Lab Communications Project 2

LTI-Systems: State Equations and Simulation

1. Preface

This lab concerns the state-space representation of linear time invariant systems, which are very important not only for the system theory but for automatic control, too. In a description of the input and output of a system with a frequency response or the impulse response the internal behaviour of a system is irrelevant. In contrast to that, the state space description provides insight into the system which is described through so-called *state variables*.

The purpose of all parts of this lab is the simulation of state-space described systems which are given or have to be calculated. The simulation of time-discrete systems on a – also discrete working – computer is possible without any problems, while time-continuous systems must be properly transformed in a discrete representation. Due to the limited range of numbers inside the computer, the discrete simulation contains numerical difficulties, as shown in task 4.4.

2. Fundamentals

In the following subsections the mathematical and system theoretical relationships are shortly summarized.

2.1. State-space Descriptions

The state-space description of a linear time invariant system with L inputs and R outputs in case of time-continuity can be written as

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\,\boldsymbol{x}(t) + \boldsymbol{B}\,\boldsymbol{v}(t), \tag{1}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\,\boldsymbol{x}(t) + \boldsymbol{D}\,\boldsymbol{v}(t), \tag{2}$$

and in case of a time discrete system

$$\boldsymbol{x}(n+1) = \boldsymbol{A}\,\boldsymbol{x}(n) + \boldsymbol{B}\,\boldsymbol{v}(n), \tag{3}$$

$$\boldsymbol{y}(n) = \boldsymbol{C}\,\boldsymbol{x}(n) + \boldsymbol{D}\,\boldsymbol{v}(n). \tag{4}$$

In a time-continuous system the vector

$$\boldsymbol{x}(t) = \Big[x_0(t),\, x_1(t), \ldots, x_{N-1}(t)\Big]^{\mathrm{T}}$$

represents the so-called *state-space vector*, which contains the particular state-space variables $x_i(t)$, $i = 0, \ldots, N-1$. The dimension N denotes the order of the system, in other words the number of independent memory cells. The L input signals $v_l(t)$, $l = 0, \ldots, L-1$ become the *input signal vector*

$$\boldsymbol{v}(t) \;=\; \Big[v_0(t),\, v_1(t), \ldots, v_{L-1}(t)\Big]^{\mathrm{T}}$$

and the R output signals $y_r(t)$, $r = 0, \ldots, R-1$ become the *output signal vector*

$$\boldsymbol{y}(t) \;=\; \Big[y_0(t),\, y_1(t), \ldots, y_{R-1}(t)\Big]^{\mathrm{T}}.$$

The *definition of the system state* can now be formulated as: The state of a system at time $t = t_0$ is defined by all elements of the vector $\boldsymbol{x}(t_0) = [x_0(t_0),\, x_1(t_0), \ldots, x_{N-1}(t_0)]^{\mathrm{T}}$. The knowledge of these states as well as the input signals $\boldsymbol{v}(t \geq t_0)$ is enough to determine the system reaction $\boldsymbol{y}(t \geq t_0)$ for all times $t \geq t_0$. The corresponding items and definitions for time-discrete systems are obtained by the substitutions $t \to n$ bzw. $t_0 \to n_0$. The matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ in their entirety are called *state-space matrices*. The meaning and dimension of the individual matrices are summarized in the following table:

Matrix	Dimension	Meaning for the system
$\boldsymbol{A}$	$N \times N$	*System matrix*, describes the behaviour of the system without input (*eigen behaviour*)
$\boldsymbol{B}$	$N \times L$	*Input matrix*, describes the connection of the system states with the input $\boldsymbol{v}$ (*steering of the system*)
$\boldsymbol{C}$	$R \times N$	*Observation matrix*, describes the coupling of the system states $\boldsymbol{x}$ with the outputs, (*system observation*)
$\boldsymbol{D}$	$R \times L$	*Pass through matrix*, direct connection of the input with the output

Fig. 1 and 2 show the vectorial signal-flow graph for a continuous respectively a discrete system.

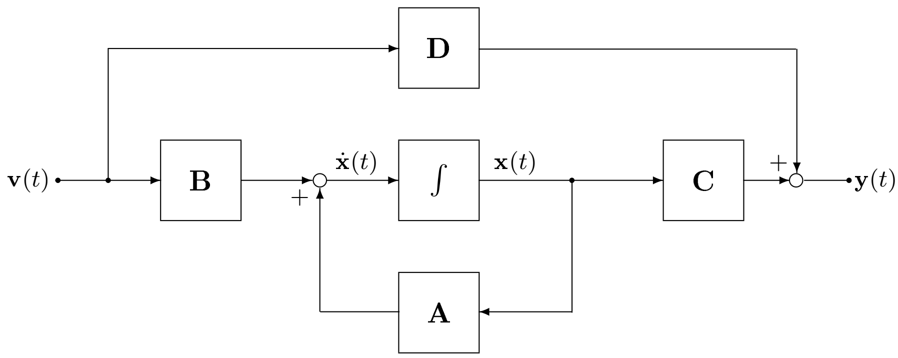

Fig. 1: Vectorial signal-flow graph of the state equations (1) and (2) for continuous systems.

