Advanced Signals and Systems – State-Space Description and System Realizations

Gerhard Schmidt

Christian-Albrechts-Universität zu Kiel
Faculty of Engineering
Institute of Electrical and Information Engineering
Digital Signal Processing and System Theory
Contents of the Lecture

Entire Semester:

- Introduction
- Discrete signals and random processes
- Spectra
- Discrete systems
- Idealized linear, shift-invariant systems
- Hilbert transform
- State-space description and system realizations
- Generalizations for signals, systems, and spectra
Contents of this Part

State-Space Description and System Realizations

- Introduction
- Basic structure
- Application example
- From difference equation to state-space representations
- Signal-flow graphs
- Signal-flow graph representation of basic structures
- Transfer matrix, impulse-response matrix, and transition matrix
- Equivalent Realizations
Introduction

**Restriction**

The ideas in this part of the lecture are restricted to linear, shift-invariant, dynamic, and causal systems.

**Up to now ...**

... we mainly dealt with systems of which we know the internal parameters:

![Diagram](image)

Linear, time-invariant system

The „inner part“ of the system was, e.g., described by its impulse response or by its Fourier transform.
**State-space description**

Basis idea:

The "**state of a system**“ is changing in dependence of the **current state vector** \( x(n) \) and of the **excitation** (the **input**) of the system:

\[
v_l(n) \quad \text{with} \quad l \in \{0, 1, ..., L - 1\}.
\]

All input signals will be grouped in a so-called **input signal vector**

\[
v(n) = [v_0(n), v_1(n), ..., v_{L-1}(n)]^T.
\]

All output signals \( y_r(n) \) with \( r \in \{0, 1, ..., R - 1\} \) are grouped in a so-called **output signal vector**

\[
y(n) = [y_0(n), y_1(n), ..., y_{R-1}(n)]^T.
\]
State-space description (continued)

Basis idea (continued):

In the same manner as for the input and output signals we will group all state-space variables $x_i(n)$ in a so-called state-space vector:

$$\mathbf{x}(n) = [x_0(n), x_1(n), \ldots, x_{N-1}(n)]^T.$$

The individual states can be regarded as memory cells (for the entire past). They are responsible for the behavior of the system in case of no input. Thus, the states describe the self or eigen behavior of the system.
**State-space description (continued)**

Using the three vectors the system is described by a **set of two equations**:

\[
x(n + 1) = f(x(n), v(n)),
\]

\[
y(n) = g(x(n), v(n)).
\]

In general, these equations can have arbitrary character. However, we will restrict ourselves here – as mentioned a few slides before – to **linear, shift-invariant, dynamic, and causal systems**. As a consequence the functions \(f(\ldots)\) and \(g(\ldots)\) have to be **linear with respect to** \(x(n)\) **and** \(v(n)\). In addition, the parameter of the functions should not depend on the time index \(n\) (due to shift invariance).
State-Space Description and System Realizations

Basic Structure – Part 4

**State-space description (continued)**

With the restrictions introduced before we can make the following *ansatz for describing linear, shift-invariant systems in the state-space domain*:

\[
\begin{align*}
    x(n+1) &= A x(n) + B v(n), \\
y(n) &= C x(n) + D v(n).
\end{align*}
\]

The quantities \(A\), \(B\), \(C\), and \(D\) have to be *matrices*, that describe linear relations with the variables \(x_r(n)\) and \(v_l(n)\). For the dimensions of the matrices we get:

- \(A : [N \times N]\),
- \(B : [N \times L]\),
- \(C : [R \times N]\),
- \(D : [R \times L]\).

**Example of a \([I \times J]\) matrix (I rows, J columns):**

\[
M = \begin{bmatrix}
m_{0,0} & m_{0,1} & \cdots & m_{0,j} & \cdots & m_{0,J-1} \\
m_{1,0} & m_{1,1} & \cdots & m_{1,j} & \cdots & m_{1,J-1} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
m_{I-1,0} & m_{I-1,1} & \cdots & m_{I-1,j} & \cdots & m_{I-1,J-1}
\end{bmatrix}
\]

\(M : [I \times J]\).
State-space description (continued)

Overview:

\[ v(n) \]

\[ \times \]

\[ x(n+1) \]

\[ \times \]

\[ x(n) \]

\[ \times \]

\[ v(n) \]

\[ D \]

\[ \times \]

\[ A \]

\[ B \]

\[ C \]

\[ y(n) \]

\[ N \text{ memory cells} \]
State-Space Description and System Realizations

Basic Structure – Part 6

State-space description (continued)

Names of the individual equations:

\[
\begin{align*}
x(n + 1) &= A x(n) + B v(n), \\
y(n) &= C x(n) + D v(n).
\end{align*}
\]

Meaning of the individual matrices:

- **A**: All feedback paths are described (system behaviour without input) in this matrix. The matrix is called system matrix.
- **B**: Connection of the system states with the input (steering of the systems). The matrix is called steering matrix.
- **C**: Coupling of the system states with the output signals. The matrix is called observation matrix.
- **D**: Direct connection of the input with the output. The matrix is called pass through matrix.

Both equations together are called the state-space description of a system!
State-space description (continued)

Discrete (and digital) systems do not have an “energy-based” memory as we find it often in continuous systems (e.g. the voltage on a capacitor). However, we often find a memory for (digital) data that can be written to in one sample and read from in the next:

At index $n$ we have $x_i(n)$ at the memory output – the input is connected to the sample that will be available at the output at index $n + 1$.

This describes a delay or a shift of one sample. In the $z$ domain we can describe this in terms of its transfer function as

$$H(z) = \frac{1}{z} = z^{-1}.$$
**State-space description (continued)**

Extended overview:

\[
\begin{align*}
\mathbf{v}(z) &= [V_0(z), V_1(z), \ldots, V_{L-1}(z)]^T, \\
\mathbf{x}(z) &= [X_0(z), X_1(z), \ldots, X_{N-1}(z)]^T, \\
\mathbf{y}(z) &= [Y_0(z), Y_1(z), \ldots, Y_{R-1}(z)]^T. \\
\mathbf{x}(z) &= A z^{-1} \mathbf{x}(z) + B z^{-1} \mathbf{v}(z), \\
\mathbf{y}(z) &= C \mathbf{x}(z) + D \mathbf{v}(z).
\end{align*}
\]
Application Example – Part 1

**Kalman filter**

The following examples are taken from the dissertation of Dr.-Ing. Henning Puder, Technische Universität Darmstadt. He has implemented a noise suppression system for hands-free communication in cars.

