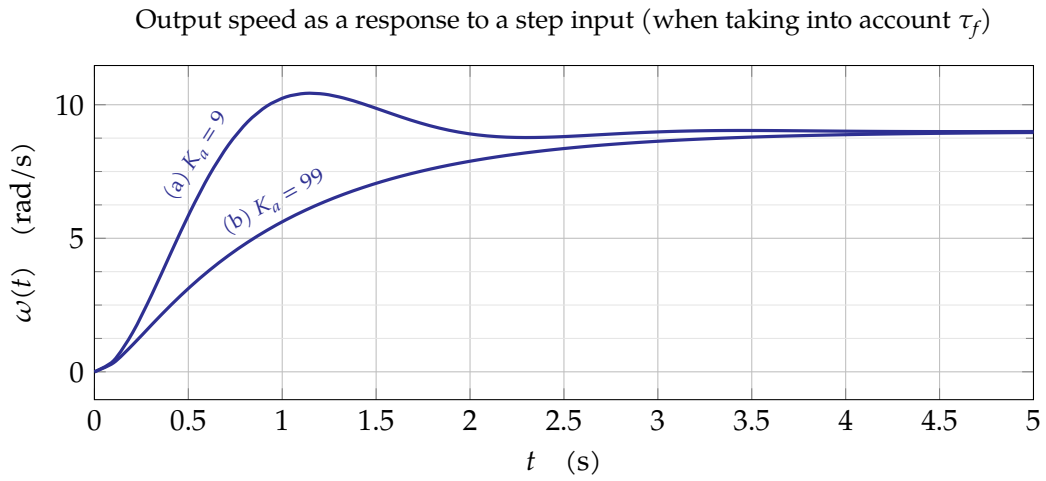


Figure 3.6: Rotational speed as a function of time of the DC motor in field control mode when not neglecting τ_f, using feedback with (a) $K_a = 9$ and $v_{IN}(t) = 50\,\mathrm{V} \cdot u(t)$ and with (b) $K_a = 99$ and $v_{IN}(t) = 45.45\,\mathrm{V} \cdot u(t)$.

system. We will see later that systems with more poles and zeros may evolve to truly unstable systems and that usually increasing the gain too much in the end has an adverse effect on stability.

3.4 Conclusion

Feedback brings many beneficial properties to the table: insensitivity to parameter variations, robustness w.r.t. disturbance signals and an increased reaction speed. To this end, the loop gain must be sufficiently high. However, the loop gain cannot be increased indefinitely, as the limited signal range and instability will spoil the party. In the next chapter we will study the stability of control systems to allow us to design stable feedback systems with sufficient loop gain that meet the specified design targets.

Chapter 4

Stability in the Laplace Domain

Once again, our goal is to use linear systems in a *controlled manner*. One of these aspects of stability is keeping the system stable.

In this chapter, we will study stability in the Laplace-domain. We will use a pure algebraic method (the Routh-Hurwitz criterion) and a graphical one (Evans' root locus method).

After having read and studied the chapter, you are expected to be able to:

- use the Routh-Hurwitz criterion to analyze the stability of a linear system,
- use Evans' root locus technique to analyze the stability of a linear feedback system as a function of the loop gain.

The root locus technique can also be applied in the digital domain (using the Z-transform). We will come back to this later, in the chapter on digital control systems on page 179.

4.1 The Routh-Hurwitz criterion

In the Laplace-domain, the transfer function of a linear time-invariant system can be written as a ratio of two polynomials in the Laplace variable s:

$$T(s) = \frac{N(s)}{D(s)}$$

The polynomial of the denominator $D(s)$ is the so-called *characteristic polynomial* of the system.[1] The roots of this polynomial are called the poles of the system. Their location in the complex plane determines the stability of the system. In the Laplace-domain, they must all be located in the left-half plane in order to yield a stable system.

Calculating the roots of $D(s)$ by hand is easy for first and second-degree polynomials, and using nowadays numerical solvers, even for higher-degree polynomials.

In MATLAB, calculating the roots of a polynome is as easy as arranging its coefficients in a vector and handing that vector over to the roots function, e.g. solving $x^3 - 3x^2 - 3 = 0$

```
c = [ 1 -3 0 -3 ];
roots( c );
```

[1] Note that in case the system is a feedback system, $T(s)$ corresponds to the overall transfer function!

However, before the abundant availability of computing power, analytical methods were developed to assess the location of the poles and zeros without actually calculating them.

An example of such an analytical technique has been independently developed by Edward John Routh [Rou77] and Adolf Hurwitz [Hur95], the former as an algorithm and the latter as a convenient tabular scheme. One could argue that due to the availability of powerful computers (and root solvers), this technique is obsolete. However, it allows for much more than the pure numerical analysis of a specific system with given coefficient values. Therefore, it makes sense to investigate it in more depth.

4.1.1 The basic idea

The Routh-Hurwitz criterion is a test to see whether the roots of the characteristic polynomial are in the left-hand plane, i.e. they have a negative real part.

We will not prove or deduct the criterion. However, to give you a clue on its nature, let's consider as example a characteristic polynomial of 3$^{\text{rd}}$ degree that has a negative real root and two complex roots with negative part (i.e. all stable poles). Assuming a, b and c to be positive reals, this boils down to:

$$T(s) = (s + a)(s + b + jc)(s + b - jc)$$

Elaborating this further yields:

$$\begin{aligned} T(s) &= (s + a)(s + b + jc)(s + c - jc) \\ &= (s + a)(s^2 + 2bs + b^2 + c^2) \\ &= s^3 + (a + 2b)s^2 + (2ab + b^2 + c^2)s + a(b^2 + c^2) \end{aligned}$$

It is clear that all the coefficients of this polynomial are positive. We will not do this, but one can generalize this for polynomials or arbitrary degree with real coefficients: if the real parts of the roots are negative, then all coefficients of the polynomial should be positive.

Obviously, this is a necessary condition for stability, however, it is not a sufficient condition. E.g. $s^3 + s^2 + 2s + 3$ has all positive coefficients, but has two poles in the right-half plane.

The Routh-Hurwitz criterion *does* provide a sufficient condition.

4.1.2 The procedure

Consider as example a characteristic polynomial of 6$^{\text{th}}$ degree:

$$D(s) = a_6 s^6 + a_5 s^5 + a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0$$

Hurwitz proposed to arrange these coefficients in a tabular format as:

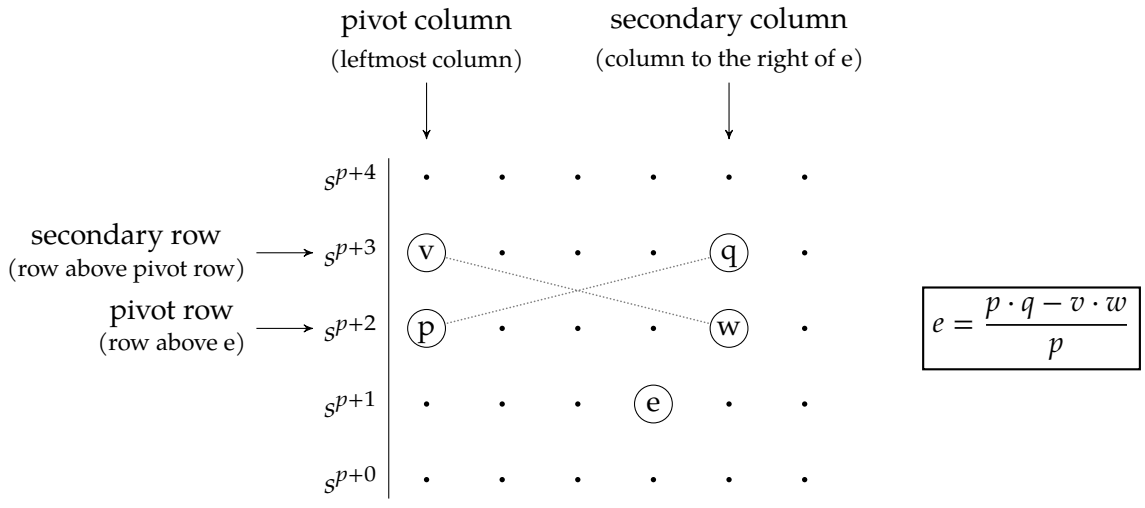
s^6	a_6	a_4	a_2	a_0
s^5	a_5	a_3	a_1	
s^4	b_1	b_2	b_3	
s^3	b_4	b_5		
s^2	b_6	b_7		
s^1	b_8			
s^0	b_9			

in which the values of b_1 to b_9 are to be calculated.

To calculate a specific value of b (which will call the *element*), find the leftmost coefficient of the row above the element and let's call that the *pivot*. To allow for a generic formulation of the rule, we will call

- the row of the pivot *the pivot row* and likewise the column of the pivot *the pivot column*,
- the row above the pivot, *the secondary row*,
- the column to the right of the element, *the secondary column*.

These terms can be inspected on the diagram below. The equation used to calculate e has been indicated next to it.



You will appreciate the clarity of the diagram when compared to a textual description of the calculation rule:
The value of e can be calculated as the division of a numerator divided by the pivot. The numerator can be calculated as 'the pivot times the value on the secondary row and secondary column minus the value on the secondary row and the pivot column times the value on the pivot row and the secondary column'.
Quite a mouthful! Luckily our brain works a little bit better on diagrams. This leads to the

following equations for b_1 to b_9:

$$b_1 = \frac{a_5 a_4 - a_6 a_3}{a_5}$$
$$b_2 = \frac{a_5 a_2 - a_6 a_1}{a_5}$$
$$b_3 = \frac{a_5 a_0 - a_6 0}{a_5} = a_0$$
$$b_4 = \frac{b_1 a_3 - a_5 b_2}{b_1}$$
$$b_5 = \frac{b_1 a_1 - a_5 b_3}{b_1}$$
$$b_6 = \frac{b_4 b_2 - b_1 b_5}{b_4}$$
$$b_7 = \frac{b_4 b_3 - b_1 0}{b_4} = b_3$$
$$b_8 = \frac{b_6 b_5 - b_4 b_7}{b_6}$$
$$b_9 = \frac{b_8 b_7 - b_6 0}{b_8} = b_7$$

The criterion can now be formulated based on this diagram:

Routh-Hurwitz criterion
The number of roots of the characteristic polynomial that have a positive part is equal to the number of sign changes in the pivot column.
If zeros occur in the pivot column, some roots are on the imaginary axis. Their value can be determined by finding the roots of auxiliary equation. This is the row above the zero, equipped with the correct powers of s.

4.1.3 Examples

Let's illustrate this using some examples.

Example 1
Consider our earlier example: $D(s) = s^3 + 2s^2 + s + 3$. The Routh-Hurwitz diagram becomes:

s^3	1	1
s^2	2	3
s^1	$-1/2$	
s^0	3	

Conclusion: the pivot column has two sign changes and therefore two roots in the right-half plane. The corresponding system is unstable. You can verify this in MATLAB (using the `roots` function) and will find:

$$p_0 = 0.08728 + j1.17131$$
$$p_1 = 0.08728 - j1.17131$$
$$p_2 = -2.17456$$

Example 2

The true value of the Routh-Hurwitz criterion shows itself when the characteristic polynomial contains parameter. For example: $s^3 + 2s^2 + s + K$, with K a parameter that one can freely choose. The key question is: for which values of K is the system stable? You again might use the `roots` function of MATLAB and do the math for a dense sampling of K. However, this is a waste of computing power. Routh-Hurwitz is much better at this problem:

$$\begin{array}{c|cc} s^3 & 1 & 1 \\ s^2 & 2 & K \\ s^1 & \frac{2-K}{2} & \\ s^0 & K & \end{array}$$

Given the fact that the two first pivot elements are positive, the bottom two also need to be positive, i.e.

$$\frac{2-K}{2} > 0$$
$$K > 0$$

This can be summarized as: $0 < K < 2$. However, the Routh-Hurwitz criterion also teaches us something about the boundary cases. If $K = 0$, the pivot element corresponding to s^0 becomes zero, i.e. there are some roots on the imaginary axis. Determining them means solving for the roots of the row above the zero:

$$\frac{2-K}{2}s = 0$$

With $K = 0$, this boils down to $s = 0$.

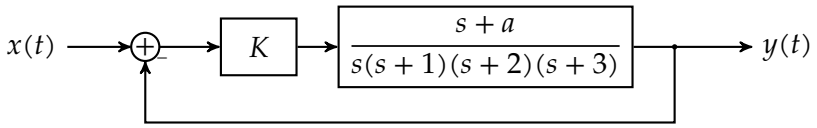
Likewise, if $K = 2$, the pivot element corresponding to s^1 becomes zero. Solving for the imaginary roots is as simple as solving for the roots of the row above:

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

With $K = 2$, this boils down to: $s = \pm j$.

Example 3

And now for the full power of the Routh-Hurwitz criterion: consider the block diagram below.



Key question: for which values of a and K is the feedback system stable? In order to be able to apply the Routh-Hurwitz criterion, the first job is to calculate the overall transfer function:

$$\begin{aligned} T(s) &= \frac{K\frac{(s+a)}{s(s+1)(s+2)(s+3)}}{1 + K\frac{(s+a)}{s(s+1)(s+2)(s+3)}} \\ &= \frac{K(s+a)}{s^4 + 6s^3 + 11s^2 + (6+K)s + aK} \end{aligned}$$

We can now write down the Routh-Hurwitz diagram:

$$\begin{array}{c|ccc} s^4 & 1 & 11 & Ka \\ s^3 & 6 & 6+K & \\ s^2 & b_1 & b_2 & \\ s^1 & b_3 & & \\ s^0 & b_4 & & \end{array}$$

with

$$\begin{aligned} b_1 &= \frac{6 \cdot 11 - 1 \cdot (6+K)}{6} = \frac{60-K}{6} \\ b_2 &= Ka \\ b_3 &= \frac{b_1 \cdot (6+K) - 6 \cdot b_2}{b_1} \\ b_4 &= Ka \end{aligned}$$

Given the fact that the first two pivot elements are positive, we also need to require $b_1 > 0, b_3 > 0$ and $b_4 > 0$. The former corresponds to: $K < 60$, the latter to $kA > 0$.

The middle one is a bit trickier, but we can be smart. Given the fact that $K < 60$, we know that $b_1 > 0$. This allows for some simplification:

$$\begin{aligned} b_3 = \frac{b_1 \cdot (6+K) - 6 \cdot b_2}{b_1} &> 0 \\ b_1 \cdot (6+K) - 6 \cdot b_2 &> 0 \\ \frac{(60-K)(6+K)}{6} - 6Ka &> 0 \\ (60-K)(6+K) - 36Ka &> 0 \end{aligned}$$

It is easier to solve the latter equation for a than to solve it for K, so that is what we will do:

$$Ka < \frac{360 + 54K - K^2}{36}$$

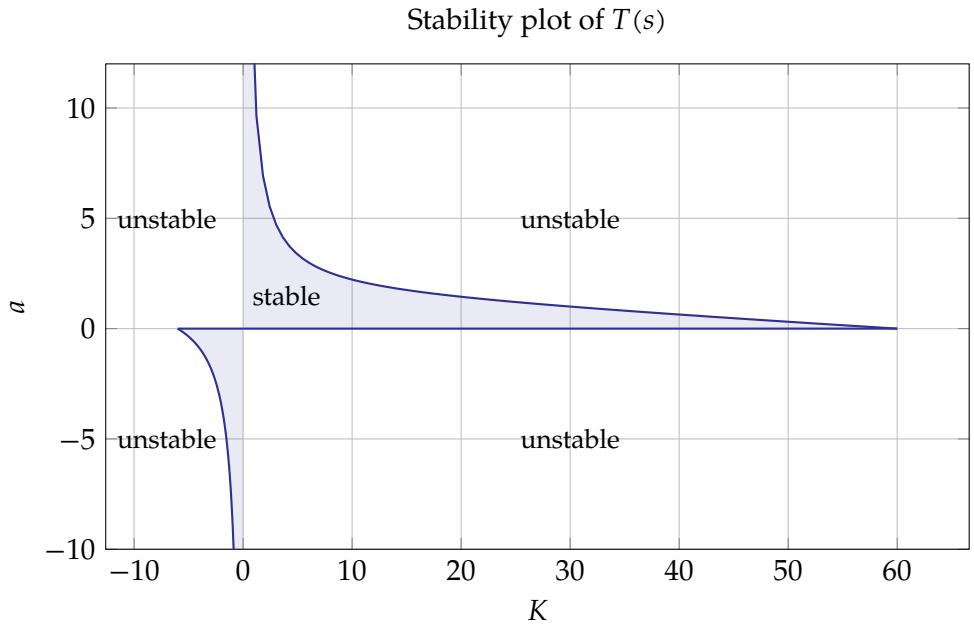
For positive K this becomes:

$$a < \frac{360 + 54K - K^2}{36K}$$

For negative K this becomes:

$$a > \frac{360 + 54K - K^2}{36K}$$

Plotting a as a function K for $-6 \leq K \leq 60$ sheds sufficient light on the region of stability:



Could you achieve the same result by root solving? If so, at what price? If you think about it, you will see the power of the work of Routh and Hurwitz.

Exercises

Exercise 4.1.3-1: Use the Routh-Hurwitz criterion to assess the stability of a system with the following transfer function:

$$T(s) = \frac{s^2 + 3s - 2}{s^4 + 6s^3 + 1s^2 - 14s + 30}$$

Exercise 4.1.3-2: Use the Routh-Hurwitz criterion to assess the stability of a system with the following transfer function:

$$T(s) = \frac{s + 3}{s^4 + 12s^3 + 52s^2 + 100s + 75}$$

Exercise 4.1.3-3: Use the Routh-Hurwitz criterion to assess the stability of a system with the following transfer function:

$$T(s) = \frac{4s - 3}{s^4 + 7.5s^3 + 12s^2 + s + 16}$$

4.2 The root locus method

4.2.1 The basic idea

Let's reconsider one of the earlier examples, a system with characteristic polynomial

$$D(s) = s^3 + s^2 + 2s + K$$

We can choose any real value for the parameter K, and in the previous section we determined (using the Routh-Hurwitz criterion) that the system is stable when $0 < K < 2$. However, it would be interesting to know where the poles of the system are located as a function of K. In fact, we'd like to perform a parameter sweep for K and calculate the poles for all considered values.

Using MATLAB this is most easy:

```
N = 11;
Kspace = linspace(-5,5,N);
poletable = zeros( N,3 );
for i = 1:N
  D = [ 1 1 2 Kspace(i) ];
  poletable(i,:) = roots( D );
end
```

The obtained values have been summarized below:

K	p_1	p_2	p_3
-5	$-1.06625 + 1.81057j$	$-1.06625 - 1.81057j$	1.13249
-4.5	$-1.00000 + 1.73205j$	$-1.00000 - 1.73205j$	1.00000
-3	$-0.92187 + 1.64493j$	$-0.92187 - 1.64493j$	0.84373
-2	$-0.82531 + 1.54687j$	$-0.82531 - 1.54687j$	0.65063
-1	$-0.69632 + 1.43595j$	$-0.69632 - 1.43595j$	0.39265
0	$-0.50000 + 1.32288j$	$-0.50000 - 1.32288j$	0.00000
1	$-0.21508 + 1.30714j$	$-0.21508 - 1.30714j$	-0.56984
2	$-0.00000 + 1.41421j$	$-0.00000 - 1.41421j$	-1.00000
3	$0.13784 + 1.52731j$	$0.13784 - 1.52731j$	-1.27568
4	$0.23898 + 1.62767j$	$0.23898 - 1.62767j$	-1.47797
5	$0.31990 + 1.71663j$	$0.31990 - 1.71663j$	-1.63980

For each value of K we obtain three roots that change location based on the value of K. We can see that when $K < 0$ the real root is positive and that for $K > 2$ the complex roots are located in the right-half plane (Fig. 4.1).

Increasing the sample density of K would even allow us to plot true lines for the paths that the roots follow. Using our MATLAB approach that would only require increasing the value of N in our script. The path that a root follows for varying values of K is the so-called *locus* of the root. Therefore, such a plot is called a *root locus plot*. Using a computer, generating these kinds of plots is most easy. By hand, this would be a true nightmare.

Luckily, analytical methods were developed by Walter Evans to sketch root locus plots 'by hand' [Eva50]. Again, one could argue that these techniques are obsolete due to the abundant availability of computing power. However, understanding the underlying principles of drawing root locus plots aids to the insight of systems and their stability. If you understand root locus plots well, you will be able to judge whether it makes sense to add an extra pole or zero to the loop gain of a system and even know where to put it!

Therefore, let's analyze this most powerful old school technique.

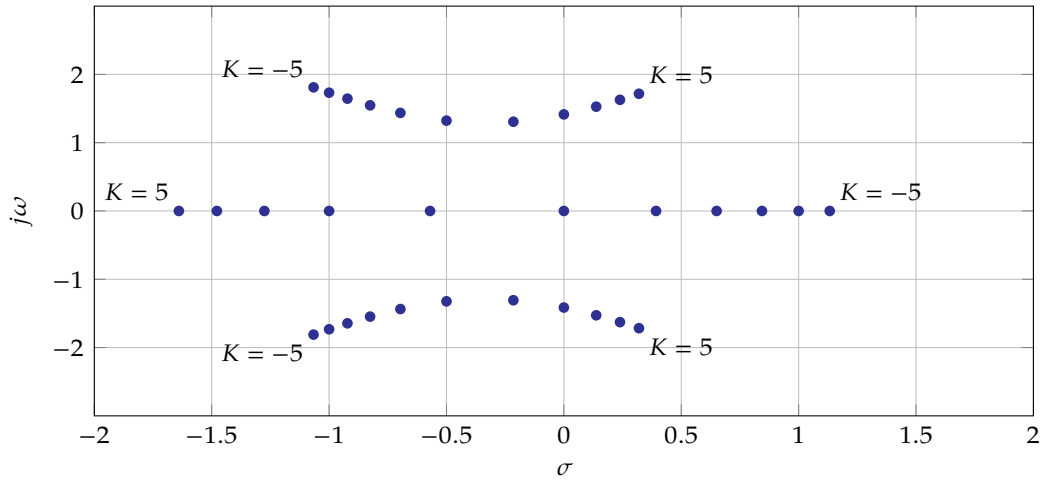


Figure 4.1: Locations of the roots of $s^3 + s^2 + 2s + K = 0$ as a function of K (integer values of K from -5 to 5).

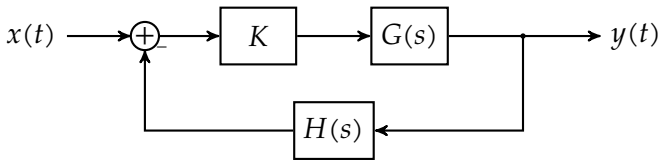
4.2.2 The procedure

The starting point of the root locus procedure is the ability to write the characteristic polynomial of your system as:

$$D(s) = 1 + K \cdot F(s) \tag{4.1}$$

with K the gain factor that is the parameter in our root locus plot.

This is not possible for all feedback systems, but for feedback systems with a single loop and a gain block in the loop, this always works. E.g. consider:



The transfer function of this feedback system can be written as:

$$T(s) = \frac{K \cdot G(s)}{1 + K \cdot \underbrace{G(s)H(s)}_{F(s)}}$$

The poles of this system occur for values of s that makes the characteristic polynomial zero, i.e.

$$1 + K \cdot F(s) = 0 \qquad \Leftrightarrow \qquad K \cdot F(s) = -1$$

As $K \cdot F(s)$ is a complex number, we are looking for numbers with magnitude 1 and a phase equaling 180° or aliases thereof. If we assume K to be a positive real number, than $K \cdot F(s) = -1$ can be written as combination of two conditions:

$$\text{magnitude/modulus condition: } K \cdot |F(s)| = 1$$
$$\text{phase condition: } \angle F(s) = 180° + m \cdot 360°, \text{ with } m \in \mathbb{Z}$$

Note that K is not present in the phase condition (as we restricted K to be a positive real number).

For LTI systems we can always decompose $F(s)$ as:

$$F(s) = \frac{\prod_{i=1}^{n_z}(s - z_i)}{\prod_{i=1}^{n_p}(s - p_i)} \qquad \text{with } n_z \text{ the number of zeros and } n_p \text{ the number of poles.}$$

And therefore:

$$1 + K \cdot F(s) = 0 \qquad \Leftrightarrow \qquad \prod_{i=1}^{n_p}(s - p_i) + K \cdot \prod_{i=1}^{n_z}(s - z_i) = 0 \tag{4.2}$$

Let's start to investigate some properties of the latter expression. Applying these properties in order will lead to a procedure to compose a root locus plot for any system whose characteristic polynomial can be written as (4.1).

In the following, we will assume that $n_p \geq n_z$, as is the case for any feasible physical system.

Property 1: the individual loci

When $K = 0$, equation (4.2) reduces to:

$$\prod_{i=1}^{n_p}(s - p_i) = 0$$

This means that for $K = 0$ the poles of the overall system coincide with the poles of $F(s)$. Conclusion: in each and every pole of $F(s)$ an individual locus starts.

When $K \to +\infty$, equation (4.2) reduces to:

$$\prod_{i=1}^{n_z}(s - z_i) = 0$$

This means that for $K \to +\infty$ the poles of the overall system coincide with the zeros of $F(s)$. Conclusion: every locus will end in a zero of $F(s)$. If $n_p > n_z$, then there are $n_p - n_z$ roots at infinity.

> **Rule no. 1:** We need to draw n_p loci, starting in poles and ending in the zeros of $F(s)$.

Property 2: the real axis

Starting from the phase condition, we can elaborate:

$$\angle F(s) = 180° + m \cdot 360° \text{ with } m \in \mathbb{Z}$$

$$\sum_{i=1}^{n_z} \angle(s - z_i) - \sum_{i=1}^{n_p} \angle(s - p_i) = 180° + m \cdot 360°$$

If we focus on values of s that are located on the real axis, then we can make a number of observations. Let's focus on $s = a$ with $a \in \mathbb{R}$ and check how poles and zeros contribute to the phase of $F(s)$ when $s = a$.

For real zeros:

- A real zero to the left of a has no phase contribution.
- A real zero to the right of a has a phase contribution of $180°$.

For real poles:

- A real pole to the left of a has no contribution to the phase value of $F(s)$.
- A real pole to the right of a has a phase contribution of $-180°$.

For complex conjugate poles or zeros[2], we can be very brief: they have no phase contribution!

Make sure you understand all these statements! If needed, sketch the situation in the Argand plane.

This leads to the following conclusion:

> **Rule no. 2:** Parts of the real axis that are located to the left of an odd number of poles or zeros of $F(s)$ are part of the root locus plot.

Property 3: symmetry As all poles and zeros occur as complex conjugate pairs, the situation does not change when flipping the pole-zero image over the real axes. Therefore:

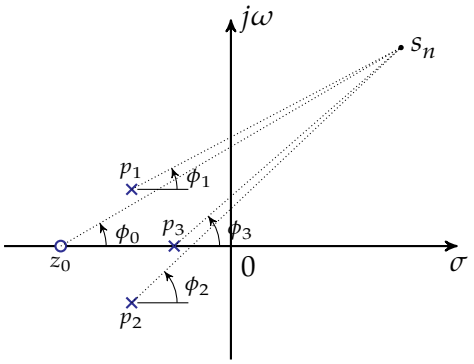
> **Rule no. 3:** The root locus plot is symmetrical w.r.t. the real axis.

Property 4: infinite zeros — asymptotes Given the fact that there are n_p - n_z zeros at infinity (for $F(s)$), there must be many asymptotes.

Direction The first question we need to ask ourselves is: in which direction do the asymptotes go to infinity?

The answer starts out in a simple way: consider a point s_n in the complex plane. If s_n is distant from the locations of the poles and zeros, all $s_n - z_i$ and $s_n - p_i$ have almost identical angles ϕ_i.

This has been illustrated in the drawing on the right.



If $s_n \to \infty$, they converge to a single ϕ.

We can use this insight in combination with the phase condition. If we label the angle under which the m-th asymptote is observed

[2]Remember: polynomials with real coefficients only have complex conjugate roots.

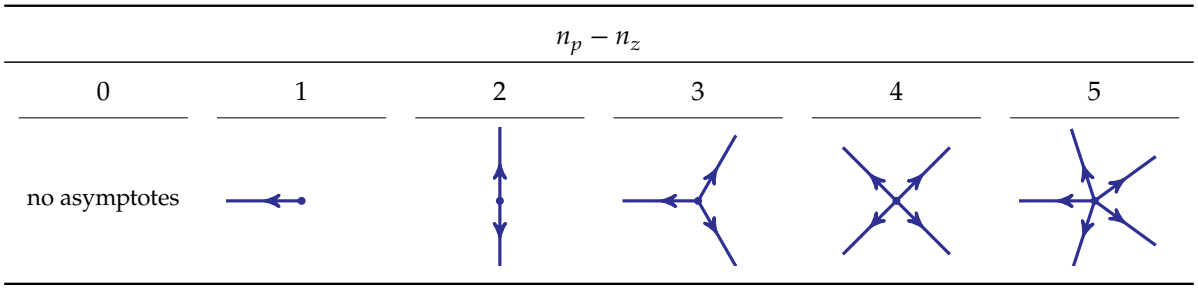


Figure 4.2: Asymptote patterns depending on $n_p - n_z$

by the poles and zeros as $\phi_{A,m}$, then

$$\sum_{i=1}^{n_z} \angle(s_n - z_i) - \sum_{i=1}^{n_p} \angle(s_n - p_i) = 180° + m \cdot 360°$$

$$n_z \phi_{A,m} - n_p \phi_{A,m} = 180° + m \cdot 360°$$

And this allows to calculate $\phi_{A,m}$ as

$$\phi_{A,m} = \frac{180° + m \cdot 360°}{n_z - n_p} \qquad \text{with } m \in \mathbb{Z}$$

This gives rise to the typical asymptote direction patterns as a function of the number of asymptotes $n_p - n_z$ that can be found in Figure 4.2.

Intersection The second question we need to ask ourselves is: where do the asymptotes intersect (the blue dot in the drawings above)? Obviously, this will be somewhere on the real axis (because of the symmetry property seen above).

To prepare ourselves for answering this question, let's redo the direction computation, but this time, analytically. We will be able to extend this reasoning to find the intersection of the asymptotes.

The starting point is the flat form of the transfer function $F(s)$:

$$F(s) = \frac{s^{n_z} + b_{n_z-1}s^{n_z-1} + \cdots + b_1 s + b_0}{s^{n_p} + a_{n_p-1}s^{n_p-1} + \cdots + a_1 s + a_0} \tag{4.3}$$

For points near to infinity $|s|$ is large and therefore we can use the following approximation:

$$F(s) \approx \frac{s^{n_z}}{s^{n_p}} = s^{n_z - n_p}$$

Plugging this approximation in the characteristic equation (with $K \in \mathbb{R}^+$) yields:

$$\begin{aligned} 1 + K \cdot s^{n_z - n_p} &= 0 \\ s^{n_p - n_z} &= -K \\ s &= \sqrt[n_p - n_z]{K} \cdot \mathrm{e}^{j\frac{\pi + m2\pi}{n_p - n_z}} \qquad \text{with } m \in \mathbb{Z} \end{aligned} \tag{4.4}$$

This — and this should not surprise us — is exactly the same result as before: for increasing K the obtained values move according to the direction patterns we found earlier. And indeed, equation (4.4) describes the asymptotes, starting from $s = 0$.

However, what we are looking for is a bundle of asymptotes starting in $s = \sigma_A$, i.e. described by:

$$s = \sigma_A + \sqrt[n_p - n_z]{K} \cdot e^{j\frac{\pi + m\pi}{n_p - n_z}} \qquad \text{with } m \in \mathbb{Z}$$

with $s = \sigma_A$ the starting point.

This would be the result of solving:

$$1 + K \cdot (s - \sigma_a)^{n_z - n_p} = 0$$

which corresponds to approximating $F(s)$ as:

$$F(s) \approx \frac{(s - \sigma_A)^{n_z}}{(s - \sigma_A)^{n_p}} = \frac{1}{(s - \sigma_A)^{n_p - n_z}}$$

Assuming $n_p > n_z$ and applying the binomial approximation trick explained at the end of appendix section A.1 on page 229 (applicable because $\sigma_A \ll |s|$), we can further simplify/approximate this result as:

$$\begin{aligned} F(s) &\approx \frac{1}{s^{n_p - n_z} - (n_p - n_z) s^{n_p - n_z - 1} \sigma_A} \\ &= \frac{1}{s^{n_p - n_z}} \cdot \frac{1}{1 - \frac{(n_p - n_z)\sigma_A}{s}} \end{aligned} \tag{4.5}$$

The remaining question is: what should σ_A's value be to make a good fit with $F(s)$ for large $|s|$? To this end, let's try to rework the original $F(s)$ into a similar shape, such that we can equate coefficients in the expressions.

Let's start again from (4.3) and consider not only keeping the terms with the highest power, but also the next ones. And from here on, the route is clear: do whatever it takes to morph this equation into the shape of (4.5). Therefore:

$$F(s) \approx \frac{s^{n_z} + b_{n_z - 1} s^{n_z - 1}}{s^{n_p} + a_{n_p - 1} s^{n_p - 1}} = \frac{s^{n_z}}{s^{n_P}} \cdot \frac{1 + \frac{b_{n_z - 1}}{s}}{1 + \frac{a_{n_z - 1}}{s}} = \frac{1}{s^{n_p - n_z}} \cdot \frac{1 + \frac{b_{n_z - 1}}{s}}{1 + \frac{a_{n_z - 1}}{s}}$$

Knowing that $|s|$ is large w.r.t. $b_{n_z - 1}$ (i.e. $|b_{n_z - 1}/s| \ll 1$), and knowing that $1 + z \approx \frac{1}{1 - z}$ for $|z| \ll 1$, we can further approximate:

$$\begin{aligned} F(s) &\approx \frac{1}{s^{n_p - n_z}} \cdot \frac{1}{\left(1 - \frac{b_{n_z - 1}}{s}\right)\left(1 + \frac{a_{n_z - 1}}{s}\right)} \\ &= \frac{1}{s^{n_p - n_z}} \cdot \frac{1}{1 - \frac{b_{n_z - 1}}{s} + \frac{a_{n_z - 1}}{s} - \frac{b_{n_z - 1}}{s}\frac{a_{n_p - 1}}{s}} \\ &\downarrow \quad \text{Neglect higher order term (small} \times \text{small} = \text{super-small)} \\ &\approx \frac{1}{s^{n_p - n_z}} \cdot \frac{1}{1 - \frac{b_{n_z - 1}}{s} + \frac{a_{n_z - 1}}{s}} \\ &= \frac{1}{s^{n_p - n_z}} \cdot \frac{1}{1 - \frac{b_{n_z - 1} - a_{n_z - 1}}{s}} \end{aligned} \tag{4.6}$$

Bingo! Equations (4.5) and (4.6) are now in the same shape and can only be equal if:

$$(n_p - n_z)\sigma_A = b_{n_z-1} - a_{n_z-1}$$
$$\sigma_A = \frac{b_{n_z-1} - a_{n_z-1}}{n_p - n_z}$$

Knowing that (try to prove this yourself):

$$b_{n_z-1} = -\sum_{i=1}^{n_z} z_i \qquad a_{n_p-1} = -\sum_{i=1}^{n_p} p_i$$

we obtain:

$$\sigma_A = \frac{\sum_{i=1}^{n_p} p_i - \sum_{i=1}^{n_z} z_i}{n_p - n_z}$$

We can summarize all this in

Rule no. 4: The asymptotes intersect in $s = \sigma_A$ and have direction angles equaling $\phi_{A,m}$, with:

$$\sigma_A = \frac{\sum_{i=1}^{n_p} p_i - \sum_{i=1}^{n_z} z_i}{n_p - n_z}$$
$$\phi_{A,m} = \frac{180° + m \cdot 360°}{n_z - n_p} \qquad \text{with } m \in \mathbb{Z}$$

Property 5: finite zeros and poles — tangent lines

Applying the phase condition to a specific finite or zero location, allows for calculating the tangent line to the root locus in that zero or pole. We will illustrate this in the examples.

This leads to:

Rule no. 5: Apply the phase condition to calculate tangent lines to the root locus in the locations of the poles and zeros of $F(s)$.

Property 6: break-away and merge-in points

Often, multiple loci break away from the real axis, or merge into the real axis, i.e. there are multiple poles coinciding at that break-away or merge-in point.

The shape in which they break away is identical to the asymptote shapes of Figure 4.2.

If there are break-away points, calculating the break-away points themselves makes sense to be able to detail our root-locus plot. The following procedure leads to the desired result (for a proof, see e.g. [DB11]).

1. Solve the characteristic polynomial for K
2. Calculate $\frac{\mathrm{d}K(s)}{\mathrm{d}s}$
3. Solve $\frac{\mathrm{d}K(s)}{\mathrm{d}s} = 0$ for s

Note that $K'(s) = 0$ is a required condition, but not a sufficient condition for being a break-away or a merge-in point. Therefore, not all roots will correspond to physical break-away or merge-in points. You might encounter:

- complex roots: these are to be rejected, as s has to be real
- real roots that are not on a part of the real axis that belongs to the locus: these are also to be rejected.

So along the same line: one should realize that the calculation only makes sense if there are break-away or merge-in points! In case there are no such points, one might still be able to solve $K'(s) = 0$ for s, but the results will be meaningless!

You can assess in advance whether it makes sense to calculate breakaway points or not.

Rule no. 6: Calculate the break-away or merge-in points by solving $K'(s) = 0$.

Property 7: oscillation points
When the poles of the overall system are on the imaginary axis, the system will start to oscillate. The system is then on the boundary of being stable. One can calculate these intersections with the imaginary axis by using the auxiliary equation from the Routh-Hurwitz criterion. We also will illustrate this in the examples.

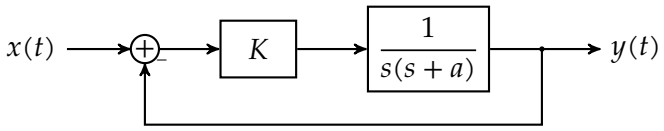
This leads to:

Rule no. 7: Determine the intersections with the imaginary axis (if any) using the auxiliary equation of the Routh-Hurwitz criterion.

4.2.3 Examples

Let's consider a number of examples to illustrate the root locus procedure.

Example 1
Consider the feedback system with input $x(t)$ and output $y(t)$. The parameters K and a are positive reals.



The overall transfer function equals:

$$H(s) = \frac{\frac{K}{s(s+a)}}{1 + \frac{K}{s(s+a)}} = \frac{K}{s(s+a) + K} \equiv \frac{N(s)}{D(s)}$$

Therefore $D(s) = s(s + a) + K$.

Considering K to be the gain and a to be a fixed parameter The first step is to morph the characteristic equation $D(s) = 0$ into the required shape: $1 + K \cdot F(s) = 0$. This is easy:

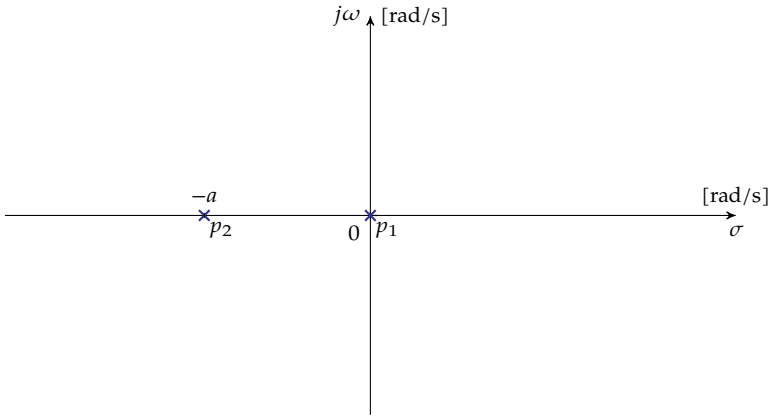
$$s(s+a) + K = 0$$
$$1 + K \cdot \underbrace{\frac{1}{s(s+a)}}_{=F(s)} = 0$$

Let's assume a to be positive. You can check the other cases yourself later.

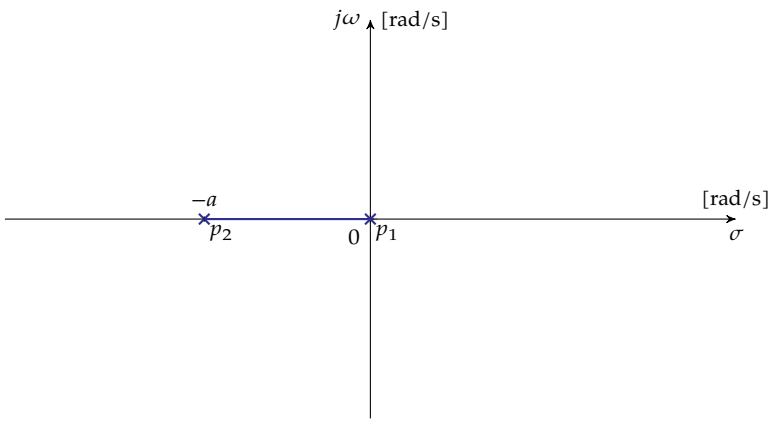
Now let's apply the *root locus rule set*:

Rule no. 1: We need to draw n_p loci, starting in poles and ending in the zeros of F(s).

Therefore, we start by drawing the pole-zero plot of $F(s)$ and realize that we need to draw 2 loci, that start in the poles and wander off to infinity.



Rule no. 2: Parts of the real axis that are located to the left of an odd number of poles or zeros of $F(s)$ are part of the root locus plot.



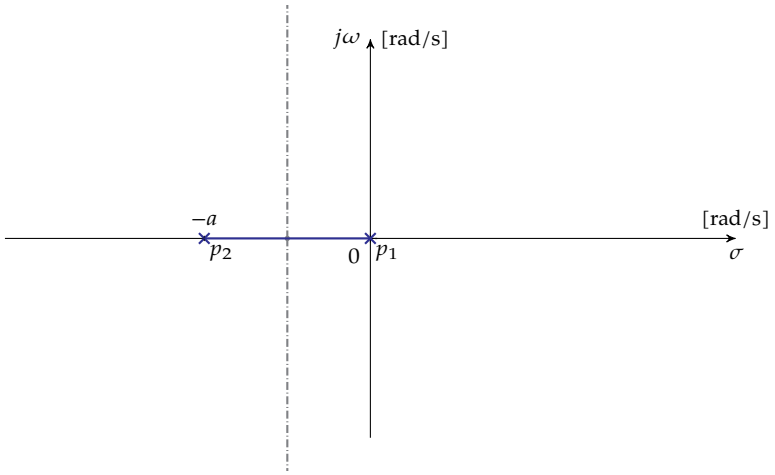
Rule no. 3: The root locus plot is symmetrical w.r.t. the real axis. Let's keep this in mind for what follows.

Rule no. 4: The asymptotes intersect in $s = \sigma_A$ and have direction angles equaling $\phi_{A,m}$.

Let's start by calculating these parameters:

$$\sigma_A = \frac{\sum p_i - \sum z_i}{n_p - n_z} = \frac{0 + (-a) - 0}{2 - 0} = -\frac{a}{2}$$
$$\phi_{A,m} = \frac{180^\circ + m \cdot 360^\circ}{n_p - n_z} = \frac{180^\circ + m \cdot 360^\circ}{2 - 0} = \pm 90^\circ$$

This yields the following asymptotes:



Rule no. 5: Apply the phase condition to calculate tangent lines to the root locus in the locations of the poles and zeros of $F(s)$.

Actually, given the fact that we already know that the real axis in between the two poles is part of the root locus, it does not make sense calculating the tangent directions. They are obviously 0° for p_2 and 180° for p_1.

Rule no. 6: Calculate the break-away or merge-in points by solving $K'(s) = 0$.

First, we solve the characteristic equation for K:

$$1 + K \cdot \frac{1}{s(s+a)} = 0$$
$$K = -s(s+a) = -s^2 - as$$

Determining K′(s) is easy:

$$K'(s) = -2s - a$$

And finally, solve

$$K'(s) = -2s - a = 0$$
$$s = -\frac{a}{2}$$

Therefore the breakaway point is located halfway in between the two poles p_1 and p_2.

Rule no. 7: Determine the intersections with the imaginary axis (if any) using the auxiliary equation of the Routh-Hurwitz criterion.

Let's start by composing the Routh-Hurwitz diagram. The starting point is the overall transfer function:

$$T(s) = \frac{K}{s^2 + as + K}$$

The diagram is easily determined to be:

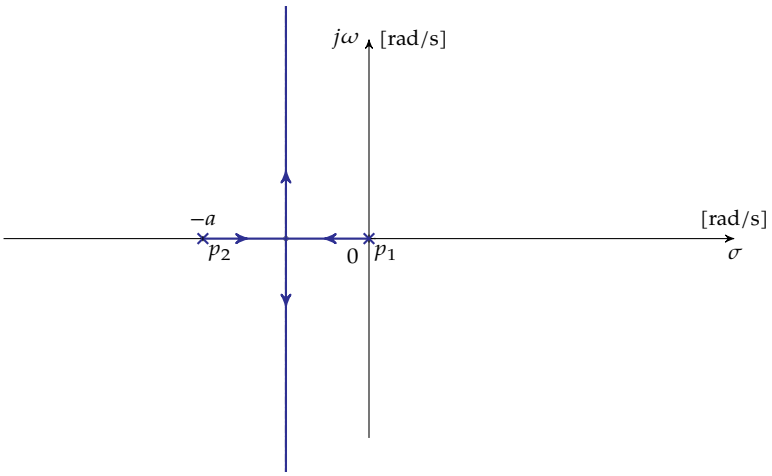
$$\begin{array}{c|cc} s^2 & 1 & K \\ s^1 & a & \\ s^0 & K & \end{array}$$

As long as $a > 0$ and $K > 0$, there are no sign changes in the pivot column and therefore no roots in the right-half plane. If $K = 0$, the bottom element becomes zero, and therefore the row above yields us the corresponding auxiliary equation, i.e.

$$as = 0$$

And indeed, for $K = 0$ we clearly have a pole in the origin.

The final result is the following root locus plot. We can see that in this case we have a stable system for $K > 0$.



Can you determine what happens if $a \leq 0$?

Considering a to be the gain and K to be a fixed parameter In this case, we will rework the characteristic equation $D(s) = 0$ into the shape $1 + a \cdot F(s) = 0$.

$$\begin{aligned} s(s + a) + K &= 0 \\ s^2 + K + as &= 0 \\ 1 + a \cdot \frac{s}{s^2 + K} &= 0 \end{aligned}$$

Let's assume K to be positive. You can check the other cases yourself later.

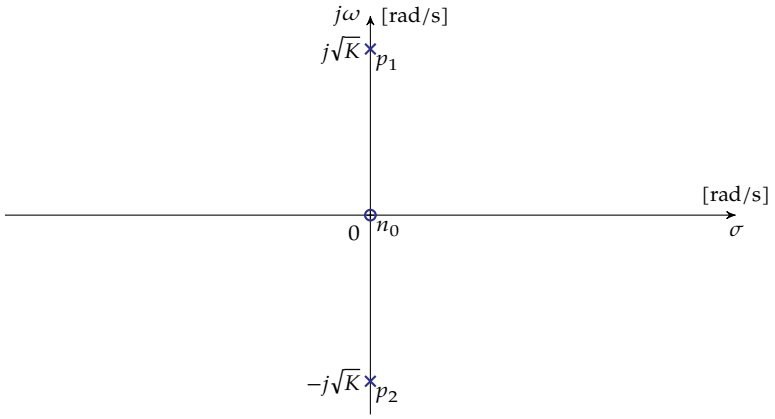
Now, let's apply the *root locus rule set*:

Rule no. 1: We need to draw n_p loci, starting in poles and ending in the zeros of F(s).

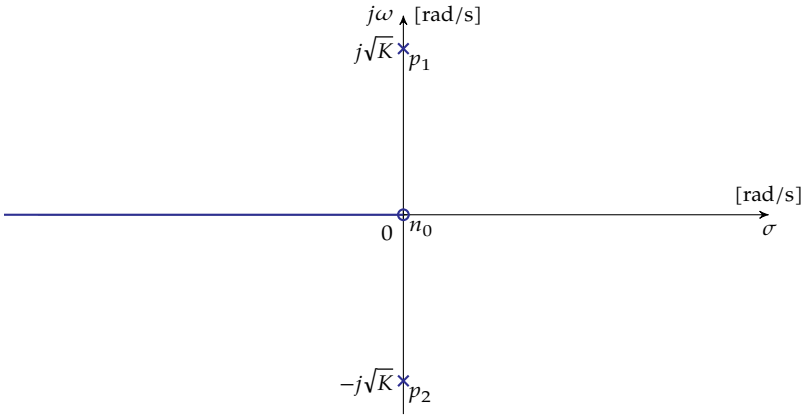
Therefore, we start by drawing the pole-zero plot of $F(s)$ and realize that we need to draw 2 loci, that start in the poles:

$$p_{1,2} = \pm j\sqrt{K}$$

One of them will wander off to infinity.



Rule no. 2: Parts of the real axis that are located to the left of an odd number of poles or zeros of $F(s)$ are part of the root locus plot.



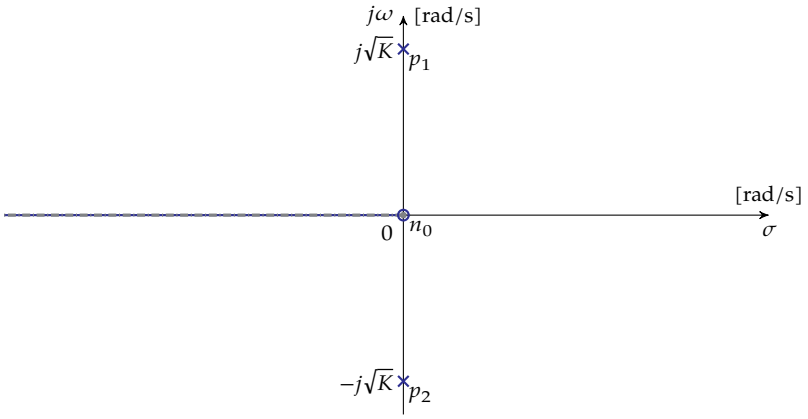
Rule no. 3: The root locus plot is symmetrical w.r.t. the real axis. Let's keep this in mind for what follows.

Rule no. 4: The asymptotes intersect in $s = \sigma_A$ and have direction angles equaling $\phi_{A,m}$.

Let's start by calculating these parameters:

$$\sigma_A = \frac{\sum p_i - \sum z_i}{n_p - n_z} = \frac{0 + j\sqrt{K} - j\sqrt{K} - 0}{2 - 1} = 0$$
$$\phi_{A,m} = \frac{180^\circ + m \cdot 360^\circ}{n_p - n_z} = \frac{180^\circ + m \cdot 360^\circ}{2 - 1} = 180^\circ$$

This yields the following asymptotes:



Rule no. 5: Apply the phase condition to calculate tangent lines to the root locus in the locations of the poles and zeros of $F(s)$.

Let's consider $s = p_1$ first. The phase condition tells us:

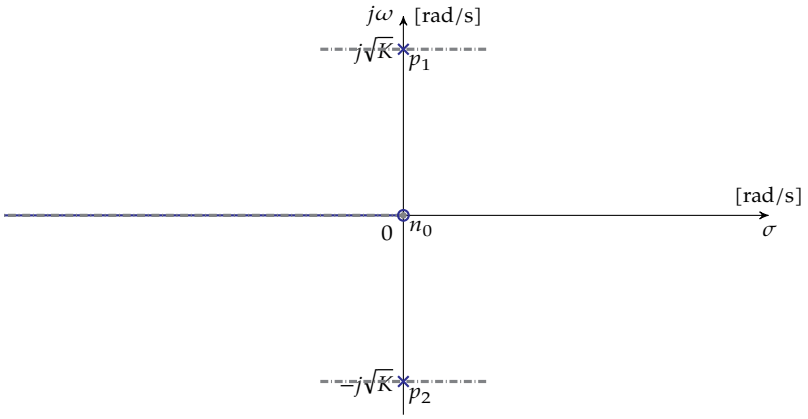
$$\angle F(s) = \angle(s - n_0) - \underbrace{\angle(s - p_1)}_{\equiv \phi_1} - \angle(s - p_2) = 180°$$

$$\downarrow \quad s = p_1$$

$$90° - \phi_1 - 90° = 180°$$

$$\phi_1 = -180°$$

Because of symmetry reasons, we don't need to calculate the tangent line direction for p_2. The symmetry property gives us the same insight without requiring extra calculations. Conclusion: the tangent lines in the poles are horizontal.



Rule no. 6: Calculate the break-away or merge-in points by solving $a'(s) = 0$.

First, we solve the characteristic equation for a:

$$1 + a \cdot \frac{s}{s^2 + K} = 0$$

$$a = -\frac{s^2 + K}{s}$$

Calculating the derivative is easy:

$$a' = -\frac{2s^2 - (s^2 + K)}{s^2}$$

And then we solve:

$$a' = -\frac{s^2 - K}{s^2} = 0$$

$$s = \pm\sqrt{K}$$

Only the negative value makes sense (the positive value occurs for $a < 0$).

Rule no. 7: Determine the intersections with the imaginary axis (if any) using the auxiliary equation of the Routh-Hurwitz criterion.

Let's start by composing the Routh-Hurwitz diagram. The starting point is the overall transfer function:

$$T(s) = \frac{K}{s^2 + as + K}$$

The diagram is easily determined to be:

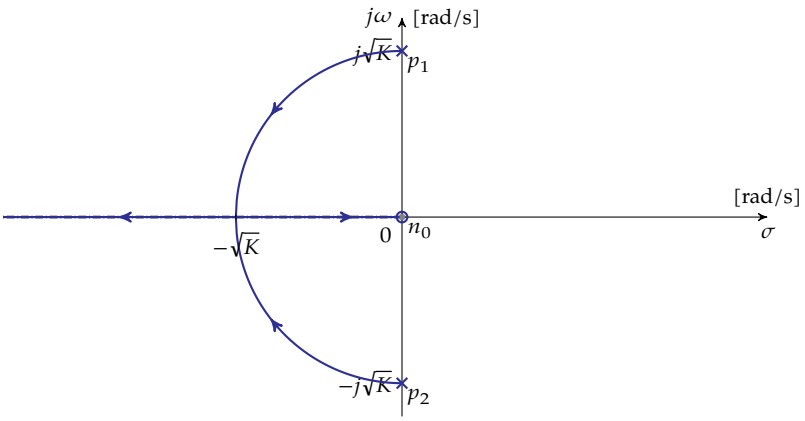
$$\begin{array}{c|cc} s^2 & 1 & K \\ s^1 & a & \\ s^0 & K & \end{array}$$

As long as $a > 0$ and $K > 0$, there are no sign changes in the pivot column and therefore no roots in the right-half plane. If $a = 0$, the middle element becomes zero, and therefore the row above yields us the corresponding auxiliary equation, i.e.

$$s^2 + K = 0$$

Again yielding us the poles of $F(s)$ that are on the imaginary axis $s = \pm j\sqrt{K}$.

The final result is the following root locus plot. How did we know the circular shape of the plot? Actually, we don't. We would have to calculate this in detail. The particular shape is also not that important if we're only interested in stability. We can see that in this case we have a stable system for $K > 0$.



Can you determine what happens if $K \leq 0$?

Before tackling a second example, note that most modern mathematics packages offer a facility for calculating and drawing root locus plots. So does MATLAB:

```
% root locus as a function of K
a = 1;
h1 = tf( [1], [1 a 0] );
rlocus( h1 );

% root locus as a function of a
K = 1;
h2 = tf( [ 1 0 ], [1 0 K ] );
rlocus( h2 );
```

Example 2
Consider a system with

$$F(s) = \frac{1}{s(s^2 + 2s + 4)}$$

This function has three poles at $p_1 = 0$ and $p_{2,3} = -1 \pm j\sqrt{3}$ and no zeros.

Rules no. 1 to 3: We therefore have 3 loci, all wandering of to infinity. The full negative real axis is on the left of an odd number of poles, so it will be part of the graph.

Rule no. 4: The asymptotes intersect in $s = \sigma_A$ and have direction angles equaling $\phi_{A,m}$:

$$\sigma_A = \frac{\sum p_i - \sum z_i}{n_p - n_z} = \frac{0 + (-1 + j\sqrt{3}) + (-1 - j\sqrt{3}) - 0}{3 - 0} = -\frac{2}{3}$$
$$\phi_{A,m} = \frac{180^\circ + m \cdot 360^\circ}{n_p - n_z} = 60^\circ + m \cdot 120^\circ$$

Rule no. 5: For symmetry reasons, the tangent line of p_1 must be horizontal. The direction ϕ_{p_2} must be -30°, as this fulfills the phase conditions:

$$\angle F(p_2) = -\angle(p_2 - p_1) - \phi_{p_2} - \angle(p_2 - p_3) = -180^\circ$$
$$-(180^\circ - \operatorname{atan}\frac{\sqrt{3}}{1}) - \phi_{p_2} - 90^\circ = -180^\circ$$
$$-120^\circ - \phi_{p_2} - 90^\circ = -180^\circ$$

For symmetry reasons the direction $\phi_{p_3} = 30^\circ$.

Rule no. 6: Calculate the break-away or merge-in points by solving $a'(s) = 0$.

Clearly, there are no parts on the real axis with multiple poles.

Rule no. 7: Determine the intersections with the imaginary axis (if any) using the auxiliary equation of the Routh-Hurwitz criterion.

Clearly $F(s)$ may stem from a feedback system like

$$T(s) = \frac{G(s)}{1 + KF(s)} = \frac{G(s) \cdot s(s^2 + 2s + 4)}{s(s^2 + 2s + 4) + K} = \frac{G(s) \cdot s(s^2 + 2s + 4)}{s^3 + 2s^2 + 4s + K}$$

This allows composing the Routh-Hurwitz diagram:

$$\begin{array}{c|cc} s^3 & 1 & 4 \\ s^2 & 2 & K \\ s^1 & \frac{8-K}{2} & \\ s^0 & K & \end{array}$$

Zeros in the pivot column occur for $K = 0$ and $K = 8$. The former corresponds to the start of the locus in the origin. The crossings with the imaginary axis of the latter can be found solving the auxiliary equation:

$$2s^2 + 8 = 0$$

resulting in $s_{1,2} = \pm j2$.

This allows for generating the following root-locus plot of Figure 4.3

Example 3

Consider a system with

$$F(s) = \frac{s+1}{s(s+2)(s+4)^2}$$

This function has 4 poles: $p_1 = 0$, $p_2 = -2$ and $p_{3,4} = -4$. There's also a single zero at $z_1 = -1$.

Rules no. 1 to 3: We therefore have 4 loci, of which 3 are wandering to infinity. The real axis for $s \leq -2$ and $-1 \leq s \leq 0$ will be part of the graph.

Rule no. 4: The asymptotes intersect in $s = \sigma_A$ and have direction angles equaling $\phi_{A,m}$:

$$\sigma_A = \frac{\sum p_i - \sum z_i}{n_p - n_z} = \frac{0 + (-2) + (-4) + (-4) - (-1)}{4 - 1} = -\frac{9}{3} = -3$$
$$\phi_{A,m} = \frac{180^\circ + m \cdot 360^\circ}{n_p - n_z} = 60^\circ + m \cdot 120^\circ$$

Rule no. 5: Because of symmetry, all tangent lines must be horizontal.

Rule no. 6: Calculate the break-away or merge-in points by solving $K'(s) = 0$.