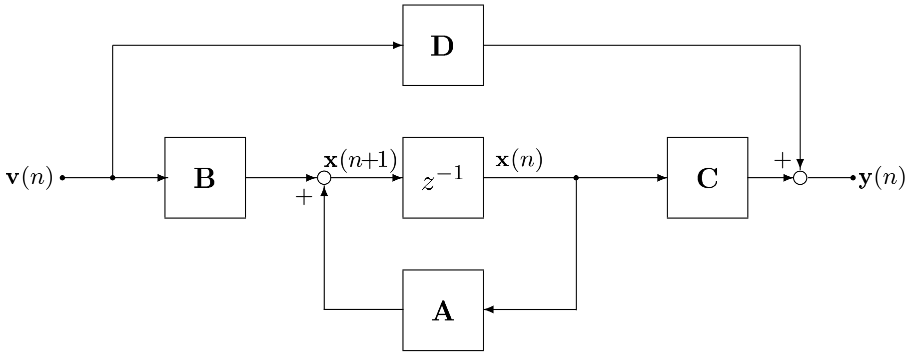

Fig. 2: Vectorial signal-flow graph of the state equations (3) and (4) for discrete systems.

2.2. Transfer Matrix and Impulse-response Matrix

Based on the state-space matrices the transfer matrix of a LTI-system can be calculated by

$$\boldsymbol{H}(s) = \boldsymbol{C}\left[s\boldsymbol{E} - \boldsymbol{A}\right]^{-1}\boldsymbol{B} + \boldsymbol{D} \tag{5}$$

with

$$\boldsymbol{H}(s) = \begin{bmatrix} H_{0,0}(s) & H_{0,1}(s) & \cdots & H_{0,L-1}(s) \\ H_{1,0}(s) & H_{1,1}(s) & \cdots & H_{1,L-1}(s) \\ \vdots & \vdots & \ddots & \vdots \\ H_{R-1,0}(s) & H_{R-1,1}(s) & \cdots & H_{R-1,L-1}(s) \end{bmatrix} \tag{6}$$

for continuous systems and

$$\boldsymbol{H}(z) = \boldsymbol{C}\left[z\boldsymbol{E} - \boldsymbol{A}\right]^{-1}\boldsymbol{B} + \boldsymbol{D} \tag{7}$$

with

$$\boldsymbol{H}(z) = \begin{bmatrix} H_{0,0}(z) & H_{0,1}(z) & \cdots & H_{0,L-1}(z) \\ H_{1,0}(z) & H_{1,1}(z) & \cdots & H_{1,L-1}(z) \\ \vdots & \vdots & \ddots & \vdots \\ H_{R-1,0}(z) & H_{R-1,1}(z) & \cdots & H_{R-1,L-1}(z) \end{bmatrix} \tag{8}$$

for discrete systems (see lecture). One element of the transfer matrix $H_{r,l}(s)$ resp. $H_{r,l}(z)$ represents the transfer function between input l and output r of the system.
Important: All elements of $\boldsymbol{H}$ have different numerator polynomials $Z_{r,l}(s)$ resp. $Z_{r,l}(z)$ but a *common* denominator polynomial $N(s)$ resp. $N(z)$,

$$H_{r,l}(s) = \frac{Z_{r,l}(s)}{N(s)}, \quad H_{r,l}(z) = \frac{Z_{r,l}(z)}{N(z)}. \tag{9}$$

In a continuous system, Eq. (9) can be explained as follows (in analogy to a discrete system with $s \to z$):
Consider Eq. (5) where $\left[s\boldsymbol{E} - \boldsymbol{A}\right]^{-1}$ is the only factor containing s. The inverse of a Matrix $\boldsymbol{M} \in \mathbb{R}^{(N,N)}$ can be described as

$$\boldsymbol{M}^{-1} = \frac{1}{\det \boldsymbol{M}}\left[(-1)^{i+j}\det \boldsymbol{M}_{ji}\right], \quad i = 0, \ldots, N-1, \quad j = 0, \ldots, N-1 \tag{10}$$

(compare with lectures in basic mathematics). There the sub determinants $\boldsymbol{M}_{ji}$ are generated by discarding the j-th row and i-th column of $\boldsymbol{M}$. By replacing $\boldsymbol{M}$ through $[s\boldsymbol{E} - \boldsymbol{A}]$, the only component of the denominator becomes det $\{s\boldsymbol{E} - \boldsymbol{A}\}$. This expression represents a polynomial in s and therefore it is identical for all matrix elements of $[s\boldsymbol{E} - \boldsymbol{A}]^{-1}$ and hence for $\boldsymbol{H}(s)$. With $\boldsymbol{h}_0(t) = \mathcal{L}^{-1}\{\boldsymbol{H}(s)\}$ respectively $\boldsymbol{h}_0(n) = \mathcal{Z}^{-1}\{\boldsymbol{H}(z)\}$ the impulse response matrix can be calculated from the state description as (compare with lecture):
For continuous systems:

$$\boldsymbol{h}_0(t) = \boldsymbol{C}\,\mathcal{L}^{-1}\left\{[s\boldsymbol{E} - \boldsymbol{A}]^{-1}\right\}\,\boldsymbol{B} + \boldsymbol{D}\,\delta_0(t) \tag{11}$$

with the impulse response matrix

$$\boldsymbol{h}_0(t) = \begin{bmatrix} h_{0_{0,0}}(t) & h_{0_{0,1}}(t) & \cdots & h_{0_{0,L-1}}(t) \\ h_{0_{1,0}}(t) & h_{0_{1,1}}(t) & \cdots & h_{0_{1,L-1}}(t) \\ \vdots & \vdots & \ddots & \vdots \\ h_{0_{R-1,0}}(t) & h_{0_{R-1,1}}(t) & \cdots & h_{0_{R-1,L-1}}(t) \end{bmatrix}, \tag{12}$$

