

Control Systems II

Gioele Zardini
gzardini@student.ethz.ch

June 3, 2017

Abstract

This "Skript" is made of my notes from the lecture *Control Systems II* of Dr. Gregor Ochsner (literature of Prof. Dr. Lino Guzzella) and from my lectures as teaching assistant of 2017 for the lecture of Dr. Guillaume Ducard.

This document, should give the chance to repeat one more time the contents of the lecture *Control Systems II* and practice them through many examples and exercises.

A special thanks goes to Mr. Nicolas Lanzetti, friend of mine and best teaching assistant I ever had: he taught me this course and created part of the material, as for example some of the drawings. The updated version of the Skript is available on n.ethz.ch/~gzardini/.

I cannot guarantee on the correctness of what is included in this Skript: it is possible that small errors occur. For this reason I am very grateful to get feedbacks and corrections, in order to improve the quality of the literature.

Enjoy your Control Systems II!

Gioele Zardini

Version Update:

Version 1: June 2017

Contents

1	Recapitulation Control Systems I	6
1.1	System Modeling	6
1.2	Linear System Analysis	7
1.2.1	Stability (Lyapunov)	7
1.2.2	Controllability	7
1.2.3	Observability	7
1.3	Feedback System Analysis	8
1.3.1	Crossover Frequency	8
1.3.2	Phase Margin φ	9
1.3.3	Gain Margin γ	9
1.4	Nyquist theorem	10
1.5	Specifications for the closed-loop	10
1.6	Examples	12
2	Synthesis of SISO Control Systems	15
2.1	Loop Shaping	15
2.1.1	Plant inversion	15
2.1.2	Loop shaping for Nonminimumphase systems	15
2.1.3	Loop shaping for unstable systems	17
2.1.4	Feasibility	17
2.2	Cascaded Control Systems	18
2.2.1	Structure of cascaded systems	18
2.2.2	Time-Scale Estimation	18
2.2.3	Design Process	19
2.2.4	Examples	20
2.3	Predictive Control	24
2.3.1	Why predictive control	24
2.3.2	The Smith Predictor	24
2.3.3	Analysis	25
2.3.4	Examples	27
2.4	Robustness	30
2.4.1	Robust Stability	30
2.4.2	Nominal Performance	30
2.4.3	Robust Performance	30
2.4.4	Examples	32
3	Implementation of Controllers	35
3.1	Setpoint Weighting for PID	35
3.1.1	Recap Goals	35
3.1.2	Method	35
3.1.3	Application to Reference Tracking	36
3.1.4	Examples	37
3.2	Feed Forward	39
3.3	Saturation Problem	40
3.3.1	Examples	42
3.4	Digital Control	46
3.4.1	Signals and Systems	46
3.4.2	Discrete-Time State-Space Description	46

3.4.3	Discrete-Time Control Systems	47
3.4.4	Controller Discretization/Emulation	49
3.4.5	Discrete Time Controller Synthesis	51
3.4.6	Examples	53
4	MIMO Systems	66
4.1	System Description	66
4.1.1	State Space Descriprion	66
4.1.2	Transfer Function	66
4.2	System Analysis	69
4.2.1	Lyapunov Stability	69
4.2.2	Controllability	69
4.2.3	Observability	69
4.3	Poles and Zeros	70
4.3.1	Zeros	70
4.3.2	Poles	70
4.3.3	Directions	70
4.4	Nyquist Theorem for MIMO Systems	70
4.5	Examples	71
4.6	Relative-Gain Array (RGA)	83
5	Frequency Response of MIMO Systems	98
5.1	Singular Value Decomposition (SVD)	98
5.1.1	Preliminary Definitions	98
5.1.2	Singular Value Decomposition	99
5.2	Frequency Responses	108
5.2.1	Maximal and minimal Gain	109
5.2.2	Robustness and Disturbance Rejection	110
6	Synthesis of MIMO Control Systems	117
6.1	The Linear Quadratic Regulator (LQR)	117
6.1.1	Definition	117
6.1.2	Kochrezept	119
6.1.3	Frequency Domain	119
6.1.4	Properties	119
6.1.5	Pole Placement	120
6.1.6	Examples	122
6.2	Extensions of the LQR problem	132
6.2.1	The LQRI	132
6.2.2	Finite Horizon LQR	138
6.2.3	Feedforward LQR Control Systems	139
6.3	Numerical Optimization	140
6.3.1	Definition	140
6.3.2	Examples	141
6.4	Model Predictive Control (MPC)	143
6.4.1	Definition	143
6.4.2	Examples	144
6.5	The Linear Quadratic Gaussian (LQG)	147
6.5.1	Observer	147
6.5.2	The LQG Controller	148
6.5.3	Kochrezept	149

6.5.4	Examples	150
6.6	Extension of LQG	155
6.6.1	The LQGI	155
6.7	LTR	156
6.7.1	Definition	156
6.7.2	β method	157
6.7.3	LQG/LQR for $m \geq p$	157
6.7.4	LQG/LQR for $p \geq m$	158
A	Linear Algebra	159
A.1	Matrix-Inversion	159
A.2	Differentiation with Matrices	159
A.3	Matrix Inversion Lemma	159
B	Rules	160
B.1	Trigo	160
B.2	Euler-Forms	160
B.3	Derivatives	160
B.4	Logarithms	160
B.5	Magnitude and Phase	161
B.6	dB-Skala	161
C	MATLAB	162
C.1	General Commands	162
C.2	Control Systems Commands	163
C.3	Plot and Diagrams	164

1 Recapitulation Control Systems I

In this section I've resumed all topics and concepts from the course *Control Systems I* that are important for *Control Systems II* as well. This chapter gives one more time the chance to have a look to fundamental concepts.

1.1 System Modeling

A whole course next semester will be focused on this section (*System Modeling*, Dr. Ducard). The normalization and linearization (around the equilibrium x_0, u_0) of a model result in the **state space description**

$$\begin{aligned}\frac{d}{dt}x(t) &= A \cdot x(t) + b \cdot u(t) \\ y(t) &= c \cdot x(t) + d \cdot u(t),\end{aligned}\tag{1.1}$$

where

$$A = \frac{\partial f_0}{\partial x} \Big|_{x=x_0, u=u_0} = \begin{pmatrix} \frac{\partial f_{0,1}}{\partial x_1} \Big|_{x=x_0, u=u_0} & \cdots & \frac{\partial f_{0,1}}{\partial x_n} \Big|_{x=x_0, u=u_0} \\ \vdots & \cdots & \vdots \\ \frac{\partial f_{0,n}}{\partial x_1} \Big|_{x=x_0, u=u_0} & \cdots & \frac{\partial f_{0,n}}{\partial x_n} \Big|_{x=x_0, u=u_0} \end{pmatrix},$$

$$b = \frac{\partial f_0}{\partial u} \Big|_{x=x_0, u=u_0} = \begin{pmatrix} \frac{\partial f_{0,1}}{\partial u} \Big|_{x=x_0, u=u_0} \\ \vdots \\ \frac{\partial f_{0,n}}{\partial u} \Big|_{x=x_0, u=u_0} \end{pmatrix}$$

$$c = \frac{\partial g_0}{\partial x} \Big|_{x=x_0, u=u_0} = \left(\frac{\partial g_0}{\partial x_1} \Big|_{x=x_0, u=u_0} \quad \cdots \quad \frac{\partial g_0}{\partial x_n} \Big|_{x=x_0, u=u_0} \right),$$

$$d = \frac{\partial g_0}{\partial u} \Big|_{x=x_0, u=u_0} = \left(\frac{\partial g_0}{\partial u} \Big|_{x=x_0, u=u_0} \right).$$

The **transfer function** of the system is given by

$$P(s) = c(s\mathbb{I} - A)^{-1}b + d = b_m \cdot \frac{(s - \zeta_1) \cdot (s - \zeta_2) \cdot \dots \cdot (s - \zeta_m)}{(s - \pi_1) \cdot (s - \pi_2) \cdot \dots \cdot (s - \pi_n)}.\tag{1.2}$$

- π_i are the **poles** of the system.
- ζ_i are the **zeros** of the system.

Both zeros and poles strong determine the system's behaviour.

Remark.

- The transfer function does not depend on the chosen system of coordinates.
- Fundamental system properties like **stability** and **controllability**, as well.

1.2 Linear System Analysis

1.2.1 Stability (Lyapunov)

Lyapunov stability analyses the behaviour of a system **near to an equilibrium point** for $u(t) = 0$ (no input). With the found formula $x(t) = x_0 \cdot e^{At}$ one can show that Lyapunov stability can be determined through the calculation of the eigenvalues λ_i of A . The outcomes are

- **Stable:** $\|x(t)\| < \infty \forall t \geq 0 \Leftrightarrow \operatorname{Re}(\lambda_i) \leq 0 \forall i$
- **Asymptotically stable:** $\lim_{t \rightarrow \infty} \|x(t)\| = 0 \Leftrightarrow \operatorname{Re}(\lambda_i) < 0 \forall i$
- **Unstable:** $\lim_{t \rightarrow \infty} \|x(t)\| = \infty \Leftrightarrow \operatorname{Re}(\lambda_i) > 0$ for at least one i

1.2.2 Controllability

Controllable: is it possible to control all the states of a system with an input $u(t)$?

A system is said to be **completely controllable**, if the **Controllability Matrix** R has full rank

$$R = \begin{pmatrix} b & A \cdot b & A^2 \cdot b & \dots & A^{n-1} \cdot b \end{pmatrix}. \quad (1.3)$$

1.2.3 Observability

Observable: is it possible to reconstruct the initial conditions of all the states of a system from the output $y(t)$?

A system is said to be **completely observable**, if the **Observability Matrix** O has full rank

$$O = \begin{pmatrix} c \\ c \cdot A \\ c \cdot A^2 \\ \vdots \\ c \cdot A^{n-1} \end{pmatrix}. \quad (1.4)$$

Detectability: A linear time-invariant system is detectable if all unstable modes are observable.

Stabilizability: A linear time-invariant system is stabilizable if all unstable modes are controllable.

1.3 Feedback System Analysis

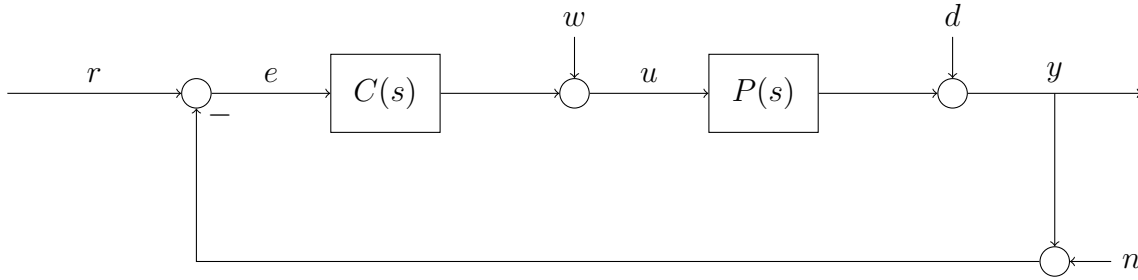


Figure 1: Standard feedback control system structure.

The **loop gain** $L(s)$ is the open-loop transfer function from $e \rightarrow y$ defined by

$$L(s) = P(s) \cdot C(s), \quad (1.5)$$

where $P(s)$ is the **plant** and $C(s)$ is the **controller**.

The **sensitivity** $S(s)$ is the closed-loop transfer function from $d \rightarrow y$ defined by

$$S(s) = \frac{1}{1 + L(s)}. \quad (1.6)$$

The **complementary sensitivity** $T(s)$ is the closed-loop transfer function from $r \rightarrow y$ defined by

$$T(s) = \frac{L(s)}{1 + L(s)}. \quad (1.7)$$

It is quite easy to show that

$$S(s) + T(s) = \frac{1}{1 + L(s)} + \frac{L(s)}{1 + L(s)} = 1. \quad (1.8)$$

holds. One can define the transfer function of the **output** signal $Y(s)$ with

$$Y(s) = T(s) \cdot R(s) + S(s) \cdot D(s) - T(s) \cdot N(s) + S(s) \cdot P(s) \cdot W(s) \quad (1.9)$$

and the transfer function of the **error** signal with

$$E(s) = S(s) \cdot (R(s) - D(s) - N(s) - P(s) \cdot W(s)), \quad (1.10)$$

where $R(s), D(s), W(s), N(s)$ are the laplace transforms of the respective signals $r(t), d(t), w(t), n(t)$.

1.3.1 Crossover Frequency

Normally, we want a high amplification at low frequencies for disturbance rejection and a low amplification at high frequencies for noise attenuation.

The **crossover frequency** ω_c is used as reference for the analysis of the system and is defined as the frequency for which it holds

$$|L(j\omega_c)| = 1 = 0 \text{ dB}. \quad (1.11)$$

- *Bode*: Frequency where $L(s)$ crosses the 0 dB line.
- *Nyquist*: Frequency where $L(s)$ crosses the unit circle.
- The bigger ω_c is, the more aggressive the system is.

1.3.2 Phase Margin φ

$$180^\circ \pm \angle L(j\omega_c). \quad (1.12)$$

- *Bode*: Distance from -180° line at crossover frequency ω_c . See Figure 2
- *Nyquist*: Angle distance between the -1 point and the point where $L(s)$ crosses the unit circle. See Figure 3

1.3.3 Gain Margin γ

γ is the inverse of the magnitude of $L(j\omega)$ if $\angle L(j\omega) = -180^\circ$. We call its frequency ω_γ, ω^* .

$$\frac{1}{\gamma} = |\operatorname{Re}(L(j\omega_\gamma))|, \quad 0 = |\operatorname{Im}(L(j\omega_\gamma))|. \quad (1.13)$$

- *Bode*: Distance from the 0dB line at frequency ω_γ . See Figure 2.
- *Nyquist*: Inverse of the distance between the origin and the intersection point of $L(s)$ and the real axis. See Figure 3

You have to think of the phase and the gain margin as **safety margins** for your system. These two are an indicator that tells you how close is the system to oscillation (instability). The **gain margin** and the **phase margin** are the gain and the phase which can be varied before the system becomes marginally stable, that is, just before instability.

This can be seen in Figure 2 and Figure 3.

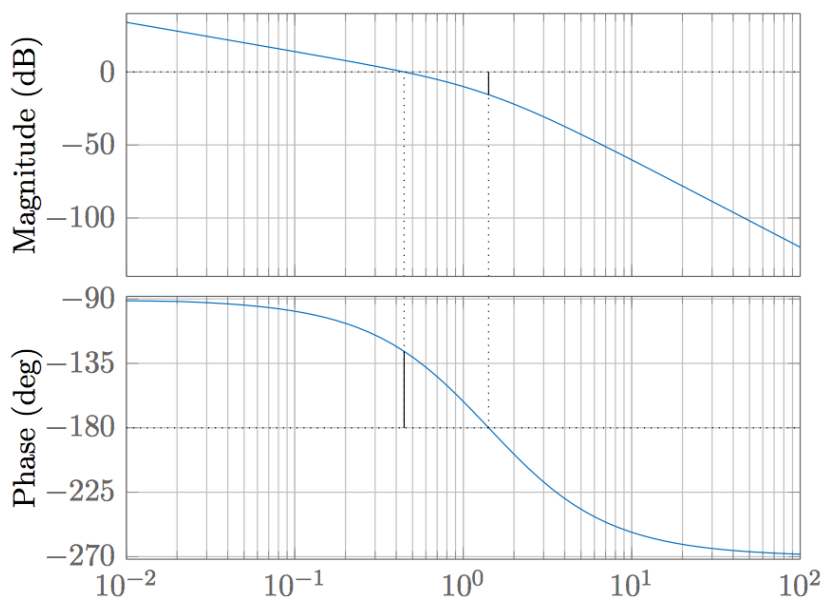


Figure 2: Bodeplot with phase and gain margin

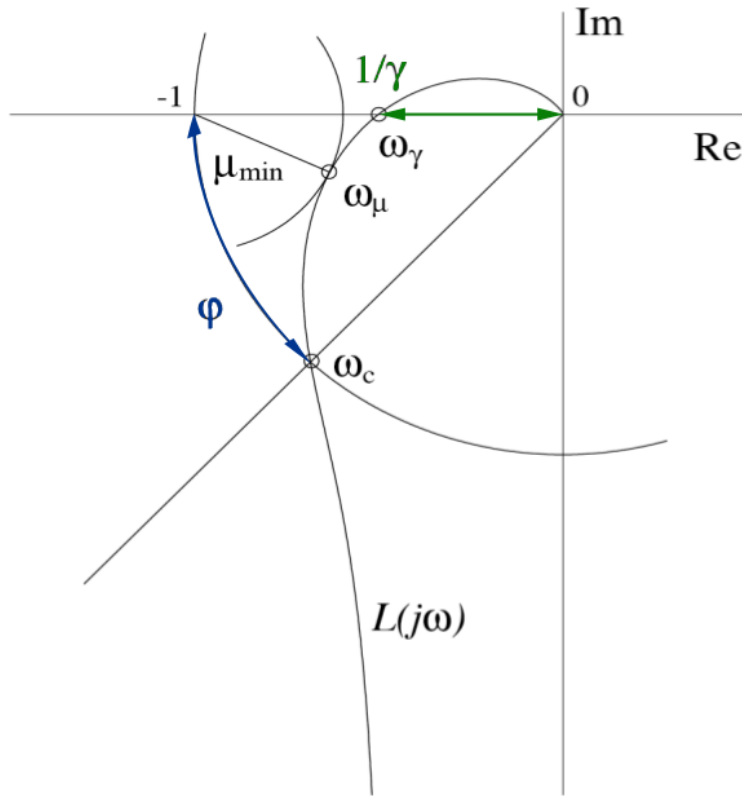


Figure 3: Nyquist plot with phase and gain margin

1.4 Nyquist theorem

A closed-loop system $T(s)$ is asymptotically stable if

$$n_c = n_+ + \frac{n_0}{2}. \quad (1.14)$$

holds, where

- n_c : Number of mathematical positive encirclements of $L(s)$ about critical point -1 (counterclockwise).
- n_+ : Number of unstable poles of $L(s)$ ($\text{Re}(\pi) > 0$).
- n_0 : Number marginal stable poles of $L(s)$ ($\text{Re}(\pi) = 0$).

1.5 Specifications for the closed-loop

The main objective of control systems is to design a controller. Before one does this, one should be sure that the system is controllable. We learned that the controllability and observability are powerful tools to do that, but they are not enough: there are some others conditions for the **crossover-frequency** ω_c that should be fulfilled for the system in order to design a controller. It holds

$$\max(10 \cdot \omega_d, 2 \cdot \omega_{\pi+}) < \omega_c < \min(0.5 \cdot \omega_{\zeta+}, 0.1 \cdot \omega_n, 0.5 \cdot \omega_{\text{delay}}, 0.2 \cdot \omega_2), \quad (1.15)$$

where

- $\omega_{\pi^+} = \text{Re}(\pi^+)$: Dominant (biggest with $\text{Re}(\pi) > 0$) unstable Pole;
- $\omega_{\zeta^+} = \text{Re}(\zeta^+)$: Dominant (smallest with $\text{Re}(\zeta) > 0$) nonminimumphase zero;
- ω_d : Biggest disturbance-frequency of the system;
- ω_n : Smallest noise-frequency of the system;
- ω_2 : Frequency with 100% uncertainty ($|W_2(j \cdot \omega_2)| = 1$),
- $\omega_{\text{delay}} = \frac{1}{T_{\text{delay}}}$: Biggest delay of the System.

1.6 Examples

Example 1. The dynamic equations of a system are given as

$$\begin{aligned}\dot{x}_1(t) &= x_1(t) - 5x_2(t) + u(t), \\ \dot{x}_2(t) &= -2x_1(t), \\ \dot{x}_3(t) &= -x_2(t) - 2x_3(t), \\ y(t) &= 3x_3(t).\end{aligned}$$

- (a) Draw the **Signal Diagram** of the system.
- (b) Find the **state space description** of the above system and write it in MATLAB.
- (c) Find out if the system is **Lyapunov** stable or not, in MATLAB.
- (d) Find the **Controllability** matrix and determine if the system is controllable in, MATLAB.
- (e) Find the **Observability** matrix and determine if the system is observable, in MATLAB.
- (f) Find the **transfer function** of the system with its poles and zeros, in MATLAB

Solution.

(a) The **Signal Diagram** reads

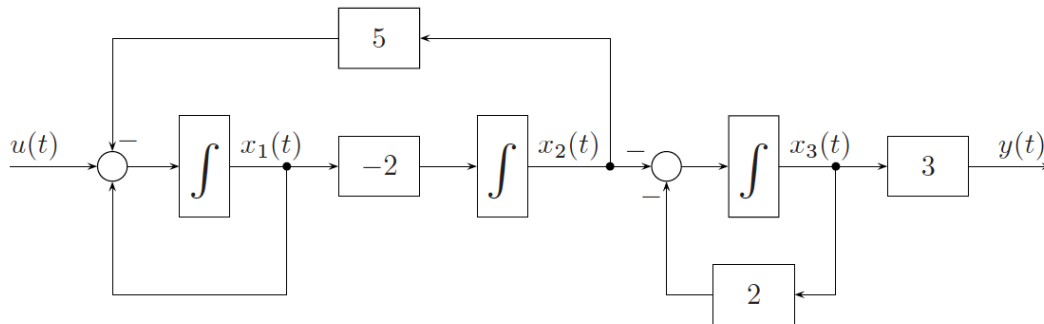


Figure 4: Signal Diagram of **Example 1**.

(b) The state space description has the following matrices:

$$A = \begin{pmatrix} 1 & -5 & 0 \\ -2 & 0 & 0 \\ 0 & -1 & -2 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad c = (0 \ 0 \ 3), \quad d = 0.$$

For the rest of solution **(b)** and solutions **(c)**-**(f)** see the attached MATLAB script: `Example_1_EC1.m`.

Example 2. Given are a plant $P(s) = \frac{1}{(s+1)(s+2)}$ and its controller $C(s) = \frac{4}{(s+1)}$.

- (a) Compute $L(s)$ and $T(s)$, with MATLAB.
- (b) Find the **gain** and the **phase** margin numerically and plot them, with MATLAB.
- (c) Write the **Bodeplot** of $P(s), C(s), L(s)$, MATLAB.
- (d) Write the **Nyquist** plot of $P(s), L(s)$, with MATLAB.
- (e) Plot the **Step response** for the open and closed loop, with MATLAB
- (f) Plot the **Impulse response** for the open and the closed loop, with MATLAB

Solution. See the attached MATLAB script: `Example_2_EC1.m`.

2 Synthesis of SISO Control Systems

2.1 Loop Shaping

2.1.1 Plant inversion

This method isn't indicated for nonminimumphase plants and for unstable plants: in those cases this would lead to nonminimumphase or unstable controllers. This method is indicated for *simple* systems for which it holds

- Plant is asymptotically stable.
- Plant is minimumphase.

The method is then based on a simple step:

$$L(s) = C(s) \cdot P(s) \Rightarrow C(s) = L(s) \cdot P(s)^{-1}. \quad (2.1)$$

The choice of the loop gain is free: it can be chosen such that it meets the specifications.

2.1.2 Loop shaping for Nonminimumphase systems

A nonminimumphase system shows a wrong response: a change in the input results in a change in sign, that is, the system initially lies. Our controller should therefore be *patient* and for this reason we use a *slow* control system. This is obtained by a crossover frequency that is smaller than the nonminimumphase zero. One begins to design the controller with a **PI-Controller**.

$$C(s) = k_p \cdot \frac{T_i \cdot s + 1}{T_i \cdot s}. \quad (2.2)$$

where parameters k_p and T_i can be chosen such that the loop gain $L(s)$ meets the known specifications. One can reach better robustness with **Lead/Lag** elements of the form

$$C(s) = \frac{T \cdot s + 1}{\alpha \cdot T \cdot s + 1}. \quad (2.3)$$

where $\alpha, T \in \mathbb{R}^+$. One can understand the Lead and the Lag as

- $\alpha < 1$: **Lead-Element**:
 - Phase margin increases.
 - Loop gain's increases.
- $\alpha > 1$: **Lag-Element**:
 - Phase margin decreases.
 - Loop gain's decreases.

Like one can see in Figure 5 and Figure 6 the maximal benefits are reached at frequencies ($\hat{\omega}$) where the drawbacks are not yet fully developed.

The element's parameters can be calculated as

$$\alpha = \left(\sqrt{\tan^2(\hat{\varphi}) + 1} - \tan(\varphi) \right)^2 = \frac{1 - \sin(\hat{\varphi})}{1 + \sin(\hat{\varphi})} \quad (2.4)$$

and

$$T = \frac{1}{\hat{\omega} \cdot \sqrt{\alpha}}. \quad (2.5)$$

where $\hat{\omega}$ the desired **center frequency** is and $\hat{\varphi} = \varphi_{new} - \varphi$ the desired **maximum phase shift** (in *rad*) is. The classic loop-shaping method reads:

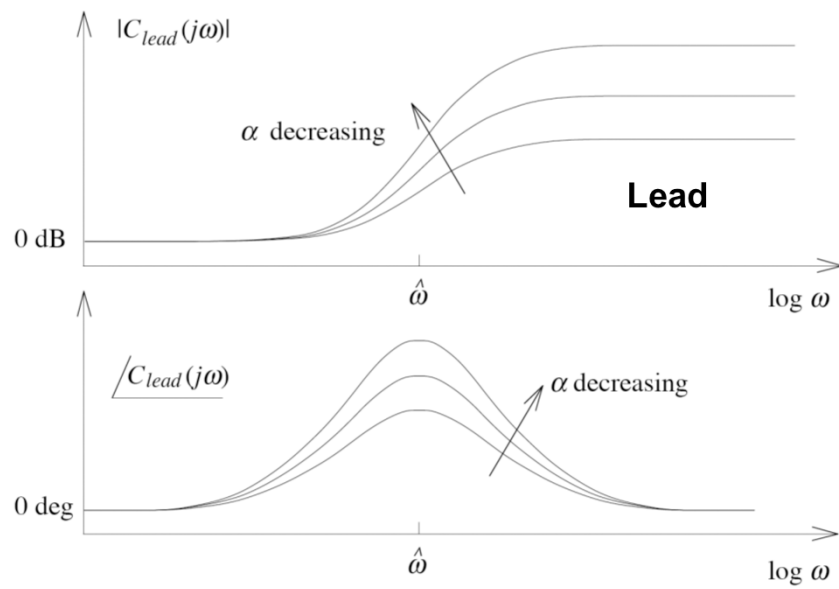


Figure 5: Bodeplot of lead Element

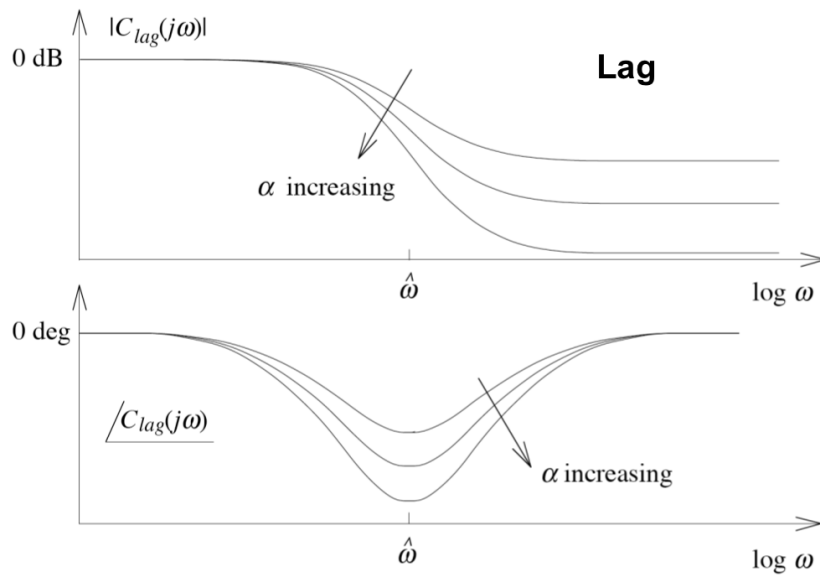


Figure 6: Bodeplot of lag Element

1. Design of a PI(D) controller.
2. Add Lead/Lag elements where needed¹
3. Set the gain of the controller k_p such that we reach the desired crossover frequency.

¹ $L(j\omega)$ often suits not the learned requirements

2.1.3 Loop shaping for unstable systems

Since the Nyquist's theorem should always hold, if it isn't the case, one has to design the controller such that $n_c = n_+ + \frac{n_0}{2}$ is valid. To remember is: stable poles *increase* the phase by 90° and minimumphase zeros *decrease* the phase by 90°

2.1.4 Feasibility

Once the controller is designed, one has to look if this is really feasible and possible to implement. That is, if the number of poles is equal or bigger than the number of zeros of the system. If that is not the case, one has to add poles at high frequencies, such that they don't affect the system near the crossover frequency. One could e.g. add to a PID controller a *Roll-Off Term* as

$$C(s) = k_p \cdot \underbrace{\left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s\right)}_{\text{PID Controller}} \cdot \underbrace{\frac{1}{(\tau \cdot s + 1)^2}}_{\text{Roll-Off Term}}. \quad (2.6)$$

2.2 Cascaded Control Systems

Single input single output (SISO) systems use, in feedback control, the output y as unique information to compute u . Clearly this doesn't represent a big part of the systems existing in the real world.

A first step towards a more precise solution, is to consider a situation where multiple outputs are available: the single input multiple output case (SIMO). To analyze such systems we need the theory and the methods that will be learned for the multiple input multiple output (MIMO) case. One can but still, without introducing this theory, use the concepts learned so far to handle such a system.

2.2.1 Structure of cascaded systems

The structure of a cascaded system can be seen in Figure 7. Special cases that you could encounter in exercises can be seen in Figure 8 and Figure 9. .

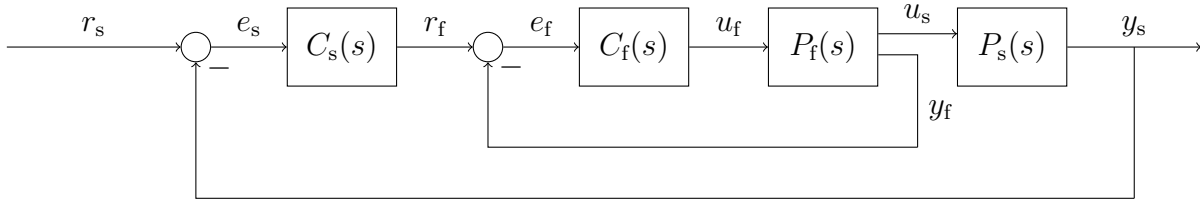


Figure 7: Control loop structure of a cascaded system.

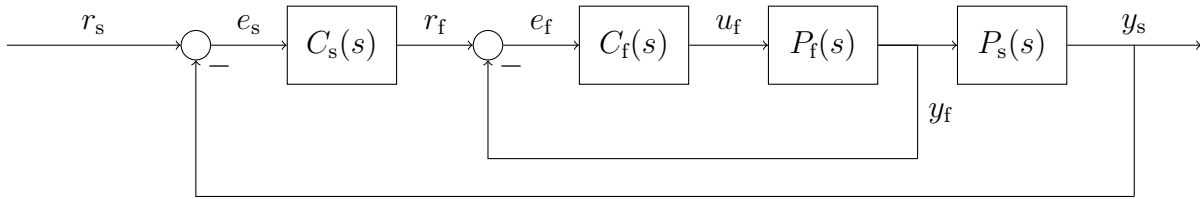


Figure 8: Control loop structure of a cascaded system (with $u_s = y_f$).

The main idea here is to divide the system basing on its *time-scale*. Usually, one sets an inner *faster* loop and an outer *slower* loop. One can identify these elements in the fast dynamic's plant $P_f(s)$ and in the slow dynamic's plant $P_s(s)$. Basically, one wants to design a *faster controller* $C_f(s)$ for the inner loop and a *slower controller* $C_s(s)$ for the outer loop.

2.2.2 Time-Scale Estimation

How can we distinguish between a *slower* and a *faster* control loop? A good way to do it consists in looking at the poles and the delays of the plant. Let's have a look at the Inverse Laplace Transfrom of the output function $Y(s)$:

$$y(t) = \sum_{i=1}^p \sum_{k=1}^{\phi_i} \frac{\rho_{i,k}}{(k-1)!} \cdot t^{k-1} \cdot e^{\pi_i \cdot t} \cdot h(t). \quad (2.7)$$

The dynamics of the plant are described by the absolutely *smallest* pole.

Example 3. We are given two plants $P_1(s)$ and $P_2(s)$. The poles π_1 and π_2 can be seen in the following table. We have to distinguish between $P_s(s)$ and $P_f(s)$.

1)	System	π_1	π_2	2)	System	π_1	π_2
	$P_1(s)$	-1	-100		$P_1(s)$	-1	100
	$P_2(s)$	-10	-50		$P_2(s)$	-1	10

1) $P_1(s)$ has the smallest pole: this means $P_s(s) = P_1(s)$ and $P_f(s) = P_2(s)$.

2) Both systems share the same smallest pole. This means that as there isn't a different time-scale, the two systems should not be controlled with the cascaded control principle.

2.2.3 Design Process

Faster Loop

The faster controller $C_f(s)$ is (e.g. with Ziegler and Nichols) designed first without taking into account the slower loop. In order to exploit the most of the available bandwidth², one chooses a P(D) controller, without the integrator.

Slower Loop

The slower controller $C_s(s)$ is (e.g. with Ziegler and Nichols) designed as second: here, one has to consider the faster closed loop as part of the *new* plant. Since this controller should obtain accuracy, one usually include the integrator here and chooses a PI(D) controller³.

²An Integrator reduces the bandwidth. The bandwidth defines where our controller is working: we want the controller to have an action/reaction as big as possible.

³The integrator reduces the static error!

2.2.4 Examples

Example 4. Figure 9 shows a single input multiple output (SIMO) loop, that should be controlled with cascaded control. The given transfer functions are

$$\begin{aligned} P_f(s) &= \frac{1}{s+1}, \\ P_s(s) &= \frac{1}{5s^2 + 6s + 1}, \\ C_f(s) &= 2. \end{aligned} \tag{2.8}$$

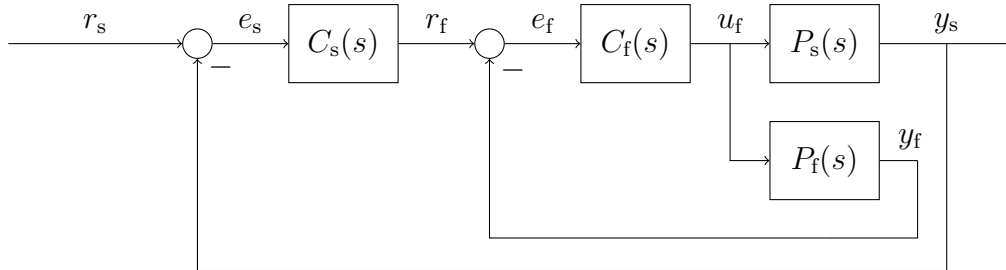


Figure 9: Control loop structure of a cascaded system.

Compute the transfer function of $P_{out}(s) : r_f \rightarrow y_s$.

Solution. To solve such problems, one has to work step by step. First, we use the frequency domain to determine the signals' dependencies. It holds:

$$Y_s(s) = P_s(s) \cdot U_f(s), \quad (2.9)$$

where

$$\begin{aligned} U_f(s) &= C_f(s) \cdot (R_f(s) - P_f(s) \cdot U_f(s)) \\ U_f(s) \cdot (1 + P_f(s) \cdot C_f(s)) &= C_f(s) \cdot R_f(s) \\ U_f(s) &= \frac{C_f(s) \cdot R_f(s)}{1 + P_f(s) \cdot C_f(s)}. \end{aligned} \quad (2.10)$$

2.10 in 2.9 gives

$$\begin{aligned} P_{out}(s) &= \frac{Y_s(s)}{R_f(s)} = P_s(s) \cdot \frac{C_f(s)}{1 + P_f(s) \cdot C_f(s)} \\ &= \frac{1}{5s^2 + 6s + 1} \cdot \frac{2 \cdot (s + 1)}{(s + 3)} \\ &= \frac{2}{(5s + 1) \cdot (s + 3)}. \end{aligned} \quad (2.11)$$

Example 5. Consider the heated water tank depicted in Figure 10. The goal of this exercise is to design a controller for this system. The **valve position** u should be controlled such that the **outlet water temperature** T_w follows a given reference $T_{w,ref}$. The water outlet temperature, as well as the **steam mass flow** \dot{m}_s , are measured. Cascaded control can be used and the chosen controller structure is depicted in Figure 11.

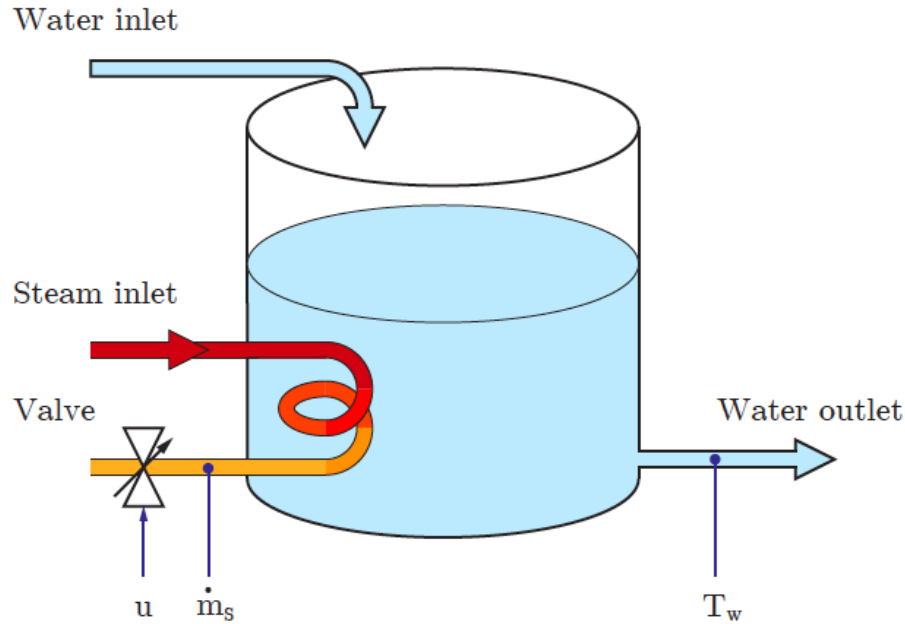


Figure 10: Heating water in the water tank.

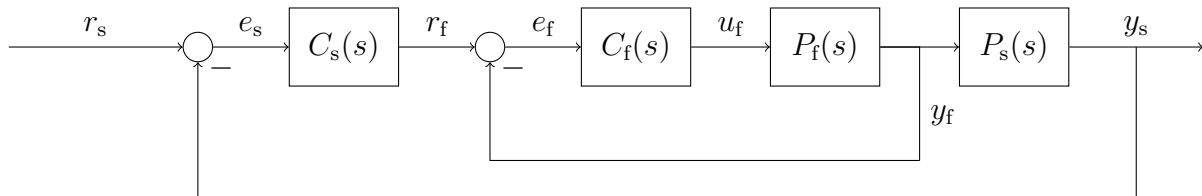


Figure 11: Control loop structure of the cascaded system.

- Which values of the system correspond to the signals r_f , y_f , r_s and y_s ?
- Derive the outer/slow closed-loop transfer function $T_{r_s \rightarrow y_s}$ as a function of the closed-loop transfer function $T_{r_f \rightarrow y_f}$.

Solution.