**Application overview**

Speech s(n)

Background noise b(n)
Kalman filter (continued)

In the state-space approach of H. Puder autoregressive models were used for the speech and the noise components.

\[ v_b(n) \rightarrow b_b \rightarrow z^{-1} \rightarrow y_b(n) \]
\[ v_s(n) \rightarrow b_s \rightarrow z^{-1} \rightarrow y_s(n) \]

Linear state-space model for background noise (here time-variant model parameters were used)

Linear state-space model for speech signals (time-variant model parameters were used)
Kalman filter (continued)

Based on a state space description of a system an algorithm can be formulated that separates a desired signal in an optimal manner from an additive distortion. Such a filter is called – according to its inventor – a Kalman filter. It has the following properties:

- the filter is *linear*,
- it generates an *unbiased* estimation,
- the filter output has *minimum error variance*, and
- the filter can be computed *recursively*.

Due to this properties such a filter is used quite often. Especially in control theory we find very many applications.
Kalman filter (continued)

**Rudolf Emil Kálmán** (born May 19, 1930) is a Hungarian-American electrical engineer, mathematical system theorist, and college professor, who was educated in the United States, and has done most of his work there. He is currently a retired professor from three different institutes of technology and universities. He is most noted for his co-invention and development of the Kalman filter, a mathematical formulation that is widely used in control systems, avionics, and outer space manned and unmanned vehicles.

Kálmán worked as a Research Mathematician at the Research Institute for Advanced Studies in Baltimore, Maryland from 1958 until 1964. He was a professor at Stanford University from 1964 until 1971, and then a Graduate Research Professor and the Director of the Center for Mathematical System Theory, at the University of Florida from 1971 until 1992. Starting in 1973, he also held the chair of Mathematical System Theory at the Swiss Federal Institute of Technology in Zürich, Switzerland.

*Source: Wikipedia*
**Kalman filter (continued)**

Structure of a Kalman filter for noise suppression:

- State-space model for car noise
- State-space model for speech signals
- Kalman gain computation
Application Example – Part 6

**Kalman filter (continued)**

Example 1: Stationary car noise

<table>
<thead>
<tr>
<th>Female speech</th>
<th>Male speech</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noisy speech signal</td>
<td>Noisy speech signal</td>
</tr>
<tr>
<td>„Conventional“ (Wiener) filter</td>
<td>„Conventional“ (Wiener) filter</td>
</tr>
<tr>
<td>Kalman filter output</td>
<td>Kalman filter output</td>
</tr>
</tbody>
</table>

Example 1: Non-stationary car noise (acceleration)

<table>
<thead>
<tr>
<th>Audio examples with permission of H. Puder, TU Darmstadt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noisy speech signal</td>
</tr>
<tr>
<td>After suppression of the engine harmonics</td>
</tr>
<tr>
<td>„Conventional“ (Wiener) filter</td>
</tr>
<tr>
<td>Kalman filter</td>
</tr>
</tbody>
</table>
**Difference equations**

A discrete, linear, shift-invariant system with one input and one output can be described by the following difference equation:

\[ y(n) + a_1 y(n - 1) + a_2 y(n - 2) + \ldots + a_{N_1} y(n - N + 1) = b_0 v(n) + \tilde{b}_1 v(n - 1) + \tilde{b}_2 v(n - 2) + \ldots + \tilde{b}_{N-1} v(n - N + 1). \]

**State-space description**

In order to get the corresponding state-space description we need ...  
- ... an equation about how the state vector is changed over time (the *system* or *state equation*). We should get a dependence on the input and on the old state-vector.  
- ... an equation that determines the system output in dependence of the input vector and the state vector (the *measurement equation*).  


Transformation into a state-space description

As a first step we will split the system into a part that describes the direct relation of the input and the output and a remaining system:

\[ y(n) + a_1 y(n-1) + a_2 y(n-2) + \ldots + a_{N-1} y(n-N+1) = b_0 v(n) + \tilde{b}_1 v(n-1) + \tilde{b}_2 v(n-2) + \ldots + \tilde{b}_{N-1} v(n-N+1) \]

\[ \implies y(n) = d v(n) + u(n) \]

with

\[ d = b_0, \]

and

\[ u(n) + a_1 u(n-1) + a_2 u(n-2) + \ldots + a_{N-1} u(n-N+1) = b_1 v(n-1) + b_2 v(n-2) + \ldots + b_{N-1} v(n-N+1). \]

Renaming according to our pass through notation.

New difference equation with a new output and without the term \( b_0 v(n) \)!
Transformation into a state-space description (continued)

Graphical explanation

Original system

Linear, discrete, shift-invariant system without pass through part, described by the difference equation

\[ u(n) + a_1 u(n - 1) + a_2 u(n - 2) + \ldots + a_{N-1} u(n - N + 1) = b_1 v(n - 1) + b_2 v(n - 2) + \ldots + b_{N-1} v(n - N + 1). \]
Transformation into a state-space description (continued)

The original system can be described by the following transfer function

\[ H(z) = \frac{Y(z)}{V(z)} = \frac{b_0 + \tilde{b}_1 z^{-1} + \tilde{b}_2 z^{-2} + \ldots}{1 + a_1 z^{-1} + a_2 z^{-2} + \ldots}. \]

We can extend the numerator in the following way

\[ H(z) = \frac{b_0 + \tilde{b}_1 z^{-1} + \tilde{b}_2 z^{-2} + \ldots + b_0 a_1 z^{-1} + b_0 a_2 z^{-2} + \ldots - b_0 a_1 z^{-1} - b_0 a_2 z^{-2} - \ldots}{1 + a_1 z^{-1} + a_2 z^{-2} + \ldots}. \]