and thus for a discrete system

$$\boldsymbol{h}_0(n) = \boldsymbol{C}\,\mathcal{Z}^{-1}\left\{[z\boldsymbol{E} - \boldsymbol{A}]^{-1}\right\}\,\boldsymbol{B} + \boldsymbol{D}\,\gamma_0(n) \tag{13}$$

with the impulse response matrix

$$\boldsymbol{h}_0(n) = \begin{bmatrix} h_{0_{0,0}}(n) & h_{0_{0,1}}(n) & \cdots & h_{0_{0,L-1}}(n) \\ h_{0_{1,0}}(n) & h_{0_{1,1}}(n) & \cdots & h_{0_{1,L-1}}(n) \\ \vdots & \vdots & \ddots & \vdots \\ h_{0_{R-1,0}}(n) & h_{0_{R-1,1}}(n) & \cdots & h_{0_{R-1,L-1}}(n) \end{bmatrix}. \tag{14}$$

2.3. Canonical Realization of Signal-flow Graphs

Important for this lab are only the *first*, *second* and *forth* canonical form, while for more (canonical and non-canonical) realizations of signal-flow graphs the reader is referred to the lecture. Canonical realizations stand out through independent energy storage within their signal-flow graph (N: Order of the system), while non-canonical realizations always contain a high number of these storage. Systems with L inputs and R outputs can be described with $L \cdot R$ signal-flow graphs, which belong to $L \cdot R$ subsystems and connect all inputs and outputs. All subsystems use the same state vector (that is to say they possess the same state-space matrix $\boldsymbol{A}$), hence the $L \cdot R$ signal-flow graphs can be rewritten to a single signal-flow graph. This signal-flow graph represents a canonical realization, if the state representations of the subsystems are given in a canonical form. An example for this is shown in part 4.3. In the following we consider systems with only *one* in- and output, so $L = 1$, $R = 1$.

The formulation of a signal-flow graph generally starts at the difference equation resp. the differential equation. Thus a continuous system of order N is described by

$$\begin{aligned} y^{(N)}(t) + \beta_{N-1}y^{(N-1)}(t) + \cdots + \beta_1\dot{y}(t) + \beta_0 y(t) = \\ \alpha_N v^{(N)}(t) + \alpha_{N-1}v^{(N-1)}(t) + \cdots + \alpha_1\dot{v}(t) + \alpha_0 v(t) \end{aligned} \tag{15}$$

with

$$v^{(n)}(t) = \frac{d^n y(t)}{(dt)^n} \tag{16}$$

and a discrete system by

$$\begin{aligned} y(k+N) + \beta_{N-1}y(k+N-1) + \cdots + \beta_1 y(k+1) + \beta_0 y(k) = \\ \alpha_N v(k+N) + \alpha_{N-1}v(k+N-1) + \cdots + \alpha_1 v(k+1) + \alpha_0 v(k) \end{aligned} \tag{17}$$

where the coefficient β_N is set to one without loss of generality. The following presented realizations describe the *same* system with the transfer function $H(s)$ resp. $H(z)$. They differ only by the internal representation of the states.

2.3.1. First canonical realization

The first canonical realization of a LTI-System is shown in Fig. 3, hence g has to be replaced by z. Furthermore the figure shows the state variables x_1 until x_N and the coefficients α_0 until α_N and β_0 until β_{N-1} from Eq. (15) resp. Eq. (17). Considering the state description

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}_1\, \boldsymbol{x}(t) + \boldsymbol{b}_1\, v(t), \\ y(t) &= {\boldsymbol{c}_1}^{\mathrm{T}}\, \boldsymbol{x}(t) + d_1\, v(t), \end{aligned}$$

resp.

$$\begin{aligned} \boldsymbol{x}(n+1) &= \boldsymbol{A}_1\, \boldsymbol{x}(n) + \boldsymbol{b}_1\, v(n), \\ y(n) &= {\boldsymbol{c}_1}^{\mathrm{T}}\, \boldsymbol{x}(n) + d_1\, v(n). \end{aligned}$$

the state matrices of the first canonical form can be got from Fig. 3

$$\boldsymbol{A}_1 = \begin{bmatrix} -\beta_{N-1} & 1 & 0 & \cdots & 0 \\ -\beta_{N-2} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\beta_1 & 0 & 0 & \cdots & 1 \\ -\beta_0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad \boldsymbol{b}_1 = \begin{bmatrix} \alpha_{N-1} - \beta_{N-1}\alpha_N \\ \alpha_{N-2} - \beta_{N-2}\alpha_N \\ \vdots \\ \alpha_1 - \beta_1\alpha_N \\ \alpha_0 - \beta_0\alpha_N \end{bmatrix}, \quad \boldsymbol{c}_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad d_1 = \alpha_N \tag{18}$$

(Proof this equations!)

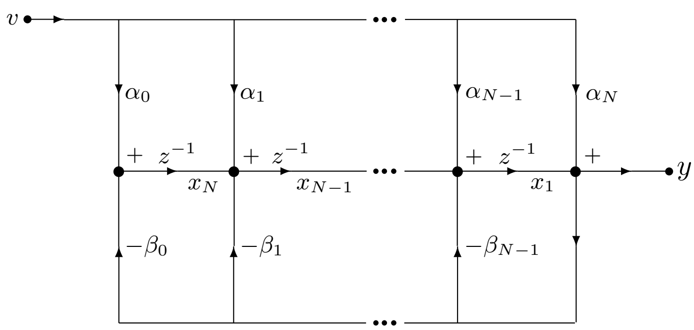

Fig. 3: First canonical realization.