- (a) The two values that describe this system are the steam mass flow \dot{m}_S and the water outlet temperature T_w . The temperature changes significantly slower than the steam mass flow, hence: the output of the outer/slower control loop y_s corresponds to the water outlet temperature T_w . Its reference signal r_s is the reference of this temperature $T_{w,ref}$. The output of the inner/fast control loop y_f corresponds to the steam mass flow \dot{m}_S . Its reference signal r_f is the reference of the steam mass flow $\dot{m}_{S,ref}$.
- (b) The transfer function of the inner/fast closed-loop is

$$\begin{aligned}
 T_{r_f \rightarrow y_f} &= T_f \\
 &= \frac{L_f}{1 + L_f} \\
 &= \frac{C_f \cdot P_f}{1 + C_f \cdot P_f}.
 \end{aligned}$$

The transfer function of the outer/slow closed-loop is

$$\begin{aligned}
 T_{r_s \rightarrow y_s} &= T_s \\
 &= \frac{L_s}{1 + L_s} \\
 &= \frac{C_s \cdot T_f \cdot P_s}{1 + C_s \cdot T_f \cdot P_s}.
 \end{aligned}$$

2.3 Predictive Control

2.3.1 Why predictive control

If a system has *substantial* delays, it is very difficult to control with a normal PID controller. The I part of the controller causes *impatience*, that is, integrates overtime. As a practical example think of taking a shower at morning: one let the water flow and of course this hasn't the desired temperature. For this reason one chooses warmer water by turning the temperature controller, the water becomes too hot and so one turns it to colder water and so on, resulting in a non optimal strategy. Moreover, the D part of the controller is practically useless⁴. What does the expression *substantial* delays mean? As indicative range one can say that it is worth using predictive control if

$$\frac{T}{T + \tau} > 0.3, \quad (2.12)$$

where T is the delay and τ is the time constant of the system. Other prerequisites are

- The plant must be asymptotically stable.
- A good model of the plant should be available.

2.3.2 The Smith Predictor

One can see the two *equivalent* structures of the Smith Predictor in Figure 12 and Figure 13.

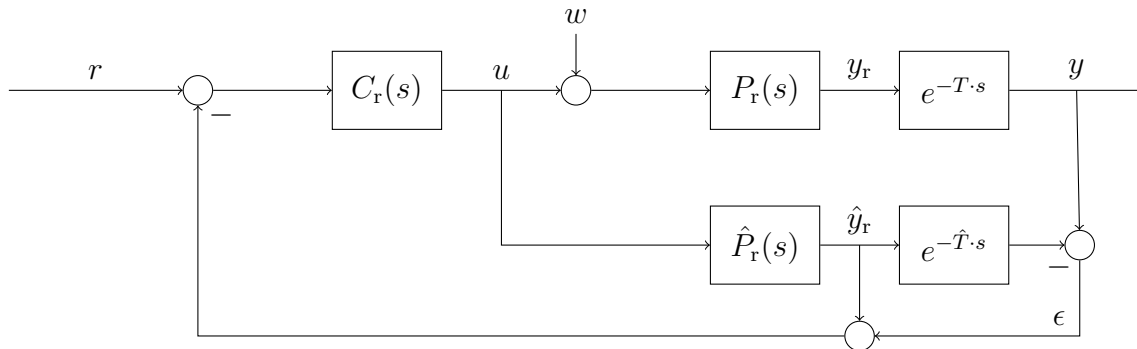


Figure 12: Structure of the Smith predictor.

If the system has big delays, one can assume that it is possible to write the delay element and the nondelayed plant as a product in the frequency domain: that's what is done in the upper right side of Figure 12. This means that the transfer function $u \rightarrow y$ can be written as

$$P(s) = P_r(s) \cdot e^{-sT}. \quad (2.13)$$

Main Idea:

As long as we have no disturbance w ($w = 0$) and our model is good enough (this means $P_r(s) = \hat{P}_r(s)$, $T = \hat{T}$)⁵, we can model a non delayed plant and get the non delayed output $\hat{y}_r(t)$ (one can see this on the lower right side of Figure 12). The feedback signal results from the sum of $\hat{y}_r(t)$ and the correction signal ϵ .

⁴Taking the derivative of a delay element doesn't help to control it

⁵We use $\hat{\cdot}$ to identify the parameters of the model

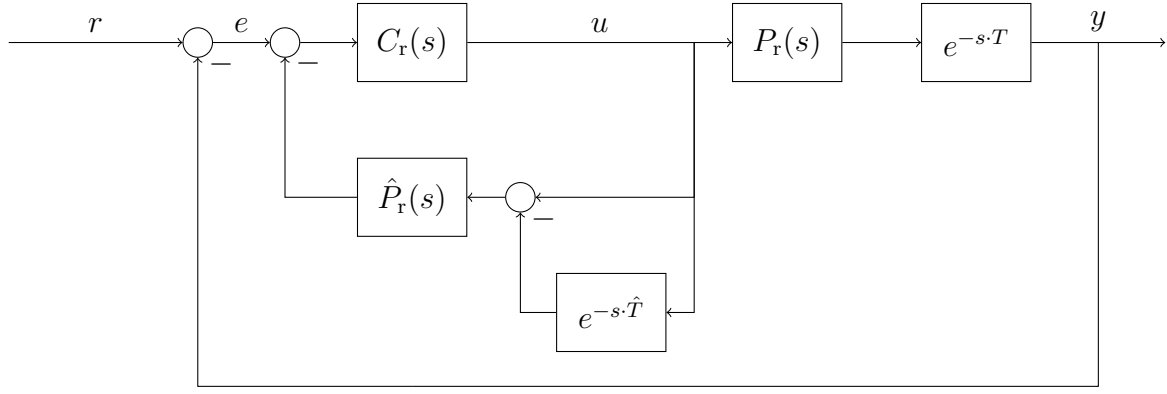


Figure 13: Structure of the Smith predictor.

2.3.3 Analysis

The controller of the system is the transfer function $e \rightarrow u$, which can be computed as

$$\begin{aligned} U(s) &= C_r(s) \cdot (R(s) - \hat{P}_r(s) \cdot U(s) + \hat{P}_r(s) \cdot e^{-s\hat{T}} \cdot U(s) - Y(s)) \\ &= C_r(s) \cdot \underbrace{(R(s) - Y(s))}_{E(s)} - C_r(s) \cdot \hat{P}_r(s) \cdot (1 - e^{-s\hat{T}}) \cdot U(s). \end{aligned} \quad (2.14)$$

If one solves for $U(s)$:

$$U(s) = \frac{C_r(s)}{1 + C_r(s) \cdot \hat{P}_r(s) \cdot (1 - e^{-s\hat{T}})} \cdot E(s) \quad (2.15)$$

and so the transfer function of the controller reads

$$C(s) = \frac{C_r(s)}{1 + C_r(s) \cdot \hat{P}_r(s) \cdot (1 - e^{-s\hat{T}})}. \quad (2.16)$$

This means that the loop gain is

$$L(s) = P(s) \cdot C(s) = \frac{C_r(s) \cdot P_r(s) \cdot e^{-sT}}{1 + C_r(s) \cdot \hat{P}_r(s) \cdot (1 - e^{-s\hat{T}})}. \quad (2.17)$$

If one assumes as stated, that the model is good enough s.t. $P_r(s) = \hat{P}_r(s)$, $T = \hat{T}$, one gets

$$\begin{aligned} T(s) &= \frac{L(s)}{1 + L(s)} \\ &= \frac{\frac{C_r(s) \cdot P_r(s) \cdot e^{-sT}}{1 + C_r(s) \cdot P_r(s) \cdot (1 - e^{-s\hat{T}})}}{1 + \frac{C_r(s) \cdot P_r(s) \cdot e^{-sT}}{1 + C_r(s) \cdot P_r(s) \cdot (1 - e^{-s\hat{T}})}} \\ &= \frac{C_r(s) \cdot P_r(s) \cdot e^{-sT}}{1 + C_r(s) \cdot P_r(s) \cdot (1 - e^{-sT}) + C_r(s) \cdot P_r(s) \cdot e^{-sT}} \\ &= \frac{C_r(s) \cdot P_r(s)}{1 + C_r(s) \cdot P_r(s)} \cdot e^{-sT} \\ &= T_{\text{ref}}(s) \cdot e^{-sT}. \end{aligned} \quad (2.18)$$

Remark.

- This result is very important: we have shown that the delay cannot be completely eliminated and that every transfer function (here $T(s)$ but also $S(s)$) will have the same delay as the plant $P(s)$.
- Advantages of the prediction are:
 - Very fast.
 - Same Robustness.
- Disadvantages of the prediction are:
 - Very difficult to implement.
 - Very difficult to analyze.
 - Problems if there are model errors.

2.3.4 Examples

Example 6. We want to control the plant $P(s) = P_r(s) \cdot e^{-s\tau}$ with a Smith Predictor. The structure of the predictor can be seen in Figure 14.

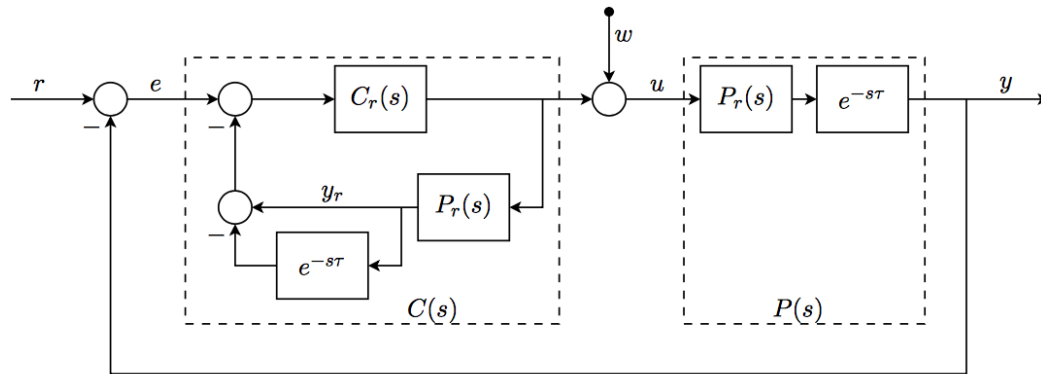


Figure 14: Smith Predictor Structure

- (a) Assume that the model isn't that good and derive the transfer functions of
 - i) The controller.
 - ii) The loop gain.
 - iii) The complementary sensitivity.
- (b) Let's say the model is now perfect. $T_r(s)$ should behave like a *first order low-pass system* with cutoff frequency a . The gain should be chosen s.t. no static error is produced. Find such a $T_r(s)$.
- (c) The plant has been modeled by a friend of you: he had no time for a complete job because of a hockey playoff match and he derived just the state space description of the system. This reads

$$\dot{x}(t) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \cdot x(t) + \begin{pmatrix} 0 \\ \frac{1}{m} \end{pmatrix} \cdot u(t)$$

and

$$\hat{y}(t) = (1 \ 0) \cdot x(t), \quad y(t) = \hat{y}(t - \tau).$$

Find the transfer function of the plant.

- (d) Calculate the transfer function of the controller $C_r(s)$.

Solution.

(a) i) Controller:

With the same derivation seen in the theory, one gets

$$C(s) = \frac{C_r(s)}{1 + C_r(s) \cdot \hat{P}_r(s) \cdot (1 - e^{-s\hat{\tau}})}.$$

ii) Loop Gain:

With the same derivation seen in the theory, one gets

$$L(s) = \frac{C_r(s) \cdot P_r(s) \cdot e^{-s\tau}}{1 + C_r(s) \cdot \hat{P}_r(s) \cdot (1 - e^{-s\hat{\tau}})}.$$

iii) Complementary Sensitivity:

With the same derivation seen in the theory, one gets

$$\begin{aligned} T(s) &= \frac{L(s)}{1 + L(s)} \\ &= \frac{C_r(s) \cdot P_r(s) \cdot e^{-s\tau}}{1 + C_r(s) \cdot \hat{P}_r(s) \cdot (1 - e^{-s\hat{\tau}}) + P_r(s) \cdot C_r(s) \cdot e^{-s\tau}}. \end{aligned}$$

(b) With a perfect model, one gets

$$\begin{aligned} T(s) &= \frac{P_r(s) \cdot C_r(s)}{1 + P_r(s) \cdot C_r(s)} \cdot e^{-s\tau} \\ &= T_r(s) \cdot e^{-s\tau}. \end{aligned}$$

The general form for a lowpass system is

$$\Sigma(s) = \frac{k}{\tau s + 1}.$$

where $\omega = \frac{1}{\tau}$ is the cutoff frequency and k is the gain. If we don't want a static error, the gain should be 1 ($k = 1$). This last step can be shown if one calculates the static error with the well-known formula. With the informations about the cutoff frequency one gets

$$\begin{aligned} T_r(s) &= \frac{1}{\frac{s}{a} + 1} \\ &= \frac{a}{s + a}. \end{aligned}$$

(c) Here one should use Laplace Analysis. The two differential equations reads

$$\begin{aligned} \dot{x}_1(t) &= x_2(t), \\ \dot{x}_2(t) &= \frac{1}{m} \cdot u(t) \end{aligned}$$

with

$$\begin{aligned} \hat{y}(t) &= x_1(t), \\ y(t) &= \hat{y}(t - \tau). \end{aligned}$$

The Laplace transform leads to

$$\begin{aligned} sX_1(s) &= X_2(s) \\ sX_2(s) &= \frac{1}{m} \cdot U(s) \end{aligned}$$

and

$$\hat{Y}(s) = X_1(s), \quad Y(s) = \hat{Y}(s) \cdot e^{-s\tau}$$

together:

$$s^2 X_1(s) = \frac{1}{m} U(s)$$

and so

$$U(s) = ms^2 X_1(s)$$

For the transfer function, it holds

$$\begin{aligned} P(s) &= \frac{Y(s)}{U(s)} \\ &= \frac{X_1(s) \cdot e^{-s\tau}}{X_1(s) \cdot ms^2} \\ &= \frac{1}{ms^2} \cdot e^{-s\tau} \\ &= P_r(s) \cdot e^{-s\tau}. \end{aligned}$$

(d) Since

$$T_r(s) = \frac{P_r(s) \cdot C_r(s)}{1 + P_r(s) \cdot C_r(s)}$$

by solving for $C_r(s)$ one gets

$$\begin{aligned} C_r(s) &= \frac{T_r(s)}{P_r(s) \cdot (1 - T_r(s))} \\ &= \frac{\frac{a}{s+a}}{\frac{1}{ms^2} \cdot \left(1 - \frac{a}{s+a}\right)} \\ &= \frac{ams^2}{s} \\ &= am \cdot s. \end{aligned}$$

2.4 Robustness

2.4.1 Robust Stability

In order to determine the stability of the closed loop gain we have used the Nyquist theorem. That isn't enough: we have to consider that the model for $L(s)$ could be unprecise. In order to ensure stability, it should hold

$$|W_2(j \cdot \omega) \cdot L(j \cdot \omega)| < |1 + L(j \cdot \omega)| \quad \forall \omega \in [0, \infty]. \quad (2.19)$$

This condition can be reformulated, in order to use the complementary sensitivity $T(s)$:

$$\begin{aligned} |W_2(j \cdot \omega) \cdot L(j \cdot \omega)| &< |1 + L(j \cdot \omega)| \quad \forall \omega \in [0, \infty] \\ \frac{|L(j \cdot \omega)|}{|1 + L(j \cdot \omega)|} &< \frac{1}{|W_2(j \cdot \omega)|} \\ |T(j \cdot \omega)| &< \frac{1}{|W_2(j \cdot \omega)|} \\ |T(j \cdot \omega)| \cdot |W_2(j \cdot \omega)| &< 1. \end{aligned} \quad (2.20)$$

This can be interpreted as an upper bound for the complementary sensitivity.

2.4.2 Nominal Performance

The conditions for the sensitivity $S(s)$ can be resumed in the transfer function $W_1(s)$ and so it must hold

$$\begin{aligned} |W_1(j \cdot \omega) \cdot S(j \cdot \omega)| &< 1 \\ \frac{1}{|1 + L(j \cdot \omega)|} \cdot |W_1(j \cdot \omega)| &< 1 \\ |W_1(j \cdot \omega)| &< |1 + L(j \cdot \omega)|. \end{aligned} \quad (2.21)$$

which can be used in with the Nyquist plot: for each frequency ω the return difference should have a distance from the critical point -1 that is larger than $|W_1(j \cdot \omega)|$. Such a system is referred to as satisfying the *nominal performance condition*.

2.4.3 Robust Performance

If a controller $C(s)$ also satisfies

$$|W_1(j \cdot \omega) \cdot S(j \cdot \omega)| + |W_2(j \cdot \omega) \cdot T(j \cdot \omega)| < 1. \quad (2.22)$$

then it satisfies the *robust performance condition*, where, as reminder

- $|W_1(j \cdot \omega)|$: Bound for the sensitivity, from specifications.
- $|W_2(j \cdot \omega)|$: Bound for the complementary sensitivity, from specifications.

Multiplying both sides by $|1 + L(j \cdot \omega)|$ one gets the rearranged condition

$$|W_1(j \cdot \omega)| + |W_2(j \cdot \omega) \cdot L(j \cdot \omega)| < |1 + L(j \cdot \omega)|. \quad (2.23)$$

Remark. The nominal and the robust performance conditions don't guarantee the stability of the system: this should be determined separately, e.g. with the Nyquist theorem. These conditions can be easily checked with Figure 15

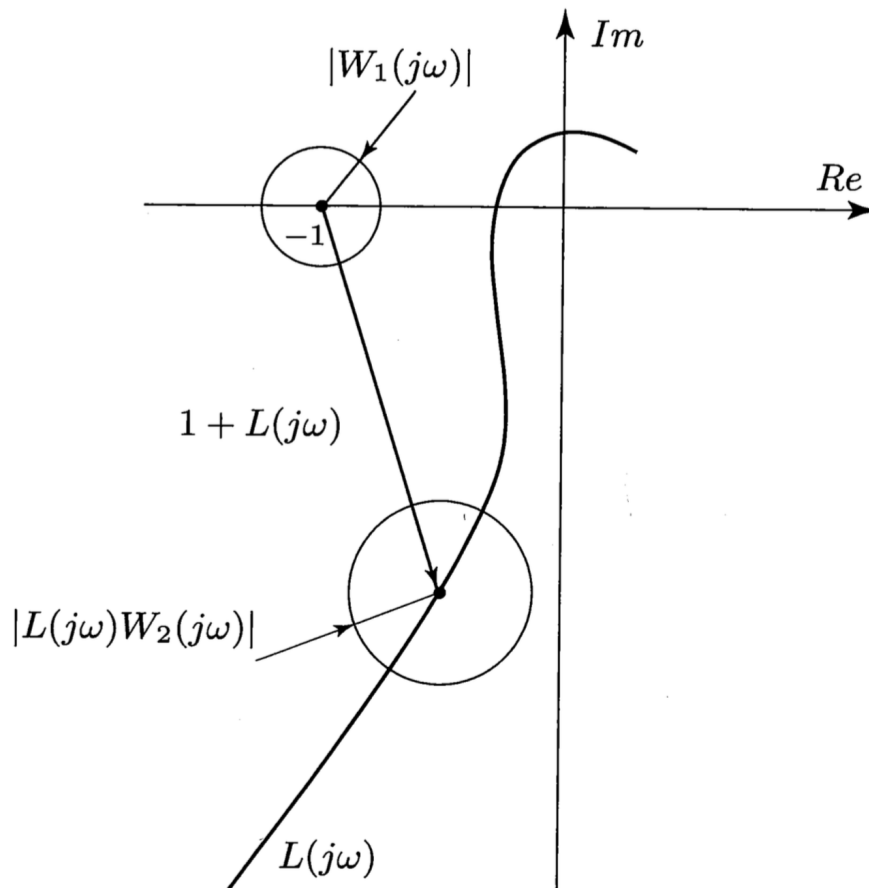


Figure 15: Nyquist with conditions

2.4.4 Examples

Example 7. In order to design a controller for the plant $P(s)$, a bound for the sensitivity of the control system is set:

$$W_1(s) = 60 \cdot \frac{2 \cdot s + 1}{200 \cdot s + 1}.$$

Iteratively, one finds a good controller that results in the complementary sensitivity

$$T(s) = \frac{\omega_0^2}{s^2 + 2 \cdot \delta \cdot \omega_0 \cdot s + \omega_0^2}.$$

where $\omega_0 = 1 \frac{\text{rad}}{\text{s}}$ and $\delta = 1$.

- (a) One can show that the maximal value of the sensitivity $|S(j\omega^*)|$ occurs at $\omega^* = \sqrt{2}$. Compute the value of $|S(j\omega^*)|$.
- (b) Show that the condition for nominal performance is fulfilled in ω^* .
- (c) What is the limit value of $|W_2(j\omega^*)|$ in order to fulfill the condition for robust performance?

Hint: It holds $|L(j\omega^*)| = 0.29$.

Solution.

(a)

It holds

$$\begin{aligned}
 S(s) &= 1 - T(s) \\
 &= 1 - \frac{1}{s^2 + 2 \cdot s + 1} \\
 &= \frac{s^2 + 2 \cdot s}{s^2 + 2 \cdot s + 1} \\
 S(j\omega) &= \frac{-\omega^2 + 2j\omega}{-\omega^2 + 2j\omega + 1} \\
 &= \frac{(-\omega^2 + 2j\omega) \cdot (1 - \omega^2 - 2j\omega)}{(-\omega^2 + 2j\omega + 1) \cdot (1 - \omega^2 - 2j\omega)} \\
 &= \frac{\omega^4 + 3\omega^2 + 2j\omega}{(1 - \omega^2)^2 + 4\omega^2} \\
 |S(j\omega^*)| &= \left| \frac{10 + j2\sqrt{2}}{9} \right| \\
 &= \sqrt{\frac{108}{81}} = 1.16.
 \end{aligned}$$

(b)

The condition reads

$$\begin{aligned}
 |S(j\omega^*)| \cdot |W_1(j\omega^*)| &< 1 \\
 \Rightarrow |S(j\omega^*)| &< \frac{1}{|W_1(j\omega^*)|}.
 \end{aligned}$$

It holds

$$\begin{aligned}
 |W_1(j\omega^*)| &= \left| 60 \cdot \frac{2j\omega^* + 1}{200j\omega^* + 1} \right| \\
 &= 60 \cdot \left| \frac{(2j\omega^* + 1) \cdot (1 - 200j\omega^*)}{(200j\omega^* + 1) \cdot (1 - 200j\omega^*)} \right| \\
 &= 60 \cdot \left| \frac{(1 + 400\omega^{*2} - 198j\omega^*)}{1 + (200 \cdot \omega^*)^2} \right| \\
 &= 60 \cdot \frac{\sqrt{(1 + 400\omega^{*2})^2 + (198\omega^*)^2}}{1 + (200 \cdot \omega^*)^2} \\
 &= 0.64.
 \end{aligned}$$

Since

$$1.16 < \frac{1}{0.64} = 1.56$$

the condition is fulfilled for ω^* .

(c)

The condition reads

$$\begin{aligned}
 |W_1(j \cdot \omega^*)| + |W_2(j \cdot \omega^*) \cdot L(j \cdot \omega^*)| &< |1 + L(j \cdot \omega^*)| \\
 |W_1(j \cdot \omega^*)| + |W_2(j \cdot \omega^*) \cdot L(j \cdot \omega^*)| &< \left| \frac{1}{S(j\omega^*)} \right| \\
 |W_2(j \cdot \omega^*)| &< \frac{1}{|L(j \cdot \omega^*)|} \cdot \left(\left| \frac{1}{S(j\omega^*)} \right| - |W_1(j \cdot \omega^*)| \right) \\
 |W_2(j \cdot \omega^*)| &< \frac{1}{0.29} \cdot \left(\frac{1}{1.16} - 0.64 \right) \\
 |W_2(j \cdot \omega^*)| &< 0.77.
 \end{aligned}$$

3 Implementation of Controllers

3.1 Setpoint Weighting for PID

3.1.1 Recap Goals

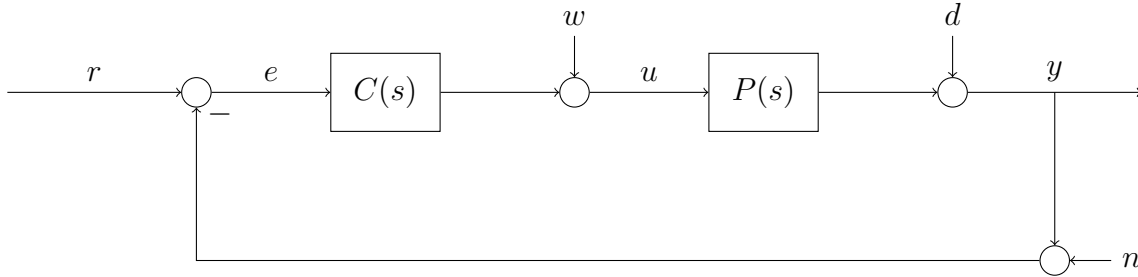


Figure 16: Standard feedback control system's structure

The two big different goals for a control system are **reference tracking** and **disturbance rejection**.

Reference Tracking: Let $y(t)$ track $r(t)$.

Disturbance Rejection: Keep $y(t)$ at $r(t)$.

3.1.2 Method

Since these two goals are often not achievable with a single controller, we define a method to adapt easily the PID controller to different needs, by introducing the **weights** a, b, c .

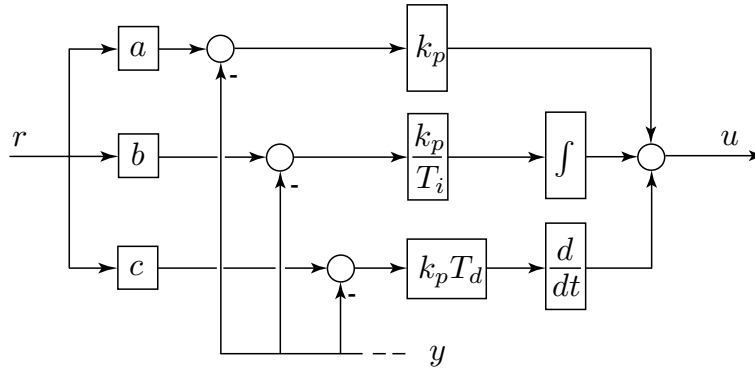


Figure 17: Setpoint weighting PID structure.

By *tuning* the weights a, b, c one can achieve the desired control action. One can write, mathematically:

$$U(s) = k_p \cdot (a \cdot R(s) - Y(s)) + \frac{k_p}{T_i \cdot s} \cdot (b \cdot R(s) - Y(s)) + k_p \cdot T_d \cdot s \cdot (c \cdot R(s) - Y(s)). \quad (3.1)$$

Remark. From this equation, one can show that the choice of a, b, c has no influence on the loop gain, that is, no influence on the stability of the system.

3.1.3 Application to Reference Tracking

Although the PID controller was designed by thinking of some disturbance rejection problem, this method could also be used to ensure reference tracking. If **disturbances** are small compared to **reference changes** and the actuator is a **limiting factor** (see saturation, next chapters), usually one does the following choice:

- **P**: The parameter a is chosen such that: $0 < a < 1$.
 - We can control the *aggressivity* of the controller.
- **I**: The parameter b is chosen such that: $b = 1$.
 - with $r = y$ one has that $e_\infty \rightarrow 0$.
- **D**: The parameter c is chosen such that: $c = 0$.
 - If there are disturbances, it is good not to have their influences on derivative terms.

3.1.4 Examples

Example 8. A plant $P(s)$ should be controlled with a PID controller. One should consider the setpoint weighting in order to better design the controller. The plant reads

$$P(s) = \frac{10}{(s+1)(s+2)(s+3)}.$$

- (a) Find
 - i) The critical time constant T^* ,
 - ii) The critical gain k_p^* ,
 - iii) The static gain of the plant.
- (b) Derive the transfer function between $r \rightarrow y$.

Solution.

(a)

With the rules learned in Control Systems I, it holds:

$$k_p^* \cdot P(j\omega^*) = -1 + 0 \cdot j$$

$$T^* = \frac{2\pi}{\omega^*}.$$

With the defined plant:

$$k_p^* \cdot P(j\omega^*) = \frac{10 \cdot k_p^*}{(j\omega^* + 1) \cdot (j\omega^* + 2) \cdot (j\omega^* + 3)}$$

$$= \frac{10 \cdot k_p^*}{-6\omega^{*2} + 6 + j \cdot (-\omega^{*3} + 11\omega^*)}.$$

The imaginary part should be 0. It holds

$$-\omega^{*3} + 11\omega^* = 0 \Rightarrow \omega^* = \sqrt{11}.$$

The real part should be -1. It holds

$$\frac{10 \cdot k_p^*}{-6\omega^{*2} + 6} = -1 \Rightarrow k_p^* = 6.$$

The static gain is

$$P(0) = \frac{10}{6}.$$

(b)

It holds

$$Y(s) = P(s) \cdot U(s)$$

$$= P(s) \cdot \left((aR(s) - Y(s)) \cdot k_p + (bR(s) - Y(s)) \cdot \frac{k_p}{T_i} \cdot \frac{1}{s} + (cR(s) - Y(s)) \cdot k_p \cdot T_d \cdot s \right).$$

One can write

$$R(s) \cdot P(s) \cdot k_p \cdot \left(a + \frac{b}{T_i \cdot s} + cT_d \right) = Y(s) \cdot \left(1 + P(s) \cdot k_p \cdot \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right) \right)$$

From the definition:

$$T(s) = \frac{Y(s)}{R(s)}$$

$$= \frac{P(s) \cdot k_p \cdot \left(a + \frac{b}{T_i \cdot s} + cT_d \right)}{1 + P(s) \cdot k_p \cdot \left(1 + \frac{1}{T_i \cdot s} + k_p \cdot T_d \cdot s \right)}$$

$$= \frac{10k_p \cdot (aT_i \cdot s + b + c \cdot T_i \cdot T_d \cdot s^2)}{T_i \cdot (s + 1) \cdot (s + 2) \cdot (s + 3) + 10k_p(T_i \cdot s + 1 + T_i \cdot T_d \cdot s^2)}.$$

Remark. From this equation one can see that the stability isn't affected from the choice of a, b, c .

3.2 Feed Forward

The concept of different controllers for different goals is generalized in the feed forward structure.

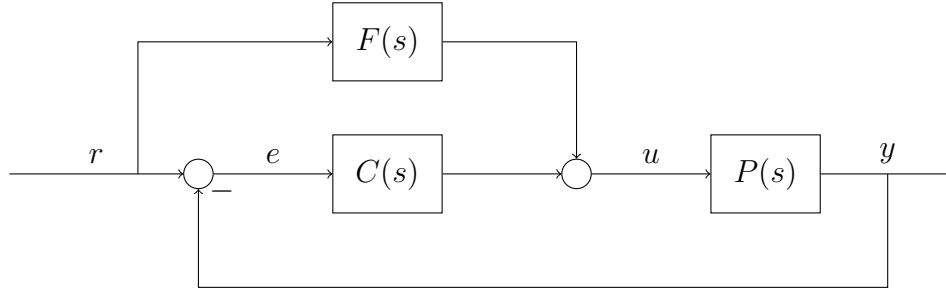


Figure 18: Feed Forward control system's structure.

Such a structure increases speed of the system because r has direct influence on u . This is really good if one wants to improve the reference tracking.

Mathematically, it holds

$$\begin{aligned}
 Y(s) &= D(s) + P(s) \cdot F(s) \cdot R(s) + P(s) \cdot C(s) \cdot R(s) - P(s) \cdot C(s) \cdot (N(s) + Y(s)) \\
 Y(s) \cdot \underbrace{(1 + P(s) \cdot C(s))}_{S(s)^{-1}} &= D(s) + (P(s) \cdot F(s) + P(s) \cdot C(s)) \cdot R(s) - P(s) \cdot C(s) \cdot N(s) \\
 \Rightarrow Y(s) &= \left(P(s) \cdot F(s) \cdot S(s) + \underbrace{P(s) \cdot C(s) \cdot S(s)}_{T(s)} \right) \cdot R(s) - \underbrace{P(s) \cdot C(s) \cdot S(s)}_{T(s)} \cdot N(s) \\
 \Rightarrow Y(s) &= (P(s) \cdot S(s) \cdot F(s) + T(s)) \cdot R(s) - T(s) \cdot N(s) + S(s) \cdot D(s).
 \end{aligned} \tag{3.2}$$

Remark.

- The closed loop stability is not affected by F .
- The tracking performance increases. Speed up the output response to input changes.

3.3 Saturation Problem

The plants treated so far, were linear systems: these of course represent just a small part of the existing real systems, where **nonlinearities** occur. An important intuitive nonlinear effect is the limitation of some actuators at the plant's input.

In general, *physical signals* cannot have all desired values because of natural limitations: easy examples are, the maximal opening area of a valve ($0 \leq A \leq A_{max}$) and the maximal torque of a motor ($|T| \leq |T_{max}|$). In order to take into account these limitations, one has to include in the loop a *saturation element*.

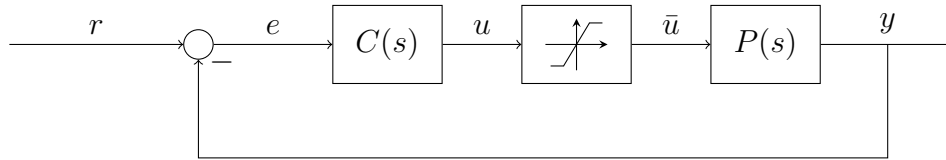


Figure 19: Actuator Saturation.

This element defines the input to give to the plant as

$$\bar{u}(t) = \begin{cases} u_{max}, & \text{if } u(t) > u_{max} \\ u(t), & \text{if } u_{max} > u(t) > u_{min} \\ u_{min}, & \text{if } u(t) < u_{min}. \end{cases} \quad (3.3)$$

A critical situation occurs when the integrative controller's output is at its saturation state: with a further integration (that is needed because on a non-zero error) one cannot decrease the error by more increasing/decreasing of $u(t)$ (that is limited).

To understand this, let's take a step back: the integral part of a PID controller makes the controller generate an output until the reference value is reached. To do this, the integral part *accumulates* in the memory the error remaining after each cycle. It is clear, that if this error won't be zero because of the saturation, the accumulation becomes a problem.

One can solve the problem with the help of a **Anti-Reset Windups (ARW)** element. The idea behind such a structure is to stop accumulating the error and *empty* the integrator.

Since the error is accumulating and we want

$$\int_0^t e(\tau) d\tau \approx 0. \quad (3.4)$$

we need for a certain big amount of time a negative error: we call this method **integral windup/reset windup**. This causes oscillations that are not welcome (see Figure 22). With the **ARW** one can empty the integrator faster and the behaviour of the system is better.

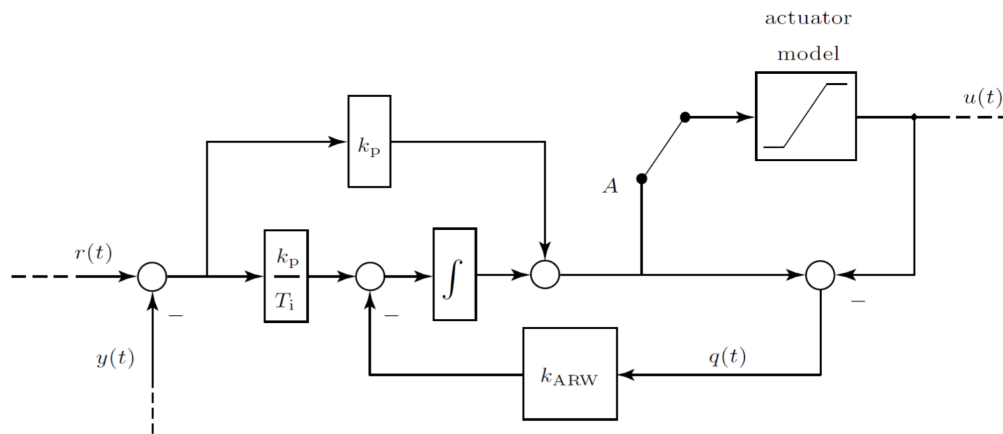


Figure 20: Anti-Reset Windups' structure.

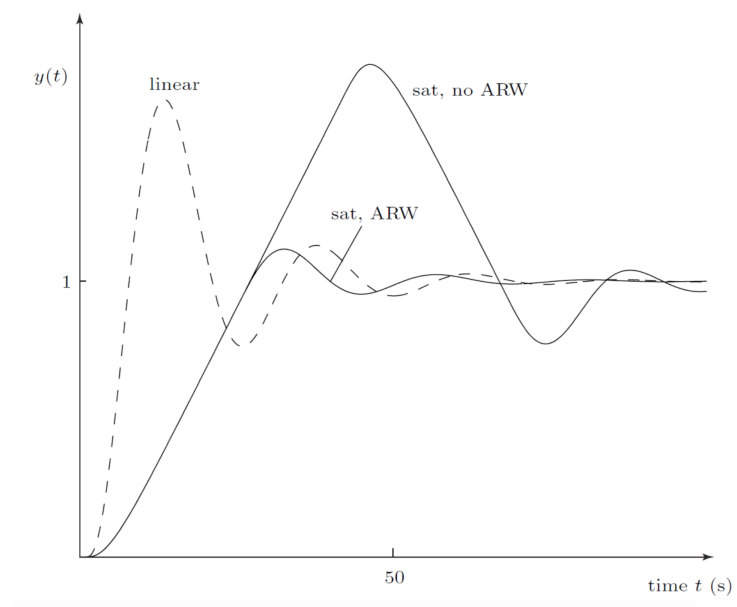


Figure 21: Effects of the Anti-Reset Windups.

3.3.1 Examples

Example 9. The plant

$$P(s) = e^{-s}.$$

should be controlled with the PI controller

$$C(s) = 0.2 + \frac{1}{2s}.$$

An input limitation/saturation is given as

$$u_b(t) \begin{cases} 0.1, & \text{if } u(t) \geq 0.1 \\ u(t), & \text{if } |u(t)| < 0.1 \\ -0.1, & \text{if } u(t) \leq -0.1. \end{cases}$$

- Draw the signals $y(t)$, $u(t)$, $u_b(t)$ in the given diagram.
- Which problem can be noticed in this problem? How could one solve this problem?

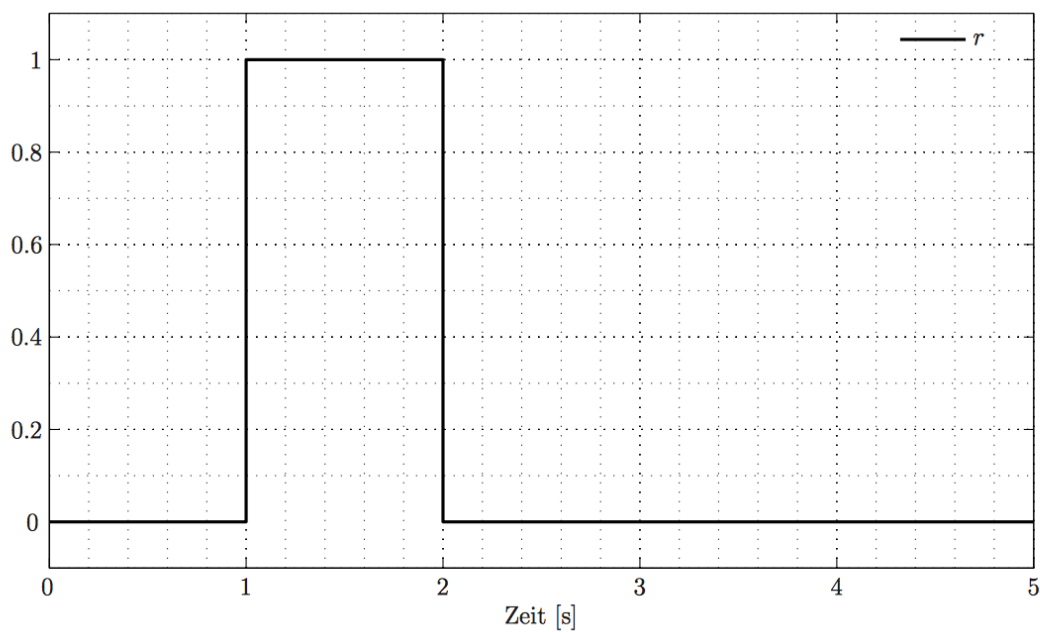


Figure 22: Diagram for the drawing. The signal $r(t)$ is given.

Solution.

(a)

First of all, we refer to a classical feedback loop with saturation (see Figure 23). For the rest of the calculations, please refer to Figure 24

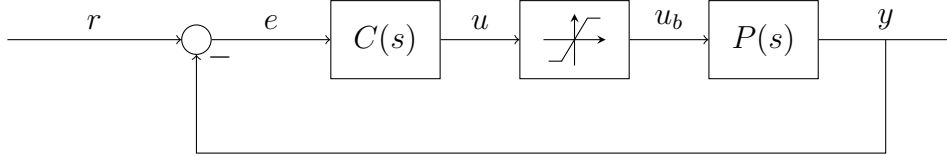


Figure 23: Actuator Saturation.

