Lecture 3: Discrete Time Stability and MIMO Systems

1 Discrete-time Systems 2.0

1.1 Discrete-time Systems Stability

We want to investigate the stability conditions for the discrete-time system given as

$$\begin{aligned} x[k+1] &= A_\mathrm{d}x[k] + B_\mathrm{d}u[k] \\ y[k] &= C_\mathrm{d}x[k] + D_\mathrm{d}u[k], \quad x[0] = x_0. \end{aligned} \tag{1.1}$$

As usual, we want to find conditions for wich the state $x[k]$ does not diverge. The free evolution (free means without input) can be written as

$$x[k+1] = A_\mathrm{d}x[k]. \tag{1.2}$$

Starting from the initial state, one can write

$$\begin{aligned} x[1] &= A_\mathrm{d}x_0 \\ x[2] &= A_\mathrm{d}^2x_0 \\ &\vdots \\ x[k] &= A_\mathrm{d}^kx_0. \end{aligned} \tag{1.3}$$

In order to analyze the convergence of this result, let's assume that $A_\mathrm{d} \in \mathbb{R}^{n\times n}$ is diagonalizable and let's rewrite A_d with the help of its diagonal form:

$$\begin{aligned} x[k] &= A_\mathrm{d}^kx_0 \\ &= \left(TDT^{-1}\right)^k x_0 \\ &= \left(TD\underbrace{T^{-1}T}_{\mathbb{I}}DT^{-1}\dots TDT^{-1}\right) \\ &= TD^kT^{-1}x_0. \end{aligned} \tag{1.4}$$

where D is the matrix containing the eigenvalues of A_d and T is the matrix containing the relative eigenvectors. One can rewrite this using the modal decomposition as

$$TD^kT^{-1}x_0 = \sum_{i=1}^{n} \alpha_i\lambda_i^k v_i, \tag{1.5}$$

where v_i are the eigenvectors relative to the eigenvalues λ_i and $\alpha_i = T^{-1}x_0$ some coefficients depending on the initial condition x_0. Considering any possible eigenvalue, i.e. $\lambda_i = \rho_i e^{j\phi_i}$, one can write

$$TD^kT^{-1}x_0 = \sum_{i=1}^{n} \alpha_i\rho_i^k e^{j\phi_i k}v_i. \tag{1.6}$$

It holds

$$\begin{aligned} |\lambda_i| &= \rho_i^k |e^{j\phi_i k}| \\ &= \rho_i^k. \end{aligned} \tag{1.7}$$

This helps us defining the following cases:

- $|\lambda_i| < 1 \ \forall i = 1, \dots, n$: the free evolution converges to 0 and the system is **asymptotically stable**.
- $|\lambda_i| \leq 1 \ \forall i = 1, \dots, n$ and eigenvalues with unit modulus have equal geometric and algebraic multiplicity: the free evolution converges (but not to 0) and the system is **marginally stable** or **stable**.
- $\exists i$ s.t. $|\lambda_i| > 1$: the free evolution diverges and the system is **unstable**.

Remark. The same analysis can be performed for non diagonalizable matrices A_d. The same conditions can be derived, with the help of the Jordan diagonal form of A_d.

Example 1. You are given the dynamics

$$x[k+1] = \underbrace{\begin{pmatrix} 0 & 0 \\ 1 & \frac{1}{2} \end{pmatrix}}_{A_\text{d}} x[k], \quad x[0] = \begin{pmatrix} x_{10} \\ x_{20} \end{pmatrix}. \tag{1.8}$$

Since A_d is a lower diagonal matrix, its eigenvalues lie in the diagonal, i.e. $\lambda_1 = 0$, $\lambda_2 = \frac{1}{2}$. Since both eigenvalues satisfy $|\lambda_i| < 1$, the system is asymptotically stable.

1.2 Discrete Time Controller Synthesis

As you learned in class, there are two ways to discretize systems. The scheme in Figure 1 resumes them.