This allows for splitting the transfer function into a pass through part and a remaining structure

\[ H(z) = \frac{b_0 + \tilde{b}_1 z^{-1} + \tilde{b}_2 z^{-2} + \ldots + b_0 a_1 z^{-1} + b_0 a_2 z^{-2} + \ldots - b_0 a_1 z^{-1} - b_0 a_2 z^{-2} - \ldots}{1 + a_1 z^{-1} + a_2 z^{-2} + \ldots} \]

\[ = \frac{b_0 (1 + a_1 z^{-1} + a_2 z^{-2} + \ldots) + (\tilde{b}_1 - b_0 a_1) z^{-1} + (\tilde{b}_2 - b_0 a_2) z^{-2} + \ldots}{1 + a_1 z^{-1} + a_2 z^{-2} + \ldots} \]

\[ = b_0 + \frac{(\tilde{b}_1 - b_0 a_1) z^{-1} + (\tilde{b}_2 - b_0 a_2) z^{-2} + \ldots}{1 + a_1 z^{-1} + a_2 z^{-2} + \ldots} \quad \Rightarrow \quad b_i = \tilde{b}_i - b_0 a_i.\]
One possible derivation (continued)

Desired structure for one input and one output signal:

Current result:

\[ v(n) \rightarrow b \rightarrow x(n + 1) \rightarrow z^{-1} \rightarrow x(n) \rightarrow c \rightarrow y(n) \]

Pass through scalar

Steering vector

Observation vector

System matrix
One possible derivation (continued)

The new difference equation is changed from so-called first direct form into second direct form. The memory elements of the second direct form contain the states of the system.
One possible derivation (continued)

We start with the difference equation of the system without direct input-output connection

\[ u(n) + a_1 u(n - 1) + a_2 u(n - 2) + \ldots + a_{N-1} u(n - N + 1) = b_1 v(n - 1) + b_2 v(n - 2) + \ldots + b_{N-1} v(n - N + 1). \]

and modify it such that we obtain two equations in the so-called second direct form:

\[ u(n) = b_1 x(n) + b_2 x(n - 1) + \ldots + b_{N-1} x(n - N + 2), \]
\[ x(n + 1) = v(n) - a_1 x(n) - a_2 x(n - 1) - \ldots - a_{N-1} x(n - N + 2). \]

In order to fulfill the equation \( y(n) = d u(n) + x(n) \), we define

\[ u(n) = c^T x(n), \]

with the observation vector (in general a matrix)

\[ c = \begin{bmatrix} b_1, \ldots, b_{N-1}, 0 \end{bmatrix}^T \]

and the state vector (in general also a matrix)

\[ x(n) = \begin{bmatrix} x(n), x(n - 1), \ldots, x(n - N + 1) \end{bmatrix}^T. \]
One possible derivation (continued)

**Desired structure** for one input and one output signal:

Current result:
From Difference Equations to State-Space Descriptions – Part 9

One possible derivation (continued)

Starting with the difference equation of a system without direct connection (pass through)

\[ x(n + 1) = v(n) - a_1 x(n) - a_2 x(n - 1) - \ldots - a_{N-1} x(n - N + 2) \]

and the definition of the state vector

\[ x(n) = [x(n), x(n - 1), \ldots, x(n - N + 1)]^T, \]

we can go over to a matrix-vector description:

\[
\begin{bmatrix}
  x(n + 1) \\
  x(n) \\
  \vdots \\
  x(n - N + 4) \\
  x(n - N + 3) \\
  x(n - N + 2) \\
\end{bmatrix}
= 
\begin{bmatrix}
  -a_1 & -a_2 & \ldots & -a_{N-2} & -a_{N-1} & 0 \\
  1 & 0 & \ldots & 0 & 0 & 0 \\
  \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
  0 & 0 & \ldots & 0 & 0 & 0 \\
  0 & 0 & \ldots & 1 & 0 & 0 \\
  0 & 0 & \ldots & 0 & 1 & 0 \\
\end{bmatrix}
\begin{bmatrix}
  x(n) \\
  x(n - 1) \\
  \vdots \\
  x(n - N + 3) \\
  x(n - N + 2) \\
  x(n - N + 1) \\
\end{bmatrix}
+ 
\begin{bmatrix}
  1 \\
  0 \\
  \vdots \\
  0 \\
  0 \\
  0 \\
\end{bmatrix}
v(n)
\]

\[ x(n + 1) = A x(n) + b v(n). \]
One possible derivation (continued)

**Desired structure** for one input and one output signal:

Current result:
One possible derivation (continued)

Remarks:

- The derivation shown in the slides before is just **one of several possibilities to transform a difference equation into a state-space description**. Please be aware, that there are **several other ways** that lead to different state-space descriptions.

- The individual steps of transforming the difference equation were restricted to systems with just one input and one output (for reasons of simplicity). Of course, the same transformation can be applied to **MIMI systems** without much more effort.

- Beside the discrete derivation it is also possible to transform **differential equations** that describe a continuous system into a **(continuous) state-space description**.
**Basics**

A signal-flow graph can help for simplified visualization of block-based system graphs. Signal-flow graphs are *directed and weighted graphs*, which means that all directions and all weights of the individual branches have to be specified.

Example:
Signal-Flow Graphs – Part 2

**Basic elements**

- **Memory / delay:**
  
  \[
  x(n+1) \quad \xrightarrow{\text{Delay}} \quad x(n) \\
  \]

- **Multiplication** with a constant factor (weight):
  
  \[
  x(\ldots) \quad \xrightarrow{a} \quad a \cdot x(\ldots) \\
  \]

- **Addition:**
  
  \[
  1 \quad \xrightarrow{a} \quad a \\
  x_1(\ldots) \quad \xrightarrow{b} \quad b \\
  x_2(\ldots) \quad \xrightarrow{c} \quad c \\
  \]

  \[
  a + b x_1(\ldots) + c x_2(\ldots) \]

  **Summation nodes are plotted as open circles!**

  **Branches without letter or number are weighted with 1!**
Basic elements (continued)

- **Cascade**: 
  \[ x(...) \xrightarrow{a} b \xrightarrow{a \cdot b} x(...) \equiv x(...) \xrightarrow{a \cdot b} x(...) \]

- **Parallel arrangement**: 
  \[ x(...) \xrightarrow{a} (a + b) x(...) \equiv x(...) \xrightarrow{(a + b)} (a + b) x(...) \]