First of all, let's define how one can get $u(t)$. From the very definition:

$$U(s) = C(s) \cdot E(s).$$

This holds in frequency domain. If we want to draw a diagram in time domain, we have to compute the inverse Laplace transform of $U(s)$:

$$\begin{aligned} u(t) &= \mathcal{L}^{-1}(U(s)) = \mathcal{L}^{-1}(C(s) \cdot E(s)) \\ &= \mathcal{L}^{-1}(0.2 \cdot E(s) + \frac{1}{2} \cdot \frac{1}{s} \cdot E(s)). \end{aligned}$$

With the known Laplace inversions it follows

$$u(t) = \mathcal{L}^{-1}(0.2 \cdot E(s) + \frac{1}{2} \cdot \frac{1}{s} \cdot E(s)) = 0.2 \cdot e(t) + \frac{1}{2} \int_0^t e(\tau) d\tau.$$

Now, let's analyze and draw the signals: we do this in intervals, in order to track every signals' change.

- $t \in [0, 1]$

The reference signal r is zero and so all the other signals.

- $t \in [1, 2]$

Since the plant is a delay element of 1 second (e^{-s}), the output $y(t)$ would be still 0. This means that in this interval, the error is $e(t) = r(t) - y(t) = 1$. We use this information to get the behaviour of $u(t)$:

$$\begin{aligned} u(t) &= 0.2 \cdot e(t) + \frac{1}{2} \int_0^t e(\tau) d\tau \\ &= 0.2 \cdot 1 + \frac{1}{2} \cdot \int_1^2 1 d\tau \\ &= 0.2 + 0.5 \cdot (t - 1). \end{aligned}$$

In order to draw this, we evaluate $u(t)$ at $t = 1$ and $t = 2$. It holds

$$\begin{aligned} u(1) &= 0.2, \\ u(2) &= 0.7. \end{aligned}$$

So, $u(t)$ is an increasing linear function from 0.2 to 0.7.

Since $u_b(t)$ is a function of $u(t)$, we can now find its shape: since $u(t) > 0.1$ in the whole interval, we use the definition and say $u_b(t) = 0.1$ (constant value).

- $t \in [2, 3)$

There should be a step in $y(t)$ here: this can be seen in

$$y(t) = u_b(t - 1) = 0.1.$$

This means that the error $e(t) = r(t) - y(t) = -0.1$. We use this information to get the behaviour of $u(t)$:

$$\begin{aligned} u(t) &= 0.2 \cdot e(t) + \frac{1}{2} \int_0^t e(\tau) d\tau \\ &= 0.2 \cdot (-0.1) + \frac{1}{2} \cdot \left(\int_1^2 1 d\tau + \int_2^t (-0.1) d\tau \right) \\ &= -0.02 + 0.5 + 0.5 \cdot (-0.1) \cdot (t - 2) \\ &= 0.48 - 0.05 \cdot (t - 2). \end{aligned}$$

In order to draw this, we evaluate $u(t)$ at $t = 2$ and $t = 3$. It holds

$$\begin{aligned} u(2) &= 0.48, \\ u(3) &= 0.43. \end{aligned}$$

So, $u(t)$ is a decreasing linear function from 0.48 to 0.43.

Since $u_b(t)$ is a function of $u(t)$, we can now find its shape: since $u(t) > 0.1$ in the whole interval, we use the definition and say $u_b(t) = 0.1$ (constant value).

- $t \in [3, 5)$

The error remains $e(t) = r(t) - y(t) = -0.1$. since $y(t) = u_b(t - 1) = 0.1$. We use this information to get the behaviour of $u(t)$:

$$\begin{aligned} u(t) &= 0.2 \cdot e(t) + \frac{1}{2} \int_0^t e(\tau) d\tau \\ &= 0.2 \cdot (-0.1) + \frac{1}{2} \cdot \left(\int_1^2 1 d\tau + \int_2^t (-0.1) d\tau \right) \\ &= -0.02 + 0.5 + 0.5 \cdot (-0.1) \cdot (t - 2) \\ &= 0.48 - 0.05 \cdot (t - 2). \end{aligned}$$

which reads exactly the same as before. In order to draw this, we evaluate $u(t)$ at $t = 2$ and $t = 3$. It holds

$$\begin{aligned} u(3) &= 0.43 \\ u(5) &= 0.33. \end{aligned}$$

So, $u(t)$ is the same decreasing linear function as before.

Since $u_b(t)$ is a function of $u(t)$, we can now find its shape: since $u(t) > 0.1$ in the whole interval, we use the definition and say $u_b(t) = 0.1$ (constant value).

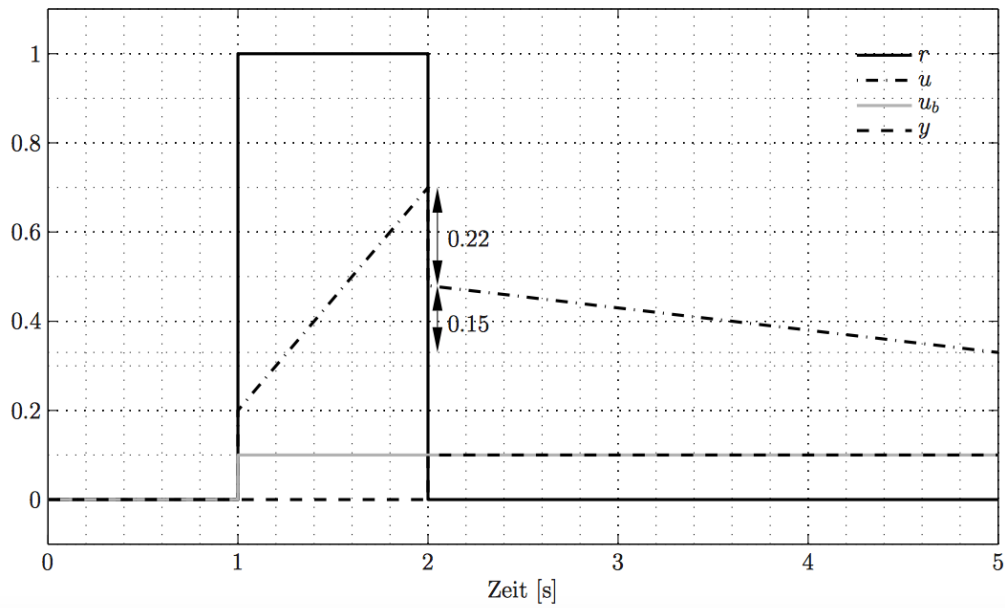


Figure 24: Solution.

(b)

This example, shows the problem that is introduced by saturation. The from the controller asked input $u(t)$ is bigger than the real feasible input $u_b(t)$. This creates the windup situation and we can use an ARW structure to avoid this.

3.4 Digital Control

3.4.1 Signals and Systems

A whole course is dedicated to this topic (see Signals and Systems of professor D'Andrea). A *signal* is a function of time that represents a physical quantity.

Continuous-Time signals are described by a function $x(t)$ such that this takes continuous values.

Discrete-Time Signals are described by a function $x[n] = x(n \cdot T_s)$ where T_s is the **sampling time**. The **sampling frequency** is defined as $f_s = \frac{1}{T_s}$.

One can see the difference between the two descriptions in Figure 25.

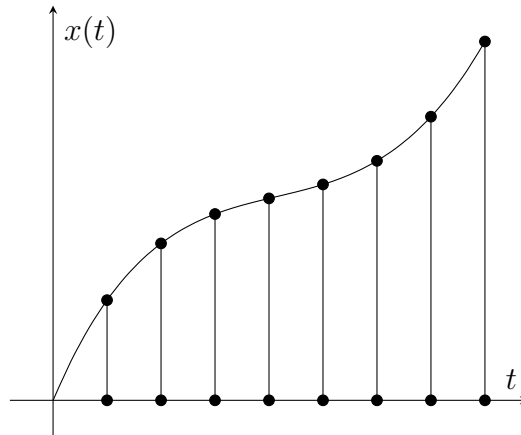


Figure 25: Continuous-Time versus Discrete-Time representation

Advantages of Discrete-Time analysis

- Calculations are easier. Moreover, integrals become sums and differentiations become finite differences.
- One can implement complex algorithms.

Disadvantages of Discrete-Time analysis

- The sampling introduces a delay in the signal ($\approx e^{-\frac{sT_s}{2}}$)
- The informations between two samplings, that is between $x[n]$ and $x[n+1]$ is lost.

Every real-world signal is some sort of discrete-time signal.

3.4.2 Discrete-Time State-Space Description

If in continuous-time we have to deal with differential equations, in discrete-time we have to work with **difference equations**: the discrete-time state-space description is very similar to the continuous-time one:

$$\begin{aligned} x[k+1] &= A \cdot x[k] + b \cdot u[k] \\ y[k] &= c \cdot x[k] + d \cdot u[k]. \end{aligned} \tag{3.5}$$

Lyapunov Stability

A system is **asymptotically stable** if and only if

$$|\lambda_i| < 1 \quad \forall i. \quad (3.6)$$

where λ_i are the eigenvalues of A . This, in other words, means that the eigenvalues should remain inside the unity circle.

3.4.3 Discrete-Time Control Systems

Nowadays, controls systems are implemented in microcontroller or in microprocessors in discrete-time and really rarely (see the lecture *Elektrotechnik II*) in continuous-time. Like defined, although the processes are faster and easier, the informations are still sampled and there is a certain loss of data.

Aliasing

If the sampling frequency is choosed too low, i.e. one measures less times pro second, the signal can become poorly determined and the loss of information is too big to reconstruct it uniquely. This situation is called **aliasing** and one can find many examples of that in the real world. Let's take an easy example: you are finished with the summer's session and you are flying to Ibiza, to finally enjoy the sun after a summer at ETH. You decide to film the turbine of the plane because, although you are on holiday, you have an engineer's spirit. You arrive to Ibiza and as you get into your hotel room, you want you have a look at your film. The rotation you observe looks different to what it is supposed to be, and since you haven't drunk yet, there must be some scientific reason. In fact, the sampling frequency of your phone camera is much lower than the turning frequency of the turbine: this results in a loss of information and hence in a wrong perception of what is going on.

Let's assume some signal

$$x_1(t) = \cos(\omega \cdot t). \quad (3.7)$$

After discretization, the sampled signal reads

$$x_1[n] = \cos(\omega \cdot T_s \cdot n) = \cos(\Omega \cdot n), \quad \Omega = \omega \cdot T_s. \quad (3.8)$$

Let's assume a second signal

$$x_2(t) = \cos\left(\left(\omega + \frac{2\pi}{T_s}\right) \cdot t\right), \quad (3.9)$$

where the frequency

$$\omega_2 = \omega + \frac{2\pi}{T_s}. \quad (3.10)$$

is given. The discretization of this second signal reads

$$\begin{aligned} x_2[n] &= \cos\left(\left(\omega + \frac{2\pi}{T_s}\right) \cdot T_s \cdot n\right) \\ &= \cos(\omega \cdot T_s \cdot n + 2\pi \cdot n) \\ &= \cos(\omega \cdot T_s \cdot n) \\ &= x_1[n]. \end{aligned} \quad (3.11)$$

Although the two signals have different frequencies, they are equal when discretized. For this reason, one has to define an interval of *good* frequencies, where aliasing doesn't occur. In particular it holds

$$|\omega| < \frac{\pi}{T_s} \quad (3.12)$$

or

$$f < \frac{1}{2 \cdot T_s} \Leftrightarrow f_s > 2 \cdot f_{max}. \quad (3.13)$$

The maximal frequency accepted is $f = \frac{1}{2 \cdot T_s}$ and is named **Nyquist Frequency**. In order to ensure good results, one uses in practice a factor of 10

$$f < \frac{1}{10 \cdot T_s} \Leftrightarrow f_s > 10 \cdot f_{max}. \quad (3.14)$$

For control systems the crossover frequency should follow

$$f_s \geq 10 \cdot \frac{\omega_c}{2\pi}. \quad (3.15)$$

Anti Aliasing Filter (AAF)

The Anti Aliasing Filter is an **analog** filter and not a discrete one. In fact, we want to eliminate unwanted frequencies before sampling, because after that is *too late* (see Figure 26). But how can one define unwanted frequencies? Those frequencies are normally the higher frequencies of a signal⁶. Because of that, as AAF one uses normally a **low-pass filter**. This type of filter, *lets* low frequencies *pass* and blocks higher ones⁷. The mathematic formulation of a first-order low-pass filter is given by

$$lp(s) = \frac{k}{\tau \cdot s + 1}. \quad (3.16)$$

where k is the gain and τ is the time constant of the system. The drawback of such a filter is problematic: the filter introduces additional unwanted phase that can lead to unstable behaviours.

Analog to Digital Converter (ADC)

At each discrete time step $t = k \cdot T$ the **ADC** converts a voltage $e(t)$ to a digital number following a sampling frequency.

Microcontroller (μP)

This is a discrete-time controller that uses the sampled discrete-signal and gives back a discrete output.

Digital to Analog Converter (DAC)

In order to convert back the signal, the **DAC** applies a **zero-order-hold (ZOH)**. This introduces an extra delay of $\frac{T}{2}$ (see Figure 27).

⁶Keep in mind: high signal frequency means problems by lower sampling frequency!

⁷This topic is exhaustively treated in the course Signals and Systems

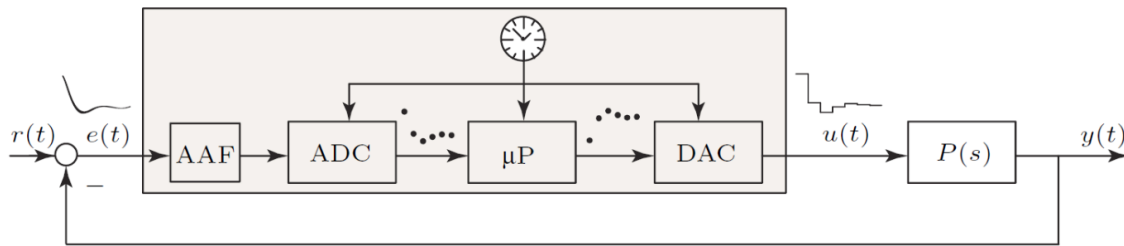


Figure 26: Control Loop with AAF.

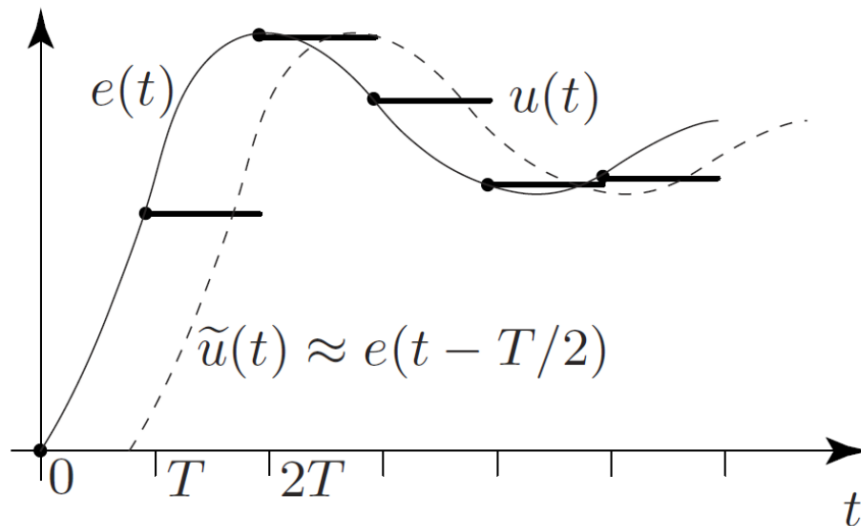


Figure 27: Zero-Order-Hold.

3.4.4 Controller Discretization/Emulation

Discrete-Time controllers can be defined by using the **z-Transform**: this transform method is to understand as some kind of Laplace transform (see the course *Analysis III*).

In general, it holds

$$y(t + T_s) = y[n + 1] \quad \Leftrightarrow \quad z \cdot Y(z). \quad (3.17)$$

$$(3.18)$$

The meaning of the variable z can change with respect to the chosen discretization approach. Here, just discretization results are presented: if you are interested in the theory behind difference rules and discretization, have a look at courses like *Computational Methods for Eng. Applications I/II*. A list of the most used transformations is:

Exakt	$s = \frac{1}{T_s} \cdot \ln(z)$	$z = e^{s \cdot T_s}$
Euler forward	$s = \frac{z - 1}{T_s}$	$z = s \cdot T_s + 1$
Euler backward	$s = \frac{z - 1}{z \cdot T_s}$	$z = \frac{1}{1 - s \cdot T_s}$
Tustin	$s = \frac{2}{T_s} \cdot \frac{z - 1}{z + 1}$	$z = \frac{1 + s \cdot \frac{T_s}{2}}{1 - s \cdot \frac{T_s}{2}}$

The different approaches are results of different Taylor's approximations⁸:

- Forward Euler:

$$z = e^{s \cdot T_s} \approx 1 + s \cdot T_s. \quad (3.19)$$

- Backward Euler:

$$z = e^{s \cdot T_s} = \frac{1}{e^{-s \cdot T_s}} \approx \frac{1}{1 - s \cdot T_s}. \quad (3.20)$$

- Tustin:

$$z = \frac{e^{s \cdot \frac{T_s}{2}}}{e^{-s \cdot \frac{T_s}{2}}} \approx \frac{1 + s \cdot \frac{T_s}{2}}{1 - s \cdot \frac{T_s}{2}}. \quad (3.21)$$

In practice, the most used approach is the Tustin transformation, but there are cases where the other transformations could be useful.

Stability

It is nice to investigate, how and if the stability conditions change in discretized systems. As a reminder, continuous-time systems are stable if their poles have a negative real part. Discrete-time systems are stable if their poles are bounded in the unity circle (see Figure 28). With Euler Backward and Tustin the stability conditions remain the same; with Forward Euler it is however possible, that the systems becomes unstable (see Figure 28).

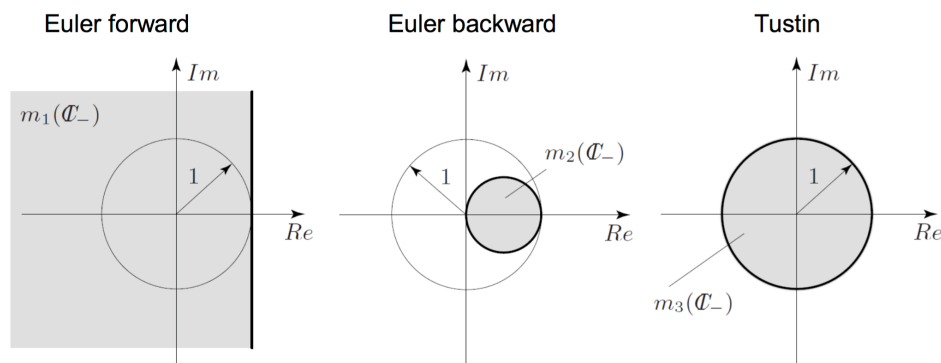


Figure 28: Pole mapping.

⁸As reminder: $e^x \approx 1 + x$.

3.4.5 Discrete Time Controller Synthesis

As you learned in class, there are two ways to discretize systems: see Figure 29.

In the previous chapter, we learned how to emulate a system: in this chapter we learn the discrete time controller synthesis (direct synthesis).

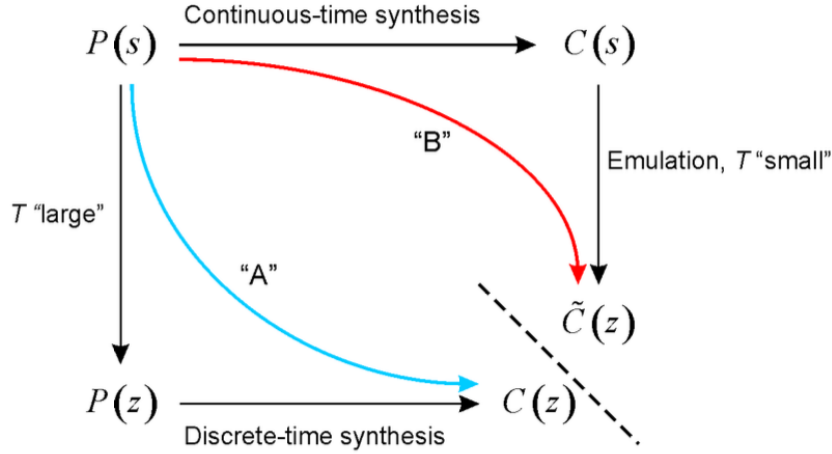


Figure 29: Emulation and Discrete time controller synthesis.

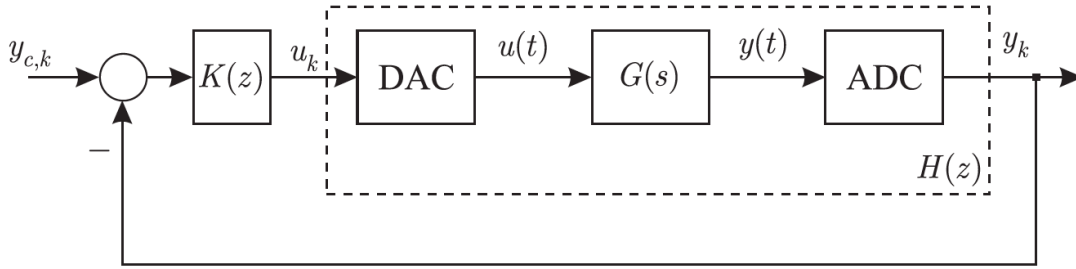


Figure 30: Discrete-time control loop.

The general situation is the following: a control loop is given as in Figure 30.

The continuous time transfer function $G(s)$ is given.

We want to compute the equivalent discrete-time transfer function $H(z)$. The loop is composed from a Digital-to-Analog Converter (DAC), the continuous-time transfer function $G(s)$ and an Analog to Digital Converter (ADC).

We want to reach a structure that looks like Figure 31.

The first thing to do, is to consider an input to analyze. What is usually chosen, is a **unit-step**

$$u(kT) = \{\dots, 0, 1, 1, \dots\}. \quad (3.22)$$

Since the z -Transform is defined as

$$X(z) = \sum_{n=0}^{\infty} x(n) \cdot z^{-n} \quad (3.23)$$

we get for u

$$U(z) = 1 + z^{-1} + z^{-2} + z^{-3} + \dots + z^{-n}. \quad (3.24)$$

This sum can be written as (see geometric series)

$$U(z) = \frac{1}{1 - z^{-1}}. \quad (3.25)$$

For $U(z)$ to be defined, this sum must converge. This can be verified by exploring the properties of the geometric series.

Recall: sum of geometric series

Let S_n denote the sum over the first n elements of a geometric series:

$$\begin{aligned} S_n &= U_0 + U_0 \cdot a + U_0 \cdot a^2 + \dots + U_0 \cdot a^{n-1} \\ &= U_0 \cdot (1 + a + a^2 + \dots + a^{n-1}). \end{aligned}$$

Then

$$a \cdot S_n = U_0 \cdot (a + a^2 + a^3 + \dots + a^n)$$

and

$$S_n - a \cdot S_n = U_0 \cdot (1 - a^n),$$

which leads to

$$S_n = U_0 \cdot \frac{1 - a^n}{1 - a}.$$

From here, it can be seen that the limit for n going to infinity converges if and only if the absolute value of a is smaller than one:

$$\lim_{n \rightarrow \infty} S_n = U_0 \cdot \frac{1}{1 - a}, \text{ iff } |a| < 1. \quad (3.26)$$

Therefore the limiting case $|a| = 1 =: r$ is called **radius of convergence**. The according **convergence criterion** is $|a| < r$.

$H(z)$ contains the converters: at first, we have the *digital -to-analog* converter. The *Laplace-Transform* of the unit-step reads generally

$$\frac{1}{s}. \quad (3.27)$$

Hence, the transfer function before the *analog-to-digital* converter reads

$$\frac{G(s)}{s}. \quad (3.28)$$

In order to consider the *analog-to-digital* converter, we have to apply the inverse Laplace transform to get

$$y(t) = \mathcal{L}^{-1} \left(\frac{G(s)}{s} \right). \quad (3.29)$$

Through a z -transform one can now get $Y(z)$. It holds

$$\begin{aligned} Y(z) &= \mathcal{Z}(y(kT)) \\ &= \mathcal{Z} \left(\mathcal{L}^{-1} \left(\frac{G(s)}{s} \right) \right). \end{aligned} \quad (3.30)$$

The transfer function is then given as

$$H(z) = \frac{Y(z)}{U(z)}. \quad (3.31)$$

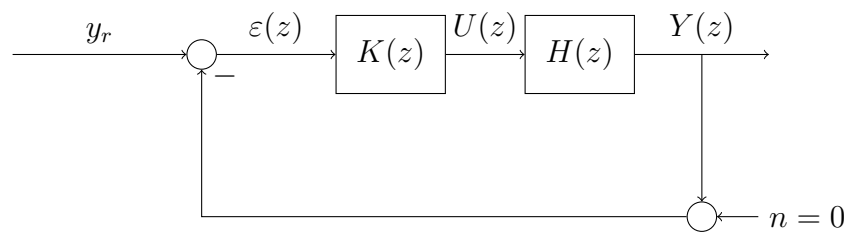


Figure 31: Discrete Syhtesis.

3.4.6 Examples

Example 10. Show that the stability conditions of a system remain the same if stable continuous-time poles are mapped to discrete-time poles.

Solution. If one looks at Tustin it holds

$$\begin{aligned} z &= \frac{1 + s \cdot \frac{T_s}{2}}{1 - s \cdot \frac{T_s}{2}} \\ &= \frac{1 + (x + i \cdot y) \cdot \frac{T_s}{2}}{1 - (x + i \cdot y) \cdot \frac{T_s}{2}} \\ &= \frac{1 + x \cdot \frac{T_s}{2} + i \cdot y \cdot \frac{T_s}{2}}{1 - x \cdot \frac{T_s}{2} - i \cdot y \cdot \frac{T_s}{2}}. \end{aligned}$$

If we compute the modulus, we get

$$|z| = \sqrt{\frac{(1 + x \cdot \frac{T_s}{2})^2 + y^2 \cdot (\frac{T_s}{2})^2}{(1 - x \cdot \frac{T_s}{2})^2 + y^2 \cdot (\frac{T_s}{2})^2}}.$$

Since the continuous-time system should be stable, it should hold

$$x < 0$$

This means

$$(1 + x \cdot \frac{T_s}{2})^2 + y^2 \cdot (\frac{T_s}{2})^2 < (1 - x \cdot \frac{T_s}{2})^2 + y^2 \cdot (\frac{T_s}{2})^2,$$

and so

$$|z| < 1.$$

The discrete-time is therefore stable.

Example 11. One of your colleagues has developed the following continuous-time controller for a continuous-time plant

$$C(s) = \frac{2s + 1}{s + \alpha}$$

where $\alpha \in \mathbb{R}$ is a tuning factor.

- (a) You now want to implement the controller on a microprocessor using the **Euler backward** emulation approach. What is the resulting function $C(z)$ for a generic sampling time $T \in \mathbb{R}^+$?
- (b) What is the range of the tuning factor α that produces an asymptotically stable discrete-time controller when applying the Euler backward emulation approach?
- (c) What is the condition on α to obtain an asymptotically stable continuous-time controller $C(s)$ and an asymptotically stable discrete-time controller $C(z)$ using the Euler backward emulation approach?

Solution.

- (a) The Euler backward emulation approach reads

$$s = \frac{z-1}{T \cdot z}.$$

If one substitutes this into the transfer function of the continuous-time controller, one gets

$$\begin{aligned} C(z) &= \frac{2 \cdot \frac{z-1}{T \cdot z} + 1}{\frac{z-1}{T \cdot z} + \alpha} \\ &= \frac{z \cdot (2 + T) - 2}{z \cdot (1 + \alpha \cdot T) - 1}. \end{aligned}$$

- (b) The controller $C(z)$ is asymptotically stable if its pole π_d fulfills the condition

$$|\pi_d| < 1.$$

The pole reads

$$\begin{aligned} z \cdot (1 + \alpha \cdot T) - 1 &= 0 \\ \pi_d &= \frac{1}{1 + \alpha \cdot T}. \end{aligned}$$

This, together with the condition for stability gives

$$-1 < \frac{1}{1 + \alpha \cdot T} < 1 \Rightarrow \alpha > 0 \text{ or } \alpha < -\frac{2}{T}.$$

- (c) For $C(s)$ to be asymptotically stable, its pole $\pi_c = -\alpha$ must lie in the left half of the complex plane:

$$\operatorname{Re}\{\pi_c\} < 0 \Rightarrow \alpha > 0.$$

Together with the results from (b), the condition on α is $\alpha > 0$.

Example 12.

(a) Choose all the signals that can be sampled without aliasing. The sampling time is $T_s = 1s$.

- ☐ $x(t) = \cos(4\pi \cdot t)$.
- ☐ $x(t) = \cos(4\pi \cdot t + \pi)$.
- ☐ $x(t) = 2 \cdot \cos(4\pi \cdot t + \pi)$.
- ☐ $x(t) = \cos(0.2\pi \cdot t)$.
- ☐ $x(t) = \cos(0.2\pi \cdot t + \pi)$.
- ☐ $x(t) = 3 \cdot \cos(0.2\pi \cdot t + \pi)$.
- ☐ $x(t) = \cos(\pi \cdot t)$.
- ☐ $x(t) = \cos(\pi \cdot t + \pi)$.
- ☐ $x(t) = 2 \cdot \cos(\pi \cdot t + \pi)$.
- ☐ $x(t) = \cos(0.2\pi \cdot t) + \cos(4\pi \cdot t)$.
- ☐ $x(t) = \sin(0.2\pi \cdot t) + \sin(0.4\pi \cdot t)$.
- ☐ $x(t) = \sum_{i=1}^{100} \cos\left(\frac{2\pi}{i+1} \cdot t\right)$.
- ☐ $x(t) = \sum_{i=1}^{100} \cos\left(\frac{2\pi}{i+2} \cdot t\right)$.

(b) The signal

$$x(t) = 2 \cdot \cos(20 \cdot \pi \cdot t + \pi) + \cos(40 \cdot \pi \cdot t) + \cos(30 \cdot \pi \cdot t).$$

is sampled with sampling frequency f_s . What is the minimal f_s such that no aliasing occurs?

Solution.

- (a) ☐ $x(t) = \cos(4\pi \cdot t)$.
☐ $x(t) = \cos(4\pi \cdot t + \pi)$.
☐ $x(t) = 2 \cdot \cos(4\pi \cdot t + \pi)$.
☒ $x(t) = \cos(0.2\pi \cdot t)$.
☒ $x(t) = \cos(0.2\pi \cdot t + \pi)$.
☒ $x(t) = 3 \cdot \cos(0.2\pi \cdot t + \pi)$.
☐ $x(t) = \cos(\pi \cdot t)$.
☐ $x(t) = \cos(\pi \cdot t + \pi)$.
☐ $x(t) = 2 \cdot \cos(\pi \cdot t + \pi)$.
☐ $x(t) = \cos(0.2\pi \cdot t) + \cos(4\pi \cdot t)$.
☒ $x(t) = \sin(0.2\pi \cdot t) + \sin(0.4\pi \cdot t)$.
☐ $x(t) = \sum_{i=1}^{100} \cos\left(\frac{2\pi}{i+1} \cdot t\right)$.
☒ $x(t) = \sum_{i=1}^{100} \cos\left(\frac{2\pi}{i+2} \cdot t\right)$.

Explanation:

If one goes back to the definition of the ranges to ensure no aliasing occurs, one gets the formula

$$f < \frac{1}{2 \cdot T_s}.$$

In this case the condition reads

$$f < \frac{1}{2 \cdot 1s} = 0.5Hz.$$

One can read the frequency of a signal from its formula: the value that multiplies t is ω and

$$\frac{\omega}{2\pi} = f.$$

The first three signals have

$$f = \frac{4\pi}{2\pi} = 2Hz.$$

which is greater than $0.5Hz$. Moreover, additional phase and gain don't play an important role in this sense. The next three signals have a frequency of

$$f = \frac{0.2\pi}{2\pi} = 0.1Hz.$$

that is lower than $0.5Hz$ and hence perfect, in order to not encounter aliasing. The next three signals have the critical frequency and theoretically speaking, one sets this as already aliased. The reason for that is that the Nyquist theorem sets a strict $<$ in the condition.

For the next two signals a special procedure applies: if one has a combination of signals, one has to look at the bigger frequency of the signal. In the first case this reads $2Hz$, that exceeds the limit frequency. In the second case this reads $0.2Hz$, that is acceptable. The last two cases are a general form of combination of signals. The leading frequency of the first sum, decreases with $\frac{1}{i+1}$ and has its biggest value with $i = 1$, namely $0.5Hz$. This is already at the limit frequency, hence not acceptable. The leading frequency of the second sum, decreases with $\frac{1}{i+2}$ and has its biggest value with $i = 1$, namely $0.33Hz$. This is lower than the limit frequency, hence acceptable.

(b) The general formula reads

$$f_s > 2 \cdot f_{max}.$$

Here it holds

$$f_{max} = \frac{40\pi}{2\pi} = 20Hz.$$

It follows

$$f_s > 2 \cdot 20Hz = 40Hz.$$

Example 13.

(a) For which A and B is the system asymptotically stable?

- ☐ $A = \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$
- ☐ $A = \begin{pmatrix} -1 & -2 \\ -1 & -2 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$
- ☐ $A = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, B = \begin{pmatrix} 0.1 \\ 0 \end{pmatrix}.$
- ☐ $A = \begin{pmatrix} -1 & -2 \\ 1 & -0.5 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$
- ☐ $A = \begin{pmatrix} -1 & -2 \\ 0 & -0.5 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$
- ☐ $A = \begin{pmatrix} 1 & -2 \\ 0 & 0.5 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$
- ☐ $A = \begin{pmatrix} -0.1 & -2 \\ 0 & -0.5 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$
- ☐ $A = \begin{pmatrix} -0.1 & -2 \\ 0 & -0.5 \end{pmatrix}, B = \begin{pmatrix} 0.1 \\ 0 \end{pmatrix}.$
- ☐ $A = \begin{pmatrix} 0.1 & -2 \\ 0 & 0.5 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$
- ☐ $A = \begin{pmatrix} 0.1 & -2 \\ 0 & 0.5 \end{pmatrix}, B = \begin{pmatrix} 0.1 \\ 0 \end{pmatrix}.$

(b) The previous exercise can be solved independently of B .

- ☐ True.
- ☐ False.

Solution.

- (a) ☐ $A = \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$
- ☐ $A = \begin{pmatrix} -1 & -2 \\ -1 & -2 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$
- ☒ $A = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, B = \begin{pmatrix} 0.1 \\ 0 \end{pmatrix}.$
- ☐ $A = \begin{pmatrix} -1 & -2 \\ 1 & -0.5 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$
- ☐ $A = \begin{pmatrix} -1 & -2 \\ 0 & -0.5 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$
- ☒ $A = \begin{pmatrix} -0.1 & -2 \\ 0 & -0.5 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$
- ☒ $A = \begin{pmatrix} -0.1 & -2 \\ 0 & -0.5 \end{pmatrix}, B = \begin{pmatrix} 0.1 \\ 0 \end{pmatrix}.$
- ☒ $A = \begin{pmatrix} 0.1 & -2 \\ 0 & 0.5 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$
- ☒ $A = \begin{pmatrix} 0.1 & -2 \\ 0 & 0.5 \end{pmatrix}, B = \begin{pmatrix} 0.1 \\ 0 \end{pmatrix}.$

Explanation:

The eigenvalues of A should fulfill

$$|\lambda_i| < 1.$$

(b)

- ☒ True.
- ☐ False.

Example 14. A continuous-time system with the following transfer function is considered:

$$G(s) = \frac{9}{s+3}.$$

- (a) Calculate the equivalent discrete-time transfer function $H(z)$. The latter is composed of a Digital-to-Analog Converter (DAC), the continuous-time transfer function $G(s)$ and an Analog-to-Digital Converter (ADC). Both converters, i.e. the DAC and ADC, have a sampling time $T_s = 1s$.
- (b) Calculate the static error if a proportional controller $K(z) = k_p$ is used and the reference input $y_{c,k}$ is a step signal.

Hint: Heavyside with amplitude equal to 1.

Solution.

- (a) Rather than taking into account all the individual elements which make up the continuous-time part of the system (DAC, plant, ADC), in a first step, these elements are lumped together and are represented by the discrete-time description $H(z)$. In this case, the discrete-time output of the system is given by

$$Y(z) = H(z) \cdot U(z),$$

where $U(z)$ is the z-transform of the discrete input u_k given to the system. Therefore, the discrete-time representation of the plant is given by the ratio of the output to the input

$$H(z) = \frac{Y(z)}{U(z)}.$$

For the sake of convenience, u_k is chosen to be the discrete-time *Heaviside* function

$$\begin{cases} u_k[k] = 1, & k \geq 0 \\ 0, & \text{else.} \end{cases}$$

This input function needs to be z-transformed. Recall the definition of the z-transform

$$X(z) = \sum_{n=0}^{\infty} x(n) \cdot z^{-n}.$$

and applying it to the above equation, with the input u_k gives ($u_k[k] = 1$ for $k \geq 0$)

$$\begin{aligned} U(z) &= X(u_k) \\ &= \sum_{k=0}^{\infty} z^{-k} \\ &= \sum_{k=0}^{\infty} (z^{-1})^k. \end{aligned}$$

For $U(z)$ to be defined, this sum must converge. Recalling the properties of geometric series one can see

$$U(z) = \frac{1}{1 - z^{-1}},$$

as long as the convergence criterion is satisfied, i.e. as long as $|z^{-1}| < 1$ or better $|z| > 1$. This signal is then transformed to continuous time using a zero-order hold DAC. The output of this transformation is again a Heaviside function $u_h(t)$. Since the signal is now in continuous time, the Laplace transform is used for the analysis. The Laplace transform of the step function is well known to be

$$\mathcal{L}(u_h(t))(s) = \frac{1}{s} = U(s).$$

The plant output in continuous time is given by

$$\begin{aligned} Y(s) &= G(s) \cdot U(s) \\ &= \frac{G(s)}{s}. \end{aligned}$$

After the plant $G(s)$, the signal is sampled and transformed into discrete time once more. Therefore, the z-transform of the output has to be calculated. However, the signal $Y(s)$

cannot be transformed directly, since it is expressed in the frequency domain. Thus, first, it has to be transformed back into the time domain ($y(t)$) using the inverse Laplace transform, where it is then sampled every $t = k \cdot T$. The resulting series of samples $\{y[k]\}$ is then transformed back into the z-domain, i.e.

$$Y(z) = X \left(\left\{ \mathcal{L}^{-1} \left(\frac{G(s)}{s} \right) (kT) \right\} \right).$$

To find the inverse Laplace transform of the output, its frequency domain representation is decomposed into a sum of simpler functions

$$\begin{aligned} \frac{G(s)}{s} &= \frac{9}{s \cdot (s + 3)} \\ &= \frac{\alpha}{s} + \frac{\beta}{s + 3} \\ &= \frac{s \cdot (\alpha + \beta) + 3 \cdot \alpha}{s \cdot (s + 3)}. \end{aligned}$$

The comparison of the numerators yields

$$\alpha = 3, \quad \beta = -3.$$

and thus

$$\begin{aligned} \frac{G(s)}{s} &= \frac{3}{s} - \frac{3}{s + 3} \\ &= 3 \cdot \left(\frac{1}{s} - \frac{1}{s + 3} \right). \end{aligned}$$

Now the terms can be individually transformed with the result

$$\begin{aligned} \mathcal{L}^{-1} \left(\frac{G(s)}{s} \right) &= 3 \cdot (1 - e^{-3t}) \cdot u_h(t) \\ &= y(t). \end{aligned}$$

The z-transform of the output sampled at discrete time instants $y(kT)$ is given by

$$\begin{aligned} X(\{y(kT)\}) &= 3 \cdot \left[\sum_{k=0}^{\infty} z^{-k} - \sum_{k=0}^{\infty} e^{-3kT} \cdot z^{-k} \right] \\ &= 3 \cdot \left[\sum_{k=0}^{\infty} z^{-1k} - \sum_{k=0}^{\infty} (e^{-3T} \cdot z^{-1})^k \right] \\ &= Y(z). \end{aligned}$$

From above, the two necessary convergence criteria are known

$$\begin{aligned} |z^{-1}| < 1 &\Rightarrow |z| > 1 \\ |e^{-3T} \cdot z^{-1}| &\Rightarrow |z| > |e^{-3T}|. \end{aligned}$$

Using the above equations the output transform converges to (given that the two convergence criteria are satisfied)

$$Y(z) = 3 \cdot \left(\frac{1}{1 - z^{-1}} - \frac{1}{1 - e^{-3T} \cdot z^{-1}} \right).$$

Finally, the target transfer function $H(z)$ is given by

$$\begin{aligned} H(z) &= \frac{Y(z)}{U(z)} \\ &= (1 - z^{-1}) \cdot Y(z) \\ &= 3 \cdot \left(1 - \frac{1 - z^{-1}}{1 - e^{-3T} \cdot z^{-1}} \right) \\ &= 3 \cdot \frac{(1 - e^{-3T}) \cdot z^{-1}}{1 - e^{-3T} \cdot z^{-1}}. \end{aligned}$$

(b) From the signal flow diagram, it can be seen that the error $\varepsilon(z)$ is composed of

$$\begin{aligned} \varepsilon(z) &= Y_c(z) - Y(z) \\ &= Y_c(z) - H(z) \cdot K(z) \cdot \varepsilon(z) \\ &= \frac{Y_c(z)}{1 + k_p \cdot H(z)}. \end{aligned}$$

The input $y_c(t)$ is a discrete step signal, for which the z-transform was calculated in (a):

$$Y_c(z) = \frac{1}{1 + z^{-1}}.$$

Therefore, the error signal reads

$$\varepsilon(z) = \frac{\frac{1}{1+z^{-1}}}{1 + 3 \cdot k_p \cdot \frac{(1-e^{-3T}) \cdot z^{-1}}{1-e^{-3T} \cdot z^{-1}}}.$$

To calculate the steady-state error, i.e. the error after infinite time, the discrete-time final value theorem⁹ is used:

$$\lim_{t \rightarrow \infty} \varepsilon(t) = \lim_{z \rightarrow 1} (1 - z^{-1}) \cdot \varepsilon(z),$$

but as z goes to 1, so does z^{-1} and therefore 1 is substituted for each z^{-1} in $\varepsilon(z)$ and the static error becomes

$$\varepsilon_{\infty} = \frac{1}{1 + 3 \cdot k_p}.$$

Note that the error does not completely vanish but can be made smaller by increasing k_p . This is the same behaviour which would have been expected from a purely proportional controller in continuous time. To drive the static error to zero, a discrete-time integrator of the form $\frac{1}{T_i \cdot (1 - z^{-1})}$ would be necessary.