First, let's calculate $K'(s)$:

$$1 + K \cdot \frac{s+1}{s(s+2)(s+4)^2} = 0$$
$$K = -\frac{s(s+2)(s+4)^2}{s+1}$$
$$= -\frac{s^4 + 10s^3 + 32s^2 + 32s}{s+1}$$
$$K' = -\frac{(4s^3 + 30s^2 + 64s + 32)(s+1) - (s^4 + 10s^3 + 32s^2 + 32s)}{(s+1)^2}$$
$$= -\frac{(4s^4 + 34s^3 + 94s^2 + 96s + 32) - (s^4 + 10s^3 + 32s^2 + 32s)}{(s+1)^2}$$
$$= -\frac{3s^4 + 24s^3 + 62s^2 + 64s + 32}{(s+1)^2}$$

Using a numerical solver on the numerator of this equation, yields two real roots:

$$K' = -3s^4 + 24s^3 + 62s^2 + 64s + 32 = 0$$

$$s = \begin{cases} -4 \\ -2.5994 \end{cases}$$

The first one was to be expected and occurs near the dual pole of $F(s)$ at $s = -4$. The second one is the true break-away location.

Rule no. 7: Determine the intersections with the imaginary axis (if any) using the auxiliary equation of the Routh-Hurwitz criterion.

Clearly $F(s)$ may stem from a feedback system like

$$T(s) = \frac{G(s)}{1 + KF(s)} = \frac{G(s) \cdot s(s+2)(s+4)^2}{s(s+2)(s+4)^2 + K(s+1)} = \frac{G(s) \cdot s(s+2)(s+4)^2}{s^4 + 10s^3 + 32s^2 + (32+K)s + K}$$

This allows composing the Routh-Hurwitz diagram:

s^4	1	32	K
s^3	10	$32 + K$	
s^2	b_1	b_2	
s^1	b_3		
s^0	b_4		

with

$$b_1 = \frac{10 \cdot 32 - 1 \cdot (32 + K)}{10} = \frac{288 - K}{10}$$

$$b_2 = K$$

$$b_3 = \frac{b_1 \cdot (32 + K) - 10 \cdot b_2}{b_1} = \frac{\frac{288-K}{10} \cdot (32 + K) - 10K}{\frac{288-K}{10}}$$

$$b_4 = b_2 = K$$

Zeros in the pivot column occur for $K = 288$ on the third row and for $b_3 = 0$ on the fourth row, i.e.

$$(288 - K)(32 + K) - 100K = 0$$

$$-K^2 + 156K + 9216 = 0$$

$$K_{1,2} = \begin{cases} -45.693 \\ 201.693 \end{cases}$$

Because we only consider positive values of K when investigating a root-locus plot, we can discard the negative value. The crossing occurs for $K = 201.693$. Finally another crossing occurs for $K = 0$ on the fifth row.

The crossing for $K = 0$ obviously corresponds to the starting position of the root locus. The next crossing occurs at the smallest value of K, i.e. $K = 201.693$. The auxiliary equation, therefore becomes:

$$\frac{288 - K}{10}s^2 + K = 0$$

$$\downarrow \quad K = 201.693$$

$$8.6307s^2 + 201.693 = 0$$

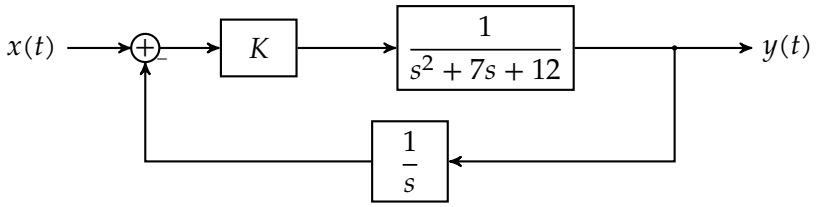
$$s = \pm j\sqrt{23.369}$$

resulting in $s_{1,2} = \pm j4.8342$.

This allows for generating the following root-locus plot of Figure 4.4.

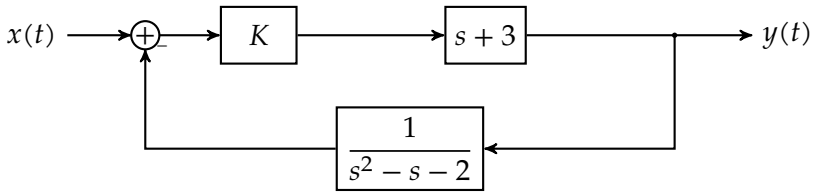
Exercises

Exercise 4.2.3-1: Determine the root locus plot of the following system w.r.t. K:



Verify the magnitude and phase and phase condition for the crossing points with the imaginary axis.

Exercise 4.2.3-2: Determine the root locus plot of the following system w.r.t. K:



Exercise 4.2.3-3: Determine the root locus plot of a system with

$$F(s) = \frac{s+1}{s^3 + 5s^2 + 8s + 6}$$

Exercise 4.2.3-4: Determine the root locus plot of a system with

$$F(s) = \frac{s}{s^4 + 8s^3 + 31s^2 + 42s + 18}$$

Exercise 4.2.3-5: (*) Determine the root locus plot of a system with

$$F(s) = \frac{1}{s^4 + 16s^3 + 96s^2 + 256s + 255}$$

Exercise 4.2.3-6: (*) Determine the root locus plot of a system with

$$F(s) = \frac{s+1}{(s+2)(s+3)(s+5)(s^2 + 8s + 17)}$$

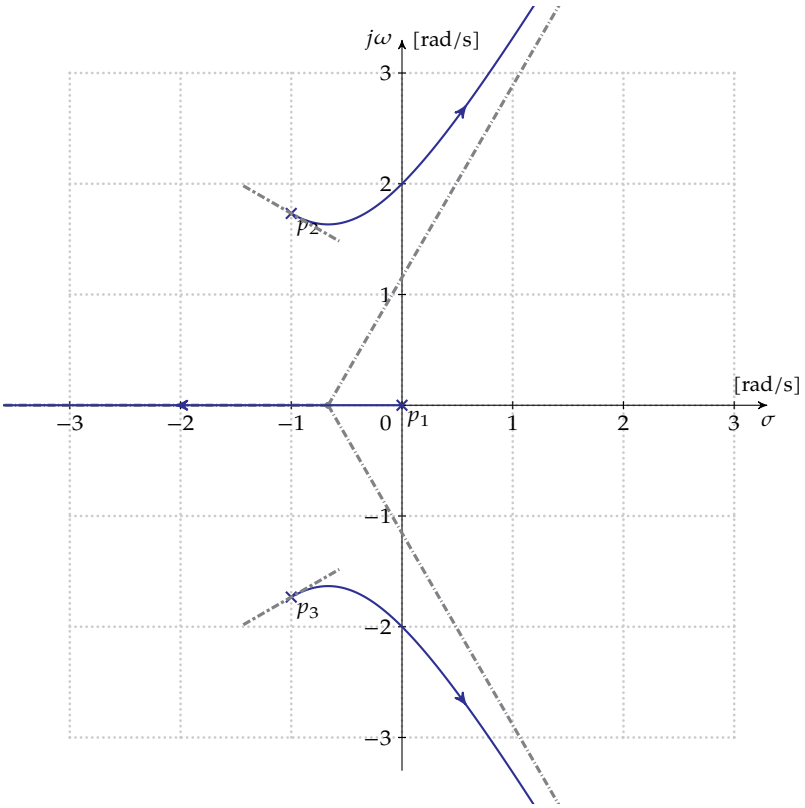


Figure 4.3: Root locus plot of example 2

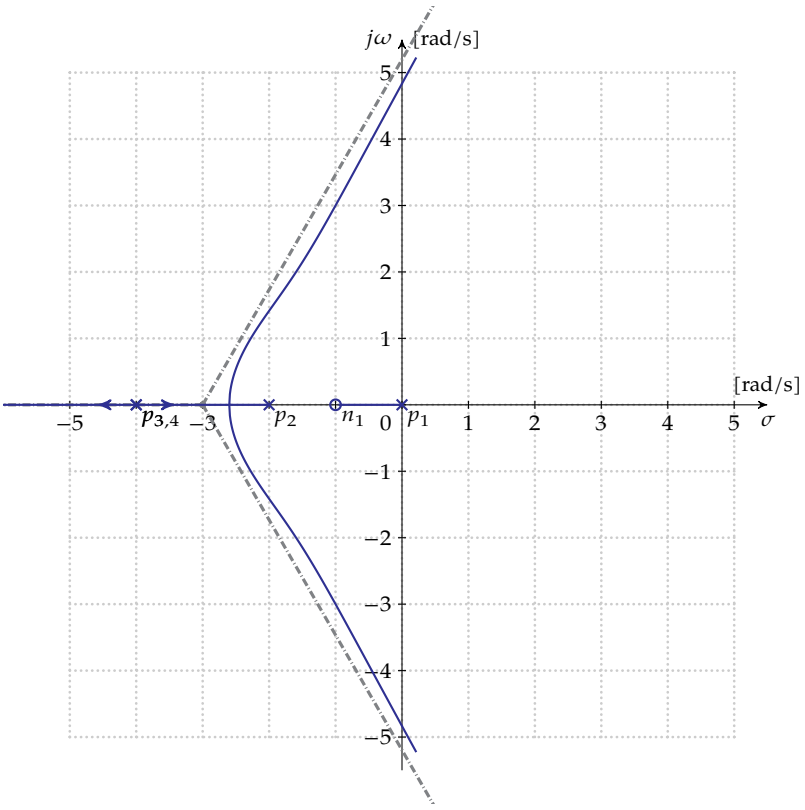


Figure 4.4: Root locus plot of example 3

H(s)	$T_{s,0.02}$
$\frac{1}{1+\tau s}$	$\frac{4}{1/\tau}$
$\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$	$\frac{4}{\zeta\omega_n}$

Table 4.1: Settling time for first and second order systems

4.2.4 Using root-locus insights to meet system specifications

In order to have insight in root-locus plots, we need to know how they are related to the system's specifications. We will treat this first. Afterwards, we will give a few examples.

Settling time, overshoot and steady-state error

The root locus plot is in direct relation to the settling time, the overshoot and the steady-state error.

Settling time Remember that we derived equations in section 2.1.1 on page 23 for the settling time as a function of the location of the system poles. You can review these equations in Table 4.1. What is noteworthy is that the denominator of these settling time equations are in fact the real pole of the first-order system and the real part of the poles of the second-order system.

In view of this, given a desired settling time T_s we can subdivide the Argand plane in two regions, separated by a specification line given by $\sigma = -4/T_s$. This has been indicated in Figure 4.5a. We have faster settling to the left of the line and slower settling to the right of the line.

And though this observation only holds for first and second-order systems, in many cases, it will be a good first estimate.

Overshoot In section 2.1.2 on page 26 we derived an equation for the overshoot of a second-order system (without zero and with $\zeta < 1$):

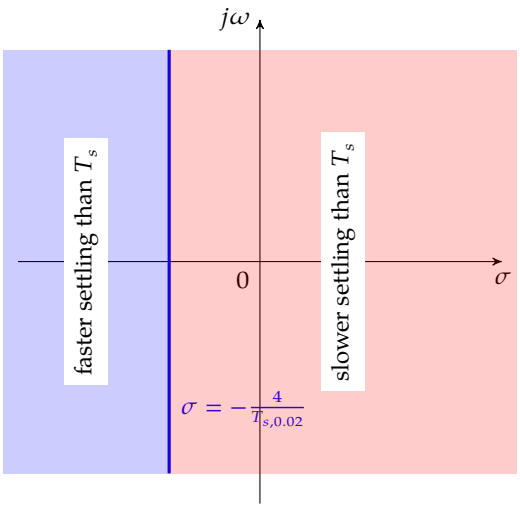
$$O = \mathrm{e}^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}$$

with ζ a measure for the angle θ the poles make with the real axis according to $\theta = \arccos\zeta$.

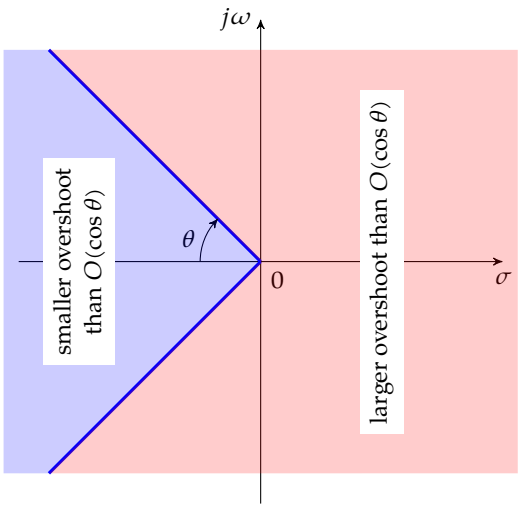
Therefore, the locus of points with constant overshoot are points with a constant ζ and thus a constant θ.

This has been indicated in Figure 4.5b. The region to the left of the blue lines is the region of smaller overshoot (larger ζ), the region to the right of larger overshoot.

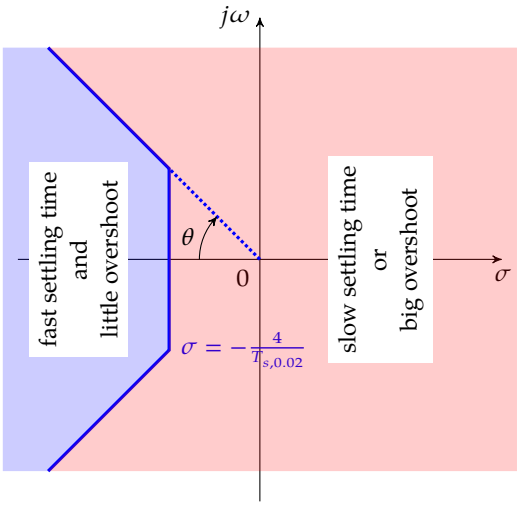
Note that the value of θ can be easily determined by substituting $\zeta = \cos(\theta)$ in (2.4) and



(a) Settling time and the Laplace domain



(b) Overshoot and the Laplace domain



(c) Settling and overshoot combined

Figure 4.5: Settling time and overshoot in relation to the root locus to illustrate the sweet region of the root locus in the Laplace domain

solving for θ. This leads to:

$$\theta = \arctan\left(\frac{\pi}{\ln\left(\frac{1}{O}\right)}\right)$$

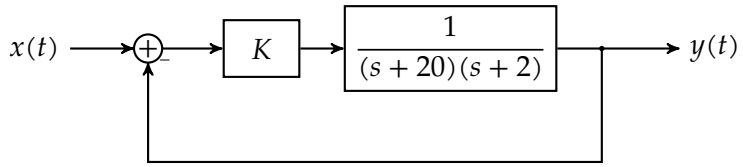
If you combine settling time and overshoot specs, you get the *sweet region* of the Laplace domain w.r.t. root locus plots.

Steady-state error In section 2.3 on page 32 we have seen that the steady-state error of feedback systems that do not have enough integrators to make the error zero, depends on the low-frequency gain of the loop (without integrators).

Therefore cranking up the gain in our loop is beneficial. Therefore, we will select an operation point on a root locus as far downstream as can be, without violating our requirements for stability, settling time and overshoot. This will become clear in the examples below.

Examples

Example 1 Consider the following system:



Let's try to design this system to the following specifications:

$$T_s \leq 0.5\,\text{s} \qquad O \leq 20\% \qquad e_{ss,u(t)} \leq 10\%$$

The key question is: which value of K realizes these specifications. The root locus plot of this example can be found in Figure 4.6. The specification lines for the settling time and the overshoot have been indicated. They are easily determined to be:

- Settling time: $\sigma = -\dfrac{4}{T_s} = -8\,\text{rad/s}$
- Overshoot: $\theta = \arctan\left(\dfrac{\pi}{\ln\left(\frac{1}{O}\right)}\right) = 62.873°$

To comply with the specifications, the roots must be located in the shaded area. The right pole enters this area for $\sigma = -8$. We can determine the corresponding value of K by solving the magnitude/modulus condition:

$$\begin{aligned} K \cdot |F(s)| &= 1 \\ K \cdot \frac{1}{|s+20| \cdot |s+2|} &= 1 \\ &\downarrow \quad s = -8 \\ K \cdot \frac{1}{|12| \cdot |-6|} &= 1 \\ K &= 12 \cdot 6 = 72 \end{aligned}$$

So, to comply with the settling time specification we must ensure $K \geq 72$.

The root locus exits the shaded area when it crosses the overshoot line. The intersection $s_i = -11 + j\omega_i$ can be easily determined applying simple trigonometry as:

$$\begin{aligned} \tan\theta &= \frac{\omega_c}{11} \\ \omega_c &= 11\tan(62.873°) = 21.472 \end{aligned}$$

Again, we can determine the corresponding value of K by solving the magnitude/modulus condition:

$$\begin{aligned} K \cdot |F(s)| &= 1 \\ K \cdot \frac{1}{|s+20| \cdot |s+2|} &= 1 \\ &\downarrow \quad s = -11 + j21.472 \\ K \cdot \frac{1}{\sqrt{9^2+21.472^2} \cdot \sqrt{9^2+21.472^2}} &= 1 \\ K &= 9^2 + 21.472^2 = 542.05 \end{aligned}$$

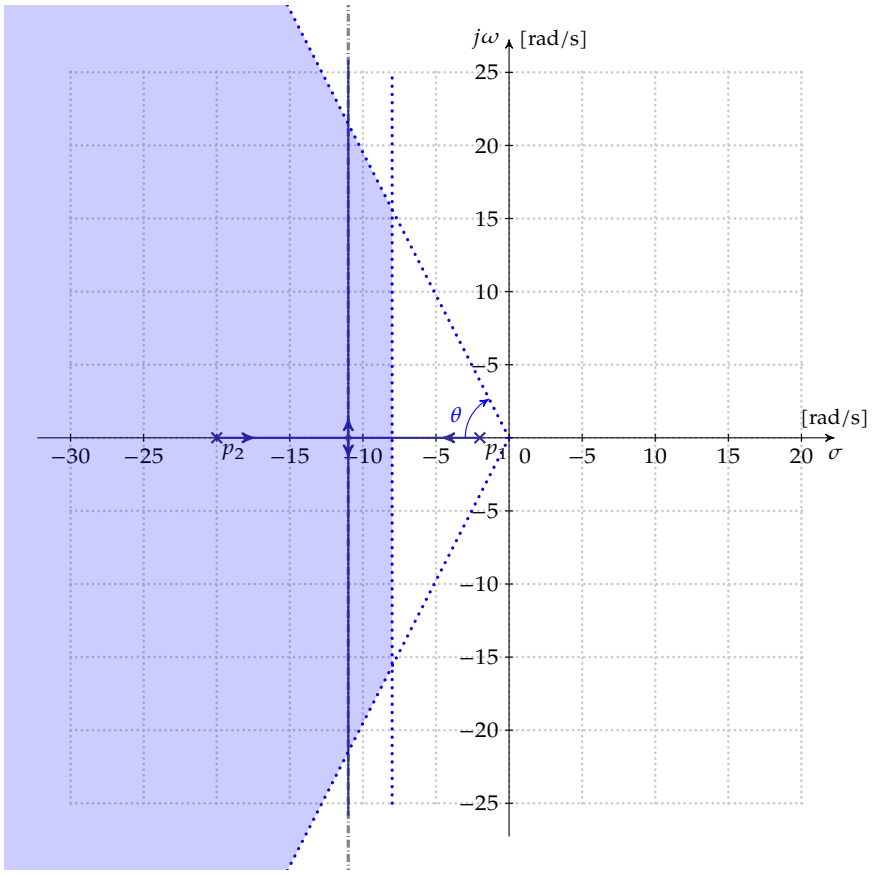


Figure 4.6: Root locus plot of the first example ($F(s) = \frac{1}{(s+20)(s+2)}$) with the settling time and overshoot specifications superimposed

Conclusion to comply with the overshoot specification, we must ensure $K \leq 542.05$.

And now for the steady-state error. The specification is given as a percentage, i.e. we need to investigate the situation for a step input step $(A \cdot u(t)$ and need to ensure that $\frac{e_{ss}}{A} \leq 0.1$.

We know that for our type 0 system

$$e_{ss} = \frac{A}{1 + K'_{DC}}$$
$$\frac{e_{ss}}{A} = \frac{1}{1 + K'_{DC}} \leq 0.1$$

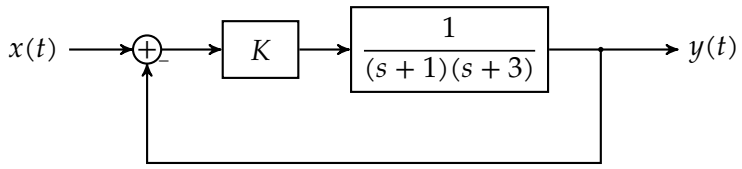
If we need to ensure that $K'_{DC} \geq 9$. Knowing that K'_{DC} is the loop gain without integrators, it is easy to see that:

$$K'_{DC} = \left| \frac{K}{(s+20)(s+2)} \right|$$
$$\downarrow \quad s = 0$$
$$= \frac{K}{20 \cdot 2} = K/40$$

This means that we need to ensure $K \geq 360$ to comply with the steady-state specification.

Conclusion: the values $360 \leq K \leq 542.05$ realize the specifications.

Example 2 In this example, we will show how adding a zero can improve a system's performance.
Consider the following system:



It looks very similar to the previous example. This time, our job is to find decent values of the gain K to ensure:

$$T_s \leq 1\,\text{s} \qquad O \leq 4.3\% \qquad e_{ss,u(t)} \leq 5\%$$

However, if we draw the root-locus plot and superimpose the specs on it (see Figure 4.7), the problem stares us right in the face: there is no spot on the root locus that will comply with the settling time specification.

There's clearly a range of values of K that fulfill the overshoot specification. Knowing that $O = 4.3\%$ corresponds to $\theta \approx 45°$ (do the math yourself), it is easy to find that the crossing of the root locus with the overshoot specification line occurs at $s = -2 + j2$. The value of K that can be derived from the magnitude/modulus condition and turns out to be $K = 5$, i.e. to comply with the overshoot specification $K \leq 5$.

However this value doesn't allow meeting the steady-state error specification, as we know that $K'_{DC} = K/3 = 1.666$ which is insufficient to obtain the required 5% which corresponds with $K'_{DC} \geq 19$.

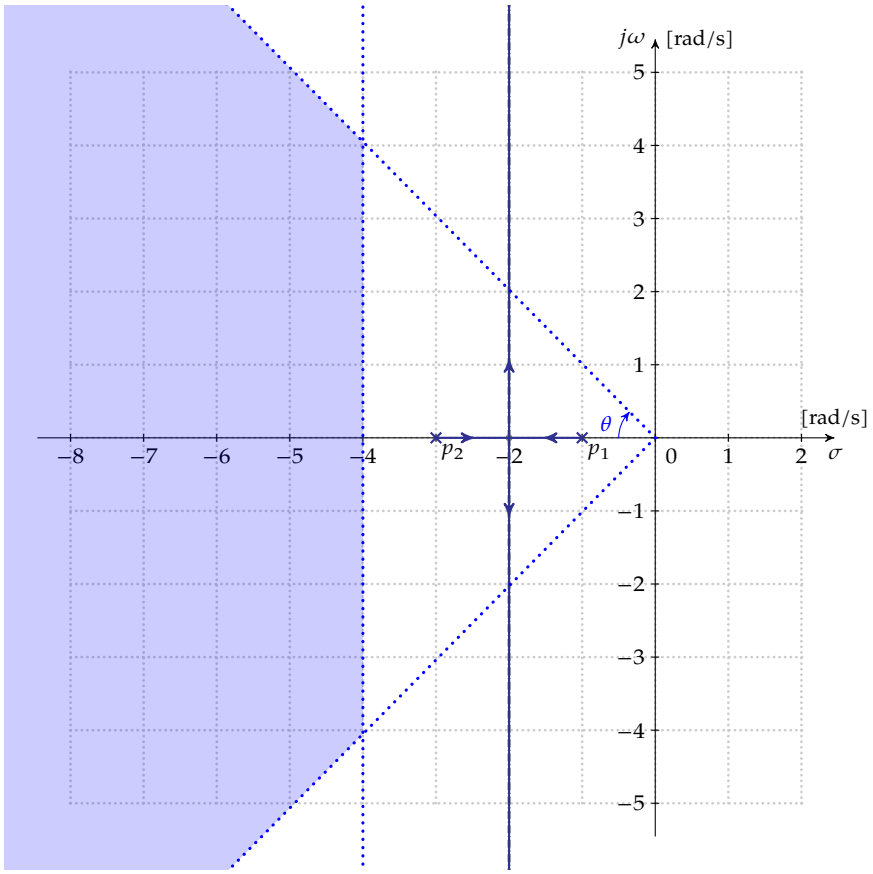


Figure 4.7: Root locus plot of the second example ($F(s) = \frac{1}{(s+1)(s+3)}$) with the settling time and overshoot specifications superimposed

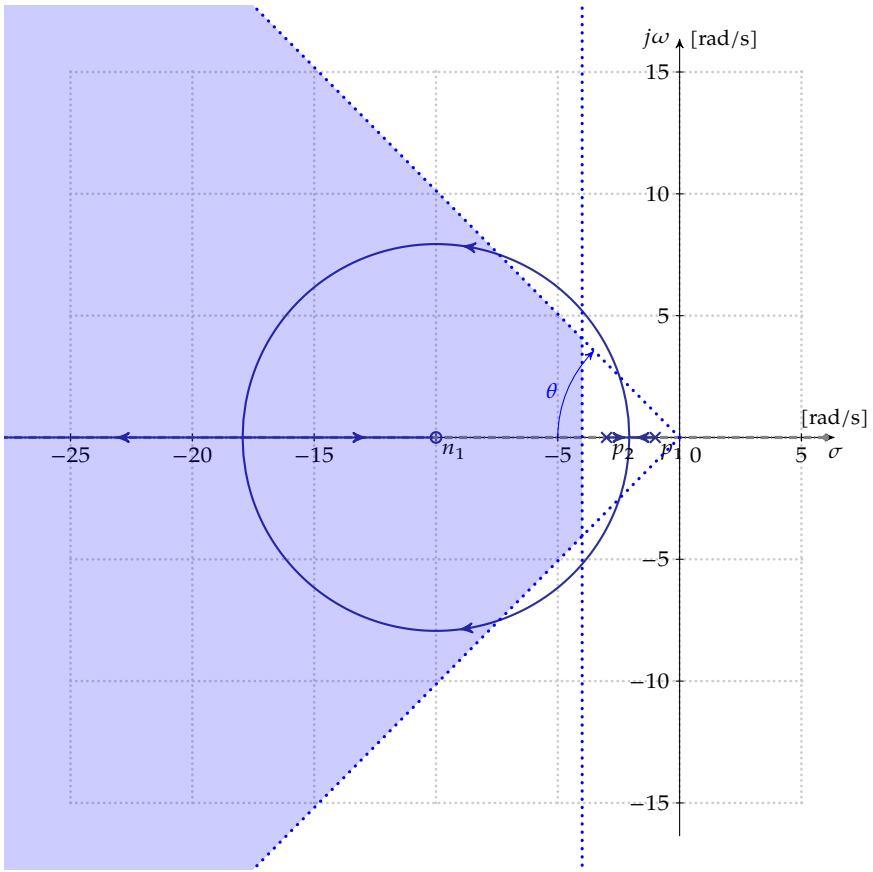
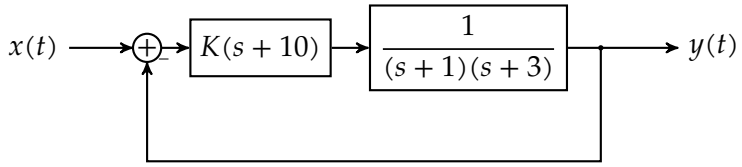


Figure 4.8: Root locus plot of the second example ($F(s) = \frac{s+10}{(s+1)(s+3)}$) with the settling time and overshoot specifications superimposed

Conclusion: no solution.

However, if we could bend the north and south going loci to the left, we might be able to obtain useful results. Adding a zero solves the problem, can you imagine where it should be added?

Let's add a zero for $s = -10$, i.e.



The corresponding root-locus plot can be found in Figure 4.8.

Finding out where the locus intersects with the overshoot specification line can be easily done by checking the phase condition:

$$\angle F(s) = \angle(s+10) - \angle(s+1) - \angle(s+3) = 180^\circ + m \cdot 360^\circ$$

$$\downarrow \quad s = -a + ja$$

$$\arctan\left(\frac{a}{10-a}\right) - \arctan\left(\frac{a}{1-a}\right) - \arctan\left(\frac{a}{3-a}\right) = 180^\circ + m \cdot 360^\circ$$

Taking the tangent of both sides of the equation, knowing that

$$\tan(\alpha+\beta+\gamma) = \frac{\tan\alpha+\tan\beta+\tan\gamma-\tan\alpha\tan\beta\tan\gamma}{1-(\tan\alpha\tan\beta+\tan\alpha\tan\gamma+\tan\beta\tan\gamma)}$$

we obtain:

$$\frac{\frac{a}{10-a}-\frac{a}{1-a}-\frac{a}{3-a}-\frac{a^3}{(10-a)(1-a)(3-a)}}{1+\frac{a^2}{(10-a)(1-a)}+\frac{a^2}{(10-a)(3-a)}-\frac{a^2}{(1-a)(3-a)}} = 0$$

$$\frac{\frac{a(1-a)(3-a)-a(10-a)(3-a)-a(10-a)(1-a)-a^3}{(10-a)(1-a)(3-a)}}{\frac{(10-a)(1-a)(3-a)+a^2(3-a)+a^2(1-a)-a^2(10-a)}{(10-a)(1-a)(3-a)}} = 0$$

$$\frac{a(1-a)(3-a)-a(10-a)(3-a)-a(10-a)(1-a)-a^3}{(10-a)(1-a)(3-a)+a^2(3-a)+a^2(1-a)-a^2(10-a)} = 0$$

$$\frac{N}{D} = 0$$

The denominator doesn't matter as the expression only can become zero for finite a if the numerator zero. Therefore, we continue with the numerator and set it to zero:

$$a(a^2-4a+3-a^2+13a-30-a^2+11a-10-a^2) = 0$$

$$a(-2a^2+20a-37) = 0$$

This expression becomes zero for $a = 0$ (obviously) and for

$$a_{1,2} = \begin{cases} 2.4505 \\ 7.5495 \end{cases}$$

which seems to be in accordance with Figure 4.8.

We now can calculate which value of K corresponds to $s = -7.5495 + j7.5495$ by writing down the modulus condition:

$$K \cdot |F(s)| = 1$$

$$K \cdot \frac{\overbrace{|s+10|}^{a}}{\underbrace{|s+1|}_{b} \cdot \underbrace{|s+3|}_{d}} = 1$$

With $s = -7.5495 + j7.5495$, we get:

$$a = \sqrt{2.4505^2+7.5495^2} = 7.9372$$

$$b = \sqrt{6.5495^2+7.5495^2} = 9.9945$$

$$c = \sqrt{4.5495^2+7.5495^2} = 8.8144$$

Therefore, we need to ensure: $K \geq 11.1$.

In order to comply with the steady-state error specification of 5%, we need at least $K'_{DC} = 19$ (as argued earlier). We know that

$$K'_{DC} = K\frac{|s+10|}{|s+1|\cdot|s+3|}\bigg|_{s=0}$$

$$= \frac{10}{3}K$$

We therefore at least need $K \geq 5.7$.

Let's consider $K = 20$. This allows for calculating the overall transfer function:

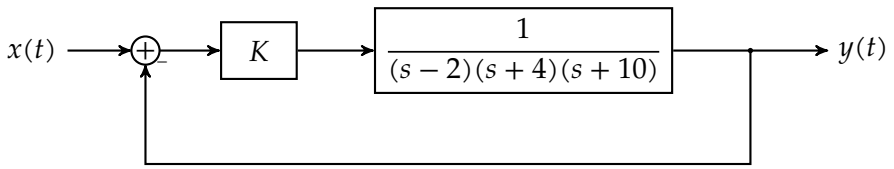
$$\begin{aligned} T(s) &= \frac{\frac{20(s+10)}{(s+1)(s+3)}}{1 + \frac{20(s+1)}{(s+1)(s+3)}} \\ &= \frac{20(s+10)}{s^2 + 4s + 3 + 20s + 200} \\ &= \frac{20(s+10)}{s^2 + 24s + 203} \end{aligned}$$

The poles of $T(s)$ are at $-12 \pm j7.6811$, which corresponds to a $\zeta = 12/\sqrt{12^2 + 7.6811^2} \approx 0.84$.

Given the fact that we are now dealing with a pure second-order system, but with a second-order system with a zero, we can

- use the original settling equation as a first estimate, however we should verify by simulation:
$$T_{s,2\%} = \frac{4}{12} = 0.333\,\text{s}$$
- use Figure 2.5 for overshoot assessment; this requires calculating
$$\alpha = \frac{z_1}{\zeta\omega_n} = \frac{10}{12} = 0.833$$
This still will lead to an overshoot of approximately 10%. It won't be easy to improve this significantly.

Example 3 In this example we will show how feedback can make an unstable system stable. Consider the following system:



Note how the system without feedback would have a pole in the right-half plane ($p_1 = 2$) and thus be unstable. Let's draw the root-locus (do the math yourself!). You can find the result in Figure 4.9.

In order to calculate the crossing points of the north-south loci with the imaginary axis, we need to use the Routh-Hurwitz criterion with:

$$\begin{aligned} T(s) &= \frac{\frac{K}{(s-2)(s+4)(s+10)}}{1 + \frac{K}{(s-2)(s+4)(s+10)}} \\ &= \frac{K}{s^3 + 12s^2 + 12s + K - 80} \end{aligned}$$

This leads to the following diagram:

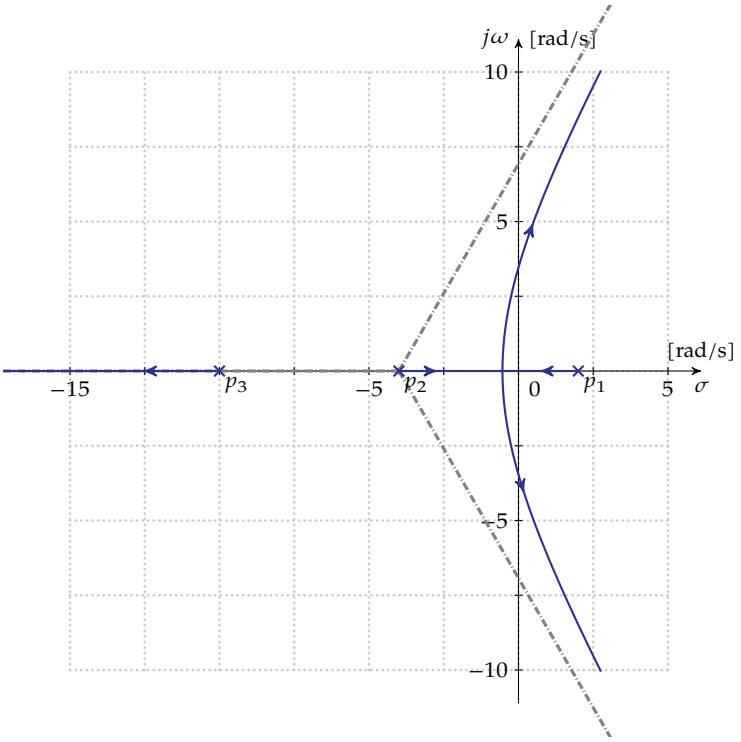


Figure 4.9: Root locus of the third example $F(s) = frac1(s-2)(s+4)(s+10)$

s^3	1	12
s^2	12	$K-80$
s^1	$\frac{224-K}{12}$	
s^0	$K-80$	

The locus of the unstable pole will go to the left-half plane of $K \geq 80$, and the north-south loci will cross the imaginary axis for $K = 224$. The auxiliary equation leading to the crossing points therefore becomes:

$$12s^2 + 144 = 0$$
$$s = \pm j\sqrt{12}$$

Note that in this case, when the amplifier (implementing K) ever dies, this system will suffer some serious damage.

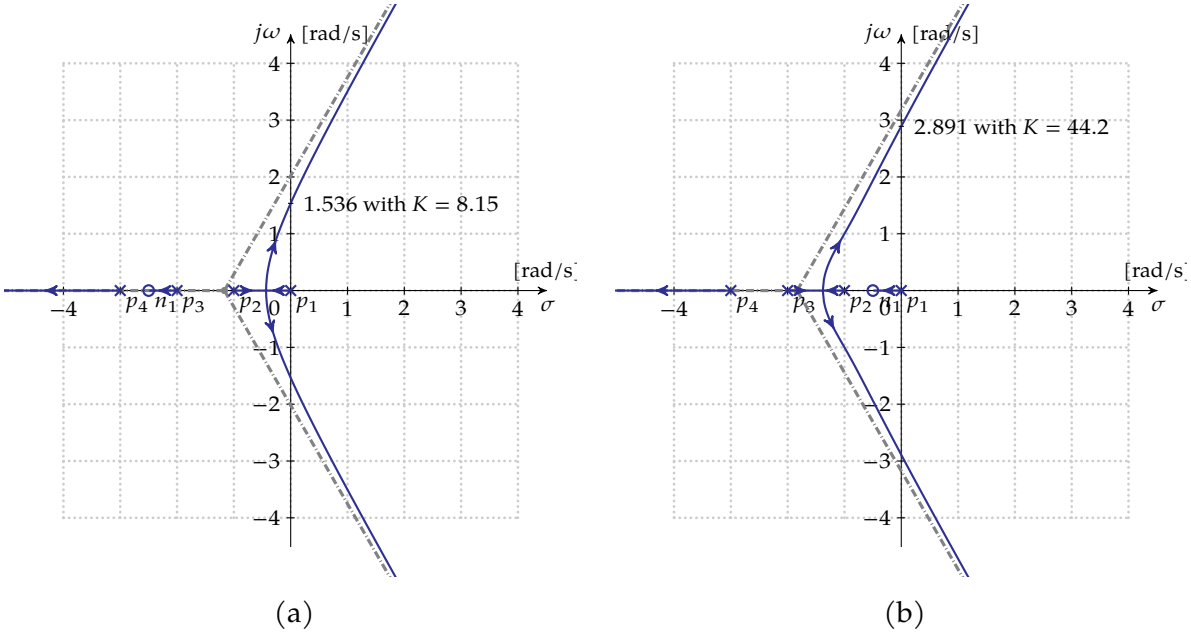
Example 4 In this example we will show once more the influence of the position of zeros on the system characteristics. The zeros of $F(s)$ are an end point point of a locus, i.e. a locus ends there for $K \to +\infty$. By placing a zero carefully, we can influence the arrangement of the root locus plot and improve the system's behavior.

Let's illustrate this with an example. Consider:

$$F(s) = \frac{s+a}{s(s+1)(s+2)(s+3)} \tag{4.8}$$

which has a zero that we can set anywhere on the real axis by selecting an appropriate value for a. We will consider two particular values $a = 2.5$ and $a = 0.5$.

We won't go through the step-by-step process of the root locus procedure. You can do that yourself (and please, do so!). We will just focus on the result for the two cases, which can be found in Figure 4.10.



(a) (b)

Figure 4.10: Root locus plots for a system with the loop gain of (4.8) for (a) $a = 2.5$ and (b) $a = 0.5$

In the case $a = 2.5$, the loci wandering off to infinity are starting from the outer poles (from p_4 and p_1 together with p_2). When you set $a = 0.5$ you place the zero such that you create a finite locus opportunity for p_1, such that the loci to infinity start from in between p_2 and p_3. This allows for a higher low-frequency loop gain and loci that are moving to infinity through higher frequencies (further away from $s = 0$). The former is good for the steady-state error and the latter is good for speed.

a	0.5	2.5
K	8.15	44.2
ω_{osc}	1.536	2.891

Another way to understand why the loci are higher in frequency, is by observing that in the equation for σ_A, more positive zeros move the locus centroid to the left.

What happens if we set the zero in the right-half plane? Would that be a good choice? Obviously no, this will cause a locus in the unstable right-half plane for all nonzero values of K.

Chapter 5

Stability in the Fourier Domain

The Fourier domain is an attractive one, as system identification in this domain is easy in the case of linear systems. Indeed, consider the following LTI system:

$$x(t) \longrightarrow \boxed{G(j\omega)} \longrightarrow y(t)$$

As sine waves are eigenfunctions of LTI systems, determining $G(j\omega)$ can be done by applying a sine wave as input

$$x(t) = \sin(\omega t)$$

and measuring the amplitude and phase delay of the output

$$y(t) = A(\omega) \cdot \sin(\omega t + \phi(\omega)).$$

The frequency transfer function $G(j\omega)$ is then easily determined as:

$$G(j\omega) = A(\omega)\, e^{j\phi(\omega)}$$

In this chapter, we will study stability by analyzing $G(j\omega)$.

We will see how different plot techniques allow assessing stability in the Fourier domain. Our journey will be one of hunting for poles in the right-half plane — with Cauchy — on a polar plot, leading to the Nyquist criterion. We will see main concepts as gain and phase margin, of which only the latter will prove to be deterministic for stability. We will translate our polar plot insights into the well known Bode plots and a mix of both, the Nichols plot. We will also debunk a common misconception w.r.t. phase and gain margin and meet the pain in the neck of control theory: systems with a time delay.

After having read and studied the chapter, you are expected to be able to:

- draw a polar plot of $G(j\omega)$ and apply the Nyquist criterion with confidence,
- explain the relationship between stability and gain and phase margin,
- determine gain and phase margins on a polar plot, a Bode plot and a Nichols plot, and
- explain why time-delay systems are a pain in the neck.

5.1 The polar plot

5.1.1 Principle

Given the fact that $G(j\omega)$ is a complex function, we can write it as

$$G(j\omega) = R(\omega) + jX(\omega)$$

and draw this complex function in the complex plane. If we vary ω, we obtain a trajectory. This is called the *polar plot*.

Let's illustrate this with a few examples.

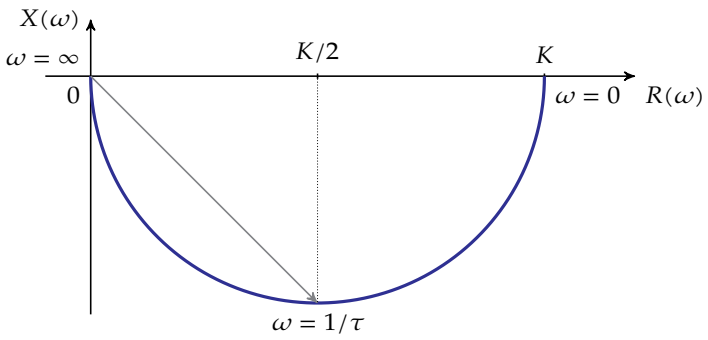
Example 1: Consider the first-order system

$$G(j\omega) = \frac{K}{1 + j\omega\tau}$$

It is easy to determine $R(\omega)$ and $X(\omega)$ as:

$$R(\omega) = \frac{K}{1 + \omega^2\tau^2} \qquad X(\omega) = -\frac{\omega\tau K}{1 + \omega^2\tau^2}$$

This leads to the following polar plot:



To illustrate, the point for $\omega = 1/\tau$ has been indicated by a gray vector. This illustrates that it is easy to read the magnitude and the phase for this kind of polar plot as well. In this case:

$$|G(j\omega)| = \sqrt{R^2(\omega) + X^2(\omega)} = \frac{K}{\sqrt{2}} \qquad \angle G(j\omega) = \arctan\left(\frac{X(\omega)}{R(\omega)}\right) = \arctan(-1) = -45^\circ$$

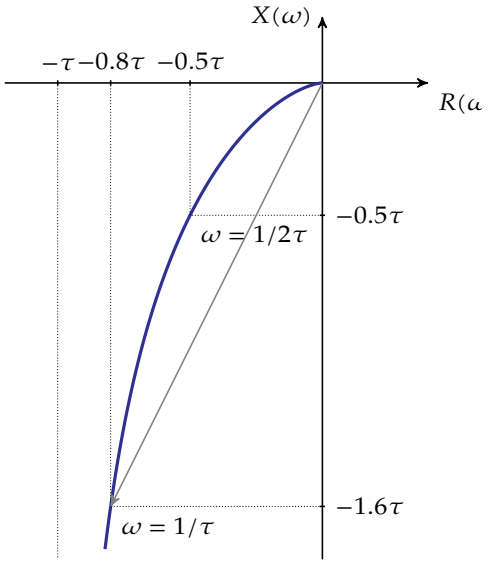
Example 2: Consider the second-order system

$$G(j\omega) = \frac{1}{j\omega(1 + j\omega\tau)}$$

Again, we can determine $R(\omega)$ and $X(\omega)$ to be:

$$R(\omega) = -\frac{\tau}{1+\omega^2\tau^2} \qquad X(\omega) = -\frac{1/\omega}{1+\omega^2\tau^2}$$

This leads to the following polar plot:



Once more, to illustrate, the point for $\omega = 1/2\tau$ has been indicated by a gray vector. The corresponding magnitude and phase is easy to read from the polar plot and can be calculated to be:

$$|G(j\omega)| = \sqrt{R^2(\omega) + X^2(\omega)} = \frac{4}{\sqrt{5}}\tau \qquad \angle G(j\omega) = -180^\circ + \arctan\left(\frac{X(\omega)}{R(\omega)}\right) = -180^\circ + \arctan(2) = -116.57^\circ$$

Admittedly, the polar plot has quite some disadvantages w.r.t. a regular Bode plot:

Plot property	Polar plot	Bode plot
Calculation & construction	difficult	easy
Changing K	requires scaling plot	shift the magnitude
Pole/zero location	unclear	easy using asymptotes
Adding pole/zero	restart from scratch	easy

Now, if it is so bad, why would we use this polar plot? Well, it is the basis for the one-and-only stability criterion in the frequency domain. Therefore, we need to investigate it. Later on, we will be able to translate that criterion to Bode plots. Now, this is important, as we have seen that it is rather easy to measure the frequency response of stable systems.

In the Laplace domain, the criterion for stability is very clear: no poles in the right-half plane. The question is: how does this criterion map to the frequency domain?

Well, we'll go hunting for poles in the right-half plane in the Laplace domain and see what that means for the Fourier domain.

The basis for this is threefold:

1. Contour mapping
2. Cauchy's theorem
3. The Nyquist criterion

The beauty of this all will be that our analysis of the stability of the feedback system will be based on the *loop gain*. Effects of increasing or reducing the loop gain are therefore easy to analyze. The Nyquist criterion (by Harry Nyquist) is a true piece of marvel.[1]

5.1.2 Contour mapping

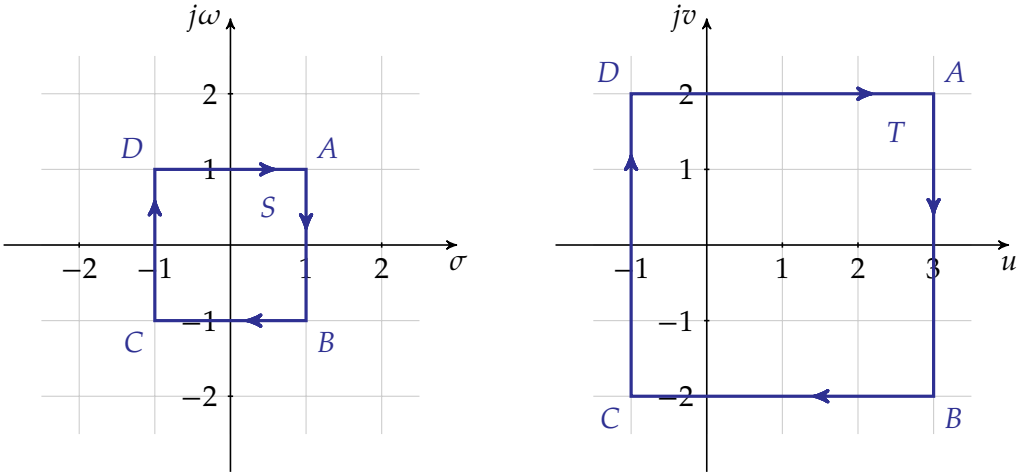
Consider a contour S in the complex plane. If we consider a complex function $F(s)$, We can map this trajectory to a new trajectory $T = F(S)$ in the complex plane. Often, we will call the domain of the original trajectory the s-plane and the domain of the target trajectory the $F(s)$-plane.

Let's illustrate this with an example. Consider $F(s) = 2s + 1$ and let's use this function to map the contour S drawn in Figure 5.1a in the s-plane to a contour T in the $F(s)$-plane.

If we decompose $s = \sigma + j\omega$ and $F(s) = u + jv$, then it is easy to see that

$$F(s) = 2s + 1 = 2(\sigma + j\omega) + 1 = \underbrace{2\sigma + 1}_{\equiv u} + j\underbrace{2\omega}_{\equiv v}$$

Using this equation we can map every point of S to a point of T. Try!



(a) Contour in the s-plane

(b) Mapped contour in the $F(s)$ plane

Figure 5.1: Mapping of a contour S to $T = F(S)$ with $F(s) = 2s + 1$

Another mapping of the same contour, but with $F(s) = \frac{s}{s+2}$, can be found in Figure 5.2.

In the frame of Cauchy's theorem, we agree to the following conventions:

[1] It must be mentioned that Felix Strecker was in fact two years earlier than Nyquist in discovering this criterion, however, history never has been fair in naming events and achievements correctly.

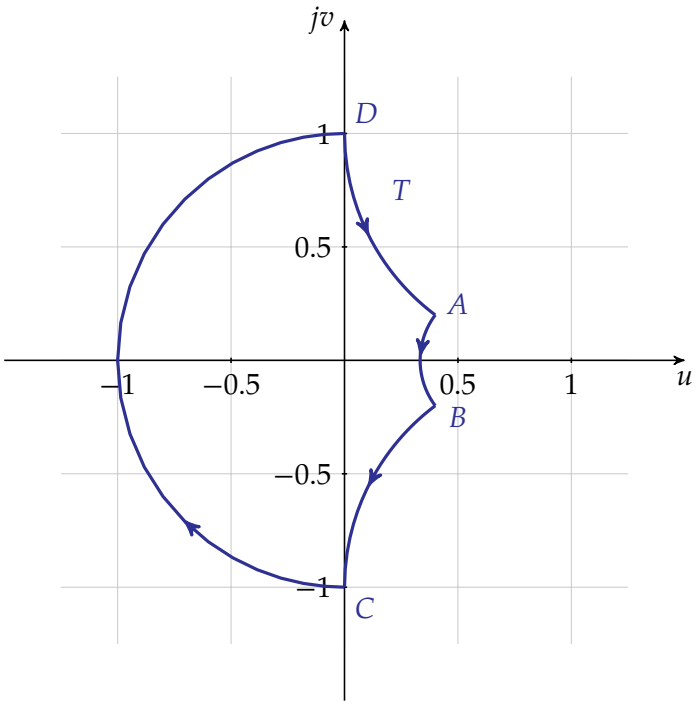


Figure 5.2: Mapped contour $T = F(S)$ with $F(s) = s/(s+2)$

- A contour can be traversed in two directions, we will call the clockwise (CW) direction the *positive direction*.
- A contour encloses a region, we will call the region to the right of the contour the *enclosed region*.

It must be said that this is a true American convention; normally we choose the counterclockwise (CCW) direction of the contour to be positive and the area to the left of the contour, the enclosed region.

5.1.3 Cauchy's theorem

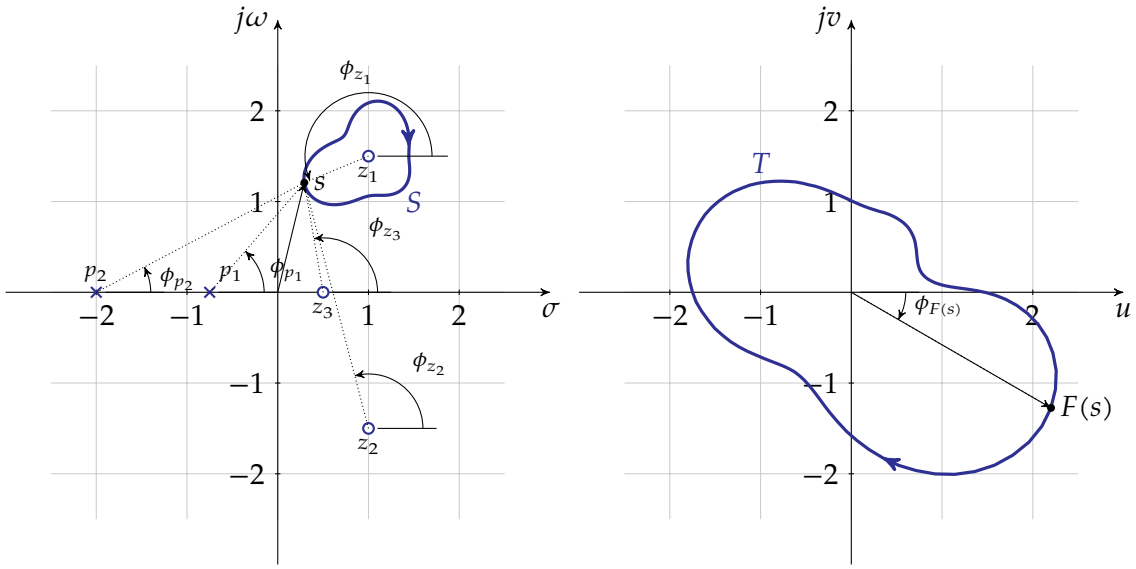
This theorem holds for $F(s)$ that are a ratio of two polynomials in s:

$$F(s) = \frac{\sum_{j=0}^{m} b_j s^j}{\sum_{i=0}^{n} a_i s^i}$$

or in factorized form:

$$F(s) = K \cdot \frac{\prod_{j=1}^{m} (s - z_j)}{\prod_{i=1}^{n} (s - p_i)}$$

If you draw the poles and zeros of $F(s)$ in the s-plane and choose a contour S that does not go through any of those poles and zeros, then the mapping $T = F(S)$ will be a finite contour in the $F(s)$-plane and will not go through the origin of the $F(s)$-plane.
Obviously, assuming for convenience reasons that $K > 0$, it is easy to calculate $F(s)$ for an



(a) Original contour S (b) Mapped contour $T = F(S)$

Figure 5.3: Example of the mapping of a contour by a function $F(s)$ that is a ratio of two polynomials in s

arbitrary s, using:

$$|F(s)| = K \cdot \frac{\prod_{j=1}^{m} |s - z_j|}{\prod_{i=1}^{n} |s - p_i|}$$

$$\underbrace{\angle F(s)}_{\phi_{F(s)}} = \sum_{j=1}^{m} \underbrace{\angle(s - z_j)}_{\phi_{z_j}} - \sum_{i=1}^{n} \underbrace{\angle(s - p_i)}_{\phi_{p_i}}$$

An example of this situation has been depicted in Figure 5.3.

Here's Cauchy's theorem:

Cauchy's theorem Assuming $F(s)$ to be a ratio of two polynomials in s, and selecting an original CW contour S in the Laplace domain that does not go through any of the poles or zeros of $F(s)$:

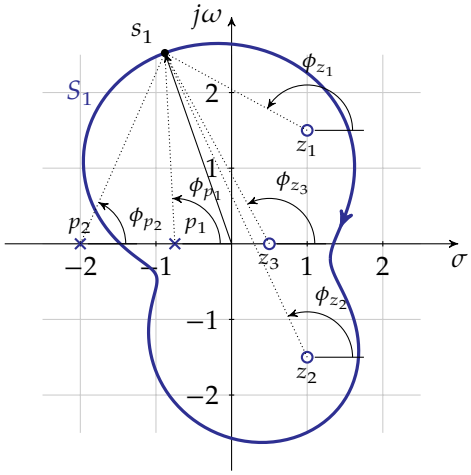
$$N = Z - P$$

with

1. N the number of CW revolutions of the mapped contour $F(S)$ around the origin,
2. Z the number of zeros of $F(s)$ inside the original contour S, and
3. P the number of poles of $F(s)$ inside the original contour S.

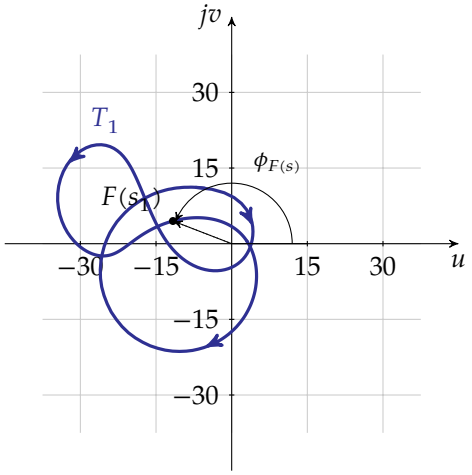
Indeed, looking at Figure 5.3, a single CW contour around a single pole (i.e. $Z = 1, P = 0$) results in a single CW contour around the origin for $F(s)$ as predicted by Cauchy's theorem: $N = Z - P = 1$.

To explore the theorem further, consider the three examples that can be found in Figure 5.4. The key values of Cauchy's theorem (N, Z and P) have been indicated below each of the drawings.



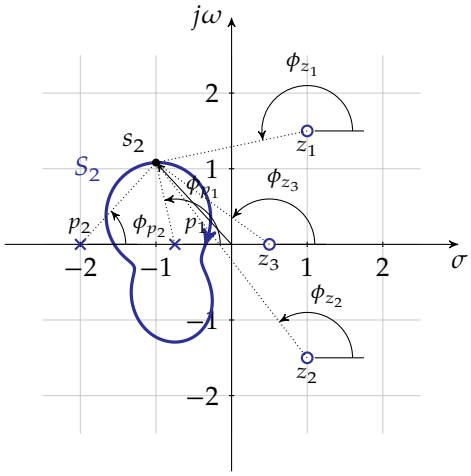
$Z = 3$ and $P = 1$

(a) Original contour S_1



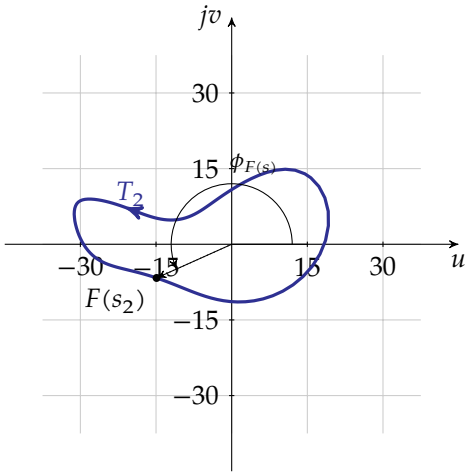
$N = 2$

(b) Mapped contour $T_1 = F(S_1)$



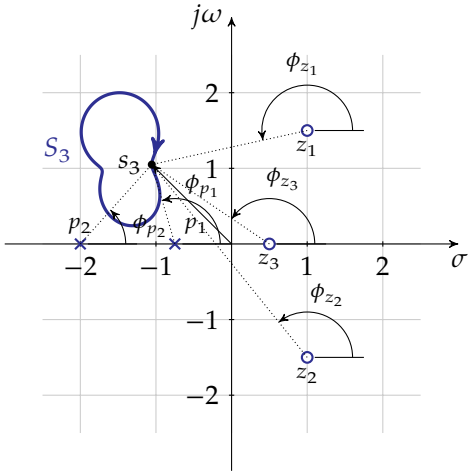
$Z = 0$ and $P = 1$

(c) Original contour S_2



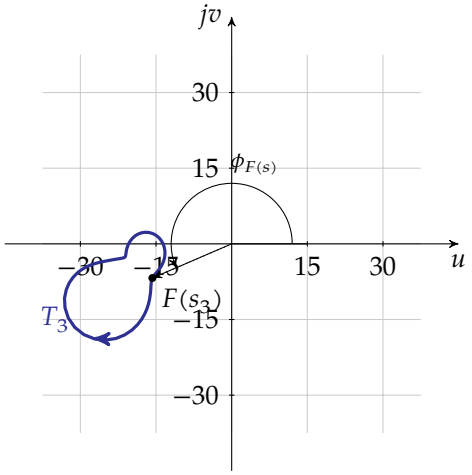
$N = -1$

(d) Mapped contour $T_2 = F(S_2)$



$Z = 0$ and $P = 0$

(e) Original contour S_3



$N = 0$

(f) Mapped contour $T_3 = F(S_3)$

Figure 5.4: Examples of the mapping by a function $F(s)$ (that is a ratio of two polynomials in the Laplace domain, with poles and zeros as indicated) of a contour not going through the poles and zeros of $F(s)$

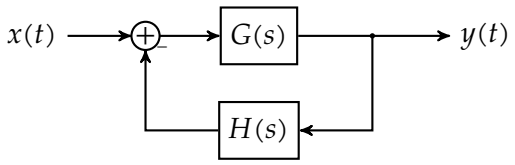
We won't give a rigorous proof, but it is easy to see in Figure 5.3a that:

- poles and zeros inside the original contour contribute a full revolution to the phase of $F(s)$, for zeros a CW revolution and for poles a CCW revolution.
  Consider for example z_1: traversing the original contour S in the CW direction will make the phase of $s - z_1$ traverse a full CW revolution as well.
  If z_1 would have been a pole, it would be in the denominator of $F(s)$ and therefore contribute negatively to the phase.
- poles and zeros outside the original contour have a phase that oscillates between specific extreme phases that can be obtained by drawing the tangential lines from the pole or zero to the contour.