2.3.2. Second Canonical Realization

The second canonical form results by *mirroring* or *transposing* of the first canonical realization. That is to say by changing places of inputs and outputs and reversing of all arrows, branch- and summation knots. Using the signal-flow graph from Fig. 4 and the state description of the first canonical form in Eq. (18), the state matrices can be written as

$$\boldsymbol{A}_2 = \boldsymbol{A}_1{}^{\mathrm{T}}, \quad \boldsymbol{b}_2 = \boldsymbol{c}_1, \quad \boldsymbol{c}_2 = \boldsymbol{b}_1, \quad d_2 = d_1. \tag{19}$$

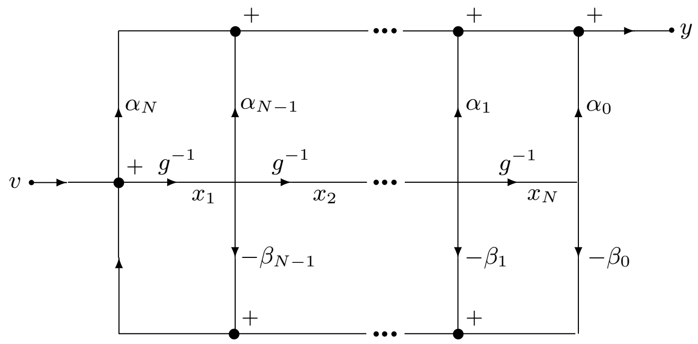

Fig. 4: Second canonical realization, resulted by mirroring of the first canonical form.

2.3.3. Forth Canonical Realization

The forth canonical realization is often called *parallel form* or *diagonal form*; These names indicate to the special structure of the signal-flow graph resp. the state-space matrix $\boldsymbol{A}$. This realization is obtained by partial fraction decomposition of a fractional, rational transfer function, shown here for the discrete system with *single* poles $z_{\infty i}, i = 0, \ldots, N-1$:

$$\begin{aligned} H(z) &= \frac{\sum\limits_{j=0}^{N} \alpha_j z^j}{\sum\limits_{k=0}^{N} \beta_k z^k} \\ &= B_0 + \sum_{i=1}^{N} \frac{B_i}{z - z_{\infty i}} \end{aligned} \tag{20}$$

Out of this follows the signal-flow graph in Fig 5, from where the name "parallel form" is explained. Assuming complete controlability, notice, that the states x_0 till x_{N-1} are decoupled from each other, i.e. a state x_i is influenced only from his own past and the input signal, but not from other states! The state-space matrices can be obtained from the signal-flow graph:

$$\boldsymbol{A}_4 = \begin{bmatrix} z_{\infty 1} & 0 & \cdots & 0 \\ 0 & z_{\infty 2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & z_{\infty n} \end{bmatrix}, \quad \boldsymbol{b}_4 = \begin{bmatrix} B_1{}' \\ B_2{}' \\ \vdots \\ B_n{}' \end{bmatrix}, \quad \boldsymbol{c}_4 = \begin{bmatrix} C_1{}' \\ C_2{}' \\ \vdots \\ C_n{}' \end{bmatrix}, \quad d_4 = B_0 \tag{21}$$

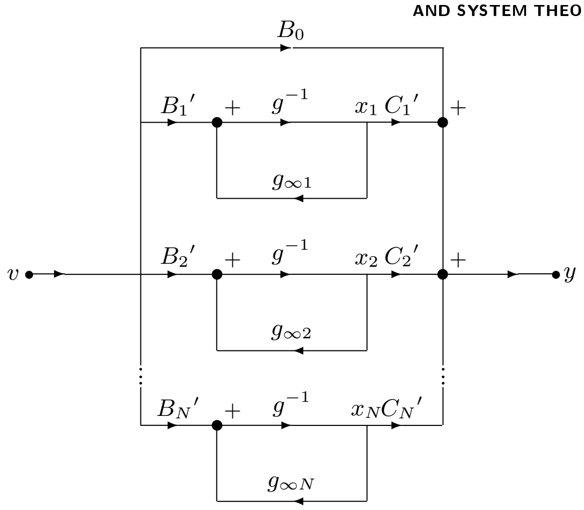

Fig. 5: Forth canonical realization (parallel or diagonal form) for n *single* poles with $B_i = B_i{}' \cdot C_i{}'$ for Eq. (20). The single states x_k are decoupled from each other.

The state-space matrix $\boldsymbol{A}$ is reduced to a pure diagonal matrix of the pole locations $z_{\infty i}$ ("diagonal form"). The B_i from Eq. (20) are here expressed as

$$B_i = B_i{}'\, C_i{}', \quad i = 0, \ldots, N-1. \tag{22}$$

At first this decomposition of B_i seems to be needless, but will be of importance when we look on controlability and observability in Section 2.5. However, this description is naturally valid for continuous systems, replace z through s for a continuous description.

2.4. Transformation of State-space Variables

In the last section it was mentioned, that different forms of realization describe the same system by possessing different state-space representations. Often it is useful to transform one state-space representation to another one (e.g. to the parallel form, see below). We will see in task 4.4 that some realizations are more numerical insensitive than others.