⁹ $\lim_{t \rightarrow \infty} \varepsilon(t) = \lim_{s \rightarrow 0} s \cdot \varepsilon(s)$

4 MIMO Systems

MIMO systems are systems with multiple inputs and multiple outputs. In this chapter we will introduce some analytical tools to deal with those systems.

4.1 System Description

4.1.1 State Space Description

The state-space description of a MIMO system is very similar to the one of a SISO system. For a linear, time invariant MIMO system with m input signals and p output signals, it holds

$$\begin{aligned} \frac{d}{dt}x(t) &= A \cdot x(t) + B \cdot u(t), & x(t) &\in \mathbb{R}^n, u(t) \in \mathbb{R}^m \\ y(t) &= C \cdot x(t) + D \cdot u(t), & y(t) &\in \mathbb{R}^p \end{aligned} \quad (4.1)$$

where

$$x(t) \in \mathbb{R}^{n \times 1}, u(t) \in \mathbb{R}^{m \times 1}, y(t) \in \mathbb{R}^{p \times 1}, A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m}, C \in \mathbb{R}^{p \times n}, D \in \mathbb{R}^{p \times m}. \quad (4.2)$$

Remark. The dimensions of the matrices A, B, C, D are very important and they are a key concept to understand problems.

The big difference from SISO systems is that $u(t)$ and $y(t)$ are here vectors and no more just numbers. For this reason B, C, D are now matrices.

4.1.2 Transfer Function

One can compute the transfer function of a MIMO system with the well known formula

$$P(s) = C \cdot (s \cdot \mathbb{1} - A)^{-1} \cdot B + D. \quad (4.3)$$

This is no more a scalar, but a $p \times m$ -matrix. The elements of that matrix are rational functions. Mathematically:

$$P(s) = \begin{pmatrix} P_{11}(s) & \cdots & P_{1m}(s) \\ \vdots & \ddots & \vdots \\ P_{p1}(s) & \cdots & P_{pm}(s) \end{pmatrix}, \quad P_{ij}(s) = \frac{b_{ij}(s)}{a_{ij}(s)}. \quad (4.4)$$

Here $P_{ij}(s)$ is the transfer function from the j -th input to the i -th output.

Remark. In the SISO case, the only matrix we had to care about was A . Since now B, C are matrices, one has to pay attention to some mathematical properties: the matrix multiplication is not **commutative** ($A \cdot B \neq B \cdot A$). Since now $P(s)$ and $C(s)$ are matrices, it holds

$$L(s) = P(s) \cdot C(s) \neq C(s) \cdot P(s). \quad (4.5)$$

Moreover, one can no more define the complementary sensitivity and the sensitivity as

$$T(s) = \frac{L(s)}{1 + L(s)}, \quad S(s) = \frac{1}{1 + L(s)}. \quad (4.6)$$

because no matrix division is defined. There are however similar expressions to describe those transfer functions: let's try to derive them.

We start from the standard control system's structure (see Figure 32).

To keep things general, let's say the plant $P(s) \in \mathbb{C}^{p \times m}$ and the controller $C(s) \in \mathbb{C}^{m \times p}$. The reference $r \in \mathbb{R}^p$, the input $u \in \mathbb{R}^m$ and the disturbance $d \in \mathbb{R}^p$. The output $Y(s)$ can as always be written as

$$Y(s) = T(s) \cdot R(s) + S(s) \cdot D(s). \quad (4.7)$$

where $T(s)$ is the transfer function of the complementary sensitivity and $S(s)$ is the transfer function of the sensitivity.

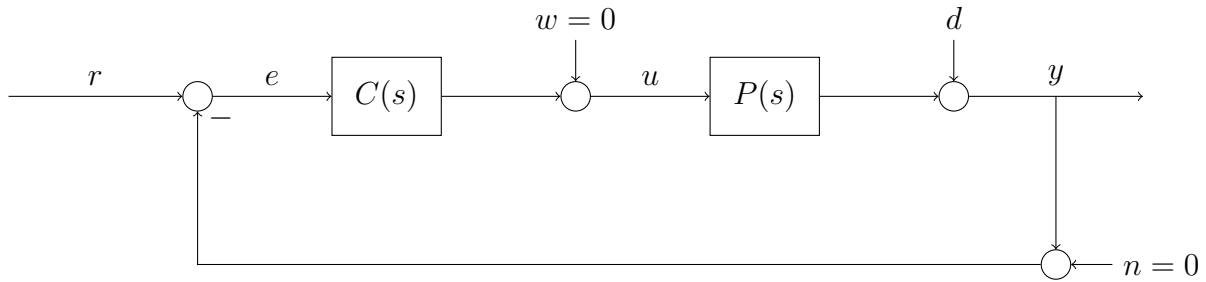


Figure 32: Standard feedback control system structure.

Starting from the error $E(s)$

If one wants to determine the matrices of those transfer functions, one can start writing (by paying attention to the direction of multiplications) with respect to $E(s)$

$$\begin{aligned} E(s) &= R(s) - P(s) \cdot C(s) \cdot E(s) - D(s), \\ Y(s) &= P(s) \cdot C(s) \cdot E(s) + D(s). \end{aligned} \quad (4.8)$$

This gives in the first place

$$E(s) = (\mathbb{1} + P(s) \cdot C(s))^{-1} \cdot (R(s) - D(s)). \quad (4.9)$$

Inserting and writing the functions as $F(s) = F$ for simplicity, one gets

$$\begin{aligned} Y &= P \cdot C \cdot E + D \\ &= P \cdot C \cdot (\mathbb{1} + P \cdot C)^{-1} \cdot (R - D) + D \\ &= P \cdot C \cdot (\mathbb{1} + P \cdot C)^{-1} \cdot R - P \cdot C (\mathbb{1} + P \cdot C)^{-1} \cdot D + \underbrace{(\mathbb{1} + P \cdot C) (\mathbb{1} + P \cdot C)^{-1}}_{\mathbb{1}} \cdot D \\ &= P \cdot C \cdot (\mathbb{1} + P \cdot C)^{-1} \cdot R + (\mathbb{1} + P \cdot C - P \cdot C) \cdot (\mathbb{1} + P \cdot C)^{-1} \cdot D \\ &= P \cdot C \cdot (\mathbb{1} + P \cdot C)^{-1} \cdot R + (\mathbb{1} + P \cdot C)^{-1} \cdot D. \end{aligned} \quad (4.10)$$

Recalling the general equation (5.2) one gets the two transfer functions:

$$\begin{aligned} T_1(s) &= P(s) \cdot C(s) \cdot (\mathbb{1} + P(s) \cdot C(s))^{-1}, \\ S_1(s) &= (\mathbb{1} + P(s) \cdot C(s))^{-1}. \end{aligned} \quad (4.11)$$

Starting from the input $U(s)$

If one starts with respect to $U(s)$, one gets

$$\begin{aligned} U(s) &= C(s) \cdot (R(s) - D(s)) - C(s) \cdot P(s) \cdot U(s), \\ Y(s) &= P(s) \cdot U(s) + D(s). \end{aligned} \quad (4.12)$$

This gives in the first place

$$U(s) = (\mathbb{1} + C(s) \cdot P(s))^{-1} \cdot C(s) \cdot (R(s) - D(s)). \quad (4.13)$$

Inserting and writing the functions as $F(s) = F$ for simplicity, one gets

$$\begin{aligned} Y &= P \cdot U + D \\ &= P \cdot (\mathbb{1} + C \cdot P)^{-1} \cdot C \cdot (R - D) + D \\ &= P \cdot (\mathbb{1} + C \cdot P)^{-1} \cdot C \cdot R + (\mathbb{1} - P \cdot (\mathbb{1} + C \cdot P)^{-1} \cdot C) \cdot D. \end{aligned} \quad (4.14)$$

Recalling the general equation (5.2) one gets the two transfer functions:

$$\begin{aligned} T_2(s) &= P(s) \cdot (\mathbb{1} + C(s) \cdot P(s))^{-1} \cdot C(s), \\ S_2(s) &= \mathbb{1} - P(s) \cdot (\mathbb{1} + C(s) \cdot P(s))^{-1} \cdot C(s). \end{aligned} \quad (4.15)$$

It can be shown that this two different results actually are the equivalent. It holds

$$\begin{aligned} S_1 &= S_2 \\ (\mathbb{1} + P \cdot C)^{-1} &= \mathbb{1} - P \cdot (\mathbb{1} + C \cdot P)^{-1} \cdot C \\ \mathbb{1} &= \mathbb{1} + P \cdot C - P \cdot (\mathbb{1} + C \cdot P)^{-1} \cdot C \cdot (\mathbb{1} + P \cdot C) \\ \mathbb{1} &= \mathbb{1} + P \cdot C - P \cdot (\mathbb{1} + C \cdot P)^{-1} \cdot (C + C \cdot P \cdot C) \\ \mathbb{1} &= \mathbb{1} + P \cdot C - P \cdot (\mathbb{1} + C \cdot P)^{-1} \cdot (\mathbb{1} + C \cdot P) \cdot C \\ \mathbb{1} &= \mathbb{1} + P \cdot C - P \cdot C \\ \mathbb{1} &= \mathbb{1} \end{aligned} \quad (4.16)$$

$$\begin{aligned} T_1 &= T_2 \\ P \cdot C \cdot (\mathbb{1} + P \cdot C)^{-1} &= P \cdot (\mathbb{1} + C \cdot P)^{-1} \cdot C \\ P \cdot C &= P \cdot (\mathbb{1} + C \cdot P)^{-1} \cdot C \cdot (\mathbb{1} + P \cdot C) \\ P \cdot C &= P \cdot (\mathbb{1} + C \cdot P)^{-1} \cdot (C + C \cdot P \cdot C) \\ P \cdot C &= P \cdot (\mathbb{1} + C \cdot P)^{-1} \cdot (\mathbb{1} + C \cdot P) \cdot C \\ P \cdot C &= C \cdot P \\ \mathbb{1} &= \mathbb{1}. \end{aligned}$$

Finally, one can show that

$$\begin{aligned} S(s) + T(s) &= (\mathbb{1} + P \cdot C)^{-1} + P \cdot C \cdot (\mathbb{1} + P \cdot C)^{-1} \\ &= (\mathbb{1} + P \cdot C) \cdot (\mathbb{1} + P \cdot C)^{-1} \\ &= \mathbb{1}. \end{aligned} \quad (4.17)$$

4.2 System Analysis

4.2.1 Lyapunov Stability

The Lyapunov stability theorem analyses the behaviour of a system near to its equilibrium points when $u(t) = 0$. Because of this, we don't care if the system is MIMO or SISO. The three cases read

- Asymptotically stable: $\lim_{t \rightarrow \infty} \|x(t)\| = 0$;
- Stable: $\|x(t)\| < \infty \forall t \geq 0$;
- Unstable: $\lim_{t \rightarrow \infty} \|x(t)\| = \infty$.

As it was done for the SISO case, one can show by using $x(t) = e^{A \cdot t} \cdot x_0$ that the stability can be related to the eigenvalues of A through:

- Asymptotically stable: $\operatorname{Re}(\lambda_i) < 0 \forall i$;
- Stable: $\operatorname{Re}(\lambda_i) \leq 0 \forall i$;
- Unstable: $\operatorname{Re}(\lambda_i) > 0$ for at least one i .

4.2.2 Controllability

Controllable: is it possible to control all the states of a system with an input $u(t)$?
A system is said to be **completely controllable**, if the **Controllability Matrix**

$$R = \begin{pmatrix} B & A \cdot B & A^2 \cdot B & \dots & A^{n-1} \cdot B \end{pmatrix} \in \mathbb{R}^{n \times (n \cdot m)}. \quad (4.18)$$

has full rank n .

4.2.3 Observability

Observable: is it possible to reconstruct the initial conditions of all the states of a system from the output $y(t)$?

A system is said to be **completely observable**, if the **Observability Matrix**

$$O = \begin{pmatrix} C \\ C \cdot A \\ C \cdot A^2 \\ \vdots \\ C \cdot A^{n-1} \end{pmatrix} \in \mathbb{R}^{(n \cdot p) \times n}. \quad (4.19)$$

has full rank n .

4.3 Poles and Zeros

Since we have to deal with matrices, one has to use the theory of *minors* (see *Lineare Algebra I/II*) in order to compute the zeros and the poles of a transfer function.

The first step of this computation is to calculate all the minors of the transfer function $P(s)$. The minors of a matrix $F \in \mathbb{R}^{n \times m}$ are the determinants of all square submatrices. By **maximal minor** it is meant the minor with the biggest dimension. From the minors one can calculate the poles and the zeros as follows:

4.3.1 Zeros

The zeros are the zeros of the numerator's *greatest common divisor* of the maximal minors, after their normalization with respect to the same denominator (*polepolynom*).

4.3.2 Poles

The poles are the zeros of the *least common denominator* of all the minors of $P(s)$.

4.3.3 Directions

In MIMO systems, the poles and the zeros are related to a **direction**. Moreover, a zero-pole cancellation occurs only if zero and pole have the same magnitude and input-output direction. The directions $\delta_{\pi,i}^{\text{in,out}}$ associated with a pole π_i are defined by

$$P(s)|_{s=\pi_i} \cdot \delta_{\pi,i}^{\text{in}} = \infty \cdot \delta_{\pi,i}^{\text{out}}. \quad (4.20)$$

The directions $\delta_{\xi,i}^{\text{in,out}}$ associated with a zero ξ_i are defined by

$$P(s)|_{s=\xi_i} \cdot \delta_{\xi,i}^{\text{in}} = 0 \cdot \delta_{\xi,i}^{\text{out}}. \quad (4.21)$$

The directions can be computed with the **singular value decomposition** (see next week) of the matrix $P(S)$.

4.4 Nyquist Theorem for MIMO Systems

The Nyquist theorem can also be written for MIMO systems. The closed-loop $T(s)$ is asymptotically stable if

$$n_c = n_+ + \frac{1}{2} \cdot n_0 \quad (4.22)$$

where

- n_c : Encirclements about the origin of

$$N(i \cdot \omega) = \det(\mathbb{1} + P(i \cdot \omega) \cdot C(i \cdot \omega)). \quad (4.23)$$

- n_+ : number of unstable poles.
- n_0 : number of marginal stable poles.

4.5 Examples

Example 15. The minors of a given matrix

$$P(s) = \begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix}$$

are:

First order:

$$a, b, c, d, e, f$$

Second order:

$$\det \begin{pmatrix} a & b \\ d & e \end{pmatrix}, \det \begin{pmatrix} a & c \\ d & f \end{pmatrix}, \det \begin{pmatrix} b & c \\ e & f \end{pmatrix}.$$

Example 16. One wants to find the poles and the zeros of the given transfer function

$$P(s) = \begin{pmatrix} \frac{s+2}{s+3} & 0 \\ 0 & \frac{(s+1) \cdot (s+3)}{s+2} \end{pmatrix}.$$

Solution. First of all, we list **all** the minors of the transfer function:

Minors:

- First order: $\frac{s+2}{s+3}$, $\frac{(s+1) \cdot (s+3)}{s+2}$, 0, 0 ;
- Second order: $s + 1$.

Poles:

The least common denominator of all the minors is

$$(s + 3) \cdot (s + 2)$$

This means that the poles are

$$\begin{aligned}\pi_1 &= -2 \\ \pi_2 &= -3.\end{aligned}$$

Zeros:

The maximal minor is $s + 1$ and we have to normalize it with respect to the polepolynom $(s + 3) \cdot (s + 2)$. It holds

$$(s + 1) \Rightarrow \frac{(s + 1) \cdot (s + 2) \cdot (s + 3)}{(s + 2) \cdot (s + 3)}$$

The numerator reads

$$(s + 1) \cdot (s + 2) \cdot (s + 3)$$

and so the zeros are

$$\begin{aligned}\zeta_1 &= -1 \\ \zeta_2 &= -2 \\ \zeta_3 &= -3.\end{aligned}$$

Example 17. One wants to find the poles and the zeros of the given transfer function

$$P(s) = \begin{pmatrix} \frac{1}{s+1} & \frac{1}{s+2} & \frac{2 \cdot (s+1)}{(s+2) \cdot (s+3)} \\ 0 & \frac{s+3}{(s+1)^2} & \frac{s+4}{s+1} \end{pmatrix}.$$

Solution. First of all, we list **all** the minors of the transfer function:

Minors:

- First order: $\frac{1}{s+1}, \frac{1}{s+2}, \frac{2 \cdot (s+1)}{(s+2) \cdot (s+3)}, 0, \frac{s+3}{(s+1)^2}, \frac{s+4}{s+1},$
- Second order: $\frac{s+3}{(s+1)^3}, \frac{s+4}{(s+1) \cdot (s+2)} - \frac{2}{(s+2) \cdot (s+1)} = \frac{1}{s+1}, -\frac{s+4}{(s+1)^2}.$

Poles:

The least common denominator of all the minors is

$$(s+1)^3 \cdot (s+2) \cdot (s+3).$$

This means that the poles are

$$\begin{aligned}\pi_1 &= -1 \\ \pi_2 &= -1 \\ \pi_3 &= -1 \\ \pi_4 &= -2 \\ \pi_5 &= -3.\end{aligned}$$

Zeros:

The numerators of the maximal minors are $(s+3)$, 1 and $-(s+4)$. We have to normalize them with respect to the polepolynom $(s+1)^3 \cdot (s+2) \cdot (s+3)$. It holds

$$\begin{aligned}(s+3) &\Rightarrow \frac{(s+3)^2 \cdot (s+2)}{(s+1)^3 \cdot (s+2) \cdot (s+3)}, \\ 1 &\Rightarrow \frac{(s+1)^2 \cdot (s+2) \cdot (s+3)}{(s+1)^3 \cdot (s+2) \cdot (s+3)}, \\ -(s+4) &\Rightarrow -\frac{(s+4) \cdot (s+1) \cdot (s+2) \cdot (s+3)}{(s+1)^3 \cdot (s+2) \cdot (s+3)}.\end{aligned}$$

The greatest common divisor of these is

$$(s+3) \cdot (s+2).$$

Hence, the zeros are

$$\begin{aligned}\zeta_1 &= -2, \\ \zeta_2 &= -3.\end{aligned}$$

Example 18. The dynamics of a system are given as

$$\begin{aligned}\dot{x}(t) &= \begin{pmatrix} 4 & 1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix} \cdot x(t) + \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \cdot u(t) \\ y(t) &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} \cdot x(t).\end{aligned}\tag{4.24}$$

Moreover the transfer function is given as

$$P(s) = \begin{pmatrix} \frac{(s-2)}{s^2-6s+9} & 0 \\ \frac{-1}{s^2-6s+9} & \frac{1}{s-2} \end{pmatrix}.\tag{4.25}$$

- (a) Is the system Lyapunov stable, asymptotically stable or unstable?
- (b) Is the system completely controllable?
- (c) Is the system completely observable?
- (d) The poles of the system are $\pi_1 = 2$ and $\pi_{2,3} = 3$. The zero of the system is $\zeta_1 = 2$. Are there any zero-pole cancellations?
- (e) Write a MATLAB code that computes the transfer function basing on A, B, C, D .

Solution.

(a)

First of all, one identifies the matrices as:

$$\begin{aligned} \dot{x}(t) &= \underbrace{\begin{pmatrix} 4 & 1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}}_A \cdot x(t) + \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}}_B \cdot u(t) \\ y(t) &= \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}}_C \cdot x(t) + \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}}_D \cdot u(t). \end{aligned} \quad (4.26)$$

We have to compute the eigenvalues of A. It holds

$$\begin{aligned} \det(A - \lambda \cdot \mathbb{1}) &= \left| \begin{pmatrix} 4 - \lambda & 1 & 0 \\ -1 & 2 - \lambda & 0 \\ 0 & 0 & 2 - \lambda \end{pmatrix} \right| \\ &= (2 - \lambda) \cdot \left| \begin{pmatrix} 4 - \lambda & 1 \\ -1 & 2 - \lambda \end{pmatrix} \right| \\ &= (2 - \lambda) \cdot ((4 - \lambda) \cdot (2 - \lambda) + 1) \\ &= (2 - \lambda) \cdot (\lambda^2 - 6\lambda + 9) \\ &= (2 - \lambda) \cdot (\lambda - 3)^2. \end{aligned}$$

Since all the three eigenvalues are bigger than zero, the system is Lyapunov **unstable**.

(b)

The controllability matrix can be found with the well-known multiplications:

$$\begin{aligned} A \cdot B &= \begin{pmatrix} 4 & 1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 4 & 0 \\ -1 & 0 \\ 0 & 2 \end{pmatrix}, \\ A^2 \cdot B &= \begin{pmatrix} 4 & 1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix} \cdot \begin{pmatrix} 4 & 0 \\ -1 & 0 \\ 0 & 2 \end{pmatrix} \\ &= \begin{pmatrix} 15 & 0 \\ -6 & 0 \\ 0 & 4 \end{pmatrix}. \end{aligned}$$

Hence, the controllability matrix reads

$$R = \begin{pmatrix} 1 & 0 & 4 & 0 & 15 & 0 \\ 0 & 0 & -1 & 0 & -6 & 0 \\ 0 & 1 & 0 & 2 & 0 & 4 \end{pmatrix}.$$

This has full rank 3: the system is completely controllable.

(c)

The observability matrix can be found with the well-known multiplications:

$$\begin{aligned} C \cdot A &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 4 & 1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix} \\ &= \begin{pmatrix} 4 & 1 & 0 \\ -1 & 2 & 2 \end{pmatrix}, \\ C \cdot A^2 &= \begin{pmatrix} 4 & 1 & 0 \\ -1 & 2 & 2 \end{pmatrix} \cdot \begin{pmatrix} 4 & 1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix} \\ &= \begin{pmatrix} 15 & 6 & 0 \\ -6 & 3 & 4 \end{pmatrix}. \end{aligned}$$

Hence, the observability matrix reads

$$O = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 4 & 1 & 0 \\ -1 & 2 & 2 \\ 15 & 6 & 0 \\ -6 & 3 & 4 \end{pmatrix}.$$

This has full rank 3: the system is completely observable.

(d)

Although $\zeta_1 = 2$ and $\pi_1 = 2$ have the same magnitude, they don't cancel out. Why? Since the system is completely controllable and completely observable, we have already the minimal realization of the system. This means that no more cancellation is possible. The reason for that is that the directions of the two don't coincide. We will learn more about this in the next chapter.

(e)

The code reads

```
P=tf(ss(A,B,C,D));
```

or alternatively

```
P=tf('s');
P=C*inv(s*eye(3)-A)*B+D;
```

Example 19. Given is the system

$$\begin{aligned}\dot{x}(t) &= \begin{pmatrix} -2 & 0 & 0 \\ 0 & -2 & 5 \\ 0 & -1 & 0 \end{pmatrix} \cdot x(t) + \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 1 \end{pmatrix} \cdot u(t) \\ y(t) &= \begin{pmatrix} -1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \cdot x(t),\end{aligned}$$

with two inputs $u_1(t)$ and $u_2(t)$ and two outputs $y_1(t)$ and $y_2(t)$. The transfer function of the system reads

$$P(s) = \begin{pmatrix} \frac{2s-1}{(s^2+2s+5) \cdot (s+2)} & \frac{s+2}{s^2+2s+5} \\ \frac{5}{s^2+2s+5} & \frac{5}{s^2+2s+5} \end{pmatrix}.$$

- (a) How many state variables are needed, in order to describe the input/output behaviour of the system?
- (b) How many outputs are needed, in order to reconstruct the initial state $x(0)$?
- (c) For every $x(0) \neq 0$ we have to ensure $\lim_{t \rightarrow \infty} x(t) = 0$. How many inputs are needed, in order to ensure this condition?

Solution.

- (a) We want to find the minimal order of the system, that is, the number of poles.

Minors 1st. Order

$$\frac{2s-1}{(s^2+2s+5) \cdot (s+2)}, \quad \frac{s+2}{s^2+2s+5}, \quad \frac{5}{s^2+2s+5}.$$

Minors 2nd. Order

$$\begin{aligned} & \frac{2s-1}{(s^2+2s+5) \cdot (s+2)} \cdot \frac{5}{s^2+2s+5} - \frac{s+2}{s^2+2s+5} \cdot \frac{5}{s^2+2s+5} \\ &= \frac{5}{s^2+2s+5} \cdot \left(\frac{2s-1}{(s^2+2s+5) \cdot (s+2)} - \frac{s+2}{s^2+2s+5} \right) \\ &= \frac{5}{s^2+2s+5} \cdot \frac{-s^2-2s-5}{(s^2+2s+5) \cdot (s+2)} \\ &= \frac{-5}{(s^2+2s+5) \cdot (s+2)}. \end{aligned}$$

The pole-polynom reads

$$(s^2+2s+5) \cdot (s+2).$$

and the poles

$$\begin{aligned} \pi_1 &= -2 \\ \pi_{2,3} &= -1 \pm 2j. \end{aligned}$$

The minimal order of the system is $n = 3$, since we have 3 poles. The given description is already in the minimal realization.

Alternative

As alternative one can compute the controllability matrix and the observability matrix as follows: for the controllability matrix it holds:

$$\begin{aligned} A \cdot B &= \begin{pmatrix} -2 & 0 & 0 \\ 0 & -2 & 5 \\ 0 & -1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 1 \end{pmatrix} \\ &= \begin{pmatrix} -2 & 0 \\ 5 & 5 \\ 0 & 0 \end{pmatrix}. \\ A^2 \cdot B &= \begin{pmatrix} -2 & 0 & 0 \\ 0 & -2 & 5 \\ 0 & -1 & 0 \end{pmatrix} \cdot \begin{pmatrix} -2 & 0 \\ 5 & 5 \\ 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 4 & 0 \\ -10 & -10 \\ -5 & -5 \end{pmatrix}. \end{aligned}$$

The matrix reads

$$R = \begin{pmatrix} 1 & 0 & -2 & 0 & 4 & 0 \\ 0 & 0 & 5 & 5 & -10 & -10 \\ 1 & 1 & 0 & 0 & -5 & -5 \end{pmatrix}.$$

This matrix has full rank $r = 3 = n$: the system is completely controllable.
For the observability matrix it holds:

$$\begin{aligned} C \cdot A &= \begin{pmatrix} -1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} -2 & 0 & 0 \\ 0 & -2 & 5 \\ 0 & -1 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 2 & -1 & 0 \\ 0 & -2 & 5 \end{pmatrix}. \\ C \cdot A^2 &= \begin{pmatrix} 2 & -1 & 0 \\ 0 & -2 & 5 \end{pmatrix} \cdot \begin{pmatrix} -2 & 0 & 0 \\ 0 & -2 & 5 \\ 0 & -1 & 0 \end{pmatrix} \\ &= \begin{pmatrix} -4 & 2 & -5 \\ 0 & -1 & -10 \end{pmatrix}. \end{aligned}$$

The matrix reads

$$O = \begin{pmatrix} -1 & 0 & 1 \\ 0 & 1 & 0 \\ 2 & -1 & 0 \\ 0 & -2 & 5 \\ -4 & 2 & -5 \\ 0 & -1 & -10 \end{pmatrix}.$$

This matrix has full rank $r = 3 = n$: the system is completely observable.

Since the system is completely observable and controllable, the given state-space description is already in its minimal realization, which means that the minimal order is $n = 3$.

(b) If we take into account just the first output $y_1(t)$, we get

$$C_1 = (-1 \quad 0 \quad 1)$$

with its observability matrix

$$\begin{pmatrix} -1 & 0 & 1 \\ 2 & -1 & 0 \\ -4 & 2 & -5 \end{pmatrix}.$$

This matrix has full rank and the partial system is completely observable: this means that $x(0)$ can be reconstructed from the first output.

Remark. This holds e.g. not for the second output $y_2(t)$. One would get

$$C_2 = (0 \quad 1 \quad 0)$$

with its observability matrix

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & -2 & 5 \\ 0 & -1 & -10 \end{pmatrix}.$$

This observability matrix has rank $r = 2$ and so the system is not completely observable: $x(0)$ cannot be reconstructed from the second output only.

(c) None. The eigenvalues of the system are

$$\begin{aligned}\lambda_1 &= -2 \\ \lambda_{2,3} &= -1 \pm 2j.\end{aligned}$$

These eigenvalues are all asymptotically stable: this means that no matter which input is given, the state will come back to its initial state.

Remark. If this wouldn't be the case, we would proceed like in task (b): if one takes into account just input $u_1(t)$, one gets

$$B_1 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

and its controllability matrix

$$\begin{pmatrix} 1 & -2 & 4 \\ 0 & 5 & -10 \\ 1 & 0 & -5 \end{pmatrix}.$$

This controllability matrix has full rank and would satisfy the condition. You can check yourselves that this wouldn't be the case for $u_2(t)$.

4.6 Relative-Gain Array (RGA)

Although we introduced the structure of MIMO systems, one could still ask himself why do we have to worry about all these new concepts and why don't we easily consider a MIMO system as a superposition of different SISO systems. In some cases, this reasoning is actually the good one, but how can one distinguish when to use this approach?

The RGA-matrix tells us how the different subplants of a MIMO plant interact: this matrix is a good indicator of *how SISO* a system is.

This matrix can be generally calculated as

$$\text{RGA}(s) = P(s) \cdot P(s)^{-T} \quad (4.27)$$

where

$$P(s)^{-T} = (P(s)^T)^{-1}. \quad (4.28)$$

and $A \cdot A$ represents the **element-wise** multiplication (**A.*A** in MATLAB).

In general, each element of the matrix gives us a special information:

$$[\text{RGA}]_{ab} = \frac{\text{gain from } u_a \text{ to } y_b \text{ with all other loops open}}{\text{gain from } u_a \text{ to } y_b \text{ with all other loops closed (perfect control)}}. \quad (4.29)$$

Remark. It's intuitive to notice, that if

$$[\text{RGA}]_{ab} \approx 1 \quad (4.30)$$

the numerator and the denominator are equal, i.e. SISO control is enough to bring u_a at y_b .

Remark. The theory behind the relative-gain array goes far beyond the aim of this course and one should be happy with the given examples. If however you are interested in this topic, you can have a look here.

Let's take the example of a 2×2 plant: in order to compute the first element $(1, 1)$ of the $\text{RGA}(s)$ we consider the system in Figure 33. We close with a SISO controller $C_{22}(s)$ the loop from y_2 to u_2 and try to compute the transfer function from u_1 to y_1 .

Everyone has his special way to decouple a MIMO system; I've always used this procedure: starting from the general equation in frequency domain

$$\begin{pmatrix} Y_1(s) \\ Y_2(s) \end{pmatrix} = \begin{pmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{pmatrix} \cdot \begin{pmatrix} U_1(s) \\ U_2(s) \end{pmatrix}. \quad (4.31)$$

one can read

$$\begin{aligned} Y_1(s) &= P_{11}(s) \cdot U_1(s) + P_{12}(s) \cdot U_2(s) \\ Y_2(s) &= P_{21}(s) \cdot U_1(s) + P_{22}(s) \cdot U_2(s). \end{aligned} \quad (4.32)$$

Since we want to relate u_1 and y_1 let's express u_2 as something we know. Using the controller $C_{22}(s)$ we see

$$\begin{aligned} U_2(s) &= -C_{22}(s) \cdot Y_2(s) \\ &= -C_{22}(s) \cdot P_{21}(s) \cdot U_1(s) - C_{22}(s) \cdot P_{22}(s) \cdot U_2(s) \\ \Rightarrow U_2(s) &= \frac{-C_{22}(s) \cdot P_{21}(s) \cdot U_1(s)}{1 + P_{22}(s) \cdot C_{22}(s)}. \end{aligned} \quad (4.33)$$

With the general equation one can then write

$$\begin{aligned} Y_1(s) &= P_{11}(s) \cdot U_1(s) + P_{12}(s) \cdot U_2(s) \\ &= P_{11}(s) \cdot U_1(s) + P_{12}(s) \cdot \frac{-C_{22}(s) \cdot P_{21}(s) \cdot U_1(s)}{1 + P_{22}(s) \cdot C_{22}(s)} \\ &= \frac{P_{11}(s) \cdot (1 + P_{22}(s) \cdot C_{22}(s)) - P_{12}(s) \cdot C_{22}(s) \cdot P_{21}(s)}{1 + P_{22}(s) \cdot C_{22}(s)} \cdot U_1(s). \end{aligned} \quad (4.34)$$

We have found the general transfer function that relates u_1 to y_1 . We now consider two extreme cases:

- We assume open loop conditions, i.e. *all other loops open*: $C_{22} \approx 0$. One gets

$$Y_1(s) = P_{11}(s) \cdot U_1(s). \quad (4.35)$$

- We assume high controller gains, i.e. *all other loops closed*: $P_{22}(s) \cdot C_{22}(s) \gg 1$. One gets

$$\begin{aligned} \lim_{C_{22}(s) \rightarrow \infty} \frac{P_{11}(s) \cdot (1 + P_{22}(s) \cdot C_{22}(s)) - P_{12}(s) \cdot C_{22}(s) \cdot P_{21}(s)}{1 + P_{22}(s) \cdot C_{22}(s)} \\ = \frac{P_{11}(s) \cdot P_{22}(s) - P_{11}(s) \cdot P_{21}(s)}{P_{22}(s)}. \end{aligned} \quad (4.36)$$

As stated before, the first element of the RGA is the division of these two. It holds

$$\begin{aligned} [\text{RGA}]_{11} &= \frac{P_{11}(s)}{\frac{P_{11}(s) \cdot P_{22}(s) - P_{11}(s) \cdot P_{21}(s)}{P_{22}(s)}} \\ &= \frac{P_{11}(s) \cdot P_{22}(s)}{P_{11}(s) \cdot P_{22}(s) - P_{11}(s) \cdot P_{21}(s)}. \end{aligned} \quad (4.37)$$

Remark. As you can see, the definition of the element of the RGA matrix does not depend on the chosen controller $C_{22}(s)$. This makes this method extremely powerful.

By repeating the procedure one can try to find $[\text{RGA}]_{22}$. In order to do that one has to close the loop from y_1 to u_1 : the result will be exactly the same:

$$[\text{RGA}]_{11} = [\text{RGA}]_{22}. \quad (4.38)$$

Let's go a step further. In order to compute the element $[\text{RGA}]_{21}$, one has to close the loop from y_1 to u_2 and find the transfer function from u_1 to y_2 .

Remark. This could be a nice exercise to test your understanding!

With a similar procedure one gets

$$[\text{RGA}]_{21} = \frac{-P_{12}(s) \cdot P_{21}(s)}{P_{22}(s) \cdot P_{11}(s) - P_{21}(s) \cdot P_{12}(s)}. \quad (4.39)$$

and as before

$$[\text{RGA}]_{21} = [\text{RGA}]_{12}. \quad (4.40)$$

How can we now use this matrix, to know is SISO control would be enough? As already stated before, $[\text{RGA}]_{ab} \approx 1$ means SISO control is enough. Moreover, if the diagonal terms differ substantially from 1, the MIMO interactions (also called *cross couplings*) are too important and a SISO control is no more recommended.

If

$$\text{RGA} \approx \mathbb{1} \quad (4.41)$$

evaluated at the relevant frequencies of the system, i.e. at $\omega_c \pm$ **one decade**, one can ignore the crosscouplings and can control the system with SISO tools *One loop at time*. If this is not the case, one has to design a MIMO controller. A bunch of observations could be useful by calculations:

- Rows and Columns add up to 1. This means one can write the matrix as

$$\begin{pmatrix} [\text{RGA}]_{11} & [\text{RGA}]_{12} \\ [\text{RGA}]_{21} & [\text{RGA}]_{22} \end{pmatrix} = \begin{pmatrix} [\text{RGA}]_{11} & 1 - [\text{RGA}]_{11} \\ 1 - [\text{RGA}]_{11} & [\text{RGA}]_{11} \end{pmatrix}. \quad (4.42)$$

This allows to calculate just one element of the matrix.

- If one looks at $\text{RGA}(s = 0)$ and the diagonal entries of the matrix are positive, SISO control is possible.
- The RGA of a triangular matrix $P(s)$ is the identity matrix.
- The RGA is invariant to scaling, i.e. for every diagonal matrix D_i it holds

$$[\text{RGA}](P(s)) = [\text{RGA}](D_1 \cdot P(s) \cdot D_2). \quad (4.43)$$

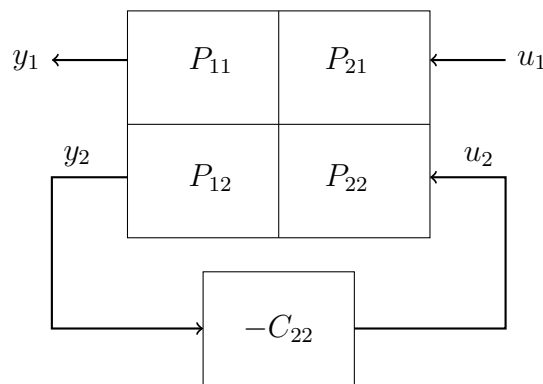


Figure 33: Derivation of the RGA-Matrix for the 2×2 case.

Example 20. For a MIMO system with two inputs and two outputs just the first element of the RGA matrix is given. This is a function of a system parameter p and is given as

$$[\text{RGA}(s)]_{11} = \frac{1}{ps^2 + 2ps + 1}.$$

- (a) Find the other elements of the RGA matrix.
- (b) For which values of p is the system for all frequencies $\omega \in [0, \infty)$ controllable with two independent SISO control loops (*one loop at the time*)?

Now, you are given the following transfer function of another MIMO system:

$$\begin{pmatrix} \frac{1}{s} & \frac{s+2}{s+1} \\ 1 & -\frac{1}{s+1} \end{pmatrix}.$$

- (c) Find the RGA matrix of this MIMO system.
- (c) Use the computed matrix to see if for frequencies in the range $\omega \in [3, 10]$ rad/s the system is controllable with two separate SISO controllers.

Solution.