Now, why do we need this strange theorem? Well the original contour is like a fishing net that we can use to hunt down any undesired poles or zeros. The mapped contour will tell us whether we caught anything. However, we can't use the theorem straight away. We will show how we can in the next section.

5.1.4 The Nyquist criterion

Consider the following generic feedback system



The function $T(s)$ can be written as a function of the loop gain $L(s)$:

$$T(s) = \frac{G(s)}{1 + \underbrace{G(s)H(s)}_{L(s)}} = \frac{G(s)}{\underbrace{1 + L(s)}_{F(s)}} = \frac{G(s)}{F(s)}$$

The system will be stable if $F(s)$ has no zeros in the right-half plane. Now, let's apply our Cauchy fishing net:

- take a CW contour S that encloses the full right-half plane
- construct the mapped contour $F(S)$ and determine the number of CW rotations around the origin
- according to Cauchy, the number of zeros we caught will be equal to:

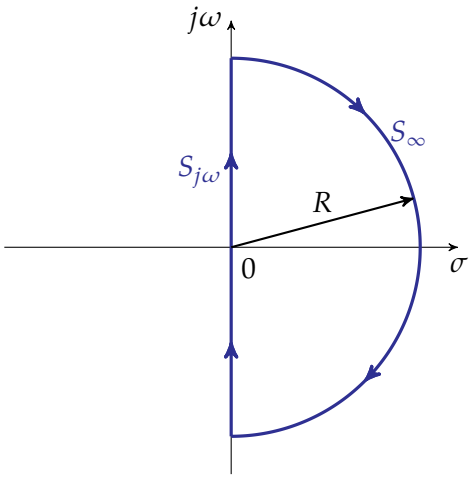
  $$Z = N + P$$

  with P the number of poles of $F(s)$ in the right-half plane. Note that these poles can be directly observed as the poles of $L(s) = G(s)H(s)$.[2]

If $Z = N + P > 0$, then the system is unstable.

[2]Indeed: poles of $F(s)$ are points for which $F(s)$ becomes infinitely large; this can only happen if $L(s)$ grows infinitely large.

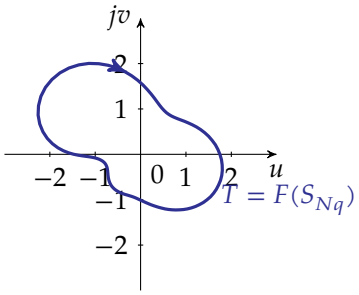
The way to construct a CW contour S that encloses the full right-half plane is obvious: start on the imaginary axis at $-j\infty$, traverse the imaginary axis to $+j\infty$ and enclose the right-half plane by returning on a half circle with a radius $R \to \infty$. We will label the trajectory on the imaginary axis as $S_{j\omega}$ and the half circle at infinity as S_{inf}. The total trajectory as S_{Nq} (with Nq for Nyquist).
This has been indicated below:

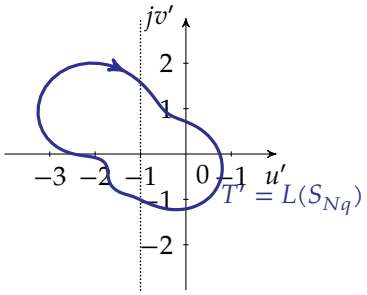


The only problem we still face, is that we need to be able to draw $F(S)$ and therefore need to know $F(s)$ (by analysis or measurement). And here comes the essence of Nyquist's theorem: *we can avoid calculating or measuring* $F(s)$, by realizing that:

$$L(s) = F(s) - 1$$

The meaning of this equation has been illustrated by the intuitive drawings below:





Therefore, the phrases out of Cauchy's theorem

1. N the number of CW revolutions of the mapped contour $T = F(S)$ around the origin,
2. Z the number of zeros of $F(s)$ inside the original contour S, and
3. P the number of poles of $F(s)$ inside the original contour S.

can be replaced by (changes have been underlined):

1. N the number of CW revolutions of the mapped contour $\underline{T' = L(S)}$ around $\underline{-1}$,
2. Z the number of zeros of $\underline{L(s)}$ inside the original contour S, and
3. P the number of poles of $\underline{L(s)}$ inside the original contour S.

Now, let's consider the two parts of our contour S_{Nq} and see how they are mapped by $L(s)$. Assuming that

$$L(s) = K \cdot \frac{\prod_{j=1}^{m}(s - z_j)}{\prod_{i=1}^{n}(s - p_i)}$$

we can make the following observations:

- **The part S_∞** is mapped *depending the degree of the numerator m minus the degree of the denominator n of $L(s)$*. Considering the limit case $s \to \infty$ results in the following conclusions:
  1. if $m - n < 0$, then for large s the loop gain $L(s)$ can be approximated by:

     $$L(s) \approx \frac{K}{s^{n-m}}$$

     Therefore the mapping will result in $L(S_\infty) = 0$.
  2. if $m - n = 0$, then the loop gain $L(s)$ can be approximated by:

     $$L(s) \approx K$$

     In this case, the mapping will result in a constant $L(S_\infty) = K$.
  3. if $m - n > 0$, then the loop gain $L(s)$ can be approximated by:

     $$L(s) \approx Ks^{m-n}$$

     The mapping will now stay at infinity and perform a number of $m - n$ half circles, starting from $s = j\infty$ clockwise, or from $s = -j\infty$ also clockwise, depending on the sign of K.

  In almost any physical case we will have $n \geq m$, as we would otherwise loop gain would increase to infinity if ω goes to infinity. Therefore, we can safely assume that this part of the contour will not contribute to enclosing $s = -1$.
- **The part $S_{j\omega}$** corresponds to substituting $s = j\omega$, i.e. calculating the Fourier transform of $L(s)$. The conclusion is obvious: The mapped contour $L(S_{j\omega}) = L(j\omega)$ is the polar plot of $L(j\omega)$ (i.e. in the frequency domain)!

Before formulating the Nyquist criterion, there is one last detail we need to discuss. We want to determine $Z = N + P$. So far, we have seen that we can count N by counting the CW revolutions of $L(s)$ around $s = -1$. However, $L(s)$ may also have poles. Therefore P may be nonzero. Let's elaborate on that:

- In most cases $L(s)$ will be stable and *have no poles in the right-half plane*, i.e. no poles within our original zero-hunting contour S_{Nq}. In that case $P = 0$.
  Note that only in this case, we will be able to determine $L(j\omega)$ by measurement. We cannot measure the frequency response of an unstable system.
- In the other case, where $L(s)$ is not stable, there *will be poles in the right-half plane*, i.e. poles within our original zero-hunting contour S_{Nq}. In that case $P \neq 0$.
  Note that in this case $L(s)$ cannot be measured. It must come from a model generated based on our knowledge of the physics of the system.

This leads to the generic Nyquist criterion:

The Nyquist criterion
Considering for a single-loop feedback system S_{Nq} in the s-domain enclosing P poles of $L(s)$, then the feedback system is stable if $L(S_{Nq})$ circles around the point -1 only P times.

In measurable cases (i.e. stable loop gain), $P = 0$ and then the Nyquist criterion can be simplified to:

The Nyquist criterion for stable loop gains
A single-loop feedback system is stable if $L(S_{Nq})$ does not circle around -1.

Let's illustrate this with some examples.

Example 1

$$L(s) = \frac{K}{1 + \tau s}$$

Knowing that the degree of the denominator $n = 1$ is larger than the degree of the numerator $m = 0$, we know that $L(S_\infty)$ will be mapped to 0.

Calculating $T' = L(S_{j\omega})$ is easy:

$$T' = L(s)\Big|_{s=j\omega} = \frac{K}{1 + j\omega\tau}$$

We have plotted this function, as part of the first example of polar plots in section 5.1.1 on page 96. Repeating the same effort, yields the polar plot of Figure 5.6.

A number of things can be observed in this plot:

- The mapped contour does not circle around -1, therefore the system is stable.
- Because of the mirror symmetry of S_{Nq} w.r.t. the σ-axis, and the complex conjugate nature of poles and zeros, the plot of $L(S_{Nq})$ is also mirror symmetric w.r.t. the u'-axis. Therefore, we can limit ourselves to calculating the plot for positive frequencies and (in our mind) mirror the result around the u'-axis.

Example 2

$$L(s) = \frac{K}{s(1 + \tau s)}$$

Knowing that the degree of the denominator $n = 2$ is larger than the degree of the numerator $m = 0$, we know that $L(S_\infty)$ will be mapped to 0.

However, for $S_{j\omega}$ another complication arises: it will cross one of the poles of the system (at the origin, $s = 0$). Therefore, we need to take a slightly deviant path, avoiding that pole. A possible solution using the arc in gray, has been indicated in Figure 5.7a.

Let's consider the four segments of S_{Nq}:

- We have found earlier that $L(S_\infty) = 0$.

- Calculating $T' = L(S_{j\omega})$ is easy:

$$T' = L(s)\Big|_{s=j\omega} = \frac{K}{j\omega(1+j\omega\tau)}$$

- $T' = L(S_{-j\omega})$ can be found by mirroring the previous result (the green curve).
- $T' = L(S_\epsilon)$ can be calculated by considering that the gray arc is described by: $s = \epsilon\, e^{j\phi}$ with $\phi = [-\pi/2, +\pi/2]$. Therefore:

$$\begin{aligned}\lim_{\epsilon\to 0} L(s)\Big|_{s=\epsilon e^{j\phi}} &= \lim_{\epsilon\to 0} \frac{K}{\epsilon\, e^{j\phi}(1+\tau\epsilon\, e^{j\phi})} \\ &= \lim_{\epsilon\to 0} \frac{K}{\epsilon\, e^{j\phi}} = \lim_{\epsilon\to 0} \frac{K}{\epsilon}\, e^{-j\phi}\end{aligned}$$

  The latter is an arc with an infinitely large radius with its argument ranging from $\pi/2$ to $-\pi/2$

We have plotted the polar plot before as part of the second example of polar plots in section 5.1.1 on page 96. Repeating the same effort, yields the polar plot of Figure 5.7b.

Similar conclusions can be drawn as for the first example:

- The mapped contour does not circle around -1 (independent of the value of K), therefore the system is absolutely stable.
- Again, we could restrict ourselves to drawing the blue and the gray curves, to draw our conclusions.

Example 3

$$L(s) = \frac{K}{s(1+\tau_1 s)(1+\tau_2 s)}$$

Again, knowing that the degree of the denominator $n = 3$ is larger than the degree of the numerator $m = 0$, we know that $L(S_\infty)$ will be mapped to 0.

A proper contour S_{Nq} can be found in Figure 5.8a.

Let's consider the four segments of S_{Nq}:

- We know that $L(S_\infty) = 0$.
- Given the fact that we did not discuss $T' = L(S_{j\omega})$ before (when discussing polar plots), let's explain in a little more detail how to plot this curve by hand.
  We know that:

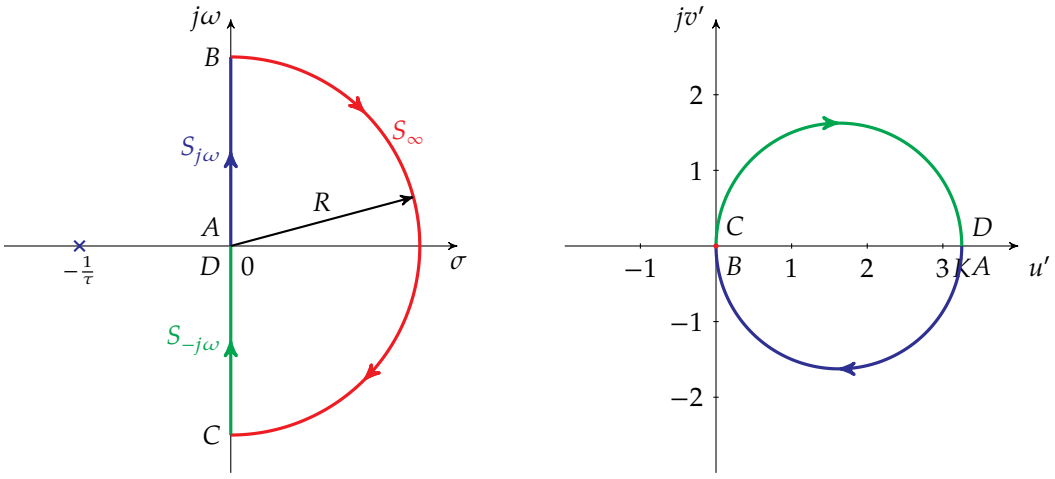
$$T' = L(s)\Big|_{s=j\omega} = \frac{K}{j\omega(1+j\omega\tau_1)(1+j\omega\tau_2)}$$

  Let's consider this equation for very small and very large values of ω, i.e. let's try to find its starting point and its ending point.

  - $\omega \to 0$
    In this case, we can approximate

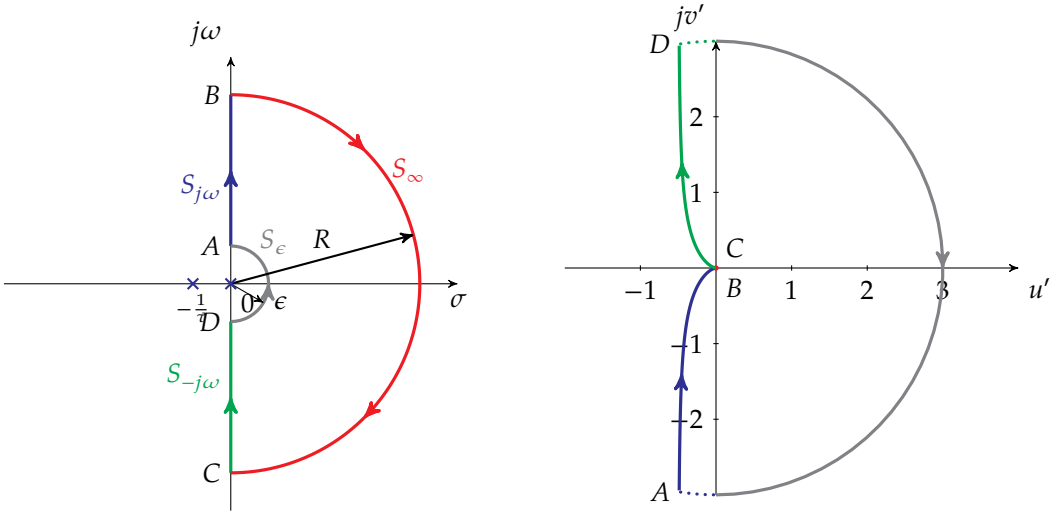
$$T' \approx \frac{K}{j\omega} = -j\frac{K}{\omega}$$

    This means that for small ω the curves start at $-j\infty$.



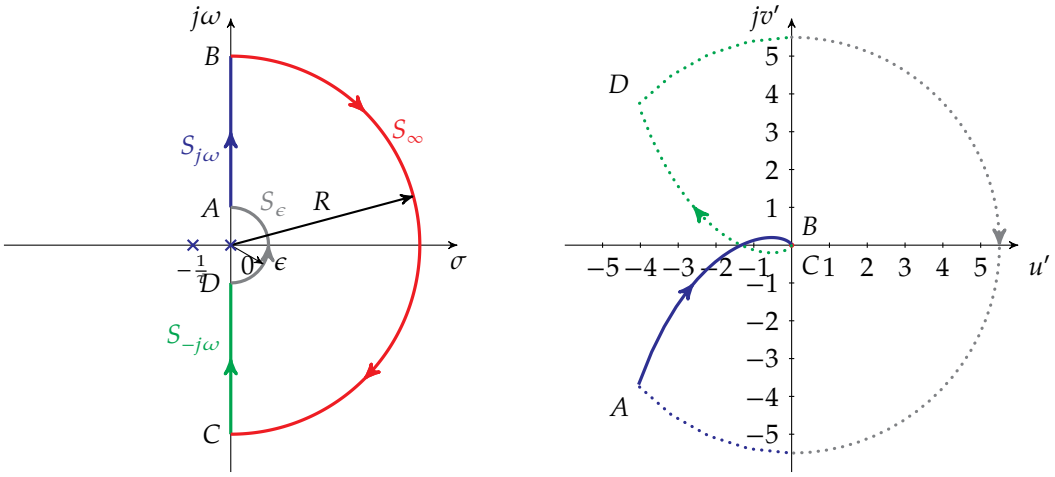
(a) Original contour $S_{Nq} = S_{-j\omega} \cup S_{j\omega} \cup S_\infty$ **(b)** Mapped contour $T' = L(S_{Nq})$

Figure 5.6: Illustration of the Nyquist criterion — example 1



(a) Original contour $S_{Nq} = S_{-j\omega} \cup S_\epsilon \cup S_{j\omega} \cup S_\infty$ **(b)** Mapped contour $T' = L(S_{Nq})$

Figure 5.7: Illustration of the Nyquist criterion — example 2



(a) Original contour $S_{Nq} = S_{-j\omega} \cup S_\epsilon \cup S_{j\omega} \cup S_\infty$ **(b)** Mapped contour $T' = L(S_{Nq})$

Figure 5.8: Illustration of the Nyquist criterion — example 3

- $\omega \to +\infty$
  In this case, we can approximate

$$T' \approx \frac{K}{j\omega \cdot j\omega\tau_1 \cdot j\omega\tau_2} = j\frac{K}{\omega^3\tau_1\tau_2}$$

  This means that for large ω the curve will approximate 0 from the positive jv'-axis.

OK, so now we have its start and ending point. What about the rest of the curve? Our main interest, is whether and where it intersects with the real axis. To this end, we determine (try yourself!) the real and imaginary parts to be:

$$u'(\omega) = -\frac{K(\tau_1 + \tau_2)}{(1 + \omega^2\tau_1^2)(1 + \omega^2\tau_2^2)}$$
$$v'(\omega) = -\frac{K(1 - \omega^2\tau_1\tau_2)}{\omega(1 + \omega^2\tau_1^2)(1 + \omega^2\tau_2^2)}$$

Setting the imaginary part $v'(\omega)$ to 0 allows us to determine any intersection with the real axis. We will label the ω for which this happens as ω_{180}:

$$\omega_{180} = \frac{1}{\sqrt{\tau_1\tau_2}}$$

In this case, the crossing point with the real axis is easily found as:

$$u'(\omega_{180}) = -K\frac{\tau_1\tau_2}{\tau_1 + \tau_2}$$

If this crossing point occurs to the left of -1, then the system will be unstable. Therefore, the condition for stability is:

$$K < \frac{\tau_1 + \tau_2}{\tau_1\tau_2}$$

- $T' = L(S_{-j\omega})$ can be found by mirroring the previous result (the dotted green curve).
- In a similar way as for the previous case, one can conclude that $L(S_\epsilon)$ is an arc with an infinitely large radius with its argument ranging from $\pi/2$ to $-\pi/2$.

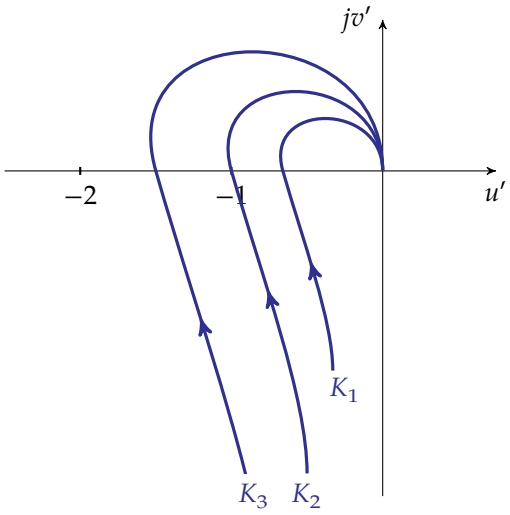
The result has been plotted in Figure 5.8b for an unstable case. We used dots for the curves that we normally don't draw but only imagine in our minds.

The unstability shows itself in the fact that the polar plot of $L(j\omega)$ circles around -1.

5.1.5 Gain margin and phase margin

Given our knowledge about root-locus plots in the Laplace domain, it is clear that K has an effect on stability. The effect of K on the polar plot of $L(j\omega)$ is obvious: it scales the plot, because the magnitude scales linearly with K and the phase is totally independent of K (except for its sign).

This is illustrated below for an arbitrary loop gain $L(s)$ and three different values of K:



For K_1 the feedback system is stable, for K_2 borderline stable and for K_3 unstable.

When given a particular polar plot (like the one for K_1 in the example above), an obvious question is: how much can I increase K_1 to get to the border of stability?

This leads to the concept of *gain margin*.

Another issue in feedback systems are time-delays (by components that take a moment to measure, compute, transmit or transport something). These time delays are in the Fourier domain translated as an extra phase delay. Indeed:

$$x(t - t_0) \xrightarrow{\mathcal{F}} X(\omega)\, \mathrm{e}^{-j\omega t_0}$$

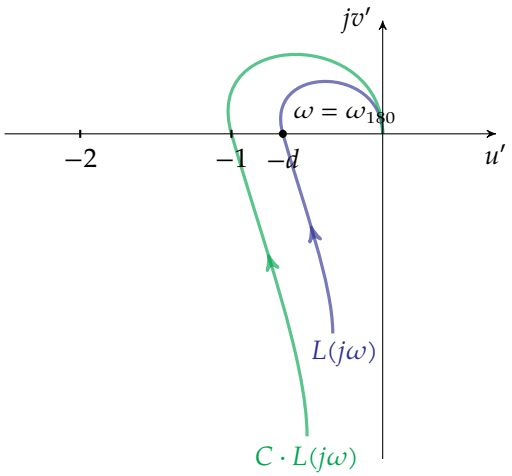
On a polar plot, an extra time delay corresponds to rotating the polar plot around the origin. An obvious questions is: how much of extra rotation (i.e time delay) can I allow, to get to the border of stability.

This leads to the concept of *phase margin*.

Gain margin

Consider the polar plot in blue below, of an arbitrary loop gain $L(s)$. It intersects with the negative real axis in $-d$ (indicated in blue on the graph below). The frequency for which it does so, is a special frequency and we will call that the *one-hundred-and-eighty-degrees phase change frequency*[3] and we will label it ω_{180}. Note that ω_{180} is not a function of K.

[3] Admittedly, this is quite a mouth full and often the term *the one-eighty frequency* is used.



Gain margin

The gain margin A_{GM} is the value of the extra loop gain C that makes the polar plot go through -1. Assuming the original intersection occurred at $-d$, this means:

$$A_{GM} = \frac{1}{d}$$

Often the gain margin is expressed in decibels, i.e.

$$A_{GM_{dB}} = 20 \log_{10} \frac{1}{d}$$

The polar plot with the correct amount of gain added (A_{GM}) is indicated on the graph above in green.

Note that the gain margin can be smaller than one (or negative when expressed in dB). In this case, we have an unstable system and we need to scale down the loop gain in order to become stable.

Considering the third example of previous section, we know that:

$$-d = -K \frac{\tau_1 \tau_2}{\tau_1 + \tau_2}$$

Assuming $K = 0.5$, $\tau_1 = \tau_2 = 1$, this yields $d = 1/4$ and therefore:

$$A_{GM} = 4 \approx 12\,\text{dB}$$

In words: we can increase the gain by a factor of 4 before the system becomes borderline stable.

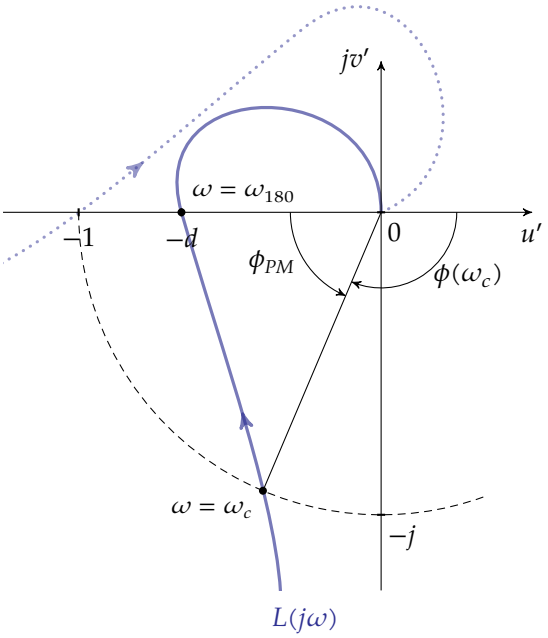
Phase margin

Consider the polar plot, in solid blue below, of an arbitrary loop gain $L(s)$. The frequency for which the polar plot crosses the unit circle (i.e. loop gain equal to 1) is a special frequency, and we will call that the *crossover frequency* and label it ω_c.[4]

An extra delay will cause the polar plot to rotate clockwise. The situation for which the system becomes borderline stable (and ω_c coincides with -1) has been plotted in dotted line on the

[4] Another term that is often used is the *unity-gain frequency*.

graph. This borderline situation is obtained by adding additional phase delay of ϕ_{PM}. This value is called the *phase margin*.



Phase margin

The phase margin ϕ_{PM} is the extra phase delay one can introduce to a polar plot before it will go through -1. Assuming the polar plot crosses the unit-gain circle at ω_c, this means:

$$\phi_{PM} = \phi(\omega_c) - (-180^\circ) = 180^\circ + \phi(\omega_c)$$

Note that the phase margin can also be negative, indicating the feedback system is unstable.

Recap

Because the two special frequencies we have introduced in this section are so important, we list their definition once again in the table below:

Symbol	Name	Definition	Purpose
ω_{180}	one-eighty frequency	$\angle L(j\omega_{180}) = -180^\circ$	Used to find the gain margin
ω_c	cross-over frequency	$\lvert L(j\omega_c)\rvert = 1$	Used to find the phase margin

Note how the gain margin uses a phase-defined frequency and the phase margin uses a gain-defined frequency. It is of utmost importance that you do not mix up these two frequencies!

5.1.6 Computational help

Luckily many mathematical suites offer tools to help you in using the Nyquist criterion. MATLAB is one of them. Analyzing a simple system like

$$L(s) = \frac{10}{4s+1}$$

is as simple as:

```
>> L = tf( 10, [4 1] );
>> nyquist( L );
```

Exercises

Exercise 5.1.6-1: Draw the Nyquist plot for the following system:

$$L(s) = \frac{1}{s(4s+1)}$$

Verify your result in MATLAB.

Exercise 5.1.6-2: Draw the Nyquist plot for the following system:

$$L(s) = \frac{5}{4s^2+2s+1}$$

Verify your result in MATLAB.

Exercise 5.1.6-3: Draw the Nyquist plot for the following system:

$$L(s) = \frac{5}{s(4s^2+2s+1)}$$

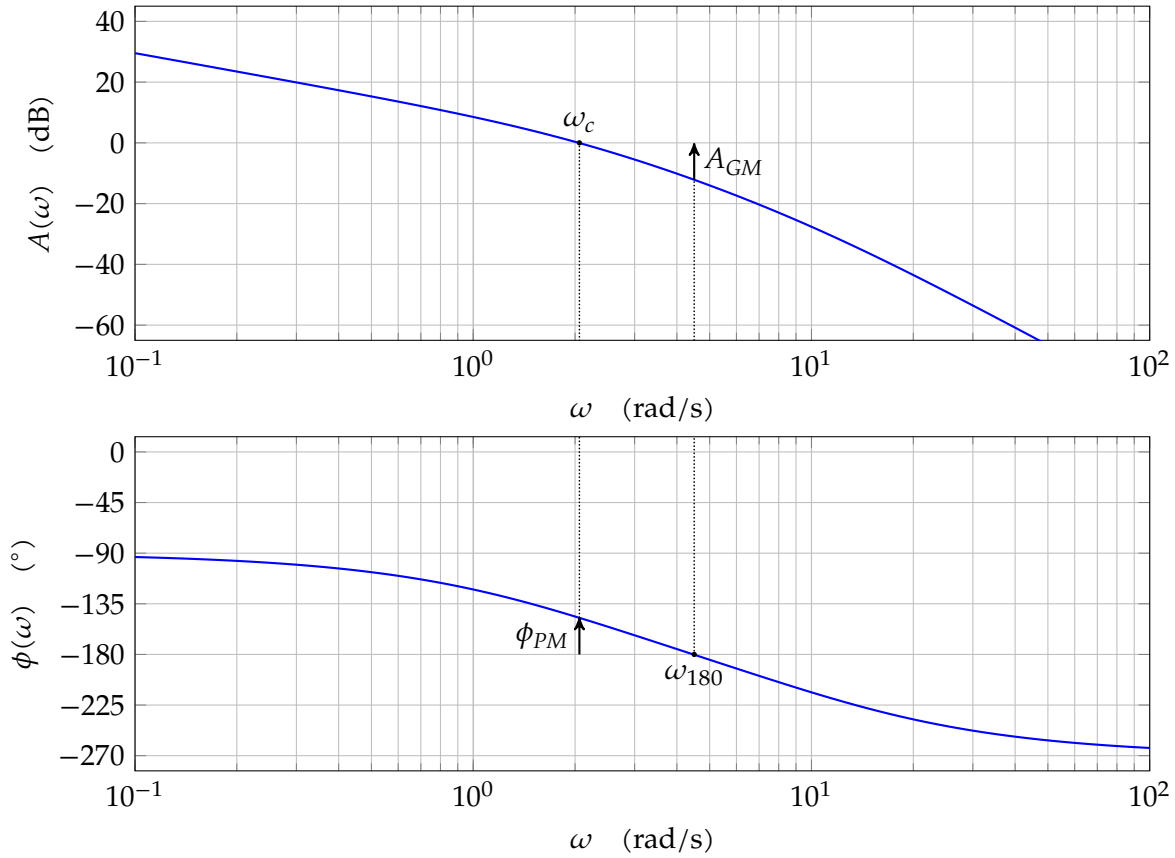
Verify your result in MATLAB.

5.2 The Bode plot

No matter how big a fan you are of polar plots, there is one thing to admit: they are a pain to compose. The Bode diagram, on the contrary, is much simpler to compose. As both plots (polar and Bode) are a different view on the same data, it should be perfectly possible to translate our insights on gain and phase margin to Bode plots. That is what we will do in this section.

5.2.1 Gain margin and phase margin

If you consider the definition of the gain and the phase margin, it is easy to understand that they are easy to read on a Bode plot.
The gain margin is read on the magnitude diagram wherever the phase delay amounts to 180° (i.e. at ω_{180}).
The phase margin is read on the phase graph wherever the magnitude crosses the 0 dB line (i.e. at ω_c).
This has been indicated on the graph below.

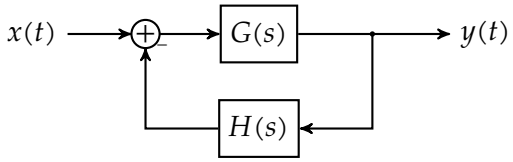


Reading on the graph that $\phi(\omega_c) = -147°$ and $A(\omega_{180}) = -12\,\text{dB}$, this leads to:

$$A_{GM} = 12\,\text{dB}$$
$$\phi_{PM} = 180° + \phi(\omega_c) = 33°$$

5.2.2 A common misconception w.r.t. stability

Consider once again the following system:



Very often one explains (erroneously) the Nyquist criterion as follows: when the phase delay of the loop gain $L(s) = G(s) \cdot H(s)$ amounts to 180°, the negative feedback becomes a positive feedback, because a phase delay of 180° corresponds to a multiplication with -1. If, in that case, the gain is larger than 1, i.e. $|L(j\omega)| > 1$, the system will become unstable.

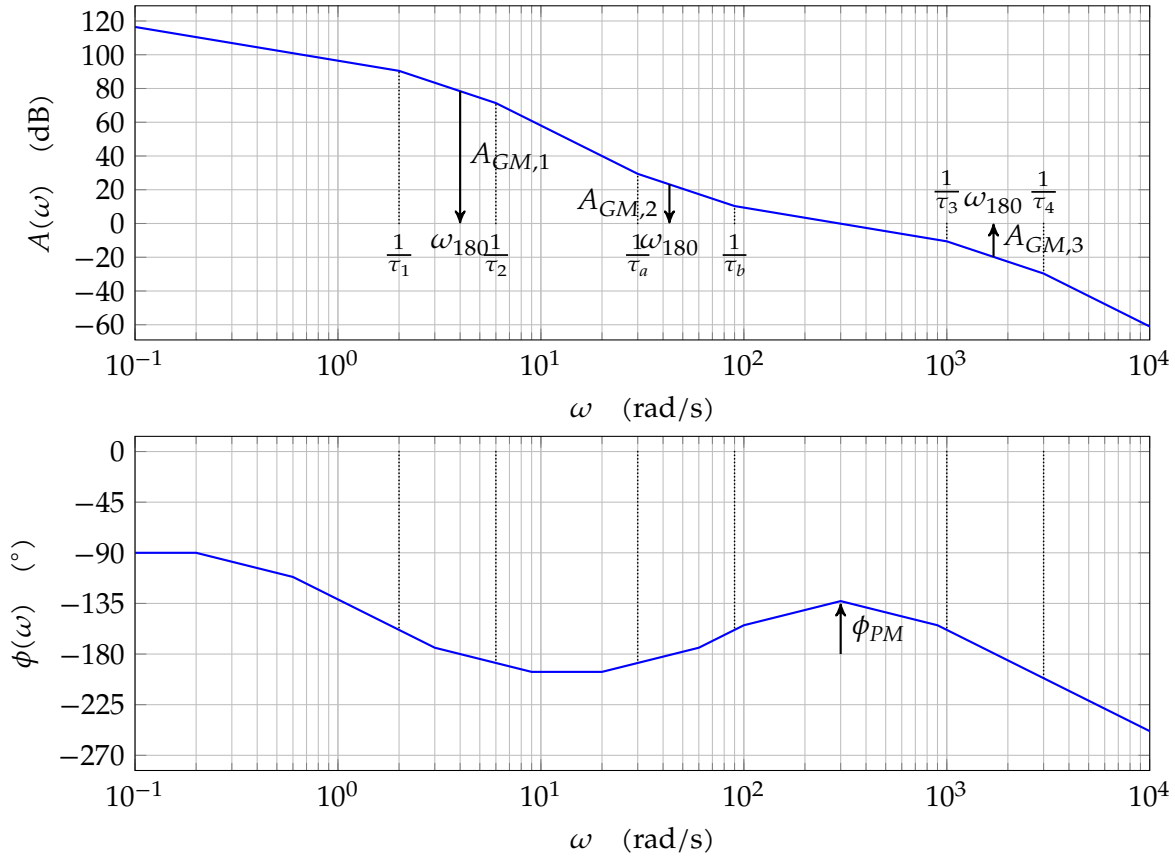
It cannot be stressed enough that this reasoning is *wrong!*

The only true stability criterion is Nyquist's criterion, and that states that the phase margin should be positive! The gain margin only tells you something in case of a stable system, more specifically how much you can increase the gain before the system becomes unstable.

We will illustrate this with an example. Consider a system with the following loop gain:

$$L(s) = K\frac{(1+\tau_a s)(1+\tau_b s)}{s(1+\tau_1 s)(1+\tau_2 s)(1+\tau_3 s)(1+\tau_4 s)} \tag{5.1}$$

Let's assume $\tau_1 > \tau_2 > \tau_a > \tau_b > \tau_3 > \tau_4$. The (asymptotic) Bode plot of the system can be inspected below:



The gain margins $A_{GM,1}$ and $A_{GM,2}$ are clearly negative. However, the system *is* stable, as the phase margin is positive, and therefore, the polar plot will not circle around -1.

You can find the corresponding polar plot in Figure 5.9, proving the latter statement. Note how the linear scaling of the polar plot again hinders an easy use of such a plots.

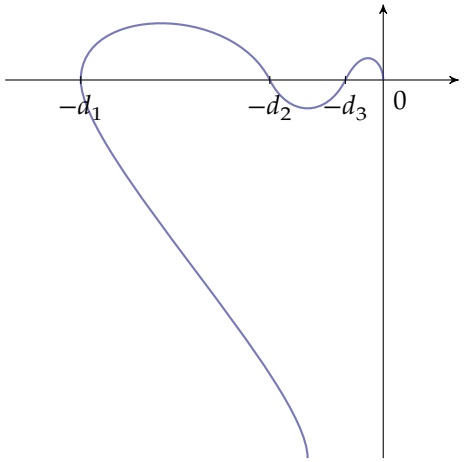
Note that if you decrease the gain with 40 dB (i.e. shifting the magnitude graph of the Bode plot downwards), the system will become unstable.

So please remember:

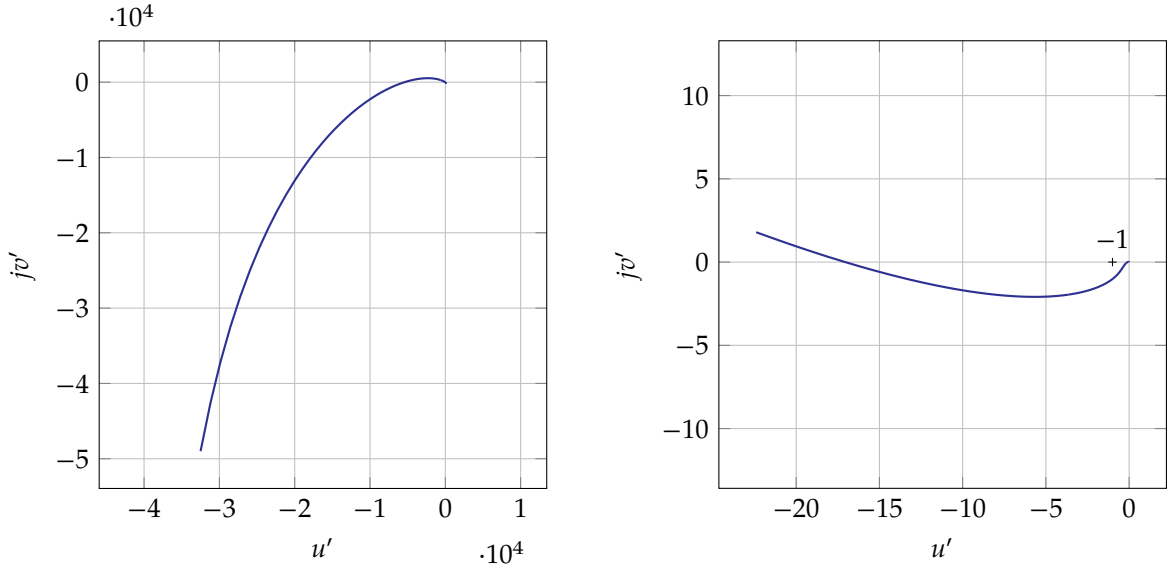
Only the phase margin determines stability!

5.3 Intermezzo: phase margin and damping factor

For a second-order system there is a simple and direct relationship between the phase margin ϕ_{PM} and the damping factor ζ. Deriving this relationship is straightforward.



(a) Artistic impression (showing the overall shape) of the polar plot



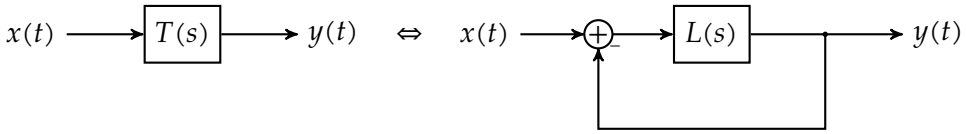
(b) Calculated polar plot on a coarse scale (left) and a finer scale (right): the polar plot starts at $-j\infty$ goes left and up to cross the (negative) real axis a first time (see left plot), to come down again and cross it again avoiding to circle -1 (see right plot), again crossing the (negative) real axis upwards close to the origin to target the origin from above (almost invisible on the plot).

Figure 5.9: Polar plot of the example system of (5.1)

Consider a second-order system in its normalized form:

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

A first question to answer is: how does the loop gain $L(s)$ look like for a feedback system that is equivalent to this second-order system $T(s)$?



Given

$$T(s) = \frac{L(s)}{1 + L(s)} \qquad \Rightarrow \qquad L(s) = \frac{T(s)}{1 - T(s)}$$

it is easy to derive:

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

In order to calculate the phase margin of this system, we need to determine the cross-over frequency ω_c defined by the following condition:

$$|L(j\omega_c)| = 1$$
$$\left|\frac{\omega_n^2}{j\omega_c(j\omega_c + 2\zeta\omega_n)}\right| = 1$$
$$\frac{\omega_n^2}{\omega_c\sqrt{\omega_c^2 + (2\zeta\omega_n)^2}} = 1$$

If we assume $\omega_c \geq 0$, then squaring does not introduce new solutions to this equation. Therefore squaring and reordering, yields:

$$\omega_n^4 = \omega_c^2(\omega_c^2 + (2\zeta\omega_n)^2)$$
$$\omega_c^4 + (2\zeta\omega_n)^2\omega_c^2 - \omega_n^4 = 0$$
$$\downarrow \quad \text{divide both sides by } \omega_n^4$$
$$\left(\frac{\omega_c}{\omega_n}\right)^4 + 4\zeta^2\left(\frac{\omega_c}{\omega_n}\right)^2 - 1 = 0$$

Solving for $(\omega_c/\omega_n)^2$ yields:

$$\left(\frac{\omega_c}{\omega_n}\right)^2 = -2\zeta^2 + \sqrt{4\zeta^4 + 1}$$

and isolating ω_c out of the latter, results in:

$$\omega_c = \omega_n\sqrt{-2\zeta^2 + \sqrt{4\zeta^4 + 1}} \tag{5.2}$$

This is a first important relationship that relates ω_n and ω_c. You can find a graphical plot of this relation in Figure 5.10.

The phase of $L(j\omega_c)$ can be found to be:

$$\phi(\omega_c) = -90^\circ - \text{atan}\left(\frac{\omega_c}{2\zeta\omega_n}\right)$$

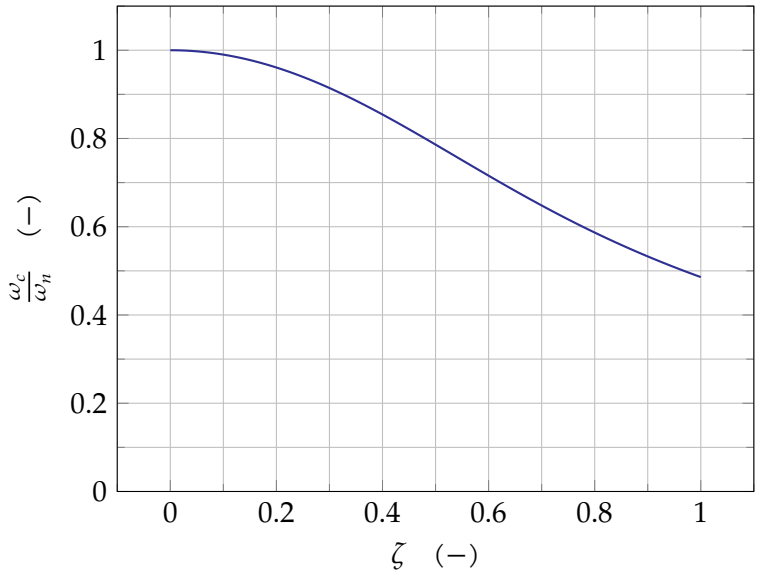


Figure 5.10: Ratio of ω_c/ω_n vs. damping factor ζ for a second-order system without zeros.

Therefore:

$$\phi_{PM} = 90^\circ - \text{atan}\left(\frac{\omega_c}{2\zeta\omega_n}\right) = 90^\circ - \text{atan}\left(\frac{\cancel{\omega_n}\sqrt{-2\zeta^2+\sqrt{4\zeta^4+1}}}{2\zeta\cancel{\omega_n}}\right)$$
$$= 90^\circ - \text{atan}\left(\frac{\sqrt{-2\zeta^2+\sqrt{4\zeta^4+1}}}{2\zeta}\right) \tag{5.3}$$

Using equations (5.2) and (5.3), we can plot the curve of Figure 5.11.

As one can see a very decent approximation (the green curve) is given by:

$$\zeta = 0.01\phi_{PM} \tag{5.4}$$

5.3.1 Computational help

Using Bode plots and the phase-margin criterion is well supported in many mathematical tools. In MATLAB plotting a Bode plot is very straightforward. Assuming as example a system with loop gain

$$L(s) = \frac{10}{s(0.001s^2 + 0.11s + 1)}$$

a Bode plot can be generate as easily as:

```
>> L = tf( 10, [ 0.001, 0.11, 1, 0] );
>> bode( L );
>> grid on;
```

An alternative version exists that shows you the phase margin superimposed on the plot:

```
>> L = tf( 10, [ 0.001, 0.11, 1, 0] );
>> margin( L );
>> grid on;
```

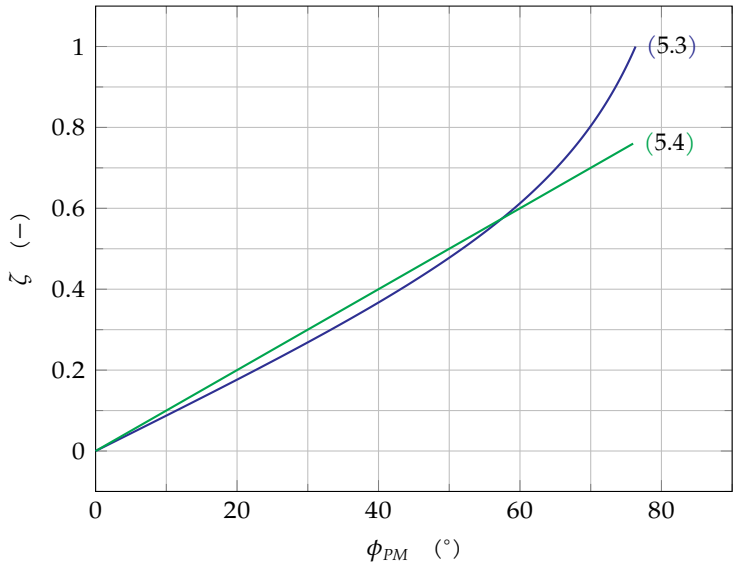


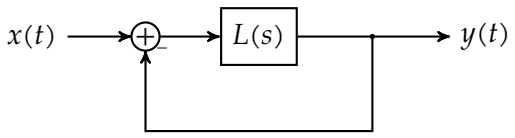
Figure 5.11: Damping factor vs. phase margin for a second-order system without zeros: true curve (in blue) and approximation (in green)

5.4 The polar plot revisited

The polar plot of loop gain, can give us useful information about the magnitude and the phase response of the overall system. The points of constant magnitude and phase on the polar plot will prove to be circles, giving rise to:

- M-circles for the magnitude
- N-circles for the phase.

We will analyze the relationship between loop gain and overall gain, for the following generic system:



5.4.1 M-circles

Let's try to find the locus of points $u + jv$ in the $L(j\omega)$-domain that exhibit a constant overall magnitude M, i.e.

$$\begin{aligned}
|T(j\omega)| &= M \\
&\Big\downarrow\ T(j\omega) = \frac{L(j\omega)}{1+L(j\omega)} \\
\left|\frac{L(j\omega)}{1+L(j\omega)}\right| &= M \\
&\Big\downarrow\ L(j\omega) = u + jv \\
\left|\frac{u+jv}{1+u+jv}\right| &= M \\
\frac{\sqrt{u^2+v^2}}{\sqrt{(1+u)^2+v^2}} &= M \\
u^2+v^2 &= M^2((1+u)^2+v^2) \\
(1-M^2)u^2 + (1-M^2)v^2 - 2M^2u &= M^2
\end{aligned}$$

Depending on the value of M we can distinguish two cases:

- $M = 1$
  In this case the equation reduces to:

  $$-2u = 1 \qquad \Leftrightarrow \qquad u = -1/2$$

  which is a straight vertical line in the polar plot (see Figure 5.12, blue line).
- $M \neq 1$
  In this case we can divide both sides of the equation by $1 - M^2$:

  $$u^2 + v^2 - 2\frac{M^2}{1-M^2}u = \frac{M^2}{1-M^2}$$

  which has the form of a circular locus. Completing the squares yields:

  $$\left(u - \frac{M^2}{1-M^2}\right)^2 + v^2 = \frac{M^2}{1-M^2} + \left(\frac{M^2}{1-M^2}\right)^2$$

  $$\left(u - \underbrace{\frac{M^2}{1-M^2}}_{\equiv a}\right)^2 + v^2 = \left(\underbrace{\frac{M}{1-M^2}}_{\equiv R}\right)^2$$

  which is the equation of circle with center $(a, 0)$ and radius R. As examples, the circles for $M = 1.5$ and $M = 2$ have been indicated in green on Figure 5.12.
  It can be easily checked that circles for a value $M = K$ (in green) have the same radius as the circle for value $M = 1/K$ (in red) and a center that is located opposite of $u = -1/2$. This has been indicated on Figure 5.12. Indeed, the locus composition shows a remarkable symmetry around $u = -1/2$.

The obtained circles are the so-called *M-circles*. On a polar plot they form equimagnitude lines that can be used to read the overall magnitude for every point of the polar plot.

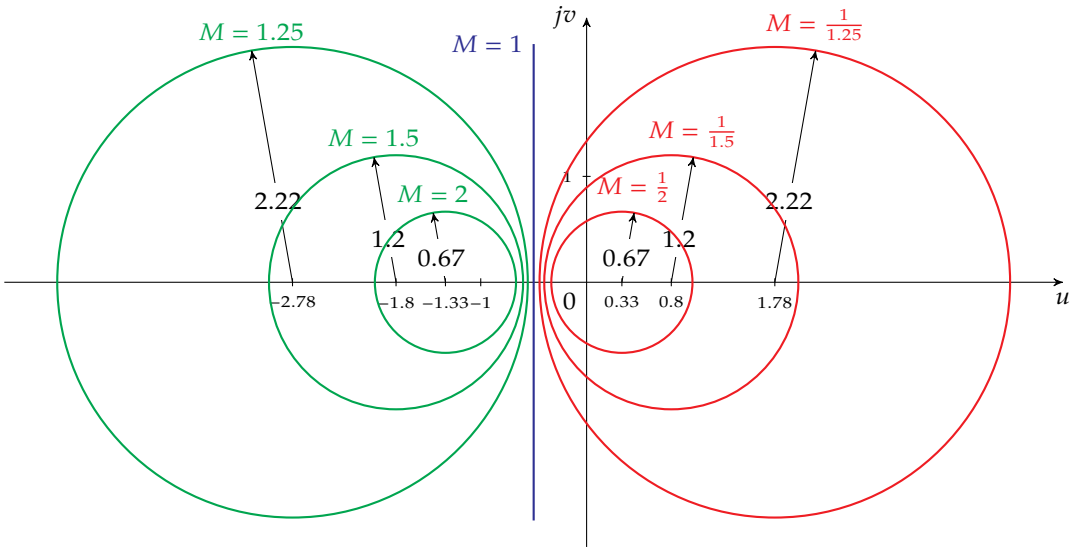


Figure 5.12: The locus of points with constant overall magnitude M gives rise to M-circles

5.4.2 N-circles

Finding the locus of points $u + jv$ in the $L(j\omega)$-domain that exhibit a constant overall phase ϕ is most similar:

$$\begin{aligned}
\angle T(j\omega) &= \phi \\
&\Big\downarrow\ T(j\omega) = \frac{L(j\omega)}{1 + L(j\omega)} \\
\angle L(j\omega) - \angle(1 + L(j\omega)) &= \phi \\
&\Big\downarrow\ L(j\omega) = u + jv \\
\operatorname{atan}\frac{v}{u} - \operatorname{atan}\frac{v}{1+u} &= \phi \\
&\Big\downarrow\ \tan \text{ of both sides and } \tan(\alpha - \beta) = \frac{\tan\alpha - \tan\beta}{1 + \tan\alpha\tan\beta} \\
\frac{\frac{v}{u} - \frac{v}{1+u}}{1 + \frac{v}{u}\frac{v}{1+u}} &= \underbrace{\tan\phi}_{\equiv N} \\
\frac{v}{u^2 + u + v^2} &= N \\
Nu^2 + Nu + Nv^2 - v &= 0
\end{aligned}$$

Depending on the value of N we can distinguish two cases:

- $N = 0$
  In this case the equation reduces to:

  $$v = 0$$

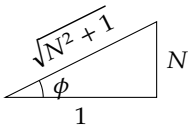
  which a straight line coinciding with the u-axis.
- $N \neq 0$
  In this case we can divde both sides of the equation by N, leading to:

  $$u^2 + u + v^2 - \frac{v}{N} = 0$$

wich has the form of a circular locus. Completing the squares yields:

$$\left(u+\frac{1}{2}\right)^2+\left(v-\frac{1}{2N}\right)^2=\frac{1}{4}+\frac{1}{4N^2}=\frac{N^2+1}{4N^2}=\left(\underbrace{\left|\frac{\sqrt{N^2+1}}{2N}\right|}_{\equiv R}\right)^2$$

This is the equation of a circle with center $(-1/2, 1/2N)$ and radius R. Considering $tan\phi = N$, the following rectangular triangle makes sense:



and therefore $R = \left|\frac{1}{2\sin\phi}\right|$.

The obtained circles are the so-called *N-circles*. On a polar plot they form equiphase lines that can be used to read the overall phase for every point of the polar plot.

5.4.3 The Hall chart

If you combine M- and N-circles on a single plot, one obtains the so-called *Hall chart* [Hal43]. You can find an empty template in Figure 5.13.

The advantage of a Hall chart is that one has a view on both the loop gain and the overall characteristics of the feedback system. However, the disadvantage is that drawing it (and altering it in case of changes) takes quite an effort.

In MATLAB the M-circles can be visualized on a Nyquist plot by turning the grid on.

5.5 The Nichols plot

To combine the ease of Bode plots (e.g. in case of a gain change) with the clarity of a Hall chart, one can make a Nichols plot [JNP47].

This plot sets out the loop gain magnitude (in dB) versus the loop gain phase. It can be composed by setting out phase vs. magnitude points of the Bode plot on a Nichols plot.

The example system with a loop gain of

$$L(s)=\frac{17}{s(1+0.1s)(1+0.02s)}$$

has a Bode plot as depicted in Figure 5.14. The corresponding Nichols plot has been depicted in Figure 5.16. It shows a trajectory for increasing values of ω.

In order for the depicted loop gain to correspond to a stable system, the plot needs to stay to the right of the point $(-180°, 0\,\text{dB})$. This is the case in this example. The phase and gain

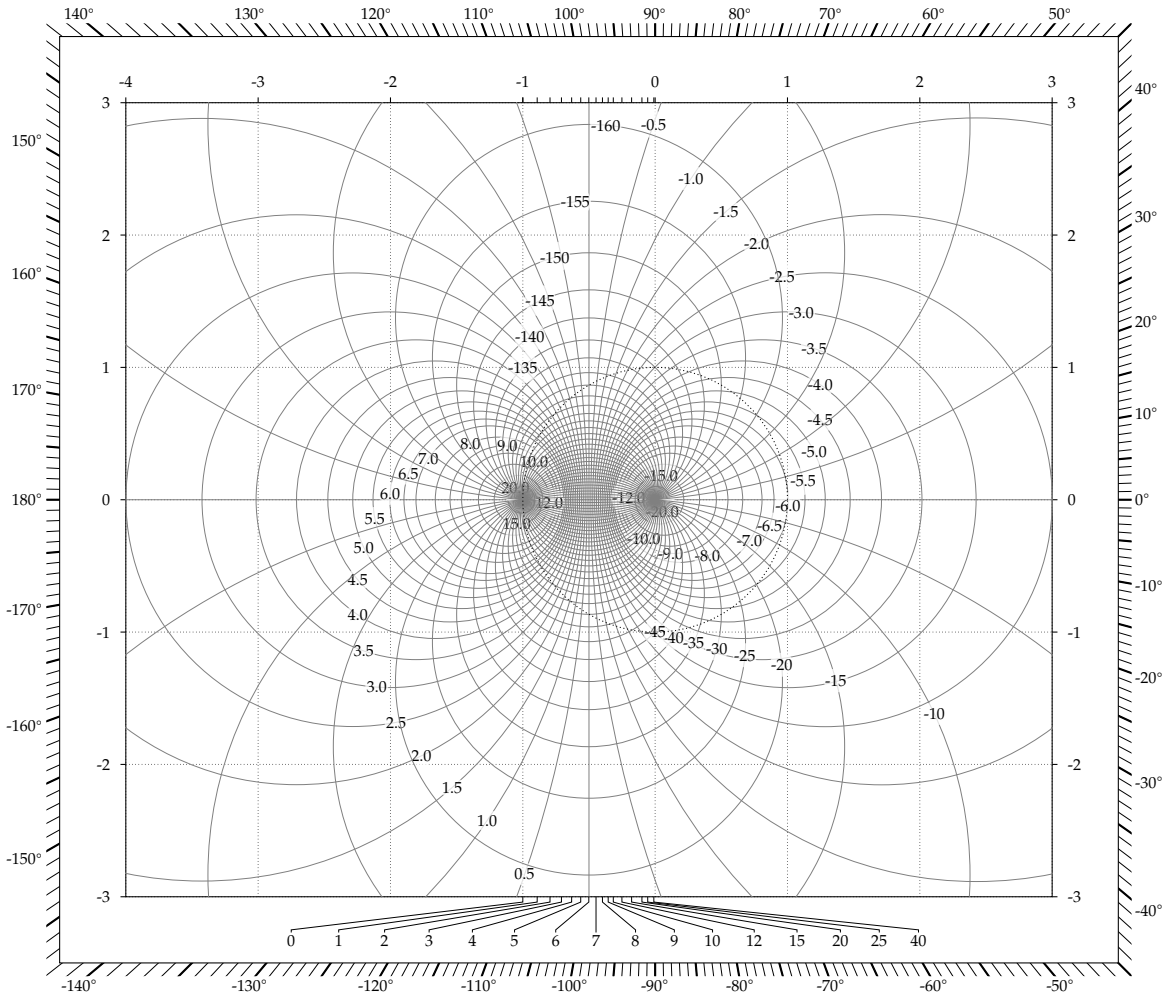


Figure 5.13: Hall chart of the loop gain with M-circles and N-circles ($[M]$ = dB, $[N]$ = °, $[GM]$ = dB)

margin have been indicated and can be easily read on this type of graph to be about 30° and 11 dB.