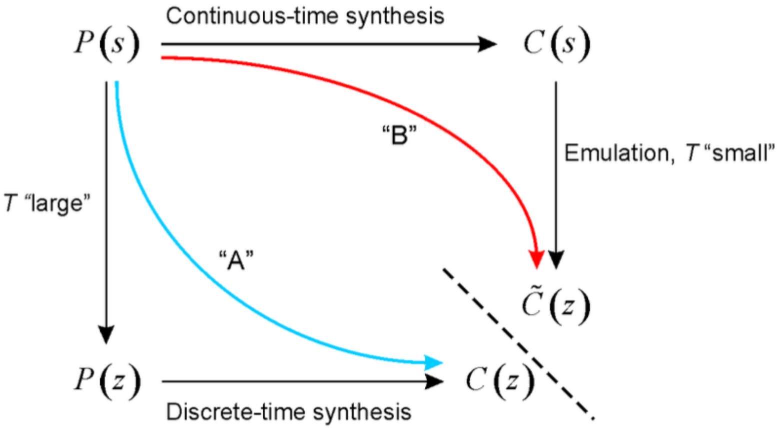

Figure 1: Emulation and Discrete time controller synthesis.

1.2.1 Emulation

In the previous chapter, we learned how to emulate a system. Let's define the recipe for such a procedure: Given a continous-time plant $P(s)$:

1. Design a continuous-time controller for the continuous-time plant.
2. Choose a sampling rate that is at least twice (ten times in practice) the crossover frequency.
3. If required, design an anti aliasing filter (AAF) to remove high frequency components of the continuous signal that is going to be sampled.
4. Modify your controller to take into accout the phase lag introduced by the discretization (up to a sampling period delay) and the AAF.
5. Discretize the controller (e.g. use the Tustin method for best accuracy).
6. Check open loop stability. If the system is unstable, change the emulation method, choose a faster sampling rate, or increase margin of phase.
7. Implement the controller.

1.2.2 Discrete-Time Synthesis

In this chapter we learn how to perform discrete time controller synthesis (also called direct synthesis). The general situation is the following: a control loop is given as in Figure 2. The continuous time transfer function $G(s)$ is given.

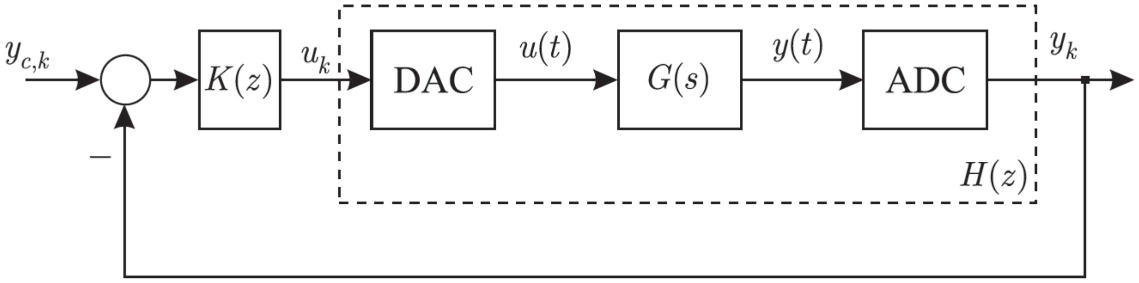

Figure 2: Discrete-time control loop.

We want to compute the equivalent discrete-time transfer function $H(z)$. The loop is composed of a Digital-to-Analog Converter (DAC), the continuous-time transfer function $G(s)$ and of an Analog to Digital Converter (ADC). We aim reaching a structure that looks like the one in Figure 3.
The first thing to do, is to consider an input to analyze. The usual choice for this type of analysis is a **unit-step** of the form

$$u(kT) = \{\ldots, 0, 1, 1, \ldots\}. \tag{1.9}$$

Since the $z-$Transform is defined as

$$X(z) = \sum_{n=0}^{\infty} x(n) \cdot z^{-n} \tag{1.10}$$

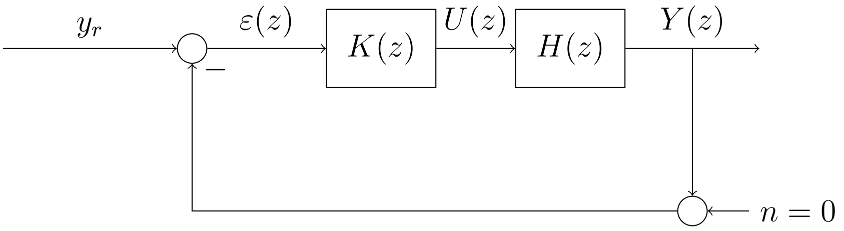

Figure 3: Discrete Syhtesis.