  - **Summation node (open)**
  - **Splitting node (closed)**

- **Coupling back (feedback)**: 
  \[ x(...) \xrightarrow{b} a_1 \xrightarrow{a_2} c \xrightarrow{h} h x(...) \equiv x(...) \xrightarrow{h} h x(...) \]

  - **Rule for feedback (control theory)**
    \[ h = b \frac{a_1}{1 - a_1 a_2} c \]
Signal flow graph for the state-space description

We obtain for the state-space description in the \( z \) domain:

\[
\begin{align*}
\mathbf{v}(n) &= \left[ v_0(n), v_1(n), \ldots, v_{L-1}(n) \right]^T, \\
\mathbf{x}(n) &= \left[ x_0(n), x_1(n), \ldots, x_{N-1}(n) \right]^T, \\
\mathbf{y}(n) &= \left[ y_0(n), y_1(n), \ldots, y_{R-1}(n) \right]^T, \\
\end{align*}
\]

\[
\begin{align*}
\mathbf{x}(n+1) &= \mathbf{A} \mathbf{x}(n) + \mathbf{B} \mathbf{v}(n), \\
\mathbf{y}(n) &= \mathbf{C} \mathbf{x}(n) + \mathbf{D} \mathbf{v}(n).
\end{align*}
\]
Example

Assume that we have a system with the following properties:

- a discrete system,
- with $N = 3$ states, $x_0(n)$, $x_1(n)$, and $x_2(n)$,
- with $L = 2$ inputs, $v_0(n)$ and $v_1(n)$, and
- with $R = 1$ output $y(n)$.

Furthermore, we assume the following system equation (with example values):

$$x(n + 1) = \begin{bmatrix} x_0(n + 1) \\ x_1(n + 1) \\ x_2(n + 1) \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \begin{bmatrix} x_0(n) \\ x_1(n) \\ x_2(n) \end{bmatrix} + \begin{bmatrix} 10 & 11 \\ 12 & 13 \\ 14 & 15 \end{bmatrix} \begin{bmatrix} v_0(n) \\ v_1(n) \end{bmatrix}.$$

For the measurement equation, we assume the following (again with example values):

$$y(n) = \begin{bmatrix} 16 & 17 & 18 \end{bmatrix} \begin{bmatrix} x_0(n) \\ x_1(n) \\ x_2(n) \end{bmatrix} + \begin{bmatrix} 19 & 20 \end{bmatrix} \begin{bmatrix} v_0(n) \\ v_1(n) \end{bmatrix}.$$
**Example (continued)**

If we want to draw a signal-flow graph according to the two equations from the last slide, the following procedure is suggested:

- Draw the memory/delay elements $z^{-1}$ with inputs and outputs,
- add the states $x_i(...)$ to the graph (either in the time or in the z domain),
- add the input and output nodes ($v_i(...)$ and $y_i(...)$),
- draw all connections that result from $A$ (including directions and weights),
- draw all connections that result from $B$ (including directions and weights),
- draw all connections that result from $c$ (including directions and weights),
- draw all connections that result from $d$ (including directions and weights).
**Example (continued)**

Please complete the signal-flow graph below! Please use the example values for the weights that were given two slides before!

\[
\begin{align*}
v_0(n) & \quad x_0(n+1) z^{-1} \quad x_0(n) \\
v_1(n) & \quad x_1(n+1) z^{-1} \quad x_1(n) \\
& \quad x_2(n+1) z^{-1} \quad x_2(n) \\
& \quad y(n)
\end{align*}
\]
Definitions

Up to know we put our focus on systems with just one input and one output:

\[ v(n) \longrightarrow y(n). \]

If we extend the system to have \( L > 1 \) inputs and \( R > 1 \) outputs, we will use an excitation vector \( \mathbf{v}(n) \) and a reaction or output vector \( \mathbf{y}(n) \).

The SISO convolutions

\[ y(n) = h_0(n) * v(n) \]

can be extended for the MIMO case in a **matrix-vector manner**:

\[ \mathbf{y}(n) = \mathbf{H}_0(n) * \mathbf{v}(n). \]

The so-called **impulse-response matrix** \( \mathbf{H}_0(n) \) has the dimension \( (R \times L) \).
**Definitions (continued)**

The individual *elements* of the *signal vectors* and of the *impulse-response matrix* are defined as followed:

\[
\begin{bmatrix}
  y_0(n) \\
  y_1(n) \\
  \vdots \\
  y_r(n) \\
  \vdots \\
  y_{R-1}(n)
\end{bmatrix} =
\begin{bmatrix}
  h_{0,0,0}(n) & h_{0,0,1}(n) & \cdots & h_{0,0,l}(n) & \cdots & h_{0,0,L-1}(n) \\
  h_{0,1,0}(n) & h_{0,1,1}(n) & \cdots & h_{0,1,l}(n) & \cdots & h_{0,1,L-1}(n) \\
  \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
  h_{0,r,0}(n) & h_{0,r,1}(n) & \cdots & h_{0,r,l}(n) & \cdots & h_{0,r,L-1}(n) \\
  \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
  h_{0,R-1,0}(n) & h_{0,R-1,1}(n) & \cdots & h_{0,R-1,l}(n) & \cdots & h_{0,R-1,L-1}(n)
\end{bmatrix}
\begin{bmatrix}
  v_0(n) \\
  v_1(n) \\
  \vdots \\
  v_l(n) \\
  \vdots \\
  v_{L-1}(n)
\end{bmatrix}
\]

*One row of the equation system* can be interpreted as a *sum over \( L \) single convolutions*:

\[
y_r(n) = \sum_{l=0}^{L-1} h_{0,r,l}(n) * v_l(n).
\]
Definitions (continued)

Similar matrix-vector based notations can be used in the \( z \) domain. We know from previous investigations that linear, shift-invariant systems have the following property:

For an input \( v(n) = V z^n \) we get at the system output
\[
y(n) = Y z^n = H(z) V z^n.
\]

If we use that property for systems with a multitude of inputs and outputs (again the MIMI system is assumed to be linear and shift invariant) we obtain:

For an input vector \( \mathbf{v}(n) = \mathbf{v} z^n \) we get at the system output
\[
\mathbf{y}(n) = \mathbf{y} z^n = \mathbf{H}(z) \mathbf{v} z^n.
\]
**Definitions (continued)**

We can conclude that also for MIMO systems the following property holds: If a linear, shift-invariant system is excited with complex exponentials of the form $z^n$ weighted with individual (complex) amplitudes, also all output sequences show the same form.