A second example

$$L(s) = \frac{5.1}{(1 + 0.025s)^3}$$

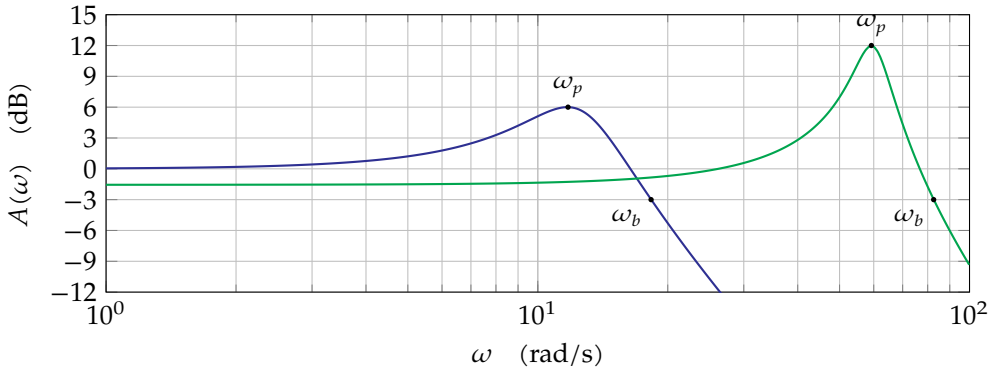
has a Bode plot as depictedin Figure 5.15. The corresponding Nichols plot has been depcited in Figure 5.17. It shows a much smaller phase and gain margin (16° and 4 dB) and the resonance is much higher.

As in a Hall chart, one can plot lines of constant overall magnitude (M-lines) and lines of constant overall phase (N-lines). This has been done in Figure 5.16 and Figure 5.17.

The main advantage of a Nichols plot over a Hall chart is that an increase in gain corresponds to shifting the plot vertically, which is much easier than scaling the plot in a Hall chart.

Remarks

- Note how in the first example, the overall gain peaks to 6 dB. In the second example the peak is even higher and amounts to 12 dB.
- Note the obvious relationship between the Nichols plots (see Figure 5.16 and Figure 5.17) and their overall Bode magnitude plots below:



The resonance peak for ω_p (touching an M-circle) and the bandwidths ω_b (frequency for which the −3 dB line is intersected) can be easily observed in both.

In MATLAB a Nichols plot can be conveniently generated as:

```
>> L tf( 5.1, [ 15.625e-6,  1.875e-3, 0.075, 1 ] );
>> nichols(L);
>> grid on;
```

Exercises

Exercise 5.5-1: (*) Derive the M-locus (i.e. the lines of constant overall magnitude) in a Nichols plot.

Exercise 5.5-2: (*) Derive the N-locus (i.e. the lines of constant overall phase) in a Nichols plot.

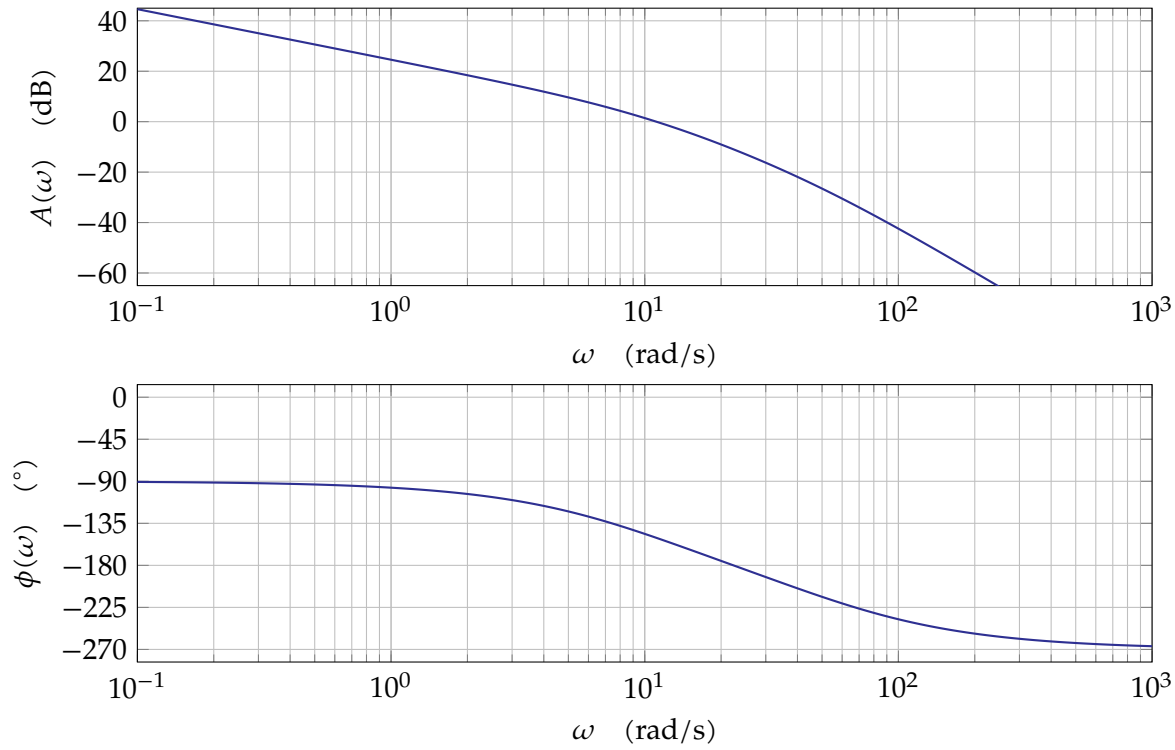


Figure 5.14: Bode plot of $L(s) = \frac{17}{s(1+0.1s)(1+0.02s)}$

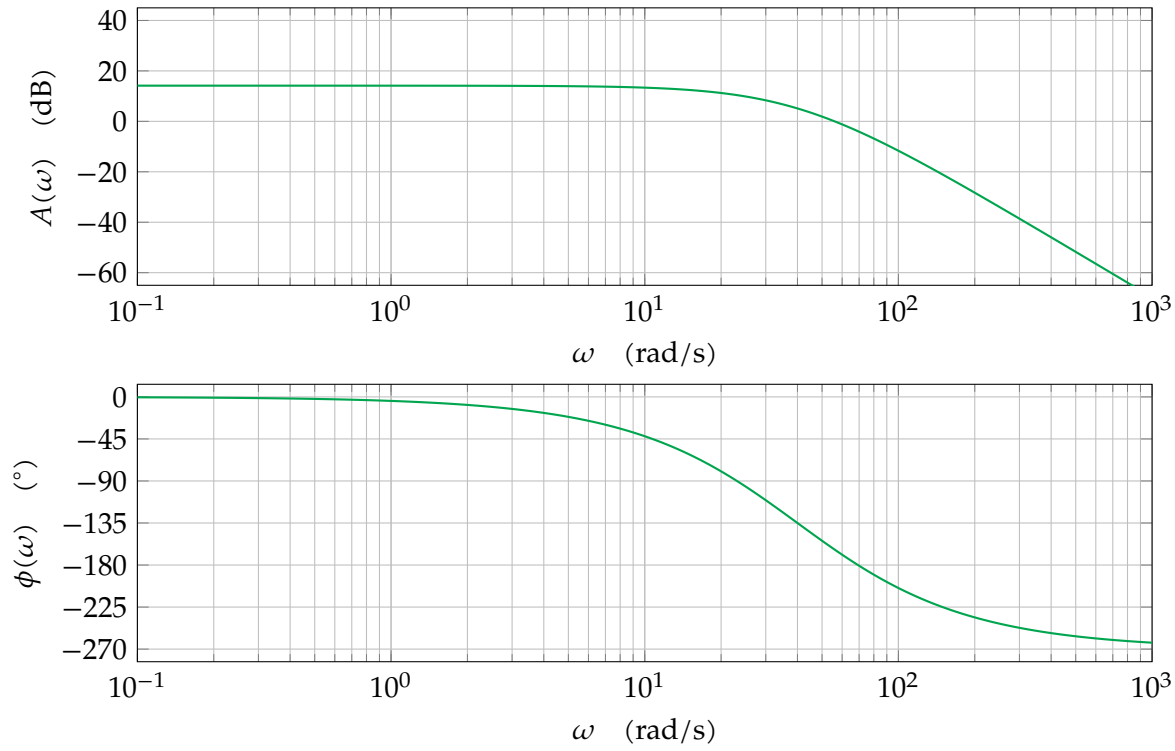


Figure 5.15: Bode plot of $L(s) = \frac{5.1}{(1+0.025s)^3}$

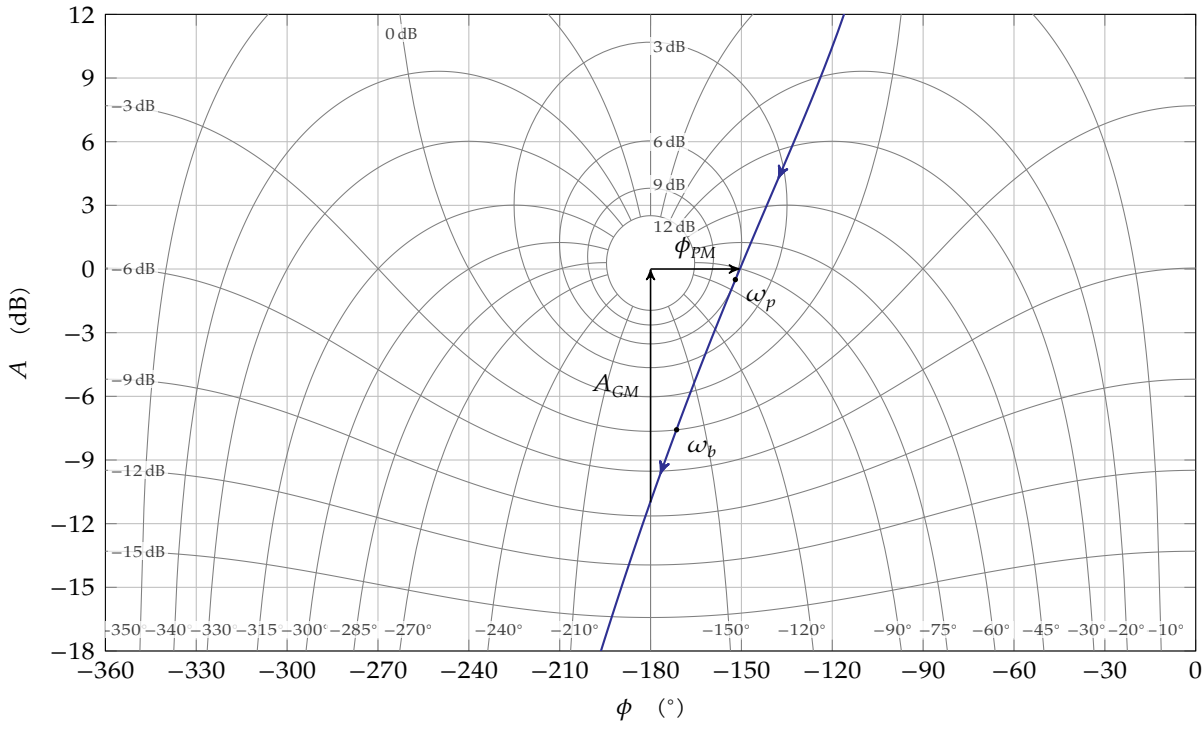


Figure 5.16: Nichols plot of $L(s) = \frac{17}{s(1+0.1s)(1+0.02s)}$ with M-lines and N-lines

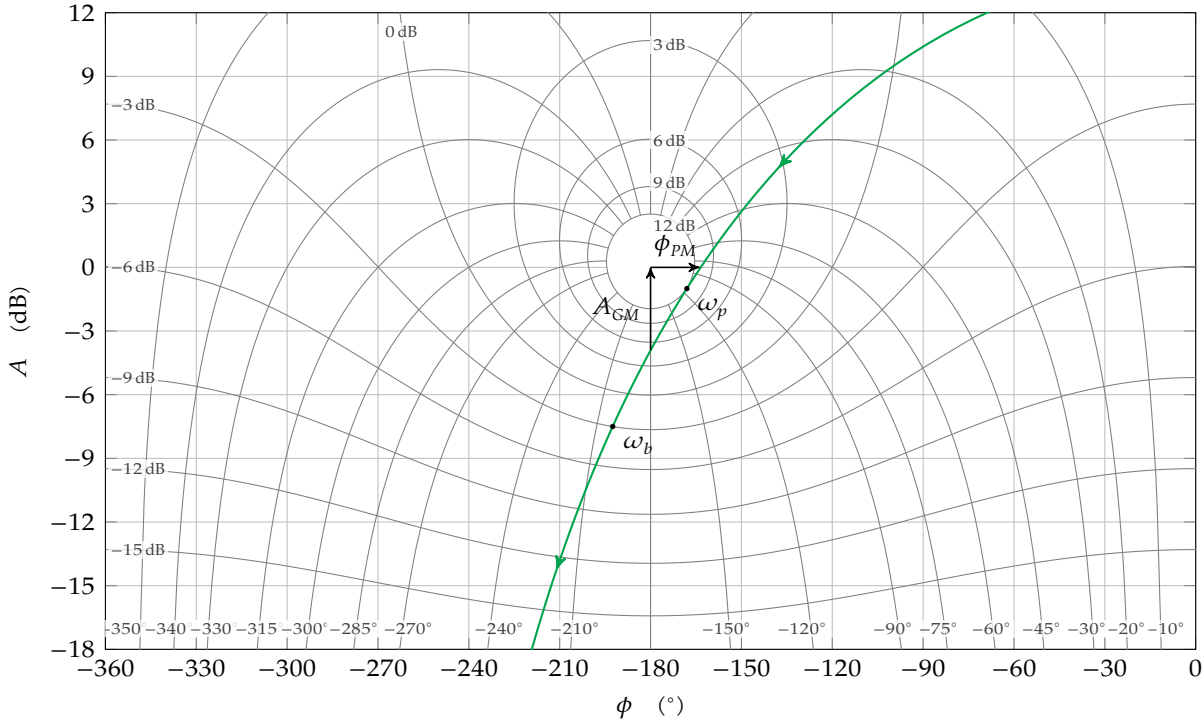


Figure 5.17: Nichols plot of $L(s) = \frac{5.1}{(1+0.025s)^3}$ with M-lines and N-lines

5.6 Systems with a time delay - a pain in the neck

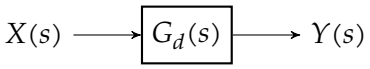
In many systems a time delay (a so-called dead time) is present. In this paragraph we investigate the influence of this time delay on the system's stability.

Problem statement

We know from Laplace theory that a time delay corresponds to multiplication with a complex exponential in the Laplace domain:

$$x(t-T) \quad \xrightarrow{\mathcal{L}} \quad X(s) \cdot \underbrace{e^{-sT}}_{\equiv G_d(s)}$$

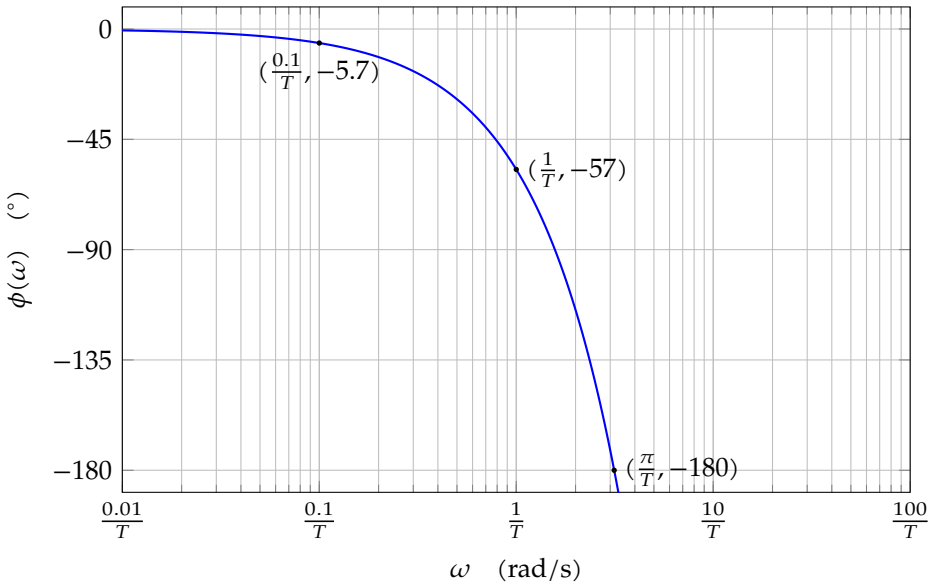
Therefore, the time delay can be modeled as:



Though this transfer function is no ratio of polynomials in s, its Bode plot is easily drawn, knowing that:

$$|G_d(j\omega)| = |e^{-j\omega T}| = 1 \qquad \text{and} \qquad \angle G_d(j\omega) = -\omega T$$

Therefore, this system keeps the modulus unaltered, but adds an extra phase delay (that is linear w.r.t. the frequency[5]). However, on a logarithmic frequency scale (which is common when drawing Bode plots), the effect is anything but linear and in fact quite brutal as can be observed below:



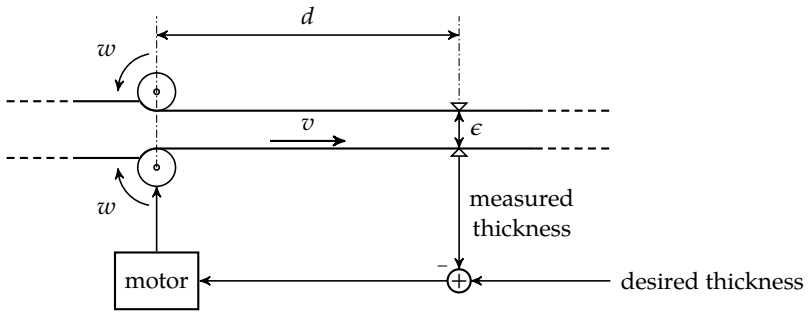
Why is this relevant? Even when the system contains a time delay, Nyquist's criterion is still valid (the time delay does not add any poles or zeros to the loop gain). Therefore, the stability of these systems is still governed by the phase margin.

[5]Remember that if you want to express $\angle G_d(j\omega))$ in degrees, you need a conversion factor $180°/\pi$.

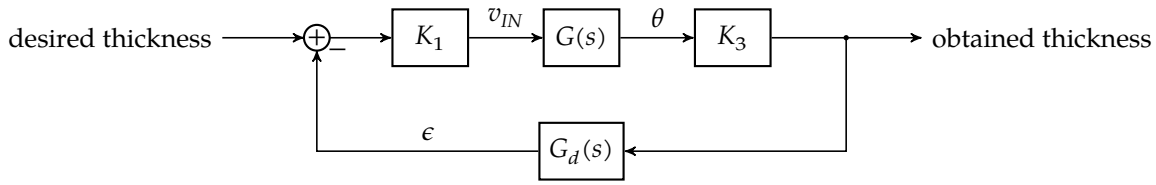
That makes it clear: the 'exponential' phase delay will be very detrimental for the phase margin of the system if the cross-over frequency has values $\omega_c \geq 0.1/T$, and therefore will significantly impair the stability of the system.

Example 1: a steel rolling mill

A small example illustrates a common origin of dead time. Stock steel is fed into a steel rolling mill to obtain sheet material that leaves the rolls at a speed v. The thickness of the obtained sheet is determined by the distance between the rolls. The thickness of the sheet material ϵ is measured a distance d downstream the feeding direction and is used to control a motor that corrects the distance between the rolls.



Measuring at a distance d introduces a delay $T = d/v$. This means that we will only be able to measure the result of an actuation of the rolls T seconds later. We can make this explicit in the following block diagram:



The desired thickness is compared to the measured thickness and then converted into a voltage v_{IN} by the gain block K_1. The voltage drives the motor controller converting it into a motor angular position θ. This angular position is related to the obtained thickness by a constant K_3. The obtained thickness is measured T seconds later and sent to the input for comparison to the desired thickness. In this block diagram $G(s)$ is the transfer function of the DC motor and $G_d(s)$ is the transfer function representing a time delay.

Let's assume that the transfer function of our DC motor equals:

$$G(s) = \frac{K_2}{s(s+1)}$$

The loop gain of the entire system therefore becomes:

$$L(s) = \frac{K_1 K_2 K_3}{s(s+1)} \cdot \mathrm{e}^{-sT}$$

Let's draw the Bode diagram of this system with and without time-delay, assuming:

$$K = K_1 K_2 K_3 = 2/\mathrm{s} \qquad v = 1\,\mathrm{m/s} \qquad d = 1\,\mathrm{m}$$

The result can been seen in Figure 5.18. Without time delay the phase margin is about 40°. With the time delay it becomes negative!

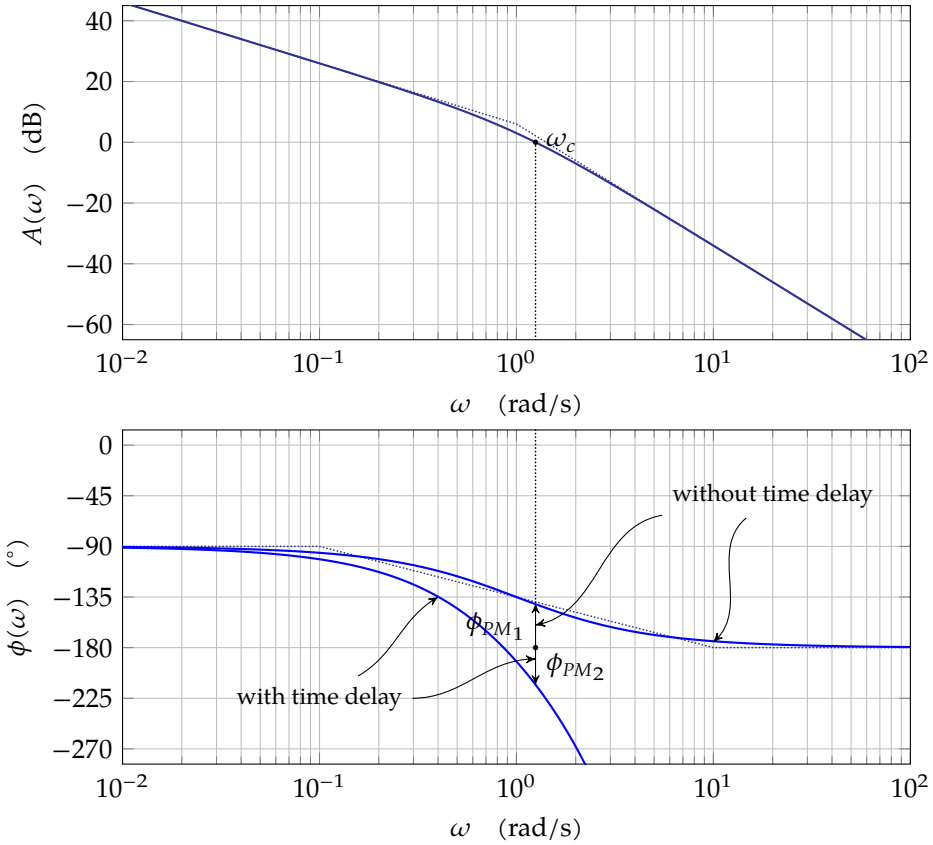


Figure 5.18: Bode plot of the steel rolling mill system, with and without time delay.

We can also compute ω_c from:

$$|L(j\omega_c)| = \frac{2}{\omega_c\sqrt{\omega_c^2+1}} = 1$$
$$\omega_c^4 + \omega_c^2 - 4 = 0 \quad \Rightarrow \quad \omega_c^2 = 1.5616 \quad \Rightarrow \quad \omega_c = 1.2496$$

This leads us to a phase margin of the system without time delay:

$$\phi_{PM_1} = 180° - 90° - \operatorname{atan}\frac{\omega_c}{1} = 38.668°$$

and a system with time delay:

$$\phi_{PM_2} = 180° - 90° - \operatorname{atan}\frac{\omega_c}{1} - \frac{180}{\pi}\omega_c = -32.930°$$

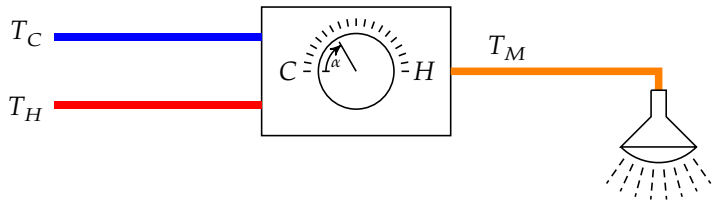
The conclusion is simple: because of the time delay the phase margin becomes negative corresponding to an unstable system. To make the system stable again, we have several options:

- Increase the speed of the rolls, i.e. increasing v; this reduces the delay.
- Reducing the distance d between the rolls and the measurement point; this also reduces the delay.

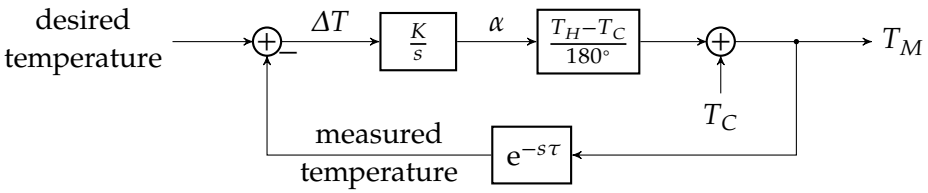
- Reducing the overall gain $K = K_1K_2K_3$, such that ω_c shifts to lower frequencies. This will keep ω_c away from the range where the phase delay kicks in. Note that this will make the system slower and more susceptible to parameter changes and disturbances.

Example 2: a mixer tap

Consider the system below, in which a mixer tap is actuated by a human being who — while showering — measures the water temperature sensed by its body and actuates the temperature by controlling the mixer knob of the tap. The tap is fed with cold water with temperature T_C and warm water with temperature T_H and produces water of temperature T_M based on the position α of the mixer tap. The time delay in between the water being mixed and hitting its body is τ.



Building a model for this system yields:



The mixer tap has been modeled based on the angle α made between the actual knob position and cold position (knob marker pointing to the left). Assuming $[\alpha] = {}^\circ$, this means:

$$T_M = T_C + \frac{T_H - T_C}{180^\circ}\alpha$$

This means:

$$\alpha = 0^\circ \quad \Rightarrow \quad T_M = T_C$$
$$\alpha = 180^\circ \quad \Rightarrow \quad T_M = T_H$$

We modeled the human actuation as setting the speed of actuation proportional to the observed temperature difference ΔT (i.e. the higher the difference, the faster the human will turn the knob):

$$\frac{\mathrm{d}\alpha}{\mathrm{d}t} = K\Delta T$$

which translates into the Laplace domain as:

$$\alpha(s) = \frac{K}{s}\Delta T$$

Obviously, if we take K too large, our human will be rotating the knob like a maniac and intuitively we can see the oscillation coming. Let's investigate what a proper value of K is.

Given:

$$L(s) = \frac{K}{s}\frac{T_H - T_C}{180^\circ} \cdot \mathrm{e}^{-s\tau}$$

We can calculate the cross-over frequency to be:

$$\begin{aligned} |L(j\omega_c)| = \frac{K}{|j\omega_c|}\frac{T_H - T_C}{180^\circ} \cdot \left|\mathrm{e}^{-j\omega\tau}\right| &= 1 \\ \frac{K}{\omega_c}\frac{T_H - T_C}{180^\circ} \cdot 1 &= 1 \\ \omega_c &= \frac{K(T_H - T_C)}{180^\circ} \end{aligned}$$

This yields the following phase margin:

$$\begin{aligned} \phi_{PM} &= 180^\circ + \angle L(j\omega_c) \\ &= 180^\circ - 90^\circ - \omega_c\tau\frac{180^\circ}{\pi} \\ &= 180^\circ - 90^\circ - \frac{K(T_H - T_C)}{180^\circ}\tau\frac{180^\circ}{\pi} \end{aligned}$$

From this equation, we can solve K to obtain an expression that allows us to find an appropriate gain value, depending on the desired phase margin:

$$\begin{aligned} 90^\circ - \phi_{PM} &= \frac{K(T_H - T_C)}{180^\circ}\tau\frac{180^\circ}{\pi} \\ K &= \frac{\pi}{\tau}\frac{90^\circ - \phi_{PM}}{T_H - T_C} \end{aligned} \tag{5.5}$$

To make drawing conclusions from our findings less abstract, we will use the following numerical values for the parameters of our problem:

$$T_C = 5\,^\circ\mathrm{C} \qquad T_H = 59\,^\circ\mathrm{C} \qquad \tau = 3\,\mathrm{s}$$

The out these values in (5.5), results in:

$$K = \frac{\pi}{3}\frac{90^\circ - \phi_{PM}}{54\,^\circ\mathrm{C}}$$

This equation has been graphed in Figure 5.19.

If we want a safe 45° phase margin, we need to allow a limited gain value of $K = 0.8$. This means that if our body senses a temperature difference of 10 °C, we only may turn the mixer tap at a rate of 8 °/s, which is very, very slow. Hopefully this sheds some light on why nowadays we are all using thermostatic mixer taps. At least it illustrates how systems with a dead time can be a true pain in the neck.

5.7 Conclusion

In this chapter, we have seen the main stability criterion in the frequency domain, the so-called Nyquist criterion: the polar plot of the loop gain may not circle −1. However, generating polar plots is not so straightforward. An alternative like the Nichols plot improves the situation. In addition, using the concept of phase margin on a Bode plot is for an electronics engineer probably the most straightforward way of tackling the problem.

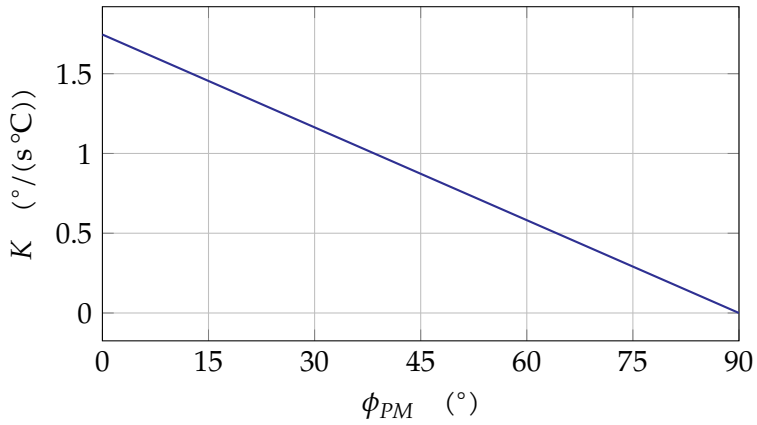
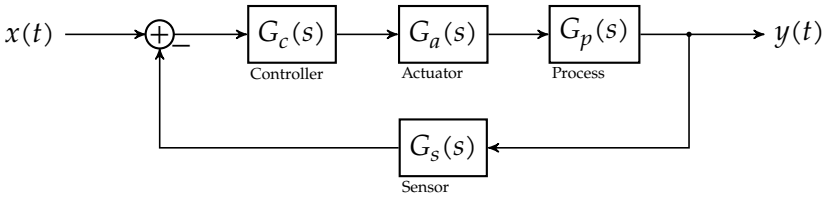


Figure 5.19: Allowable gain in the shower network as a function of the desired phase margin

Chapter 6

Analog control systems

A linear control system can be represented in general using the following block diagram:



The loop gain determines stability and therefore, we need to study:

$$G(s) = G_c(s) \underbrace{G_a(s)G_p(s)G_s(s)}_{\equiv G(s)}$$

So far, the controllers we used were proportional, i.e. they were a simple gain block, a so-called *P-controller*.

However, we can add additional poles and zeros in $G_c(s)$. This allows to further tailor the overall transfer function to the desired specifications.

The most simple template for such a $G_c(s)$ is a controller with a single zero $-z$ and a single pole $-p$, i.e.

$$G_c(s) = K \cdot \frac{s+z}{s+p} \tag{6.1}$$

Depending on whether the pole or zero comes first in the spectrum, we can discern:

- Phase lead controllers: zero before pole
- Phase lag controllers: pole before zero

The operational principle of these two controllers is fundamentally different. Both principles each give rise to a common industrial version of them: the PD-controller and the PI-controller. We will finalize the chapter with a combination of both: the well known *PID-controller*.

6.1 Single pole-zero phase lead controller

Let's start with the phase lead controller. We will study its characteristics subsequently in the frequency, the Laplace and the time domain. A good understanding of this controller will allow us to understand the operation of the more common PD-controller.

6.1.1 The normal form

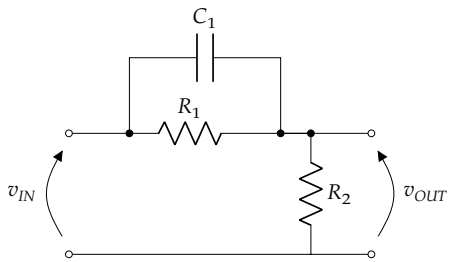
The transfer function of a single pole-zero phase lead controller can be easily derived in the so-called *normal form*, i.e. ensuring $p = \alpha z$ with $\alpha > 1$ and setting the high-frequency gain to 1. In this way, we obtain:

$$G_c(s) = \frac{s+z}{s+p} \qquad \text{with} \qquad z = \frac{1}{\alpha}p$$

One can call α the pole-zero distance factor (with $\alpha > 1$). Note that this is a special case of (6.1) with $z < p$ and $K = 1$.

6.1.2 Implementation example

Normal form A pole-zero phase lead controller can be implemented in many domains: mechanically, hydraulically, ...One way that interests us particularly is using electronic components. The following network does the job:



The transfer function is easily composed using Kirchhoff's laws and the branch equations of the elements, or by recognizing this circuit as a voltage divider:

$$\begin{aligned} G_c(s) = \frac{V_{OUT}(s)}{V_{IN}(s)} &= \frac{R_2}{R_2 + \frac{R_1}{1+R_1C_1s}} \\ &= \frac{R_2}{R_1+R_2} \cdot \frac{1+R_1C_1s}{1+\frac{R_1R_2}{R_1+R_2}C_1s} \\ &\Big\downarrow \frac{R_1R_2}{R_1+R_2}C_1 \equiv \tau \text{ and } \frac{R_1+R_2}{R_2} \equiv \alpha \\ &= \frac{1}{\alpha} \cdot \frac{1+\alpha\tau s}{1+\tau s} \\ &= \frac{s+\frac{1}{\alpha\tau}}{s+\frac{1}{\tau}} \end{aligned}$$

Non-normal form An extra gain factor K can be added by adding a simple op-amp amplifier in cascade, with gain:

$$K = 1 + \frac{R_4}{R_3}$$

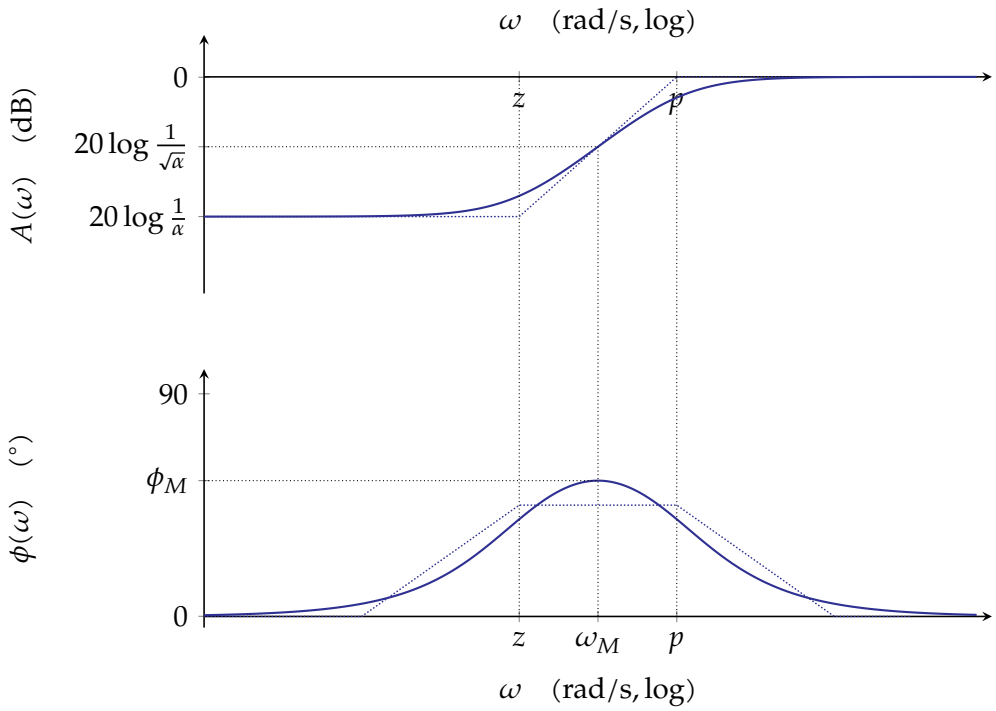
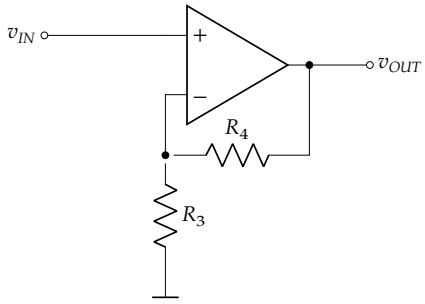


Figure 6.1: Bode plot of the single pole-zero phase lead controller



6.1.3 In the frequency domain

Let's analyze our controller in the frequency domain. Let's consider the normal form. Going to the non-normal form is as simple as adding an extra gain factor K. The Bode plot corresponding to this controller in normal form can be found in Figure 6.1.

We can clearly see that for low frequencies the gain is $1/\alpha$ and the phase is $0°$. Then the zero lifts the gain up, also increasing the phase (making it a leading phase), finally the pole stops the gain increase and brings the phase down again to $0°$.

The phase increase is maximal right in between the zero and the pole, i.e.

$$\omega_M = \sqrt{zp} = \sqrt{\alpha} z = \frac{p}{\sqrt{\alpha}}$$

At that point the gain has increased to

$$|G_c(j\omega_M)| = \frac{1}{\sqrt{\alpha}}$$

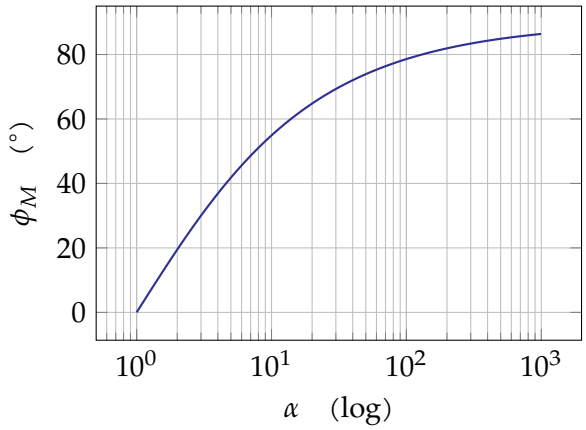
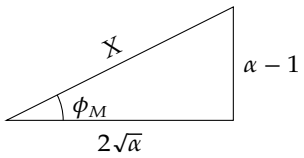


Figure 6.2: Maximal phase lead as a function of the pole-zero distance α

and the maximal phase deviation amounts to

$$\begin{aligned}
\phi_M &= \operatorname{atan}\frac{\omega_M}{z} - \operatorname{atan}\frac{\omega_M}{p} \\
&\downarrow \\
&= \operatorname{atan}\sqrt{\alpha} - \operatorname{atan}\frac{1}{\sqrt{\alpha}} \\
&\downarrow \tan \text{ of both sides} \\
\tan\phi_M &= \frac{\sqrt{\alpha} - \frac{1}{\sqrt{\alpha}}}{1 + \sqrt{\alpha}\frac{1}{\sqrt{\alpha}}} = \frac{\sqrt{\alpha} - \frac{1}{\sqrt{\alpha}}}{2} = \frac{\alpha - 1}{2\sqrt{\alpha}}
\end{aligned} \tag{6.2}$$

This relationship has been plotted in Figure 6.2. It is clear that the more the pole and zero are apart, the higher the maximal phase lead. However, it is unpractical to go above 80°. If a higher phase lead is required, one can cascade two phase lead controllers (effectively summing both phase characteristics creating a doubling of the effect). As our design strategy will later require, we need an align that expresses α as a function of ϕ_M. Solving (6.2) for α would be a viable approach, though cumbersome. We prefer a simple trigonometric insight to do simplify our job, and that is jotting down a rectangular triangle for which the tangent equals (6.2):



The hypotenuse X is easily determined as:

$$X = \sqrt{(\alpha - 1)^2 + (2\sqrt{\alpha})^2} = \cdots = \alpha + 1$$

A simpler align to determine α is now present as

$$\sin\phi_M = \frac{\alpha - 1}{\alpha + 1}$$

which can be easily solved for α, yielding:

$$\alpha = \frac{1 + \sin\phi_M}{1 - \sin\phi_M}$$

The key question remains: how are we going to use this controller? We summarize this in a design strategy:

Phase lead design strategy in the frequency domain

1. Select your desired
   - cross-over frequency ω_c
   - phase margin
2. Check the current phase margin at the desired ω_c and determine how much phase increase you need to obtain a decent phase margin. This determines α.
3. Put the pole p and zero z such that $\omega_M = \omega_c$ (the desired crossover frequency). This provides a maximal increase in phase margin.
4. Select a proper K to ensure the gain equals 1 for the desired ω_c.
5. Verify, adjust.

Some examples will make this design strategy clear.

Example 1: Design a single pole-zero phase lead controller for

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

such that the settling time $T_{s,2\%} < 0.5\,\text{s}$ and the overshoot to a step response is limited to 20%.

The system without controller It always makes sense to draw a Bode plot of the system without controller to be able to assess the situation. You can find it in Figure 6.3.

Determining an appropriate value of ω_c and a desired ϕ_{PM}
Many approach angles can be taken in determining appropriate values. We highlight just one possible route. The idea is to bluntly abuse the equations for the settling time and the overshoot of a second-order system without zeros as a (wrong but decent) first estimate.

From overshoot to ζ to ϕ_{PM} — we can translate the overshoot requirement into a proper ζ for a second-order system, using Figure 2.2 on page 28. The conclusion from that graph is that we need a damping $\zeta \geq 0.45$. Using the approximation of (5.4), this translates into a required phase margin of $\phi_{PM} \approx 45°$.

From settling time to $\zeta\omega_n$ to ω_c — for a second-order system without zeros we can approximate the settling time as:

$$T_{s,2\%} \approx \frac{4}{\zeta\omega_n}$$

Therefore, we need to ensure $\omega_n \approx 4/(\zeta T_{s,2\%}) = 17.8\,\text{rad/s}$.
Subsequently, equation (5.2) or reading figure Figure 5.10 on page 117 allows us to obtain a first estimate for $\omega_c \approx 0.825\omega_n = 14.7\,\text{rad/s}$.

Check the current phase margin at the desired ω_c
Given the phase characteristic of our double integrator system, this question is easily answered: the phase is stuck at $-180°$ and therefore the phase margin is $0°$. Conclusion: we need a phase boost of about $45°$ to obtain a proper phase margin.

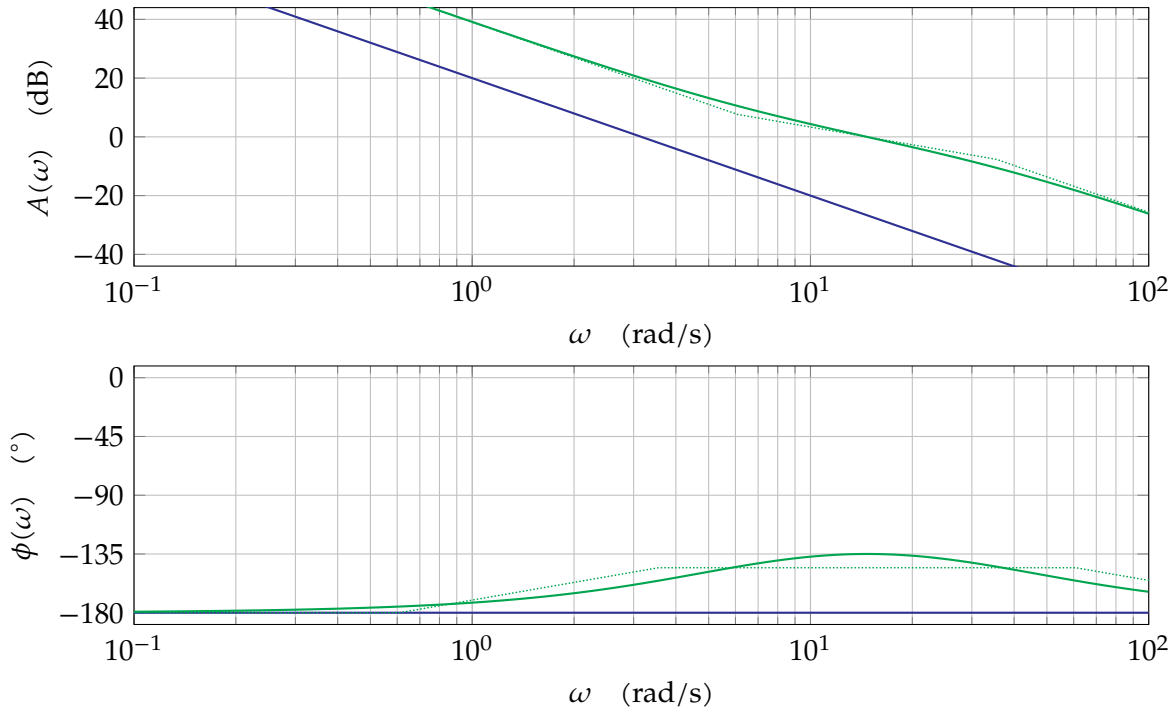


Figure 6.3: Bode plot of $\frac{10}{s^2}$, without controller (in blue), with phase lead controller (in green) with $\alpha = 5.83, z = 6.09\,\text{rad/s}, p = 35.5\,\text{rad/s}$ and $K = 52.1$.

This allows for calculating

$$\alpha = \frac{1 + \sin\phi_M}{1 - \sin\phi_M} = \frac{1 + \sin 45°}{1 - \sin 45°} = 5.83$$

Determine p and z

We need our phase boost at ω_c, i.e.

$$\omega_M = \omega_c = 14.7\,\text{rad/s}$$

and

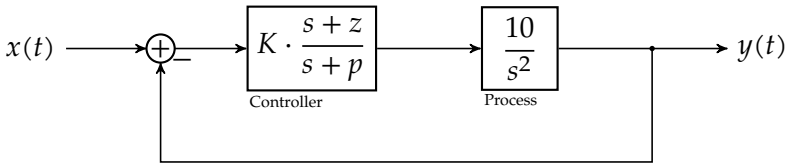
$$z = \omega_M/\sqrt{\alpha} = 6.09\,\text{rad/s}$$
$$p = \omega_M \cdot \sqrt{\alpha} = 35.5\,\text{rad/s}$$

Determine a proper K

We need to select K such that the overall gain is exactly 1 at ω_c. At ω_M ($= \omega_c$ in our case) the controller adds a gain of $1/\sqrt{\alpha}$. In view of this we can write the following modulus condition:

$$|K \cdot G_c(j\omega_c)G(j\omega_c)| = 1$$
$$K \cdot \underbrace{|G_c(j\omega_c)|}_{=1/\sqrt{\alpha}} \cdot |G(j\omega_c)| = 1$$
$$K \cdot \frac{1}{\sqrt{\alpha}} \cdot \frac{10}{\omega_c^2} = 1$$
$$K = \sqrt{\alpha}\frac{\omega_c^2}{10} = 52.1$$

The resulting feedback system can be found below:



The corresponding loop gain Bode-plot can be found in green in Figure 6.3.

Verify, adjust

Let's check the overall transfer function:

$$T(s) = \frac{K \cdot 10 \cdot (s+z)}{s^2(s+p) + K \cdot 10 \cdot (s+z)} = \frac{10K(s+z)}{s^3 + ps^2 + 10Ks + 10Kz} = \frac{521(s+6.09)}{s^3 + 35.5s^2 + 521s + 3173}$$

This transfer function has a zero $z = -6.09$ and three poles:

$$p_1 = -14.8$$
$$p_{2,3} = -10.4 \pm j10.4$$

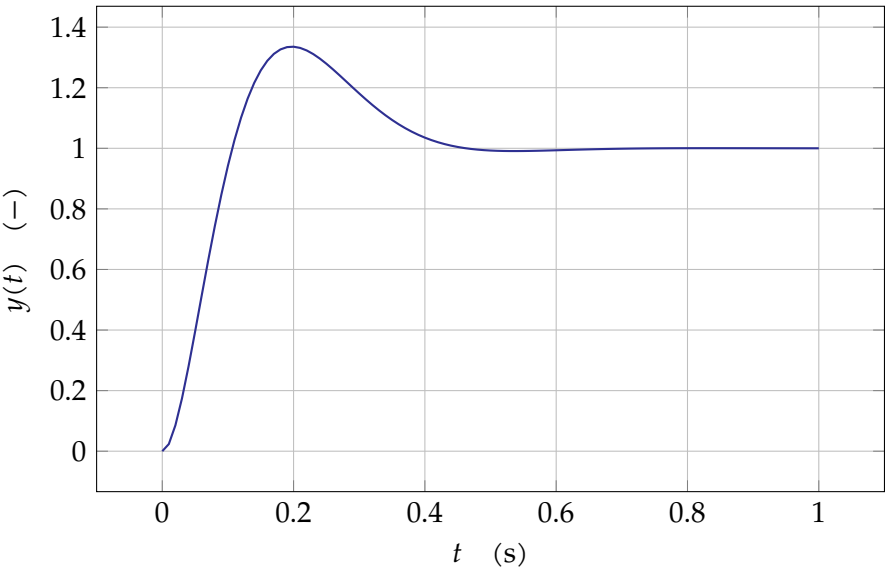
From this, we can deduce:

$$\omega_n = \sqrt{(10.4)^2 + (10.4)^2} = 14.7\,\text{rad/s}$$
$$\zeta = \frac{-10.4}{\omega_n} = 0.707$$

Note that these values are quite different from what we intended. Further tweaking might be needed to improve the situation. If we simulate this system using a step input, e.g. using MATLAB:

```
T = tf( 521*[1 6.09], [ 1 35.5 521 3173] );
[y,t] = step(T);
```

we obtain:



From this simulation, we can deduce the settling time $T_{s,2\%}$ using the following MATLAB code that continues after calculating the step response.

```
settlingindex = 0;
for i = length(y):-1:1
  if abs(1-y(i)) > 0.02
    settlingindex = i + 1;
    break;
  end
end
Ts = t(settlingindex);
```

Likewise, it is easy to determine the overshoot:

```
maxdev = 0;
for i = 1:length(y)
  dev = y(i) - 1;
  if dev  > maxdev
    maxdev = dev;
  end
end
O = maxdev;
```

The result turns out to be $T_{s,2\%} = 0.42\,\text{s}$ which is a little bit better than our target value, and an overshoot of 33.5% which is clearly outside of our specification. The culprit is the zero. We will come back to this later. Clearly a redesign is needed. However, we will stop the example here.

Example 2: Determine the minimal achievable settling time for the following system using a proportional controller:

$$G(s) = \frac{1}{s(s+2)(s+10)}$$

Try to improve this specification using a single pole-zero phase lead controller.

The system without controller Again, it makes sense to assess the situation by drawing a Bode plot of the system without controller. You can find it in Figure 6.4. Just looking at the plot teaches us that in order to obtain a marginally stable system (i.e. $\phi_{PM} = 0$ and therefore $\angle G(j\omega) = -180°$) we can add a gain of about 45 dB.

We can verify this numerically:

$$\angle G(j\omega) = -90° - \operatorname{atan}\frac{\omega}{2} - \operatorname{atan}\frac{\omega}{10}$$

From this align, we can derive the frequency for which the phase becomes $-180°$ (using our — by now — classical trick using the tangent):

$$\begin{aligned} -180° &= -90° - \operatorname{atan}\frac{\omega_{180}}{2} - \operatorname{atan}\frac{\omega_{180}}{10} \\ \tan(180° - 90°) &= \tan\left(\operatorname{atan}\frac{\omega_{180}}{2} + \operatorname{atan}\frac{\omega_{180}}{10}\right) \\ \tan 90° &= \frac{\frac{\omega_{180}}{2} + \frac{\omega_{180}}{10}}{1 - \frac{\omega_{180}}{2}\frac{\omega_{180}}{10}} \end{aligned}$$

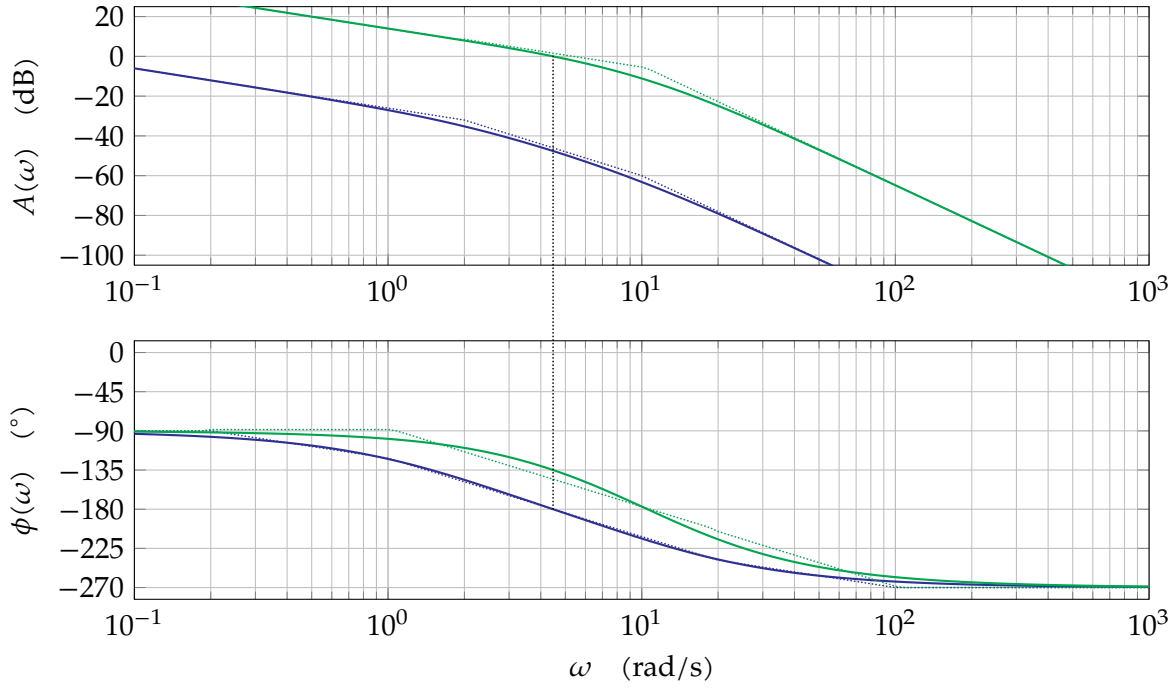


Figure 6.4: Bode plot of $\frac{1}{s(s+2)(s+10)}$, without controller (in blue), with phase lead controller (in green) with $\alpha = 5.83, z = 1.85\,\text{rad/s}, p = 10.80\,\text{rad/s}$ and $K = 579.4$

The left-hand side of the align is infinite. This is only possible if the denominator of the right-hand side becomes zero, i.e.

$$1 - \frac{\omega_{180}}{2}\frac{\omega_{180}}{10} = 0$$
$$\omega_{180} = \sqrt{20}$$

This corresponds to our findings in the Bode plot. The magnitude of the system without controller at ω_{180} amounts to:

$$|G(j\omega_{180}| = \frac{1}{\sqrt{20}\sqrt{20+4}\sqrt{20+100}} = 4.1667 \times 10^{-3}$$

allowing a proportional controller with a gain of $K = \sqrt{20}\sqrt{20+4}\sqrt{20+100} = 240$ or 47.604 dB before the system becomes unstable.

A fundamental problem with settling time is that it can only be calculated analytically for the most simple problems (e.g. second order without zeros). In more complex cases, it pays off to simulate it, e.g. in MATLAB using:

```
K = 10;
G = tf( K, [ 1 12 20 0 ] );
T = feedback( G, 1 );
[y,t] = step( T );
```

Performing this simulation for multiple values of K, allows composing Table 6.1. A first observation is that the settling time does not decrease monotonically with K. However, in many cases the following rule of thumb holds:

Design guideline Taking $PM = 45°$ is a common choice in order to obtain a good control system.

K (−)	ω_c (rad/s)	ϕ_{PM} (°)	$T_{s,2\%}$ (s)	O (%)
5	0.25	81	13.6	0
10	0.49	73	5.3	0
20	0.91	60	4.4	8
30	1.26	50	4.9	17
40	1.56	43	4.4	25
50	1.82	37	5.2	32
100	2.80	20	7.7	57
200	4.08	4	30	85

Table 6.1: Simulation results for proportionally controlled system $\frac{1}{s(s+2)(s+10)}$ for various values of K

Now, let's take it one step further and add a single pole-zero phase lead controller.

Determining an appropriate value of ω_c and a desired ϕ_{PM}
Let's pick $\omega_c = \omega_{180}$ of the system without controller and aim for our rule of thumb, i.e. $\phi_{PM} = 45°$.

Check the current phase margin at the desired ω_c
The phase margin for the system without controller was 0°, so we need an extra 45° phase lead from our controller, i.e.:

$$\alpha = \frac{1+\sin\phi_M}{1-\sin\phi_M} = \frac{1+\sin 45°}{1-\sin 45°} = 5.83$$

Determine p and z
We need our phase boost at ω_c, i.e.

$$\omega_M = \omega_c = \omega_{180} = 4.47\,\text{rad/s}$$

and

$$z = \omega_M/\sqrt{\alpha} = 1.85\,\text{rad/s}$$
$$p = \omega_M \cdot \sqrt{\alpha} = 10.80\,\text{rad/s}$$

Determine a proper K
At ω_M, the gain of the controller is $1/\sqrt{\alpha}$. Therefore we can solve the following modulus condition to make the gain equal to 1 at that frequency:

$$|KG_c(j\omega_c)G(j\omega_c)| = 1$$
$$K\underbrace{|G_c(j\omega_c)|}_{=1/\sqrt{\alpha}}|G(j\omega_c)| = 1$$
$$K\frac{1}{\sqrt{\alpha}}\frac{1}{\omega_c\sqrt{\omega_c^2+4}\sqrt{\omega_c^2+100}} = 1$$

And therefore:

$$K = \sqrt{\alpha}\sqrt{20}\sqrt{20+4}\sqrt{20+100} = 579.4$$

K (−)	ω_c (rad/s)	ϕ_{PM} (°)	$T_{s,2\%}$ (s)	O (%)
100	0.9	82	4.0	0
200	1.7	73	1.6	0
300	2.6	64	1.3	4
400	3.3	56	1.6	11
576	4.5	45	1.6	24
1000	6.5	26	2.3	47

Table 6.2: Simulation results for a phase lead controlled system $\frac{1}{s(s+2)(s+10)}$ for various values of K

Verify, adjust

Again, we can verify our result by simulation. This yields the values of Table 6.2. Obviously a K that is a little bit smaller than the one we originally have chosen yields an even better settling time. Not that the improvement over the proportionally controlled case is significant (about a factor of three).

Conclusion: using a phase lead controller, we can significantly speed up the system. At the same time, it allows for a higher loop gain, reducing the influence of parameter changes and disturbance signals. Analyzing the controller in the frequency domain, clearly has shown the positive effect of the controller: it improves the phase margin by its phase lead effect. However, the analysis in the frequency domain does not show the effect on the location of the poles and zeros of the overall system. To this end, we will look at a design procedure in the Laplace domain.