one gets for u

$$U(z) = 1 + z^{-1} + z^{-2} + z^{-3} + \ldots + z^{-n}. \tag{1.11}$$

This sum can be written as (see geometric series)

$$U(z) = \frac{1}{1 - z^{-1}}. \tag{1.12}$$

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

Remark. **Recall: sum of geometric series** Let S_n denote the sum over the first n elements of a geometric series:

$$\begin{aligned} S_n &= U_0 + U_0 \cdot a + U_0 \cdot a^2 + \ldots + U_0 \cdot a^{n-1} \\ &= U_0 \cdot (1 + a + a^2 + \ldots + a^{n-1}). \end{aligned} \tag{1.13}$$

Then

$$a \cdot S_n = U_0 \cdot (a + a^2 + a^3 + \ldots + a^n) \tag{1.14}$$

and

$$S_n - a \cdot S_n = U_0 \cdot (1 - a^n), \tag{1.15}$$

which leads to

$$S_n = U_0 \cdot \frac{1 - a^n}{1 - a}. \tag{1.16}$$

From here, it can be shown that the limit for n going to infinity convergenges if and only if the absolute value of a is smaller than one, i.e.

$$\lim_{n \to \infty} S_n = U_0 \cdot \frac{1}{1 - a}, \textbf{ iff } |a| < 1. \tag{1.17}$$

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

$$\frac{1}{s}. \tag{1.18}$$

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

$$\frac{G(s)}{s}. \tag{1.19}$$

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

$$y(t) = \mathcal{L}^{-1}\left(\frac{G(s)}{s}\right). \tag{1.20}$$

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

$$\begin{aligned} Y(z) &= \mathcal{Z}\left(y(kT)\right) \\ &= \mathcal{Z}\left(L^{-1}\left(\frac{G(s)}{s}\right)\right). \end{aligned} \tag{1.21}$$

The transfer function is then given as

$$H(z) = \frac{Y(z)}{U(z)}. \tag{1.22}$$

Example 2. A continuous-time system with the following transfer function is considered:

$$G(s) = \frac{9}{s+3}. \tag{1.23}$$

(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

$$Y(z) = H(z) \cdot U(z), \tag{1.24}$$

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

$$H(z) = \frac{Y(z)}{U(z)}. \tag{1.25}$$

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

$$\begin{cases} u_k[k] = 1, & k \geq 0 \\ 0, & \text{else.} \end{cases} \tag{1.26}$$

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

$$X(z) = \sum_{n=0}^{\infty} x(n) \cdot z^{-n}. \tag{1.27}$$

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

$$\begin{aligned} U(z) &= X(u_k) \\ &= \sum_{k=0}^{\infty} z^{-k} \\ &= \sum_{k=0}^{\infty} (z^{-1})^k. \end{aligned} \tag{1.28}$$

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

$$U(z) = \frac{1}{a - z^{-1}}, \tag{1.29}$$

as long as the convergence criterion is satisfied, i.e. as long as $|z^{-1}| < 1$ or better $|z| > 1$ ($a = 1$). 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 to for the analysis. The Laplace transform of the step function is well known to be

$$\mathcal{L}(u_h(t))(s) = \frac{1}{s} = U(s). \tag{1.30}$$

The plant output in continuous time is given by

$$\begin{aligned} Y(s) &= G(s) \cdot U(s) \qquad . \\ &= \frac{G(s)}{s}. \end{aligned} \tag{1.31}$$

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 (i.e. into $y(t)$) 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.

$$Y(z) = X\left(\{\mathcal{L}^{-1}\left(\frac{G(s)}{s}\right)(kT)\}\right). \tag{1.32}$$

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

$$\begin{aligned} \frac{G(s)}{s} &= \frac{9}{s \cdot (s+3)} \\ &= \frac{\alpha}{s} + \frac{\beta}{s+3} \\ &= \frac{s \cdot (\alpha + \beta) + 3 \cdot \alpha}{s \cdot (s+3)}. \end{aligned} \tag{1.33}$$

The comparison of the numerators yields

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

and thus

$$\begin{aligned}\frac{G(s)}{s} &= \frac{3}{s} - \frac{3}{s+3}\\ &= 3 \cdot \left(\frac{1}{s} - \frac{1}{s+3}\right).\end{aligned} \tag{1.34}$$