We are looking for a linear transformation $\boldsymbol{T}$, which creates from a given state-space representation with $\boldsymbol{x}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}$ a new representation with $\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{A}}, \tilde{\boldsymbol{B}}, \tilde{\boldsymbol{C}}, \tilde{\boldsymbol{D}}$. Inserting

$$\boldsymbol{x} = \boldsymbol{T}\,\tilde{\boldsymbol{x}}, \quad \tilde{\boldsymbol{x}} = \boldsymbol{T}^{-1}\,\boldsymbol{x}, \tag{23}$$

in the state-space representation according to Eq. (1) (here for discrete systems),

$$\begin{aligned} \boldsymbol{T}\tilde{\boldsymbol{x}}(n+1) &= \boldsymbol{A}\boldsymbol{T}\tilde{\boldsymbol{x}}(n) + \boldsymbol{B}\boldsymbol{v}(n) \\ \tilde{\boldsymbol{x}}(n+1) &= \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}\tilde{\boldsymbol{x}}(n) + \boldsymbol{T}^{-1}\boldsymbol{B}\boldsymbol{v}(n), \end{aligned}$$

and Eq. (2),

$$\boldsymbol{y}(n) = \boldsymbol{C}\boldsymbol{T}\tilde{\boldsymbol{x}}(n) + \boldsymbol{D}\boldsymbol{v}(n) \tag{24}$$

the transformed state-space matrices follow as

$$\tilde{\boldsymbol{A}} = \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}, \quad (25)$$
$$\tilde{\boldsymbol{B}} = \boldsymbol{T}^{-1}\boldsymbol{B}, \quad (26)$$
$$\tilde{\boldsymbol{C}} = \boldsymbol{C}\boldsymbol{T}, \quad (27)$$
$$\tilde{\boldsymbol{D}} = \boldsymbol{D}. \quad (28)$$

In this lab given state-space representations should be transformed to the parallel form, which contains a state matrix $\boldsymbol{A}$ in diagonal form. So we are looking for a matrix $\boldsymbol{T}$ which creates a diagonal matrix $\tilde{\boldsymbol{A}} = \mathrm{diag}\{z_{\infty i}\}$, $i = 0, \ldots, N-1$ as long as $\boldsymbol{T}$ is applied to $\boldsymbol{A}$ according (25). It is well known from the linear algebra, that this happens, if $\boldsymbol{T}$ consists of column-wise ordered *Eigenvectors* from $\boldsymbol{A}$! Then the originating diagonal matrix consists of the *Eigenvalues* from $\boldsymbol{A}$, which are identical with the pole locations $z_{\infty i}$ of the system (assuming simple poles) [1]:

$$\boldsymbol{T} = \left[\boldsymbol{t}_i\right] \text{ mit } \boldsymbol{A}\,\boldsymbol{t}_i = z_{\infty i}\,\boldsymbol{t}_i \quad (29)$$

2.5. Controlability and Observability

The system matrix $\boldsymbol{A}$ can always be transformed in a diagonal matrix $\tilde{\boldsymbol{A}}$, where the roots of the denominator stay on the main diagonal. These roots of the denominator $z_{\infty i}$, $i = 0, \ldots, N-1$ are pole locations of each single transfer function $H_{r,l}(s)$ resp. $H_{r,l}(z)$ from equations (6) resp. (8), on condition that the numerator polynomial is not getting zero at that location, i.e. $z_{\infty i} = z_{0i}$. In such a situation the state-space representation of a system has got N states, but the belonging transfer function $H_{r,l}$ only the denominator order $N_{r,l} < N$. — In other words: Not all eigenvalues of $\tilde{\boldsymbol{A}}$ are pole locations of $\boldsymbol{H}$, whereas the remaining eigenvalues can get lost only by multiplication of $\tilde{\boldsymbol{A}}$ with $\tilde{\boldsymbol{B}}$ and/or $\tilde{\boldsymbol{C}}$ (see Fig. 1 resp. 2). In such a case either missing *controlability* or *observability* of one or more states occurs. That is explained for discrete systems in the following (mind the analogy to continuous systems).

A system is *completely controlable*, if each beginning state $\boldsymbol{x}(n_0)$ can be advanced in any ending state $\boldsymbol{x}(n_1)$ (with $n_1 > n_0$, i.e. to a later time step) by an input signal $\boldsymbol{v}(l)$. Starting from the parallel form (look above), this means, that each input of the system has to be connected with each state. This connection is made according Fig. 2 with the "System control matrix" $\boldsymbol{B}$. Transformation of this matrix to the parallel form leads to a matrix $\tilde{\boldsymbol{B}}$, and it follows: The n-th state is not controllable by the l-th input, if the n-th row of $\tilde{\boldsymbol{B}}$ contains a zero at the l-th position, since the matrix product $\tilde{\boldsymbol{B}} \cdot \boldsymbol{v}$ from Eq. (3) leads to a vector which does not contain neither value of input v_l at the n-th position ($n = 0, \ldots, N-1$, $l = 0, \ldots, L-1$). A system is *completely observable*, if in any later time step ($n_1 > n_0$) each state $\boldsymbol{x}(n_0)$ can be reproduced from knowledge of the output signal $\boldsymbol{y}(n_1)$, without any input signal (i.e. $\boldsymbol{v}(n) = \boldsymbol{0}$). For this purpose each state within the system realization as a parallel form must be connected with each output of the system. Derived from Fig. 2 this connection is made by the "system observation matrix" $\boldsymbol{C}$. Transformation of this matrix to the parallel form leads to a matrix $\tilde{\boldsymbol{C}}$, and it follows analogical:

The n-th state is not observable by the r-th output, if the r-th row of $\tilde{\boldsymbol{C}}$ contains a zero at the n-th position, since the matrix product $\tilde{\boldsymbol{C}} \cdot \boldsymbol{x}$ from Eq. (4) leads to a vector

[1] Of course, the same is valid for continuous systems with the substitution $z \rightarrow s$.

which does not contain neither value of input x_n at the r-th position ($r = 0, \ldots, R-1$, $n = 0, \ldots, N-1$).