6.1.4 In the Laplace domain

The disadvantage of designing a single pole-zero phase lead controller in the frequency domain, is that you don't have any control over where the overall poles and zeros will end up. Designing directly in the Laplace domain is a better approach in this view. The following design strategy does the job:

Phase lead design strategy in the Laplace domain

1. Choose the location of the dominant poles of the overall system. This location is determined by the desired settling time and damping factor.
2. Position the zero of the controller wisely. Rule of thumb: $z = 4/T_s$.
3. Determine the location of the pole of the controller such that the root locus of the compensated system visits the chosen location of the dominant poles. Use the phase condition for this.
4. Determine the proper gain K that corresponds to the location of the dominant poles. Use the magnitude condition for this.

Let's illustrate this design strategy and the philosophy behind it using the first example of previous section.

Example: Design a single pole-zero phase lead controller for

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

such that the settling time $T_{s,2\%} < 0.5\,\text{s}$ and the overshoot to a step response is limited to 20%.

Select desired location of the dominant poles

We have in the previous section before, that the settling time requirement corresponds to $\zeta = 0.45$. Similarly we have shown in the previous section that the desired $\omega_n = 17.8\,\text{rad/s}$.

A second-order dominant pole pair can be written in terms of ω_n and ζ. When $0 \leq \zeta \leq 1$, the following holds:

$$p_{a,b} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$$

Given the values of ζ and ω_n this translates into:

$$p_{a,b} \approx -8 \pm j15.9$$

This is our desired location for the dominant poles.

Position the zero of the controller

If we consider the root locus of the system with only proportional control (see Figure 6.5), we can see the fundamental problem of this controller: the locus is on the imaginary axis and therefore, this system will oscillate.

The fundamental idea is that we can move the origin of the asymptotes $(\sigma_A, 0)$ to the left without changing the number of asymptotes by adding a zero and a pole.
Indeed, we know that:

$$\sigma_A = \frac{\sum_{i=1}^{n_p} p_i - \sum_{i=1}^{n_z} z_i}{n_p - n_z} \tag{6.3}$$

So, if we put in an extra pole that is more negative than the extra zero we can shift the origin of the asymptotes to the left, away from the unstable imaginary axis. As it is likely that there will be a locus in between the extra zero and pole (corresponding to the location of an overall pole), it makes sense, to put them into the left half plane. To make sure this extra overall pole does not become dominant we use the rule of thumb $z = 4/T_s = 8$ (i.e. the zero is at s=-8).

The value of this zero will contribute adversely to (6.3) and therefore, we put the extra pole far enough to the left to provide a shift of σ_A such that it ends up to the left of $\sigma = -8$. The effect has been illustrated in Figure 6.6.

Determine the location of the pole of the controller

Now let's calculate where exactly we need to put the pole. To this end, we use the phase condition, i.e. for the desired pole location $(s = -8 \pm j15.9)$ the following should hold:

$$\angle\left(\frac{s+z}{s+p}\frac{10}{s^2}\right) = 180^\circ + m \cdot 360^\circ, \text{ with } m \in \mathbb{Z}$$

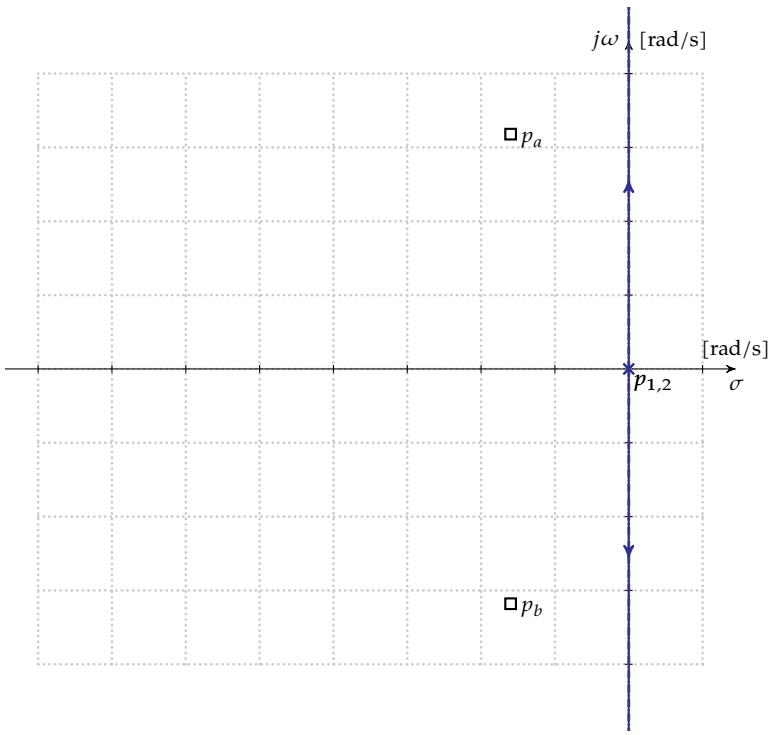


Figure 6.5: Root locus of a feedback system with loop gain $G(s) = \frac{10}{s^2}$ showing a clear oscillation for every $K \geq 0$. The desired location of the dominant poles $p_{a,b}$ has been indicated.

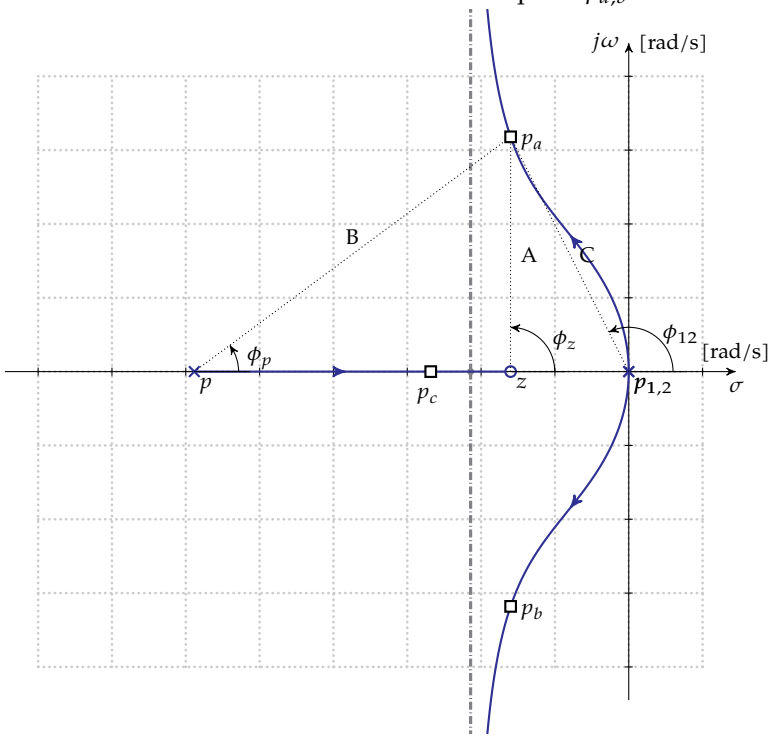


Figure 6.6: Root locus of a feedback system with loop gain $G(s) = \frac{s+8}{s+29.4}\frac{10}{s^2}$. The desired location of the dominant poles $p_{a,b}$ has been indicated.

Considering the indicated angles on Figure 6.6, we can rewrite this as:

$$\begin{aligned}
\phi_z - \phi_p - 2\phi_{12} &= 180^\circ + m \cdot 360^\circ \\
\phi_p &= \phi_z - 2\phi_{12} - 180^\circ - m \cdot 360^\circ \\
&= 90^\circ - 2\left(180^\circ - \operatorname{atan}\frac{15.9}{8}\right) - 180^\circ - m \cdot 360^\circ \\
&= 90^\circ - 2 \cdot 116.71^\circ - 180^\circ - m \cdot 360^\circ \\
&\downarrow m = -1 \\
&= 36.582^\circ
\end{aligned}$$

Applying a bit of trigonometry results in:

$$|p_c| = 8 + \frac{15.9}{\tan \phi_p} = 29.423$$

The angles have been indicated on Figure 6.6.

Determine the proper gain K of the controller

To this end, we can use the magnitude condition:

$$\left| K \cdot \frac{s+z}{s+p} \cdot \frac{10}{s^2} \right|_{s=j\omega} = 1$$

for $s = -8 \pm j15.9$.

Considering the indicated lengths on Figure 6.6, we can rewrite this condition as:

$$K \cdot \frac{A}{B} \cdot \frac{10}{C^2} = 1$$

Solving for K and calculating the lengths A, B and C yields:

$$K = \frac{B \cdot C^2}{10 \cdot A} = \frac{\sqrt{21.423^2 + 15.9^2} \cdot (8^2 + 15.9^2)}{10 \cdot 15.9} = 53.158$$

One question remains: was the choice of $z = 4/T_s$ a good one? To this end, we performed the same design procedure for different values of z and simulated a step response of the resulting system to determine the settling time and the overshoot. You can find the result below:

z (rad/s)	p (rad/s)	α (−)	K (−)	ω_c (rad/s)	ϕ_{PM} (°)	T_s (s)	O (%)
0.0	16.0	∞	31.7	14.6	47.6	0.47	21
1.0	17.1	17.1	33.4	14.8	45.2	1.00	26
2.0	18.2	9.1	35.2	15.0	42.9	0.78	31
4.0	21.0	5.3	39.7	15.6	39.0	0.64	39
8.0	29.4	3.7	53.2	17.2	34.7	0.47	46
12.0	46.5	3.9	80.4	19.0	35.5	0.45	45
16.0	99.4	6.2	165.1	20.6	40.4	0.44	39
18.0	213.9	11.9	348.4	21.2	44.1	0.43	36
19.8	627299.8	31681.8	1003685.8	21.7	47.6	0.43	33

We can draw a number of conclusions out of this table:

- both settling time and overshoot keep improving for zeroes that are higher in frequency; however, note the overshoot being out of spec! We need to investigate this problem in the time-domain.
- for a value of z that is a little bit higher than 19.8 the pole will move to infinity. This situation corresponds to a PD-controller that we will see later.

$$\lim_{p\to+\infty} K\cdot\frac{s+z}{s+p}\cdot\frac{10}{s^2} = \lim_{p\to+\infty} \underbrace{\frac{K}{p}\cdot(s+z)}_{\text{PD-controller}}\cdot\frac{10}{s^2}$$

The remaining limit in the expression will evolve to a finite value.

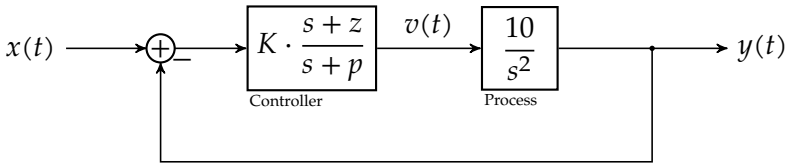
6.1.5 In the time domain

The previous two sections each contained a design procedure for our single pole-zero phase lead controller. This section will not yield a new design procedure but will prevent us from making stupid mistakes when it comes to the time-domain behavior of the controlled system.

A significant problem that remained unattended so far, is that the zero of the controller is also a (low-frequency) zero of the overall system and therefore will create a large overshoot. This was visible in the previous example.

In addition to this, the actuation signal level at the output of the controller will become very large and may lead to clipping, even further deteriorating the response speed of the system. Let's investigate this.

Our example system looks like this:



The overall transfer function of this system is easily calculated to be:

$$T(s) = \frac{10K(s+z)}{s^2(s+p)+10K(s+z)} = \frac{10K(s+z)}{(s-p_a)(s-p_b)(s-p_c)}$$

with p_a, p_b and p_c the three poles as indicated in Figure 6.6. The zero of our controller will deteriorate the overshoot, the pole of the controller will counteract this effect a little bit.

If we calculate the transfer function of the output of the controller (the actuation signal) w.r.t. the input, we obtain:

$$\frac{V(s)}{X(s)} = \frac{Ks^2(s+z)}{s^2(s+p)+10K(s+z)} = \frac{Ks^2(s+z)}{(s-p_a)(s-p_b)(s-p_c)}$$

This transfer function has three zeros, resulting in an extra big overshoot when we apply a step

input. The initial value theorem makes this clear:

$$v(0) = \lim_{s\to\infty} s \cdot \frac{V(s)}{X(s)} \cdot \underbrace{\frac{1}{s}}_{\text{step input}}$$
$$= \lim_{s\to\infty} \frac{Ks^2(s+z)}{(s-p_a)(s-p_b)(s-p_c)} = K$$

As K can become quite high (e.g. with $z = 8, p = 29.4$, we obtained $K = 53.2$), the actuation signal may harm the process or more likely, clipping will occur. In itself clipping is no major problem, but it will further deteriorate the speed of our system.

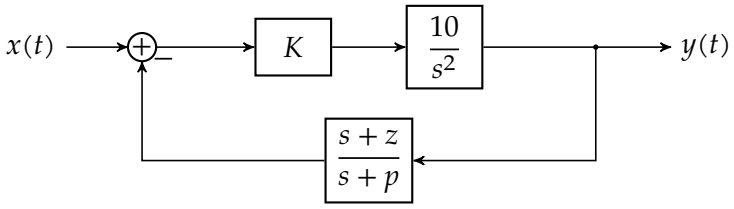
Can we overcome this problem? Yes. We have two options:

- Put the controller in the feedback path
- Add a prefilter

Let's go into detail.

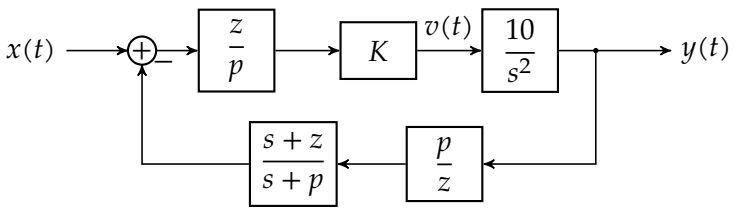
Put the controller in the feedback path

If we put the controller in the feedback path, we obtain the following system.



Note that we keep the gain in the forward path to diminish the influence of parameter variations and disturbance signals.

Performing this shift results in an overall DC gain that is z/p times smaller than with the controller in the forward path. We compensate for this, by adding a correction factor in the forward path ánd in the feedback path, the latter to make sure the DC loop gain is unchanged.



In this way, the overall transfer function becomes:

$$T(s) = \frac{10K\frac{z}{p}(s+p)}{s^2(s+p) + 10K(s+z)}$$

The result is that now that the controller's pole has become the zero. Because it is α times higher in frequency, this will improve the overshoot significantly.

Indeed, if we design the controller using the same values we used earlier ($z = 8, p = 29.4$ and $K = 53.2$) and simulate, we obtain an overshoot of only 8.6%.

The actuation signal as a response to a step input becomes:

$$V(s) = \frac{K\frac{z}{p}s^2(s+p)}{s^2(s+p) + 10K(s+z)}\frac{1}{s}$$

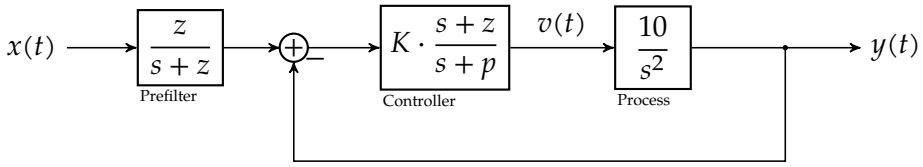
Using the initial value theorem, we can calculate the initial peak:

$$v(0) = \lim_{s\to\infty} \frac{K\frac{z}{p}s^2(s+p)}{s^2(s+p) + 10K(s+z)}\frac{1}{s} = K\frac{z}{p}$$

i.e. the value is a factor α smaller and (given $\alpha = 3.7$) equals $v(0) = 14.4$, a significant improvement! This will reduce the risk of clipping and therefore will be beneficial for settling time as well.

Add a prefilter

An alternative is to use a prefilter. This results in the following block diagram:



The pole of the prefilter equals the zero of the controller, such that they cancel each other in the overall transfer function of the system. Indeed:

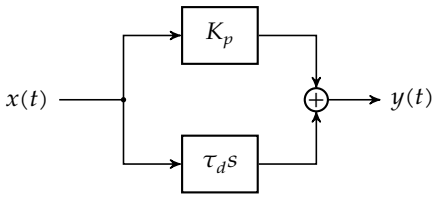
$$T(s) = \frac{z}{s+z}\frac{10K(s+z)}{s^2(s+p) + 10K(s+z)} = \frac{Kz}{(s-p_a)(s-p_b)(s-p_c)}$$

Simulating this transfer function, yields a settling time $T_{s,2\%} = 0.476\,\text{s}$ and an overshoot $O = 6.25\%$. This is a comparable result to using the controller in the feedback loop. However, it is more expensive due to the additional block, may result in poor performance if due to parameter variation the pole of the prefilter loses track of the zero of the controller, and doesn't offer the beneficial reduced clipping risk.

6.2 The PD-controller

6.2.1 The normal form

The PD-controller is the sum of a P-controller (a gain block with gain K_p) and a D-controller (a differentiator with gain $\tau_d s$):



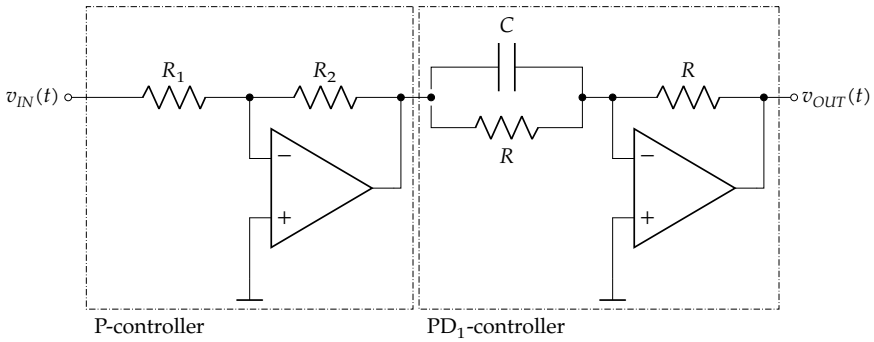
The overall transfer function of the controller is:

$$G_c(s) = K_p + \tau_d s = K_p\left(1 + \frac{s}{z}\right) = K_p\frac{s+z}{z} \qquad \text{with } z = \frac{K_p}{\tau_d}$$

Note that this is a special case of the single pole-zero phase lead controller. This one has no pole (you can consider the pole to occur at infinity).

6.2.2 Implementation example

In practice, a PD-controller is most often realized as a cascade of a P-controller and a PD_1-controller (i.e. a PD-controller with gain 1). The following schematic does the job:



The P-controller offers a negative gain $G_p = -R_2/R_1$. The PD_1-controller corrects the sign by inverting itself with a gain $G_{PD_1} = -(1 + RCs)$. The overall transfer function becomes:

$$H(s) = \frac{R_2}{R_1}(1 + RCs) = \underbrace{\frac{R_2}{R_1}}_{\equiv K_p}\left(1 + \frac{s}{z}\right) \qquad \text{with } z = \frac{1}{RC}$$

One can make the gain adjustable by making R_2 a variable resistor. One can make the zero (independently) adjustable by taking a variable capacitor or a dual pot-meter for R.

6.2.3 In the frequency domain

The Bode plot of the PD-controller can be found in Figure 6.7. For low frequencies, the gain equals a constant K_p, for high frequencies it evolves as $\tau_d\omega$. The zero occurs at the intersection of those two lines (on a logarithmic scale), i.e. $K_p = \frac{1}{\tau_i\omega}$.

Let's find out how to use the PD-controller from the Fourier domain's perspective using a number of examples.

Example: Design a PD-controller for a process given by:

$$G(s) = \frac{K_G}{(1 + \tau_1 s)(1 + \tau_2 s)(1 + \tau_3 s)} \tag{6.4}$$

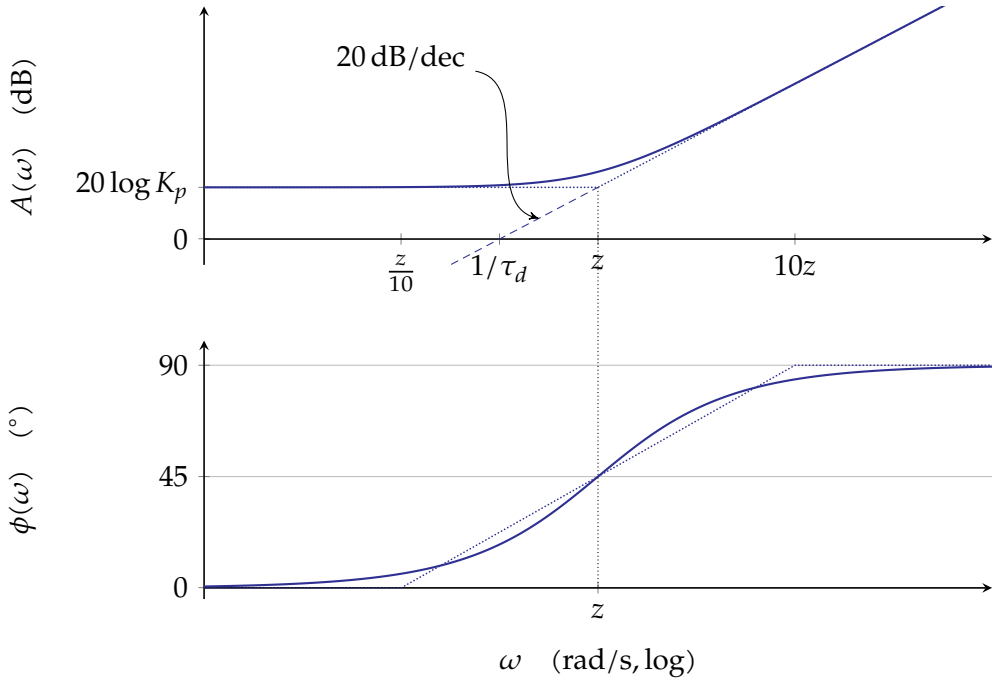


Figure 6.7: Bode plot of the PD-controller

assuming $\tau_1 \gg \tau_2 \gg \tau_3$. The design goal is to maximize ω_c while keeping a decent phase margin $\phi_{PM} = 45°$.

As the poles are far apart, we may assume the phase to be exactly $-135°$ at $\omega = 1/\tau_2$. A DC-gain K to obtain $\omega_c = 1/\tau_2$ equals $K = \tau_1/\tau_2$. The Bode plot corresponding to this situation can be found in Figure 6.8. This would be the optimal situation when using a proportional controller with $K_p = K/K_G$.

What benefit can we get from a PD-controller? The key question is where to put the zero of the PD-controller. We will try three options and calculate for each what a proper value of K_p would be. The corresponding Bode plots can be found in Figure 6.9. The original Bode plot has been indicated in blue dashed line for your convenience. The new plots (including the PD-controller) have been indicated in green).

- $z = 1/\tau_1$: The zero of the controller will neutralize the first pole of the process. Therefore, we will only reach a $-135°$ phase for $\omega = 1/\tau_3$. To obtain $\omega_c = 1/\tau_3$ we need to set the overall gain to $K = \tau_2/\tau_3$ i.e. $K_p = K/K_g$. If the ratio τ_3/τ_2 is bigger than τ_2/τ_1 this is an improvement. If not, there is no benefit in setting the controller's zero at the first pole. In any case, the benefit (if any) is marginal. This situation has been sketched in Figure 6.9a. We conclude that this is not a good option.
- $z = 1/\tau_2$: The zero of the controller will neutralize the second pole of the process. Therefore, again, we will only reach a $-135°$ phase for $\omega = 1/\tau_3$. To obtain $\omega_c = 1/\tau_3$, we need to set the overall gain to $K = \tau_1/\tau_3$, i.e. $K_p = K/K_g$. Note that this allows setting much higher low-frequency gain as in the first option. This situation has been sketched in Figure 6.9b. We conclude that this is a good option.
- $z = 1/\tau_3$: The zero of the controller will neutralize the third pole of the process. However, this does not improve the situation w.r.t. case with the proportional controller. The phase of $-135°$ is still achieved at $\omega = 1/\tau_2$ and the appropriate gain

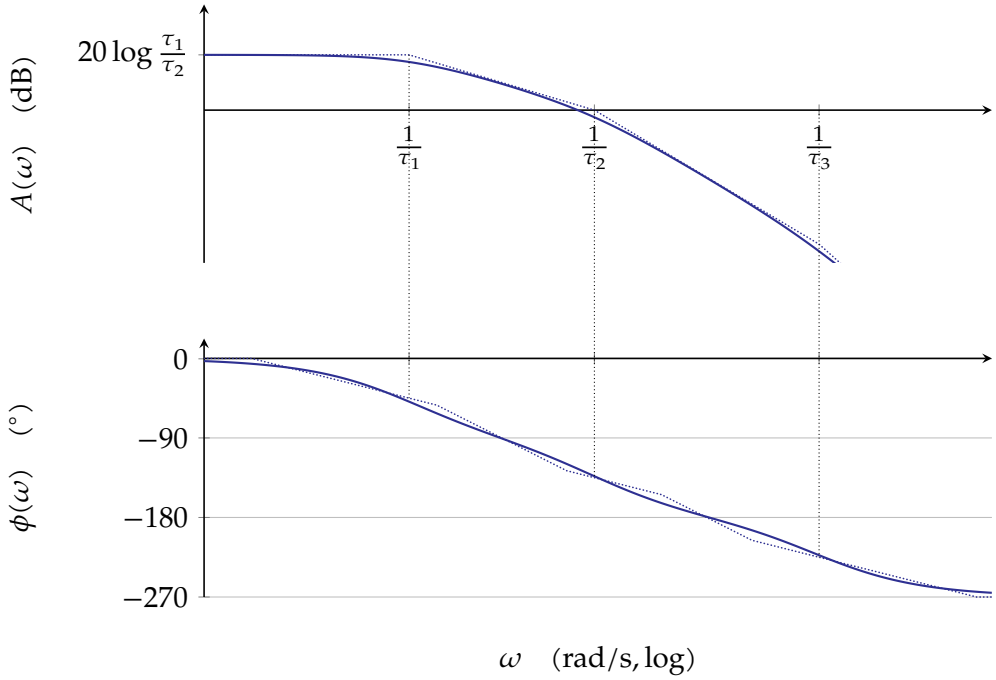


Figure 6.8: Bode plot of the example of (6.4)

is τ_1/τ_2. This situation has been sketched in Figure 6.9c. We conclude that this is not a good option.

From this analysis, we derive the following design strategy:

PD-controller design strategy in the frequency domain

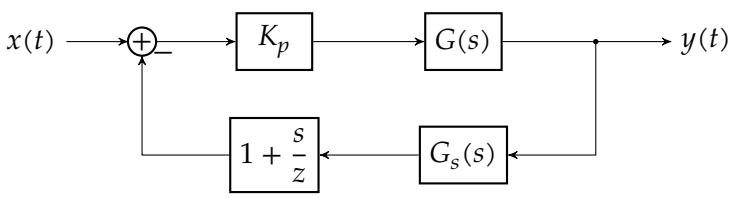
1. Put the zero of the controller near the second pole of $G(s)$.
2. Calculate the gain to provide for the desired phase margin.

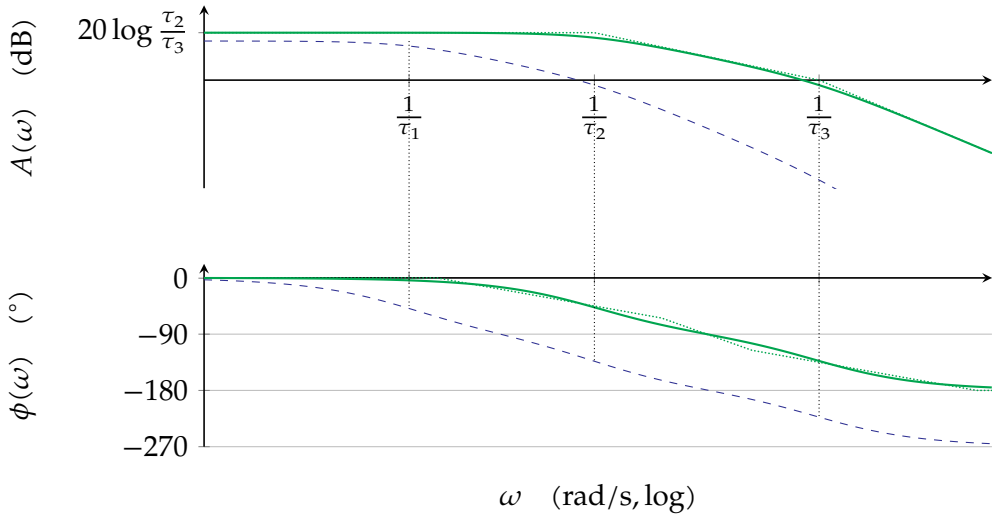
6.2.4 In the Laplace domain

The same design strategy can be used as for the single pole-zero phase lead controller.

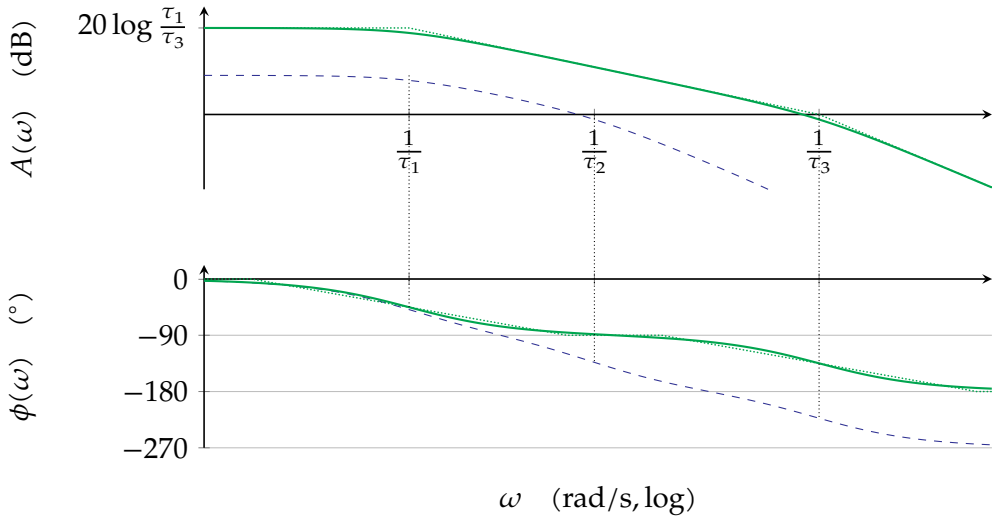
6.2.5 In the time domain

Again, to avoid overshoot, it is advisable to put the PD-controller in the feedback path:

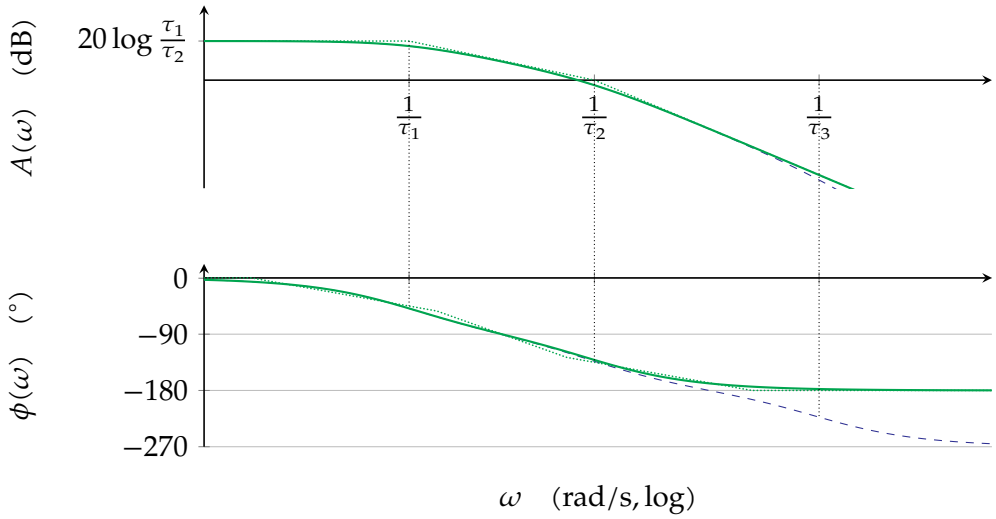




(a) Setting the controller's zero at the first pole (green curve); the original Bode plot has been indicated in blue dashed line.



(b) Setting the controller's zero at the second pole (green curve); the original Bode plot has been indicated in blue dashed line.



(c) Setting the controller's zero at the third pole (green curve); the original Bode plot has been indicated in blue dashed line.

Figure 6.9: Bode plots of the PD-controlled example of (6.4).

6.3 Single pole-zero phase lag controller

Secondly, let's study the phase lag controller. We will study its characteristics subsequently in the frequency, the Laplace and the time domain. A good understanding of this controller will allow us to understand the operation of the more common PI-controller.

6.3.1 The normal form

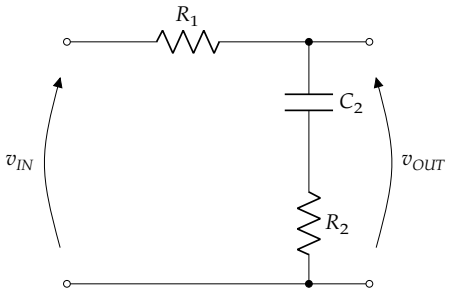
The transfer function of a single pole-zero phase lag controller can be easily derived in the so-called *normal form*, i.e. ensuring $z = \alpha p$ with $\alpha > 1$ and setting the low-frequency gain to 1. In this way, we obtain:

$$G_c(s) = \frac{1}{\alpha}\frac{s+z}{s+p} \qquad \text{with} \qquad z = \alpha p$$

One can call α the pole-zero distance factor (with $\alpha > 1$). Note that this is a special case of (6.1) with $p < z$ and $K = 1$.

6.3.2 Implementation example

Normal form A pole-zero phase lead controller can be implemented in many domains: mechanically, hydraulically, ...One way that interests us particularly is using electronic components. The following network does the job:



The transfer function is easily composed using Kirchhoff's laws and the branch equations of the elements, or by recognizing this circuit as a voltage divider:

$$\begin{aligned} G_c(s) = \frac{V_2(s)}{V_1(s)} &= \frac{R_2 + \frac{1}{sC_2}}{R_1 + R_2 + \frac{1}{sC_2 s}} \\ &= \frac{1 + R_2 C_2 s}{1 + (R_1 + R_2)C_2 s} \\ &\Big\downarrow \; R_2C_2 \equiv \tau \;\text{ and }\; \frac{R_1 + R_2}{R_2} \equiv \alpha \\ &= \frac{1 + \tau s}{1 + \alpha \tau s} \end{aligned}$$

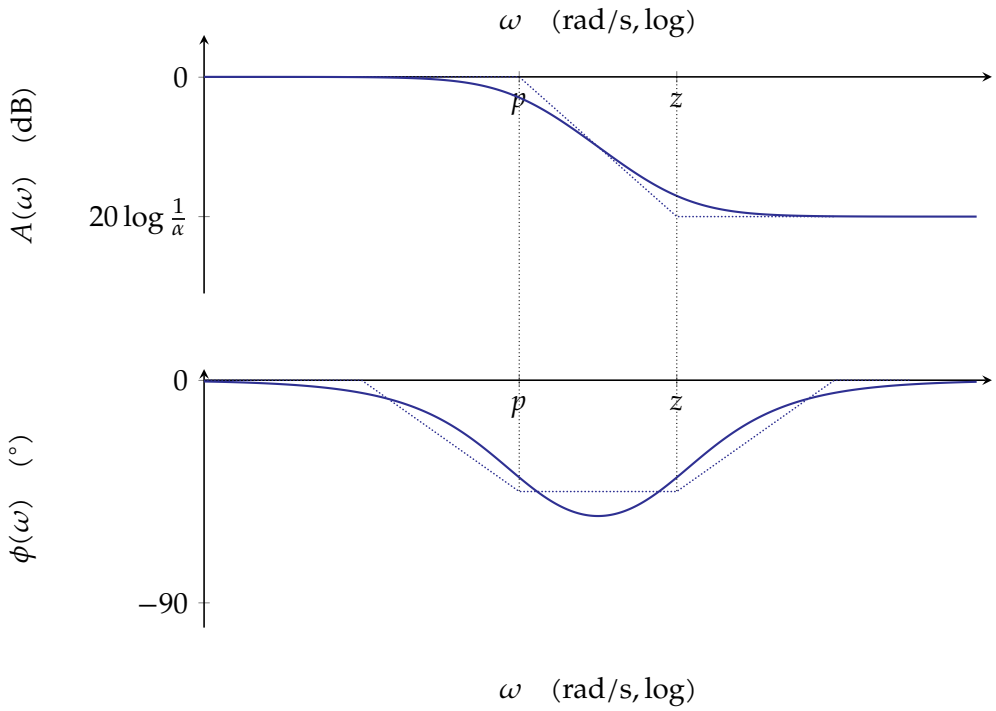


Figure 6.10: Bode plot of the single pole-zero phase lag controller

Non-normal form An extra gain factor K can be added by adding a simple op-amp amplifier in cascade as explained in section 6.1.2 on page 134.

6.3.3 In the frequency domain

Let's analyze our controller in the frequency domain. Let's consider the normal form. Going to the non-normal form is as simple as adding an extra gain factor K. The Bode plot corresponding to this controller in normal form can be found in Figure 6.10.

We can clearly see that for low frequencies the gain is 0 and the phase is 0°. Then the pole reduces the gain, also adding an extra amount of phase shift (making it a lagging phase), finally the zero stops the gain decrease and brings the phase up again to 0°.

Note that we did not indicate the maximal phase decrease in between the pole and the zero (as we did in the phase lead case). The reason for this is simple: the benefit of this controller lies not in its phase behavior, but in the gain decrease it offers for $\omega \gg z$ without affecting the phase for these frequencies. In this way, we are not interested in the phase behavior, only in the magnitude effect. The magnitude drops by $20\log(\alpha)$ dB.

The key question remains: how are we going to use this controller? We will use it in a fundamentally different way than we use the phase lead controller. As there is no benefit in the phase behavior of the controller (it worsens the situation), we will use it to provide extra low-frequency loop gain.

We propose the following design procedure. Because low-frequency gain is ill defined for systems containing an integrator, we will always consider the low-frequency gain without integrators. The *tick* after to the symbols K stands for 'without integrators'.

To fully understand the procedure, it may be useful to see the procedure as designing a non-normal controller:

$$H_c(s) = K_p \frac{s+z}{s+p}$$

The idea is to calculate the gain K_p first, and then selecting α to get the low-frequency gain (without the integrators) up to the desired level.

Phase lag design strategy in the frequency domain

1. Given $G(s)$, calculate its low-frequency gain without integrators K'_{DC}
2. Determine K_p such that $K_p G(j\omega)$ provides the desired phase margin (i.e. design a proportional controller). To this end you need to calculate ω_c.
3. Given the desired low-frequency loop gain (without integrators) K'_{des}, set

$$\alpha = \frac{K'_{des}}{K_p K'_{DC}}$$
$$z \ll \omega_c$$
$$p = z/\alpha$$

4. Verify, adjust.

Note that setting z much lower than the target ω_c ensures that the phase lag effect of the single pole-zero phase lag controller is almost negligible and does not deteriorate the phase margin. In this way the controller ensures a high low-frequency gain and a smaller gain at ω_c to allow for a better phase margin.

The effect of the controller is sometimes described using two metaphors:

1. using a sledge hammer, that keeps the low-frequency gain intact while hammering down the high-frequency part;
2. using an hydraulic jack that lifts the low-frequency part, while keeping the high-frequency part on the floor (i.e. where it was).

Therefore, seasoned control engineers may speak of a sledgehammer or a hydraulic jack approach.

Some examples will make this design strategy clear.

Example 1: Design a single pole-zero phase lag controller for

$$G(s) = \frac{14000}{(s+20)(s+100)}$$

The desired phase margin is 50° and the desired low-frequency gain $K'_{des} = 70$.

The system without controller It always makes sense to draw a Bode plot of the system without controller to be able to assess the situation. You can find it in Figure 6.11.

The next steps correspond one to one to our design strategy.

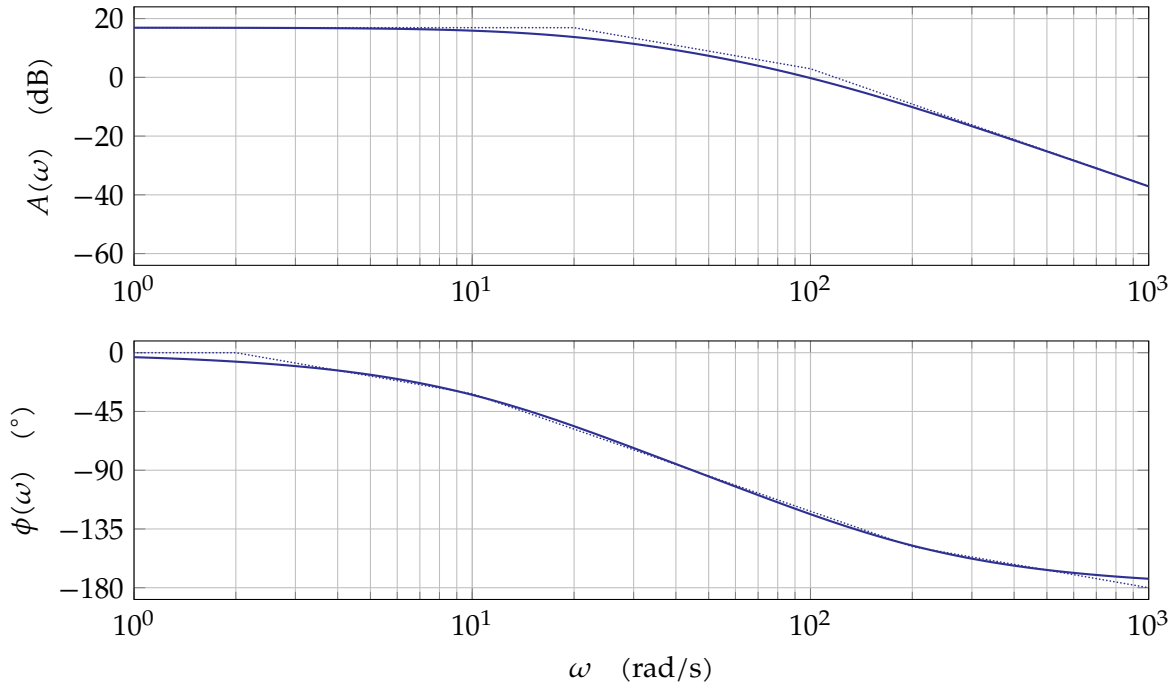


Figure 6.11: Bode plot of $\frac{14000}{(s+20)(s+100)}$, without controller

Calculate the low-frequency gain without integrators

$$K'_{DC} = |G(j0)| = \frac{14000}{20 \cdot 100} = 7$$

Design a proportional controller to realize the required phase margin The phase behavior of $G(s)$ can be described as:

$$\phi(\omega) = \angle G(j\omega) = -\operatorname{atan}\frac{\omega}{20} - \operatorname{atan}\frac{\omega}{100}$$

We will reach a phase margin of 50° if

$$\begin{aligned}
\phi_{PM} = 180° + \phi(\omega_c) &= 50° \\
130° &= \operatorname{atan}\frac{\omega_c}{20} + \operatorname{atan}\frac{\omega_c}{100} \\
&\downarrow \quad \tan \text{ of both sides and } \tan(\alpha+\beta) = \frac{\tan\alpha + \tan\beta}{1 - \tan\alpha\tan\beta} \\
-1.1918 &= \frac{\frac{\omega_c}{20} + \frac{\omega_c}{100}}{1 - \frac{\omega_c}{20}\frac{\omega_c}{100}}
\end{aligned}$$

This can be reworked into:

$$\begin{aligned}
1.1918\omega_c^2 - 120\omega_c - 2383.5 &= 0 \\
\omega_c &= \langle \begin{matrix} 117.7\,\text{rad/s} \\ -17.0\,\text{rad/s} \end{matrix}
\end{aligned}$$

of which of course only the positive value makes sense. This value corresponds to Figure 6.11, so we probably didn't make any calculation mistakes.

The gain K_p that sets the overall loop gain at this frequency to the required 0 dB can be solved

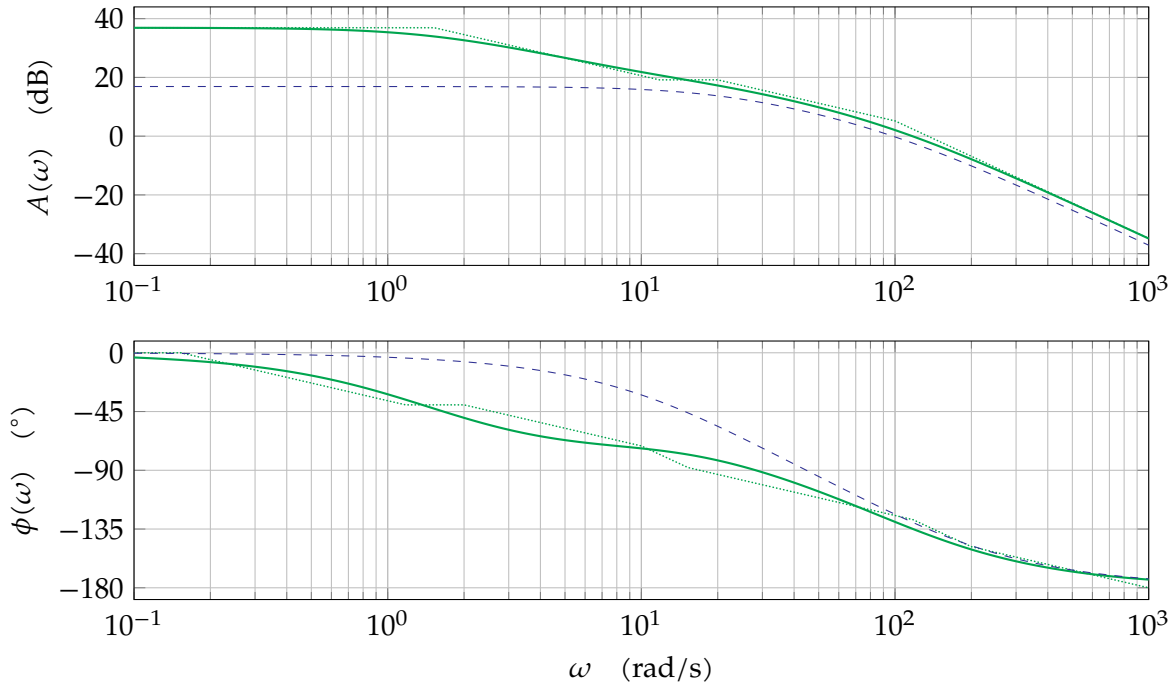


Figure 6.12: Bode plot of the loop gain with single pole-zero phase lag controller (in green); the original Bode plot of the system without controller has been indicated in dashed blue line.

from:

$$K_p|G(j\omega_c)| = 1$$
$$K_p \frac{14000}{\sqrt{\omega_c^2 + 20^2}\sqrt{\omega_c^2 + 100^2}} = 1$$
$$K_p = 1.2983$$

Again, inspection of Figure 6.11 confirms our result, as $20 \log K_p = 2.27\,\text{dB}$.

Design an appropriate phase lag controller According to the third step of our design strategy, we set:

$$\alpha = \frac{K'_{des}}{K_p K'_{DC}} = \frac{70}{1.2983 \cdot 7} = 7.7024$$
$$z = \frac{\omega_c}{10} = 11.77\,\text{rad/s}$$
$$p = \frac{z}{\alpha} = 1.528\,1\,\text{rad/s}$$

Verify, adjust
The result has been graphed in Figure 6.12. We can see that

- for high frequencies the loop gain slightly increased (with 2.27 dB) with respect to the uncontrolled loop gain,
- for the low frequencies the controller was able to give a significant boost to the low frequencies (i.e. the 'hydraulic jack' effect).

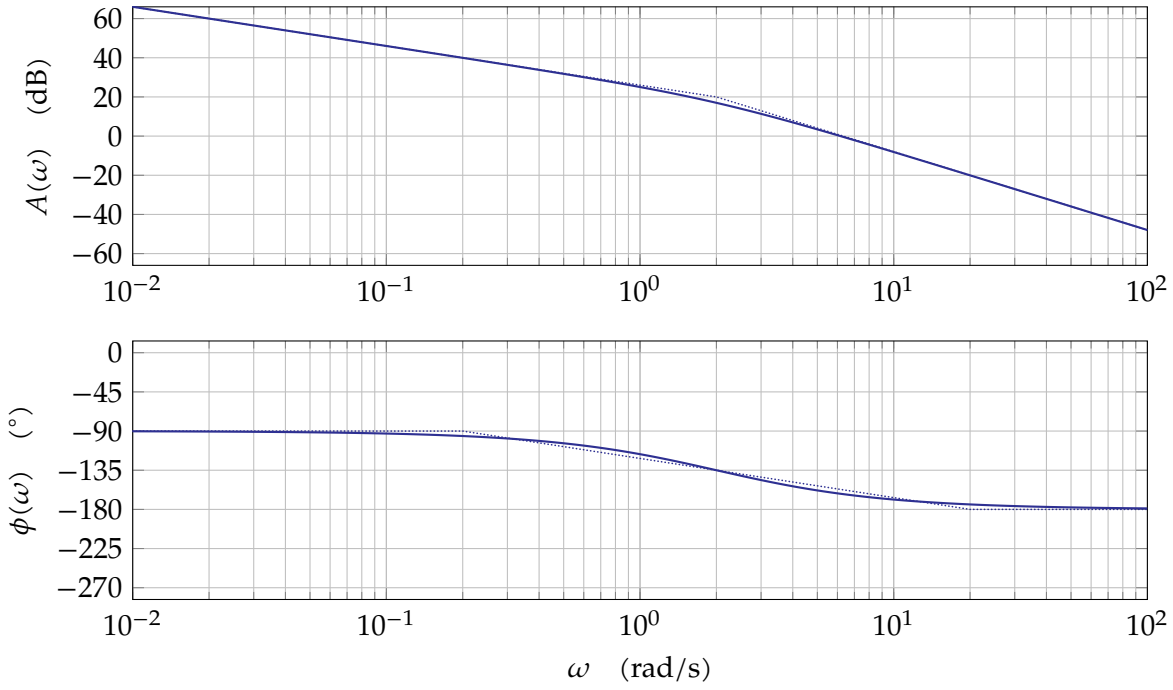


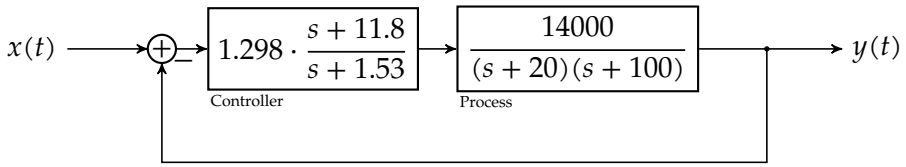
Figure 6.13: Bode plot of $\frac{40}{s(s+2)}$, without controller

Checking the phase margin again, yields:

$$\begin{aligned}\phi_{PM} &= 180^\circ + \operatorname{atan}\frac{\omega_c}{11.77} - \operatorname{atan}\frac{\omega_c}{1.53} - \operatorname{atan}\frac{\omega_c}{20} - \operatorname{atan}\frac{\omega_c}{100} \\ &\approx 45^\circ\end{aligned}$$

This is due to the fact that the phase of the controller is not yet zero at $\omega = 10z$. We might have taken that into account by designing for a slightly higher phase margin. As a rule of thumb, it makes sense to aim for an extra safety margin of 5°.

The resulting feedback system can be found below:



Example 2: Design a single pole-zero phase lag controller for

$$G(s) = \frac{40}{s(s+2)}$$

such that the phase margin amounts to 45°. The desired low-frequency loop gain without integrators is $K'_{des} = 20$.

The system without controller Again, it makes sense to assess the situation by drawing a Bode plot of the system without controller. You can find it in Figure 6.14.

Calculate the low-frequency gain without integrators

$$K'_{DC} = |G(j0)| = \frac{40}{2} = 20$$

Design a proportional controller to realize the required phase margin
The phase behavior of $G(s)$ can be described as:

$$\phi(\omega) = \angle G(j\omega) = -90° - \operatorname{atan}\frac{\omega}{2}$$

Adding a safety margin of 5° (see the rule of thumb we stated when discussing example 1), we aim for a phase margin of 50°. Therefore we need ω_c to comply with:

$$\begin{aligned} \phi_{PM} &= 180° + \phi(\omega_c) = 50° \\ 180° - 90° - \operatorname{atan}\frac{\omega_c}{2} &= 50° \\ \operatorname{atan}\frac{\omega_c}{2} &= 40° \\ \omega_c &= 2\tan 40° = 1.68\,\mathrm{rad/s} \end{aligned}$$

The gain K_p that sets the overall loop gain at this frequency to the required 0 dB can be solved from:

$$\begin{aligned} K_p|G(j\omega_c)| &= 1 \\ K_p\frac{40}{\omega_c\sqrt{\omega_c^2 + 2^2}} &= 1 \\ K_p &= 0.10954 \end{aligned}$$

Inspecting Figure 6.14 confirms our result, as $20\log K_p \approx -19$ dB.

Design an appropriate phase lag controller According to the third step of our design strategy, we set:

$$\begin{aligned} \alpha &= \frac{K'_{des}}{K_p K'_{DC}} = \frac{20}{0.10954 \cdot 20} = 9.1291 \\ z &= \frac{\omega_c}{10} = 0.168\,\mathrm{rad/s} \\ p &= \frac{z}{\alpha} = 0.018\,4\,\mathrm{rad/s} \end{aligned}$$

Verify, adjust
The result has been graphed in Figure 6.14. We can clearly see the phase margin is now approximately 45°.

6.3.4 In the Laplace domain

Let's start by reflecting on the way we have been using the phase lag controller so far. Our first step was to design a proportional controller, such that the system had an appropriate phase margin. Designing a proportional controller corresponds to drawing a root-locus plot in the Laplace domain. As we always designed the phase lag controller (almost) not to interfere with ω_c and the phase margin, we must assume that we can start from a root-locus plot that goes through our desired dominant poles for a specific value of K_p.

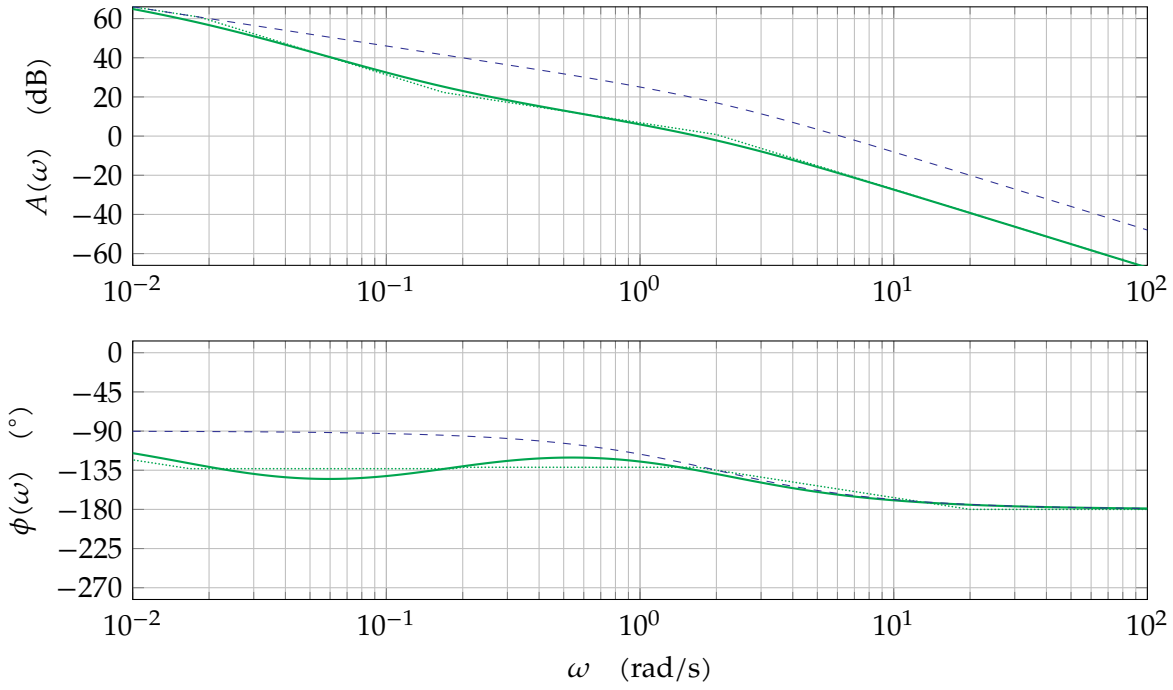


Figure 6.14: Bode plot of the loop gain with single pole-zero phase lag controller (in green); the original Bode plot of the system without controller has been indicated in dashed blue line.

Our goal is to be able to increase that gain factor K_p by adding a single pole-zero phase lag controller. However, adding the pole and zero of the controller will displace the intersection of the root locus asymptotes with the real axis (i.e. σ_A) in the root locus plot. If the original intersection (without the controller) is

$$\sigma_A = \frac{\sum p_i - \sum z_i}{n_p - n_z}$$

then adding the controller $\frac{s+z}{s+p}$ will add a term to σ_A:[1]

$$\sigma_A = \frac{\sum p_i - \sum z_i - p + z}{n_p - n_z}$$

The displacement equals:

$$\begin{aligned} \Delta_{\sigma_A} &= \frac{z - p}{n_p - n_z} \\ &\downarrow \quad p = z/\alpha \\ \Delta_{\sigma_A} &= \frac{z(1 - \frac{1}{\alpha})}{n_p - n_z} \end{aligned}$$

Clearly, if we want to keep the displacement small, we need to go for small values of z. In case of $n_p > n_z$ the displacement will be to the right, otherwise to the left in the complex plane.

By setting a maximally allowable displacement $\Delta_{\max}$, we can calculate z as:

$$z = (n_p - n_z)\frac{\alpha}{\alpha - 1}\Delta_{\max}$$

[1]Remember: the pole and zero of the controller are $-p$ and $-z$ respectively.

Phase lag design strategy in the Laplace domain

1. Design a proportional controller by drawing a root locus plot and determining a proper spot for the dominant poles. Calculate the corresponding value of K_p.
2. Decide by how much you want to increase that gain. This factor is α.
3. Decide on an maximal allowable displacement of the asymptotes and use it to calculate z and p:

$$z = (n_p - n_z)\frac{\alpha}{\alpha - 1}\Delta_{\max}$$
$$p = z/\alpha$$

Let's illustrate this design strategy using the second example of previous section.

Example: Design a single pole-zero phase lead controller in the Laplace domain for

$$G(s) = \frac{K}{s(s+2)} \tag{6.5}$$

and

- assume the desired position of the dominant poles of the overall system to be $p_{a,b} = -1 \pm 2j$,
- aim for a low-frequency loop gain without integrators of $K'_{DC} = 20$.

Design a proportional controller

Let's start by drawing a root locus plot corresponding to (6.5). You can find the plot in Figure 6.15.

The gain corresponding to the dominant poles can be determined by solving the magnitude condition for K:

$$|G(p_a)| = 1$$
$$\frac{K}{|p_a||p_a + 2|} = 1$$
$$K = \sqrt{(-1)^2 + 2^2}\sqrt{(-1+2)^2 + 2^2} = \sqrt{5}\sqrt{5} = 5$$

Determine α

This value of K will result in a low-frequency gain without integrators $K'_{DC} = 2.5$. As we need $K'_{DC} = 20$ we need a jacking-up operation by a factor of 8. Therefore $\alpha = 8$.