Now the terms can be individually transformed with the result

$$\begin{aligned}\mathcal{L}^{-1}\left(\frac{G(s)}{s}\right) &= 3 \cdot \left(1 - e^{-3t}\right) \cdot u_h(t)\\ &= y(t).\end{aligned} \tag{1.35}$$

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

$$\begin{aligned}X(\{y(kT)\}) &= 3 \cdot \left[\sum_{k=0}^{\infty} z^{-k} - \sum_{k=0}^{\infty} e^{-3kT} \cdot z^{-k}\right]\\ &= 3 \cdot \left[\sum_{k=0}^{\infty} z^{-1k} - \sum_{k=0}^{\infty} (e^{-3T} \cdot z^{-1})^k\right]\\ &= Y(z).\end{aligned} \tag{1.36}$$

From above, the two necessary convergence criteria are known:

$$\begin{aligned}|z^{-1}| < 1 &\Rightarrow |z| > 1\\ |e^{-3T} \cdot z^{-1}| &\Rightarrow |z| > |e^{-3T}|.\end{aligned} \tag{1.37}$$

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

$$Y(z) = 3 \cdot \left(\frac{1}{1 - z^{-1}} - \frac{1}{1 - e^{-3T} \cdot z^{-1}}\right). \tag{1.38}$$

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

$$\begin{aligned}H(z) &= \frac{Y(z)}{U(z)}\\ &= (1 - z^{-1}) \cdot Y(z)\\ &= 3 \cdot \left(1 - \frac{1 - z^{-1}}{1 - e^{-3T} \cdot z^{-1}}\right)\\ &= 3 \cdot \frac{(1 - e^{-3T}) \cdot z^{-1}}{1 - e^{-3T} \cdot z^{-1}}.\end{aligned} \tag{1.39}$$

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

$$\begin{aligned}\varepsilon(z) &= Y_c(z) - Y(z)\\ &= Y_c(z) - H(z) \cdot K(z) \cdot \varepsilon(z)\\ &= \frac{Y_c(z)}{1 + k_p \cdot H(z)}.\end{aligned} \tag{1.40}$$

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

$$Y_c(z) = \frac{1}{1+z^{-1}}. \tag{1.41}$$

Therefore, the error signal reads

$$\varepsilon(z) = \frac{\frac{1}{1+z^{-1}}}{1+3\cdot k_p \cdot \frac{(1-e^{-3T})\cdot z^{-1}}{1-e^{-3T}\cdot z^{-1}}}. \tag{1.42}$$

To calculate the steady-state error, i.e. the error after infinite time, the discrete-time final value theorem[1] is used:

$$\lim_{t\to\infty} \varepsilon(t) = \lim_{z\to 1}(1-z^{-1})\cdot\varepsilon(z), \tag{1.43}$$

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

$$\varepsilon_\infty = \frac{1}{1+3\cdot k_p}. \tag{1.44}$$

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.

[1] $\lim_{t\to\infty}\varepsilon(t) = \lim_{s\to 0} s\cdot\varepsilon(s)$

2 MIMO Systems

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

2.1 System Description

2.1.1 State Space Descriprion

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

$$\begin{aligned} \dot{x}(t) &= A \cdot x(t) + B \cdot u(t), \qquad && x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m \\ y(t) &= C \cdot x(t) + D \cdot u(t), && y(t) \in \mathbb{R}^p \end{aligned} \tag{2.1}$$

where

$$x(t) \in \mathbb{R}^{n\times 1},\ u(t) \in \mathbb{R}^{m\times 1},\ y(t) \in \mathbb{R}^{p\times 1},\ A \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times m},\ C \in \mathbb{R}^{p\times n},\ D \in \mathbb{R}^{p\times m}. \tag{2.2}$$

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 not scalars anymore. For this reason B, C, D are now matrices.

2.1.2 Transfer Function

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

$$P(s) = C \cdot (s \cdot \mathbb{I} - A)^{-1} \cdot B + D. \tag{2.3}$$

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

$$P(s) = \begin{pmatrix} P_{11}(s) & \cdots & P_{1m}(s) \\ \vdots & \ddots & \vdots \\ P_{p1}(s) & \cdots & P_{pm}(s) \end{pmatrix}, \qquad P_{ij}(s) = \frac{b_{ij}(s)}{a_{ij}(s)}. \tag{2.4}$$