(a) Using the theory we learned, it holds

$$[\text{RGA}(s)]_{11} = [\text{RGA}(s)]_{22} = \frac{1}{ps^2 + 2ps + 1}$$

and

$$\begin{aligned} [\text{RGA}(s)]_{12} &= [\text{RGA}(s)]_{21} \\ &= 1 - [\text{RGA}(s)]_{11} \\ &= 1 - \frac{1}{ps^2 + 2ps + 1} \\ &= \frac{ps \cdot (s + 2)}{ps^2 + 2ps + 1}. \end{aligned}$$

(b) In order to use two independent SISO control loops, the diagonal elements of the RGA matrix should be ≈ 1 and the anti diagonal elements should be ≈ 0 . It's easy to see that this is the case for $p = 0$. In fact, if one sets $p = 0$ one gets

$$\text{RGA}(s) = \mathbb{1}.$$

Hence, independently of the frequency one has, i.e. $\omega \in [0, \infty)$, the control problem can be solved with two independent SISO controllers.

(c) Using the learned theory, it holds

$$\begin{aligned} [\text{RGA}(s)]_{11} &= [\text{RGA}(s)]_{22} \\ &= \frac{P_{11}(s) \cdot P_{22}(s)}{P_{11}(s) \cdot P_{22}(s) - P_{12}(s) \cdot P_{21}(s)} \\ &= \frac{-\frac{1}{s \cdot (s+1)}}{-\frac{1}{s \cdot (s+1)} - \frac{s+2}{s+1}} \\ &= \frac{1}{1 + s \cdot (s + 2)} \\ &= \frac{1}{s^2 + 2s + 1} \\ &= \frac{1}{(s + 1)^2}. \end{aligned}$$

and

$$\begin{aligned} [\text{RGA}(s)]_{12} &= [\text{RGA}(s)]_{21} \\ &= 1 - [\text{RGA}(s)]_{11} \\ &= 1 - \frac{1}{(s + 1)^2} \\ &= \frac{s \cdot (s + 2)}{(s + 1)^2}. \end{aligned}$$

(d) In order to evaluate the RGA matrix in this range, we have to express it with its frequency dependence, i.e. $s = j\omega$. For the magnitudes it holds

$$\begin{aligned} |[\text{RGA}(j\omega)]_{11}| &= |[\text{RGA}(j\omega)]_{22}| \\ &= \frac{1}{|j\omega + 1|^2} \\ &= \frac{1}{1 + \omega^2}. \end{aligned}$$

and

$$\begin{aligned}
 |[\text{RGA}(j\omega)]_{12}| &= |[\text{RGA}(j\omega)]_{21}| \\
 &= \frac{1}{|j\omega + 1|^2} \cdot |j\omega| \cdot |j\omega + 2| \\
 &= \frac{\omega \cdot \sqrt{4 + \omega^2}}{1 + \omega^2}.
 \end{aligned}$$

We can now insert the two limit values of the given range and get

$$\begin{aligned}
 |[\text{RGA}(j \cdot 3)]_{11}| &= |[\text{RGA}(j \cdot 3)]_{22}| \\
 &= \frac{1}{10} \\
 &= 0.10. \\
 |[\text{RGA}(j \cdot 3)]_{12}| &= |[\text{RGA}(j \cdot 3)]_{21}| \\
 &= \frac{3 \cdot \sqrt{13}}{10} \\
 &\approx 1.08.
 \end{aligned}$$

and

$$\begin{aligned}
 |[\text{RGA}(j \cdot 10)]_{11}| &= |[\text{RGA}(j \cdot 10)]_{22}| \\
 &= \frac{1}{101} \\
 &= 0.01. \\
 |[\text{RGA}(j \cdot 10)]_{12}| &= |[\text{RGA}(j \cdot 10)]_{21}| \\
 &= \frac{10 \cdot \sqrt{104}}{101} \\
 &\approx 1.01.
 \end{aligned}$$

In both cases the diagonal elements are close to 0 and the antidiagonal elements are close to 1. This means that the system is **diagonal dominant** and SISO control *one loop at time* is permitted. We just need to pay attention to what should be controlled: since the antidiagonal elements are close to 1, we need to use u_1 for y_2 and u_2 for y_1 .

Example 21. Figure 34 shows a 2×2 MIMO system. Sadly, we don't know anything about the transfer functions $P_{ij}(s)$ but

$$P_{12}(s) = 0.$$

Your boss wants you to use a *one loop at the time* approach as you see in the picture.

- Why is your boss' suggestion correct?
- Just a reference r_i is affecting both outputs y_i , which one?
- Compute the transfer function $r_i \rightarrow y_j$ for $i \neq j$?

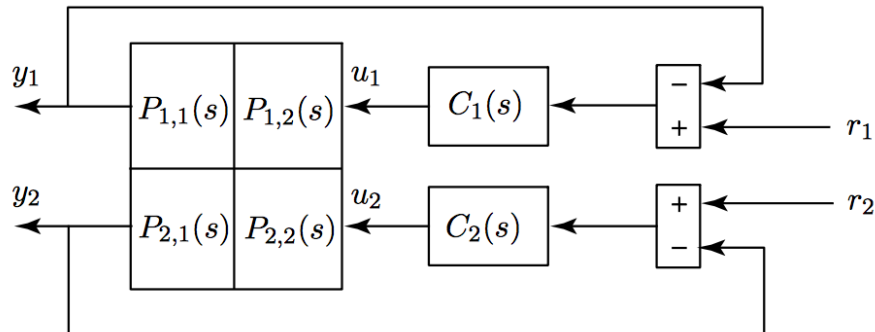


Figure 34: Structure of MIMO system.

Solution.

- (a) To check if the suggestion is correct let's have a look at the RGA matrix: it holds

$$\begin{aligned}
 [\text{RGA}]_{11} &= [\text{RGA}]_{22} \\
 &= \frac{P_{11}(s) \cdot P_{22}(s)}{P_{11}(s) \cdot P_{22}(s) - P_{12}(s) \cdot P_{21}(s)} \\
 &= 1. \\
 [\text{RGA}]_{12} &= [\text{RGA}]_{21} \\
 &= 1 - [\text{RGA}]_{11} \\
 &= 0.
 \end{aligned}$$

since $P_{12}(s) = 0$. This means that the RGA matrix is identical to the identity matrix, resulting in a perfect diagonal dominant system, which can be controlled with the *one loop at the time* approach.

- (b) Let's analyze the signals from Figure 34. Since $P_{12}(s) = 0$, the output y_1 is not affected from u_2 . Moreover, this means that the reference signal r_2 , which influences u_2 , cannot affect the output y_1 . The only reference that acts on both y_1 and y_2 is r_1 : directly through $C_1(s)$ on y_1 and with crosscouplings through $P_{21}(s)$ on y_2 .
- (c) As usual we set to 0 the reference values we don't analyze: here $r_2 = 0$. Starting from the general equation in frequency domain

$$\begin{aligned}
 \begin{pmatrix} Y_1(s) \\ Y_2(s) \end{pmatrix} &= \begin{pmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{pmatrix} \cdot \begin{pmatrix} U_1(s) \\ U_2(s) \end{pmatrix} \\
 &= \begin{pmatrix} P_{11}(s) & 0 \\ P_{21}(s) & P_{22}(s) \end{pmatrix} \cdot \begin{pmatrix} U_1(s) \\ U_2(s) \end{pmatrix}.
 \end{aligned}$$

one can read

$$\begin{aligned}
 Y_1(s) &= P_{11}(s) \cdot U_1(s) \\
 Y_2(s) &= P_{21}(s) \cdot U_1(s) + P_{22}(s) \cdot U_2(s).
 \end{aligned}$$

Since we want to relate r_1 and y_2 let's express u_1 as something we know. Using Figure 34 one gets

$$\begin{aligned}
 R_1(s) \cdot C_1(s) &= U_1(s) + P_{11}(s) \cdot C_1(s) \cdot U_1(s) \\
 U_1 &= \frac{R_1(s) \cdot C_1(s)}{1 + P_{11}(s) \cdot C_1(s)}.
 \end{aligned}$$

Inserting this into the second equation one gets

$$Y_2(s) = P_{21}(s) \cdot \frac{R_1(s) \cdot C_1(s)}{1 + P_{11}(s) \cdot C_1(s)} + P_{22}(s) \cdot U_2(s).$$

One have to find an expression for $U_2(s)$. To do that, we look at the second loop in Figure 34 an see

$$\underbrace{R_2(s) \cdot C_2(s)}_{=0} - Y_2(s) \cdot C_2(s) = U_2(s)$$

$$U_2(s) = -Y_2(s) \cdot C_2(s).$$

Inserting this into the second equation one gets

$$\begin{aligned}
 Y_2(s) &= P_{21}(s) \cdot \frac{R_1(s) \cdot C_1(s)}{1 + P_{11}(s) \cdot C_1(s)} + P_{22}(s) \cdot U_2(s) \\
 &= P_{21}(s) \cdot \frac{R_1(s) \cdot C_1(s)}{1 + P_{11}(s) \cdot C_1(s)} + P_{22}(s) \cdot (-Y_2(s) \cdot C_2(s)) \\
 Y_2(s) \cdot (1 + P_{22}(s) \cdot C_2(s)) &= P_{21}(s) \cdot \frac{R_1(s) \cdot C_1(s)}{1 + P_{11}(s) \cdot C_1(s)} \\
 Y_2(s) &= \underbrace{\frac{P_{21}(s) \cdot C_1(s)}{(1 + P_{11}(s) \cdot C_1(s)) \cdot (1 + P_{22}(s) \cdot C_2(s))}}_{F(s)} \cdot R_1(s).
 \end{aligned}$$

where $F(s)$ is the transfer function we wanted.

Example 22. Figure 35 shows the structure of a MIMO system, composed of three subsystems $P_1(s)$, $P_2(s)$ and $P_3(s)$. It has inputs u_1 and u_2 and outputs y_1 and y_2 . The three subsystems

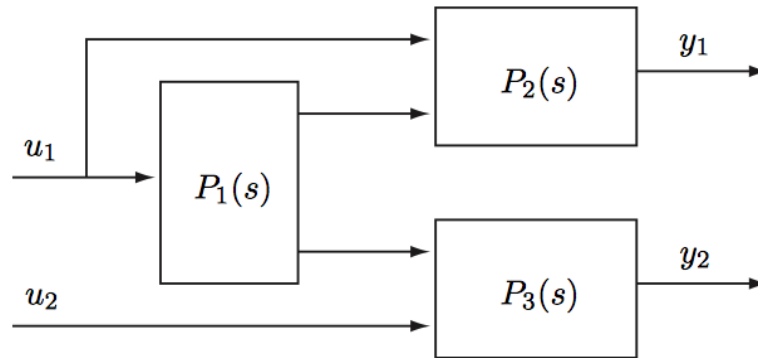


Figure 35: Structure of MIMO system.

are given as

$$P_1(s) = \begin{pmatrix} \frac{s-5}{s+3} \\ \frac{1}{s+4} \end{pmatrix}, \quad P_2(s) = \begin{pmatrix} \frac{1}{s+3} & \frac{s+4}{s-5} \end{pmatrix}, \quad P_3(s) = \begin{pmatrix} \frac{s+2}{s+5} & \frac{1}{s+1} \end{pmatrix}.$$

Compute the transfer function of the whole system.

Solution. One should think with matrix dimensions here. Let's redefine the subsystem's matrices more generally:

$$P_1(s) = \begin{pmatrix} P_1^{11} \\ P_1^{21} \end{pmatrix}, \quad P_2(s) = \begin{pmatrix} P_2^{11} & P_2^{12} \end{pmatrix}, \quad P_3(s) = \begin{pmatrix} P_3^{11} & P_3^{12} \end{pmatrix}$$

Together with the structure of the system one gets

$$\begin{aligned} Y_1 &= P_2^{11} \cdot U_1 + P_2^{12} \cdot P_1^{11} \cdot U_1, \\ Y_2 &= P_3^{11} \cdot P_1^{21} \cdot U_1 + P_3^{12} \cdot U_2. \end{aligned}$$

This can be written in the general matrix for the transfer function:

$$\begin{aligned} P(s) &= \begin{pmatrix} P_2^{11} + P_2^{12} \cdot P_1^{11} & 0 \\ P_3^{11} \cdot P_1^{21} & P_3^{12} \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{s+3} + \frac{s+4}{s-5} \cdot \frac{s-5}{s+3} & 0 \\ \frac{s+2}{s+5} \cdot \frac{1}{s+4} & \frac{1}{s+1} \end{pmatrix} \\ &= \begin{pmatrix} \frac{s+5}{s+3} & 0 \\ \frac{s+2}{(s+5) \cdot (s+4)} & \frac{1}{s+1} \end{pmatrix}. \end{aligned}$$

Example 23. Figure 36 shows the structure of a MIMO system, composed of two subsystems $P_1(s), P_2(s)$. It has inputs u_1 and u_2 and outputs y_1 and y_2 . The subsystem $P_1(s)$ is given with

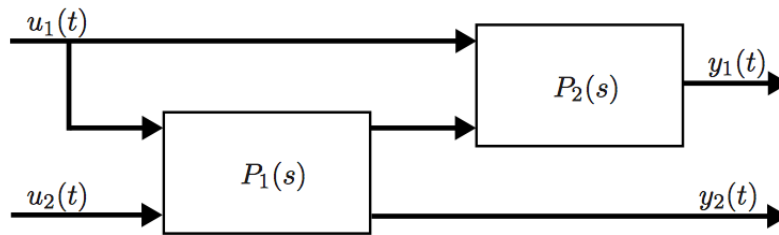


Figure 36: Structure of MIMO system.

its state space description:

$$A_1 = \begin{pmatrix} -3 & 0 \\ 2 & 1 \end{pmatrix}, \quad B_1 = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}, \quad C_1 = \begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix}, \quad D_1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}.$$

and the subsystem $P_2(s)$ is given as

$$P_2(s) = \begin{pmatrix} \frac{1}{s-2} & \frac{s-1}{(s+4) \cdot (s-2)} \end{pmatrix}.$$

Compute the transfer function of the whole system.

Solution. First of all, we compute the transfer function in frequency domain of the first subsystem $P_1(s)$. It holds

$$\begin{aligned}
 P_1(s) &= C_1 \cdot (s \cdot \mathbb{1} - A_1)^{-1} \cdot B_1 + D_1 \\
 &= \begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix} \cdot \begin{pmatrix} s+3 & 0 \\ -2 & s-1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \\
 &= \begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix} \cdot \frac{1}{(s+3) \cdot (s-1)} \cdot \begin{pmatrix} s-1 & 0 \\ 2 & s+3 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \\
 &= \frac{1}{(s+3) \cdot (s-1)} \cdot \begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix} \cdot \begin{pmatrix} s-1 & 0 \\ 2s+8 & s+3 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \\
 &= \frac{1}{(s+3) \cdot (s-1)} \cdot \begin{pmatrix} 4s+16 & 2s+6 \\ 5s+5 & s+3 \end{pmatrix} \\
 &= \begin{pmatrix} \frac{4s+16}{(s+3) \cdot (s-1)} & \frac{2}{s-1} \\ \frac{5s+5}{(s+3) \cdot (s-1)} & \frac{1}{s-1} \end{pmatrix}.
 \end{aligned}$$

One should think with matrix dimensions here. Let's redefine the subsystem's matrices more generally:

$$P_1(s) = \begin{pmatrix} P_1^{11} & P_1^{12} \\ P_1^{21} & P_1^{22} \end{pmatrix}, \quad P_2(s) = \begin{pmatrix} P_2^{11} & P_2^{12} \\ P_2^{21} & P_2^{22} \end{pmatrix}.$$

Together with the structure of the system one gets

$$\begin{aligned}
 Y_1 &= P_2^{11} \cdot U_1 + P_1^{11} \cdot P_2^{12} \cdot U_1 + P_1^{12} \cdot P_2^{12} \cdot U_2, \\
 Y_2 &= P_1^{21} \cdot U_1 + P_1^{22} \cdot U_2.
 \end{aligned}$$

This can be written in the general matrix for the transfer function:

$$\begin{aligned}
 P(s) &= \begin{pmatrix} P_2^{11} + P_1^{11} \cdot P_2^{12} & P_1^{12} \cdot P_2^{12} \\ P_1^{21} & P_1^{22} \end{pmatrix} \\
 &= \dots \\
 &= \begin{pmatrix} \frac{s+7}{(s+3) \cdot (s-2)} & \frac{s+1}{(s+4) \cdot (s-2)} \\ \frac{5s+5}{(s+3) \cdot (s-1)} & \frac{1}{s-1} \end{pmatrix}.
 \end{aligned}$$

Example 24. The system in Figure 37 can be well controlled with two separate SISO controllers.

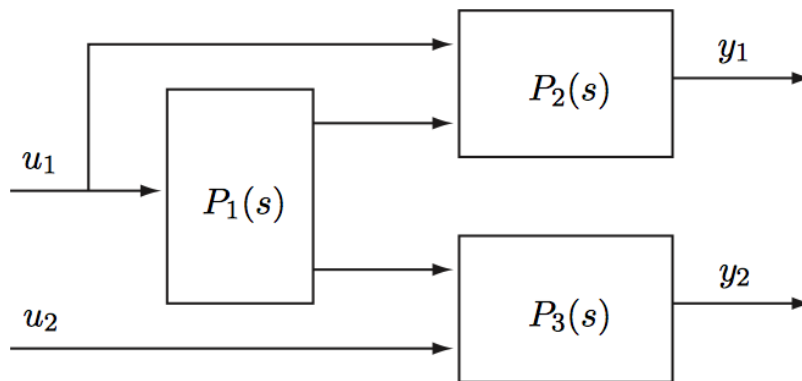


Figure 37: Structure of MIMO system.

☐ True.

☐ False.

Solution.☒ True.☐ False.**Explanation:**

One can observe that the input u_2 affects only the output y_2 . This means that the transfer function matrix has a **triangular** form and hence, that the RGA matrix is identical to the identity matrix: this means that we can reach good control with two separate SISO controllers.

5 Frequency Response of MIMO Systems

5.1 Singular Value Decomposition (SVD)

The Singular Value Decomposition plays a central role in MIMO frequency response analysis. Let's recall some concepts from the course *Lineare Algebra I/II*:

5.1.1 Preliminary Definitions

The **induced** norm $\|A\|$ of a matrix that describes a linear function like

$$y = A \cdot u \quad (5.1)$$

is defined as

$$\begin{aligned} \|A\| &= \max_{u \neq 0} \frac{\|y\|}{\|u\|} \\ &= \max_{\|u\|=1} \|y\|. \end{aligned} \quad (5.2)$$

Remark. In the course *Lineare Algebra I/II* you learned that there are many different norms and the expression $\|A\|$ could look too generic. However, in this course we will always use the euclidean norm $\|A\| = \|A\|_2$. This norm is defined as

$$\begin{aligned} \|A\|_2 &= \sqrt{\mu_{\max}} \\ &= \sqrt{\text{maximal eigenvalue of } A^* \cdot A} \end{aligned} \quad (5.3)$$

where A^* is the **conjugate transpose** or **Hermitian transpose** of the matrix A . This is defined as

$$A^* = (\text{conj}(A))^T. \quad (5.4)$$

Example 25. Let

$$A = \begin{pmatrix} 1 & -2-i \\ 1+i & i \end{pmatrix}.$$

Then it holds

$$\begin{aligned} A^* &= (\text{conj}(A))^T \\ &= \begin{pmatrix} 1 & -2+i \\ 1-i & -i \end{pmatrix}^T \\ &= \begin{pmatrix} 1 & 1-i \\ -2+i & -i \end{pmatrix}. \end{aligned}$$

Remark. If we deal with $A \in \mathbb{R}^{n \times m}$ holds of course

$$A^* = A^T. \quad (5.5)$$

One can list a few useful properties of the euclidean norm:

- (i) Remember: $A^T \cdot A$ is always a quadratic matrix!
- (ii) If A is orthogonal:

$$\|A\|_2 = 1. \quad (5.6)$$

(iii) If A is symmetric:

$$\|A\|_2 = \max(|e_i|). \quad (5.7)$$

(iv) If A is invertible:

$$\begin{aligned} \|A^{-1}\|_2 &= \frac{1}{\sqrt{\mu_{\min}}} \\ &= \frac{1}{\sqrt{\text{minimal eigenvalue of } A^* \cdot A}}. \end{aligned} \quad (5.8)$$

(v) If A is invertible and symmetric:

$$\|A^{-1}\|_2 = \frac{1}{\min(|e_i|)}. \quad (5.9)$$

where e_i are the eigenvalues of A .

In order to define the SVD we have to go a step further. Let's consider a Matrix A and the linear function 5.1. It holds

$$\begin{aligned} \|A\|^2 &= \max_{\|u\|=1} y^* \cdot y \\ &= \max_{\|u\|=1} (A \cdot u)^* \cdot (A \cdot u) \\ &= \max_{\|u\|=1} u^* \cdot A^* \cdot A \cdot u \\ &= \max_i \mu(A^* \cdot A) \\ &= \max_i \sigma_i^2. \end{aligned} \quad (5.10)$$

where σ_i are the **singular values** of matrix A . They are defined as

$$\sigma_i = \sqrt{\mu_i} \quad (5.11)$$

where μ_i are the eigenvalues of $A^* \cdot A$.

Combining 5.2 and 5.11 one gets

$$\sigma_{\min}(A) \leq \frac{\|y\|}{\|u\|} \leq \sigma_{\max}(A). \quad (5.12)$$

5.1.2 Singular Value Decomposition

Our goal is to write a general matrix $A \in \mathbb{C}^{p \times m}$ as product of three matrices: U , Σ and V . It holds

$$A = U \cdot \Sigma \cdot V^* \text{ with } U \in \mathbb{C}^{p \times p}, \Sigma \in \mathbb{R}^{p \times m}, V \in \mathbb{C}^{m \times m}. \quad (5.13)$$

Remark. U and V are orthogonal, Σ is a diagonal matrix.

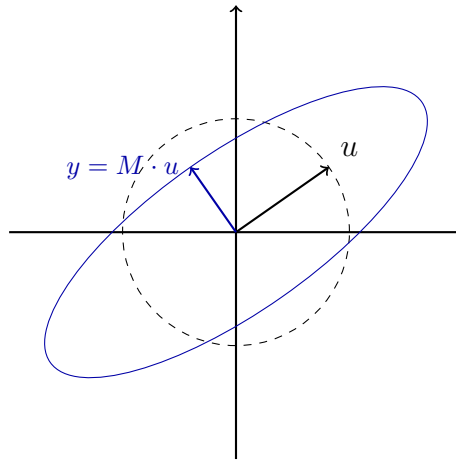


Figure 38: Illustration of the singular values.

Kochrezept:

Let $A \in \mathbb{C}^{p \times m}$ be given:

- (I) Compute all the eigenvalues and eigenvectors of the matrix

$$A^* \cdot A \in \mathbb{C}^{m \times m}.$$

and sort them as

$$\mu_1 \geq \mu_2 \geq \dots \geq \mu_r > \mu_{r+1} = \dots = \mu_m = 0 \quad (5.14)$$

- (II) Compute an orthogonal basis from the eigenvectors v_i and write it in a matrix as

$$V = (v_1 \dots v_m) \in \mathbb{C}^{m \times m}. \quad (5.15)$$

- (III) We have already found the singular values: they are defined as

$$\sigma_i = \sqrt{\mu_i} \text{ for } i = 1, \dots, \min\{p, m\}. \quad (5.16)$$

We can then write Σ as

$$\Sigma = \begin{pmatrix} \sigma_1 & & 0 & \dots & 0 \\ & \ddots & \vdots & & \vdots \\ & & \sigma_m & 0 & \dots & 0 \end{pmatrix} \in \mathbb{R}^{p \times m}, \quad p < m \quad (5.17)$$

$$\Sigma = \begin{pmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_n & \\ 0 & \dots & 0 & \\ \vdots & & \vdots & \\ 0 & \dots & 0 & \end{pmatrix} \in \mathbb{R}^{p \times m}, \quad p > m. \quad (5.18)$$

- (IV) One finds u_1, \dots, u_r from

$$u_i = \frac{1}{\sigma_i} \cdot A \cdot v_i \text{ for all } i = 1, \dots, r \text{ (for } \sigma_i \neq 0) \quad (5.19)$$

(V) If $r < p$ one has to complete the basis u_1, \dots, u_r (with ONB) to obtain an orthogonal basis, with U orthogonal.

(VI) If you followed the previous steps, you can write

$$A = U \cdot \Sigma \cdot V^*.$$

Motivation for the computation of Σ , U und V .

$$\begin{aligned} A^* \cdot A &= (U \cdot \Sigma \cdot V^*)^* \cdot (U \cdot \Sigma \cdot V^*) \\ &= V \cdot \Sigma^* \cdot U^* \cdot U \cdot \Sigma \cdot V^* \\ &= V \cdot \Sigma^* \cdot \Sigma \cdot V^* \\ &= V \cdot \Sigma^2 \cdot V^*. \end{aligned} \tag{5.20}$$

This is nothing else than the **diagonalization** of the matrix $A^* \cdot A$. The columns of V are the eigenvectors of $A^* \cdot A$ and the σ_i^2 the eigenvalues.

For U :

$$\begin{aligned} A \cdot A^* &= (U \cdot \Sigma \cdot V^*) \cdot (U \cdot \Sigma \cdot V^*)^* \\ &= U \cdot \Sigma \cdot V^* \cdot V \cdot \Sigma \cdot U^* \\ &= U \cdot \Sigma^* \cdot \Sigma \cdot U^* \\ &= U \cdot \Sigma^2 \cdot U^*. \end{aligned} \tag{5.21}$$

This is nothing else than the **diagonalization** of the matrix $A \cdot A^*$. The columns of U are the eigenvectors of $A \cdot A^*$ and the σ_i^2 the eigenvalues.

Remark. In order to derive the previous two equations I used that:

- The matrix $A^* \cdot A$ is symmetric, i.e.

$$\begin{aligned} (A^* \cdot A)^* &= A^* \cdot (A^*)^* \\ &= A^* \cdot A. \end{aligned}$$

- $U^{-1} = U^*$ (because U is orthogonal).
- $V^{-1} = V^*$ (because V is orthogonal).

Remark. Since the matrix $A^* \cdot A$ is always symmetric and positive semidefinite, the singular values are always real numbers.

Remark. The MATLAB command for the singular value decomposition is

`[U,S,V]=svd`

One can write A^T as `A.'=transpose(A)` and A^* as `A'=conj(transpose(A))`. Those two are equivalent for real numbers.

Remark. Although I've reported detailed informations about the calculation of U and V , this **won't be relevant for the exam**. It is however good and useful to know the reasons that are behind this topic.

Example 26. Let u be

$$u = \begin{pmatrix} \cos(x) \\ \sin(x) \end{pmatrix}$$

with $\|u\| = 1$. The matrix M is given as

$$M = \begin{pmatrix} 2 & 0 \\ 0 & \frac{1}{2} \end{pmatrix}.$$

We know that the product of M and u defines a linear function

$$\begin{aligned} y &= M \cdot u \\ &= \begin{pmatrix} 2 & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \cdot \begin{pmatrix} \cos(x) \\ \sin(x) \end{pmatrix} \\ &= \begin{pmatrix} 2 \cdot \cos(x) \\ \frac{1}{2} \cdot \sin(x) \end{pmatrix}. \end{aligned}$$

We need the maximum of $\|y\|$. In order to avoid square roots, one can use that the x that maximizes $\|y\|$ should also maximize $\|y\|^2$.

$$\|y\|^2 = 4 \cdot \cos^2(x) + \frac{1}{4} \cdot \sin^2(x)$$

has maximum

$$\begin{aligned} \frac{d\|y\|^2}{dx} &= -8 \cdot \cos(x) \cdot \sin(x) + \frac{1}{2} \cdot \sin(x) \cdot \cos(x) \stackrel{!}{=} 0 \\ \Rightarrow x_{\max} &= \left\{ 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2} \right\}. \end{aligned}$$

Inserting back for the maximal $\|y\|$ one gets:

$$\|y\|_{\max} = 2, \quad \|y\|_{\min} = \frac{1}{2}.$$

The singular values can be calculated with $M^* \cdot M$:

$$\begin{aligned} M^* \cdot M &= M^T \cdot M \\ &= \begin{pmatrix} 4 & 0 \\ 0 & \frac{1}{4} \end{pmatrix} \Rightarrow \lambda_i = \left\{ 4, \frac{1}{4} \right\} \Rightarrow \sigma_i = \left\{ 2, \frac{1}{2} \right\}. \end{aligned}$$

As stated before, one can see that $\|y\| \in [\sigma_{\min}, \sigma_{\max}]$. The matrix U has eigenvectors of $M \cdot M^T$ as columns and the matrix V has eigenvectors of $M^T \cdot M$ as columns.

In this case

$$M \cdot M^T = M^T \cdot M,$$

hence the two matrices are equal. Since their product is a diagonal matrix one should recall from the theory that the eigenvectors are easy to determine: they are nothing else than the standard basis vectors. This means

$$U = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}, \quad V = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Interpretation:

Referring to Figure 39, let's interpret these calculations. One can see that the maximal *amplification* occurs at $v = V(:, 1)$ and has direction $u = U(:, 1)$, i.e. the vector u is doubled (σ_{\max}). The minimal *amplification* occurs at $v = V(:, 2)$ and has direction $u = U(:, 2)$, i.e. the vector u is halved (σ_{\min}).

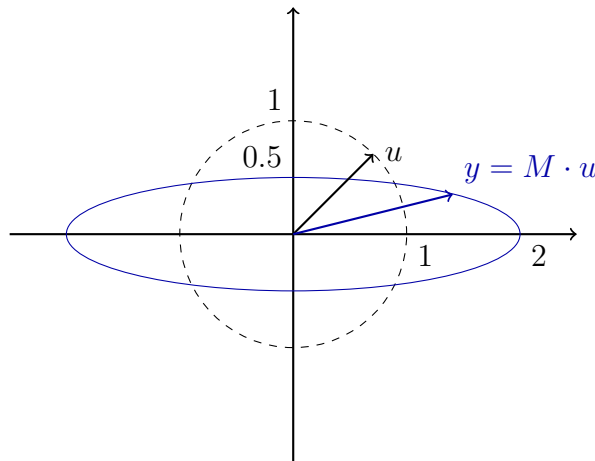


Figure 39: Illustration von der Singularwertzerlegung.

Example 27. Let

$$A = \begin{pmatrix} -3 & 0 \\ 0 & 3 \\ \sqrt{3} & 2 \end{pmatrix}$$

be given.

Question: Find the singular values of A and write down the matrix Σ .

Solution. Let's compute $A^T A$:

$$A^T A = \begin{pmatrix} -3 & 0 & \sqrt{3} \\ 0 & 3 & 2 \end{pmatrix} \cdot \begin{pmatrix} -3 & 0 \\ 0 & 3 \\ \sqrt{3} & 2 \end{pmatrix} = \begin{pmatrix} 12 & 2\sqrt{3} \\ 2\sqrt{3} & 13 \end{pmatrix}$$

One can see easily that the eigenvalues are

$$\lambda_1 = 16, \lambda_2 = 9.$$

The singular values are

$$\sigma_1 = 4, \sigma_2 = 3.$$

One writes in this case

$$\Sigma = \begin{pmatrix} 4 & 0 \\ 0 & 3 \\ 0 & 0 \end{pmatrix}.$$

Example 28. A transfer function $G(s)$ is given as

$$\begin{pmatrix} \frac{1}{s+3} & \frac{s+1}{s+3} \\ \frac{s+1}{s+3} & \frac{1}{s+3} \end{pmatrix}$$

Find the singular values of $G(s)$ at $\omega = 1 \frac{\text{rad}}{s}$.

Solution. The transfer function $G(s)$ evaluated at $\omega = 1 \frac{\text{rad}}{s}$ has the form

$$G(j) = \begin{pmatrix} \frac{1}{j+3} & \frac{j+1}{j+3} \\ \frac{j+1}{j+3} & \frac{1}{j+3} \end{pmatrix}$$

In order to calculate the singular values, we have to compute the eigenvalues of $H = G^* \cdot G$:

$$\begin{aligned} H &= G^* \cdot G \\ &= \begin{pmatrix} \frac{1}{-j+3} & \frac{-j+1}{-j+3} \\ \frac{-j+1}{-j+3} & \frac{1}{-j+3} \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{j+3} & \frac{j+1}{j+3} \\ \frac{j+1}{j+3} & \frac{1}{j+3} \end{pmatrix} \\ &= \begin{pmatrix} \frac{3}{10} & \frac{2}{10} \\ \frac{2}{10} & \frac{3}{10} \end{pmatrix} \\ &= \frac{1}{10} \cdot \begin{pmatrix} 3 & 2 \\ 2 & 3 \end{pmatrix}. \end{aligned}$$

For the eigenvalues it holds

$$\begin{aligned} \det(H - \lambda \cdot \mathbb{1}) &= \det \begin{pmatrix} \frac{3}{10} - \lambda & \frac{2}{10} \\ \frac{2}{10} & \frac{3}{10} - \lambda \end{pmatrix} \\ &= \left(\frac{3}{10} - \lambda \right)^2 - \left(-\frac{2}{10} \right)^2 \\ &= \lambda^2 - \frac{6}{10}\lambda + \frac{5}{100} \\ &= \left(\lambda - \frac{1}{10} \right) \cdot \left(\lambda - \frac{5}{10} \right). \end{aligned}$$

It follows

$$\begin{aligned} \lambda_1 &= \frac{1}{10} \\ \lambda_2 &= \frac{5}{10} \end{aligned}$$

and so

$$\begin{aligned} \sigma_1 &= \sqrt{\frac{1}{10}} \\ &\approx 0.3162. \\ \sigma_2 &= \sqrt{\frac{5}{10}} \\ &\approx 0.7071. \end{aligned}$$

Example 29. Let be

$$A = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}$$

$$B = \begin{pmatrix} j & 1 \end{pmatrix}.$$

Find the singular values of the two matrices.

Solution.

- Let's begin with matrix A . It holds

$$\begin{aligned} H &= A^* \cdot A \\ &= \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 2 \\ 2 & 5 \end{pmatrix}. \end{aligned}$$

In order to find the eigenvalues of H we compute

$$\begin{aligned} \det(H - \lambda \cdot \mathbb{1}) &= \det \begin{pmatrix} 1 - \lambda & 2 \\ 2 & 5 - \lambda \end{pmatrix} \\ &= (1 - \lambda) \cdot (5 - \lambda) - 4 \\ &= \lambda^2 - 6\lambda + 1. \end{aligned}$$

This means that the eigenvalues are

$$\begin{aligned} \lambda_1 &= 3 + 2\sqrt{2} \\ \lambda_2 &= 3 - 2\sqrt{2}. \end{aligned}$$

The singular values are then

$$\begin{aligned} \sigma_1 &\approx 2.4142 \\ \sigma_2 &\approx 0.4142. \end{aligned}$$

- Let's look at matrix B . It holds

$$\begin{aligned} F &= B^* \cdot B \\ &= \begin{pmatrix} -j \\ 1 \end{pmatrix} \cdot \begin{pmatrix} j & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & -j \\ j & 1 \end{pmatrix}. \end{aligned}$$

In order to find the eigenvalues of F we compute

$$\begin{aligned} \det(F - \lambda \cdot \mathbb{1}) &= \det \begin{pmatrix} 1 - \lambda & -j \\ j & 1 - \lambda \end{pmatrix} \\ &= (1 - \lambda)^2 - 1 \\ &= \lambda^2 - 2\lambda \\ &= \lambda \cdot (\lambda - 2). \end{aligned}$$

This means that the eigenvalues are

$$\begin{aligned} \lambda_1 &= 0 \\ \lambda_2 &= 2. \end{aligned}$$

The singular values are then

$$\begin{aligned} \sigma_1 &= 0 \\ \sigma_2 &= \sqrt{2}. \end{aligned}$$

5.2 Frequency Responses

As we learned for SISO systems, if one excites a system with an *harmonic* signal

$$u(t) = h(t) \cdot \cos(\omega \cdot t), \quad (5.22)$$

the answer after a big amount of time is still an harmonic function with equal frequency ω :

$$y_\infty(t) = |P(j \cdot \omega)| \cos(\omega \cdot t + \angle(P(j \cdot \omega))). \quad (5.23)$$

One can generalize this and apply it to MIMO systems. With the assumption of $p = m$, i.e. equal number of inputs and outputs, one excite a system with

$$u(t) = \begin{pmatrix} \mu_1 \cdot \cos(\omega \cdot t + \phi_1) \\ \vdots \\ \mu_m \cdot \cos(\omega \cdot t + \phi_m) \end{pmatrix} \cdot h(t) \quad (5.24)$$

and get

$$y_\infty(t) = \begin{pmatrix} \nu_1 \cdot \cos(\omega \cdot t + \psi_1) \\ \vdots \\ \nu_m \cdot \cos(\omega \cdot t + \psi_m) \end{pmatrix}. \quad (5.25)$$

Let's define two diagonal matrices

$$\begin{aligned} \Phi &= \text{diag}(\phi_1, \dots, \phi_m) \in \mathbb{R}^{m \times m}, \\ \Psi &= \text{diag}(\psi_1, \dots, \psi_m) \in \mathbb{R}^{m \times m} \end{aligned} \quad (5.26)$$

and two vectors

$$\begin{aligned} \mu &= (\mu_1 \quad \dots \quad \mu_m)^T, \\ \nu &= (\nu_1 \quad \dots \quad \nu_m)^T. \end{aligned} \quad (5.27)$$

With these one can compute the Laplace Transform of the two signals as:

$$U(s) = e^{\frac{\Phi \cdot s}{\omega}} \cdot \mu \cdot \frac{s}{s^2 + \omega^2}. \quad (5.28)$$

and

$$Y(s) = e^{\frac{\Psi \cdot s}{\omega}} \cdot \nu \cdot \frac{s}{s^2 + \omega^2}. \quad (5.29)$$

With the general equation for a systems one gets

$$\begin{aligned} Y(s) &= P(s) \cdot U(s) \\ e^{\frac{\Psi \cdot s}{\omega}} \cdot \nu \cdot \frac{s}{s^2 + \omega^2} &= P(s) \cdot e^{\frac{\Phi \cdot s}{\omega}} \cdot \mu \cdot \frac{s}{s^2 + \omega^2} \\ e^{\frac{\Psi \cdot j \cdot \omega}{\omega}} \cdot \nu &= P(s) \cdot e^{\frac{\Phi \cdot j \cdot \omega}{\omega}} \cdot \mu \\ e^{\Psi \cdot j} \cdot \nu &= P(s) \cdot e^{\Phi \cdot j} \cdot \mu. \end{aligned} \quad (5.30)$$

We then recall that the **induced norm** for the matrix of a linear transformation $y = A \cdot u$ from 5.2. Here it holds

$$\begin{aligned} \|P(j \cdot \omega)\| &= \max_{e^{\Phi \cdot j} \cdot \mu \neq 0} \frac{\|e^{\Psi \cdot j} \cdot \nu\|}{\|e^{\Phi \cdot j} \cdot \mu\|} \\ &= \max_{\|e^{\Phi \cdot j} \cdot \mu\|=1} \|e^{\Psi \cdot j} \cdot \nu\|. \end{aligned} \quad (5.31)$$

Since

$$\|e^{\Phi \cdot j} \cdot \mu\| = \|\mu\| \quad (5.32)$$

and

$$\|e^{\Psi \cdot j} \cdot \nu\| = \|\nu\|. \quad (5.33)$$

One gets

$$\begin{aligned} \|P(j \cdot \omega)\| &= \max_{\mu \neq 0} \frac{\|\nu\|}{\|\mu\|} \\ &= \max_{\|\mu\|=1} \|\nu\|. \end{aligned} \quad (5.34)$$

Here one should get the feeling of why we introduced the singular value decomposition. From the theory we've learned, it is clear that

$$\sigma_{\min}(P(j \cdot \omega)) \leq \|\nu\| \leq \sigma_{\max}(P(j \cdot \omega)). \quad (5.35)$$

and if $\|\mu\| \neq 1$

$$\sigma_{\min}(P(j \cdot \omega)) \leq \frac{\|\nu\|}{\|\mu\|} \leq \sigma_{\max}(P(j \cdot \omega)). \quad (5.36)$$

with σ_i singular values of $P(j \cdot \omega)$. These two are worst case ranges and is important to notice that there is no exact formula for $\nu = f(\mu)$.

5.2.1 Maximal and minimal Gain

You are given a singular value decomposition

$$P(j \cdot \omega) = U \cdot \Sigma \cdot V^*. \quad (5.37)$$

One can read out from this decomposition several informations: the maximal/minial gain will be reached with an excitation in the direction of the column vectors of V . The response of the system will then be in the direction of the coulumn vectors of U .