Determine z and p for a maximal displacement of σ_A

As can be seen from Figure 6.15 and easily calculated, the intersection of the asymptotes with the real axis (for the proportionally controlled system) is:

$$\sigma_A = \frac{\sum_i p_i - \sum_j z_i}{n_p - n_z} = \frac{-2}{2} = -1$$

As $n_p - n_z > 0$, the displacement of adding our controller will be to the right (i.e. positive). Let's allow for a maximal displacement of 5%, i.e. $\Delta_{\max} = 0.05$. This allows for calculation z

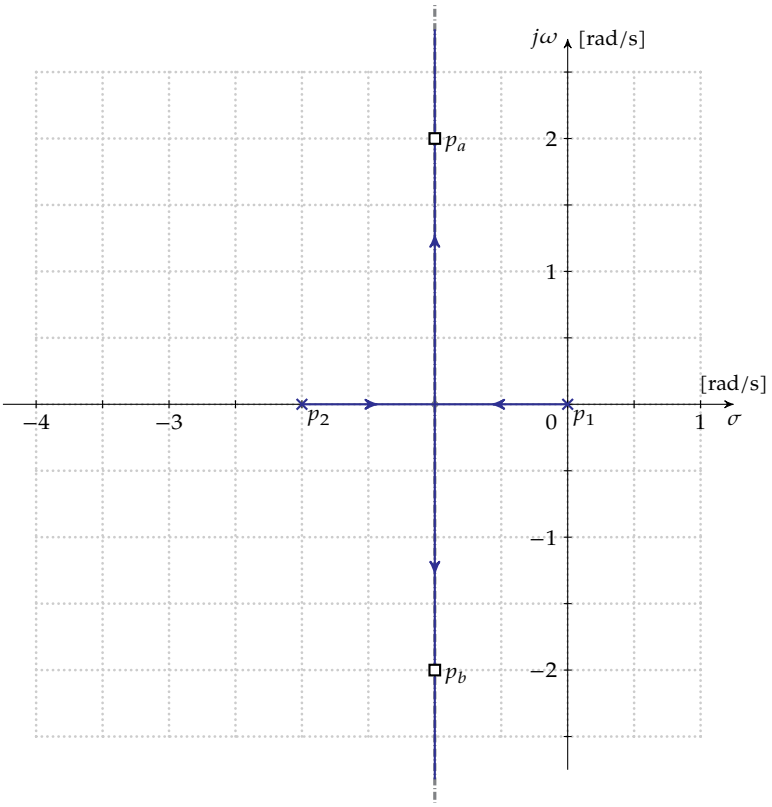


Figure 6.15: Root locus of a feedback system with loop gain $G(s) = \frac{K}{s(s+2)}$. The desired location of the dominant poles $p_{a,b}$ has been indicated.

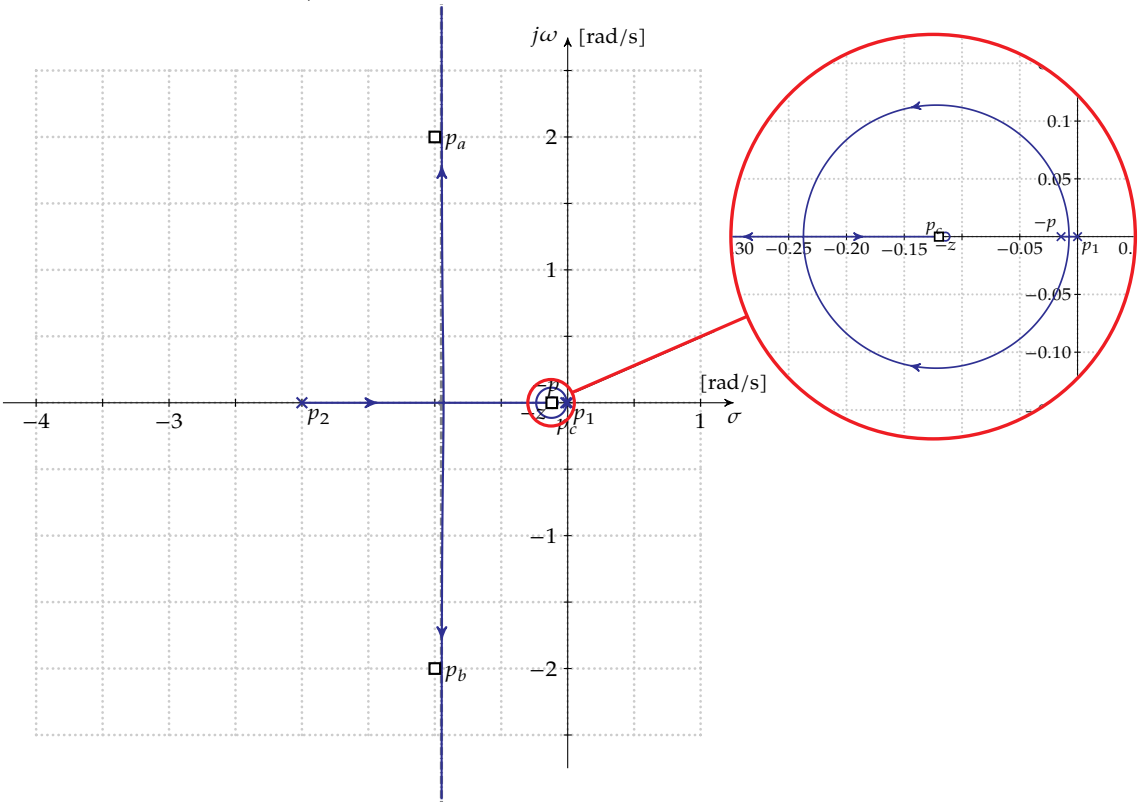


Figure 6.16: Root locus of a feedback system with loop gain $G(s) = \frac{s+0.114}{s+0.0143}\frac{K}{s(s+2)}$. The desired location of the dominant poles $p_{a,b}$ has been indicated. The area around $-z$ has been magnified.

and p. Indeed:

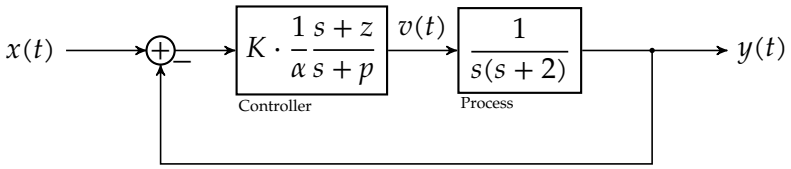
$$z = (n_p - n_z)\frac{\alpha}{\alpha - 1}\Delta_{\max} = 0.114\,\text{rad/s}$$
$$p = z/\alpha = 0.014\,3\,\text{rad/s}$$

The corresponding root locus of the system with controller can now be drawn. You can find the result in Figure 6.16. As one can see, the location of the asymptotes is slightly shifted to the right.

6.3.5 In the time domain

The previous two sections each contained a design procedure for our single pole-zero phase lead controller. This section will not yield a new design procedure but will prevent us from making stupid mistakes when it comes to the time-domain behavior of the controlled system.

If we consider the system of our last example, the configuration looks like this:



The overall transfer function of this system is easily calculated to be:

$$T(s) = \frac{K/\alpha(s+z)}{s(s+2)(s+p) + K/\alpha(s+z)} = \frac{K}{\alpha}\frac{s+z}{(s-p_a)(s-p_b)(s-p_c)}$$

As $z \gg p$ it makes no sense to put the controller in the feedback path: this would only increase the overshoot. In addition, if K is sufficiently large, p_c will almost coincide with (and therefore cancel) the zero of the controller (as can be seen in the enlargement of Figure 6.16).

In that case the transfer function can be approximated as:

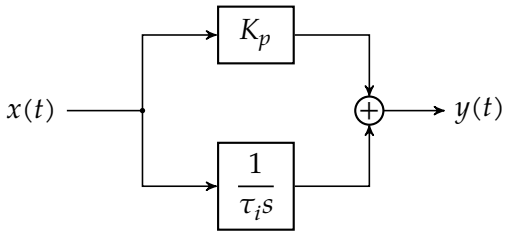
$$T(s) \approx \frac{K/\alpha}{(s-p_a)(s-p_b)}$$

Note that this was our intent: the controller almost does not influence the overall behavior, only offers an increased low-frequency loop gain.

6.4 The PI-controller

6.4.1 The normal form

The PI-controller is the sum of a P-controller (a gain block with gain K_p) and an I-controller (an integrator with gain $1/\tau_i s$):



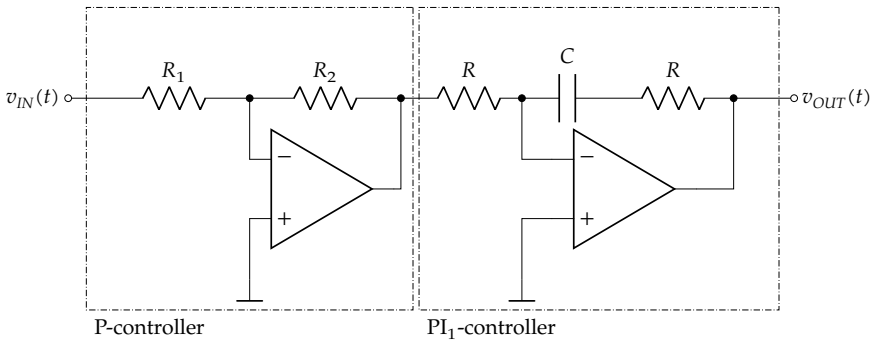
The overall transfer function of the controller therefore becomes:

$$G_c(s) = K_p + \frac{1}{\tau_i s} = K_p \frac{s+z}{s} \qquad \text{with } z = \frac{1}{K_p \tau_i}$$

Note that this is a special case of the single pole-zero phase lag controller. This one has a pole at $s = 0$.

6.4.2 Implementation example

In practice, a PI-controller is most often realized as a cascade of a P-controller and a PI_1-controller (i.e. a PI-controller with gain 1). The following schematic does the job:



The P-controller offers a negative gain $G_p = -R_2/R_1$. The PI_1-controller corrects the sign by inverting itself with a gain $G_{PI_1} = -\frac{RCs+1}{RCs}$. The overall transfer function becomes:

$$H(s) = \frac{R_2}{R_1}\frac{RCs+1}{RCs} = \underbrace{\frac{R_2}{R_1}}_{\equiv K_p}\frac{s+z}{s} \qquad \text{with } z = \frac{1}{RC}$$

One can make the gain adjustable by making R_2 a variable resistor. One can make the zero (independently) adjustable by taking a variable capacitor or a dual pot-meter for R.

6.4.3 In the frequency domain

The Bode plot of the PI-controller can be found in Figure 6.17. For high frequencies, the gain equals a constant K_p, for low frequencies it evolves as $\frac{1}{\tau_i \omega}$. The zero occurs at the intersection of those two lines (on a logarithmic scale), i.e. $K_p = \tau_d \omega$.

Let's find out how to use the PI-controller from the Fourier domain's perspective using a number of examples.

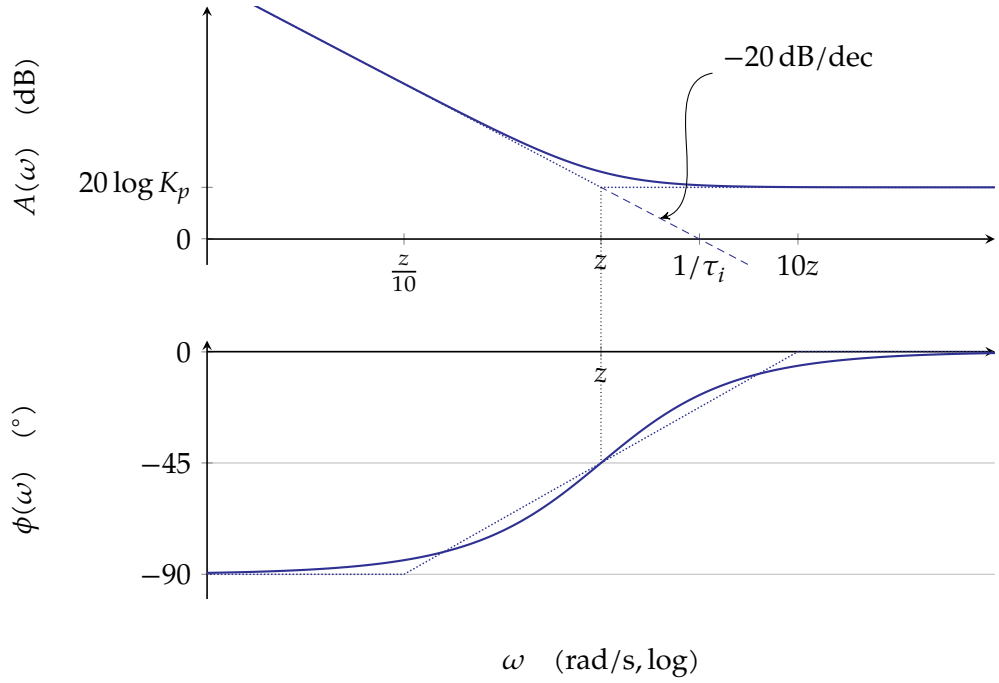


Figure 6.17: Bode plot of the PI-controller

Example 1: Design a PI-controller for a process given by:

$$G(s) = \frac{K_G}{(1+\tau_1 s)(1+\tau_2 s)(1+\tau_3 s)}$$

assuming $\tau_1 \gg \tau_2 \gg \tau_3$. The design goal is to maximize ω_c while keeping a decent phase margin $\phi_{PM} = 45°$.

As the poles are far apart, we may assume the phase to be exactly $-135°$ at $\omega = 1/\tau_2$. A DC-gain K to obtain $\omega_c = 1/\tau_2$ equals $K = \tau_1/\tau_2$. The Bode plot corresponding to this situation can be found in Figure 6.8 on page 152. This would be the optimal situation when using a proportional controller, i.e. $K_p = K/K_G$.

What benefit can we get from a PI-controller? As this type of controller cannot improve the phase, we need to position it such that it does not interfere with the phase margin at ω_c.

In view of this insight, we formulate the following design strategy:

PI-controller design strategy in the frequency domain (for systems without time delay)

1. Put the zero of the controller (more than) a decade before the second pole.
2. Optimize the position of the zero in a trade-off between phase margin and an increased low-frequency DC gain.

In our case, this would correspond to putting the zero of the controller at the first pole. This changes the system into a type 1 system, offering the benefit of a zero steady-state error for a step input. This situation has been graphed in Figure 6.18.

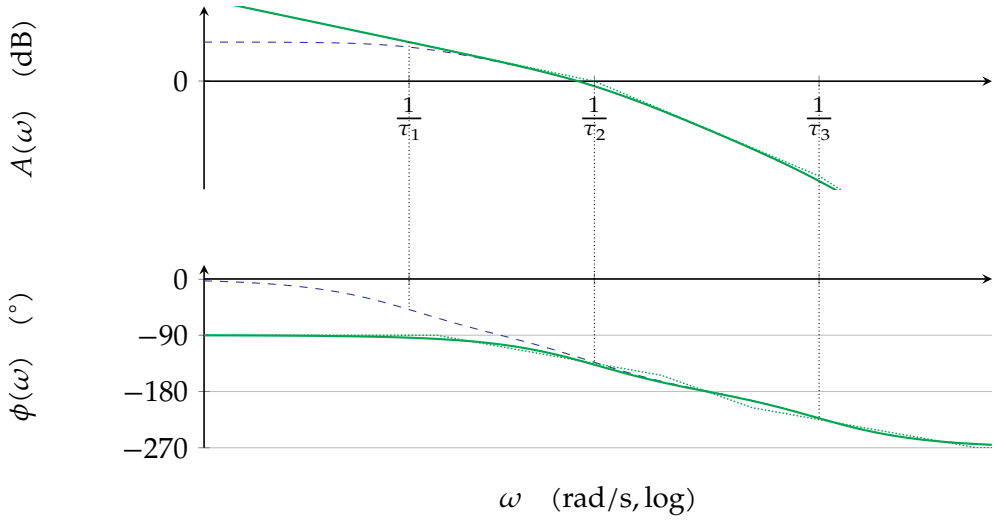


Figure 6.18: The system of (6.4) proportionally controlled (in blue dashed line) and controlled with a PI-controller with the zero set at the first pole (in green)

Example 2: Design a PI-controller for a process given by

$$G(s) = \frac{K_G\, \mathrm{e}^{-sT}}{1 + \tau_1 s} \tag{6.7}$$

and assume that the time delay is dominated by the first pole. Make sure there is about 45° phase margin.

In this case (as we have seen before) the phase margin of the loop gain is determined by the time delay. The phase of the loop gain can be derived from (6.7) and is described by the following expression:

$$\phi(\omega) = \angle G(j\omega) = -\omega T \frac{180^\circ}{\pi} - \mathrm{atan}(\omega\tau_1)$$

Note the conversion factor from radians to degrees in the first term.

Let's start by designing a proportional controller, resulting in a loop gain equaling:

$$L(s) = K_p K_G \frac{\mathrm{e}^{-sT}}{1 + \tau_1 s}$$

For a proportional controller our job is to set a gain K_p such that the cross-over frequency ω_c is set at a frequency that still offers the appropriate phase margin (i.e. where the phase line crosses -135°): The frequency for which this happens is fixed by:

$$\phi(\omega) = -\omega T \frac{180^\circ}{\pi} - \mathrm{atan}(\omega\tau_1) = -135^\circ$$

Solving this equation for ω is difficult and can only be done numerically. To simplify things, let's assume $\omega \gg 1/\tau_1$, i.e. that the atan- term already arrived at -90°, i.e.

$$\phi(\omega) = -\omega T \frac{180^\circ}{\pi} - 90^\circ = -135^\circ$$

Solving ω from this equation is easy and yields:

$$\omega = \frac{45^\circ}{180^\circ} \frac{\pi}{T} = \frac{\pi}{4T}$$

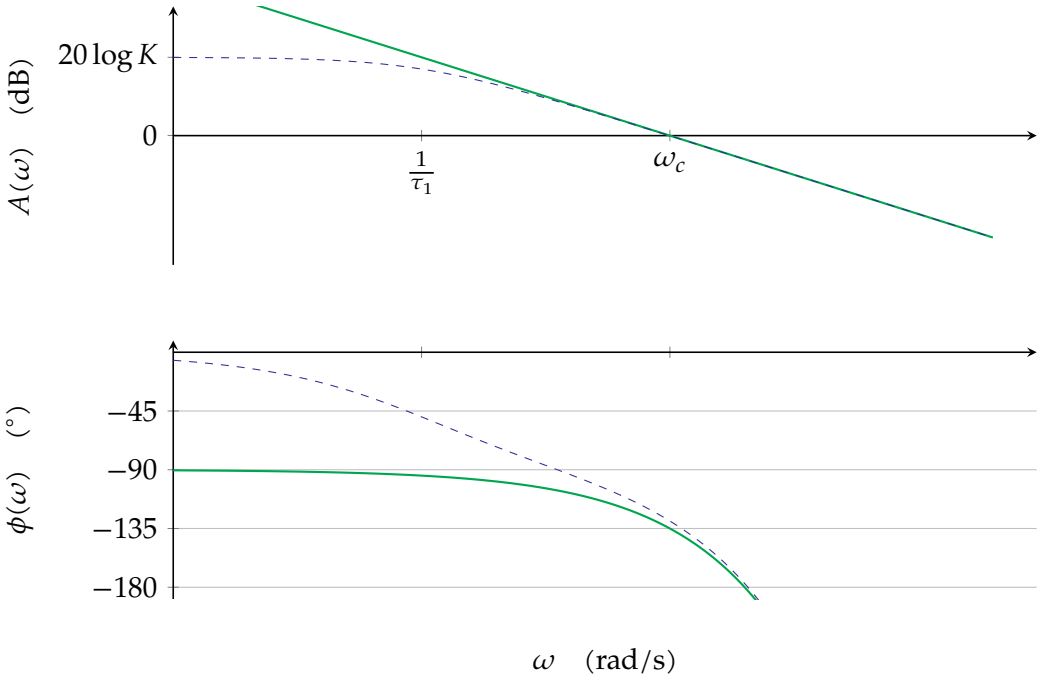


Figure 6.19: The system of (6.7) (with a time-delay dominated by the first pole) controlled with a P-controller (in blue dashed line) and with a PI-controller with the zero set at the first pole (in green)

If we set the cross-over frequency ω_c to this value, the gain at that frequency should equal 1, i.e.

$$|L(j\omega_c)| = 1$$
$$\frac{K_p K_G}{\sqrt{(\omega_c \tau_1)^2 + 1^2}} = 1$$

Assuming $\omega_c \gg 1/\tau_1$ as we did earlier, we can simplify:

$$\frac{K_p K_G}{\omega_c \tau_1} = 1$$
$$K_p = \frac{\omega_c \tau_1}{K_G}$$

This has been graphed by the blue dashed line in Figure 6.19.

The design strategy for our PI-controller is again the same, i.e. make it not interfere with the phase margin, e.g. by setting the controller's zero equal to the first pole: $z = 1/\tau_1$.

PI-controller design strategy in the frequency domain (for systems with non-dominating time delay) Put the zero of the controller (more than) a decade before the cross-over frequency ω_c obtained from the proportional controller design, e.g. at the dominating pole.

In this way the controller cancels the dominant pole, resulting again in a type 1 system. You can find the resulting loop gain as the green line in Figure 6.19.

In the next example we will illustrate what happens if the time delay dominates the dominant pole.

Example 3: Design a PI-controller for a process given by

$$G(s) = \frac{K_G\, e^{-sT}}{1 + \tau_1 s} \tag{6.8}$$

but now assume that the time delay dominates the pole. Make sure there is about 45° phase margin.

Take a look at the Bode plot (in dashed blue line) that corresponds to this situation in Figure 6.20.

The phase can be calculated from (6.7):

$$\phi(\omega) = \angle G(j\omega) = -\omega T \frac{180^\circ}{\pi} - \text{atan}(\omega\tau_1)$$

Let's again try to design a proportional controller first. In the frequency range that would correspond with $\phi_{PM} = 45^\circ$, it is clear that we can approximate the phase as:

$$\phi(\omega) \approx -\omega T \frac{180^\circ}{\pi}$$

This allows calculating an appropriate ω_c, to obtain $\phi_{PM} = 45^\circ$, i.e.

$$\phi(\omega_c) \approx -\omega_c T \frac{180^\circ}{\pi} = -135^\circ$$
$$\omega_c = \frac{3}{4}\frac{\pi}{T}$$

However, at that frequency (check Figure 6.20) the gain is still way above 0 dB! We need a way to bring that gain down. The downwards slope of the PI-controller (for low frequencies) will do that for us. In addition, the proportional part of the controller makes no sense anymore as it only acts for frequencies way above the cross-over frequency ω_c, and therefore, we will use a pure I-controller (i.e. we set the zero of the controller at infinity).

PI-controller design strategy in the frequency domain (for systems with dominating time delay) Use an I-controller K_i/s and determine K_i such that ω_c provides for sufficient phase margin.

In view of the above, the loop gain can be described to be:

$$L(s) \approx \frac{K_i}{s} K_G\, e^{-sT}$$

Because the I-controller influences the phase significantly (it adds an extra delay of 90°, we cannot but redo the work to calculate a decent cross-over frequency ω_c, by re-evaluating the phase:

$$\phi(\omega) = \angle L(j\omega) = -90^\circ - \omega T \frac{180^\circ}{\pi}$$

We can solve for the cross-over frequency ω_c from our desire to have a phase margin of 45°:

$$\phi_{PM} = 180^\circ + \phi(\omega_c) = 45^\circ$$
$$180^\circ - 90^\circ - \omega_c T \frac{180^\circ}{\pi} = 45^\circ$$
$$\omega_c = \frac{1}{4}\frac{\pi}{T}$$

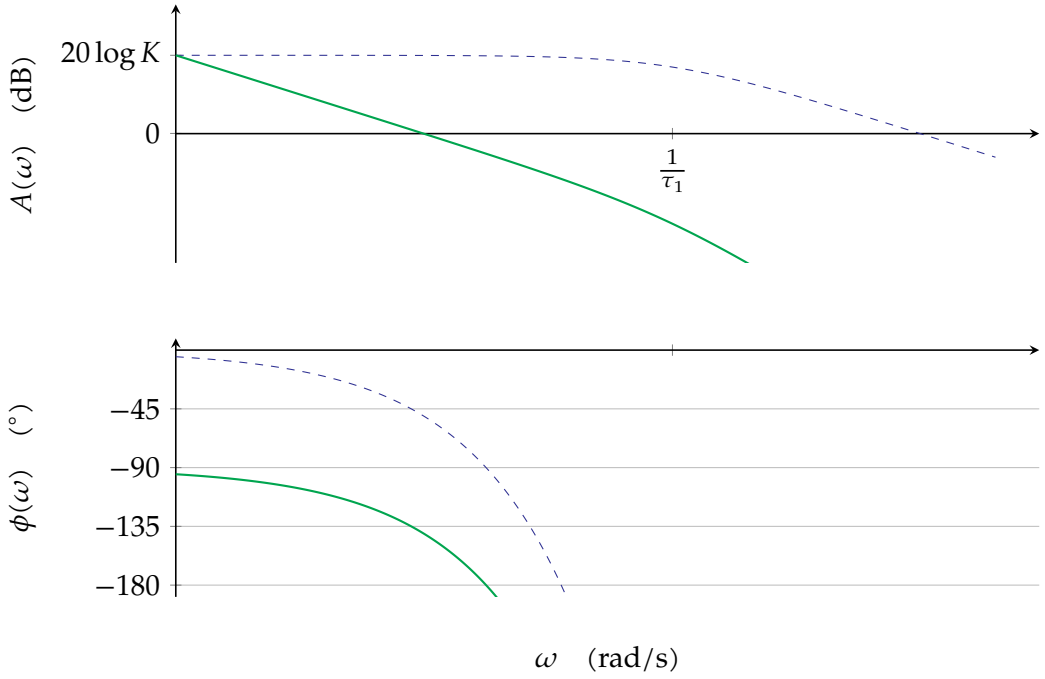


Figure 6.20: The system of (6.8) (with a time-delay that dominates the poles) that is uncontrollable with a P-controller (in blue dashed line) and appropriately controlled with an I controller (in green)

At that frequency, K_i should be such that the loop gain dropped to 0 dB, i.e.

$$\begin{aligned} |L(j\omega_c)| &= 1 \\ \frac{K_i}{\omega_c} K_G &= 1 \\ K_i &= \frac{\omega_c}{K_G} \end{aligned}$$

This situation has been depicted as the green line in Figure 6.20. Problem solved.

6.4.4 In the Laplace domain

The same design strategy can be used as for the single pole-zero phase lag controller.

6.4.5 In the time domain

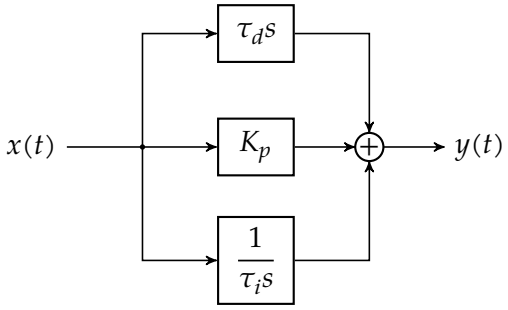
As for the single pole-zero phase lag controller, the PI-controller is to be put in the forward path in any case.

6.5 The PID controller

The PID-controller combines the advantages of a PD-controller and a PI-controller.

6.5.1 The normal form

The PID-controller is the sum of a P-controller (a gain block with gain K_p), a D-controller (a differentiator with gain $\tau_d s$) and an I-controller (an integrator with gain $1/\tau_i s$):



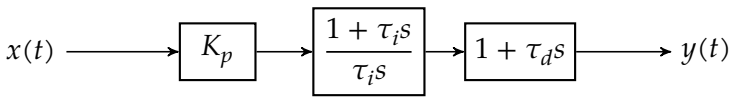
The overall transfer function of the controller can therefore be written as:

$$G_c(s) = \tau_d s + K_p + \frac{1}{\tau_i s} = \frac{\tau_d s^2 + K_p s + \frac{1}{\tau_i}}{s}$$

In view of this, the controller has a pole in the origin and two zeros. The zeros are determined by the combination of the three parameters. This hinders a intuitive (interactive) setting of the parameters, however, it allows for complex conjugate zeros. This can be of advantage when designing the controller in the Laplace domain.

6.5.2 The cascade form

The cascade form allows for an independent control of the zeros (and mid-band gain), however, it does not allow for complex conjugate zeros:



The zero of the PI_1 controller is $1/\tau_i$, the zero of the PD_1 controller equals $1/\tau_d$. These are both real. The corresponding transfer function is:

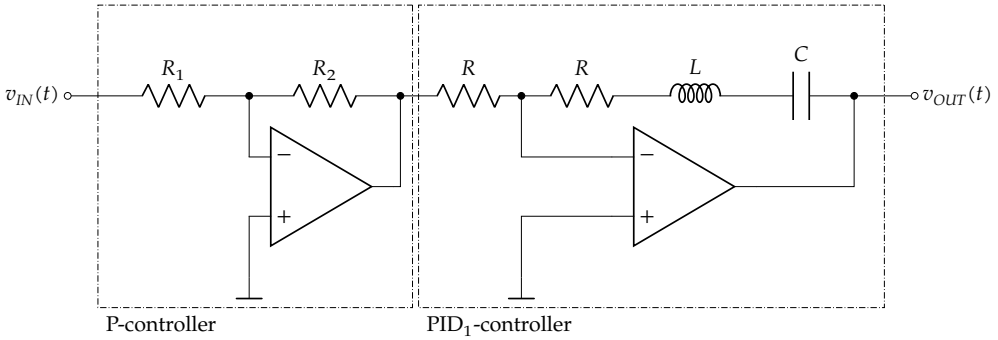
$$G_c(s) = K_p \frac{(1+\tau_i s)(1+\tau_d s)}{\tau_i s}$$

6.5.3 Implementation example

Cascade form In practice, a PID-controller is most often realized as a cascade of a P-controller, a PD_1-controller and a PI_1-controller, whose implementations can be seen in section 6.2.2 on page 150 and section 6.2.2 on page 150.

If the PD_1- and PI_1-controllers are implemented using an inverting op-amp configuration, of course the P-controller must be non-inverting.

Complex form The complex form is most often realized as a cascade of a P-controller and a PID_1-controller. The following schematic does the job:



The P-controller offers a negative gain $G_p = -R_2/R_1$. The PID_1-controller corrects the sign by inverting itself with a gain $G_{PI_1} = -\frac{LCs^2+RCs+1}{RCs}$. The overall transfer function becomes:

$$\begin{aligned} H(s) &= \frac{R_2}{R_1}\frac{LCs^2 + RCs + 1}{RCs} \\ &= \frac{R_2}{R_1}\frac{1}{\tau s}\frac{s^2 + 2\zeta\omega_0 s + \omega_0^2}{\omega_0^2} \qquad \text{with } \omega_0 = \frac{1}{\sqrt{LC}},\ \zeta = \frac{R}{2}\sqrt{\frac{C}{L}},\ \tau = RC \end{aligned}$$

One can make the gain adjustable by making R_2 a variable resistor. Careful selection of R, L, and C allow for controlling the behavior of the second-order double zero system.

6.5.4 In the frequency domain

The Bode plot of the PID-controller can be found in Figure 6.21, under the assumption the zeros are real, i.e.:

$$G_c(s) = K_p\frac{1 + \tau_i s}{\tau_i s}(1 + \tau_d s)$$

For the mid frequencies, the magnitude of the gain equals a constant K_p, for low frequencies it evolves as $\frac{K_p}{\tau_i\omega}$, for high frequencies as $K_p\tau_d\omega$.

Let's find out how to use the PID-controller from the Fourier domain's perspective using an example.

Example: Design a PID-controller for

$$G(s) = \frac{K_G\,\mathrm{e}^{-sT}}{(1 + \tau_1 s)(1 + \tau_2 s)(1 + \tau_3 s)}$$

assuming $\tau_1 \gg \tau_2 \gg \tau_3$. The design goal is to maximize the cross-over frequency ω_c while keeping a decent phase margin $\phi_{PM} = 45°$.

Depending on the significance of the time delay T, we need to distinguish the following cases:

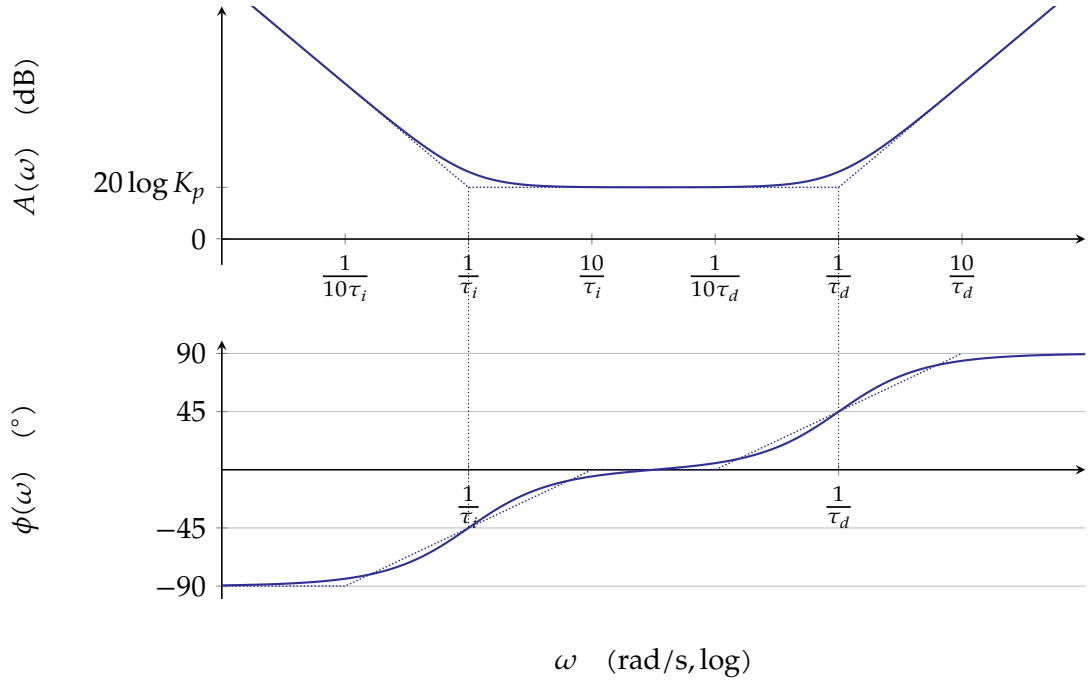


Figure 6.21: Bode plot of the PID-controller

$T < \tau_3$: The delay is non-dominant. We can use the PID-controller to compensate for the first and the second pole, such that the phase margin is again determined by the time delay. Therefore, setting

$$\tau_i = \tau_1 \qquad\qquad \tau_d = \tau_2$$

the loop gain becomes:

$$L(s) = G(s)G_c(s) = \frac{K_p K_G \, \mathrm{e}^{-sT}}{(1 + \tau_3 s)\tau_i s}$$

leading to the following phase behavior:

$$\phi(\omega) = \angle L(j\omega) = -90° - \arctan(\omega\tau_3) - \frac{180°}{\pi}\omega T$$

and the following phase margin:

$$\phi_{PM} = 180° + \phi(\omega_c) = 180° - 90° - \arctan(\omega_c\tau_3) - \cancel{\frac{180°}{\pi}\omega_c T}$$

As we assumed $T \ll \tau_3$, we assume the phase transition because of τ_3 to happen way before the delay kicks in, and therefore, we can neglect the last term.
Conclusion: a cross-over frequency $\omega_c = \frac{1}{\tau_3}$ will provide for a 45° phase margin. This allows for determining the proper gain to ensure $|L(j\omega_c)| = 1$:

$$\frac{K_p K_G}{\sqrt{1 + (\tau_3\omega_c)^2}\tau_i\omega_c} = 1$$

$$K_p = \frac{\sqrt{2}}{K_G}\frac{\tau_1}{\tau_3}$$

$\tau_3 < T < \tau_2$: We still can use the PID-controller to compensate for the first and the second pole:

$$\tau_i = \tau_1 \qquad \tau_d = \tau_2$$

However, in this case the phase is dominated by the delay (and therefore, we neglect the last term):

$$\phi(\omega) = \angle L(j\omega) = -90° - \frac{180°}{\pi}\omega T - \cancel{\arctan(\omega\tau_3)}$$

Ensuring $\phi_{PM} = 45°$ now requires:

$$\omega_c = \frac{\pi}{4T}$$

Again, this allows for determining the proper gain to ensure $|L(j\omega_c)| = 1$. Assuming $\omega_c \ll \frac{1}{\tau_3}$ allows simplifying:

$$\frac{K_p K_G}{\tau_i \omega_c \sqrt{1 + \cancel{(\tau_3\omega_c)^2}}} = 1$$

$$K_p = \frac{1}{K_G}\frac{\pi}{4}\frac{\tau_1}{T}$$

$\tau_2 < T < \tau_1$: In this case, canceling the second pole using the second zero of the controller does not make any sense. On the contrary, it is dangerous, as it will introduce an extra gain for higher frequencies, resulting in an increase in cross-over frequency ω_c. We need to control ω_c in such a way that cross-over occurs at a phase margin that is still decent. Therefore, the advice is to use a PI-controller instead. Revert to the section on PI-controllers to see how to cope with this situation.

$\tau_1 < T$: In this case, the time delay is dominant. We have seen this case when treating the PI-controller. The same advice holds here: use a pure I-controller in this case. Again, revert to the section on PI-controllers to see how to cope with this situation.

6.5.5 In the Laplace domain

We will use an example to illustrate a proper design strategy in the complex frequency domain.

However, let's first investigate our complex PID controller. We know that in its normal form:

$$\begin{aligned} G_c(s) &= K_p + \frac{1}{\tau_i s} + \tau_d s \\ &= \frac{K_p s + 1 + \tau_d s^2}{\tau_i s} \\ &= \tau_d \frac{s^2 + \frac{K_p}{\tau_d}s + \frac{1}{\tau_d\tau_i}}{s} \\ &= \tau_d \frac{(s - n_1)(s - n_2)}{s} \end{aligned}$$

with $n_2 = \overline{n_1}$ the complex conjugate zeros of the controller. Note that next to the complex zeros also a pole at the origin is added.

Example: Design a PID-controller in the Laplace domain for a process with transfer function:

$$G(s) = \frac{60}{(s+2)(s+3)} \tag{6.9}$$

Let's first investigate what a P-controller could do for us, by drawing the root-locus for this system. This has been done in Figure 6.22. As can be readily seen, the settling time will be limited to $T_{s,2\%} \approx \frac{4}{\zeta\omega_n} = 4/2.5 = 1.6\,\text{s}$. Increasing the gain, will only lead to less damping and hence more overshoot.

The problem lies in the fact that the asymptotes of root locus plot that do not offer the possibility for the resulting overall roots to penetrate more into the left half plane. The basic idea of using the complex zero pair of the PID-controller is to provide alternative paths to the poles of the overall system.

Look at Figure 6.23 to see what happens to the root locus if we choose $n_{1,2} = -4 \pm 2j$. Instead of the being forced north and south, the rightmost two loci are offered an exit at the controller's zeros.

This choice of complex zeros for the controller, results in a transfer function for the controller of:

$$G_c(s) = \tau_d \frac{s^2 + 8s + 20}{s}$$

Let's fix the remaining parameter τ_d (controlling the gain, given the fixed zeros) and determine K_p and τ_i:

$$\tau_d = 0.1\,\text{s} \qquad K_p = 8\tau_d = 0.8 \qquad \tau_i = \frac{1}{20\tau_d} = 0.5\,\text{s}$$

This results in the following poles for the overall system:

$$r_{1,2} = -3 \pm 3.9j \qquad r_3 = -5$$

These have been indicated in Figure 6.23. We can simulate this system using:

```
%% System
G = tf( 60, conv([1 2], [1 3] ) );

%% PID controller
taud = 0.1;
taui = 1/20/taud;
Kp = 8*taud;
Gc = tf( taud * [ 1 Kp/taud 1/taud/taui ], [ 1 0] );

%% Loop gain
L = G * Gc;

%% Overall gain
T = feedback( L, 1 );

%% Calculate overall roots
pole(T)

%% simulate to determine settling time
step(T);
```

This reveals a 2% settling time of approximately 0.92 s.

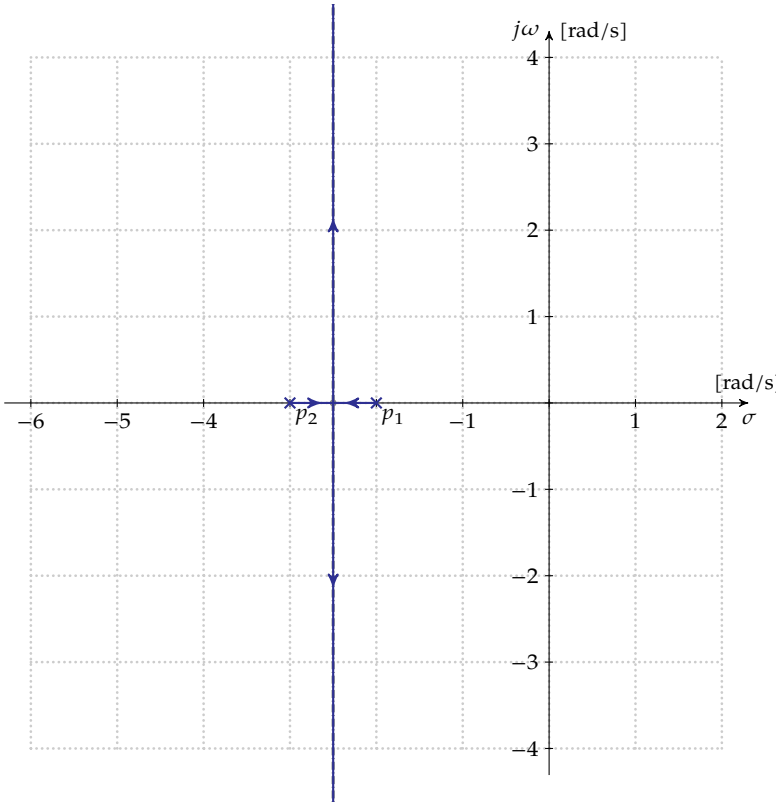


Figure 6.22: Root locus for the system of (6.9), illustrating the potential of a proportional controller

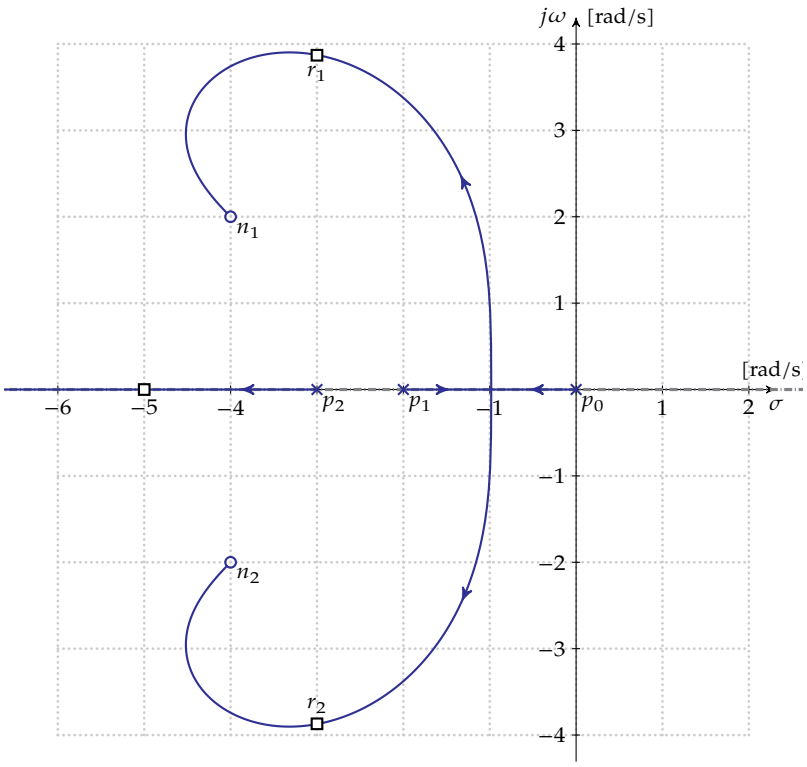


Figure 6.23: Root locus for the system of (6.9), illustrating the potential of a complex PID controller

Of course, this can be improved upon by increasing the controller's overall gain τ_d is increased. Setting $\tau_d = 1\,\text{s}$ while keeping the locations of $n_{1,2}$ unaltered can only be achieved by:

$$\tau_d = 1\,\text{s} \qquad K_p = 8 \qquad \tau_i = 0.05\,\text{s}$$

Running the Matlab script above with adapted parameters reveals

$$r_{1,2} = -4.1 \pm 2.1j \qquad r_3 = -56.8$$

and a decreased settling time of $T_{s,2\%} = 0.275\,\text{s}$.

If we keep increasing the controller's gain, the roots $r_{1,2}$ will end up at $n_{1,2}$, allowing to simplify the overall transfer function. Indeed:

$$T(s) = \frac{G_c(s)G(s)}{1 + G_c(s)G(s)} = \frac{60\tau_d(s - n_1)(s - n_2)}{(s - r_1)(s - r_2)(s - r_3)} \approx \frac{60\tau_d}{(s - r_3)}$$

In theory we can make τ_d infinitely large without incurring stability problems. However, in practice clipping of the signal after the controller will occur, annulling the advantage of a large τ_d.

6.5.6 In the time domain

In general, for a cascaded controller, it makes sense to:

- Keep the I-action in the forward path
- Move the D-action to the feedback path

Chapter 7

Digital control systems

7.1 Introduction

So far, we only considered using analog controllers to adapt the properties of the systems that we want to control. However, using a digital computer as controller is also an option. This leads to the straightforward schematic to control a linear process $G(s)$ that can be measured with a sensor with transfer function $G_s(s)$ of Fig. 7.1.

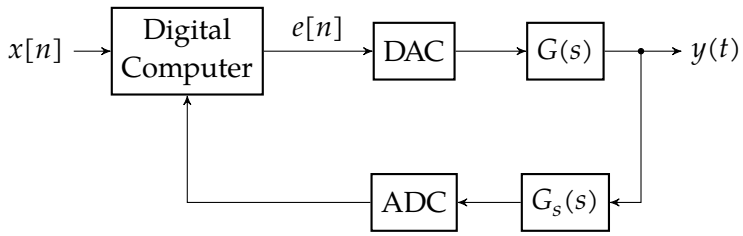


Figure 7.1: Basic block diagram of a linear digital control system

Both process and sensor are analog. We therefore need to convert the output of the computer $e[n]$ back to the analog domain using a DAC and we need to convert the output of sensor to a digital signal for the computer to be able to process it. Also the control input $x[n]$ that is sent to the digital computer is digital.

Using this setup has many advantages:

- The DAC and ADC can be positioned very close to the process while the computer can be positioned in a control room, quite distant from the process. The fact that the digital transmission of data can be made more robust than analog transmission, allows for this. In this way, the controllers of different processes can be centralized in a single control room.
- We can program the computer to perform any control action we'd like (phase lead, phase lag, P, I, PI, PD, PID, ...). Even model predictive control strategies can be implemented!
- Because of the fact that our controller lives in software, it can be very easily adapted.
- The digital controller is very stable over time compared to the gradual degradation of an analog controller due to component aging. The disadvantage is that end of life occurs often very sudden, while an analog controller may reveal its degradation early by poor process performance.

The main question is: how can we analyze this mixed-signal setup to assess its stability, settling time, overshoot, a.s.o. To come up with a satisfactory answer to this question, we need to revise our understanding of sampling and reconstruction and the relationship between the $\mathcal{L}$- and the $\mathcal{Z}$-transform.

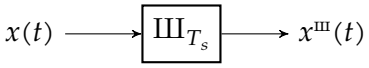
7.2 Sampling and reconstruction in the $\mathcal{L}$-domain

Sampling Consider a continuous-time signal $x(t)$ that we will sample at a rate $1/T_s$, resulting in a discrete-time signal $x^{\text{ш}}(t)$. The ш-exponent denotes the fact that the signal has been sampled. The round brackets and the continuous time argument t imply that - though we sampled - this is still a continuous-time model of that signal.

In your basic course on DSP you have seen that sampling corresponds to multiplication with a unit impulse train:[1]

$$x^{\text{ш}}(t) = x(t) \cdot \text{ш}_{T_s}(t) = x(t) \cdot \sum_{n=-\infty}^{+\infty} \delta(t - nT_s) \tag{7.1}$$

Graphically, we denote the sampling process as:



Let's call the block in this diagram a *Sha-block*.

Note that the above equation is a model for the discrete-time signal in the continuous time domain.
The discrete time-domain signal $x[n]$ itself can we written as:

$$x[n] = x(nT_s)$$

Let's investigate the Laplace transform of this sampled-data signal. We start from its definition:

$$\begin{aligned} x^{\text{ш}}(t) &= x(t) \cdot \sum_{n=-\infty}^{+\infty} \delta(t - nT_s) \\ &\Big\downarrow \text{ Fourier series decomposition: } \sum_{n=-\infty}^{+\infty} \delta(t - nT_s) = \frac{1}{T_s} \sum_{n=-\infty}^{+\infty} \mathrm{e}^{-jn\omega_s t} \\ &= x(t) \cdot \frac{1}{T_s} \sum_{n=-\infty}^{+\infty} \mathrm{e}^{-jn\omega_s t} \end{aligned}$$

Transforming this equation to the Laplace domain results in:

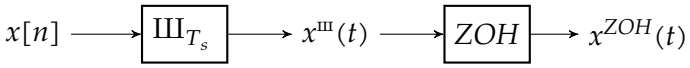
$$\begin{aligned} X^{\text{ш}}(s) &= \int_0^{+\infty} \frac{1}{T_s} x(t) \sum_{n=-\infty}^{+\infty} \mathrm{e}^{-jn\omega_s t} \mathrm{e}^{-st} \, \mathrm{d}t \\ &= \frac{1}{T_s} \sum_{n=-\infty}^{+\infty} \int_0^{+\infty} x(t) \, \mathrm{e}^{-(s+jn\omega_s)t} \, \mathrm{d}t = \frac{1}{T_s} \sum_{n=-\infty}^{+\infty} X(s + jn\omega_s) \end{aligned} \tag{7.2}$$

Note how this equation confirms our understanding of the periodic nature (with period ω_s) of the frequency spectrum for a sampled-data signal.

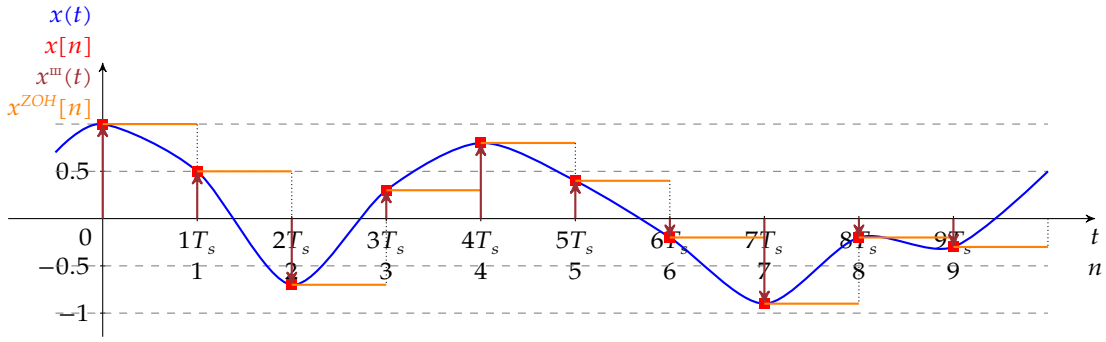
[1] Note that we did not add the appropriate extra factor T_s to the sampling Dirac comb. Remember that we also omitted this factor in the derivation of the discrete-time Fourier transform (DtFT) from the Fourier transform.

Reconstruction If we want to reconstruct the original signal from the samples, for practical reasons (we can't really make Dirac impulses), we use the Zeroth-Order Hold approximation (ZOH). The Dirac impulses are replaced by rectangles with the same area.

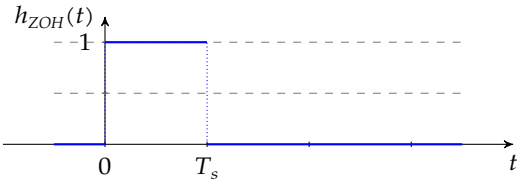
Graphically:



Note that we reused the same block diagram symbol for the conversion of a number series into a sampled-data signal, as we did to convert a continuous time-domain signal into a sampled-data signal. Given the fact that the end result of both operations is the same, that makes sense.



It is easily seen that the impulse response corresponding to the zeroth-order hold effect is a rectangle:



The corresponding transfer function in the Laplace domain can be directly obtained, starting from the time-domain description:

$$\begin{aligned} h_{ZOH}(t) &= u(t) - u(t - T_s) \\ &\downarrow \mathcal{L} \\ H_{ZOH}(s) &= \frac{1}{s} - \frac{1}{s}\mathrm{e}^{-sT_s} = \frac{1 - \mathrm{e}^{-sT}}{s} \end{aligned}$$

7.3 Equivalence of $\mathcal{L}$- and $\mathcal{Z}$-transform

7.3.1 Equivalence for sampled-data signals

When we derived the $\mathcal{Z}$-transform, we started out from the DtFT. However, we also might start from the Laplace transform of our discretized signal $x^{\text{Ш}}(t)$ according to (7.1).

$$x^{\text{Ш}}(t) = x(t) \cdot \sum_{n=-\infty}^{+\infty} \delta(t - nT_s)$$

Assuming that $x(t)$ is causal, we can easily calculate its $\mathcal{L}$-transform:

$$\begin{aligned} X^{\text{III}}(s) &= \int_0^{+\infty} x(t) \cdot \sum_{n=-\infty}^{+\infty} \delta(t - nT_s)\ \mathrm{e}^{-st}\,\mathrm{d}t \\ &= \sum_{n=-\infty}^{+\infty} \int_0^{+\infty} x(t)\,\delta(t - nT_s)\ \mathrm{e}^{-st}\,\mathrm{d}t \\ &\downarrow \text{sifting property of } \delta(t - nT_s) \\ &= \sum_{n=0}^{+\infty} x(nT_s)\ \mathrm{e}^{-snT_s} \\ &\downarrow \text{substitution: } s = \frac{\ln z}{T_s} \\ X(z) &= \sum_{n=0}^{+\infty} x(nT_s) z^{-n} \end{aligned}$$

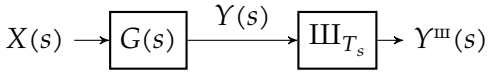
This is an alternative route to derive the $\mathcal{Z}$-transform. It's significance is that it illustrates that for a sampled-data signal the $\mathcal{Z}$-transform contains the full $\mathcal{L}$-domain information be it in a disguised (transformed) way. Therefore, for sampled-data signals, the $\mathcal{Z}$-transform and the $\mathcal{L}$-transform are truly equivalent:

$$X^{\text{III}}(s) = X(z) \qquad \text{knowing that } s = \frac{\ln z}{T_s}$$

7.3.2 Equivalence for transfer functions

In the previous section, we have shown the equivalence of the $\mathcal{L}$- and $\mathcal{Z}$-transform for sampled-data functions. An obvious question is whether one can find a similar equivalence for transfer functions? As the major prerequisite for equivalence was the sampled nature of the signal, we also assume that sampling will be a prerequisite for transfer functions. We can consider sampling at the input of a block, at the output of a block or at both input and output.

Sampling at the output Consider the system below which filters an input signal $X(s)$ using a transfer function $G(s)$ and produces an output $Y(s)$ that is sampled into $Y^{\text{III}}(s)$:



We can derive:

$$Y(s) = G(s) \cdot X(s)$$

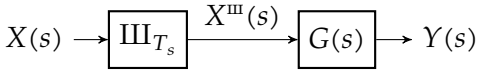
and using (7.2), we easily obtain:

$$Y^{\text{III}}(s) = \frac{1}{T_s} \sum_{n=-\infty}^{+\infty} Y(s + jn\omega_s) = \frac{1}{T_s} \sum_{n=-\infty}^{+\infty} G(s + jn\omega_s) X(s + jn\omega_s)$$

Note that we cannot apply the conclusion of the previous section to replace these Laplace expressions by Z-expressions, as the equivalence between Laplace and $\mathcal{Z}$-transform only holds for sampled-data signals. The time-domain counterparts of $G(s)$, $X(s)$ or their translated products (as the appear in the equation above) are not sampled-data signals.

Therefore, let's consider sampling at the input.

Sampling at the input Consider the system below which first samples an input signal $X(s)$ into $X^{\text{III}}(s)$ to process it using the transfer function $G(s)$ to obtain an output signal $Y(s)$.



We can derive:

$$Y(s) = G(s) \cdot X^{\text{III}}(s)$$

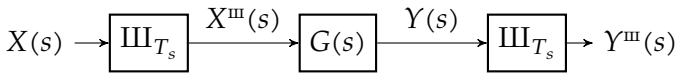
and using (7.2), we easily obtain:

$$Y(s) = G(s) \cdot \underbrace{\frac{1}{T_s} \sum_{n=-\infty}^{+\infty} X(s + jn\omega_s)}_{\equiv X(z)}$$

Nice, but it does not get us very far. We're still stuck on the fact that we don't know whether $G(s)$ is equivalent with $G(z)$, because we cannot state that $G(s)$ is the Laplace transform of a sampled-data impulse response. Most likely, it won't be.

In order to experience the fun of the Laplace-Z-equivalence party in full, we need to sample both at the input as well as at the output. The latter can even be considered to be a virtual sampling, in the sense that we don't have to provide for a physical ADC. We just can only consider the output at the sample time instances.

Sampling at both the input and the output Consider the system below which first samples an input signal $X(s)$ into $X^{\text{III}}(s)$ to process it using the transfer function $G(s)$ to obtain an output signal $Y(s)$ that on itself again is sampled into $Y^{\text{III}}(s)$.



Here comes the fun:

$$\begin{aligned} Y(s) &= G(s) \cdot X^{\text{III}}(s) \\ &= G(s) \cdot \frac{1}{T_s} \sum_{n=-\infty}^{+\infty} X(s + jn\omega_s) \end{aligned} \tag{7.3}$$

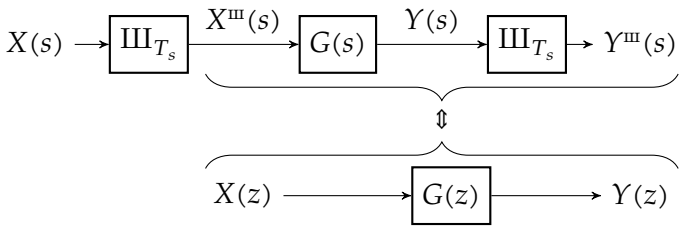
Sampling the latter results in:

$$Y^{\text{III}}(s) = \frac{1}{T_s} \sum_{m=-\infty}^{+\infty} Y(s + jm\omega_s)$$

If we now use (7.3) and replace $s \mapsto s + jm\omega_s$, we obtain:

$$\begin{aligned}
Y^{\text{III}}(s) &= \frac{1}{T_s}\sum_{m=-\infty}^{+\infty} G(s+jm\omega_s)\frac{1}{T_s}\sum_{n=-\infty}^{+\infty} X(s+jm\omega_s+jn\omega_s) \\
&\quad \downarrow \text{substitute } n \text{ by } k \text{ using } k = m+n \\
&= \frac{1}{T_s}\sum_{m=-\infty}^{+\infty} G(s+jm\omega_s)\underbrace{\frac{1}{T_s}\sum_{k=-\infty}^{+\infty} X(s+jk\omega_s)}_{\equiv X^{\text{III}}(s)\neq f(m)} \\
&= \underbrace{\left(\frac{1}{T_s}\sum_{m=-\infty}^{+\infty} G(s+jm\omega_s)\right)}_{\equiv G^{\text{III}}(s)} X^{\text{III}}(s) \\
&= G^{\text{III}}(s)\cdot X^{\text{III}}(s) \\
&= G(z)\cdot X(z) \qquad \text{knowing that: } s = \frac{\ln z}{T_s}
\end{aligned}$$

or even more concise: $Y(z) = G(z)\cdot X(z)$.
Graphically:



Avoiding mistakes Note that if we replace $G(s)$ by a cascade of two systems $A(s)$ and $B(s)$, i.e. $G(s) = A(s)\cdot B(s)$, we only can write

$$Y(z) = G(z)\cdot X(z)$$

and specifically not

$$Y(z) = A(z)\cdot B(z)\cdot X(z)$$

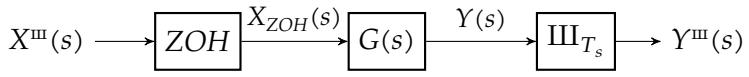
i.e. we need to consider $G(s)$ as a whole. This is sometimes referred to as "collapsing A and B into G". To this end, the following compact notation (joining the A and B together into one glyph AB) might come in handy:

$$G^{\text{III}}(s) = (A(s)B(s))^{\text{III}} = AB^{\text{III}}(s) = AB(z)$$

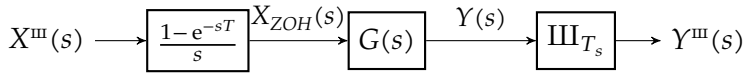
The essence to remember: calculate the product of A and B in the $\mathcal{L}$-domain first, and then transform to the $\mathcal{Z}$-domain.