With a similar derivation we can show the same for general harmonic exponentials. We get:

For $v(n) = v e^{j\Omega n}$ we obtain at the system output

$$y(n) = y e^{j\Omega n} = H(e^{j\Omega}) v e^{j\Omega n}.$$
Definitions (continued)

The matrices introduced before (impulse response matrix $H_0(n)$, transfer matrix $H(z)$, and frequency response matrix $H(e^{j\Omega})$) are extensions of the known scalar quantities. All relations that allow to transform the individual scalar quantities are still valid and we obtain for the MIMO case:

$$H(z) = \mathcal{Z}\{H_0(n)\},$$
$$H(e^{j\Omega}) = \mathcal{F}\{H_0(n)\}.$$ 

Now the transforms are assumed to be applied individually to the single matrix elements, e.g.

$$H_{r,l}(z) = \mathcal{Z}\{h_{0,r,l}(n)\}.$$ 

Transfer function / impulse response from input $l$ the output $r$.
Transfer matrix

Next, we will discuss the connection of the state-space description in terms of the parameters (the matrices $A, B, C,$ and $D$) and the transfer function matrix of MIMO systems. We start with the following relation:

If all input sequences of a linear, shift invariant system are of (complex) exponential type $v(n) = v z^n$ then all system states and all output sequences are also of (complex) exponential type:

$$x(n) = x z^n,$$
$$y(n) = y z^n.$$

If we compute the delayed version of the state vector, we obtain:

$$x(n) = x z^n,$$
$$x(n + 1) = x z^{n+1} = x z z^n.$$
Transfer matrix (continued)

By inserting the result into the system equation we get:

\[ x(n+1) = A \, x(n) + B \, v(n) \]

... inserting harmonic exponentials as input and state sequences ...

\[ x \, z \, z^n = A \, x \, z^n + B \, v \, z^n \]

... adding a unity matrix ...

\[ z \, I \, x \, z^n = A \, x \, z^n + B \, v \, z^n \]

... truncation of \( z^n \) ...

\[ z \, I \, x = A \, x + B \, v \]

... bringing all terms with \( x \) on one side ...

\[ [z \, I - A] \, x = B \, v \]

... solving for \( x \) ...

\[ x = [z \, I - A]^{-1} \, B \, v. \]
Transfer matrix (continued)

If we do the same modification for the measurement equations we obtain:

\[
\begin{align*}
y(n) &= C \, x(n) + D \, v(n) \\
\end{align*}
\]

... inserting harmonic exponentials as input and state sequences ...

\[
\begin{align*}
y \, z^n &= C \, x \, z^n + D \, v \, z^n \\
\end{align*}
\]

... truncation of \( z^n \) ...

\[
\begin{align*}
y &= C \, x + D \, v \\
\end{align*}
\]

... inserting the result obtained for \( x \) (see last slide) ...

\[
\begin{align*}
y &= C \, [z \, I - A]^{-1} \, B \, v + D \, v \\
\end{align*}
\]

... excluding \( v \) ...

\[
\begin{align*}
y &= \left( C \, [z \, I - A]^{-1} \, B + D \right) \, v.
\end{align*}
\]

By comparing this result with the earlier found relation \( y = H(z) \, v \), we get the relation of the state-space parameters and the transfer matrix:

\[
H(z) = C \, [z \, I - A]^{-1} \, B + D.
\]

For the frequency response matrix we obtain the same result but with \( z = e^{j \Omega} \).
Impulse-response matrix

If we are interested to get an impulse response matrix out of the state-space matrices, we can use that the individual impulse responses can be obtained by inverse z transform of the corresponding entries of the transfer matrix. Thus, we get

\[ H_0(n) = Z^{-1}\{H(z)\} \]
\[ = Z^{-1}\{C [z I - A]^{-1} B + D\} \]
\[ = C Z^{-1}\{[z I - A]^{-1}\} B + D Z^{-1}\{1\} \]
\[ = C Z^{-1}\{[z I - A]^{-1}\} B + D \gamma_0(n). \]

**Remark:** Keep in mind that for the inverse z transform only those parts that are dependent on \(z\) are of special „interest“.

For further simplification and understanding of the impulse response matrix \(H_0(n)\) we will have a closer look on the inverse z transform of \([z I - A]^{-1}\) on the next slide.
Transfer Matrix, Impulse-Response Matrix, and Transition Matrix – Part 10

**Impulse-response matrix (continued)**

For the inverse transform of the matrix \([z I - A]^{-1}\) we will first look at the already known scalar transform pair

\[
Z^{-1} \left\{ \frac{z}{z - a} \right\} = a^n \gamma_{-1}(n).
\]

Extending (multiplying) the z-domain part with \(z^{-1}\) results in the time-domain part as a delay by one sample. We get:

\[
Z^{-1} \left\{ \frac{1}{z - a} \right\} = a^{n-1} \gamma_{-1}(n - 1).
\]

Extending this result to the MIMO case leads to:

\[
Z^{-1} \{[z I - A]^{-1}\} = A^{n-1} \gamma_{-1}(n - 1).
\]

*The exponent is applied in an elementwise manner – meaning that the result is again a matrix!*
Transfer Matrix, Impulse-Response Matrix, and Transition Matrix – Part 11

**Impulse-response matrix (continued)**

Inserting this result in the corresponding impulse response matrix and using it in the measurement equation leads to

$$H_0(n) = C A^{n-1} B \gamma_1(n-1) + D \gamma_0(n).$$

As a result the measurement equation can be exchanged to

$$y(n) = C \sum_{\kappa=1}^{\infty} A^{\kappa-1} B v(n-\kappa) + D v(n).$$

$$= x(n)$$
Some Questions

Transfer matrix, impulse-response matrix, and transition matrix (continued)

Partner work – Please think about the following questions and try to find answers (first group discussions, afterwards broad discussion in the whole group).

- What does it mean if a system has no pass-through part? What does this mean for the state-space description and what can you say about the impulse response matrix of such systems?