Let's look at an example and try to understand how to use these informations:

Example 30. We consider a system with $m = 2$ inputs and $p = 3$ outputs. We are given its singular value decomposition at $\omega = 5 \frac{\text{rad}}{\text{s}}$:

$$\begin{aligned} \Sigma &= \begin{pmatrix} 0.4167 & 0 \\ 0 & 0.2631 \\ 0 & 0 \end{pmatrix}, \\ V &= \begin{pmatrix} 0.2908 & 0.9568 \\ 0.9443 - 0.1542 \cdot j & -0.2870 + 0.0469 \cdot j \end{pmatrix}, \\ U &= \begin{pmatrix} -0.0496 - 0.1680 \cdot j & 0.1767 - 0.6831 \cdot j & -0.6621 - 0.1820 \cdot j \\ 0.0146 - 0.9159 \cdot j & -0.1059 + 0.3510 \cdot j & -0.1624 + 0.0122 \cdot j \\ 0.0349 - 0.3593 \cdot j & 0.1360 - 0.5910 \cdot j & 0.6782 + 0.2048 \cdot j \end{pmatrix}. \end{aligned}$$

For the singular value $\sigma_{\max} = 0.4167$ the eigenvectors are $V(:, 1)$ and $U(:, 1)$:

$$\begin{aligned} V_1 &= \begin{pmatrix} 0.2908 \\ 0.9443 - 0.1542 \cdot j \end{pmatrix}, & |V_1| &= \begin{pmatrix} 0.2908 \\ 0.9568 \end{pmatrix}, & \angle(V_1) &= \begin{pmatrix} 0 \\ -0.1618 \end{pmatrix}, \\ U_1 &= \begin{pmatrix} -0.0496 - 0.1680 \cdot j \\ 0.0146 - 0.9159 \cdot j \\ 0.0349 - 0.3593 \cdot j \end{pmatrix}, & |U_1| &= \begin{pmatrix} 0.1752 \\ 0.9160 \\ 0.3609 \end{pmatrix}, & \angle(U_1) &= \begin{pmatrix} -1.8581 \\ -1.5548 \\ -1.4741 \end{pmatrix}. \end{aligned}$$

The maximal gain is then reached with

$$u(t) = \begin{pmatrix} 0.2908 \cdot \cos(5 \cdot t) \\ 0.9568 \cdot \cos(5 \cdot t - 0.1618) \end{pmatrix}.$$

The response of the system is then

$$y(t) = \sigma_{\max} \cdot \begin{pmatrix} 0.1752 \cdot \cos(5 \cdot t - 1.8581) \\ 0.9160 \cdot \cos(5 \cdot t - 1.5548) \\ 0.3609 \cdot \cos(5 \cdot t - 1.4741) \end{pmatrix} = 0.4167 \cdot \begin{pmatrix} 0.1752 \cdot \cos(5 \cdot t - 1.8581) \\ 0.9160 \cdot \cos(5 \cdot t - 1.5548) \\ 0.3609 \cdot \cos(5 \cdot t - 1.4741) \end{pmatrix}.$$

Since the three signals $y_1(t)$, $y_2(t)$ and $y_3(t)$ are not in phase, the maximal gain will never be reached. One can show that

$$\max_t \|y(t)\| \approx 0.4160 < 0.4167 = \sigma_{\max}$$

The reason for this difference stays in the phase deviation between $y_1(t)$, $y_2(t)$ and $y_3(t)$. The same analysis can be computed for σ_{\min} .

5.2.2 Robustness and Disturbance Rejection

Let's redefine the matrix norm $\|\cdot\|_{\infty}$ as

$$\|G(s)\|_{\infty} = \max_{\omega} (\max_i (\sigma_i(G(i \cdot \omega))). \quad (5.38)$$

Remark. It holds

$$\|G_1(s) \cdot G_2(s)\|_{\infty} \leq \|G_1(s)\|_{\infty} \cdot \|G_2(s)\|_{\infty}. \quad (5.39)$$

As we did for SISO systems, we can resume some important indicators for robustness and noise amplification:

	SISO	MIMO
Robustness	$\mu = \min_{\omega} (1 + L(j \cdot \omega))$	$\mu = \min_{\omega} (\sigma_{\min}(\mathbb{I} + L(j \cdot \omega)))$
Noise Amplification	$\ S\ _{\infty} = \max_{\omega} (S(j \cdot \omega))$	$\ S\ _{\infty} = \max_{\omega} (\sigma_{\max}(S(j \cdot \omega)))$

Example 31. Given the MIMO system

$$P(s) = \begin{pmatrix} \frac{1}{s+3} & \frac{1}{s+1} \\ \frac{1}{s+1} & \frac{3}{s+1} \end{pmatrix}.$$

Starting at $t = 0$, the system is excited with the following input signal:

$$u(t) = \begin{pmatrix} \cos(t) \\ \mu_2 \cos(t + \varphi_2) \end{pmatrix}.$$

Find the parameters φ_2 and μ_2 such that for steady-state conditions the output signal

$$\begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix}$$

has $y_1(t)$ equal to zero.

Solution. For a system excited using a harmonic input signal

$$u(t) = \begin{pmatrix} \mu_1 \cos(\omega t + \varphi_1) \\ \mu_2 \cos(\omega t + \varphi_2) \end{pmatrix}$$

the output signal $y(t)$, after a transient phase, will also be a harmonic signal and hence have the form

$$y(t) = \begin{pmatrix} \nu_1 \cos(\omega t + \psi_1) \\ \nu_2 \cos(\omega t + \psi_2) \end{pmatrix}.$$

As we have learned, it holds

$$e^{\Psi \cdot j} \cdot \nu = P(j\omega) \cdot e^{\Phi \cdot j} \cdot \mu.$$

One gets

$$\begin{pmatrix} e^{\psi_1 \cdot j} & 0 \\ 0 & e^{\psi_2 \cdot j} \end{pmatrix} \cdot \begin{pmatrix} \nu_1 \\ \nu_2 \end{pmatrix} = \begin{pmatrix} P_{11}(j\omega) & P_{12}(j\omega) \\ P_{21}(j\omega) & P_{22}(j\omega) \end{pmatrix} \cdot \begin{pmatrix} e^{\varphi_1 \cdot j} & 0 \\ 0 & e^{\varphi_2 \cdot j} \end{pmatrix} \cdot \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}.$$

For the first component one gets

$$e^{\psi_1 \cdot j} \cdot \nu_1 = P_{11}(j\omega) \cdot e^{\varphi_1 \cdot j} \cdot \mu_1 + P_{12}(j\omega) \cdot e^{\varphi_2 \cdot j} \cdot \mu_2.$$

For $y_1(t) = 0$ to hold we must have $\nu_1 = 0$. In the given case, some parameters can be easily copied from the signals:

$$\begin{aligned} \mu_1 &= 1 \\ \varphi_1 &= 0 \\ \omega &= 1. \end{aligned}$$

With the given transfer functions, one gets

$$\begin{aligned} 0 &= \frac{1}{j+3} + \mu_2 \cdot \frac{1}{j+1} \cdot e^{\varphi_2 \cdot j} \\ 0 &= \frac{3-j}{10} + \mu_2 \cdot \frac{1-j}{2} \cdot e^{\varphi_2 \cdot j} \\ 0 &= \frac{3-j}{10} + \mu_2 \cdot \frac{1-j}{2} \cdot (\cos(\varphi_2) + j \sin(\varphi_2)) \\ 0 &= \frac{3}{10} + \mu_2 \cdot \frac{1}{2} \cdot (\cos(\varphi_2) + \sin(\varphi_2)) + j \cdot \left(\mu_2 \cdot \frac{1}{2} \cdot (\sin(\varphi_2) - \cos(\varphi_2)) - \frac{1}{10} \right). \end{aligned}$$

Splitting the real to the imaginary part, one can get two equations that are easily solvable:

$$\begin{aligned} \mu_2 \cdot \frac{1}{2} \cdot (\cos(\varphi_2) + \sin(\varphi_2)) + \frac{3}{10} &= 0 \\ \mu_2 \cdot \frac{1}{2} \cdot (\sin(\varphi_2) - \cos(\varphi_2)) - \frac{1}{10} &= 0. \end{aligned}$$

Adding and subtracting the two equations one can reach two better equations:

$$\begin{aligned} \mu_2 \cdot \sin(\varphi_2) + \frac{1}{5} &= 0 \\ \mu_2 \cdot \cos(\varphi_2) + \frac{2}{5} &= 0. \end{aligned}$$

One of the solutions (periodicity) reads

$$\begin{aligned} \mu_2 &= \frac{1}{\sqrt{5}} \\ \varphi_2 &= \arctan\left(\frac{1}{2}\right) + \pi. \end{aligned}$$

Example 32. A 2×2 linear time invariant MIMO system with transfer function

$$P(s) = \begin{pmatrix} \frac{1}{s+1} & \frac{2}{s+1} \\ \frac{s^2+1}{s+10} & \frac{1}{s^2+2} \end{pmatrix}$$

is excited with the signal

$$u(t) = \begin{pmatrix} \mu_1 \cdot \cos(\omega \cdot t + \varphi_1) \\ \mu_2 \cdot \cos(\omega \cdot t + \varphi_2) \end{pmatrix}.$$

Because we bought a cheap signal generator, we cannot know exactly the constants $\mu_{1,2}$ and $\varphi_{1,2}$. A friend of you just found out with some measurements, that the excitation frequency is $\omega = 1 \frac{\text{rad}}{\text{s}}$. The cheap generator, cannot produce signals with magnitude of μ bigger than 10, i.e. $\sqrt{\mu_1^2 + \mu_2^2} \leq 10$. This works always at maximal power, i.e. at 10. Choose all possible responses of the system after infinite time.

☐ $y_\infty(t) = \begin{pmatrix} 5 \cdot \sin(t + 0.114) \\ \cos(t) \end{pmatrix}.$

☐ $y_\infty(t) = \begin{pmatrix} 5 \cdot \sin(t + 0.114) \\ \cos(2 \cdot t) \end{pmatrix}.$

☐ $y_\infty(t) = \begin{pmatrix} \sin(t + 0.542) \\ \sin(t + 0.459) \end{pmatrix}.$

☐ $y_\infty(t) = \begin{pmatrix} 19 \cdot \cos(t + 0.114) \\ \cos(t + 1.124) \end{pmatrix}.$

☐ $y_\infty(t) = \begin{pmatrix} 5 \cdot \cos(t + 0.114) \\ 5 \cdot \cos(t) \end{pmatrix}.$

☐ $y_\infty(t) = \begin{pmatrix} 10 \cdot \sin(t + 2.114) \\ 11 \cdot \sin(t + 1.234) \end{pmatrix}.$

Solution.

$$\checkmark y_{\infty}(t) = \begin{pmatrix} 5 \cdot \sin(t + 0.114) \\ \cos(t) \end{pmatrix}.$$

$$\square y_{\infty}(t) = \begin{pmatrix} 5 \cdot \sin(t + 0.114) \\ \cos(2 \cdot t) \end{pmatrix}.$$

$$\square y_{\infty}(t) = \begin{pmatrix} \sin(t + 0.542) \\ \sin(t + 0.459) \end{pmatrix}.$$

$$\square y_{\infty}(t) = \begin{pmatrix} 19 \cdot \cos(t + 0.114) \\ \cos(t + 1.124) \end{pmatrix}.$$

$$\checkmark y_{\infty}(t) = \begin{pmatrix} 5 \cdot \cos(t + 0.114) \\ 5 \cdot \cos(t) \end{pmatrix}.$$

$$\checkmark y_{\infty}(t) = \begin{pmatrix} 10 \cdot \sin(t + 2.114) \\ 11 \cdot \sin(t + 1.234) \end{pmatrix}.$$

Explanation

We have to compute the singular values of the matrix $P(j \cdot 1)$. These are

$$\begin{aligned} \sigma_{max} &= 1.8305 \\ \sigma_{min} &= 0.3863. \end{aligned}$$

With what we have learned it follows

$$10 \cdot \sigma_{min} = 3.863 \leq \|\nu\| \leq 18.305 = 10 \cdot \sigma_{max}.$$

The first response has $\|\nu\| = \sqrt{26}$ that is in this range. The second response also has $\|\nu\| = \sqrt{26}$ but the frequency in its second element changes and that isn't possible for linear systems. The third response has $\|\nu\| = \sqrt{2}$ that is too small to be in the range. The fourth response has $\|\nu\| = \sqrt{362}$ that is too big to be in the range. The fifth response has $\|\nu\| = \sqrt{50}$ that is in the range. The sixth response has $\|\nu\| = \sqrt{221}$ that is in the range.

Example 33. A 3×2 linear time invariant MIMO system is excited with the input

$$u(t) = \begin{pmatrix} 3 \cdot \sin(30 \cdot t) \\ 4 \cdot \cos(30 \cdot t) \end{pmatrix}.$$

You have forgot your PC and you don't know the transfer function of the system. Before coming to school, however, you have saved the MATLAB plot of the singular values of the system on your phone (see Figure 40). Choose all the possible responses of the system.

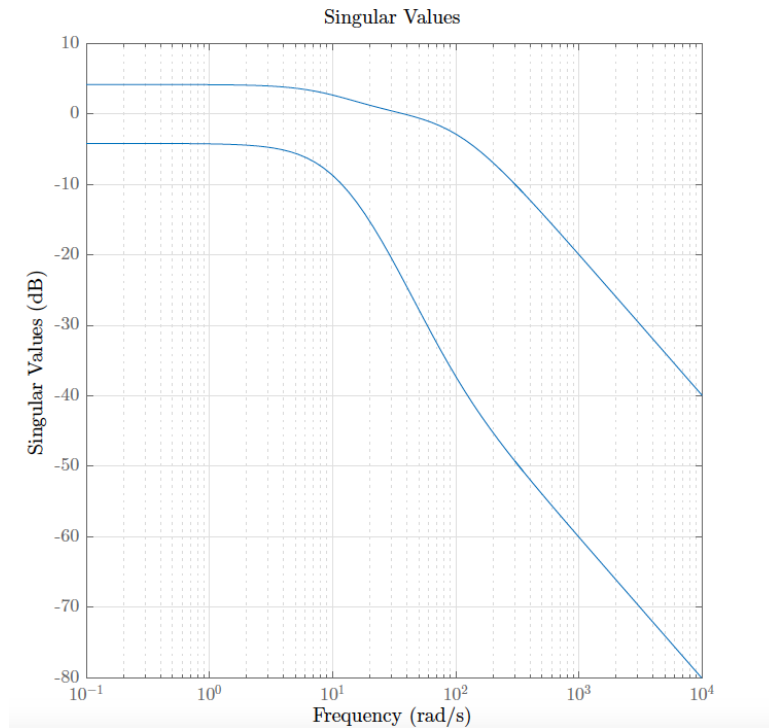


Figure 40: Singular values behaviour.

$$\square y_{\infty}(t) = \begin{pmatrix} 0.5 \cdot \sin(30 \cdot t + 0.314) \\ 0.5 \cdot \cos(30 \cdot t) \\ 0.5 \cdot \cos(30 \cdot t + 1) \end{pmatrix}.$$

$$\square y_{\infty}(t) = \begin{pmatrix} 4 \cdot \sin(30 \cdot t + 0.314) \\ 3 \cdot \cos(30 \cdot t) \\ 2 \cdot \cos(30 \cdot t + 1) \end{pmatrix}.$$

$$\square y_{\infty}(t) = \begin{pmatrix} 0.1 \cdot \sin(30 \cdot t + 0.314) \\ 0.1 \cdot \cos(30 \cdot t) \\ 0.1 \cdot \cos(30 \cdot t + 1) \end{pmatrix}.$$

$$\square y_{\infty}(t) = \begin{pmatrix} 0 \\ 4 \cdot \cos(30 \cdot t) \\ 2 \cdot \cos(30 \cdot t + 1) \end{pmatrix}.$$

$$\square y_{\infty}(t) = \begin{pmatrix} 2 \cdot \cos(30 \cdot t + 0.243) \\ 2 \cdot \cos(30 \cdot t + 0.142) \\ 2 \cdot \cos(30 \cdot t + 0.252) \end{pmatrix}.$$

Solution.

$$\checkmark y_{\infty}(t) = \begin{pmatrix} 0.5 \cdot \sin(30 \cdot t + 0.314) \\ 0.5 \cdot \cos(30 \cdot t) \\ 0.5 \cdot \cos(30 \cdot t + 1) \end{pmatrix}.$$

$$\square y_{\infty}(t) = \begin{pmatrix} 4 \cdot \sin(30 \cdot t + 0.314) \\ 3 \cdot \cos(30 \cdot t) \\ 2 \cdot \cos(30 \cdot t + 1) \end{pmatrix}.$$

$$\square y_{\infty}(t) = \begin{pmatrix} 0.1 \cdot \sin(30 \cdot t + 0.314) \\ 0.1 \cdot \cos(30 \cdot t) \\ 0.1 \cdot \cos(30 \cdot t + 1) \end{pmatrix}.$$

$$\checkmark y_{\infty}(t) = \begin{pmatrix} 0 \\ 4 \cdot \cos(30 \cdot t) \\ 2 \cdot \cos(30 \cdot t + 1) \end{pmatrix}.$$

$$\checkmark y_{\infty}(t) = \begin{pmatrix} 2 \cdot \cos(30 \cdot t + 0.243) \\ 2 \cdot \cos(30 \cdot t + 0.142) \\ 2 \cdot \cos(30 \cdot t + 0.252) \end{pmatrix}.$$

Explanation

From the given input one can read

$$\|\mu\| = \sqrt{3^2 + 4^2} = 5.$$

From the plot one can read at $\omega = 30 \frac{\text{rad}}{\text{s}}$ $\sigma_{\min} = 0.1$ and $\sigma_{\max} = 1$. It follows

$$5 \cdot \sigma_{\min} = 0.5 \leq \|\nu\| \leq 5 = 5 \cdot \sigma_{\max}.$$

The first response has $\|\nu\| = \sqrt{0.75}$ that is in the range. The second response has $\|\nu\| = \sqrt{29}$ that is too big to be in the range. The third response has $\|\nu\| = \sqrt{0.03}$ that is too small to be in the range. The fourth response has $\|\nu\| = \sqrt{20}$ that is in the range. The fifth response has $\|\nu\| = \sqrt{12}$ that is in the range.

6 Synthesis of MIMO Control Systems

We have learned that MIMO systems are more complicated than SISO systems and therefore their control as well. In fact, it is very difficult to apply intuitive concepts like loop shaping, that apply to SISO systems, because of the cross couplings/interactions between the inputs and the outputs. This means that systematic methods should be used, in order to achieve good control properties. We define two big groups of methods, that are used nowadays:

- \mathcal{H}_2 methods: Optimization problems in time domain which try to minimize the error signal over all frequencies.
- \mathcal{H}_∞ methods: Optimization problems in frequency domain which try to minimize the error signal over all frequencies.

6.1 The Linear Quadratic Regulator (LQR)

We first look at one \mathcal{H}_2 method: the linear quadratic regulator.

6.1.1 Definition

With the given state-space description

$$\frac{d}{dt}x(t) = A \cdot x(t) + B \cdot u(t), \quad A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m}, x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m \quad (6.1)$$

we are looking for a state-feedback controller

$$u(t) = f(x(t), t), \quad (6.2)$$

that brings $x(t)$ asymptotically to zero (with $x(0) \neq 0$). In other words it should hold.

$$\lim_{t \rightarrow \infty} x(t) = 0. \quad (6.3)$$

Remark. The input $u(t)$ we want to control depends only on $x(t)$ and not on other dynamic state variables.

In order to achieve the best control, we want to minimize the criterium

$$J(x, u) = \int_0^\infty \underbrace{(x^\top \cdot Q \cdot x)}_{\text{error}} + \underbrace{u^\top \cdot R \cdot u}_{\text{given energy}} dt, \quad (6.4)$$

where

$$Q \in \mathbb{R}^{n \times n} \quad (6.5)$$

is a *symmetric positive semidefinite* matrix and

$$R \in \mathbb{R}^{m \times m} \quad (6.6)$$

is a *symmetric positive definite matrix*. These two matrices represent the weights, i.e. the importance, of two things: how fast does the controller bring $x(t)$ to zero and how much energy should it use.

Remark. In the SISO case, we deal with scalars and things are *nicer*:

$$J(x, u) = \int_0^\infty (q \cdot x^2 + r \cdot u^2) dt, \quad (6.7)$$

Example 34. Let's try to visualize this: a criterion could look like

$$J(x, u) = \int_0^\infty (x_1^2 + 6 \cdot x_1 \cdot x_2 + 100 \cdot x_2^2 + 6 \cdot u_1^2 + 10 \cdot u_2^2) dt.$$

The matrices Q and R are

$$Q = \begin{pmatrix} 1 & 3 \\ 3 & 100 \end{pmatrix}$$

and

$$R = \begin{pmatrix} 6 & 0 \\ 0 & 10 \end{pmatrix}.$$

The solution¹⁰ $u(t) = f(x(t), t)$ (that minimizes (6.4)) is given as

$$u(t) = -K \cdot x(t), \quad (6.8)$$

with

$$K = R^{-1} \cdot B^\top \cdot \Phi. \quad (6.9)$$

where Φ is the solution of the Riccati equation

$$\Phi \cdot B \cdot R^{-1} \cdot B^\top \cdot \Phi - \Phi \cdot A - A^\top \cdot \Phi - Q = 0. \quad (6.10)$$

Remark. The actual procedure to determine this equation is not trivial. The main point you should be concerned with in this course is the following: we are trying to minimize the given criterium and we are dealing with matrices. This means that in the end, we calculate matrix derivations and minimize, obtaining matrix equations. If you are interested in the minimization procedure have a look here.

Remark. Don't forget what we are trying to do! This equations give us the optimal controller K : this means but not that it will be always the best one.

The feedback loop of such a system can be seen in Figure 41.

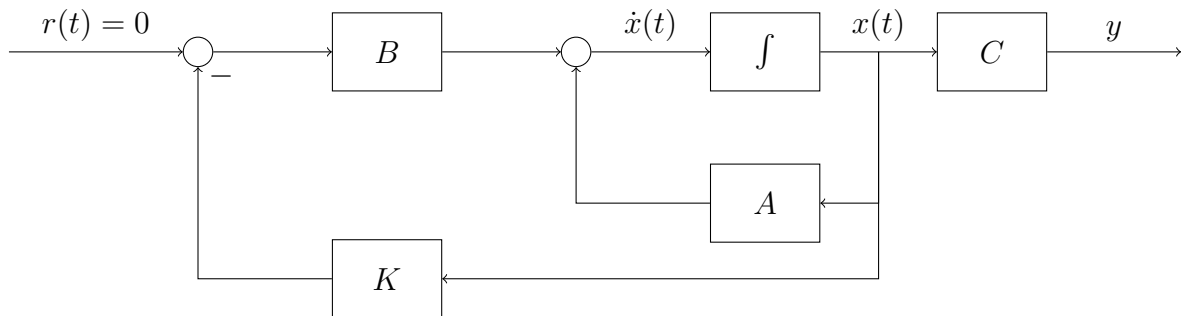


Figure 41: LQR Problem: Closed loop system.

¹⁰Here, it is assumed that $x(t)$ is already available. This is normally not the case and we will therefore introduce an element (observer) that gives us an estimation of it (see next chapters).

6.1.2 Kochrezept

(I) Choose the matrices Q and R :

- The matrix Q is usually chosen as follows:

$$Q = \bar{C}^T \cdot \bar{C}, \quad \bar{C} \in \mathbb{R}^{p \times n} \quad (6.11)$$

where

$$p = \text{rank}(Q) \leq n. \quad (6.12)$$

Equation 6.11 represents the **full-rank** decomposition of Q . This makes sure that the term with Q in the criterion contains enough informations about the state's behaviour. \bar{C} has strictly speaking nothing to do with the state space description's matrix C . We will however discover, that the choice $\bar{C} = C$ is really interesting.

- The matrix R is usually chosen as follows:

$$R = r \cdot \mathbb{1}_{m \times m}. \quad (6.13)$$

(II) Find the **symmetric, positive definite** solution Φ of the Riccati equation

$$\Phi \cdot B \cdot R^{-1} \cdot B^T \cdot \Phi - \Phi \cdot A - A^T \cdot \Phi - Q = 0. \quad (6.14)$$

Remark. Such a solution exists if

- The pair $\{A, B\}$ is completely controllable and
- The pair $\{A, \bar{C}\}$ is completely observable.

This doesn't mean that these conditions should be always fulfilled: **sufficient, not necessary**.

(III) Compute the matrix of the controller K with

$$K = R^{-1} \cdot B^T \cdot \Phi. \quad (6.15)$$

6.1.3 Frequency Domain

By looking at Figure 41 one can write

$$L_{\text{LQR}}(s) = K \cdot (s \cdot \mathbb{1} - A)^{-1} \cdot B \quad (6.16)$$

and

$$T_{\text{LQR}}(s) = C \cdot (s \cdot \mathbb{1} - (A - B \cdot K))^{-1} \cdot B. \quad (6.17)$$

6.1.4 Properties

A bunch of nice properties can be listed:

- The matrix of the closed-loop system

$$A - B \cdot K \quad (6.18)$$

is guaranteed to be a **Hurwitz** matrix.

Remark. A Hurwitz matrix is a matrix that has all eigenvalues with *negative* real parts.

But wait, what does it mean? We have learned that eigenvalues with negative real parts contribute to the stability of the closed-loop system. In this case is the closed-loop system **asymptotically stable**.

- If one chooses $R = r \cdot \mathbb{1}$ one gets for the minimal return difference
- If $Q \gg R$ we say that we have **cheap control energy**.
- If $R \gg Q$ we say that we have **expensive control energy**.

$$\mu_{\text{LQR}} = \min_{\omega} \{\sigma_{\min}(\mathbb{1} + L_{\text{LQR}})\} \geq 1. \quad (6.19)$$

- Closed-loop behaviour:

$$\max_{\omega} \{\sigma_{\max}(S(j \cdot \omega))\} \leq 1, \quad (6.20)$$

$$\max_{\omega} \{\sigma_{\max}(T(j \cdot \omega))\} \leq 2. \quad (6.21)$$

- Of course you are not expected in real life to solve the Riccati equation by hand. Since these calculations contain many matrix multiplications, numerical algorithms are developed. MATLAB offers the command

`K=lqr(A,B,Q,R);`

- Remind that

$$\Phi = \Phi^T. \quad (6.22)$$

6.1.5 Pole Placement

Another intuitive way of approaching this problem is the so called *pole placement*. Given that one knows that the state feedback matrix is $A - B \cdot K$, one can choose such a K matrix, that the desired poles are achieved. With this method one can have direct influence on the speed of the control system. The disadvantage, however, is that one has no guarantee of robustness. If one looks at Figure 42, one can derive the following complete state space description of the plant and of the controller:

$$\begin{aligned} \dot{x}(t) &= A \cdot x(t) + B \cdot u(t), \\ y(t) &= C \cdot x(t) + D \cdot u(t), \\ \dot{z}(t) &= F \cdot z(t) + G \cdot e(t), \\ u(t) &= H \cdot z(t), \end{aligned} \quad (6.23)$$

where $e(t)$ is the error of the controller.

Open Loop Conditions

By assuming for this case $D = 0$ and plugging the information of the controller in the plant, one gets

$$\begin{aligned} \dot{x}(t) &= A \cdot x(t) + B \cdot H z(t), \\ y(t) &= C \cdot x(t), \\ \dot{z}(t) &= F \cdot z(t) + G \cdot e(t). \end{aligned} \quad (6.24)$$

By defining two new variables $\tilde{x}(t) = \begin{pmatrix} x(t) \\ z(t) \end{pmatrix}$, $\tilde{u}(t) = \begin{pmatrix} u(t) \\ e(t) \end{pmatrix}$ and referring to Figure 43 one can write

$$\begin{aligned} A_{\text{open loop}} &= \begin{pmatrix} A & B \cdot H \\ 0 & F \end{pmatrix}, \\ B_{\text{open loop}} &= \begin{pmatrix} 0 \\ G \end{pmatrix}, \\ C_{\text{open loop}} &= (C \ 0). \end{aligned} \quad (6.25)$$

Remark. Because of the rule for determinant of matrices build up from other submatrices, we automatically know that the eigenvalues of $A_{\text{open loop}}$ are the eigenvalues of A and the eigenvalues of F . Mathematically

$$\lambda(A_{\text{open loop}}) = \lambda(A) + \lambda(F). \quad (6.26)$$

Closed Loop Conditions

It is still assumed $D = 0$. If one closes the loop it holds

$$e(t) = -y(t) \Rightarrow e(t) = -C \cdot x(t). \quad (6.27)$$

This gives the state space description

$$\begin{aligned} \dot{x}(t) &= A \cdot x(t) + B \cdot Hz(t), \\ y(t) &= C \cdot x(t), \\ \dot{z}(t) &= F \cdot z(t) + G \cdot (-C \cdot x(t)). \end{aligned} \quad (6.28)$$

By defining a new variable $\tilde{x}(t) = \begin{pmatrix} x(t) \\ z(t) \end{pmatrix}$ and referring to Figure 44 one can write

$$A_{\text{closed loop}} = \begin{pmatrix} A & B \cdot H \\ -G \cdot C & F \end{pmatrix}. \quad (6.29)$$

Remark. The eigenvalues of $A_{\text{closed loop}}$ are not well defined. Moreover it holds

$$\lambda(A_{\text{closed loop}}) = f(F, G, H). \quad (6.30)$$

This makes this method extremely superficial and not generally useful.

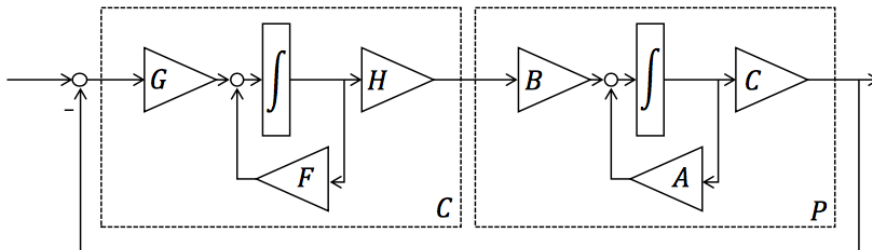


Figure 42: Signal Flow Diagram with Controller.

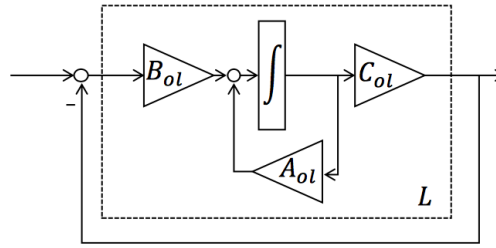


Figure 43: Open loop case.

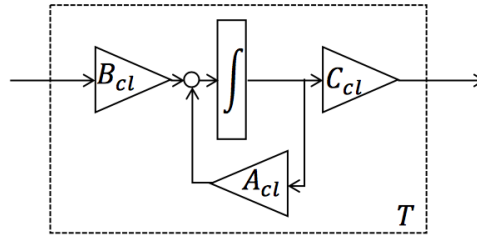


Figure 44: Closed loop case.

6.1.6 Examples

Example 35. Your task is to keep a space shuttle in its trajectory. The deviations from the desired trajectory are well described by

$$\begin{aligned} \frac{d}{dt}x(t) &= A \cdot x(t) + b \cdot u(t) \\ &= \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \cdot x(t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot u(t). \end{aligned}$$

$x_1(t)$ is the position of the space shuttle and $x_2(t)$ is its velocity. Moreover, $u(t)$ represents the propulsion. The position and the velocity are known for every moment and for this reason we can use a state-feedback regulator. In your first try, you want to implement a LQR with

$$Q = \begin{pmatrix} 1 & 0 \\ 0 & 2.25 \end{pmatrix}$$

$$R = 4.$$

- Show that the system is completely controllable.
- Find the state feedback matrix K .

After some simulations, you decide that the controller you've found is not enough for your task. Your boss doesn't want you to change the matrices Q and R and for this reason you decide to try with direct pole placement. The specifications are

- The system should not overshoot.
 - The error of the regulator should decrease with $e^{-3 \cdot t}$.
- Find the poles such that the specifications are met.
 - Find the new state feedback matrix K_2 .

Solution.

- (a) The controllability of the system is easy to show: the controllability matrix is

$$R_2 = \begin{pmatrix} b & A \cdot b \end{pmatrix} \\ = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

The rank of R is clearly 2: completely controllable.

- (b) In order to find the matrix K , one has to compute the solution of the Riccati equation related to this problem. First, one has to look at the form that this solution should have. Here b is a 2×1 vector. This means that since $\Phi = \Phi^T$ we are dealing with a 2×2 Φ of the form

$$\Phi = \begin{pmatrix} \varphi_1 & \varphi_2 \\ \varphi_2 & \varphi_3 \end{pmatrix}$$

With the Riccati equation, it holds

$$\Phi \cdot B \cdot R^{-1} \cdot B^T \cdot \Phi - \Phi \cdot A - A^T \cdot \Phi - Q = 0$$

Plugging in the matrices one gets

$$\begin{aligned} \Phi \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot 4^{-1} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix}^T \cdot \Phi - \Phi \cdot \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}^T \cdot \Phi - \begin{pmatrix} 1 & 0 \\ 0 & 2.25 \end{pmatrix} &= \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\ \frac{1}{4} \cdot \begin{pmatrix} \varphi_2 \\ \varphi_3 \end{pmatrix} \cdot (0 \quad 1) \cdot \begin{pmatrix} \varphi_1 & \varphi_2 \\ \varphi_2 & \varphi_3 \end{pmatrix} - \begin{pmatrix} 0 & \varphi_1 \\ 0 & \varphi_2 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ \varphi_1 & \varphi_2 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 2.25 \end{pmatrix} &= \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\ \frac{1}{4} \cdot \begin{pmatrix} 0 & \varphi_2 \\ 0 & \varphi_3 \end{pmatrix} \cdot \begin{pmatrix} \varphi_1 & \varphi_2 \\ \varphi_2 & \varphi_3 \end{pmatrix} - \begin{pmatrix} 1 & \varphi_1 \\ \varphi_1 & 2 \cdot \varphi_2 + 2.25 \end{pmatrix} &= \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\ \frac{1}{4} \cdot \begin{pmatrix} \varphi_2^2 & \varphi_2 \cdot \varphi_3 \\ \varphi_2 \cdot \varphi_3 & \varphi_3^2 \end{pmatrix} - \begin{pmatrix} 1 & \varphi_1 \\ \varphi_1 & 2 \cdot \varphi_2 + 2.25 \end{pmatrix} &= \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} \frac{\varphi_2^2}{4} - 1 & \frac{\varphi_2 \cdot \varphi_3}{4} - \varphi_1 \\ \frac{\varphi_2 \cdot \varphi_3}{4} - \varphi_1 & \frac{\varphi_3^2}{4} - 2 \cdot \varphi_2 - 2.25 \end{pmatrix} &= \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}. \end{aligned}$$

From this it follows:

- $\frac{\varphi_2^2}{4} - 1 \Rightarrow \varphi_2 = \pm 2$.
- From the fourth term, we get two cases:
 - $\varphi_2 = 2$: $\varphi_3 = \sqrt{25} = \pm 5$.
 - $\varphi_2 = -2$: $\varphi_3 = j \cdot \sqrt{7}$.
- Because of the *Sylvester's Criterion*, φ_1 should be positive, in order to get a positive definite matrix. Plugging into the second term: $\varphi_1 = \frac{\varphi_2 \cdot \varphi_3}{4} = 2.5$ is the only positive solution. This means that the only reasonable choice is $\varphi_2 = 2$ and $\varphi_3 = 5$.

We therefore get

$$\Phi = \begin{pmatrix} 2.5 & 2 \\ 2 & 5 \end{pmatrix}.$$

Because of the *Sylvester's Criterion* and the fact that Φ should be positive definite, this is a good matrix!

For the matrix K it holds

$$\begin{aligned} K &= R^{-1} \cdot B^T \cdot \Phi \\ &= \frac{1}{4} \cdot \begin{pmatrix} 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 2.5 & 2 \\ 2 & 5 \end{pmatrix} \\ &= \frac{1}{4} \cdot \begin{pmatrix} 2 & 5 \end{pmatrix}. \end{aligned}$$

- (c) Overshoot or in general oscillations, are due to the complex part of some poles. The real part of these poles is given by the decrease function of the error:

$$\pi_1 = \pi_2 = -3$$

- (d) The closed loop has feedback matrix

$$A - B \cdot K_2.$$

We have to choose K_2 such that the eigenvalues of the state feedback matrix are both -3 . The dimensions of K_2 are the same as the dimensions of K , namely 1×2 . It holds

$$\begin{aligned} A - B \cdot K_2 &= \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} k_1 & k_2 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ k_1 & k_2 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 1 \\ -k_1 & -k_2 \end{pmatrix}. \end{aligned}$$

The eigenvalues of this matrix are

$$\pi_{1,2} = \frac{-k_2 \pm \sqrt{k_2^2 - 4k_1}}{2}.$$

Since the two eigenvalues should have the same value, we know the the part under the square rooth should vanish. This means that $-\frac{k_2}{2} = -3 \Rightarrow k_2 = 6$. Moreover:

$$\begin{aligned} k_1 &= \frac{k_2^2}{4} \\ &= 9. \end{aligned}$$

The matrix finally reads

$$K_2 = \begin{pmatrix} 9 & 6 \end{pmatrix}.$$

Example 36. A system is given as

$$\dot{x}_1 = 3 \cdot x_2$$

$$\dot{x}_2 = 3 \cdot x_1 - 2 \cdot x_2 + \frac{1}{2} \cdot u$$

$$y = 4 \cdot x_1 + \frac{7}{3} \cdot x_2.$$

- (a) Solve the LQR problem for the criterion

$$J = \int_0^\infty 7 \cdot x_1^2 + 3 \cdot x_2^2 + \frac{1}{4} \cdot u^2 dt$$

and find K .

- (b) Find the eigenvalues of the closed-loop and the LQ regulator K .
(c) Does it change something if we use the new criterion?

$$J_{\text{new}} = \int_0^\infty 70 \cdot x_1^2 + 30 \cdot x_2^2 + \frac{10}{4} \cdot u^2 dt.$$

Solution.

- (a) From the theory of the course *Lineare Algebra I/II* about the quadratic forms, one can read

$$Q = \begin{pmatrix} 7 & 0 \\ 0 & 3 \end{pmatrix}$$

and

$$R = \frac{1}{4}.$$

The state-space description of the system can be rewritten as

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 3 \\ 3 & -2 \end{pmatrix}}_A \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \underbrace{\begin{pmatrix} 0 \\ \frac{1}{2} \end{pmatrix}}_B \cdot u$$

$$y = \underbrace{\begin{pmatrix} 4 & \frac{7}{3} \end{pmatrix}}_C \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \underbrace{0}_D \cdot u.$$

Plugging this into the Riccati equation one gets:

$$\begin{aligned} \Phi \cdot \begin{pmatrix} 0 \\ \frac{1}{2} \end{pmatrix} \cdot 4 \cdot \begin{pmatrix} 0 & \frac{1}{2} \end{pmatrix} \cdot \Phi - \Phi \cdot \begin{pmatrix} 0 & 3 \\ 3 & -2 \end{pmatrix} - \begin{pmatrix} 0 & 3 \\ 3 & -2 \end{pmatrix} \cdot \Phi - \begin{pmatrix} 7 & 0 \\ 0 & 3 \end{pmatrix} &= \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 2\varphi_2 \\ 2\varphi_3 \end{pmatrix} \cdot \begin{pmatrix} \varphi_2 & \varphi_3 \\ 2 & 2 \end{pmatrix} - \begin{pmatrix} 3\varphi_2 & 3\varphi_1 - 2\varphi_2 \\ 3\varphi_3 & 3\varphi_2 - 2\varphi_3 \end{pmatrix} - \begin{pmatrix} 3\varphi_2 & 3\varphi_3 \\ 3\varphi_1 - 2\varphi_2 & 3\varphi_2 - 2\varphi_3 \end{pmatrix} - \begin{pmatrix} 7 & 0 \\ 0 & 3 \end{pmatrix} &= \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} \varphi_2^2 & \varphi_2\varphi_3 \\ \varphi_2\varphi_3 & \varphi_3^2 \end{pmatrix} - \begin{pmatrix} 6\varphi_2 + 7 & 3\varphi_1 - 2\varphi_2 + 3\varphi_3 \\ 3\varphi_1 - 2\varphi_2 + 3\varphi_3 & 6\varphi_2 - 4\varphi_3 + 3 \end{pmatrix} &= \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}. \end{aligned}$$

Hence, one gets 3 equations (two elements are equal because of symmetry):

$$\begin{aligned} \varphi_2^2 - 6 \cdot \varphi_2 - 7 &= 0 \\ \varphi_2 \cdot \varphi_3 - 3 \cdot \varphi_1 + 2 \cdot \varphi_2 - 3 \cdot \varphi_3 &= 0 \\ \varphi_3^2 - 6 \cdot \varphi_2 + 4 \cdot \varphi_3 - 3 &= 0. \end{aligned}$$

From here, with the same procedure we have used before, one gets

$$\Phi = \begin{pmatrix} \frac{34}{3} & 7 \\ 7 & 5 \end{pmatrix}.$$

This gives K as

$$\begin{aligned} K &= 4 \cdot \begin{pmatrix} 0 & \frac{1}{2} \end{pmatrix} \cdot \begin{pmatrix} \frac{34}{3} & 7 \\ 7 & 5 \end{pmatrix} \\ &= \begin{pmatrix} 14 & 10 \end{pmatrix}. \end{aligned}$$

- (b) The matrix to analyze is

$$\begin{aligned} A - B \cdot K &= \begin{pmatrix} 0 & 3 \\ 3 & -2 \end{pmatrix} - \begin{pmatrix} 0 \\ \frac{1}{2} \end{pmatrix} \cdot \begin{pmatrix} 14 & 10 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 3 \\ 3 & -2 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 7 & 5 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 \\ -4 & -7 \end{pmatrix}. \end{aligned}$$

The eigenvalues of the closed loop system are given with

$$\begin{aligned}\det((A - B \cdot K) - \lambda \cdot \mathbb{1}) &= 0 \\ \lambda^2 + 7\lambda + 12 &= 0\end{aligned}$$

from which it follows: $\lambda_1 = -3$ and $\lambda_2 = -4$.