7.4 The ZOH-method

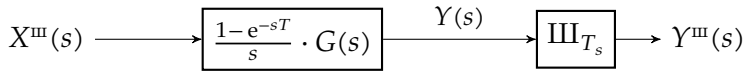
Now, let's consider a practical system in which we have a discrete-time input signal $X(z)$ (or its Laplace counterpart $X^{\text{III}}(s)$) that we will convert to analog using a ZOH setup, send through a process $G(s)$, to finally consider its output $Y(s)$ on the sampling time instances as $Y^{\text{III}}(s)$:



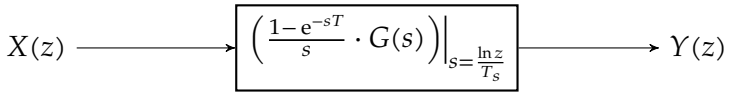
This can be expanded into:



To prepare for $\mathcal{L}/\mathcal{Z}$ equivalence, we collapse the ZOH block and $G(s)$:



Or in the $\mathcal{Z}$-domain:



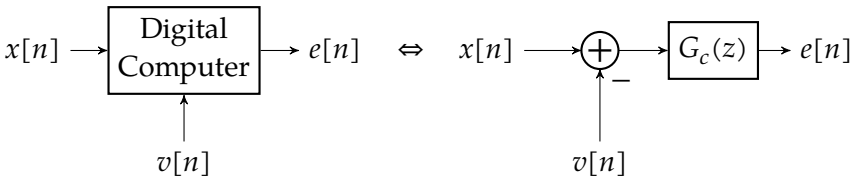
Applying this kind of transformation, is the so-called *ZOH-method*. It is a method to transform an analog system that is fed with a discrete-time signal (using the ZOH approximation) and sampled at the output, to a digital representation in the $\mathcal{Z}$-domain.

It is actually performed by using a mapping table between the Laplace and the Z-transform (see Table 7.1). The table looks a bit deviant w.r.t. the ordinary transformation tables in the sense that it explicitly mentions T_s.

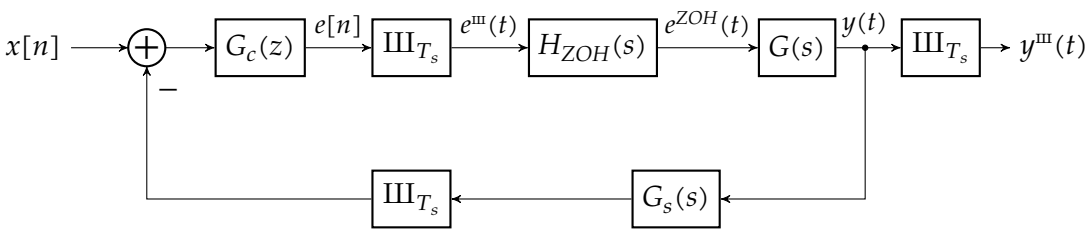
You can use the Matlab method `c2d` in its default `'ZOH'` mode to do the work for you.

7.5 Mixed-signal control systems

We can use the ZOH-method to analyze mixed-signal control system of of Fig. 7.1. Let's assume our digital computer (on the left below) implements the following digital controller (on the right below):



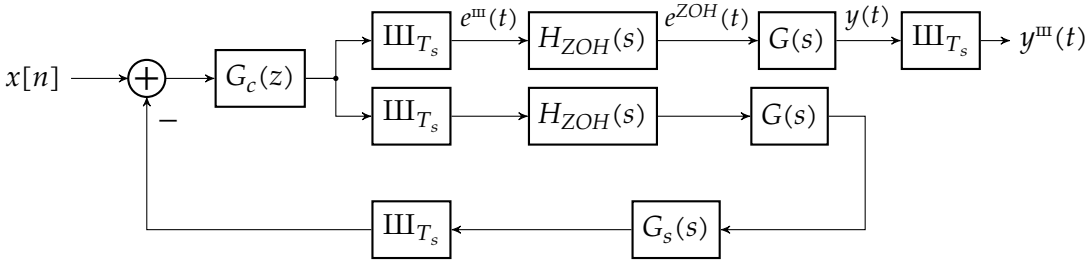
Using the notation used in the previous sections, we can redraw Fig. 7.1 as follows:



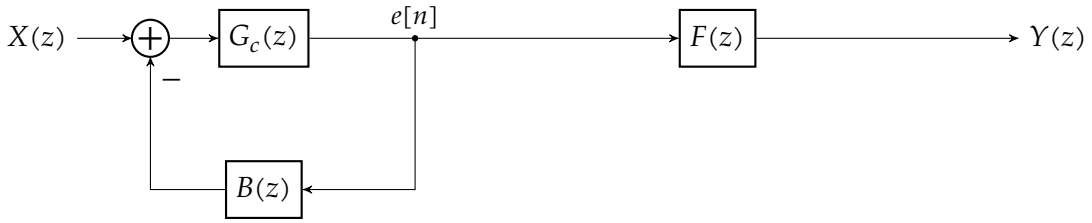
$x(t)$	$X(s)$	$X(z)$
1	$\frac{1}{s}$	$\frac{z}{z-1}$
t	$\frac{1}{s^2}$	$\frac{T_s z}{(z-1)^2}$
t^2	$\frac{2}{s^3}$	$\frac{T_s^2 z(z+1)}{(z-1)^3}$
e^{-at}	$\frac{1}{s+a}$	$\frac{z}{z-\mathrm{e}^{-aT_s}}$
$1-\mathrm{e}^{-at}$	$\frac{1}{s}\frac{a}{s+a}$	$\frac{z(1-\mathrm{e}^{-aT_s})}{(z-1)(z-\mathrm{e}^{-aT_s})}$
$t\,\mathrm{e}^{-at}$	$\frac{1}{(s+a)^2}$	$\frac{T_s z\,\mathrm{e}^{-aT_s}}{(z-\mathrm{e}^{-aT_s})^2}$
$t^2\,\mathrm{e}^{-at}$	$\frac{2}{(s+a)^3}$	$\frac{T_s^2\,\mathrm{e}^{-aT_s} z(z+\mathrm{e}^{-aT_s})}{(z-\mathrm{e}^{-aT_s})^3}$
$b\,\mathrm{e}^{-bt}-a\,\mathrm{e}^{-at}$	$\frac{(b-a)s}{(s+a)(s+b)}$	$\frac{z(z(b-a)-(b\,\mathrm{e}^{-aT_s}-a\,\mathrm{e}^{-bT_s}))}{(z-\mathrm{e}^{-aT_s})(z-\mathrm{e}^{-bT_s})}$
$\sin\omega t$	$\frac{\omega}{s^2+\omega^2}$	$\frac{z\sin\omega T_s}{z^2-2z\cos\omega T_s+1}$
$\cos\omega t$	$\frac{s}{s^2+\omega^2}$	$\frac{z(z-\cos\omega T_s)}{z^2-2z\cos\omega T_s+1}$
$\mathrm{e}^{-at}\sin\omega t$	$\frac{\omega}{(s+a)^2+\omega^2}$	$\frac{z\,\mathrm{e}^{-aT_s}\sin\omega T_s}{z^2-2z\,\mathrm{e}^{-aT_s}\cos\omega T_s+\mathrm{e}^{-2aT_s}}$
$\mathrm{e}^{-at}\cos\omega t$	$\frac{s+a}{(s+a)^2+\omega^2}$	$\frac{z(z-\mathrm{e}^{-aT_s}\cos\omega T_s)}{z^2-2z\,\mathrm{e}^{-aT_s}\cos\omega T_s+\mathrm{e}^{-2aT_s}}$
$1-\mathrm{e}^{-at}\left(\cos\omega t+\frac{a}{\omega}\sin\omega t\right)$	$\frac{1}{s}\frac{a^2+\omega^2}{(s+a)^2+\omega^2}$	$\frac{z(Az+b)}{(z-1)(z^2-2z\,\mathrm{e}^{-aT_s}\cos\omega T_s+\mathrm{e}^{-2aT_s})}$
		$A=1-\mathrm{e}^{-aT_s}\cos\omega T_s-\frac{a}{\omega}\mathrm{e}^{-aT_s}\sin\omega T_s$
		$B=\mathrm{e}^{-2aT_s}-\mathrm{e}^{-aT_s}\cos\omega T_s+\frac{a}{\omega}\mathrm{e}^{-aT_s}\sin\omega T_s$

Table 7.1: $\mathcal{L}$ to $\mathcal{Z}$ mapping table assuming causal $x(t)$

Our goal is to identify straight paths in between two 'sha-blocks'. To reveal the appropriate paths, we shift the junction of $y(t)$ (where the feedback path starts) to the left:



Assuming $F(s) = H_{ZOH}(s) \cdot G(s)$, and $B(s) = H_{ZOH}(s) \cdot G(s) \cdot G_s(s)$ we can now further use the equivalence between $\mathcal{L}$- and $\mathcal{Z}$-domain, to obtain:



This allows considering the loop gain

$$L(z) = G_c(z) \cdot B(z)$$

for stability and the overall transfer function

$$T(z) = \frac{G_c(z)F(z)}{1 + G_c(z)B(z)}$$

to assess the overall behavior. Note that in this case also the poles of $F(z)$ should be checked for stability. In addition, it might be that one of the poles of $F(z)$ is canceled by a zero of the preceding loop or vice versa.

The analysis of these discrete-time models for the mixed-signal control systems is most similar to their analog counterparts. We can use:

- the root locus method (in the $\mathcal{L}$-domain); we only need to remember that stable systems require poles within the unit circle.
- the Nyquist plot (in the $\mathcal{F}$-domain); the 'zero-catching contour' that is the center of Nyquist's application of the Cauchy's theorem, is now located in the $\mathcal{Z}$-domain and corresponds to the unit circle in counterclockwise direction. Again, Nyquist's criterion is not to circle the point -1 in the $\mathcal{Z}$-domain. Gain and phase margin
- Because of the similarity in Nyquist's criterion, it also makes sense to inspect phase and gain margin on a Bode plot or on a Nichols plot.

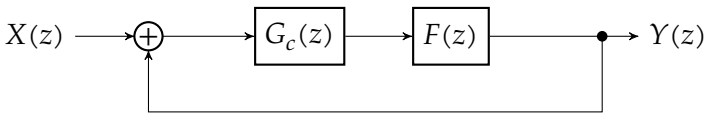
7.6 Example

Consider the mixed-signal control system of Fig. 7.1 with

$$G(s) = \frac{1}{s(s+1)} \qquad G_s(s) = 1$$

and the digital computer implementing a digital controller with transfer function $G_c(z)$.

This leads to the following discrete-time model for the system:



with $F(z)$ the $\mathcal{Z}$-transform corresponding to $F(s) = \frac{1-\mathrm{e}^{-sT_s}}{s}G(s)$.

Let's therefore determine $F(z)$ by partial fraction decomposition of $F(s)$.

$$\begin{aligned} F(s) &= \frac{1-\mathrm{e}^{-sT_s}}{s}\frac{1}{s(s+1)} \\ &= \left(1-\mathrm{e}^{-sT_s}\right)\frac{1}{s^2(s+1)} \\ &= \left(1-\mathrm{e}^{-sT_s}\right)\left(\frac{1}{s^2}+\frac{-1}{s}+\frac{1}{s+1}\right) \end{aligned}$$

Using Table 7.1 we can find the corresponding $F(z)$ to be

$$\begin{aligned} F(z) &= \left(1-z^{-1}\right)\left(\frac{T_s z}{(z-1)^2}-\frac{z}{z-1}+\frac{z}{z-\mathrm{e}^{-T_s}}\right) \\ &= \frac{z-1}{z}\left(\frac{T_s z}{(z-1)^2}-\frac{z}{z-1}+\frac{z}{z-\mathrm{e}^{-T_s}}\right) \\ &= \frac{T_s}{z-1}-1+\frac{z-1}{z-\mathrm{e}^{-T_s}} \\ &= \frac{(T_s+\mathrm{e}^{-T_s}-1)z-(T_s\,\mathrm{e}^{-T_s}+\mathrm{e}^{-T_s}-1)}{z^2-(1+\mathrm{e}^{-T_s})z+\mathrm{e}^{-T_s}} \end{aligned}$$

Assuming $T_s = 1\,\mathrm{s}$ we can quantify this transfer function to:

$$F(z) = \frac{0.3679z + 0.2642}{z^2 - 1.3679z + 0.3679}$$

Proportional controller Assuming a proportional controller, i.e. $G_c(z) = K$, this results in a loop gain of:

$$L(z) = KF(z) = K\frac{0.3679z + 0.2642}{z^2 - 1.3679z + 0.3679}$$

and an overall transfer function of:

$$\begin{aligned} T(z) &= \frac{L(z)}{1+L(z)} \\ &= \frac{K(0.3679z + 0.2642)}{z^2 - 1.3679z + 0.3679 + K(0.3679z + 0.2642)} \\ &= \frac{K(0.3679z + 0.2642)}{z^2 + (0.3679 \cdot K - 1.3679)z + 0.2642 \cdot K + 0.3679} \end{aligned} \tag{7.4}$$

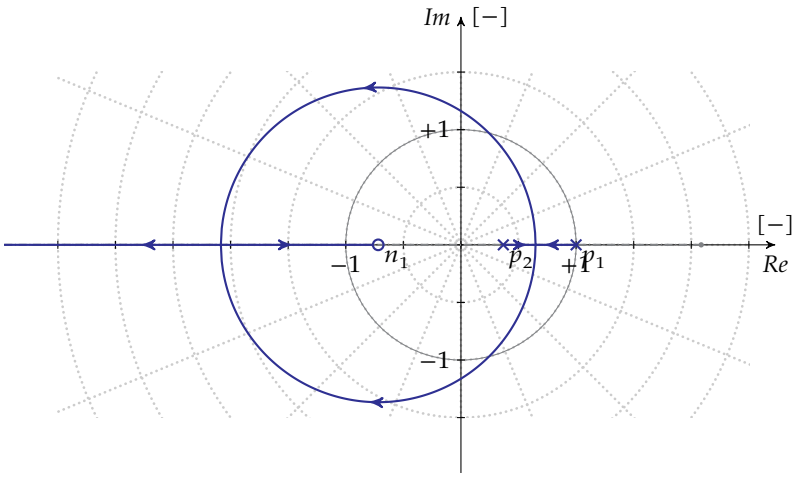
Let's draw a root locus for this system. The loop gain has two poles:

$$\begin{aligned} p_1 &= 1 \\ p_2 &= 0.3679 \end{aligned}$$

and two zeros:

$$\begin{aligned} z_1 &= -0.7183 \\ z_2 &= -\infty \end{aligned}$$

The root locus plot corresponding to this pole-zero-configuration can be easily sketched to be:



As we can clearly see, for low values of K the system is stable, until the root locus crosses the unit circle. At that point the (complex conjugate) poles of the overall transfer function $T(z)$ have a magnitude of one. If we denote these roots as $e^{\pm j\phi}$, then we can write the denominator of $T(z)$ as:

$$D(z) = (z - e^{j\phi})(z - e^{-j\phi}) = z^2 - 2\cos(\phi)z + 1$$

Equating this expression to the denominator of (7.4), results in two equations:

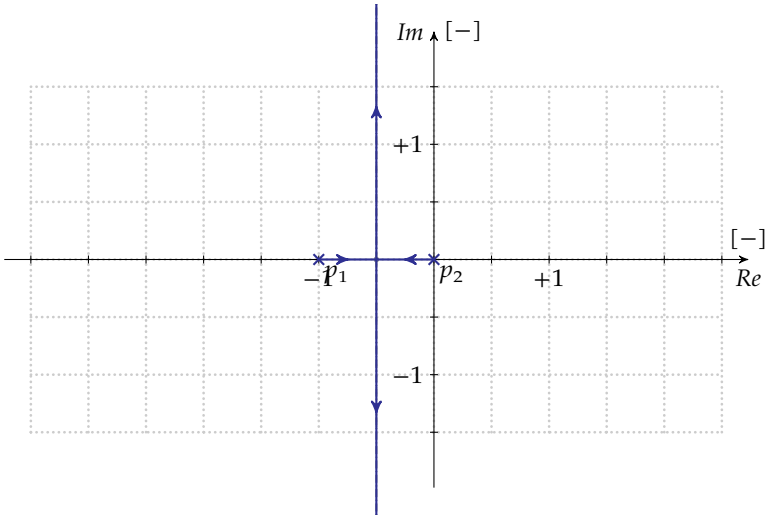
$$\begin{cases} -2\cos\phi = 0.3679 \cdot K - 1.3679 \\ \qquad 1 = 0.2642 \cdot K + 0.3679 \end{cases}$$

This allows for the calculation of K and ϕ, resulting in:

$$\begin{cases} K = 2.3925 \\ \phi = 1.324\,5\,\text{rad} = 75.886^\circ \end{cases}$$

The conclusion is simple: to obtain a stable system, we must ensure $0 < K < 2.3925$.

However, a final consideration must be made. Note how the continuous-time second order system we started from, had the following root locus plot, and therefore could not be unstable, except for when $K = 0$ (marginally stable).



How come the discretized system does become unstable for large values of K? The culprit is the discretization itself. The ZOH block introduces a delay in the system of $T_s/2$ on average. It is this delay that causes an extra phase lag and therefore results in an unstable system. You can easily check yourself, that if we reduce T_s to 100 ms, the maximal K increases by a factor 8.5.

In any case, a smaller T_s is to be preferred in view of stability. However, a faster sampling rate corresponds to more expensive hardware and more data to be transferred between the control computer (in the control room) and the DAC/ADC (in the field). As a rule of thumb: choose T_s to be 10 times smaller than the smallest relevant time constant in the system (i.e. among the time constants that determine the phase margin).

First-order controller So far, our computer only acted as a proportional controller. We can do better of course. Programming a controller with poles and zeros is not difficult at all.

Consider using a first-order single pole-zero controller:

$$G_c(z) = K\frac{z-a}{z-b}$$

How to program it in the time domain results from transferring this system description to the time domain:

$$\begin{aligned}
\frac{Y(z)}{X(z)} &= K\frac{z-a}{z-b}\cdot\frac{z^{-1}}{z^{-1}} \\
&= K\frac{1-az^{-1}}{1-bz^{-1}} \\
Y(z) &= bz^{-1}Y(z) + K(1-az^{-1})X(z) \\
&\downarrow \quad \mathcal{Z}^{-1} \\
y[n] &= by[n-1] + K(x[n] - ax[n-1])
\end{aligned}$$

Implementing this difference equation at every sample clock cycle should be a piece of cake. Of course, we still need to design the controller, i.e. determine appopriate values for K, a and b.

Let's assume our target is to obtain a second-order system with $\omega_n = \omega_s/4$ and $\zeta = 1/2$, assuming $\omega_s = 2\pi/T_s$ with T_s the sampling period. The characteristic polynomial of such a second-order system in the Laplace domain is:

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

Its poles are:

$$s_{1,2} = \omega_n\left(-\zeta \pm j\sqrt{1-\zeta^2}\right)$$

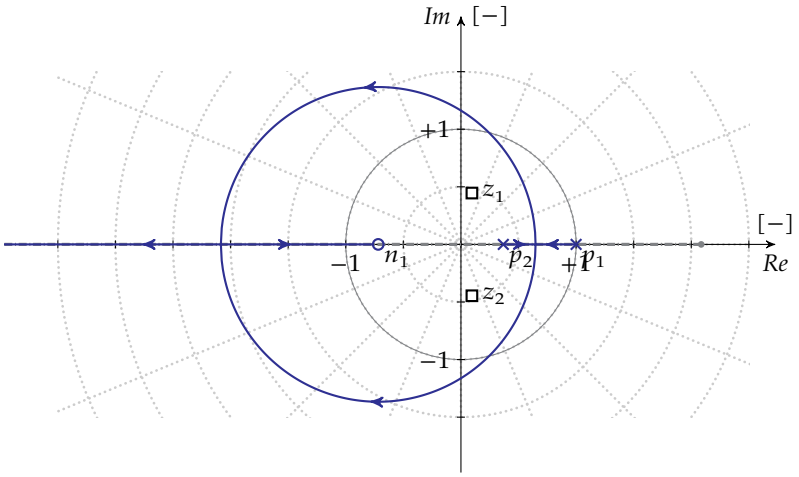
The corresponding pole locations in the Z domain can be calculated using $z = \mathrm{e}^{sT_s}$, i.e.

$$\begin{aligned} z_{1,2} = \mathrm{e}^{s_{1,2}T_s} &= \mathrm{e}^{-\zeta\omega_n T_s}\,\mathrm{e}^{\pm j\omega_n T_s\sqrt{1-\zeta^2}} \\ &= \mathrm{e}^{-\frac{\pi}{4}}\,\mathrm{e}^{\pm j\frac{\pi\sqrt{3}}{4}} = 0.456\,\mathrm{e}^{\pm j1.36} = 0.0952 \pm j0.446 \end{aligned}$$

This corresponds to the following characteristic polynomial:

$$z^2 - 0.1905z + 0.2079 \tag{7.5}$$

Reconsider the original root locus plot for the proportional controller below. We have indicated the desired location for the poles as we determined them above:



To achieve this, we propose $a = p_2$, such that the controller's zero cancels out the leftmost process pole.

The loop gain becomes:

$$L(z) = G_c(z)G(z) = K\frac{z-0.3679}{z-b}\frac{0.3679z+0.2642}{(z-1)(z-0.3679)} = K\frac{0.3679z+0.2642}{(z-b)(z-1)}$$

The controller's pole can now be set at any desired real location. The result of this, is that the overall shape of the root locus plot will remain the same. Putting the controller's pole to the

left of p_2 will allow to shift the root locus circle to the left and make it smaller such that it crosses the desired overall pole locations. To this end, we calculate the overall transfer function

$$T(z) = \frac{L(z)}{1+L(z)} = \frac{K(0.3679z + 0.2642)}{(z-b)(z-1) + K(0.3679z + 0.2642)} = \frac{K(0.3679z + 0.2642)}{z^2 - (b + 1 - 0.3679K)z + b + 0.2642K}$$

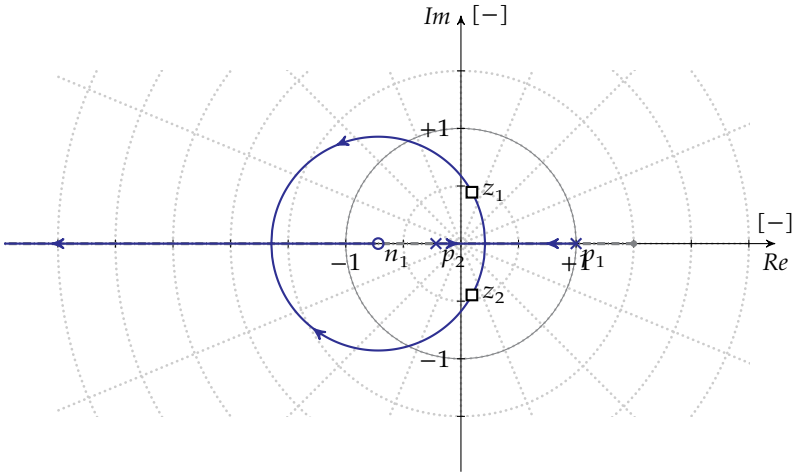
and equate it's characteristic polynomial to (7.5):

$$z^2 - (b + 1 - 0.3679K)z + b + 0.2642K = z^2 - 0.1905z + 0.2079$$

This leads to:

$$\begin{cases} b + 1 - 0.3679K = 0.1905 \\ b + 0.2642K = 0.2079 \end{cases} \Leftrightarrow \begin{cases} b = -0.217 \\ K = 1.61 \end{cases}$$

The resulting root locus can be found below:



The control equation to be implemented is:

$$y[n] = -0.217y[n-1] + 1.61x[n] - 0.592x[n-1])$$

7.7 Conclusion

Using digital controllers has many advantages. Using the ZOH-method we can analyze mixed-signal systems where the process is in the analog domain and the controller is in the discrete-time domain. This allows for designing controllers in the Z-domain. Judiciously placing controller poles and zeros allows taming the process to be stable, yet responsive.

g

Chapter 8

State-space control

8.1 Introduction

Consider as an example the circuit below with as input the current source and as output the voltage over the resistor:

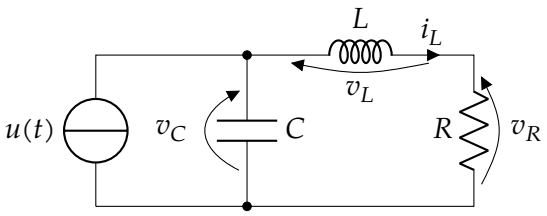


Figure 8.1: Example network

The governing differential equation is:

$$LC\frac{\mathrm{d}^2}{\mathrm{d}t^2}v_R(t) + RC\frac{\mathrm{d}}{\mathrm{d}t}v_R(t) + v_R(t) = Ru(t)$$

It must be said: it took quite some effort to derive this equation. Try yourself on a rainy day! However, there is a simpler approach: we can describe this system using a set of n first-order linear differential equations.

In the case of our example, this is easily done by

1. selecting as *key variables* the current through the inductor and the voltage over the capacitor
2. writing down Kirchhoff's equations:

$$\begin{cases} u(t) - i_C(t) - i_L(t) = 0 & \text{(KCL)} \\ v_C(t) - v_L(t) - V_R(t) = 0 & \text{(KVL)} \end{cases}$$

3. writing down the branch equations of the components

$$\begin{cases} i_c(t) = C\dfrac{\mathrm{d}v_C(t)}{\mathrm{d}t} \\ v_L(t) = L\dfrac{\mathrm{d}i_L(t)}{\mathrm{d}t} \\ v_R(t) = Ri_L(t) \end{cases}$$

4. use this set of five equations to eliminate all but the key variables, their derivatives and the input.

This leads to:

$$u(t) - i_L(t) = C\frac{\mathrm{d}v_C(t)}{\mathrm{d}t}$$
$$v_C(t) - Ri_L(t) = L\frac{\mathrm{d}i_L(t)}{\mathrm{d}t}$$

In jargon, we call the key variables the *state variables* and the equations above, the *state-space equations*. Often the state variables are named x_i (omitting the time aspect in the notation) and derivatives are denoted using Newton's dot notation, i.e. $\frac{\mathrm{d}x}{\mathrm{d}t} = \dot{x}$.

In our example, setting $x_1 = v_C(t)$ and $x_2 = i_L(t)$, leads to:

$$\begin{cases} u - x_2 = C\dot{x}_1 \\ x_1 - Rx_2 = L\dot{x}_2 \end{cases}$$

which is often rearranged by only allowing the derivatives in the left-hand sides of the equations:

$$\begin{cases} \dot{x}_1 = -\frac{1}{C}x_2 + \frac{1}{C}u \\ \dot{x}_2 = \frac{1}{L}x_1 - \frac{R}{L}x_2 \end{cases}$$

This system can be most conveniently described in matrix form:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{C} \\ \frac{1}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{C} \\ 0 \end{bmatrix} u \tag{8.1}$$

Using the notational conventions of section A.2 on page 230, we can define:

$$A = \begin{bmatrix} 0 & -\frac{1}{C} \\ \frac{1}{L} & -\frac{R}{L} \end{bmatrix} \qquad b = \begin{bmatrix} \frac{1}{C} \\ 0 \end{bmatrix}$$

We call A the *state matrix*, or *system matrix* and b the *input matrix*. Simply defining

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}$$

allows rewriting (8.1) into

$$\dot{x} = Ax + bu$$

which we will conveniently refer to as the *state equation*.

One important detail is still missing, the output v_R is not present in this description. To this end, we add an *output equation*:

$$v_R(t) = Ri_L(t)$$

or defining $y = v_R(t)$ and considering our state variables, we can write:

$$y = \underbrace{\begin{bmatrix} 0 & R \end{bmatrix}}_{\equiv c^\top} x + \underbrace{0}_{\equiv d} u$$

or in short:

$$y = c^\top x + du$$

This leads to the full state-space description:

$$\begin{cases} \dot{x} = Ax + bu \\ y = c^\top x + du \end{cases}$$

Note that the choice of state-space variables is not unique. One might as well have chosen $x_1 = v_C(t)$ and $x_2 = v_L(t)$ as state-space variables. This will lead to the following state-space description (try yourself!):

$$\dot{x} = Ax + bu$$
$$y = c^\top x + du$$

with

$$A = \begin{bmatrix} -\frac{1}{RC} & \frac{1}{RC} \\ -\frac{1}{RC} & \frac{1}{RC} - \frac{R}{L} \end{bmatrix} \qquad b = \begin{bmatrix} \frac{1}{C} \\ \frac{1}{C} \end{bmatrix}$$

and

$$c^\top = \begin{bmatrix} 1 & -1 \end{bmatrix} \qquad d = 0$$

Note that, though the structure of the description is the same, the composing matrices are totally different!

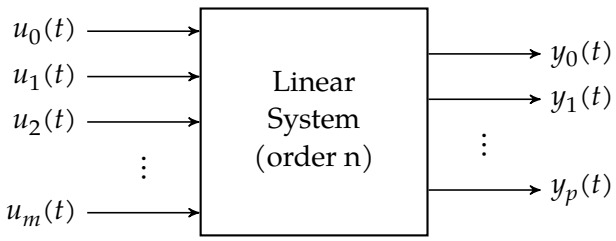
As a side note: there are proper ways to get a good set of state variables for linear electrical circuits

Systematic state-space approach for linear electrical circuits

1. Select all capacitors and all inductors in the circuit. These form your list of candidate elements.
2. Short all voltage sources in the circuit. If the capacitors on your list form a loop, remove one of the capacitors from your candidate list to break the loop.
3. Remove all current sources from the circuit. If removing all the inductors from your circuit, splits it in two or more unconnected parts, remove a minimal amount of inductors from your list to avoid the splitting effect.
4. Now select the voltages over the capacitors and the currents through the inductors that remain on your list as state variables.

8.2 State-space description for MIMO systems

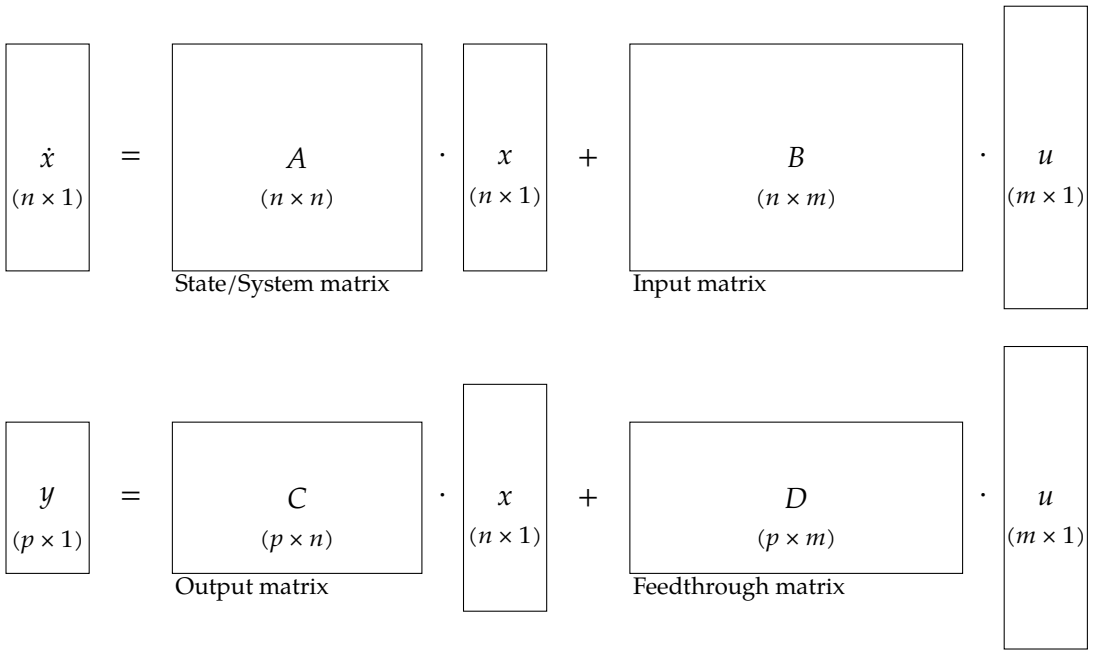
Consider the generic linear MIMO system below:



It has m inputs and p outputs. We can write the dynamic behavior of this system using the following first-order differential state-space description:

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

The dimensions of the vectors and the matrices appearing in this equation have been indicated on the illustration below together with their proper name:



This is a very compact and powerful description for MIMO linear systems.

8.3 State-space description for SISO systems

Though a state-space description is very convenient in the case of MIMO systems, it also makes sense to use it in case of a SISO system.

A number of questions pop up in this case:

- Is there a relationship with the concept of a transfer function?
- Is there a simple stability criterion given a state-space description?

The answer is 'yes' in both cases.

8.3.1 Relationship with transfer functions

We will elaborate the relationship between transfer functions and the state-space description, by showing how to go from one to the other and vice versa.

From transfer function to state-space description

The problem to solve is: given a transfer function, find *a* corresponding state-space description. It is important to stress the *a* in the previous sentence, as a state-space description is not unique.

We can use different techniques:

- the direct form II method
- the transposed form II method
- the method of the partial fraction expansion

Other options exist (e.g. direct and transposed form I, but these are less compact).

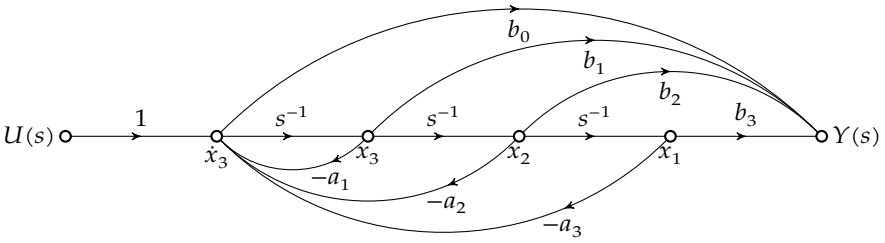
We will discuss these methods using a third-order system as an example. This example can be easily generalized to arbitrary order. The transfer function of our system is:

$$T(s) = \frac{b_0 s^3 + b_1 s^2 + b_2 s + b_3}{s^3 + a_1 s^2 + a_2 s + a_3} \tag{8.2}$$

The direct form II method By dividing numerator and denominator by s^3, we can elaborate:

$$\frac{Y(s)}{X(s)} = \frac{b_0 + b_1 s^{-1} + b_2 s^{-2} + b_3 s^{-3}}{1 - (-a_1 s^{-1} - a_2 s^{-2} - a_3 s^{-3})}$$

As one can check using Mason's rule, this corresponds to the signal flow graph below:



Because a lot of the arrows are departing from the intermediate nodes, we'd like to have them in the right-hand side of the state space equations and therefore, we taken them to be the state variables.

This allows writing the following equations:

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3 \\
\dot{x}_3 &= -a_3 x_1 - a_2 x_2 - a_1 x_3 + u \\
y &= b_3 x_1 + b_2 x_2 + b_1 x_3 + b_0 \dot{x}_3 = (b_3 - b_0 a_3) x_1 + (b_2 - b_0 a_2) x_2 + (b_1 - b_0 a_1) x_3 + b_0 u
\end{aligned}$$

To this end, we used the fact that multiplication with s in the Laplace domain corresponds to derivation w.r.t. time in the time domain.

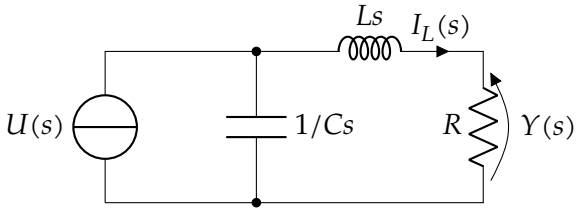
This leads to the following state space description:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ -a_3 & -a_2 & -a_1\end{bmatrix} \cdot \begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 1\end{bmatrix} \cdot u$$

$$y = \begin{bmatrix}b_3 - b_0a_3 & b_2 - b_0a_2 & b_1 - b_0a_1\end{bmatrix} \cdot \begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} + b_0 \cdot u$$

This form is also known as the *controller canonical form*. The disadvantage of this method is that we have no clue on the nature of the state variables.

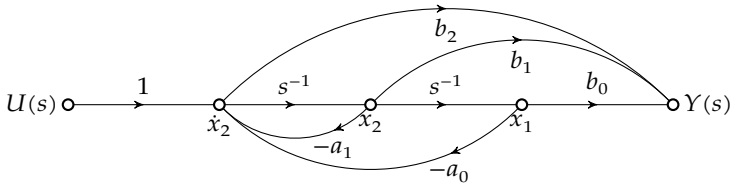
Let's illustrate this with the example of Figure 8.1 that was used earlier. In the Laplace domain (assuming zero initial conditions) this becomes:



Using the equation of a current divider, we can calculate:

$$\begin{aligned} I_L(s) &= \frac{\frac{1}{Cs}}{R + Ls + \frac{1}{Cs}} U(s) \\ Y(s) = RI_L(s) &= \frac{\frac{R}{Cs}}{R + Ls + \frac{1}{Cs}} U(s) \\ T(s) = \frac{Y(s)}{U(s)} &= \frac{\frac{R}{LC}}{s^2 + \frac{R}{L}s + \frac{1}{LC}} \end{aligned} \tag{8.3}$$

This corresponds to a second-order system with the following signal flow graph:



with $b_0 = \frac{R}{LC}$, $b_1 = b_2 = 0$, $a_1 = \frac{R}{L}$ and $a_2 = \frac{1}{LC}$.

Leading to the following state space description:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ -a_2 & -a_1\end{bmatrix}\cdot\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}\cdot u$$
$$y = \begin{bmatrix}b_2 - b_0a_2 & b_1 - b_0a_1\end{bmatrix}\cdot\begin{bmatrix}x_1\\ x_2\end{bmatrix} + b_0\cdot u$$

And after filling out the coefficient values:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\frac{1}{LC} & -\frac{R}{L}\end{bmatrix}\cdot\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}\cdot u$$
$$y = \begin{bmatrix}\frac{R}{LC} & 0\end{bmatrix}\cdot\begin{bmatrix}x_1\\ x_2\end{bmatrix} + 0\cdot u$$

The limited size of the example still allows to determine the nature of x_1 and x_2:

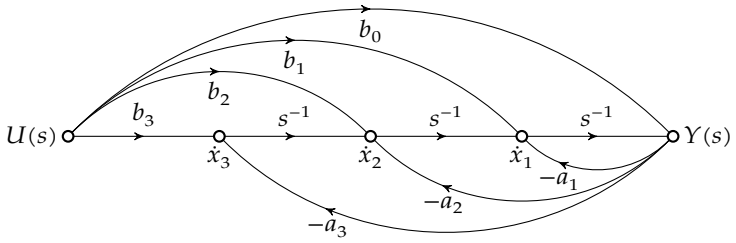
$$x_1 = LCi_L \qquad\qquad x_2 = Cv_L$$

However, in more complex examples, this might not be so easy. Also note that the selected state-space variables are not very obvious nor intuitive.

The transposed form II method Again, by dividing numerator and denominator of (8.2) by s^3, we can elaborate:

$$\frac{Y(s)}{X(s)} = \frac{b_0 + b_1s^{-1} + b_2s^{-2} + b_3s^{-3}}{1 - (-a_1s^{-1} - a_2s^{-2} - a_3s^{-3})}$$

The following signal flow graph corresponds to this expression (as one can check using Mason's rule):



Because a lot of the arrows are arriving in the intermediate nodes, we'd like to have them in the left-hand side of the state space equations and therefore, we taken them to be the derivatives of the state variables.

This allows writing the following equations:

$$\begin{aligned}
y &= x_1 + b_0u\\
\dot{x}_3 &= b_3u - a_3y = b_3u - a_3x_1 - a_3b_0u\\
\dot{x}_2 &= b_2u + x_3 - a_2y = b_2u + x_3 - a_2x_1 - a_2b_0u\\
\dot{x}_1 &= b_1u + x_2 - a_1y = b_1u + x_2 - a_1x_1 - a_1b_0u
\end{aligned}$$

In matrix form, we get our state-space description:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = \begin{bmatrix}-a_1 & 1 & 0\\ -a_2 & 0 & 1\\ -a_3 & 0 & 0\end{bmatrix}\cdot\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} + \begin{bmatrix}b_1 - a_1 b_0\\ b_2 - a_2 b_0\\ b_3 - a_3 b_0\end{bmatrix}\cdot u$$

$$y = \begin{bmatrix}1 & 0 & 0\end{bmatrix}\cdot\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} + b_0 \cdot u$$

This form is also known as the *observer canonical form*. Note that the matrices that appear in this form are the transpose of the matrices that appear in the *controller canonical form*.

Illustrating this with the earlier example of Figure 8.1, leads to the following state-space description:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}-a_1 & 1\\ -a_2 & 0\end{bmatrix}\cdot\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}b_1 - a_1 b_0\\ b_2 - a_2 b_0\end{bmatrix}\cdot u$$

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\cdot\begin{bmatrix}x_1\\ x_2\end{bmatrix} + b_0 \cdot u$$

and knowing that $b_2 = \frac{R}{LC}$, $b_1 = b_0 = 0$, $a_1 = \frac{R}{L}$ and $a_2 = \frac{1}{LC}$, this leads to:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}-\frac{R}{L} & 1\\ -\frac{1}{LC} & 0\end{bmatrix}\cdot\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ \frac{R}{LC}\end{bmatrix}\cdot u$$

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\cdot\begin{bmatrix}x_1\\ x_2\end{bmatrix} + 0 \cdot u$$

Again it is not trivial to determine the nature of the state variables, but in this case:

$$x_1 = Ri_L \qquad\qquad x_2 = \frac{R}{L}v_C$$

Note, that again, this is not an obvious choice.

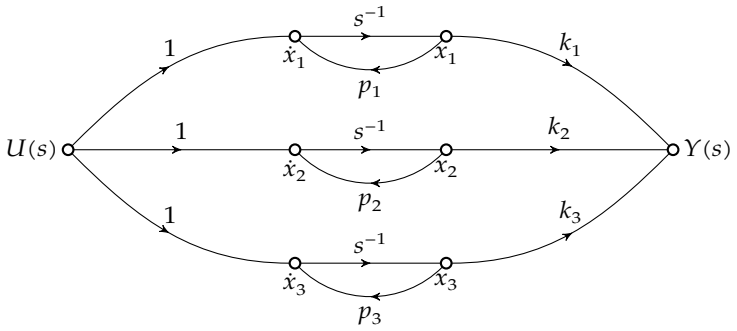
The method of the partial fraction expansion Partial fraction expansion of a transfer function leads to a sum of first order sections. For a third-order transfer function, this means:

$$T(s) = \frac{k_1}{s - p_1} + \frac{k_2}{s - p_2} + \frac{k_3}{s - p_3}$$

Multiplying numerator and denominator of these fractions by s^{-1} leads to:

$$T(s) = \frac{k_1 s^{-1}}{1 - p_1 s^{-1}} + \frac{k_2 s^{-1}}{1 - p_2 s^{-1}} + \frac{k_3 s^{-1}}{1 - p_3 s^{-1}}$$

which can be translated in the following obvious signal flow graph:



In case the poles are not real, one can keep the complex conjugate ones together, or we just can simple allow for complex state variables, without affecting the real nature of the input and the output.

Based on the graph above, we can write down the following equations:

$$\begin{aligned}\dot{x}_1 &= p_1 x_1 + u \\ \dot{x}_2 &= p_2 x_2 + u \\ \dot{x}_3 &= p_3 x_3 + u \\ y &= k_1 x_1 + k_2 x_2 + k_3 x_3\end{aligned}$$

In matrix form, we again see the state-space description emerging:

$$\begin{bmatrix}\dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3\end{bmatrix} = \begin{bmatrix}p_1 & 0 & 0 \\ 0 & p_2 & 0 \\ 0 & 0 & p_3\end{bmatrix} \cdot \begin{bmatrix}x_1 \\ x_2 \\ x_3\end{bmatrix} + \begin{bmatrix}1 \\ 1 \\ 1\end{bmatrix} \cdot u$$

$$y = \begin{bmatrix}k_1 & k_2 & k_3\end{bmatrix} \cdot \begin{bmatrix}x_1 \\ x_2 \\ x_3\end{bmatrix} + 0 \cdot u$$

This description is als known as the *diagonal canonical form*.

Note: the nature of the state variables is still unclear. Also note the specific diagonal form of the system matrix.

Exercises

Exercise 8.3.1-1: (*) Show how the signal flow graph of the direct form II can be derived from the direct form I by altering the cascade of the input and output part, allowing to merge the integrators.

Exercise 8.3.1-2: (*) Show how the signal flow graph of the transposed form II can be derived from the direct form I, by applying transposition and altering the cascade of the input and output part, allowing to merge the integrators.

From state-space description to transfer function Now, let's tackle the reverse problem: given a SISO system with its state-space description, compose the transfer function.

Let's tackle this from a generic point of view. The state-space equations for a SISO system have the following form:

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

Transforming this description to the Laplace domain yields:

$$sX(s) = AX(s) + bU(s) \tag{8.4}$$
$$Y(s) = CX(s) + dU(s) \tag{8.5}$$

in which we assumed zero initial conditions.

Solving $X(s)$ from (8.4) can be done in a few steps. With I the square unity matrix, we can write:

$$\begin{aligned} sX(s) &= AX(s) + bU(s) \\ (sI - A)X(s) &= bU(s) \\ X(s) &= (sI - A)^{-1}bU(s) \end{aligned}$$

where in the final step $(sI - A)^{-1}$ denotes the matrix inverse of $sI - A$.

Substituting this result into (8.5) yields:

$$\begin{aligned} Y(s) &= C(sI - A)^{-1}bU(s) + dU(s) \\ &= \left(C(sI - A)^{-1}b + d\right)U(s) \end{aligned}$$

allowing to calculate the transfer function $T(s) = Y(s)/U(s)$ in the Laplace domain:

$$T(s) = C(sI - A)^{-1}b + d \tag{8.6}$$

Let's illustrate this again with the example network of Figure 8.1. Earlier analysis resulted in the the following state space equations:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{C} \\ \frac{1}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{C} \\ 0 \end{bmatrix} u$$
$$y = \begin{bmatrix} 0 & R \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

This allows calculating

$$sI - A = \begin{bmatrix} s & \frac{1}{C} \\ -\frac{1}{L} & s + \frac{R}{L} \end{bmatrix}$$

and calculating its inverse:

$$(sI - A)^{-1} = \frac{\text{adj}\,(sI - A)}{\det\,(sI - A)}$$

in which the numerator contains the adjugate matrix (a.k.a. as the transposed cofactor matrix) and the denominator contains the determinant of the original matrix. You might want to brush up your matrix inversion skills.

For a two by two matrix this simply yields:

$$\text{adj}\,(sI - A) = \begin{bmatrix} s + \frac{R}{L} & -\frac{1}{C} \\ \frac{1}{L} & s \end{bmatrix}$$

$$\det(sI - A) = s\left(s + \frac{R}{L}\right) + \frac{1}{LC}$$

and allows for calculating the transfer function:

$$T(s) = \frac{\begin{bmatrix} 0 & R \end{bmatrix} \begin{bmatrix} s + \frac{R}{L} & -\frac{1}{C} \\ \frac{1}{L} & s \end{bmatrix} \begin{bmatrix} \frac{1}{C} \\ 0 \end{bmatrix}}{\det(sI - A)}$$

$$= \frac{\frac{R}{LC}}{s^2 + \frac{R}{L}s + \frac{1}{LC}}$$

which corresponds to the transfer function of (8.3).

8.3.2 Stability

Note that because in (8.6):

$$(sI - A)^{-1} = \frac{\text{adj}\,(sI - A)}{\det(sI - A)}$$

the determinant of the matrix $sI - A$ becomes the denominator of the transfer function. Therefore, the characteristic equation of the system (that allows calculating poles in order to determine its stability) is:

$$\det(sI - A) = 0$$

In linear algebra the determination of the values of s for which the above equation becomes zero is known as calculating the eigenvalues of the matrix A.

This is a major conclusion:

> The poles of a linear system are the eigenvalues of its system matrix.

Note that when discussing the conversion of transfer functions into state-space descriptions using the method of the partial fraction expansion, the resulting system matrix was:

$$A = \begin{bmatrix} p_1 & 0 & 0 \\ 0 & p_2 & 0 \\ 0 & 0 & p_3 \end{bmatrix}$$

According to our conclusion above, the poles can be found as the eigenvalues of the system matrix, i.e. solving:

$$\det(sI - A) = 0$$

$$\begin{vmatrix} s - p_1 & 0 & 0 \\ 0 & s - p_2 & 0 \\ 0 & 0 & s - p_3 \end{vmatrix} = 0$$

$$(s - p_1)(s - p_2)(s - p_3) = 0$$

This illustrates that the eigenvalues (i.e. the roots of the equation above) are indeed the poles of the system.

8.4 Calculating transient responses

As the state-space description is in essence a (set of) linear differential equations, we might also solve this using the Laplace transformation:

$$\begin{aligned}
\dot{x}(t) &= Ax(t) + Bu(t) \\
&\downarrow \quad \mathcal{L} \\
sX(s) - x(0) &= AX(s) + BU(s) \\
(sI - A)\, X(s) &= x(0) + BU(s) \\
&\downarrow \quad \text{if } \det(sI - A) \neq 0 \\
X(s) &= (sI - A)^{-1} (x(0) + BU(s)) \\
&\downarrow \quad \mathcal{L}^{-1} \\
x(t) &= \mathcal{L}^{-1} \left[(sI - A)^{-1} (x(0) + BU(s)) \right]
\end{aligned}$$

Together with the output equation, we can summarize in the Laplace domain:

$$\begin{aligned}
X(s) &= (sI - A)^{-1} (x(0) + BU(s)) \\
Y(s) &= CX(s) + DU(s)
\end{aligned}$$

Example Let's apply this to our simple example:

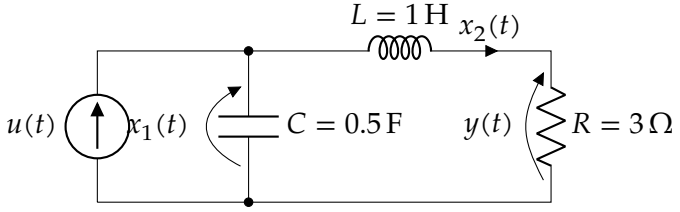


Figure 8.2: Example network with state variables indicated

Its state-space description is:

$$\begin{aligned}
\dot{x}(t) &= Ax(t) + bu(t) \\
y(t) &= Cx(t) + du(t)
\end{aligned}$$

with

$$A = \begin{bmatrix} 0 & -\frac{1}{C} \\ \frac{1}{L} & -\frac{R}{L} \end{bmatrix} = \begin{bmatrix} 0 & -2 \\ 1 & -3 \end{bmatrix} \qquad b = \begin{bmatrix} \frac{1}{C} \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & R \end{bmatrix} = \begin{bmatrix} 0 & 3 \end{bmatrix} \qquad d = 0$$

As $sI - A$ plays such a crucial role, we start by calculating it:

$$sI - A = \begin{bmatrix} s & 2 \\ -1 & s+3 \end{bmatrix}$$

Then, let's calculate its inverse:

$$(sI-A)^{-1} = \frac{\operatorname{adj}(sI-A)}{\det(sI-A)} = \frac{\begin{bmatrix} s+3 & -2 \\ 1 & s \end{bmatrix}}{s(s+3)+2} = \frac{\begin{bmatrix} s+3 & -2 \\ 1 & s \end{bmatrix}}{(s+1)(s+2)}$$

This results in:

$$\begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} = \frac{\begin{bmatrix} s+3 & -2 \\ 1 & s \end{bmatrix}}{(s+1)(s+2)} \left(\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} + \begin{bmatrix} 2 \\ 0 \end{bmatrix} U(s) \right)$$

$$Y(s) = \begin{bmatrix} 0 & 3 \end{bmatrix} \begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} = 3X_2(s)$$

As a first case, let's assume $U(s) = 0$, $x_1(0) = 1\,\mathrm{V}$, $x_2(0) = 2\,\mathrm{A}$

$$\begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} = \frac{\begin{bmatrix} s+3 & -2 \\ 1 & s \end{bmatrix}}{(s+1)(s+2)} \left(\begin{bmatrix} 1 \\ 2 \end{bmatrix} \right) = \frac{\begin{bmatrix} s-1 \\ 2s+1 \end{bmatrix}}{(s+1)(s+2)}$$

Therefore:

$$X_1(s) = \frac{s-1}{(s+1)(s+2)} \overset{PFE}{=} \frac{-2}{s+1} + \frac{3}{s+2} \xrightarrow{\mathcal{L}^{-1}} x_1(t) = u(t)\left(-2\,\mathrm{e}^{-t} + 3\,\mathrm{e}^{-2t}\right)$$

$$X_2(s) = \frac{2s+1}{(s+1)(s+2)} \overset{PFE}{=} \frac{-1}{s+1} + \frac{3}{s+2} \xrightarrow{\mathcal{L}^{-1}} x_2(t) = u(t)\left(-\,\mathrm{e}^{-t} + 3\,\mathrm{e}^{-2t}\right)$$

and

$$Y(s) = 3X_2(s) \xrightarrow{\mathcal{L}^{-1}} y(t) = 3x_2(t) = 3u(t)\left(-\,\mathrm{e}^{-t} + 3\,\mathrm{e}^{-2t}\right)$$

As a second case, let's assume $U(s) = 1/s$, $x_1(0) = 0\,\mathrm{V}$, $x_2(0) = 0\,\mathrm{A}$
Note how the final values $x_1(+\infty) = 3\,\mathrm{V}$ and $x_2(+\infty) = 1\,\mathrm{A}$ can be checked to be correct as, in steady state, the capacitor is an open and the inductor is a short.

8.5 Application: analyzing the population count

Using state variables, it is possible to model systems in general in an intuitive way. As an example, consider modeling the evolution of the population count in relationship to

- the birth rate,
- the mortality rate,

- an incurable disease,
- its contagiousness, and
- the people's immunity for the disease.

Let's model this system using the following state variables:

Variable	Meaning
x_1	the number of people who are prone to the disease
x_2	the number of people who suffer the disease
x_3	the number of people that are immune to the disease

Next, let's consider the following model parameters expressing annual rates:

Parameter	Meaning
B	the birth rate
M_1	the mortality rate for healthy people
M_2	the mortality rate for sick people
S	the spontaneous sickness rate for non-immune people
I	the infection rate for non-immune people
V	the vaccination rate (making people immune)

leading to the following *annual growth* model of Figure 8.3.

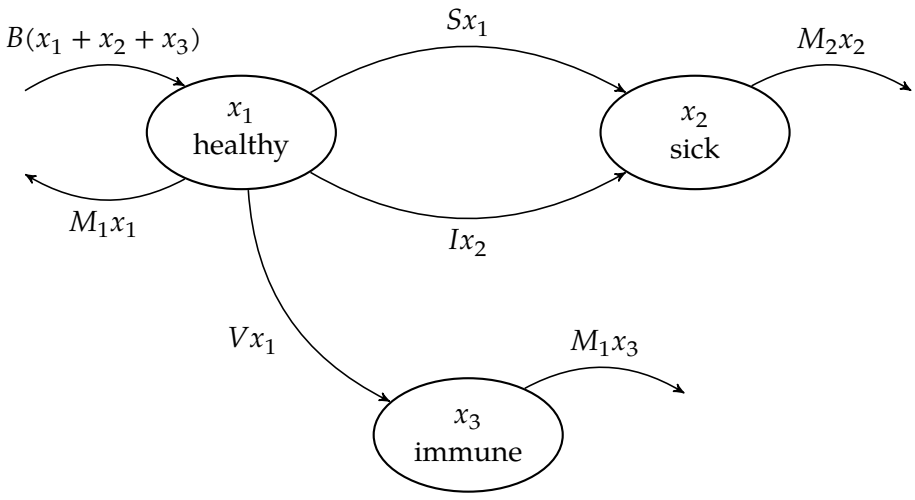


Figure 8.3: Annual population growth model

This model can be mathematically described using the following state-space description:

$$\begin{aligned}
\dot{x}_1 &= B(x_1 + x_2 + x_3) - M_1x_1 - Sx_1 - Ix_2 - Vx_1 \\
\dot{x}_2 &= Sx_1 + Ix_2 - M_2x_2 \\
\dot{x}_3 &= Vx_1 - M_1x_3
\end{aligned}$$

or using our matrix formalism:

$$\dot{x} = Ax \qquad \text{with} \quad A = \begin{bmatrix} B - M_1 - S - V & B - I & B \\ S & I - M_2 & 0 \\ V & 0 & -M_1 \end{bmatrix}$$

The system matrix A and its eigenvalues determine the stability of the system. A model as the one above may help a government to check for the influence of an additional vaccination campaign, anti conception program or other health-related measures on the global population count.

We can use MATLAB/OCTAVE to analyze this model. The following script will allow you to tinker with the parameters and check for the effects:

```
%% file  : popcount.m
%% author: M. Vanpaemel

%% parameters
B  = 0.02;
M1 = 0.01;
M2 = 0.1;
S  = 0.02;
I  = 0.01;
V  = 0.02;

%% simulation length (in years)
T  = 500;

%% state space description
A = [
  B-M1-S-V, B-I,  B;
  S,        I-M2, 0;
  V,        0,   -M1
];
b = [0;0;0] ; d = 0 ; C = [1 1 1];
sys = ss( A, b, C, d );

%% check the eigen values
eig(A)

%% simulation data: initial conditions, time vector and input vector
x0=[10 0 0];   % in millions
t=[0:T/100:T]; % in years
u=0*t;         % no external input

%% simulate
[y,t,x]=lsim(sys,u,t,x0);

%% plot results
subplot(221),plot(t,x(:,1));
xlabel('year'),ylabel('x_1 : healthy (10^6)'); grid on;
subplot(222),plot(t,x(:,2));
xlabel('year'),ylabel('x_2 : sick (10^6)'); grid on;
subplot(223),plot(t,x(:,3));
xlabel('year'),ylabel('x_3 : immune (10^6)'); grid on;
subplot(224),plot(t,y);
xlabel('year'),ylabel('y : total population count (10^6)'); grid on;
```

Try different scenario's and assess the results. E.g.