3. Lab Preparation

3.1. Discrete System

Assume that we have a causal system with the transfer function

$$H(z) = \frac{z^2 - 4.5z - 2.5}{z^2 - 5.5z + 2.5}. \tag{30}$$

(a) Calculate the poles and roots of $H(z)$. Is the system stable? What is the impulse response $h(n) \circ\!\!-\!\!\bullet\, H(z)$?

(b) Determine the state-space matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ in the first and second canonical form and draw the corresponding signal-flow graph.

(c) Transform the second canonical form to the parallel form. For this calculate the transformation matrix $\mathbf{T}$ at first and perform the transformation by matrix multiplication. Draw also the signal-flow graph of the parallel form.

***Hint* :** Use Eq. (10) for inversion of the transformation matrix $\mathbf{T}$.

***Hint* 2 :** Calculate the eigenvectors of $\mathbf{A}$ such that the sum of squared elements becomes equal to 1.

***Important* :** Calculate with *closed* form notations of numbers, and avoid rounding. (e.g. $\sqrt{3}/2$ instead of $0.866\ldots$).

3.2. Analog Lowpass Filter

Fig. 6 shows the circuit of an analog lowpass filter.

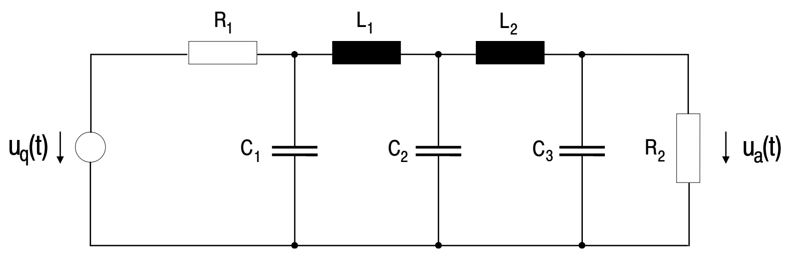

Fig. 6: Circuit of the analog filter.

(a) What is the system order? Which order would the system have, if there was a capacity parallel to each inductivity?

(b) Determine the state and output equations, where $u_q(t)$ resp. $u_a(t)$ represent the input resp. output time function. State the state matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$.

***Hint* :** As a state vector suits e.g.

$$\boldsymbol{x}(t) = \begin{bmatrix} i_{L_1}(t) & i_{L_2}(t) & u_{C_1}(t) & u_{C_2}(t) & u_{C_3}(t) \end{bmatrix}^T. \tag{31}$$

4. Lab Execution

Important: *All* results must be printed for later studies!

4.1. Simulation of an Analog Filter

Assume we have the analog filter from Fig. 6 in Sec. 3.2 of the preparation. A so-called *Butterworth–Filter* is obtained, if the electronic components are chosen e.g. as follows:

$$\begin{aligned} R_1 &= R_2 = 1\,\mathrm{k}\Omega, \\ C_1 &= C_3 = 88.2\,\mathrm{nF}, \\ L_1 &= L_2 = 0.231\,\mathrm{H}, \\ C_2 &= 285\,\mathrm{nF}. \end{aligned} \tag{32}$$

1. Perform a resistor and frequency normalization of the electronic components specified above according

$$R' = \frac{R}{R_\mathrm{N}} = 1, \quad C' = 2\pi f_\mathrm{P}\, R_\mathrm{N}\, C, \quad L' = \frac{2\pi f_\mathrm{P}\, L}{R_N}, \tag{33}$$

   where the normalized resistor and the critical pass frequency are $R_\mathrm{N} = 1\,\mathrm{k}\Omega$ and $f_\mathrm{P} = 1\,\mathrm{kHz}$. Explain the sense of such a normalization!

2. Simulate all state variables and outputs by stimulation with an impulse and a step for a period of 20 s, and choose the time-step width T between two time-steps as $T = 0.01\,\mathrm{s}$ (Command `impulse_new`, `lsim_new`). Use the calculated state matrices from your preparation and the *normalized* values from part 1. Plot the impulse-response and step-response as well as all state-space variables (Command `subplot`!) and discuss your results.

4.2. Time Continuous System and Frequency Response

A time continuous system $H(s) = Z(s)/N(s)$ is given in a state-space description:

$$\boldsymbol{A} = \begin{bmatrix} -0.86224 & 1 & 0 & 0 & 0 \\ -1.6217 & 0 & 1 & 0 & 0 \\ -0.87895 & 0 & 0 & 1 & 0 \\ -0.53661 & 0 & 0 & 0 & 1 \\ -0.10825 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.10825 \end{bmatrix}, \tag{34}$$

$$\boldsymbol{c} = [1 \quad 0 \quad 0 \quad 0 \quad 0]^T, \qquad d = 0. \tag{35}$$

1. Which canonical form describes this state description?

   If possible, make a prediction about the controlability and/or observability of the system! Give reasons for your answer!