(c) No. K remains the same, because it holds $J_{\text{new}} = 10 \cdot J$.

Example 37. You have to design a LQ regulator for a plant with 2 inputs, 3 outputs and 6 state variables.

- (a) What are the dimensions of A , B , C and D ?
- (b) What is the dimension of the transfer function $u \rightarrow y$?
- (c) What is the dimension of the matrix Q of J_{LQR} ?
- (d) What is the dimension of the matrix R of J_{LQR} ?
- (e) What is the dimension of the matrix K ?

Solution.

(a) One can find the solution by analyzing the meaning of the matrices:

- Since we are given 6 states variables, the matrix A should have 6 rows and 6 columns, i.e. $A \in \mathbb{R}^{6 \times 6}$.
- Since we are given 2 inputs, the matrix B should have 2 columns and 6 rows, i.e. $B \in \mathbb{R}^{6 \times 2}$.
- Since we are given 3 outputs, the matrix C should have 6 columns and 3 rows, i.e. $C \in \mathbb{R}^{3 \times 6}$.
- Since we are given 2 inputs and 3 outputs, the matrix D should have 2 columns and 3 rows, i.e. $D \in \mathbb{R}^{3 \times 2}$.

(b) Since we are dealing with a system with 2 inputs and 3 outputs, $P(s) \in \mathbb{R}^{3 \times 2}$. Moreover, $P(s)$ should have the same dimensions of D because of its formula.

(c) From the formulation of Q one can easily see that its dimensions are the same of the dimensions of A , i.e. $Q \in \mathbb{R}^{6 \times 6}$.

(d) From

$$u(t) = -K \cdot x(t).$$

we can see that K should have 6 columns and 2 rows, i.e. $K \in \mathbb{R}^{2 \times 6}$.

Example 38. You design with MATLAB a LQ Regulator:

```
1  A = [1 0 0 0; 1 1 0 0; 1 1 1 0; 0 0 1 1];
2  B = [1 1 1 1; 0 1 0 2];
3  C = [0 0 0 1; 0 0 1 1; 0 1 1 1];
4
5  nx = size(A,1); Number of state variables of the plant, in Script: n
6  nu = size(B,2); Number of input variables of the plant, in Script: m
7  ny = size(C,1); Number of output variables of the plant, in Script: p
8
9  q = 1;
10 r = 1;
11 Q = q*eye(###);
12 R = r*eye(###);
13
14 K = lqr(A,B,Q,R);
```

Fill the following rows:

```
11 : ### =
12 : ### =
```

Solution. The matrix Q is a weight for the states and the matrix R is a weight for the inputs. The correct filling is

11 : ### = nx
12 : ### = nu

6.2 Extensions of the LQR problem

6.2.1 The LQRI

This is an extension of what was presented in the previous section. In the previous formulation, **no integral action** was present: in this section we introduce it. The main idea is to introduce a (P)I element to each output channel, in order to control the static error.

Definition

Two possible structure's representations can be seen in Figure 45 and Figure 46. Let's analyze this system by referring to Figure 46. Let's define a new state vector

$$\tilde{x}(t) = \begin{pmatrix} x(t) \\ v(t) \end{pmatrix} \in \mathbb{R}^{n+m}. \quad (6.31)$$

Two assumptions are then used:

- $D = 0$,
- we have the same number of inputs and outputs (square systems).

By subdividing different B matrices for $u(t)$, $w(t)$ and $r(t)$ one gets the dynamics

$$\frac{d}{dt}\tilde{x}(t) = \tilde{A} \cdot \tilde{x}(t) + \tilde{B}_u \cdot u(t) + \tilde{B}_r \cdot r(t) + \tilde{B}_w \cdot w(t), \quad (6.32)$$

where

$$\tilde{A} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}, \quad \tilde{B}_u = \tilde{B}_w = \begin{bmatrix} B \\ 0 \end{bmatrix}, \quad \tilde{B}_r = \begin{bmatrix} 0 \\ \mathbf{1} \end{bmatrix}. \quad (6.33)$$

This can be solved using the LQR approach and the new weight matrices

$$\begin{aligned} \tilde{C} &= \begin{pmatrix} \bar{C} & 0 \\ 0 & \gamma \cdot \mathbf{1} \end{pmatrix} \\ \Rightarrow \tilde{Q} &= \tilde{C}^T \cdot \tilde{C} \\ &= \begin{pmatrix} Q & 0 \\ 0 & \gamma^2 \cdot \mathbf{1} \end{pmatrix}. \end{aligned} \quad (6.34)$$

and

$$\tilde{R} = R. \quad (6.35)$$

The solution we want to find is then

$$\tilde{K} = (K \quad -K_I), \quad (6.36)$$

where $K \in \mathbb{R}^{m \times n}$ and $K_I \in \mathbb{R}^{m \times p}$.

Remark.

- One could ask, why the matrix R is unchanged. Recalling the role of R , one sees that it is the weight for the input, that remains unchanged with respect to the initial LQR problem.
- $\gamma > 0$ is the tuning parameter, that defines how strong will the integral action be.

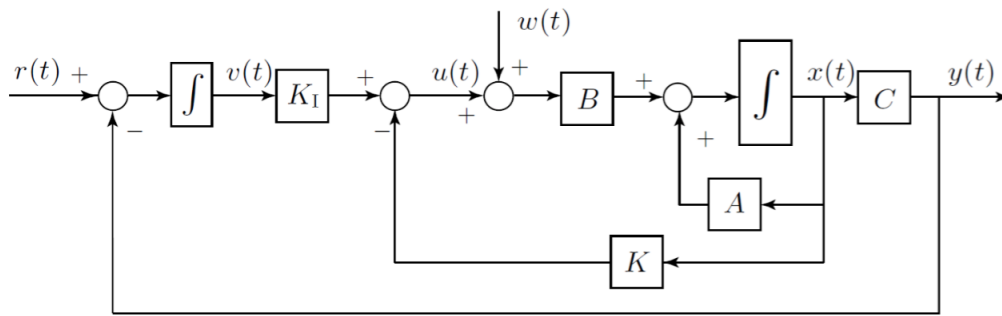


Figure 45: Structure of LQRI.

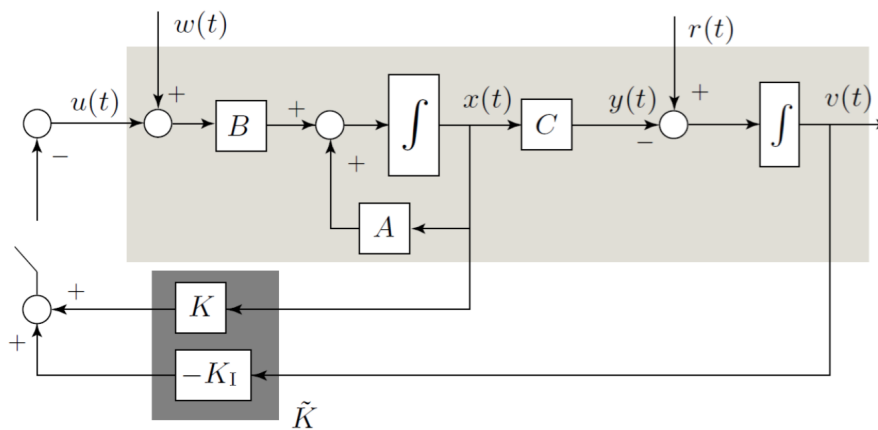


Figure 46: Structure of LQRI.

Kochrezept

- (I) Define the new system

$$\frac{d}{dt}\tilde{x}(t) = \tilde{A} \cdot \tilde{x}(t) + \tilde{B}_u \cdot u(t) + \tilde{B}_r \cdot r(t) + \tilde{B}_w \cdot w(t), \quad \tilde{x}(t) = \begin{pmatrix} x(t) \\ v(t) \end{pmatrix}. \quad (6.37)$$

with

$$\tilde{A} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}, \quad \tilde{B}_u = \tilde{B}_w = \begin{bmatrix} B \\ 0 \end{bmatrix}, \quad \tilde{B}_r = \begin{bmatrix} 0 \\ \mathbb{1} \end{bmatrix}. \quad (6.38)$$

- (II) Choose the desired weight matrices

$$\tilde{Q} = \begin{pmatrix} Q & 0 \\ 0 & \gamma^2 \cdot \mathbb{1} \end{pmatrix}, \quad \tilde{R} = R. \quad (6.39)$$

The controller is then

$$\tilde{K} = \begin{pmatrix} K & -K_I \end{pmatrix}, \quad (6.40)$$

- (III) Find \tilde{K} using the standard LQR formulation with $\{\tilde{A}, \tilde{B}_u, \tilde{Q}, \tilde{R}\}$ instead of $\{A, B, Q, R\}$.

- (IV) Use the MATLAB command

```
K_tilde=lqr(A_tilde,B_u_tilde,C_tilde'*C_tilde,r*eye(m,m)).
```

Examples

Example 39. The dynamics of a system are given as

$$\begin{aligned}\dot{x}(t) &= 3 \cdot x(t) + 2 \cdot u(t) \\ y(t) &= 2 \cdot x(t).\end{aligned}$$

First, you try to solve the problem by using the standard LQR problem statement. However, after some measurements, you notice that the controller you've found actually is not so good: there is a stationary controller's error. You decide to use the LQRI formulation, in order to eliminate this error. Find the new controller K_e using $R = 1$, $\tilde{C} = 2$ and integral action $\gamma = 1$.

Solution. The matrices that describe the dynamics of the system are

$$\begin{aligned} A &= (3), \\ B &= (2), \\ C &= (2). \end{aligned}$$

The new matrices $\{\tilde{A}, \tilde{B}, \tilde{C}\}$ are

$$\begin{aligned} \tilde{A} &= \begin{pmatrix} A & 0 \\ -C & 0 \end{pmatrix} \\ &= \begin{pmatrix} 3 & 0 \\ -2 & 0 \end{pmatrix}, \\ \tilde{B} &= \begin{pmatrix} B \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \\ \tilde{C} &= \begin{pmatrix} \bar{C} & 0 \\ 0 & \gamma \cdot \mathbb{1} \end{pmatrix} \\ &= \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}. \end{aligned}$$

The matrix \tilde{Q} reads

$$\begin{aligned} \tilde{Q} &= \bar{C}^T \cdot \bar{C} \\ &= \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}^2 \\ &= \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}. \end{aligned}$$

With $\Phi = \begin{pmatrix} \varphi_1 & \varphi_2 \\ \varphi_2 & \varphi_3 \end{pmatrix}$ one gets for the Riccati equation:

$$\begin{aligned} \Phi \cdot \tilde{B} \cdot \tilde{R}^{-1} \cdot \tilde{B}^T \cdot \Phi - \Phi \cdot \tilde{A} - \tilde{A}^T \cdot \Phi - \tilde{Q} &= 0 \\ \Phi \cdot \begin{pmatrix} 2 \\ 0 \end{pmatrix} \cdot 1 \cdot \begin{pmatrix} 2 & 0 \end{pmatrix} \cdot \Phi - \Phi \cdot \begin{pmatrix} 3 & 0 \\ -2 & 0 \end{pmatrix} - \begin{pmatrix} 3 & -2 \\ 0 & 0 \end{pmatrix} \cdot \Phi - \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix} &= \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 4\varphi_1^2 & 4\varphi_1\varphi_2 \\ 4\varphi_1\varphi_2 & 4\varphi_2^2 \end{pmatrix} - \begin{pmatrix} 3\varphi_1 - 2\varphi_2 & 0 \\ 3\varphi_2 - 2\varphi_3 & 0 \end{pmatrix} - \begin{pmatrix} 3\varphi_1 - 2\varphi_2 & 3\varphi_2 - 2\varphi_3 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix} &= \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 4\varphi_1^2 - 6\varphi_1 + 4\varphi_2 - 4 & 4\varphi_1\varphi_2 - 3\varphi_2 + 2\varphi_3 \\ 4\varphi_1\varphi_2 - 3\varphi_2 + 2\varphi_3 & 4\varphi_2^2 - 1 \end{pmatrix} &= \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}. \end{aligned}$$

Let's now analyze the equations:

- From the fourth term of the matrix, one can compute φ_2 . One gets $\varphi_2 = \pm \frac{1}{2}$.
- By plugging these two into the first term of the equation one can get the following results:
 - $\varphi_2 = \frac{1}{2}$, from which it follows $\varphi_1 = \frac{3 \pm \sqrt{17}}{4}$.
 - $\varphi_2 = -\frac{1}{2}$, from which it follows $\varphi_1 = \frac{3 \pm \sqrt{33}}{4}$.

Because of the Sylvester's criterion, one has to eliminate all negative solutions for φ_1 . This means we are left with two solutions:

$$\begin{aligned} - \varphi_2 &= \frac{1}{2}, \varphi_1 = \frac{3+\sqrt{17}}{4}. \\ - \varphi_2 &= -\frac{1}{2}, \varphi_1 = \frac{3+\sqrt{33}}{4}. \end{aligned}$$

- Let's plug these solutions into the second term of the equation, where $\varphi_3 = \frac{-4\varphi_1\varphi_2+3\varphi_2}{2}$:

$$- \varphi_2 = \frac{1}{2}, \varphi_1 = \frac{3+\sqrt{17}}{4}, \text{ from which it follows}$$

$$\begin{aligned} \varphi_3 &= -\frac{3+\sqrt{17}}{4} + \frac{3}{4} \\ &= -\frac{\sqrt{17}}{4}. \end{aligned}$$

$$- \varphi_2 = -\frac{1}{2}, \varphi_1 = \frac{3+\sqrt{33}}{4}, \text{ from which it follows}$$

$$\begin{aligned} \varphi_3 &= \frac{3+\sqrt{33}}{4} - \frac{3}{4} \\ &= \frac{\sqrt{33}}{4}. \end{aligned}$$

This means that the two possible solutions read

$$\begin{aligned} \Phi_1 &= \begin{pmatrix} \frac{3+\sqrt{17}}{4} & \frac{1}{2} \\ \frac{1}{2} & -\frac{\sqrt{17}}{4} \end{pmatrix}, \\ \Phi_2 &= \begin{pmatrix} \frac{3+\sqrt{33}}{4} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{\sqrt{33}}{4} \end{pmatrix}. \end{aligned}$$

Since these solutions already contain just the φ_1 that could lead to a positive definite matrix ($\varphi_1 > 0$), we can have a look at the determinants of them. In order to get a positive definite matrix, we should choose the matrix with a positive determinant.

It holds

$$\begin{aligned} \det(\Phi_1) &= \frac{3+\sqrt{17}}{4} \cdot -\frac{\sqrt{17}}{4} - \frac{1}{4} \\ &= -\frac{3\sqrt{17}+17}{16} - \frac{1}{4} \\ &< 0, \\ \det(\Phi_2) &= \frac{3+\sqrt{33}}{4} \cdot \frac{\sqrt{33}}{4} - \frac{1}{4} \\ &= \frac{3\sqrt{33}+33}{16} - \frac{1}{4} \\ &> 0. \end{aligned}$$

From which it follows that the only positive definite and therefore correct matrix is Φ_2 .

The new matrix K_e can now be computed:

$$\begin{aligned} K_e &= \begin{pmatrix} K & K_I \end{pmatrix} \\ &= R^{-1} \cdot \tilde{B}^T \cdot \Phi_2 \\ &= \begin{pmatrix} 2 & 0 \end{pmatrix} \cdot \begin{pmatrix} \frac{3+\sqrt{33}}{4} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{\sqrt{33}}{4} \end{pmatrix} \\ &= \begin{pmatrix} \frac{3+\sqrt{33}}{2} & -1 \end{pmatrix}. \end{aligned}$$

6.2.2 Finite Horizon LQR

Definition

Assuming a **time-dependent** LQR problem, we can write a time-dependent description of the system:

$$\frac{d}{dt}x(t) = A(t) \cdot x(t) + B(t) \cdot u(t), \quad x(t) \in \mathbb{R}^n, \quad u(t) \in \mathbb{R}^m, \quad t \in [t_a, t_b], \quad x(t_a) = x_a, \quad (6.41)$$

with the matrices same as before, but now time-dependent. The criterion for the LQR formulation can be written as

$$J(u) = x^T(t_b) \cdot P \cdot x(t_b) + \int_{t_a}^{t_b} (x^T(u(t)) \cdot Q(t) \cdot x(u(t)) + u^T(t) \cdot R(t) \cdot u(t)) dt, \quad (6.42)$$

with $P \in \mathbb{R}^{n \times n}$ and $P = P^T \geq 0$.

Remark. This formulation isn't relevant for the course: you have to make only sure to understand the point in considering time-dependent systems.

The equations for $u(t)$ and for K are the same as before and the Riccati equation will be a *differential equation*. Referring to Figure 47, we see that a **memory** element is present. In fact, the new matrix $K(t)$ is **computed in advance**, stored at every step in this memory and recovered everytime this is needed.

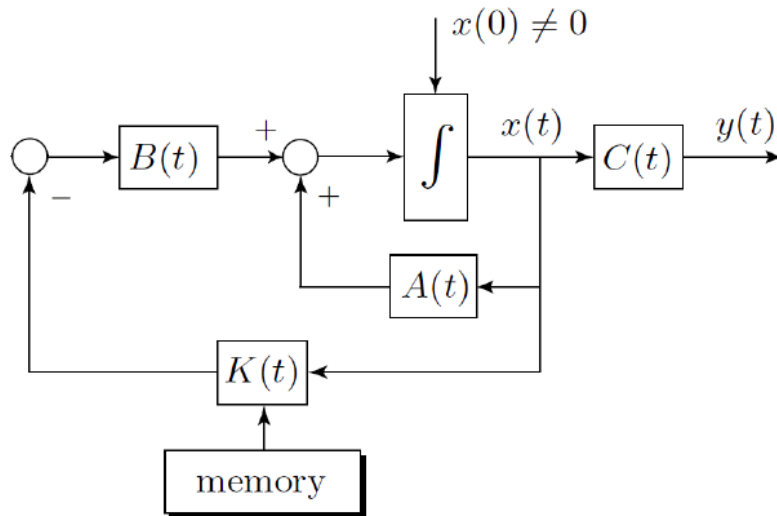


Figure 47: Structure of the finite horizon LQR.

6.2.3 Feedforward LQR Control Systems

Definition

Referring to Figure 48, we can see the structure of the feedforward LQR. The problem is still set up in the interval $[t_a, t_b]$. The objective of the controller is to follow a known reference signal $r(t)$ and to not care how much energy will this cost to us. The feedforward input $u_{ff}(t)$ can be computed in advance as well and stored in another memory.

The method to do this, will be explained in the next section.

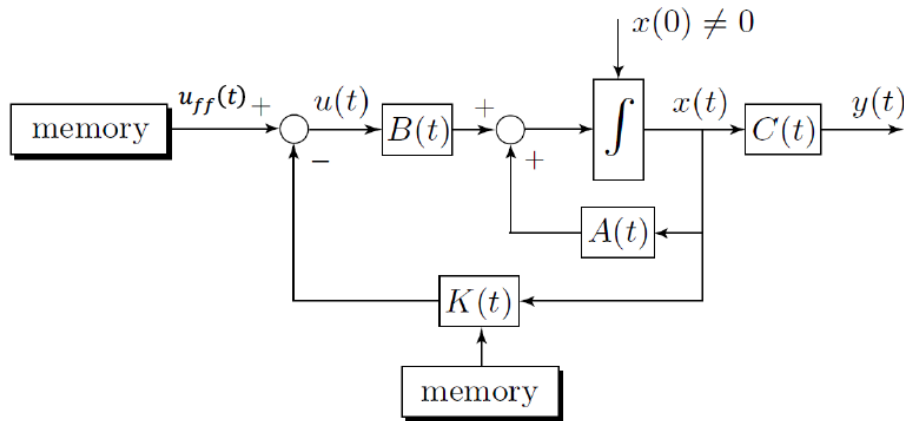


Figure 48: Structure of the feedforward LQR.

6.3 Numerical Optimization

6.3.1 Definition

The optimal input

$$u^*(t \in [0, T]) \quad (6.43)$$

is computed numerically offline for the whole planning window and is used as feedforward signal, i.e. $u_{ff} = u^*(t)$. The numerical optimization is possible, because the planning window has a limited dimension T .

The criterion we want to minimize is defined as

$$\begin{aligned} \min_{u(t)} J_T(x_0, u(t)) &= \min_{u(t)} \int_0^T \underbrace{l(x(t), u(t))}_{\text{Stage cost}} dt + \underbrace{m(x(t))}_{\text{Terminal cost}} \\ \text{s.t. } \dot{x}(t) &= f(t, x(t), u(t)), \\ x(0) &= x_0, \\ x(t) &\in \mathcal{X}, u(t) \in \mathcal{U}, \\ x(T) &\in \mathcal{X}_f, \end{aligned} \quad (6.44)$$

where

- $\dot{x}(t) = f(t, x(t), u(t))$: Model of the systems;
- $x(0) = x_0$: Initial Condition;
- $x(t) \in \mathcal{X}, u(t) \in \mathcal{U}$: “state and input constraint sets”;
- $x(T) \in \mathcal{X}_f$: “terminal state constraint set”.

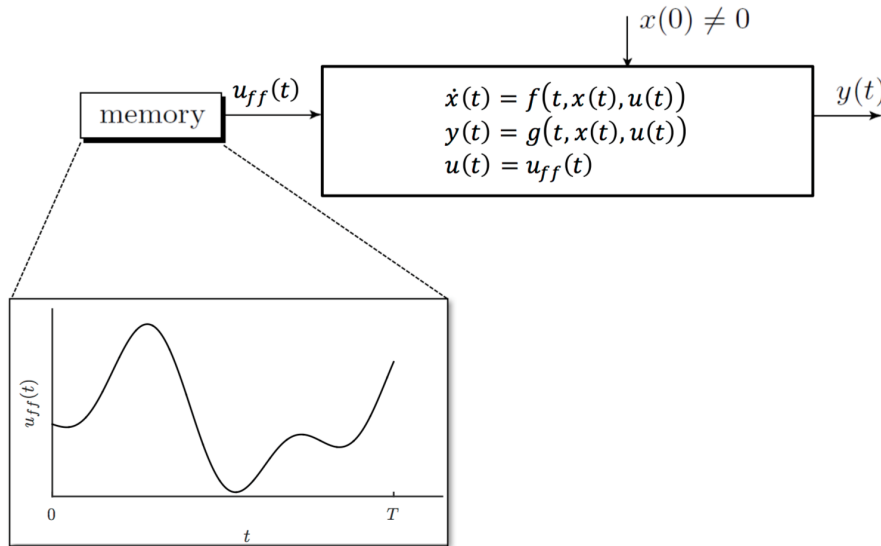


Figure 49: Numerical Optimization.

Remark. A bunch of other courses is offered in order to learn how to approach such an optimization problem. The aim of its introduction in this course, is only that of recognizing the elements that compose it and understand how they connect to each other.

6.3.2 Examples

Example 40. You would like to autonomously drive a train from Zürich to Lugano without stops in two and a half hours by consuming as little energy as possible. You therefore formulate the optimal control problem as

$$\begin{aligned}
 & \min_{P_m(t) \in [0, T]} \int_0^T P_{el}(P_m(t)) dt \\
 & \text{s.t. } \dot{s}(t) = \sqrt{\frac{2}{m} \cdot E_{kin}(t)}, \\
 & \dot{E}_{kin}(t) = P_m(t) - F_{drag}(s(t), E_{kin}(t)) \cdot \sqrt{\frac{2}{m} \cdot E_{kin}(t)}, \\
 & s(0) = E_{kin}(0) = E_{kin}(T) = 0, \quad s(T) = S_{end} \\
 & |P_m(t)| \leq P_{max} \\
 & E_{kin}(t) \leq \frac{m}{2} \cdot v_{max}(s(t))^2,
 \end{aligned} \tag{6.45}$$

where P_m is the power generated by the electric motors, $P_{el} : \mathbb{R} \rightarrow \mathbb{R}$ a nonlinear function describing the relation between mechanical power and power extracted from the electric grid, $s(t)$ is the current position of the train, $E_{kin}(t)$ its current kinetic energy, $F_{drag}(s(t), E_{kin}(t))$ the total drag force, S_{end} the total distance between Zürich and Lugano, P_{max} the maximum power that can be exerted by the motors and $v_{max}(s(t))$ the position-dependent maximum speed profile.

- Identify the state variables $x(t)$ and the input variables $u(t)$.
- Identify the system dynamics $\dot{x}(t) = f(x(t), u(t))$ and the initial conditions x_0 . Is it a nonlinear system?
- What is the stage cost function $l(x(t), u(t))$? Is it linear, quadratic or nonlinear? What about the terminal cost function $m(x(T))$ and the objective function $J_T(x_0, u(t))$?
- What is the value of T in seconds?
- Identify the state, input and terminal state constraint sets $\mathcal{X}, \mathcal{U}, \mathcal{X}_f$.

Solution.

- (a) The state variables are easy to see: they are the train position and the its kinetic energy. The state vector reads

$$x(t) = \begin{pmatrix} s(t) \\ E_{kin}(t) \end{pmatrix}.$$

We are given only an input: the power exerted by the motors. Mathematically:

$$u(t) = P_m(t).$$

- (b) The system dynamics are computed by taking the derivative of the state vector. Reading from the problem's description one gets

$$\dot{x}(t) = \begin{pmatrix} \sqrt{\frac{2}{m} \cdot x_2(t)} \\ u(t) - F_{drag}(x_1(t), x_2(t)) \cdot \sqrt{\frac{2}{m} \cdot x_2(t)} \end{pmatrix},$$

with $x_1(t)$, $x_2(t)$ state variables. Of course, this represents a nonlinear dynamical system. The initial conditions are $x_0 = 0$.

- (c) The stage cost function is

$$l(x(t), u(t)) = P_{el}(u(t))$$

and is nonlinear.

The terminal cost function is zero, i.e.

$$m(x(T)) = 0.$$

The objective function reads

$$J_T(x_0, u(t)) = \int_0^T P_{el}(u(t)) dt.$$

- (d) Since we would like to complete our journey within 2.5 hours, it holds

$$T = 3600s/h \cdot 2.5h = 9000s.$$

- (e) The state constraint set is

$$\mathcal{X} = \{x \in \mathbb{R}^2 : 0 \leq x_1 \leq S_{end}, x_2 \leq \frac{m}{2} \cdot v_{max}^2(x_1)\}.$$

The input constraint set is

$$\mathcal{U} = \{u \in \mathbb{R} : |u| \leq P_{max}\}.$$

The terminal state constraint set is

$$\mathcal{X}_f = \left\{ \begin{pmatrix} S_{end} \\ 0 \end{pmatrix} \right\}.$$

6.4 Model Predictive Control (MPC)

A whole course about this topic is offered in the master at IDSC (Prof. Zeilinger).

6.4.1 Definition

Basically, with MPC one solves in each time step the numerical optimization problem defined in the previous section. The solution is always given as feedback to the system.

Method

For every Δt this procedure is applied:

- (I) Measure or estimate the actual state, i.e.

$$x(t) = z. \quad (6.46)$$

- (II) Find the optimal input for the whole planning window T :

$$u^*([0, T]) = \arg \min_{u([0, T])} J_T(z, u([0, T])), \quad (6.47)$$

with the model and the constraints introduced in 6.44.

Remark. $\arg \min f(x)$ is the argument that minimizes $f(x)$.

- (III) Implement just the first part

$$u^*([0, \Delta t]) \quad (6.48)$$

of the calculated solution.

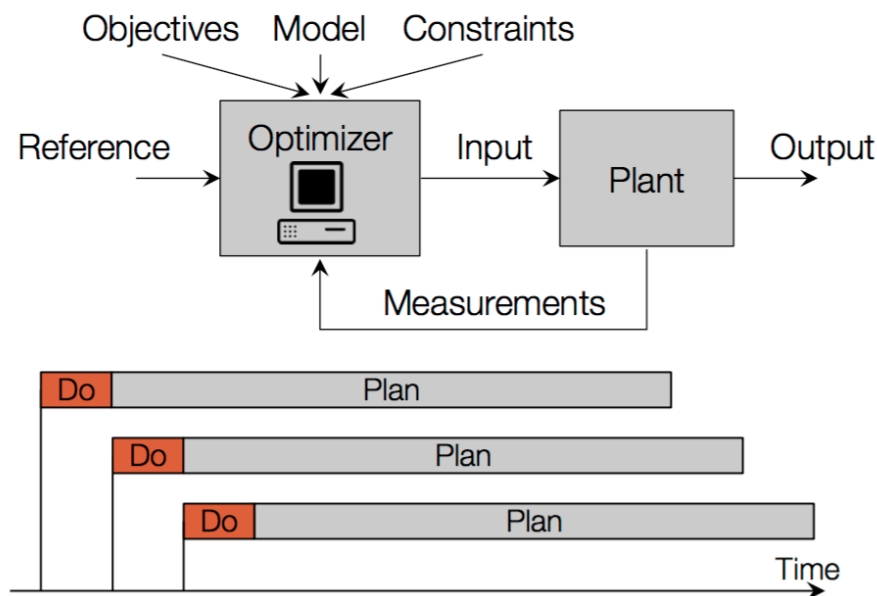


Figure 50: MPC method description.

One of the big advantages of Model Predictive Control, is that this method can work with every model, i.e. also with nonlinear models and delay elements. Another advantage is that the criterion (objective function) can be freely chosen. However, stability is not guaranteed and good models are computationally demanding.

6.4.2 Examples

Example 41. Consider the Model Predictive Control scheme $\mathcal{P}(x_0)$

$$\begin{aligned} \min_{u(t \in [0, T])} \int_0^T q \cdot x^2(t) + r \cdot u^2(t) dt \\ \text{s.t. } \dot{x}(t) = u(t), \quad x(0) = x_0 \\ u(t) \in \mathcal{U}, \quad x(t) \in \mathcal{X}, \quad x(T) \in \mathcal{X}_f. \end{aligned}$$

- (a) How can we solve the problem for $T \rightarrow \infty$ and $\mathcal{X} = \mathcal{U} = \mathcal{X}_f = \mathbb{R}$? What is the solution for $u^*(t)$?
- (b) What happens if the value of r is decreased? And what if it is increased? Include the notion of *cheap* and *expensive* control in your reasoning.
- (c) In this case would you need to iteratively solve the optimization problem? Why?
- (d) Consider now a finite horizon T and the constraint sets

$$\begin{aligned} \mathcal{X} &= \mathbb{R} \\ \mathcal{U} &= \{u \in \mathbb{R} : |u| \leq u_{max}\} \\ \mathcal{X}_f &= \{0\}. \end{aligned}$$

Define the feasible set of this MPC scheme, i.e. find the set

$$\mathcal{X}_T = \{x_0 \in \mathbb{R} : \mathcal{P}(x_0) \text{ admits a feasible solution}\}.$$

What happens if $x_0 \notin \mathcal{X}_T$?

- (e) Now also consider

$$\mathcal{X} = \{x \in \mathbb{R} : 0 \leq x \leq x_{max}\}.$$

How does the \mathcal{X}_T change?

Solution.

- (a) The MPC optimization problem becomes

$$\min_{u(t)} \int_0^\infty q \cdot x^2(t) + r \cdot u^2(t) dt \text{ s.t. } \dot{x}(t) = u(t), x(0) = x_0.$$

This is a classic unconstrained infinite-horizon LQR problem that we can tackle solving the scalar version of the Continuous-time Algebraic Riccati Equation (CARE):

$$\frac{1}{r} \cdot B^2 \cdot \Phi^2 - 2 \cdot A \cdot \Phi - q = 0.$$

Since in our case

$$\begin{aligned} A &= 0 \\ B &= 1 \end{aligned}$$

and $\Phi > 0$, we obtain

$$\Phi = \sqrt{q \cdot r}.$$

Therefore, the static gain for the optimal state-feedback regulator is

$$\begin{aligned} K &= \frac{B}{r} \cdot \Phi \\ &= \sqrt{\frac{q}{r}}. \end{aligned}$$

We know that

$$u^*(t) = -K \cdot x(t),$$

where $x(t)$ can be computed from the dynamics of the closed loop

$$\dot{x}(t) = -K \cdot x(t).$$

This is an easy differential equation and the solution reads

$$\begin{aligned} x^*(t) &= x_0 \cdot e^{-K \cdot t} \\ &= x_0 \cdot e^{-\sqrt{\frac{q}{r}} \cdot t}. \end{aligned}$$

Plugging this into the wanted input we find

$$\begin{aligned} u^*(t) &= -K \cdot x^*(t) \\ &= -\sqrt{\frac{q}{r}} \cdot x_0 \cdot e^{-\sqrt{\frac{q}{r}} \cdot t}. \end{aligned}$$

- (b) This case is interesting to analyze, since the control gain K is the same as the convergence rate of the state variable. If the value of r is decreased we would do cheap control: the *cheaper* control energy would result in a fastest convergence rate $K = \sqrt{\frac{q}{r}}$ at the expense of a larger control gain. If we increase r , making the control energy more *expensive*, the control gain will be reduced at the expensive of a slower convergence rate.
- (c) It is not necessary to solve the optimization problem iteratively, as for this special case the optimal input $u^*(t)$ can be computed as a closed form solution. Typically, this would not be possible, as the presence of state and input constraints would not allow to solve the optimization problem analytically. In that case, a numerical solver must be employed.

- (d) The state dynamics are given by the open integrator $\dot{x}(t) = u(t)$. Hence the maximum change achievable in x over the infinite horizon T can be computed integrating the minimum and maximum values allowed for the input, i.e. $u(t) = \pm u_{max}$. Since the input constraint is symmetric, we obtain for the maximum achievable distance for both negative and positive directions

$$\Delta x_{max} = \int_0^T u_{max} dt = u_{max} \cdot T.$$

As we do not have state constraints, but we are only forcing the state variable to be at the origin at the end of the horizon, i.e.

$$x(T) \in \mathcal{X}_f = \{0\} \Rightarrow x(T) = 0,$$

the feasible set is defined by all the initial conditions whose distance from the origin is less than the maximum achievable distance Δx_{max} , i.e.

$$\begin{aligned} \mathcal{X}_T &= \{x_0 \in \mathbb{R} : \mathcal{P}(x_0) \text{ admits a feasible solution} \\ &= \{x_0 \in \mathbb{R} : |x_0| \leq u_{max} \cdot T\}. \end{aligned}$$

If at some point $x_0 \notin \mathcal{X}_T$, which means $|x_0| > \Delta x_{max}$, there will be no admissible control trajectory able to steer the state variable to the origin within the horizon T , and therefore the MPC problem will be unfeasible and no solution will be found.

- (e) Since we are now imposing the additional constraint

$$x(t) \in \mathcal{X} = \{x \in \mathbb{R} : 0 \leq x \leq x_{max}\},$$

x_0 must always be inside of \mathcal{X} . Hence, \mathcal{X}_T results from the intersection of the admissible set computed in part (c) $\mathcal{X}_T^{\text{old}}$ and the state constraint set \mathcal{X} as

$$\mathcal{X}_T^{\text{old}} \cap \mathcal{X} = \{x_0 \in \mathbb{R} : 0 \leq x_0 \leq \min\{u_{max} \cdot T, x_{max}\}.$$

6.5 The Linear Quadratic Gaussian (LQG)

In the previous section we assumed that all the state variables of the system were given. In non academic cases, however, this is not the case: one knows just the output $y(t)$ and the input $u(t)$. Hence, one has to figure out how to get the actual state $x(t)$. The idea is to use an **observer** to get an estimate of $x(t)$, also called $\hat{x}(t)$. A whole course about estimation is offered at IDSC in the master by Prof. D'Andrea: *Recursive Estimation*.

6.5.1 Observer

For the following explanations, I refer to Figure 51. The idea is to *copy* the internal description of the plant and its dynamics, in order to predict $\hat{x}(t)$. This is done by connecting two feedback loops through the observer gain L , that can be tuned for different applications. First of all, let's define the observation error as

$$\bar{x}(t) = x(t) - \hat{x}(t) \in \mathbb{R}^n. \quad (6.49)$$

The dynamics of the error are described by its derivative, namely

$$\begin{aligned} \frac{d}{dt}\bar{x}(t) &= \frac{d}{dt}x(t) - \frac{d}{dt}\hat{x}(t) \\ &= A \cdot x(t) + B \cdot u(t) - (\hat{A} \cdot \hat{x}(t) + \hat{B} \cdot u(t) + L \cdot (C \cdot x(t) - \hat{C} \cdot \hat{x}(t))) \\ &= A \cdot \bar{x}(t) - L \cdot C \cdot \bar{x}(t) \\ &= (A - L \cdot C) \cdot \bar{x}(t), \end{aligned} \quad (6.50)$$

by assuming the **exact** duplication of the system's dynamics, i.e. $A = \hat{A}$, $B = \hat{B}$ and $C = \hat{C}$. If the matrix $A - L \cdot C$ has **asymptotically stable** eigenvalues, the observation error will converge asymptotically to 0. The speed of convergence can be *tuned* with the gain L .

The whole point here, is to find such a matrix with asymptotically stable eigenvalues. Since with the LQR approach we are guaranteed to have asymptotically stable eigenvalues for the matrix $A - B \cdot K$, we can find similarities between the two: in fact the roles of B and C are interchanged. By converting the problem, in order to get the same role for the matrices, one gets

$$A - L \cdot C \Rightarrow (A - L \cdot C)^T = A^T - C^T \cdot L^T \quad (6.51)$$

This makes the things easier: we can now, according to Table 1, approach this problem as a LQR problem with changed matrices. By plugging the new matrices into the Riccati equation, one gets

$$\frac{1}{q} \cdot \Psi \cdot C^T \cdot C \cdot \Psi - \Psi \cdot A^T - A \cdot \Psi - \bar{B} \cdot \bar{B}^T = 0, \quad (6.52)$$

where Ψ is its solution. The matrix L is then given as

$$L^T = \frac{1}{q} \cdot C \cdot \Psi. \quad (6.53)$$

LQR	LQG
$[A - B \cdot K]$	$[(A - L \cdot C)^\top] = [(A^\top - C^\top \cdot L^\top)]$
A	A^\top
B	C^\top
$Q = \bar{C}^\top \cdot \bar{C}$	$\bar{B} \cdot \bar{B}^\top$
$R = r \cdot \mathbb{1}$	$q \cdot \mathbb{1}$

Table 1: LQG and LQR.