- What quantities of the excitation signal vectors have to be the same and which can be chosen individually in order to do all the modifications of the last slides (resulting in the frequency response matrix and in the transfer matrix)?
**Transition matrix**

In order to apply the results from our previous studies of linear, shift invariant systems – especially those which have *transfer functions that are polynomials in the numerator and in the denominator* – we will again look on the transfer matrix. In the previous slides we found the following result:

\[
H(z) = C \left[ zI - A \right]^{-1} B + D.
\]

The involved matrix \([zI - A]\) is of the following form:

\[
[zI - A] = \begin{bmatrix}
z - a_{0,0} & -a_{0,1} & -a_{0,2} & \ldots & -a_{0,N-1} \\
-a_{1,0} & z - a_{1,1} & -a_{1,2} & \ldots & -a_{1,N-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
-a_{N-1,0} & -a_{N-1,1} & -a_{N-1,2} & \ldots & z - a_{N-1,N-1}
\end{bmatrix}.
\]
Transition matrix (continued)

If we want to compute the inverse of this matrix we can use the following relation (always true, not only for the special type of matrix that we face here):

\[
[z \mathbf{I} - \mathbf{A}]^{-1} = \frac{1}{\det[z \mathbf{I} - \mathbf{A}]} \text{adj}[z \mathbf{I} - \mathbf{A}].
\]
Transition matrix (continued)

For a 3x3 matrix of the form

\[
M = \begin{bmatrix}
a & b & c \\
d & e & f \\
g & h & i \\
\end{bmatrix}
\]

the **determinant** is computed as followed:

\[
\det\{M\} = aei + bfg + cdi - ceg - bdi - afh.
\]

For the **adjoint** we get:

\[
\text{adj}\{M\} = \begin{bmatrix}
\det \begin{bmatrix} e & f \\
h & i \\
\end{bmatrix} & -\det \begin{bmatrix} d & f \\
g & i \\
\end{bmatrix} & \det \begin{bmatrix} d & e \\
g & h \\
\end{bmatrix} \\
-\det \begin{bmatrix} b & c \\
h & i \\
\end{bmatrix} & \det \begin{bmatrix} a & c \\
g & i \\
\end{bmatrix} & -\det \begin{bmatrix} a & b \\
g & h \\
\end{bmatrix} \\
\det \begin{bmatrix} b & c \\
e & f \\
\end{bmatrix} & -\det \begin{bmatrix} a & c \\
d & f \\
\end{bmatrix} & \det \begin{bmatrix} a & b \\
d & e \\
\end{bmatrix}
\end{bmatrix}^T
\]

\[
= \begin{bmatrix}
ei - hf & ch - bi & bf - ce \\
fg - di & ai - cg & cd - af \\
dh - eg & bg - ah & ae - bd
\end{bmatrix}
\]
Transition matrix (continued)

If we apply our knowledge about the determinant and the adjoint, we can see that we obtain polynomials of \( z \) for all elements the inverse matrix. Important here is the degree of the polynomial that we obtain:

\[
[z I - A]^{-1} = \frac{1}{\det\{zI - A\}} \text{adj}\{zI - A\}.
\]

Polynomial in the numerator of degree \( \leq N-1 \)

Polynomial in the denominator of degree \( N \)

The polynomial that appears in the denominator of all elements is called *characteristic polynomial*. It is defined via the determinant of the matrix \([z I - A]\)

\[
N(z) = \det\{zI - A\}.
\]
Transition matrix (continued)

We can finally conclude that all elements of the inverted matrix \( [z \mathbf{I} - \mathbf{A}]^{-1} \) are broken rational functions with identical denominator polynomials \( N(z) \). We obtain for the elements of the transfer matrix:

\[
H_{r,l}(z) = \frac{Z_{r,l}(z)}{N(z)} + d_{r,l}
\]

... modifying such that a common denominator is used (leading also to a numerator degree of \( N \)) ...

\[
= \frac{Z_{r,l}(z) + d_{r,l} N(z)}{N(z)}
\]

... combining all terms and renaming the coefficients ...

\[
\sum_{\mu=0}^{N} \alpha_{\mu}^{(r,l)} \cdot z^{\mu}
\]

\[
= \frac{\sum_{\nu=0}^{N} \beta_{\nu} \cdot z^{\nu}}{N}
\]
**Transition matrix (continued)**

This kind of description is known from earlier parts of this lecture. We can transform the description that is based on a sum into a *product form*. Here we obtain

\[
H_{r,l}(z) = \frac{\sum_{\mu=0}^{N} \alpha_{\mu}^{(r,l)} \cdot z^\mu}{\sum_{\nu=0}^{N} \beta_{\nu} \cdot z^\nu} = \frac{\prod_{\mu=0}^{N-1} z - z_{0,\mu}^{(r,l)}}{\prod_{\nu=0}^{N-1} z - z_{\infty,\nu}}.
\]

*Remark: The zeros in the numerator might be different for each matrix element. The poles (the zeros of the denominator) are equal for all matrix elements!*
Transition matrix (continued)

If all pole locations are different, we can apply a simple partial fraction expansion. We obtain in that case

\[ H_{r,l}(z) = d_{r,l} + \sum_{\nu=0}^{N-1} \frac{B^{(r,l)}_{\nu}}{z - z_{\infty,\nu}}. \]

If a pole appears more than once an extended partial fraction expansion is necessary and we obtain

\[ H_{r,l}(z) = d_{r,l} + \sum_{\nu=0}^{\tilde{N}-1} \sum_{\kappa=1}^{N_{\nu}} \frac{B^{(r,l)}_{\nu,\kappa}}{(z - z_{\infty,\nu})^{\kappa}}. \]
Transfer matrix, impulse-response matrix, and transition matrix

Partner work – Please think about the following questions and try to find answers (first group discussions, afterwards broad discussion in the whole group).