2. Simulate and plot the impulse and step response in a time period of 70 s with $T = 0.1\,\mathrm{s}$ (Commands `impulse_new`, `lsim_new`). Make a presumption on the basis of the impulse and step response, whether the system describes a highpass or a lowpass. Give reasons for your answer!

3. Simulate the reaction to sine signals with the frequencies $f = 0.08\,\text{Hz}$ and $f = 0.4\,\text{Hz}$ in $70\,\text{s}$, and set each input and output together in a common plot. (Commands `lsim_new`, `subplot`). Compare the results with your presumptions from part 2.

4. Determine and plot the magnitude frequency response of the system (Commands `freqs`, `ss2tf`) and validate your presumptions from parts 2 and 3. Select the frequency axis $\omega = [0 : 0.001 : 0.5] * 2\pi$.

4.3. Controlability and Observability

A time continuous system with two inputs and two outputs has got the following state-space description:

$$\boldsymbol{A} = \begin{bmatrix} -4 & 1 \\ -3 & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{D} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{36}$$

1. In which canonical form is the state-space description given?

2. Transform the state-space matrices to the parallel form (use the command `eig` for calculation of the transformation matrix) and illustrate the signal-flow graph (command `sfg(A,B,C,D)`). Decide whether the system is completely controllable and observable!

   Why is a system representation in the parallel form for the evaluation of control and observability better suited as a representation in another (canonical) form?

3. Determine the impulse responses of the system between all inputs by simulation over $5\,\text{s}$ with the step size $T = 0.01\,\text{s}$ (command `impulse_new`). Plot all impulse responses in one diagram (command `subplot`). Furthermore calculate the transfer matrix $\boldsymbol{H}(s)$ (command `ss2tf`).

4. The system above will be modified now, by replacing the matrices $\boldsymbol{B}$ und $\boldsymbol{C}$ by

$$\boldsymbol{B} = \begin{bmatrix} 1 & 1 \\ 1 & 3 \end{bmatrix} \quad \text{und} \quad \boldsymbol{C} = \begin{bmatrix} -3 & 1 \\ 1 & -1 \end{bmatrix} \tag{37}$$

   Matrices $\boldsymbol{A}$ und $\boldsymbol{D}$ remain unchanged. Perform subtasks 2 and 3 again with the changed system. Which changes concerning controlability and observability of the system do you notice?

4.4. Numerical Sensitivity in the Different Forms of Realization

1. a) Simulate the impulse response $h(n)$ of the discrete system $H(z)$ from Section 3.1 of the preparation. Use the *second* canonical form and select 100 samples ($0 \le k \le 99$). Plot the obtained impulse response $h(n)$ and state variables x_1 and x_2 in one MATLAB-Figure (Command `subplot`). Please note that the input signal must have a length of 100 samples (see Appendix A). What result do you get? (Commands among others: `dlsim_new`, `displot`, `subplot`)

   b) Visualize the signal-flow graph with `sfg(A,B,C,D)`.

2. a) Transform the state matrices of the second canonical form to the parallel form (use the command `eig` for calculating the transformation matrix) and compare the result with your calculation from Section 3.1(c) of the preparation.

   b) Again simulate the first 100 samples of the impulse response and plot the impulse response and state-space variables.

   c) Visualize the signal-flow graph with `sfg(A,B,C,D)`.

   Explain the observations you make!

3. Now type in the exact state matrices of the parallel form, which you have calculated in 3.1(c) and repeat the simulation and graphical illustration from part 2(b) and (c).

4. Try to explain the different behaviour of the three simulated forms of realization by comparing the signal-flow graphs of parts 1, 2 and 3.

A. Additional MATLAB-commands

dlsim_new Simulation of discrete LTI-systems

Call: `[Y,X] = dlsim_new(A,B,C,D,U)`

Parameters:

- `A` : $N \times N$ System matrix
- `B` : $N \times L$ Input matrix
- `C` : $R \times N$ Observation matrix
- `D` : $R \times L$ Pass through matrix
- `U` : System inputs, $Q \times L$ matrix (Q: Length of input signals)
- `Y` : System reaction, $Q \times R$ matrix
- `X` : Time course of the states, $Q \times N$ matrix

`dlsim_new` simulates the reaction of the time discrete LTI-system with order N (L inputs, R outputs) in the state-space description

$$\begin{aligned} \boldsymbol{x}(n+1) &= \boldsymbol{A}\,\boldsymbol{x}(n) + \boldsymbol{B}\,\boldsymbol{v}(n) \\ \boldsymbol{y}(n) &= \boldsymbol{C}\,\boldsymbol{x}(n) + \boldsymbol{D}\,\boldsymbol{v}(n). \end{aligned}$$

There the matrices `A`, `B`, `C` and `D` represent the state-space matrices of the system. The matrix `U` describes the stimulation of the system and has got as many columns as existing inputs, in other words: The i-th column of `U` describes the stimulation at the i-th input of the system. The number of rows Q of `U` is determined by the length of the input signals. The matrix `Y` contains the simulation result, where the number of columns matches the number of system outputs R, and the number of rows depends on the length Q of the input signals. The value of the state vector during the simulation is stored row-wise in `X` for each step n.

See also `lsim_new`, `impulse_new`.

eig Calculation of eigenvalue and eigenvector

Call: `[V,D] = eig(A)`

Parameters:

- `A` : $N \times N$ matrix, whose eigenvalues and eigenvectors will be calculated
- `D` : Diagonal matrix, containing the i-th eigenvalue λ_i of `A` as the i-th element on the main diagonal
- `V` : Matrix of eigenvectors, containing in the i-th *column* the normalized eigenvector x_i which belongs to the i-th eigenvalue λ_i.