- $V = 0.02$
- $V = 0$

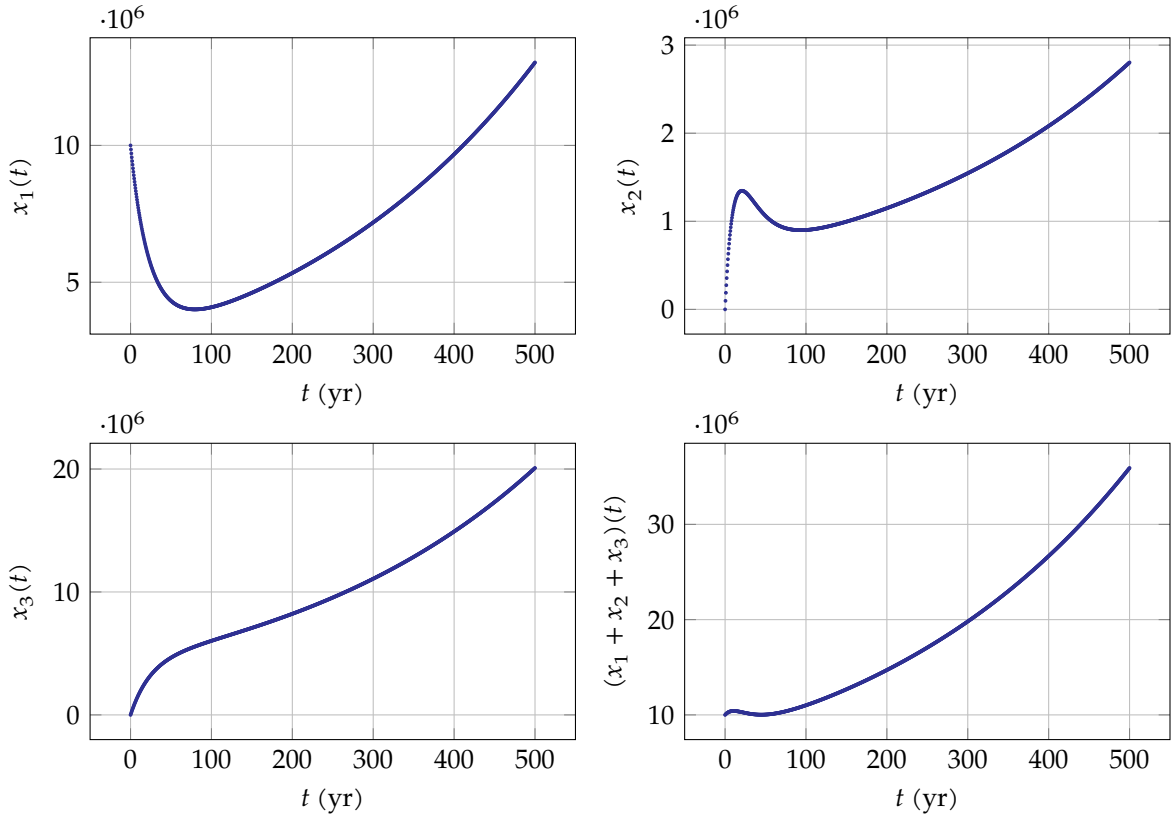


Figure 8.4: Simulation of the linear population count model, for $B = 0.02, M_1 = 0.01, M_2 = 0.1, S = 0.02, I = 0.01, V = 0.02$

- $V = 0.1$

Take a look at the eigenvalues and try to relate them to your simulation results. For your reference, you can find the plots for $V = 0.02$ in Figure 8.4.

8.6 Discretizing the state-space equation

The true power of using a state-space description emerges when we discretize a continuous-time model to a discrete-time model. This is a step that is mandatory if we want to simulate complex systems using a computer.

There are multiple options to perform this discretization step. We will use the most simple approaches, i.e. a forward Euler method and a backward Euler method. It should be noted that the only benefit of this approach is its simplicity. In practice more complex (and better) discretization strategies are used.

8.6.1 Forward Euler method

The forward Euler method replaces the derivative $\dot{x}$ by a forward difference quotient

$$\dot{x} = \frac{\mathrm{d}x(t)}{\mathrm{d}t} \approx \frac{x(t + T_s) - x(t)}{T_s}$$

assuming a uniformly sampled signal at a rate $f_s = 1/T_s$.

Applying this discretization to the state equation, allows the following derivation:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ \frac{x(t+T_s) - x(t)}{T_s} &= Ax(t) + Bu(t) \\ x(t+T_s) &= (T_sA + I)x(t) + T_sBu(t) \end{aligned}$$

Replacing the continuous time variable t by a discrete counter n, yields:

$$x[n+1] = (T_sA + I)x[n] + T_sBu[n] \tag{8.7}$$

This incremental equation allows for calculating the next state (for time index $n+1$) using the current state and the current input (both for time index n). A proper choice for T_s is paramount. The smaller T_s, the better the approximation, however, the higher the computational load. To obtain a reasonable result, T_s should be smaller than the smallest time constant of the system.

Example 1 Let's apply this technique to the state-space description of our example network of Figure 8.2 on page 206.

It had two poles $p_1 = 1$ and $p_2 = 2$. Because of this, the smallest time constant is $\tau_2 = 1/p_2 = 0.5\,\text{s}$. Therefore a time step of $T_s = 0.1\,\text{s}$ should do the job.

The composing matrices to write down (8.7) are:

$$A = \begin{bmatrix} 0 & -2 \\ 1 & -3 \end{bmatrix} \Rightarrow T_sA + I = \begin{bmatrix} 1 & -0.2 \\ 0.1 & 0.7 \end{bmatrix} \qquad b = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \Rightarrow T_sb = \begin{bmatrix} 0.2 \\ 0 \end{bmatrix}$$

and therefore:

$$x[n+1] = \begin{bmatrix} 1 & -0.2 \\ 0.1 & 0.7 \end{bmatrix} x[n] + \begin{bmatrix} 0.2 \\ 0 \end{bmatrix} u[n] \tag{8.8}$$

Let's again consider the earlier two cases:

As a first case, let's assume $u(t) = 0$, $x_1(0) = 1\,\text{V}$, $x_2(0) = 2\,\text{A}$
We therefore set $x[0] = [12]^\top$ and evaluate (8.8) multiple times. The result can be found in the graphs of Figure 8.6. As one can see there is a significant difference between the simulated points and the analytical solution. For $T_s = 0.01\,\text{s}$ the data points would almost perfectly coincide with the analytical solution.

As a second case, let's assume $u(t) = 1$, $x_1(0) = 0\,\text{V}$, $x_2(0) = 0\,\text{A}$
We therefore set $x[0] = [0\ 0]^\top$ and evaluate (8.8) multiple times keeping a constant input $u[n] = 1$. The result can be found in the graphs of Figure 8.6. As one can see there is a significant difference between the simulated points and the analytical solution. For $T_s = 0.01\,\text{s}$ the data points would almost perfectly coincide with the analytical solution.

This illustrates that the discretization approach can be used to simulate linear time-invariant systems. However, the method can also be used in the case of nonlinear and also time variant systems! Let's illustrate this with another example.

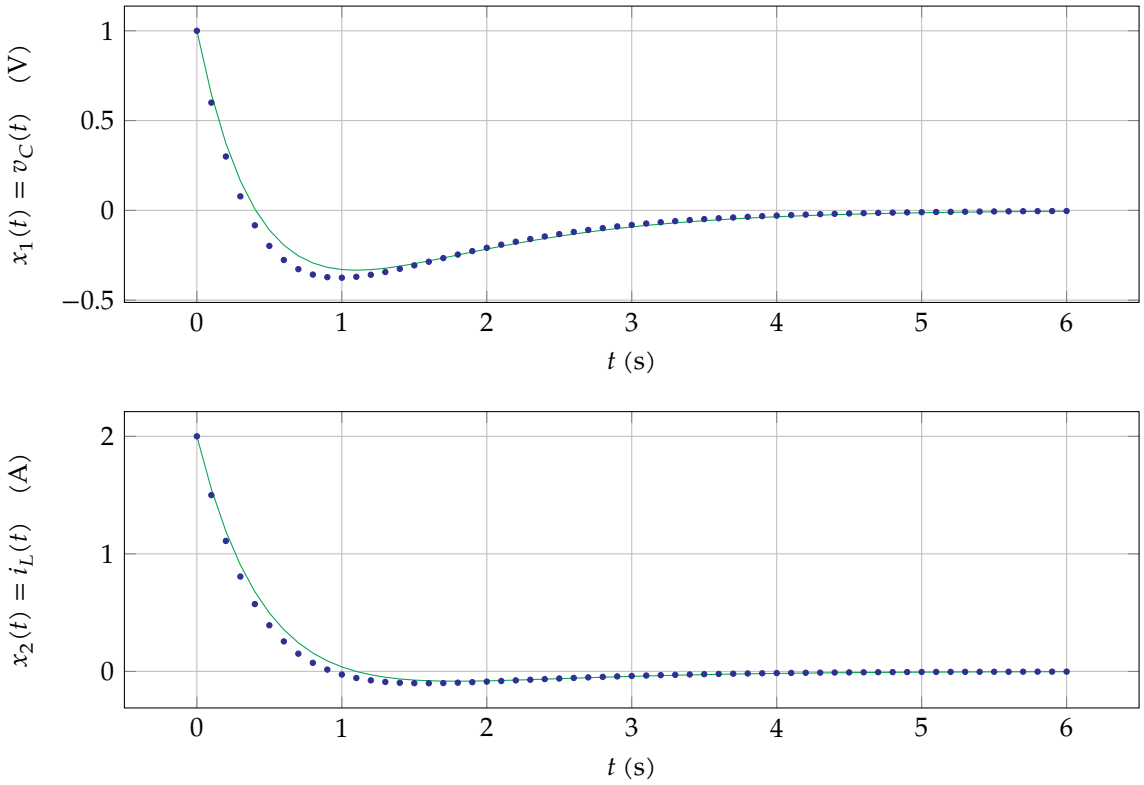


Figure 8.5: Simulation of the network of Figure 8.2 on page 206 with zero input and $x[0] = [1\ 2]^\mathsf{T}$ using forward Euler discretization with $T_s = 0.1\,\mathrm{s}$ (blue dots), compared to the analytical solution (green solid line)

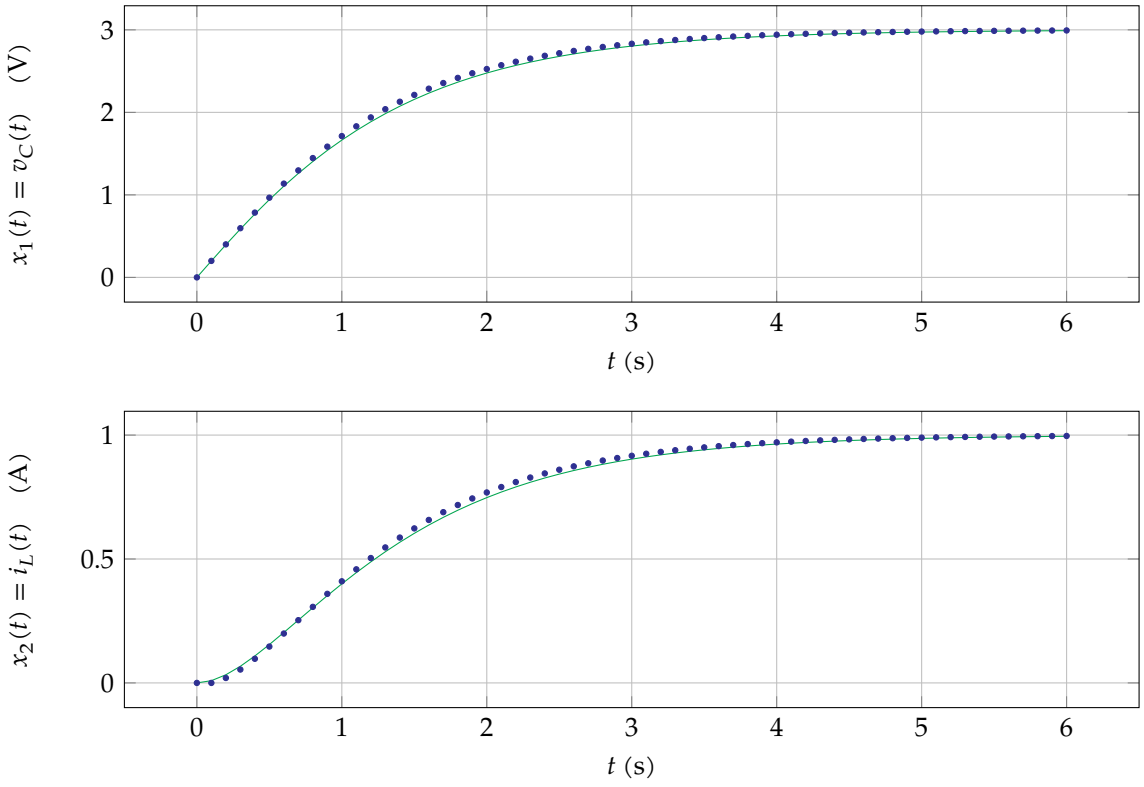


Figure 8.6: Simulation of the network of Figure 8.2 on page 206 with zero initial conditions and $u[n] = 1$ using forward Euler discretization with $T_s = 0.1\,\mathrm{s}$ (blue dots), compared to the analytical solution (green solid line)

Example 2 Let's reconsider our population count application. Actually, the model we proposed for the population growth contains a flaw. Try to simulate with $I = 0.1$ instead of $I = 0.01$ and see what happens: the state variables become negative (as can be seen in Figure 8.7), which corresponds to a non-physical situation. It's not the simulation that is to blame, but the model. As the decline of x_1 due to infection is modeled as $-Ix_2$, in case of large x_2, this negative term might be bigger than the remaining x_1, i.e. more people are modeled to be infected than there are healthy people left. Obviously, this is not possible. A fix for this problem could be to model the number of infections as Ix_1x_2. This results in an adapted model:

$$\begin{aligned}\dot{x}_1 &= B(x_1 + x_2 + x_3) - M_1x_1 - Sx_1 - Ix_1x_2 - Vx_1\\ \dot{x}_2 &= Sx_1 + Ix_1x_2 - M_2x_2\\ \dot{x}_3 &= Vx_1 - M_1x_3\end{aligned}$$

It is no longer possible to derive a linear matrix equation with a system matrix.

However, we still can use our forward Euler discretization, resulting in the following finite difference equations:

$$\begin{aligned}x_1[n+1] &= ((B - M_1 - S - V)T_s + 1)\, x_1[n] + BT_sx_2[n] + BT_sx_3[n] - IT_sx_1[n]x_2[n]\\ x_2[n+1] &= ST_sx_1[n] + (-M_2T_S + 1)x_2[n] + IT_sx_1[n]x_2[n]\\ x_3[n+1] &= VT_sx_1[n] + (-M_1T_s + 1)x_3[n]\end{aligned}$$

These equations can be used in conjunction with a T_s that is small enough (e.g. $T_s = 1/2$yr) to calculate a sequence of states. The fact that there is a product of two state variables in there does not make the computational load significantly higher than in the linear case. In addition, making some parameters time dependent (e.g. an increasing value for V after a vaccination campaign) is also easily incorporated. Given a specific $V[n]$, we can easily use the following equation set:

$$\begin{aligned}x_1[n+1] &= ((B - M_1 - S - V[n])T_s + 1)\, x_1[n] + BT_sx_2[n] + BT_sx_3[n] - IT_sx_1[n]x_2[n]\\ x_2[n+1] &= ST_sx_1[n] + (-M_2T_S + 1)x_2[n] + IT_sx_1[n]x_2[n]\\ x_3[n+1] &= V[n]T_sx_1[n] + (-M_1T_s + 1)x_3[n]\end{aligned}$$

One could go further and make any of the parameters dependent on one of the state variables (e.g. the more sick people, the more people will be inclined to get a vaccine).

Implementing this in Octave/Matlab is straightforward:

```
%% file  : popcount_nonlin.m
%% author: W.Daems

%% parameters
B  = 0.02;
M1 = 0.01;
M2 = 0.1;
S  = 0.02;
I  = 0.1;
V  = 0.1;

%% simulation length (in years)
T  = 500;
Ts = 0.5

%% discretized description (forward Euler)
function Xnext = model( X, Ts, B, M1, M2, S, I, V )
  Xnext(1) = ((B - M1 - S - V)*Ts+1)*X(1)+B*Ts*X(2)+B*Ts*X(3)-I*Ts*X(1)*X(2);
```

```
  Xnext(2) = S*Ts*X(1) + (1-M2*Ts)*X(2)+I*Ts*X(1)*X(2);
  Xnext(3) = V*Ts*X(1) + (1-M1*Ts)*X(3);
end

X(:,1) = [ 10, 0, 0 ];
t = linspace(0,T,T/Ts+1);
for i = 1:1000
  X(:,i+1) = model( X(:,i), Ts, B, M1, M2, S, I, V );
end

%% plot results
figure;
subplot(221),plot(t,X(1,:));
xlabel('year'),ylabel('x_1 : healthy (10^6)'); grid on;
subplot(222),plot(t,X(2,:));
xlabel('year'),ylabel('x_2 : sick (10^6)'); grid on;
subplot(223),plot(t,X(3,:));
xlabel('year'),ylabel('x_3 : immune (10^6)')' grid on;
subplot(224),plot(t,X(1,:)+X(2,:)+X(3,:));
xlabel('year'),ylabel('y : total population count (10^6)'); grid on;
```

The results of a simulation with $T_s = 1/2\text{yr}$ can be found in Figure 8.8.

8.6.2 Backward Euler method

The backward Euler method is only a little bit more complicated that the forward Euler method. However, it has better convergence properties.

The backward Euler method replaces the derivative $\dot{x}$ by a backward difference quotient:

$$\dot{x} = \frac{\mathrm{d}x(t)}{\mathrm{d}t} \approx \frac{x(t) - x(t - T_s)}{T_s}$$

assuming a uniformly sampled signal at a rate $f_s = 1/T_s$.

Applying this discretization to the state equation, allows the following derivation:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ \frac{x(t) - x(t - T_s)}{T_s} &= Ax(t) + Bu(t) \\ (I - T_s A)x(t) &= x(t - T_s) + T_s Bu(t) \\ x(t) &= (I - T_s A)^{-1} x(t - T_s) + T_s Bu(t) \end{aligned}$$

Replacing the continuous time variable t by a discrete counter n, yields:

$$x[n] = (I - T_s A)^{-1} \left(x[n-1] + T_s Bu[n]\right)$$

This incremental equation allows for calculating the current state (for time index n) based on the previous state and the current input. A proper choice for T_s is also in this case paramount and the same observations hold as for the forward Euler case.

8.6.3 Summary

Many computer programs apply this technique to simulate the time-domain behavior of electronic circuits (SPICE-like simulators) or more generic models (e.g. Simulink). These

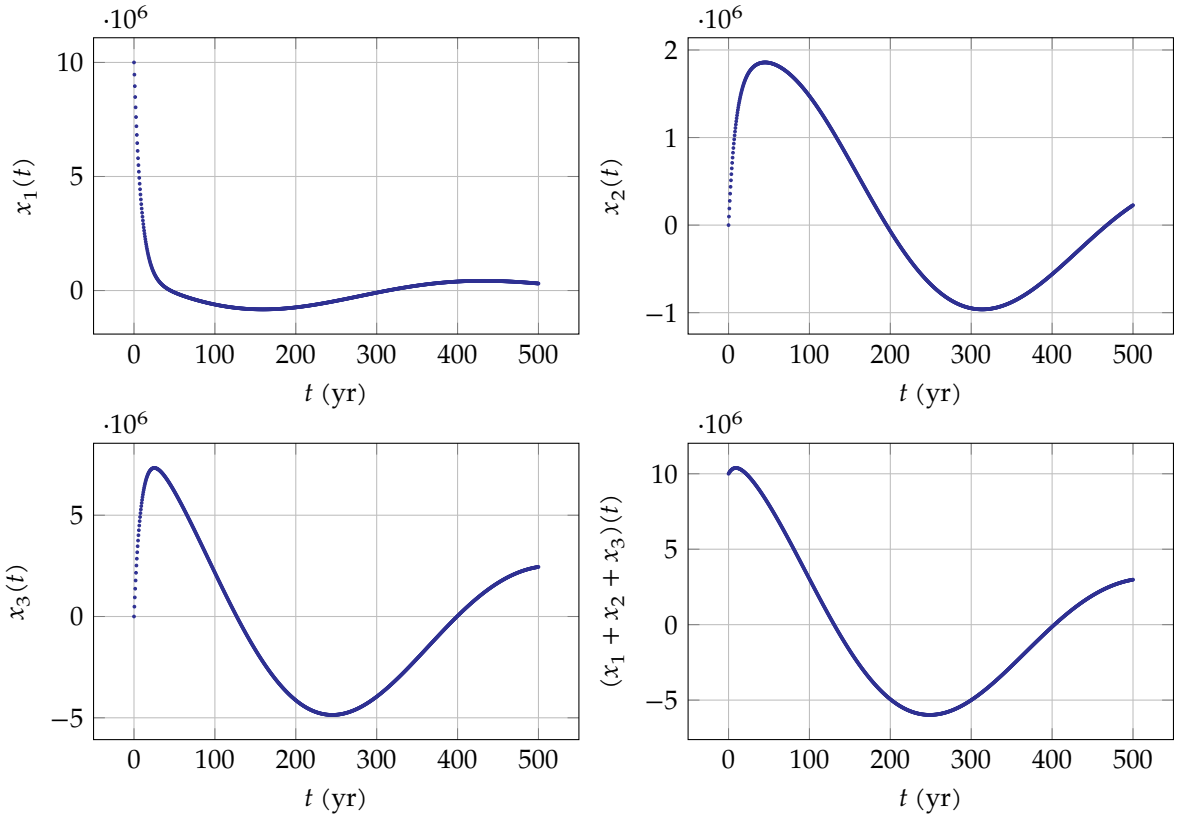


Figure 8.7: Simulation of the linear population count model, for $B = 0.02, M_1 = 0.01, M_2 = 0.1, S = 0.02, I = V = 0.1$ — the negative counts clearly indicate a model error.

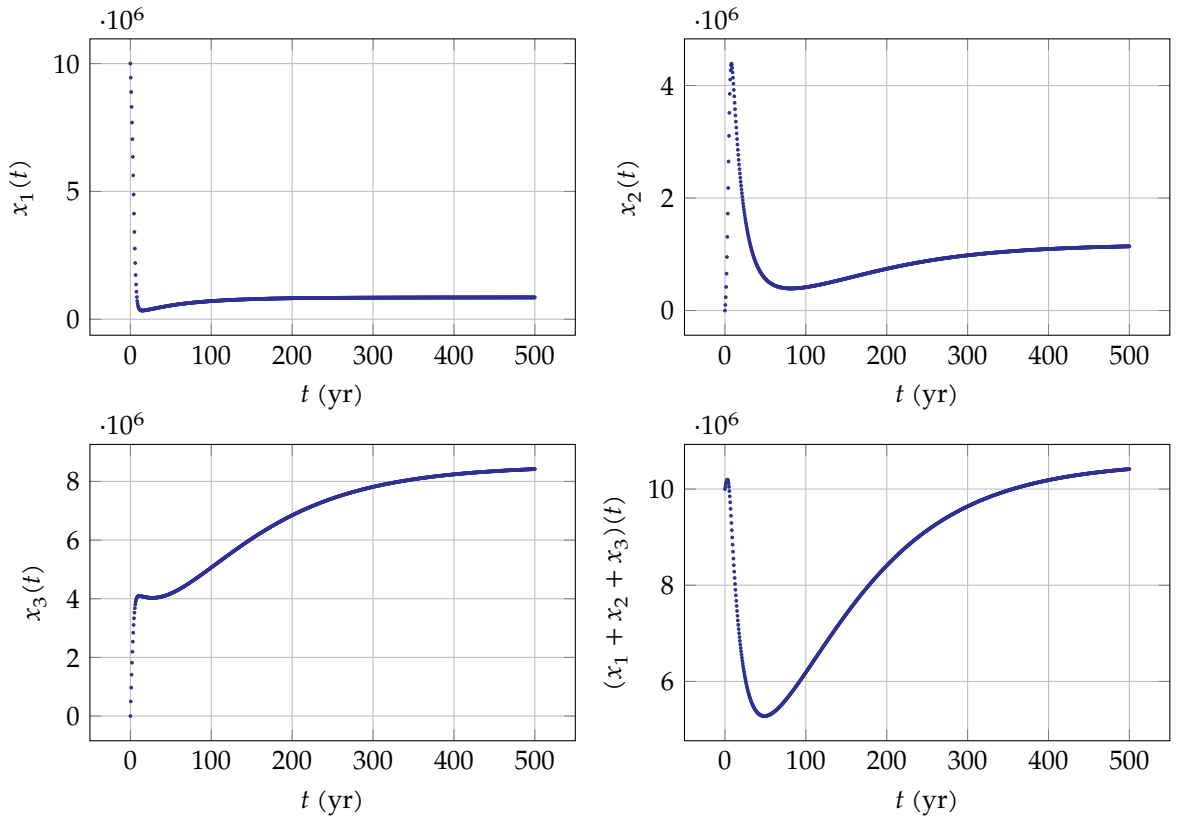


Figure 8.8: Simulation of the nonlinear (corrected) population count model, for $B = 0.02, M_1 = 0.01, M_2 = 0.1, S = 0.02, I = V = 0.1$ — all counts are now positive.

programs often have an extensive set of integration rules (e.g., forward and backward Euler, bilinear mapping, Gear, Runge-Kutta, ...). Together with an appropriate time step, these allow for detailed simulation of generic systems. Often these programs autonomously select an appropriate integration rule and time step, allowing the user to overrule that choice.

8.7 State-space control

Now, we're ready for the main topic of this chapter: how can we control these multi-input, multi-output systems described by a state-space model?

Let's consider the LTI system of Figure 8.9 with 3 inputs and 2 state variables. We neglect the true outputs of the system for now and only focus on the state variables as output.

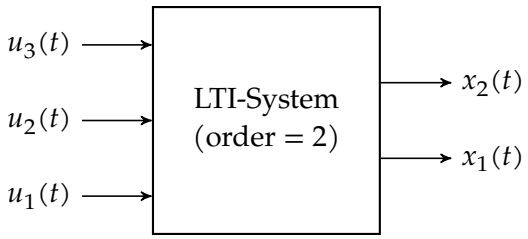


Figure 8.9: Example 2nd-order LTI system with 3 inputs $u_i(t)$ and 2 state variables $x_i(t)$

Its system equation $\dot{x} = Ax + Bu$ can be written in full as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

The system matrix A will determine the dynamic behavior of the system, as the poles of the system are determined by its eigenvalues, i.e. by $\det(sI - A) = 0$.

8.7.1 Internal feedback

Now, let's assume that we will be using the first two inputs to provide the system with feedback on its state variables. We call this *internal feedback* because it does not interfere with the remaining external input.

In view of this, it makes sense to separate the internal inputs u_1 and u_2 from the external input u_3, as in:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}}_{\equiv B_i} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \underbrace{\begin{bmatrix} b_{13} \\ b_{23} \end{bmatrix}}_{\equiv B_e} u_3$$

Consider e.g. the straight feedback scheme below:

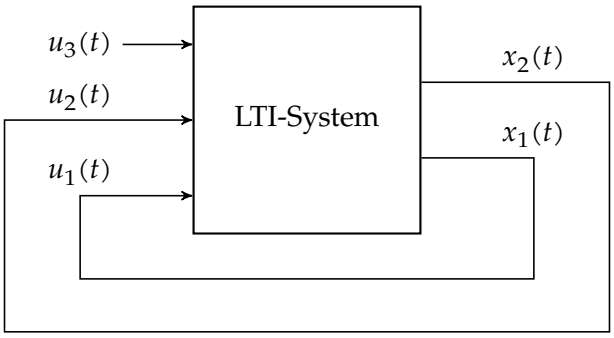


Figure 8.10: Example 2nd-order LTI system with straight feedback

In this case, the system equation becomes:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} \\ a_{21} + b_{21} & a_{22} + b_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_{13} \\ b_{23} \end{bmatrix} u_3$$

However, in general, the feedback does not need to be straight, but can make use of a feedback network, as in

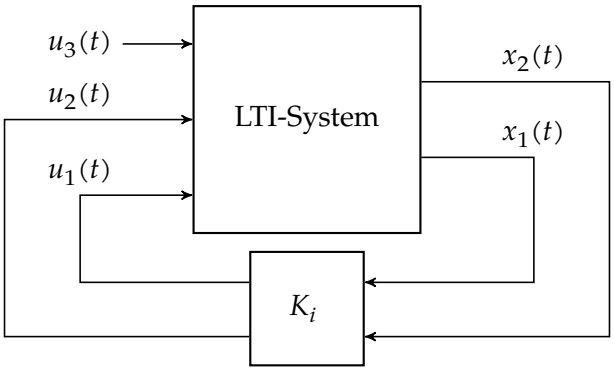


Figure 8.11: Example 2nd-order LTI system with feedback gain K_i

with

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \underbrace{\begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}}_{\equiv K_i} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

resulting in the following system equation:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \underbrace{\left(\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \right)}_{\equiv A_i} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_{13} \\ b_{23} \end{bmatrix} u_3$$

We can use K_i to tailor the system matrix A_i (and therefore the poles of the system) as we want.

In general:

$$\dot{x} = (A + B_i K_i)\, x + B_e u$$

8.7.2 External feedback

An alternative way to create a feedback system, is to explicit add the generated feedback to the external inputs. We therefore call this principle *external feedback*.

The setup is drawn in Figure 8.12.

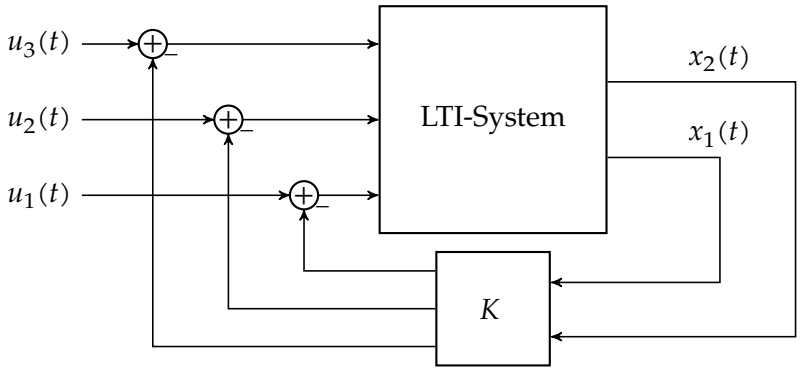


Figure 8.12: Example 2nd-order LTI system with external feedback

This results in the following system equation:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}b_{11} & b_{12} & b_{13}\\ b_{21} & b_{22} & b_{23}\end{bmatrix}\left(\begin{bmatrix}u_1\\ u_2\\ u_3\end{bmatrix} - \begin{bmatrix}k_{11} & k_{12}\\ k_{21} & k_{22}\\ k_{31} & k_{32}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}\right)$$

$$= \underbrace{\left(\begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix} - \begin{bmatrix}b_{11} & b_{12} & b_{13}\\ b_{21} & b_{22} & b_{23}\end{bmatrix}\begin{bmatrix}k_{11} & k_{12}\\ k_{21} & k_{22}\\ k_{31} & k_{32}\end{bmatrix}\right)}_{\equiv A_e}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}b_{11} & b_{12} & b_{13}\\ b_{21} & b_{22} & b_{23}\end{bmatrix}\begin{bmatrix}u_1\\ u_2\\ u_3\end{bmatrix}$$

We can use K_e to tailor the system matrix A_e (and therefore the poles of the system) as we want.

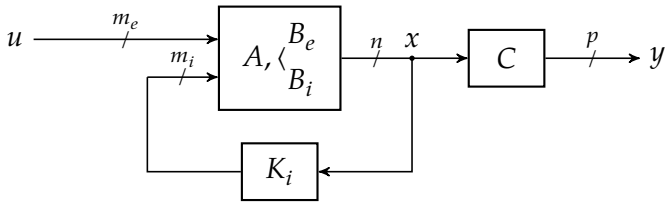
In general:

$$\dot{x} = (A - BK_e)\,x + Bu$$

8.7.3 Simplified diagrams

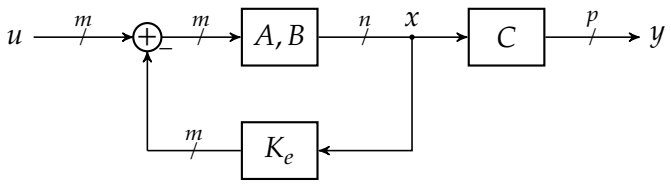
Often a simplified diagram is made, that from a distance closely resembles the simple SISO control systems that we treated earlier. However, the signals are not scalars, but vectors.

Assuming the feedthrough matrix $D = 0$, we obtain the block diagram of Figure 8.13 for the system with internal feedback and the diagram of Figure 8.14 for the system with external feedback. We assumed n state variables, m_e external inputs, m_i internal inputs, giving the LTI system to be controlled a total of m inputs and p outputs.



$$\dot{x} = (A + B_i K_i)\, x + B_e u$$
$$y = Cx$$

Figure 8.13: Simplified schematic and corresponding state-space equations for an LTI system with internal feedback, without feedthrough. The signals have the following dimensions: u is $m_e \times 1$, x is $n \times 1$ and y is $p \times 1$. The matrices have the following dimensions: A is $n \times n$, B_e is $n \times m_e$, B_i is $n \times m_i$, K_i is $m_i \times n$ and C is $p \times n$.



$$\dot{x} = (A - BK_e)\, x + Bu$$
$$y = Cx$$

Figure 8.14: Simplified schematic and corresponding state-space equations for an LTI system with external feedback, without feedthrough. The signals have the following dimensions: u is $m \times 1$, x is $n \times 1$ and y is $p \times 1$. The matrices have the following dimensions: A is $n \times n$, B is $n \times m$, K_e is $m \times n$ and C is $p \times n$.

8.7.4 Examples

Example 1 Consider the following uncontrolled system:

$$\dot{x} = Ax + bu$$
$$y = c^\top x$$

with

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -3 & -5 \end{bmatrix} \qquad b = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad c^\top = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

In this case the number of inputs $m = 1$, the number of state variables $n = 3$ and the number of outputs $p = 1$. Imposing external feedback, corresponds to choosing an $m \times n$ matrix, i.e. a 1×3 row vector

$$\boldsymbol{k}_e^\top = \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix}$$

that results in the following controlled system:

$$\dot{x} = \left(A - b\boldsymbol{k}_e^\top\right) x + b\, u$$
$$y = c^\top x$$

The system matrix of the controlled system, therefore is:

$$\begin{aligned} A_e = A - bk_e^T &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -3 & -5 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -3 & -5 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ k_1 & k_2 & k_3 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 - k_1 & -3 - k_2 & -5 - k_3 \end{bmatrix} \end{aligned}$$

This leads to the following characteristic equation:

$$\det(sI - A_e) = 0$$
$$\begin{vmatrix} s & -1 & 0 \\ 0 & s & -1 \\ 2 + k_1 & 3 + k_2 & s + 5 + k_3 \end{vmatrix} = 0$$
$$s^3 + (5 + k_3)s^2 + (3 + k_2)s + 2 + k_1 = 0 \tag{8.9}$$

Let's assume we want to give this three pole system the following poles:

$$p_{1,2} = -4.8 \pm j3.6 \qquad p_3 = -4.8$$

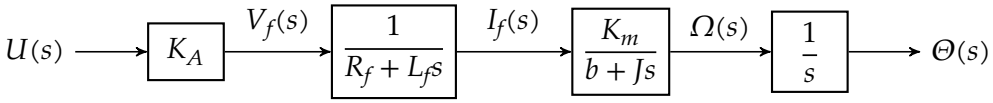
This corresponds to a characteristic polynomial:

$$\begin{aligned} T(s) &= (s-p_1)(s-p_2)(s-p_3) \\ &= (s+4.8+j3.6)(s+4.8-j3.6)(s+4.8) \\ &= (s^2+9.6s+36)(s+4.8) \\ &= s^3+14.4s^2+82.08s+172.8 \end{aligned} \tag{8.10}$$

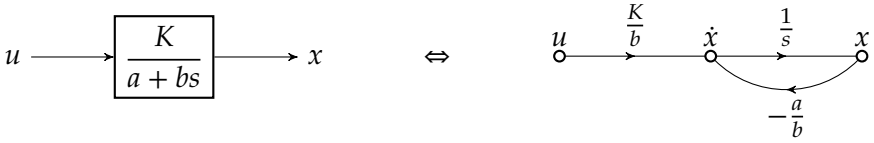
Equation of the coefficients of (8.9) and (8.10), results in:

$$\begin{cases} 5+k_3 = 14.4 \\ 3+k_2 = 82.08 \\ 2+k_1 = 172.8 \end{cases} \Leftrightarrow \begin{cases} k_1 = 170.8 \\ k_2 = 79.08 \\ k_3 = 9.4 \end{cases}$$

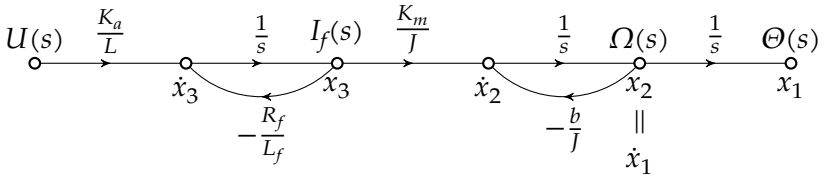
Example 2 As a second example, let's reconsider our DC motor of section 1.1.3 on page 4 in field control mode. Its block diagram in the Laplace domain was:



Using the following equivalence, we can transform the block diagram above to an *all integrator signal flow graph*. The equivalence is easily checked using Mason's rule.



The resulting *all integrator signal flow graph* is:



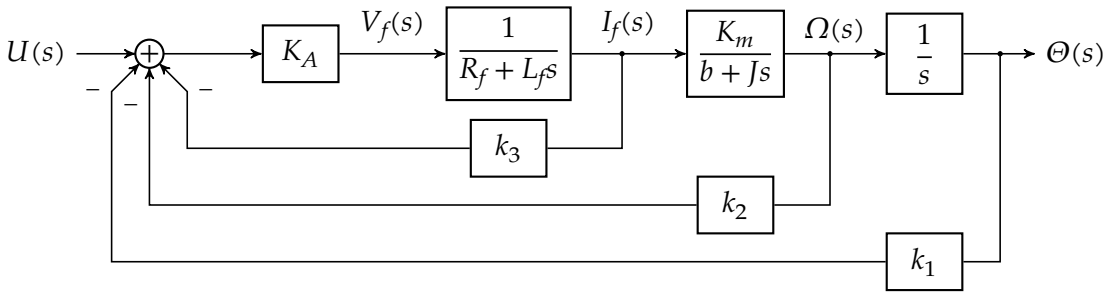
Choosing as state variables $x_1 = \theta$, $x_2 = \omega$ and $x_3 = i_f$ (as indicated on the graph above), allows writing down the following state equation:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{b}{J}x_2 + \frac{K_m}{J}x_3 \\ \dot{x}_3 &= \frac{K_a}{L}u - \frac{R_f}{L_f}x_3 \end{aligned}$$

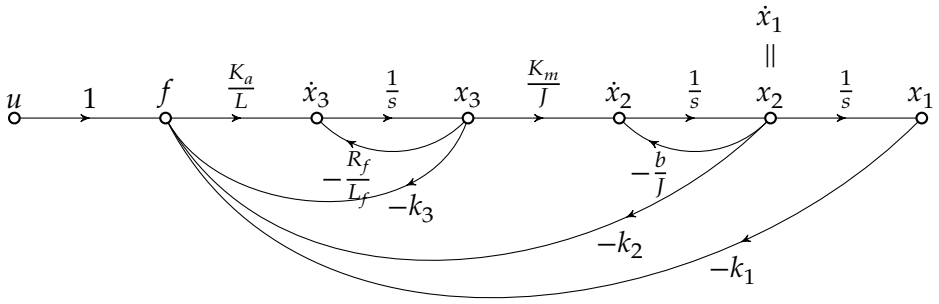
In our matrix formalism, this becomes:

$$\dot{x} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{b}{J} & \frac{K_m}{J} \\ 0 & 0 & -\frac{R_f}{L_f} \end{bmatrix}}_{\equiv A} x + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \frac{K_a}{L} \end{bmatrix}}_{\equiv b} u$$

Now, let's try to control our motor using three feedback paths (feeding back every state variable to be added to the input, i.e. an external feedback configuration). Implementing this idea in the block diagram results in:



The signal flow graph equivalent becomes:



By simple inspection, we can write the following system equation:

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -\frac{b}{J}x_2 + \frac{K_m}{J}x_3 \\
\dot{x}_3 &= \frac{K_a}{L}f - \frac{R_f}{L_f}x_3 = \frac{K_a}{L}\left(u - k_3x_3 - k_2x_2 - k_1x_1\right) - \frac{R_f}{L_f}x_3
\end{aligned}$$

In our matrix formalism, this becomes:

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{b}{J} & \frac{K_m}{J} \\ -k_1\frac{K_a}{L} & -k_2\frac{K_a}{L} & -\frac{R_f}{L_f} - k_3\frac{K_a}{L} \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ \frac{K_a}{L} \end{bmatrix} u$$

Note that the system matrix of the feedback system corresponds to $A - bk_e^T$ with $k_e^T = \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix}$.

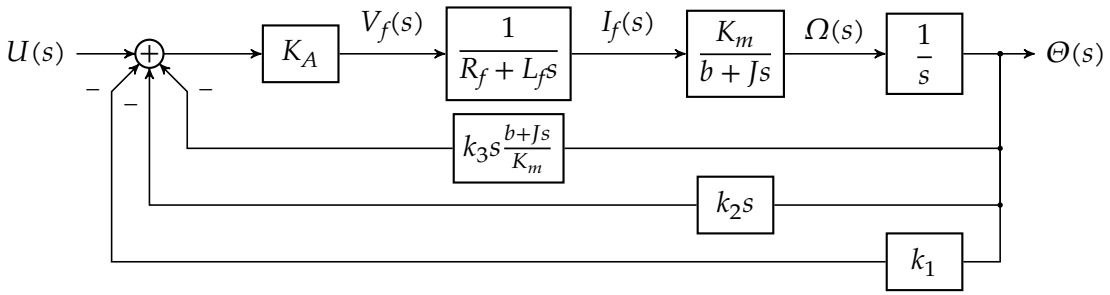
The characteristic equation can be derived as:

$$\det\left(sI - (A - bk_e^T)\right) = 0$$

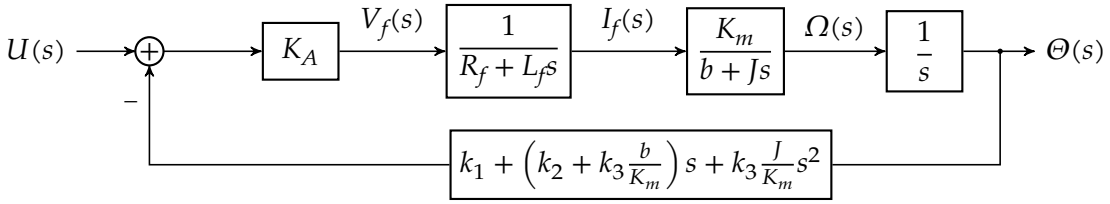
The obvious next step is to choose k_e^T such that we set the poles of the system at desired locations.

Before doing so, we'd first like to illustrate the beneficial effect of state feedback (i.e. using the knowledge of the individual states in controlling the system). We therefore start again from

the block diagram with state feedback. We can move the tapping points of the feedback paths to the output, leading to:



This allows merging the three feedback paths, resulting in:



Now, we can clearly see that the state feedback introduces two extra zeros in the feedback loop. It is as if we added two PD-controllers in the loop, significantly increasing ω_c of the feedback system! In addition, the zeros are in the feedback path, such that they don't cause an extreme increase in overshoot.

On an intuitive basis, we can also understand the benefit of state feedback: the intermediate state feedback (through k_3 and k_2) gives the system an 'early warning' system. The system doesn't have to wait until the signal propagates to the output before it can take corrective action. It measures early on and therefore allows for an early and quick response, (1) improving the speed and (2) reducing the overshoot of the system.

Let's return to choosing an appropriate k_e^T. The characteristic equation can be derived by elaborating:

$$\det\left(sI - (A - bk_e^T)\right) = 0$$

$$\det\left(sI - \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{b}{J} & \frac{K_m}{J} \\ -k_1\frac{K_a}{L} & -k_2\frac{K_a}{L} & -\frac{R_f}{L_f} - k_3\frac{K_a}{L} \end{bmatrix}\right) = 0$$

Let's assume:

$$R_f = 5\,\Omega \qquad L_f = 1\,\mathrm{H} \qquad b = 1\,\mathrm{Nm/(rad/s)} \qquad J = 1\,\mathrm{Nm/(rad/s^2)} \qquad K_m = 1\,\mathrm{N\,m/A}$$

$$\det\left(sI - \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \\ -k_1K_a & -k_2K_a & -5-k_3K_a \end{bmatrix}\right) = 0$$

$$\begin{vmatrix} s & -1 & 0 \\ 0 & s+1 & -1 \\ k_1K_A & k_2K_a & s+5+k_3K_a \end{vmatrix} = 0$$

$$s((s+1)(s+5+k_3K_a)+k_2K_a)+k_1K_a = 0$$

$$s^3 + s^2(6+k_3K_a) + s(5+(k_3+k_2)K_a) + k_1K_a = 0$$

Assuming we want the following pole locations:

$$p_{1,2} = -5 \pm j$$
$$p_3 = -10$$

the corresponding characteristic equation must be:

$$s^3 + 20s^2 + 126s + 260 = 0$$

Equating the matching coefficients, yields:

$$\begin{cases} 6 + k_3K_a = 20 \\ 5 + (k_3+k_2)K_a = 126 \\ k_1K_a = 260 \end{cases}$$

i.e. we have one remaining degree of freedom we have to fix. Setting $k_1 = 1$ yields:

$$\begin{cases} k_1 = 1 \\ k_2 = 0.4115 \\ k_3 = 0.0538 \\ K_a = 260 \end{cases}$$

Of course, we still need to verify whether these values represent acceptable gains and corresponding signals that are not clipped. E.g., choosing a different value for k_1 may allow reducing the value of K_a.

Example 3 An intriguing problem in control theory is the problem of the inverted pendulum. We all have tried to keep a broomstick upright on the palm of our hand by making fast hand movements. The same problem holds for a rocket that has thrusters at its bottom, that need to be controlled to keep the rocket upright during launch.

We can mathematically model this problem using the inverted pendulum of Figure 8.15. We assume the rod of length L has no mass and only has a mass m attached to its end. The input is the acceleration u we can give to the hinge H.

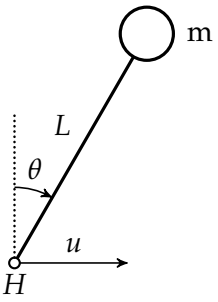


Figure 8.15: Inverted pendulum

The free body diagram for the mass in the coordinate system attached to the hinge (that accelerates with an acceleration u to the right), can be found in Figure 8.16.

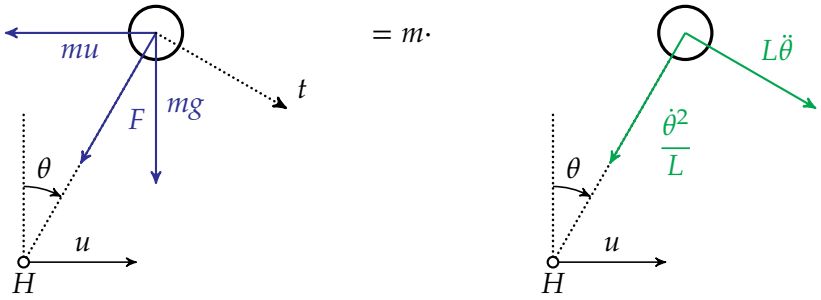


Figure 8.16: Free body diagram of the inverted pendulum referenced to a coordinate system that accelerates linearly with u; forces (in blue) on the left, accelerations (in green) on the right.

The forces on m have been indicated in the left-hand part. We have the gravitational force, and the inertial forces related to the accelerating coordinate system. The rod exercises a force F on the mass. The resulting accelerations have been indicated on the right-hand part. As the mass must move on a circular trajectory around H, we can consider a centripetal acceleration and a tangential acceleration that both are related to the angular position θ.

Because we are not interested in force F, we can just write down Newton's law for the tangential direction (along the t-axis), i.e.

$$\begin{aligned} -mu\cos\theta + mg\sin\theta &= mL\ddot{\theta} \\ L\ddot{\theta} - g\sin\theta &= -u\cos\theta \end{aligned} \tag{8.11}$$

For small values of θ, we can linearize the trigonometric functions in this equation as:

$$\sin\theta \approx \theta \qquad\qquad \cos\theta \approx 1$$

This leads to the following linearized description:

$$L\ddot{\theta} - g\theta = -u$$

Choosing the state variables to be $x_1 = \theta$ and $x_2 = \dot{\theta}$, we can write:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{g}{L}x_1 - \frac{1}{L}u \end{aligned}$$

In our matrix formalism, this becomes:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ \frac{g}{L} & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ -\frac{1}{L} \end{bmatrix} u$$

If we calculate the characteristic equation,

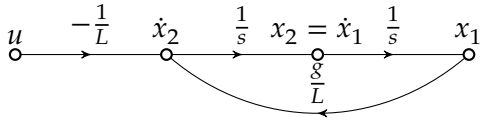
$$\det(sI - A) = 0$$
$$s^2 - \frac{g}{L} = 0$$

then the obvious is confirmed: the system is unstable, as it has a pole in the right half plane:

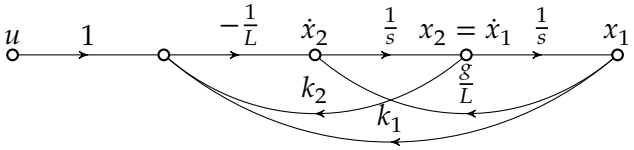
$$s_{1,2} = \pm\sqrt{g}L$$

This corresponds to our expectation, as in practice keeping a broomstick upright starting from $u = 0$ and $\theta = 0$ will fail: it is an unstable equilibrium.

Can we make this system stable? Again, let's try an external feedback configuration. The original system corresponds to the following signal flow graph:



Adding external feedback, based on the state variables x_1 and x_2, corresponds to the following graph:



Inspecting this graph, the following state equation can be written down:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -\frac{1}{L}\left(k_1 x_1 + k_2 x_2 + u\right) + \frac{g}{L} x_1$$

Equivalently:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ \frac{g-k_1}{L} & -\frac{k_2}{L} \end{bmatrix} x + \begin{bmatrix} 0 \\ -\frac{1}{L} \end{bmatrix} u$$

This allows for calculating the corresponding characteristic equation:

$$\det\left(\begin{bmatrix} s & -1 \\ -\frac{g-k_1}{L} & s + \frac{k_2}{L} \end{bmatrix}\right) = 0$$
$$s^2 + \frac{k_2}{L} s + \frac{k_1 - g}{L} = 0$$

The condition for this equation to only yield poles in the left half plane, is (based on the Routh-Hurwitz criterion):

$$\begin{cases} \dfrac{k_1 - g}{L} > 0 \\ \dfrac{k_2}{L} > 0 \end{cases}$$

i.e.

$$\begin{cases} k_1 > g \\ k_2 > 0 \end{cases}$$

The second condition, teaches us that the system cannot be made stable without feeding back the information on x_2 (i.e $\dot{\theta}$, the angular speed of the rod around the hinge).

Note that besides allowing for an elegant description of this second-order case, the state-space description also enables simulating the system without the linearization. We start again from (8.11):

$$\begin{gathered} L\ddot{\theta} - g\sin\theta = -u\cos\theta \\ \downarrow \quad x_1 = \theta \text{ and } x_2 = \dot{\theta} \\ L\dot{x}_2 = g\sin x_1 - u\cos x_1 \end{gathered}$$

This allows writing the following nonlinear first-order state equations:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \dfrac{g}{L}\sin x_1 - u\dfrac{1}{L}\cos x_1 \end{cases}$$

Applying the forward Euler approximation for the derivatives results in:

$$\begin{cases} x_1[n+1] = x_1[n] + T_s x_2[n] \\ x_2[n+1] = x_2[n] + T_s\left(\dfrac{g}{L}\sin(x_1[n]) - u[n]\dfrac{1}{L}\cos(x_1[n])\right) \end{cases}$$

For the system with external feedback $u = k_1x_1 + k_2x_2$ and therefore the equations become:

$$\begin{cases} x_1[n+1] = x_1[n] + T_s x_2[n] \\ x_2[n+1] = x_2[n] + T_s\left(\dfrac{g}{L}\sin(x_1[n]) - \dfrac{k_1x_1[n] + k_2x_2[n]}{L}\cos(x_1[n])\right) \end{cases}$$

Exercises

Exercise 8.7.4-1: Draw the root locus for the inverted pendulum with characteristic equation:

$$s^2 + \frac{k_2}{L}s + \frac{k_1 - g}{L} = 0$$

as a function of k_1.
Use $g = 10\,\mathrm{m/s^2}$ and $L = 0.4\,\mathrm{m}$. Draw two cases: (1) $k_2 = 0$ and (2) $k_2 = 2\,\mathrm{m/s}$.

Exercise 8.7.4-2: Draw the root locus for the inverted pendulum with characteristic equation:

$$s^2 + \frac{k_2}{L}s + \frac{k_1 - g}{L} = 0$$

as a function of k_2.
Use $g = 10\,\mathrm{m/s^2}$ and $L = 0.4\,\mathrm{m}$. Draw two cases: (1) $k_1 = 0$ and (2) $k_1 = 20\,\mathrm{m/s^2}$.

8.8 Conclusion

The state-space description for linear systems allows for a detailed control over the closed loop poles of the resulting feedback system using two basic configurations: internal feedback and external feedback. It also proves to be useful as a basis to simulate non LTI-systems.

Appendix A

Mathematical Bits and Pieces

A.1 The binomial theorem

A binomial is a sum of two terms that are represented by a different variable, e.g., $x + y$ or $a + b$.

The binomial theorem helps us in calculating natural powers of binomials, i.e. $(x + y)^n$. It states:

$$(x + y)^n = \sum_{i=0}^{n} \binom{n}{i} x^{n-i} y^i$$

with

$$\binom{n}{i} = \frac{n!}{(n-i)!i!}$$

The latter is the so-called *binomial coefficient* that also plays a role in combinatorics. It allows calculating in how many ways one can pick a set of i elements out of a set of n elements in the case the picking order is not relevant.

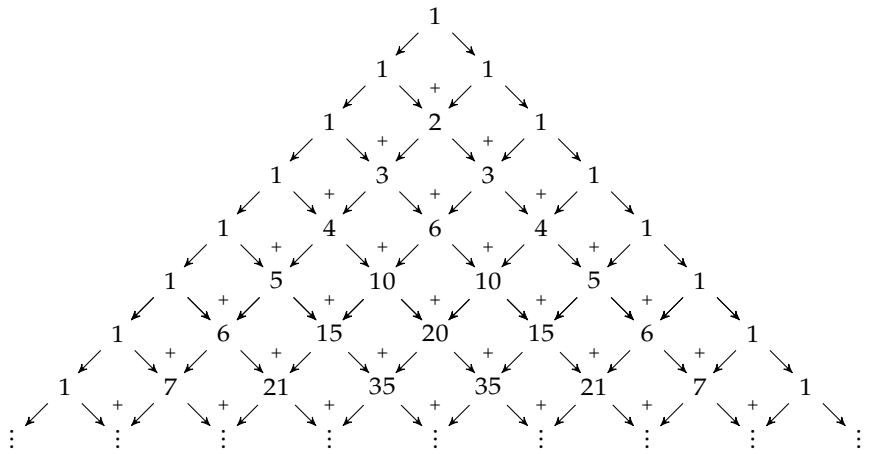
We will not prove the binomial theorem, but we will provide a little more insight by relating it to Pascal's well known triangle.

Let's start by considering powers of binomials for increasing number n. You will find it easy to verify:

$$\begin{aligned}
(x + y)^0 &= 1 \\
(x + y)^1 &= 1 \cdot x + 1 \cdot y \\
(x + y)^2 &= 1 \cdot x^2 + 2 \cdot xy + 1 \cdot y^2 \\
(x + y)^3 &= 1 \cdot x^3 + 3 \cdot x^2 y + 3 \cdot xy^2 + 1 \cdot y^3 \\
(x + y)^4 &= 1 \cdot x^4 + 4 \cdot x^3 y + 6 \cdot x^2 y^2 + 4 \cdot xy^3 + 1 \cdot y^4 \\
(x + y)^5 &= 1 \cdot x^5 + 5 \cdot x^4 y + 10 \cdot x^3 y^2 + 10 \cdot x^2 y^3 + 5 \cdot xy^4 + 1 \cdot y^5 \\
(x + y)^6 &= \cdots
\end{aligned}$$

It does not take long to see a regular pattern in the coefficients of these resulting polynomials. This pattern is known as Pascal's triangle. It was not invented by Pascal though. One of the earliest sources of the triangle is due to Acharya Pingala, an Indian mathematician, in 200BC. You can generate the triangle with a one at the top, keep copying that 1 on the outside elements diagonally down to the left and to the right and filling the spaces in between with the sum of the two elements above.

This results in the following triangle:



A special application of the binomial theorem, is that if $x \gg y$, the following approximation is good (and better than just neglecting y):

$$(x+y)^n \approx x^n + nx^{n-1}y$$

A.2 Vector and matrix notational conventions

When writing down vectors and matrices in the context of linear algebra, some simple notational conventions can help you to keep an overview.

A.2.1 Printed texts

Vector coordinates are often formalized into column matrices. The symbol denoting them is lowercase upright and printed in boldface. E.g.

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

In case row matrices are used, the symbol denoting them is equipped with a superscript T denoting the tranpose. E.g.

$$x^\top = \begin{bmatrix} x_1 & x_2 \end{bmatrix}$$

The symbol to denote matrices that have more than a single row or column is uppercase upright and printed in boldface. E.g.

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

Scalar variables are still denoted in italics as usual.

A.2.2 Handwritten texts

Writing boldface using a pencil or a pen is not convenient. Therefore the boldface is not used in handwritten texts. Sometimes a vector arrow is used.

Bibliography

[DB11] Richard C. Dorf and Robert H. Bishop. *Modern Control Systems*. Prentice Hall, 12th edition, 2011.

[Eva50] Walter R. Evans. Control systems synthesis by root locus method. 67(1), 1950.

[Hal43] Albert C. Hall. *The Analysis and Synthesis of Linear Servomechanisms*. Technology Press, M.I.T., Cambridge, MA, 1943.

[Hur95] Adolf Hurwitz. Ueber die bedingungen, unter welchen eine gleichung nur wurzeln mit negativen reellen theilen besitzt. *Mathematischen Annalen*, 46(2), 1895.

[JNP47] Hubert M. James, Nathaniel B. Nichols, and Ralph S. Phillips. *Theory of Servomechanisms*. McGraw-Hill Book Company, 1st edition, 1947.

[Mil40] J. Millman. A useful network theorem. *Proceedings of the IRE*, 28(9):413–417, 1940.

[Rou77] Edward John Routh. *A Treatise on the Stability of a Given State of Motion: Particularly Steady Motion*. Macmillan, 1877.

[Van18] Mark Vanpaemel. *Regeltechniek* (*in Dutch*). Universiteit Antwerpen, 2018.

Index