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 for the MIMO case B,C,D are matrices, one has to pay attention to a fundamental mathematical property: the matrix multiplication is not **commutative** (i.e. $A \cdot B \neq B \cdot A$). Since now $P(s)$ and $C(s)$ are matrices, considering Figure 4, it holds

$$L_{\mathrm{O}}(s) = P(s) \cdot C(s) \neq C(s) \cdot P(s) = L_{\mathrm{I}}(s), \tag{2.5}$$

where $L_{\mathrm{O}}(s)$ is the outer loop transfer function and $L_{\mathrm{I}}(s)$ is the inner loop transfer function. Moreover, one can no more define the complementary sensitivity and the sensitivity as

$$T(s) = \frac{L(s)}{1 + L(s)},\ S(s) = \frac{1}{1 + L(s)}. \tag{2.6}$$

because no matrix division is defined. There are however similar expressions to describe those transfer functions:

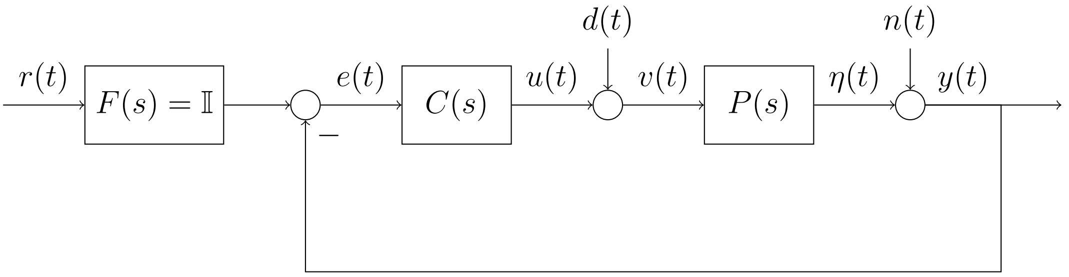

Figure 4: Standard feedback control system structure.

Output Sensitivity Functions

Referring to Figure 5, one can write

$$\begin{aligned}Y(s) &= N(s)+\eta(s)\\ &= N(s)+P(s)V(s)\\ &= N(s)+P(s)\left(D(s)+U(s)\right)\\ &= N(s)+P(s)\left(D(s)+C(s)E(s)\right)\\ &= N(s)+P(s)\left(D(s)+C(s)(R(s)-Y(s))\right),\end{aligned}\tag{2.7}$$

from which follows

$$\begin{aligned}(\mathbb{I}+P(s)C(s))Y(s) &= N(s)+P(s)D(s)+P(s)C(s)R(s)\\ Y(s) &= (\mathbb{I}+P(s)C(s))^{-1}\left(N(s)+P(s)D(s)+P(s)C(s)R(s)\right).\end{aligned}\tag{2.8}$$

It follows

- **Output sensitivity function** $(n\to y)$
$$S_{\mathrm{O}}(s)=(\mathbb{I}+L_{\mathrm{O}}(s))^{-1}.\tag{2.9}$$
- **Output complementary sensitivity function** $(r\to y)$
$$T_{\mathrm{O}}(s)=(\mathbb{I}+L_{\mathrm{O}}(s))^{-1}L_{\mathrm{O}}(s).\tag{2.10}$$

Input Sensitivity Functions

Referring to Figure 5, one can write

$$\begin{aligned}U(s) &= C(s)E(s)\\ &= C(s)\left(R(s)-Y(s)\right)\\ &= C(s)R(s)-C(s)\left(N(s)+\eta(s)\right)\\ &= C(s)R(s)-C(s)N(s)-C(s)P(s)V(s)\\ &= C(s)R(s)-C(s)N(s)-C(s)P(s)\left(D(s)+U(s)\right),\end{aligned}\tag{2.11}$$

from which it follows

$$\begin{aligned}(\mathbb{I}+C(s)P(s))U(s) &= C(s)R(s)-C(s)N(s)-C(s)P(s)D(s)\\ -U(s) &= (\mathbb{I}+C(s)P(s))^{-1}\left(-C(s)R(s)+C(s)N(s)+C(s)P(s)D(s)\right).\end{aligned}\tag{2.12}$$

It follows

- **Input sensitivity function** $(d \to v)$

$$S_{\mathrm{I}}(s) = (\mathbb{I} + L_{\mathrm{I}}(s))^{-1}. \tag{2.13}$$

- **Input complementary sensitivity function** $(d \to -u)$

$$T_{\mathrm{I}}(s) = (\mathbb{I} + L_{\mathrm{I}}(s))^{-1} L_{\mathrm{I}}(s). \tag{2.14}$$