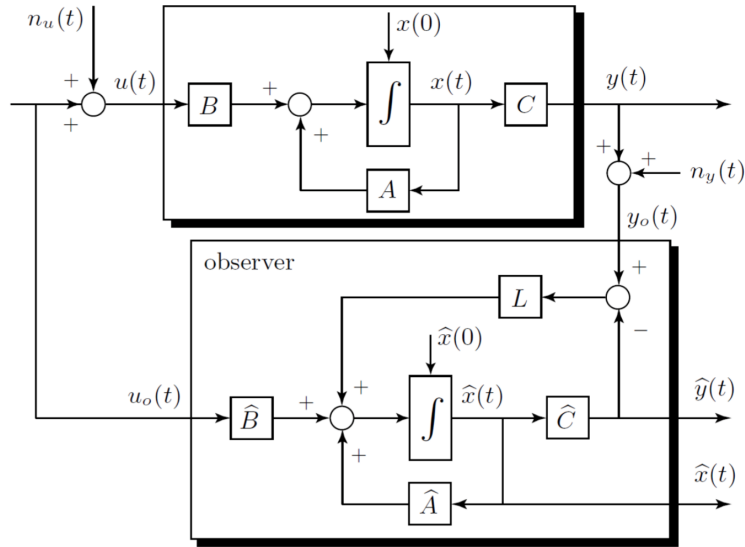


Figure 51: Structure of a state observer.

6.5.2 The LQG Controller

The LQG controller uses the LQR approach and the concept of the observer and connect the two into a unique system. Starting from the solution of the LQR problem, that contained the controller K and the observed signal $\hat{x}(t)$ from the observer, one can write as learned

$$u(t) = -K \cdot \hat{x}(t). \quad (6.54)$$

Since we have now an estimate of $x(t)$, we can approach the problem with a normal LQR method. Let's define

$$\tilde{x} = \begin{pmatrix} x(t) \\ \hat{x}(t) \end{pmatrix}. \quad (6.55)$$

By connecting the observer and the plant with the gain $-K$ one can actually analyze the open loop and the closed loop system conditions:

Open Loop Conditions

For open loop conditions, we can write the state dynamics as

$$\frac{d}{dt} \tilde{x}(t) = \tilde{A}_{\text{open loop}} \cdot \tilde{x}(t) + \tilde{B}_{\text{open loop}} \cdot e(t). \quad (6.56)$$

By looking at Figure 52, we can derive

$$\begin{aligned}\tilde{A}_{\text{open loop}} &= \begin{pmatrix} A & -B \cdot K \\ 0 & A - B \cdot K - L \cdot C \end{pmatrix}, \\ \tilde{B}_{\text{open loop}} &= \begin{pmatrix} 0 \\ -L \end{pmatrix}, \\ \tilde{C}_{\text{open loop}} &= (C \ 0).\end{aligned}\tag{6.57}$$

Closed Loop Conditions

For closed loop conditions, we can write the state dynamics as

$$\frac{d}{dt}\tilde{x}(t) = \tilde{A}_{\text{closed loop}} \cdot \tilde{x}(t) + \tilde{B}_{\text{closed loop}} \cdot e(t)\tag{6.58}$$

and analogously it follows from Figure 52 that

$$\begin{aligned}\tilde{A}_{\text{closed loop}} &= \begin{pmatrix} A & -B \cdot K \\ L \cdot C & A - B \cdot K - L \cdot C \end{pmatrix}, \\ \tilde{B}_{\text{closed loop}} &= \begin{pmatrix} 0 \\ -L \end{pmatrix}, \\ \tilde{C}_{\text{closed loop}} &= (C \ 0).\end{aligned}\tag{6.59}$$

If one analyzes this in the frequency domain, one gets

$$\begin{aligned}L_{\text{LQG}}(s) &= P(s) \cdot C(s) \\ &= \underbrace{C \cdot (s \cdot \mathbb{1} - A)^{-1} \cdot B \cdot K}_{P(s)} \cdot \underbrace{(s \cdot \mathbb{1} - (A - B \cdot K - L \cdot C))^{-1} \cdot L}_{C(s)}\end{aligned}\tag{6.60}$$

and

$$T_{\text{LQG}}(s) = L_{\text{LQG}}(s) \cdot (\mathbb{1} + L_{\text{LQG}}(s))^{-1}.\tag{6.61}$$

Remark.

- If the LQ regulator and the observer are stable, then so is the LQG closed loop.
- If the LQ regulator and the observer are robust, nothing can be said about the robustness of the LQG closed loop. This means that we don't have guarantee on robustness and it must be investigated a posteriori.

6.5.3 Kochrezept

(I) Find K : Solve a standard LQR problem with

$$\{A, B, Q = \bar{C}^T \cdot \bar{C}, R\}\tag{6.62}$$

(II) Find L : Solve a standard LQR problem with

$$\{A^T, C^T, Q = \bar{B} \cdot \bar{B}, R = q \cdot \mathbb{1}\}\tag{6.63}$$

(III) Use the MATLAB command

$$K = \text{lqr}(A, B, C_tilde' * C_tilde, r * \text{eye}(m, m)), L = \text{lqr}(A', C', B_bar * B_bar', q * \text{eye}(p, p))'.$$

Remark. It must hold

- $\{A, C\}$ completely observable
- $\{A, \bar{B}\}$ completely controllable.

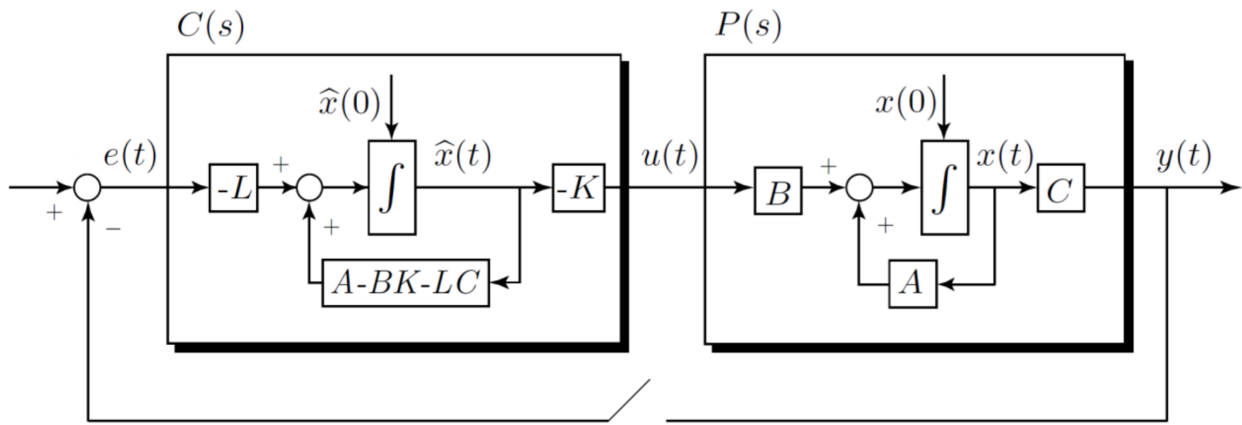


Figure 52: Structure of LQG controller.

6.5.4 Examples

Example 42. The dynamics of a system are given as

$$\begin{aligned}\dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= u(t) \\ y(t) &= x_1(t).\end{aligned}$$

You want to design a state observer. The observer should use the measurements for $y(t)$ and $u(t)$ in order to estimate the state variables $\hat{x}(t) \sim x(t)$.

- Which dimension has the observer matrix L ? Draw the signal diagram of such an observer.
- Compute the observer matrix L for $q = 1$.
- You have already computed a state feedback matrix $K = \begin{pmatrix} 1 & 1 \end{pmatrix}$ for the system above.

What is the complete transfer function of the controller $C(s)$? Use $L = \begin{pmatrix} \sqrt{2} \\ 1 \end{pmatrix}$.

Solution.

- (a) Since C is a 1×2 matrix and $L \cdot C$ is of the same dimensions of A , 2×2 matrix, L is a 2×1 matrix, i.e.

$$L = \begin{pmatrix} L_1 \\ L_2 \end{pmatrix}.$$

By adapting the general structure of an observer one gets Figure 53.

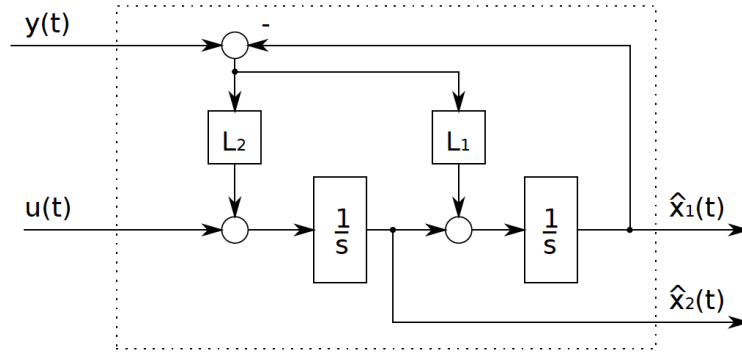


Figure 53: Structure of the state observer.

- (b) First of all let's read the system matrices:

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

$$D = 0.$$

Plugging these matrices into the Riccati equation one gets:

$$\frac{1}{q} \cdot \Psi \cdot C^T \cdot C \cdot \Psi - \Psi \cdot A^T - A \cdot \Psi - B \cdot B^T = 0$$

$$\Psi \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \end{pmatrix} \cdot \Psi - \Psi \cdot \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \cdot \Psi - \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} \psi_1^2 & \psi_1 \cdot \psi_2 \\ \psi_1 \cdot \psi_2 & \psi_2^2 \end{pmatrix} - \begin{pmatrix} 2\psi_2 & \psi_3 \\ \psi_3 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

The matrix Ψ is symmetric and positive definite and with these informations we can compute its elements:

- From the last term of the equation one gets

$$\psi_2^2 = 1 \Rightarrow \psi_2 = \pm 1.$$

- By plugging this into the first equation one gets $\psi_1 = \pm\sqrt{2}$. Because the positive definite condition, one gets $\psi_1 = \sqrt{2}$, $\psi_2 = 1$.
- Because of the form of C we don't care about ψ_3 .

From these calculations it follows

$$\begin{aligned} L^T &= \frac{1}{q} \cdot C \cdot \Psi \\ &= \frac{1}{1} \cdot (1 \ 0) \cdot \begin{pmatrix} \sqrt{2} & 1 \\ 1 & * \end{pmatrix} \\ &= (\sqrt{2} \ 1), \end{aligned}$$

and so

$$L = \begin{pmatrix} \sqrt{2} \\ 1 \end{pmatrix}.$$

- (c) As it is shown in the theory, the formula to calculate the controller's transfer function reads

$$C(s) = K \cdot (s \cdot \mathbb{1} - (A - B \cdot K - L \cdot C))^{-1} \cdot L.$$

By plugging in the found matrices one gets

$$\begin{aligned} (A - B \cdot K - L \cdot C) &= \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot (1 \ 1) - \begin{pmatrix} \sqrt{2} \\ 1 \end{pmatrix} \cdot (1 \ 0) \\ &= \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} - \begin{pmatrix} \sqrt{2} & 0 \\ 1 & 0 \end{pmatrix} \\ &= \begin{pmatrix} -\sqrt{2} & 1 \\ -2 & -1 \end{pmatrix}. \end{aligned}$$

It follows

$$\begin{aligned} (s \cdot \mathbb{1} - (A - B \cdot K - L \cdot C))^{-1} &= \begin{pmatrix} s + \sqrt{2} & -1 \\ 2 & s + 1 \end{pmatrix}^{-1} \\ &= \frac{1}{(s + \sqrt{2}) \cdot (s + 1) + 2} \cdot \begin{pmatrix} s + 1 & 1 \\ -2 & s + \sqrt{2} \end{pmatrix}. \end{aligned}$$

By plugging this into the formula one gets

$$\begin{aligned} C(s) &= K \cdot (s \cdot \mathbb{1} - (A - B \cdot K - L \cdot C))^{-1} \cdot L \\ &= (1 \ 1) \cdot \frac{1}{(s + \sqrt{2}) \cdot (s + 1) + 2} \cdot \begin{pmatrix} s + 1 & 1 \\ -2 & s + \sqrt{2} \end{pmatrix} \cdot \begin{pmatrix} \sqrt{2} \\ 1 \end{pmatrix} \\ &= \frac{1}{(s + \sqrt{2}) \cdot (s + 1) + 2} \cdot (s - 1 \ s + 1 + \sqrt{2}) \cdot \begin{pmatrix} \sqrt{2} \\ 1 \end{pmatrix} \\ &= \frac{1}{(s + \sqrt{2}) \cdot (s + 1) + 2} \cdot (\sqrt{2}s - \sqrt{2} + s + 1 + \sqrt{2}) \\ &= \frac{(\sqrt{2} + 1)s + 1}{(s + \sqrt{2}) \cdot (s + 1) + 2}. \end{aligned}$$

Example 43. You are working for your semester thesis at a project which includes a water reservoir. Your task is to determine the disturbance $d(t)$ that acts on the reservoir. Figure ?? shows the situation. The only state of the system is the water volume $x(t) = V(t)$. The volume flows in the reservoir $\dot{V}_{in}^*(t)$ are the known system input $u(t)$ and the unknown disturbance $d(t) > 0$. The volume flow of the system is assumed to be only dependent on the water volume, i.e.

$$\dot{V}_{out}^*(t) = -\beta \cdot x(t).$$

The system output $y(t)$ is the water level $h(t)$. The model of this reservoir reads

$$\begin{aligned} \frac{dx(t)}{dt} &= -\beta \cdot x(t) + u(t) + d(t), \\ y(t) &= \frac{1}{\alpha} \cdot x(t), \quad \alpha > 0, \beta > 0. \end{aligned} \quad (6.64)$$

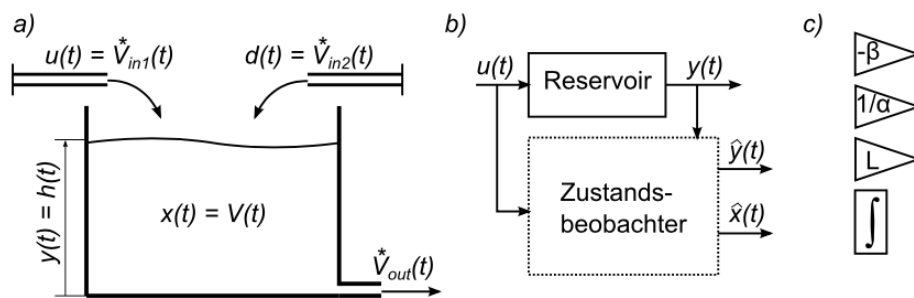


Figure 54: a) Drawing of the reservoir; b) Inputs and Outputs of the observer; c) Blocks for signal flow diagram.

The goal is to determine $d(t)$. Your supervisor has already tried to solve the model equations for $d(t)$: he couldn't determine the change in volume $\frac{dx(t)}{dt}$ with enough precision. Hence, you want to solve this problem with a state observer.

- Draw the signal flow diagram of such a state observer. Use the blocks of Figure ??c).
- The state feedback matrix L is in this case some scalar value. Which value can L be, in order to get an asymptotically stable state observer?
- Introduce a new signal $\hat{d}(t)$ in the state observer. This should approximate the real disturbance $d(t)$.
- Find the state space description of the observer with inputs $u(t)$ and $y(t)$ and output $\hat{d}(t)$.

Solution.

- (a) The signal flow diagram can be seen in Figure 55.

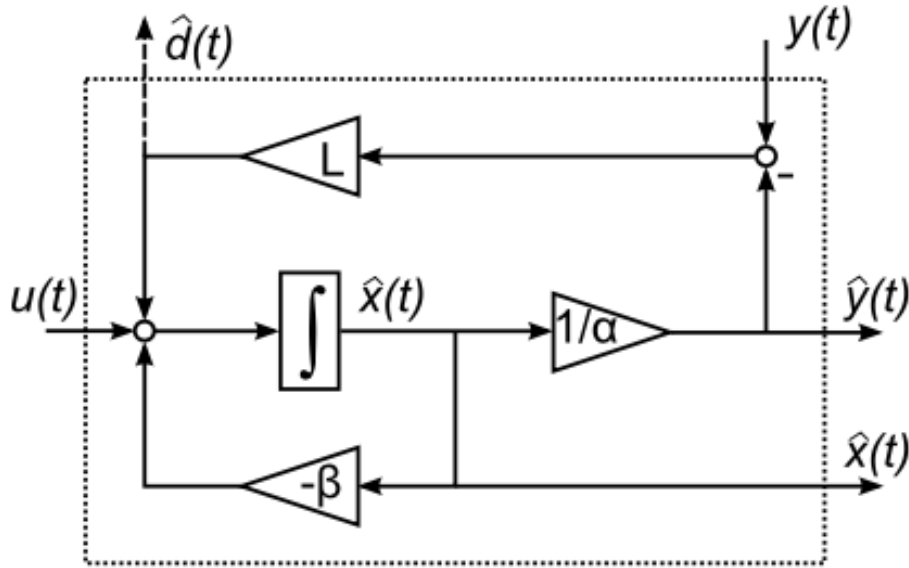


Figure 55: Signal flow diagram of the state observer.

- (b) The stability of the observer depends on the eigenvalues of $A - L \cdot C$. In this case, since $A - L \cdot C$ is a scalar,

$$A - L \cdot C < 0$$

should hold. This leads to

$$L > \frac{A}{C}$$

. With the given informations it follows

$$L > -\frac{\beta}{\alpha}.$$

- (c) The dashed line in Figure 55 represents the new output $\hat{d}(t)$. The integrator in Figure 55 has now 3 inputs. The arrow from downwards from the reservoir is $V_{\text{out}}^*(t)$, the arrow from left is the input flow $u(t) = V_{\text{in}1}^*(t)$. If we simulate the system without the dashed arrow, there is a deviation between the measured $y(t)$ and the simulated $\hat{y}(t)$. This results from the extra inflow $d(t) = V_{\text{in}}^*(t)$, which is not considered in the simulation.

- (d) The new state-space description reads

$$\begin{aligned} \frac{d\hat{x}(t)}{dt} &= \left(-\beta - \frac{L}{\alpha}\right) \cdot \hat{x}(t) + (1) \cdot u(t) + (L) \cdot y(t) \\ \hat{d}(t) &= \left(-\frac{L}{\alpha}\right) \cdot \hat{x}(t) + (0) \cdot u(t) + (L) \cdot y(t). \end{aligned}$$

6.6 Extension of LQG

6.6.1 The LQGI

As for the LQR the LQRI problem can be introduced, one can introduce a LQGI as well. The concept is really similar and the new matrices to compute are K , K_I and L . Please refer for the following computations to Figure 56.

Kochrezept

(I) Find K and K_I by solving a standard LQR problem (see LQRI) with

$$\{\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}\} \quad (6.65)$$

(II) Find L by solving a standard LQR problem with

$$\{\tilde{A}^T, \tilde{C}^T, Q = \bar{B} \cdot \bar{B}^T, R = q \cdot \mathbb{1}\} \quad (6.66)$$

This can be one more time analyzed with open loop and closed loop conditions:

Open Loop Conditions

The state space description for open loop conditions reads

$$\begin{aligned} \frac{d}{dt} \begin{pmatrix} x(t) \\ \hat{x}(t) \\ v(t) \end{pmatrix} &= \begin{pmatrix} A & -B_u \cdot K & B_u \cdot K_I \\ 0 & A - B_u \cdot K - L \cdot C & B_u \cdot K_I \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x(t) \\ \hat{x}(t) \\ v(t) \end{pmatrix} + \begin{pmatrix} 0 \\ -L \\ \mathbb{1} \end{pmatrix} \cdot e(t) \\ y(t) &= (C \ 0 \ 0) \cdot \begin{pmatrix} x(t) \\ \hat{x}(t) \\ v(t) \end{pmatrix}. \end{aligned} \quad (6.67)$$

Closed Loop Conditions

The state space description for closed loop conditions reads

$$\begin{aligned} \frac{d}{dt} \begin{pmatrix} x(t) \\ \hat{x}(t) \\ v(t) \end{pmatrix} &= \begin{pmatrix} A & -B_u \cdot K & B_u \cdot K_I \\ L \cdot C & A - B_u \cdot K - L \cdot C & B_u \cdot K_I \\ -C & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x(t) \\ \hat{x}(t) \\ v(t) \end{pmatrix} + \begin{pmatrix} B_u \\ B_u \\ \mathbb{1} \end{pmatrix} \cdot r(t) \\ y(t) &= (C \ 0 \ 0) \cdot \begin{pmatrix} x(t) \\ \hat{x}(t) \\ v(t) \end{pmatrix}. \end{aligned} \quad (6.68)$$

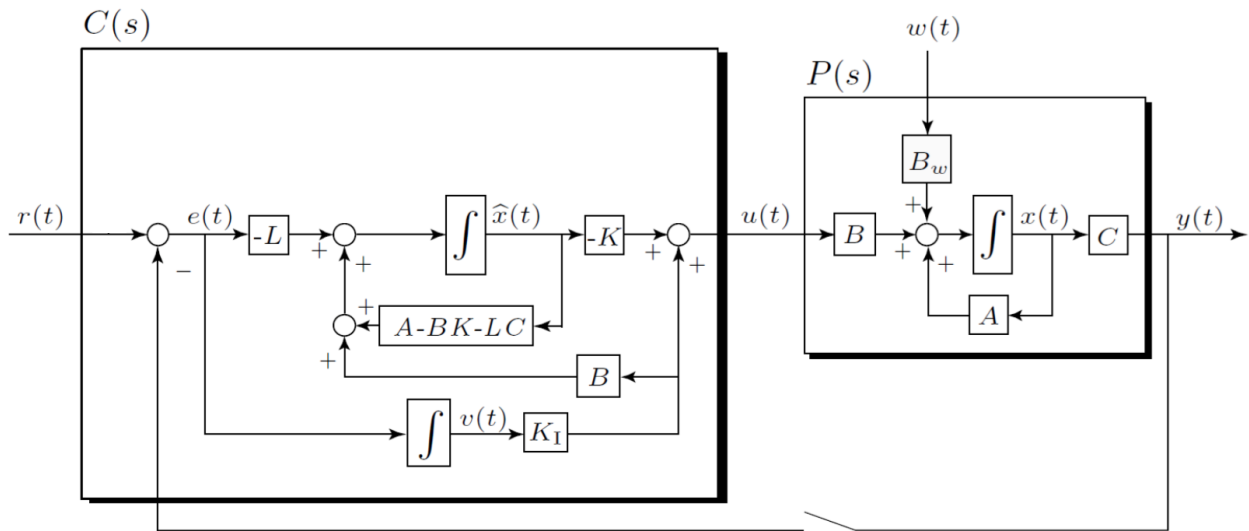


Figure 56: Structure of LQGI controller.

6.7 LTR

With the methods we have learned so far, we have no guarantee about robustness. The goal of the LTR method is to improve the robustness of the system, in particular if an observer is introduced.

6.7.1 Definition

The main idea is to design a LQR or an observer such that the system is enough *good* and in a second step, design a LQG with tuning parameter, to improve the robustness. This can be seen in Figure 57

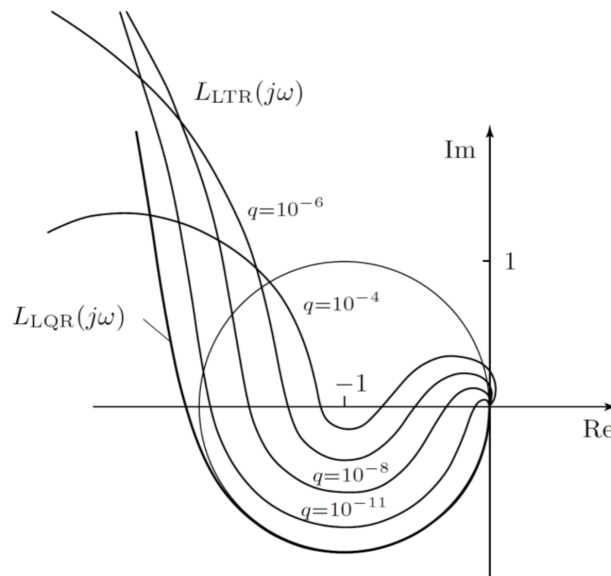


Figure 57: LTR.

6.7.2 β method

In order to improve the robustness, the β method is often chosen:

Kochrezept

(I) LQR: Compute K with

$$\frac{1}{r \cdot \beta} \cdot \Phi \cdot B \cdot B^\top \cdot \Phi - \Phi \cdot A - A^\top \cdot \Phi - Q = 0, \quad (6.69)$$

$$K = \frac{1}{r} \cdot B^\top \cdot \Phi. \quad (6.70)$$

Matlab: `Phi = care(A,B,Q,beta*r*eye(nu)), K = 1/r*B'*Phi.`

(II) LQG: Compute L with

$$\frac{1}{q \cdot \beta} \cdot \Psi \cdot C^\top \cdot C \cdot \Psi - \Psi \cdot A^\top - A \cdot \Psi - \bar{B} \cdot \bar{B}^\top = 0, \quad (6.71)$$

$$L^\top = \frac{1}{q} \cdot C \cdot \Psi. \quad (6.72)$$

Matlab: `Psi = care(A',C',B_bar*B_bar',beta*q*eye(ny)), L = (1/q*C*Psi)'`.

Remark. In order to compute K and L the command `lqr` was used so far. Since β appears only in the Riccati equations for Φ and Ψ , one should first solve the Riccati equation with the command `care` and then just in a second step compute K and L .

6.7.3 LQG/LQR for $m \geq p$

1. Modelization of the plant, linearization and normalization;
2. Definition of the specifications in frequency domain;
3. Design of the observer:

- (a) `Psi = care(A',C',B_bar*B_bar',beta*q*eye(ny));`
- (b) `L = (1/q*C*Psi)';`
- (c) Find $B_bar \cdot B_bar' \in \mathbb{R}^{n \times n}$, $q > 0$, and $\beta \leq 1$, s.t. the specifications are fulfilled.
- (d) Compute L_{obs} with

$$L_{obs}(s) = C \cdot (s \cdot \mathbb{I} - A)^{-1} \cdot L; \quad (6.73)$$

4. Design of the “state feedback regulator”:

- (a) `K = lqr(A,B,C'*Q_y*C,rho*R_u);`
- (b) Find $Q_y \in \mathbb{R}^{p \times p}$ and $R_u \in \mathbb{R}^{m \times m}$ (often $Q_y = \text{eye}(ny)$ and $R_u = \text{eye}(nu)$);
- (c) Choose `rho` smaller until all the specifications are fulfilled;
- (d) Loop gain:

$$L_{LTR}(s) = C \cdot (s \cdot \mathbb{I} - A)^{-1} \cdot B \cdot K \cdot (s \cdot \mathbb{I} - (A - B \cdot K - L \cdot C))^{-1} \cdot L; \quad (6.74)$$

5. Implement the controller.

6.7.4 LQG/LQR for $p \geq m$

1. Modelization of the plant, linearization and normalization;
2. Definition of the specifications in frequency domain;
3. Design the LQR:

(a) $\text{Phi} = \text{care}(A, B, Q, \text{beta} \cdot q \cdot \text{eye}(\text{nu}))$;

(b) $K = 1/r \cdot B' \cdot \text{Phi}$;

(c) Find $Q \in \mathbb{R}^{n \times n}$, $r > 0$, and $\beta \leq 1$, s.t. the specifications are fulfilled;

(d) Compute L_{obs} with

$$L_{\text{LQR}}(s) = K \cdot (s \cdot \mathbb{1} - A)^{-1} \cdot B; \quad (6.75)$$

4. Design the LTR “state feedback regulator” :

(a) $L = \text{lqr}(A', C', B \cdot Q_u \cdot B', \mu \cdot R_y)'$;

(b) Find $Q_u \in \mathbb{R}^{m \times m}$ and $R_y \in \mathbb{R}^{p \times p}$ (often $Q_u = \text{eye}(\text{nu})$ and $R_y = \text{eye}(\text{ny})$);

(c) Choose μ smaller until all the specifications are fulfilled;

(d) Loop gain:

$$L_{\text{LTR}}(s) = C \cdot (s \cdot \mathbb{1} - A)^{-1} \cdot B \cdot K \cdot (s \cdot \mathbb{1} - (A - B \cdot K - L \cdot C))^{-1} \cdot L; \quad (6.76)$$

5. Implement the controller.

A Linear Algebra

A.1 Matrix-Inversion

$$A^{-1} = \frac{1}{\det(A)} \cdot \text{adj}(A), \quad \{\text{adj}(A)\}_{ij} = (-1)^{i+j} \cdot \det(A_{ij})$$

Special Cases:

- $n = 2$:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \Rightarrow A^{-1} = \frac{1}{a \cdot d - b \cdot c} \cdot \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

- $n = 3$:

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \Rightarrow A^{-1} = \frac{1}{\det(A)} \begin{bmatrix} e \cdot i - f \cdot h & c \cdot h - b \cdot i & b \cdot f - c \cdot e \\ f \cdot g - d \cdot i & a \cdot i - c \cdot g & c \cdot d - a \cdot f \\ d \cdot h - e \cdot g & b \cdot g - a \cdot h & a \cdot e - b \cdot d \end{bmatrix}$$

- $(a + b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$

A.2 Differentiation with Matrices

$$\frac{d}{dx} A \cdot x = A^T$$

$$\frac{d}{dx} x^T \cdot A \cdot x = (A^T + A) \cdot x$$

A.3 Matrix Inversion Lemma

$$[M + v \cdot v^T]^{-1} = M^{-1} - \frac{1}{1 + v^T \cdot M^{-1} \cdot v} \cdot M^{-1} \cdot v \cdot v^T \cdot M^{-1}$$

B Rules

B.1 Trigo

$\alpha[^\circ]$	0	30	45	60	90	120	180
$\alpha[\text{rad}]$	0	$\frac{\pi}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{3}$	$\frac{\pi}{2}$	$\frac{2\pi}{3}$	π
$\sin(\alpha)$	0	$\frac{1}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{3}}{2}$	1	$\frac{\sqrt{3}}{2}$	0
$\cos(\alpha)$	1	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$	-1
$\tan(\alpha)$	0	$\frac{\sqrt{3}}{3}$	1	$\sqrt{3}$	$\pm\infty$	$-\sqrt{3}$	0
$\cot(\alpha)$	$\pm\infty$	$\sqrt{3}$	1	$\frac{\sqrt{3}}{2}$	0	$-\frac{\sqrt{3}}{2}$	$\pm\infty$

B.2 Euler-Forms

$$e^{ix} = \cos(x) + i \cdot \sin(x)$$

$$a + i \cdot b = |a + i \cdot b| \cdot e^{i \cdot \angle(a + i \cdot b)}$$

$$\sin(x) = \frac{1}{2i}(e^{ix} - e^{-ix})$$

$$\cos(x) = \frac{1}{2}(e^{ix} + e^{-ix})$$

B.3 Derivatives

$$(\log_a |x|)' = (\log_a e) \frac{1}{x} = \frac{1}{x \ln a}$$

$$(a^{cx})' = (c \ln a) a^{cx}$$

$$(\tan x)' = \frac{1}{\cos^2 x} = 1 + \tan^2 x$$

$$(\arcsin x)' = \frac{1}{\sqrt{1-x^2}}$$

$$(\arccos x)' = -\frac{1}{\sqrt{1-x^2}}$$

$$(\arctan x)' = \frac{1}{1+x^2}$$

B.4 Logarithms

$$\ln |y| \cdot C = \ln |y^C|$$

$$-\ln |r| = \ln |r^{-1}|$$

$$\ln(1) = \log(1) = 0$$

B.5 Magnitude and Phase

In the Bode diagram are magnitude and phase separate from each other. Magnitude and phase for complex fractions are:

$$\left| \frac{a + i \cdot b}{c + i \cdot d} \right| = \sqrt{\frac{a^2 + b^2}{c^2 + d^2}}$$

$$\angle \left(\frac{a + i \cdot b}{c + i \cdot d} \right) = \arg \left\{ \frac{a + i \cdot b}{c + i \cdot d} \right\} = \arctan \left(\frac{bc - ad}{ac + bd} \right)$$

$$\arg \{a + i \cdot b\} = \arctan \left(\frac{b}{a} \right)$$

$$\arg \{(a + i \cdot b)^c\} = c \cdot \arg \{a + i \cdot b\}$$

$$\arg \left\{ \frac{c}{(a + i \cdot b)} \right\} = \arg \{c\} - \arg \{a + i \cdot b\}$$

B.6 dB-Skala

Typically the unit for the magnitude is dB:

$$|\Sigma(j\omega)|_{\text{dB}} = 20 \cdot \log_{10} |\Sigma(j\omega)|$$

$$|\Sigma(j\omega)| = 10^{\frac{|\Sigma(j\omega)|_{\text{dB}}}{20}}$$

$$\frac{1}{X} \Big|_{\text{dB}} = -X \Big|_{\text{dB}}$$

$$(X \cdot Y) \Big|_{\text{dB}} = X \Big|_{\text{dB}} + Y \Big|_{\text{dB}}$$

Value	0.001	0.01	0.1	0.5	$\frac{1}{\sqrt{2}}$	1	$\sqrt{2}$	2	10	100
dB	-60	-40	-20	≈ -6	≈ -3	0	≈ 3	≈ 6	20	40

C MATLAB

C.1 General Commands

Command	Description
<code>A(i,j)</code>	Element of A in position <i>i</i> (row) and <i>j</i> (column)
<code>abs(X)</code>	Magnitude of all elements of X
<code>angle(X)</code>	Phase of all elements of X
<code>X'</code>	Complex conjugate and transpose of X
<code>X.'</code>	Transpose, not complex conjugate of X
<code>conj(X)</code>	Complex conjugate of all elements of X
<code>real(X)</code>	Realteil von allen Einträge von X
<code>imag(X)</code>	Imaginary part of all elements of X
<code>eig(A)</code>	Eigenvalues of A
<code>[V,D]=eig(A)</code>	Eigenvalues D (diagonal elements), eigenvectors V (column vectors)
<code>s=svd(A)</code>	singular values of A
<code>[U,Sigma,V]=svd(A)</code>	Singular Values Decomposition of A
<code>rank(A)</code>	Rank of A
<code>det(A)</code>	Determinant of A
<code>inv(A)</code>	Inverse of A
<code>diag([a1,...,an])</code>	Diagonalmatrix with a1,...,an as diagonal elements
<code>zeros(x,y)</code>	Zero matrix of dimension $x \times y$
<code>zeros(x)</code>	Zero matrix of dimension $x \times x$
<code>eye(x,y)</code>	Identity matrix of dimension $x \times y$
<code>eye(x)</code>	Identity matrix of dimension $x \times x$
<code>ones(x,y)</code>	One-Matrix (all elements = 1) of dimension $x \times y$
<code>ones(x)</code>	One-Matrix (all elements = 1) of dimension $x \times x$
<code>max(A)</code>	Largest element in vector A (A Matrix: Max in column vectors)
<code>min(A)</code>	Smallest element in vector A (A Matrix: Max in column vectors)
<code>sum(A)</code>	Sum of elements of A (A Matrix: Sum row pro row)
<code>dim=size(A)</code>	Dimension of A (size =[#rows #columns])
<code>dim=size(A,a)</code>	a =1: dim =#rows, a =2: dim =#columns, sonst dim =1
<code>t=a:i:b</code>	t =[a,a+i,a+2i,...,b-i,b] (row vector)
<code>y=linspace(a,b)</code>	row vector with 100 “linear-spaced” points in range [a,b]
<code>y=linspace(a,b,n)</code>	row vector with n “linear-spaced” points in range [a,b]
<code>y=logspace(a,b)</code>	row vector with 50 “logarithmically-spaced” points in range [$10^a, 10^b$]
<code>y=logspace(a,b,n)</code>	row vectors with n “logarithmically-spaced” points in range [$10^a, 10^b$]
<code>I=find(A)</code>	I : Index of non zero elements of A
<code>disp(A)</code>	Print on screen of A (String: 'name')

C.2 Control Systems Commands

Befehl	Beschreibung
<code>sys=ss(A,B,C,D)</code>	State-Space M. with A, B, C, D in time domain
<code>sys=ss(A,B,C,D,Ts)</code>	State-Space M. with A, B, C, D and sampling T_s (discrete-time)
<code>sys=zpk(Z,P,K)</code>	State-Space M. with zeros Z , poles P and gain K
<code>sys=zpk(Z,P,K,Ts)</code>	State-Space M. with zeros Z , poles P , gain K and sampling T_s
<code>sys=tf([bm ... b0],[an ... a0])</code>	Transfer function with bn in numerator and an in denom.
<code>P=tf(sys)</code>	Transfer function of <code>sys</code>
<code>P.iodelay=...</code>	Inserts to P delay.
<code>pole(sys)</code>	Poles of System
<code>zero(sys)</code>	Zeros of System
<code>[z,p,k]=zpkdata(sys)</code>	z : Zeros, p : Poles, k : static gain
<code>ctrb(sys)</code> or <code>ctrb(A,b)</code>	Controllability Matrix
<code>obsv(sys)</code> or <code>obsv(A,c)</code>	Observability Matrix
<code>series(sys1,sys2)</code>	series of <code>sys1</code> and <code>sys2</code>
<code>feedback(sys1,sys2)</code>	<code>sys1</code> with <code>sys2</code> as (negative) Feedback
<code>[Gm,Pm,Wgm,Wpm]=margin(sys)</code>	G_m : gain margin, P_m : phase margin, W_{pm} : crossover freq.
<code>[y,t]=step(sys,Tend)</code>	y : step response von <code>sys</code> until T , t : time
<code>[y,t]=impulse(sys,Tend)</code>	y : impulse response of <code>sys</code> until $Tend$, t : time
<code>y=lsim(sys,u,t)</code>	Simulation of <code>sys</code> with input u for the time t
<code>sim('Simulink model',Tend)</code>	Simulation of Simulink Model' until $Tend$
<code>p0=dcgain(sys)</code>	static gain ($P(0)$)
<code>K=lqr(A,B,Q,R)</code>	Gain Matrix K (solution of the LQR-Problem)
<code>[X,L,K]=care(A,B,Q)</code>	X : solution of the Riccati equation, G : Gain matrix
<code>Paug=augw(G,W1,W3,W2)</code>	Space State M. for \mathcal{H}_∞
<code>[K,C1,gamma]=hinfsyn(Paug)</code>	\mathcal{H}_∞ : K : Controller
<code>fr=evalfr(sys,f)</code>	<code>sys</code> evaluated in f ($s = f$)
<code>sysd=c2d(sys,Ts,method)</code>	Discretization of <code>sys</code> with <code>method</code> with Sampling Time T_s

C.3 Plot and Diagrams

Befehl	Beschreibung
<code>nyquist(sys)</code>	Nyquist diagram of the system <code>sys</code>
<code>nyquist(sys, {a,b})</code>	Nyquist diagram in interval <code>[a,b]</code> of the system <code>sys</code>
<code>bode(sys)</code>	Bode diagram of the system <code>sys</code>
<code>bode(sys, {a,b})</code>	Bode diagram in interval <code>[a,b]</code> of the system <code>sys</code>
<code>bodemag(sys)</code>	Bode diagram (just magnitude) of the system <code>sys</code>
<code>bodemag(sys, {a,b})</code>	Bode diagram (just magnitude) in interval <code>[a,b]</code> of the system. <code>sys</code>
<code>rlocus(sys)</code>	Root Locus diagram
<code>impulse(sys)</code>	Impulse Response of the system <code>sys</code>
<code>step(sys)</code>	Step response of the system <code>sys</code>
<code>pzmap(sys)</code>	Poles and zeros mapping of the system <code>sys</code>
<code>svd(sys)</code>	Singular values dynamics of the system <code>sys</code>
<code>plot(X,Y)</code>	Plot of <code>Y</code> as function of <code>X</code>
<code>plot(X,Y,...,Xn,Yn)</code>	Plot of <code>Yn</code> as function of <code>Xn</code> (for all <code>n</code>)
<code>stem(X,Y)</code>	Discrete plot of <code>Y</code> as function of <code>X</code>
<code>stem(X,Y,...,Xn,Yn)</code>	Discrete plot of <code>Yn</code> as function of <code>Xn</code> (for all <code>n</code>)
<code>xlabel('name')</code>	Name of the x-Axis
<code>ylabel('name')</code>	Name of the y-Axis
<code>title('name')</code>	Title of the plot
<code>xlim([a b])</code>	Range for the x-Axis (Plot between <code>a</code> and <code>b</code>)
<code>ylim([a b])</code>	Range for the y-Axis (Plot between <code>a</code> and <code>b</code>)
<code>grid on</code>	Grid
<code>title('name')</code>	Title of the plot
<code>legend('name1',..., 'name')</code>	Legend
<code>subplot(m,n,p)</code>	Grid <code>m×n</code> , Plot in Position <code>p</code>
<code>semilogx(X,Y)</code>	Logarithmic Plot with y-Axis linear