`eig` calculates the solutions of the eigenvalue problem

$$\boldsymbol{A}\boldsymbol{x}_i = \lambda_i \boldsymbol{x}_i, \quad i = 0, \ldots, N-1. \tag{38}$$

impulse_new Simulation of the impulse response of continuous LTI-systems

Call: `[Y,X] = impulse_new(A,B,C,D,iu,t)`

Parameters:

- `A` : $N \times N$ System matrix
- `B` : $N \times L$ Input matrix
- `C` : $R \times N$ Observation matrix
- `D` : $R \times L$ Pass through matrix
- `iu` : Index of the system input, which is stimulated with the impulse
- `t` : Equidistant timeaxis for the simulation
- `Y` : Impulse responses, $Q \times R \cdot L$ matrix
- `X` : Time course of the states, $Q \times N$ matrix (Q: Length of the impulse responses)

`impulse_new` simulates the impulse response of the continuous LTI-system with order N (L inputs, R outputs) in the state description

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\,\boldsymbol{x}(t) + \boldsymbol{B}\,\boldsymbol{v}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{C}\,\boldsymbol{x}(t) + \boldsymbol{D}\,\boldsymbol{v}(t). \end{aligned}$$

The internal transformation of the continuous system to a discrete system, which can be used for the simulation, is done by using the impulse invariant transformation. The matrices `A`, `B`, `C`, `D`, `U`, `Y` and `X` have the same meaning as in the function `dlsim_new`. `Y` contains the impulse responses of the system between all L inputs and R outputs and has got the dimension $Q \times R \cdot L$, where Q is the length of the single impulse responses in samples. The parameter `iu` is optional; is it specified, the simulation of the impulse response is only simulated between the `iu`-th input and all R outputs. Hence, the dimension of the impulse response matrix `y` is reduced to $Q \times R$. `t` is a time axis for the simulation, which has to be given by the user. If the time axis is specified, `iu` must be used, too.

See also `dlsim_new`, `lsim_new`.

lsim_new Simulation of continuous LTI-systems

Call: `[Y,X] = lsim_new(A,B,C,D,U,t)`

Parameter:

- `A` : $N \times N$ System matrix
- `B` : $N \times L$ Input matrix
- `C` : $R \times N$ Observation matrix
- `D` : $R \times L$ Pass through matrix
- `U` : System inputs, $Q \times L$ matrix (Q: Length of input signals)
- `t` : Equidistant time axis for the simulation
- `Y` : System reaction, $Q \times R$ matrix
- `X` : Time course of the states, $Q \times N$ matrix

`lsim_new` simulates the reaction of the continuous LTI-system with order N (L inputs, R outputs) in the state-space description

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\,\boldsymbol{x}(t) + \boldsymbol{B}\,\boldsymbol{v}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{C}\,\boldsymbol{x}(t) + \boldsymbol{D}\,\boldsymbol{v}(t). \end{aligned}$$

There the internal transformation of the continuous system to a discrete system, usable for simulation, is done by the impulse invariant transformation. The matrices `A`, `B`, `C`, `D`, `U`, `Y` and `X` have the same meaning as in the function `dlsim_new`.

See also `dlsim_new`, `impulse_new`.

sfg Graphical illustration of signal-flow graphs for systems of order two

Call: `sfg(A,B,C,D)`

Parameter:

- `A` : System matrix
- `B` : Input matrix
- `C` : Observation matrix
- `D` : Pass through matrix

The system described by the matrices `A`, `B`, `C` and `D` must have two inputs and two outputs. Moreover the order mustn't be greater then two!

ss2tf Calculation of the system transfer function from the state-space description

Call: `[ZAE,nen] = ss2tf(A,B,C,D,iu)`

Parameter:

- `A` : $N \times N$ System matrix
- `B` : $N \times L$ Input matrix
- `C` : $R \times N$ Observation matrix
- `D` : $R \times L$ Pass through matrix
- `iu` : Index l (see below) of the system input, from which the transfer function to all outputs should be calculated
- `ZAE` : Numerator of the s-transfer function, $R \times (N+1)$ matrix
- `nen` : Denominator of the s-transfer function, row vector of length $N+1$

`ss2tf` determines the transfer function of a continuous system from the state-space description, according to

$$\boldsymbol{h}^{(l)}(s) = \boldsymbol{C}\Big[s\boldsymbol{E} - \boldsymbol{A}\Big]^{-1}\mathbf{b}^{(l)} + \boldsymbol{d}^{(l)}, \tag{39}$$

where $\boldsymbol{E}$ represent the identity matrix and $\boldsymbol{b}^{(l)}$ resp. $\boldsymbol{d}^{(l)}$ are the l-th *column* of the matrices $\boldsymbol{B}$ resp. $\boldsymbol{D}$. The vector

$$\boldsymbol{h}^{(l)}(s) = \Big[H_{0,l}(s)\, H_{1,l}(s) \ldots H_{R-1,l}(s)\Big]^T \tag{40}$$

contains all transfer functions of the l-th input to all R outputs of the system. Each part of the transfer function $H_{r,l}(s)$ can be decomposed to a fractional, rational function with a common denominator polynomial (look above), whose coefficients are specified in the variable `nen`. The numerator polynomial belonging to $H_{r,l}(s)$ stands in the r-th *row* of the numerator polynomial matrix `ZAE`.