Example 3. In order to understand how to work with these matrices, let's analyze the following problem. We start from the standard control system's structure (see Figure 5). 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

$$Y(s) = T(s) \cdot R(s) + S(s) \cdot D(s). \tag{2.15}$$

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

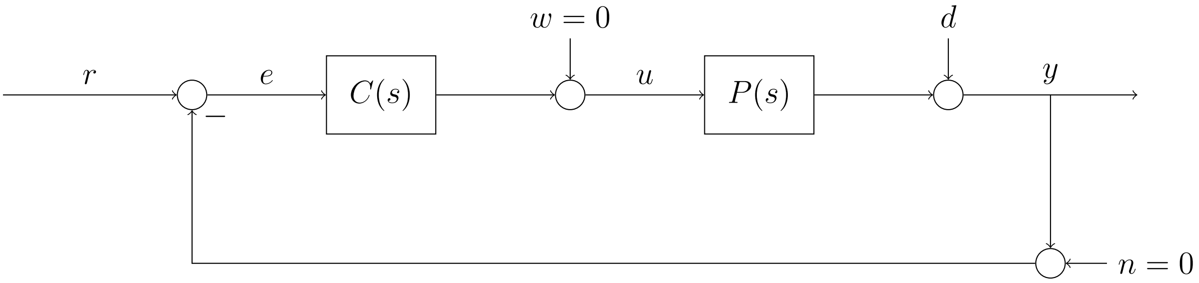

Figure 5: Standard feedback control system structure.

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 $E(s)$

$$\begin{aligned} E(s) &= R(s) - P(s) \cdot C(s) \cdot E(s) - D(s), \\ Y(s) &= P(s) \cdot C(s) \cdot E(s) + D(s). \end{aligned} \tag{2.16}$$

This gives in the first place

$$E(s) = (\mathbb{I} + P(s) \cdot C(s))^{-1} \cdot (R(s) - D(s)). \tag{2.17}$$

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

$$\begin{aligned} Y &= P \cdot C \cdot E + D \\ &= P \cdot C \cdot (\mathbb{I} + P \cdot C)^{-1} \cdot (R - D) + D \\ &= P \cdot C \cdot (\mathbb{I} + P \cdot C)^{-1} \cdot R - P \cdot C\,(\mathbb{I} + P \cdot C)^{-1} \cdot D + \underbrace{(\mathbb{I} + P \cdot C)\,(\mathbb{I} + P \cdot C)^{-1}}_{\mathbb{I}} \cdot D \\ &= P \cdot C \cdot (\mathbb{I} + P \cdot C)^{-1} \cdot R + (\mathbb{I} + P \cdot C - P \cdot C) \cdot (\mathbb{I} + P \cdot C)^{-1} \cdot D \\ &= P \cdot C \cdot (\mathbb{I} + P \cdot C)^{-1} \cdot R + (\mathbb{I} + P \cdot C)^{-1} \cdot D. \end{aligned} \tag{2.18}$$

Recalling the general equation (2.15) one gets the two transfer functions:

$$\begin{aligned} T_1(s) &= P(s)\cdot C(s)\cdot(\mathbb{I}+P(s)\cdot C(s))^{-1}, \\ S_1(s) &= (\mathbb{I}+P(s)\cdot C(s))^{-1}. \end{aligned} \tag{2.19}$$

Starting from the input $U(s)$

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

$$\begin{aligned} U(s) &= C(s)\cdot(R(s)-D(s))-C(s)\cdot P(s)\cdot U(s), \\ Y(s) &= P(s)\cdot U(s)+D(s). \end{aligned} \tag{2.20}$$

This gives in the first place

$$U(s) = (\mathbb{I}+C(s)\cdot P(s))^{-1}\cdot C(s)\cdot(R(s)-D(s)). \tag{2.21}$$

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

$$\begin{aligned} Y &= P\cdot U+D \\ &= P\cdot(\mathbb{I}+C\cdot P)^{-1}\cdot C\cdot(R-D)+D \\ &= P\cdot(\mathbb{I}+C\cdot P)^{-1}\cdot C\cdot R+\left(\mathbb{I}-P\cdot(\mathbb{I}+C\cdot P)^{-1}\cdot C\right)\cdot D. \end{aligned} \tag{2.22}$$