- Why do we have the same pole locations for all individual transfer functions?
  - ……………………………………………………………………………………………………………………………………
  - ……………………………………………………………………………………………………………………………………

- What can you conclude for the number of zeros (the degree difference between numerator and denominator) if a system has no pass-through part?
  - ……………………………………………………………………………………………………………………………………
  - ……………………………………………………………………………………………………………………………………

- How can you determine the impulse response if you know the partial fraction expansion of a transfer function as shown in the last slide?
  - ……………………………………………………………………………………………………………………………………
  - ……………………………………………………………………………………………………………………………………
Impulse response matrix (continued)

When investigating discrete systems in the z domain we found the following relations:

\[
\mathcal{Z}^{-1}\left\{ \frac{1}{z-a} \right\} = a^{n-1}\gamma_{-1}(n-1),
\]

\[
\mathcal{Z}^{-1}\left\{ \frac{1}{(z-a)^\kappa} \right\} = a^{n-\kappa}\binom{n-1}{\kappa-1}\gamma_{-1}(n-\kappa).
\]

If we use these transform pairs now, we obtain for the individual elements of the impulse response matrix ....

- ... in case of a single pole:

\[
h_{0,r,l}(n) = d_{r,l} \gamma_0(n) + \sum_{\nu=0}^{N-1} B_{\nu}^{(r,l)} z_{\infty,\nu}^{n-1} \gamma_{-1}(n-1).
\]

- ... in case of poles that appear more than once:

\[
h_{0,r,l}(n) = d_{r,l} \gamma_0(n) + \sum_{\nu=0}^{\tilde{N}-1} \sum_{\kappa=1}^{N_{\nu}} B_{\nu,\kappa}^{(r,l)} z_{\infty,\nu}^{n-\kappa} \binom{n-1}{\kappa-1} \gamma_{-1}(n-\kappa).
\]
**Impulse response matrix (continued)**

Remarks:

- All elements of the impulse response matrix $H_0(n)$ are linear combinations of exponential sequences $z_n^\infty$.

- If a pole is appearing more than once the sequences are weighted in addition with terms of the form $n^{\kappa - 1}$.

- The poles $z_{\infty,\nu}$ of $H(z)$ are determining the impulse responses. They are obtained by finding the zeros of the so-called characteristical polynomial

$$N(z_{\infty,\nu}) = \det\{z_{\infty,\nu}I - A\} = 0.$$  

These zeros are also the Eigen values of the system matrix $A$!

**Remember:**

$$Ax = \lambda x$$  
$$Ax = \lambda I x$$  
$$0 = (\lambda I - A)x. \quad \text{... can be solved if ...} \quad \det\{\lambda I - A\} = 0.$$
Impulse response matrix (continued)

In earlier parts of this lecture we treated rational transfer functions (z domain). The individual elements of the transfer matrix of the MIMO systems that result from state-space descriptions are of the same type. Thus, we have the same stability criterion:

A stable MIMO system should have all poles inside the unit circle.
Equivalent Realizations – Part 1

**Basics**

The state-space description can be used to change the signal processing structure. To explain this in more detail we assume that we have the following **two system realizations**:

- A first realization with
  \[ \mathbf{x}(n) = [x_1(n), x_2(n), \ldots, x_N(n)]^T, \quad A_x, B_x, C_x, D_x \Rightarrow H_x(z), h_{0,x,i,j}(n). \]

- A second realization with
  \[ \mathbf{q}(n) = [q_1(n), q_2(n), \ldots, q_N(n)]^T, \quad A_q, B_q, C_q, D_q \Rightarrow H_q(z), h_{0,q,i,j}(n). \]

We assume that both realizations have the **same input-output behavior**, this means that we require

\[
H_x(z) \equiv H_q(z), \\
h_{0,x,i,j}(n) \equiv h_{0,q,i,j}(n), \quad \forall i, j.
\]
Basics (continued)

In order to fulfill the assumption of the same input-output behavior we make the following ansatz:

\[ x(n) = T q(n). \]

This means that each state component \( x_i(n) \) of the first system is a linear combination of all state components \( q_j(n) \) of the second system:

\[
\begin{align*}
x_1(n) &= T_{1,1} q_1(n) + T_{1,2} q_2(n) + \ldots + T_{1,N} q_N(n), \\
x_2(n) &= T_{2,1} q_1(n) + T_{2,2} q_2(n) + \ldots + T_{2,N} q_N(n), \\
\vdots &= \vdots \\
x_N(n) &= T_{N,1} q_1(n) + T_{N,2} q_2(n) + \ldots + T_{N,N} q_N(n).
\end{align*}
\]

The idea behind that ansatz is to obtain a better (usually this means a uniform) amplitude range for all signal and state variables. This is important for the numerical behavior of digital systems.
Equivalent Realizations – Part 3

**Basics (continued)**

Since we require *equivalence* of both approaches we have to make sure that no system information gets lost. This means that the transformation of the system matrix should be *invertible*:

\[ x(n) = T q(n) \implies q(n) = T^{-1} x(n). \]

Thus, the transformation matrix \( T \) has to be *regular*:

\[ TT^{-1} = I. \]
Equivalent state-space descriptions

If we insert our ansatz $x(n) = T q(n)$ into the first state-space description we obtain:

$$
x(n+1) = A x(n) + B v(n),$$

... inserting the ansatz $x(n) = T q(n)$...

$$T q(n+1) = A T q(n) + B v(n),$$

... multiplying from the left with $T^{-1}$ ...

$$q(n+1) = \overbrace{T^{-1} A T q(n) + T^{-1} B v(n),}^{A_q B_q}$$

... inserting abbreviation ...

$$q(n+1) = A_q q(n) + B_q v(n).$$

In order to be as generic as possible, we have used vectors as inputs and outputs, meaning that we have used MIMO (multiple-input multiple-output) systems.
Equivalent state-space descriptions (continued)

In addition to the system equation we can also modify the measurement equation:

\[ y(n) = C x(n) + D v(n), \]

... inserting the ansatz \( x(n) = T q(n) \)... 

\[ y(n) = \begin{bmatrix} C T \\ C_q \end{bmatrix} q(n) + \begin{bmatrix} D \\ D_q \end{bmatrix} v(n), \]

... inserting abbreviations ... 

\[ y(n) = C_q q(n) + D_q v(n). \]

As a result we can summarize what we get for the matrices of the transformed system:

\[ A_q = T^{-1} A T, \]
\[ B_q = T^{-1} B, \]
\[ C_q = C T, \]
\[ D_q = D. \]

Due to the transformation all system-internal feedback paths and also the state variables are changed \( q_i(n) \neq x_i(n) \).