Recalling the general equation (2.15) one gets the two transfer functions:

$$\begin{aligned} T_2(s) &= P(s)\cdot(\mathbb{I}+C(s)\cdot P(s))^{-1}\cdot C(s), \\ S_2(s) &= \mathbb{I}-P(s)\cdot(\mathbb{I}+C(s)\cdot P(s))^{-1}\cdot C(s). \end{aligned} \tag{2.23}$$

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

$$\begin{aligned} S_1 &= S_2 \\ (\mathbb{I}+P\cdot C)^{-1} &= \mathbb{I}-P\cdot(\mathbb{I}+C\cdot P)^{-1}\cdot C \\ \mathbb{I} &= \mathbb{I}+P\cdot C-P\cdot(\mathbb{I}+C\cdot P)^{-1}\cdot C\cdot(\mathbb{I}+P\cdot C) \\ \mathbb{I} &= \mathbb{I}+P\cdot C-P\cdot(\mathbb{I}+C\cdot P)^{-1}\cdot(C+C\cdot P\cdot C) \\ \mathbb{I} &= \mathbb{I}+P\cdot C-P\cdot(\mathbb{I}+C\cdot P)^{-1}\cdot(\mathbb{I}+C\cdot P)\cdot C \\ \mathbb{I} &= \mathbb{I}+P\cdot C-P\cdot C \\ \mathbb{I} &= \mathbb{I} \\ \\ T_1 &= T_2 \\ P\cdot C\cdot(\mathbb{I}+P\cdot C)^{-1} &= P\cdot(\mathbb{I}+C\cdot P)^{-1}\cdot C \\ P\cdot C &= P\cdot(\mathbb{I}+C\cdot P)^{-1}\cdot C\cdot(\mathbb{I}+P\cdot C) \\ P\cdot C &= P\cdot(\mathbb{I}+C\cdot P)^{-1}\cdot(C+C\cdot P\cdot C) \\ P\cdot C &= P\cdot(\mathbb{I}+C\cdot P)^{-1}\cdot(\mathbb{I}+C\cdot P)\cdot C \\ P\cdot C &= C\cdot P \\ \mathbb{I} &= \mathbb{I}. \end{aligned} \tag{2.24}$$

Finally, one can show that

$$\begin{aligned} S(s)+T(s) &= (\mathbb{I}+P\cdot C)^{-1}+P\cdot C\cdot(\mathbb{I}+P\cdot C)^{-1} \\ &= (\mathbb{I}+P\cdot C)\cdot(\mathbb{I}+P\cdot C)^{-1} \\ &= \mathbb{I}. \end{aligned} \tag{2.25}$$

2.2 Poles and Zeros

Since we have to deals 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.

Example 4. The minors of a given matrix

$$P(s) = \begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix} \tag{2.26}$$

are:
First order:

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

Second order (maximal minors):

$$det\begin{pmatrix} a & b \\ d & e \end{pmatrix}, \; det\begin{pmatrix} a & c \\ d & f \end{pmatrix}, \; det\begin{pmatrix} b & c \\ e & f \end{pmatrix}. \tag{2.27}$$

From the minors one can calculate the poles and the zeros as follows:

2.2.1 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*).

2.2.2 Poles

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

2.2.3 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 $\delta_{\pi,i}^{in,out}$ associated with a pole π_i are defined by

$$P(s)\big|_{s=\pi_i} \cdot \delta_{\pi,i}^{\text{in}} = \infty \cdot \delta_{\pi,i}^{\text{out}}. \tag{2.28}$$

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

$$P(s)\big|_{s=\xi_i} \cdot \delta_{\xi,i}^{\text{in}} = 0 \cdot \delta_{\xi,i}^{\text{out}}. \tag{2.29}$$

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

Example 5. 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

$$\pi_1 = -2$$
$$\pi_2 = -3.$$

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

$$\zeta_1 = -1$$
$$\zeta_2 = -2$$
$$\zeta_3 = -3.$$

Example 6. 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

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

The greatest common divisor of these is

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

Hence, the zeros are

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

References

[1] Analysis III notes, ETH Zurich.

[2] Karl Johan Amstroem, Richard M. Murray *Feedback Systems for Scientists and Engineers*. Princeton University Press, Princeton and Oxford, 2009.

[3] Sigurd Skogestad, *Multivariate Feedback Control*. John Wiley and Sons, New York, 2001.