Thus, the state variable will have a different behavior!
Proof of equivalence

Due to our starting assumptions we have already ensured that the original and the transformed system have the same output $y(n)$ if they are excited with the same input $v(n)$. However, we can also formally show this equivalence.

For the transfer matrices of both system realizations we obtain:

- for the system with $x(n)$:
  $$H_x(z) = C [z I - A]^{-1} B + D.$$

- for the system with $q(n)$:
  $$H_q(z) = C_q [z I - A_q]^{-1} B_q + D_q.$$

By inserting the matrix definitions (see last slide) we obtain for the system with $q(n)$:

$$H_q(z) = \underbrace{CT}_{C_q} \underbrace{[z I - A_T]}_{A_q}^{-1} \underbrace{AT}_{B_q} + \underbrace{D}_{D_q}. $$
Proof of equivalence (continued)

If we use the following modifications

\[ H_q(z) = C \left[ z I - T^{-1} A T \right]^{-1} T^{-1} B + D \]

... inserting a "double inversion" \( [M^{-1}]^{-1} = M \) ...

\[ = C \left[ T \left[ z I - T^{-1} A T \right]^{-1} T^{-1} \right]^{-1} B + D \]

... exploiting that \( [ABC]^{-1} = C^{-1} B^{-1} A^{-1} \) ...

\[ = C \left[ T \left[ z I - T^{-1} A T \right] T^{-1} \right]^{-1} B + D \]

... multiplying the left and right matrices ...

\[ = C \left[ z T I T^{-1} - T T^{-1} A T T^{-1} \right]^{-1} B + D \]

... simplifying the subterms ...

\[ = C \left[ z I - A \right]^{-1} B + D = H_x(z). \]

we obtain the transfer matrix of the system with the state variables \( x(n) \).
Proof of equivalence (continued)

Remember:

From the previous slides we know that the poles of the system are also the zeros of the so-called characteristic polynomial:

\[ \det\{ zI - A \} = 0. \]

These zeros are also the Eigen values of the system matrix \( A \). This means that we have

\[ A e_i = z_{\infty,i} e_i. \]

In the equation above \( e_i \) are the Eigen vectors that correspond to the individual eigen values \( z_{\infty,i} \). If all eigen values are different we have in addition

\[ e_i^T e_j = \begin{cases} 1, & i = j, \\ 0, & \text{else}. \end{cases} \]

We will exploit this for deriving a special equivalence transformation in the next slides.
Equivalent Realizations – Part 9

**Transforming to a diagonal structure**

We will make the following ansatz for the transformation matrix:

\[ T = E = [e_1, e_2, ..., e_N], \]

meaning that the *transformation matrix consists of the Eigen vectors of the system matrix* \( A \). We assume here that *all poles are different*. In this case we obtain by matrix multiplication from the right:

\[
AT = A [e_1, e_2, ..., e_N]
\]

... exploiting that \( A e_i = z_{\infty,i} e_i \) ...

\[
= [z_{\infty,1} e_1, z_{\infty,2} e_2, ..., z_{\infty,N} e_N]
\]

... rearranging into a product of a transformation matrix and a diagonal matrix ...

\[
= [e_1, e_2, ..., e_N] \text{ diag} \left\{ [z_{\infty,1}, z_{\infty,2}, ..., z_{\infty,N}]^T \right\}
\]

... inserting the definition transformation matrix ...

\[
= E \text{ diag} \left\{ [z_{\infty,1}, z_{\infty,2}, ..., z_{\infty,N}]^T \right\}.
\]

**Diagonal matrix having the individual Eigen value on the main diagonal!**
Transforming to a diagonal structure (continued)

If we multiply (from the left) with the transformation matrix, we obtain:

\[ T^{-1} A T = T^{-1} E \text{ diag} \left\{ \begin{bmatrix} z_{\infty,1}, z_{\infty,2}, \ldots, z_{\infty,N} \end{bmatrix}^T \right\} \]

... inserting the ansatz ...

\[ = E^{-1} E \text{ diag} \left\{ \begin{bmatrix} z_{\infty,1}, z_{\infty,2}, \ldots, z_{\infty,N} \end{bmatrix}^T \right\} \]

... simplifying (matrix times its inverse = unity matrix) ...

\[ = \text{ diag} \left\{ \begin{bmatrix} z_{\infty,1}, z_{\infty,2}, \ldots, z_{\infty,N} \end{bmatrix}^T \right\} \]

... detailed notation ...

\[
\begin{bmatrix}
    z_{\infty,1} & 0 & 0 & \ldots & 0 \\
    0 & z_{\infty,2} & 0 & \ldots & 0 \\
    0 & 0 & z_{\infty,3} & \ldots & 0 \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    0 & 0 & 0 & \ldots & z_{\infty,N}
\end{bmatrix}
\]

If all Eigen values (poles) are different, then also the existence of the inverse transformation matrix can be ensured!
Transforming to a diagonal structure (continued)

We obtain for the transformed equivalent system:

- System matrix: \( A_q = \text{diag}\left\{ [z_{\infty,1}, z_{\infty,2}, \ldots, z_{\infty,N}]^T \right\} \),
- Steering matrix: \( B_q = E^{-1} B \),
- Observation matrix: \( C_q = C E \),
- Pass through matrix: \( D_q = D \).

Making the system matrix diagonal by means of a transformation using a matrix containing the Eigen vectors (respectively the inverse of this matrix) leads to the so-called parallel form (forth canonical structure). This structure is also called diagonal form.

In this structure all Eigen values (and thus all onsets) are decoupled. This leads usually to an increase in robustness (important for fixed-point architecture realizations, etc.).
Transforming to a diagonal structure (continued)

Structure of the resulting state-space description

\[ A_q = \text{diag}\{z_{\infty,1}, \ldots, z_{\infty,N}\}^T \]
Contents of the Part on State-Space Description and System Realizations

**This part:**
- Introduction
- Basic structure
- Application example
- From difference equations to state-space representations
- Signal-flow graphs
- Signal-flow graph representation of basic structures
- Transfer matrix, impulse-response matrix, and transition matrix
- Equivalent Realizations

**Next part:**
- Extensions