Modeling and Analysis of Systems
Lectures #4/5 - State-space Representation of LTI Systems

Guillaume Drion
Academic year 2016-2017
Outline (lectures #4 and #5)

- Zero-state response and zero-input response
- Block diagram and state-space representation (discrete and continuous)
- Solutions of state-space equations: transition matrix and matrix exponential
- Computing the matrix exponential
- State-space transformation
- Input-output approach vs state-space approach
Outline

- Zero-state response and zero-input response
- Block diagram and state-space representation (discrete and continuous)
- Solutions of state-space equations: transition matrix and matrix exponential
- Computing the matrix exponential
- State-space transformation
- Input-output approach vs state-space approach
Can we **mathematically describe** a LTI system using the following relationship?

\[ y = \mathcal{S}(u) \]

Yes, a LTI system is "fully" characterized by its **impulse response** \( h[\cdot] \) or \( h(\cdot) \). The impulse response is the response of the system to a pulse \( \delta[\cdot] \) or \( \delta(\cdot) \).

\[
y[\cdot] = \sum_{k \in \mathbb{Z}} u[k] h[\cdot - k]
\]

\[
y(\cdot) = \int_{-\infty}^{+\infty} u(\tau) h(\cdot - \tau) d\tau = u \ast h
\]
Can we derive a solution of a model using the impulse response?

Ex: RC circuit.

\[ \dot{v}_C + \frac{v_C}{RC} = \frac{V}{RC} \]

Impulse response of the RC circuit: \[ h(t) = \frac{1}{RC} \mathbb{I}(t)e^{-\frac{t}{RC}} \] (trust me for now).

It gives the solution: \[ v_C(t_0 + t) = \frac{1}{RC} \int_{-\infty}^{t_0+t} e^{-\frac{(t_0+t-\tau)}{RC}} V(\tau) d\tau \]
Input-output representation of LTI systems

- Ex: RC circuit.

\[ \dot{v}_C + \frac{v_C}{RC} = \frac{V}{RC} \]

- Solution using impulse response:
  \[ v_C(t_0 + t) = \frac{1}{RC} \int_{-\infty}^{t_0+t} e^{-\frac{(t_0+t-\tau)}{RC}} V(\tau) \, d\tau \]

- Solution using the tools you learned in calculus:
  \[ v_C(t_0+t) = e^{-\frac{t}{RC}} v_C(t_0) + \frac{1}{RC} \int_{t_0}^{t_0+t} e^{-\frac{(t_0+t-\tau)}{RC}} V(\tau) \, d\tau \]
Zero state response of a LTI system

- **Solution using impulse response:** \( v_C(t_0 + t) = \frac{1}{RC} \int_{-\infty}^{t_0+t} e^{-\frac{(t_0+t-\tau)}{RC}} V(\tau) d\tau \)

- **Solution using the tools you learned in mathematical analysis:**

\[
v_C(t_0+t) = e^{-\frac{t}{RC}} v_C(t_0) + \frac{1}{RC} \int_{t_0}^{t_0+t} e^{-\frac{(t_0+t-\tau)}{RC}} V(\tau) d\tau
\]

- The solutions are equivalent if
  - \( v_C(t_0) = 0 \): system at rest (blue).
  - \( u(t) = 0 \) \( \forall t < t_0 \): no input in the past (red).

Zero state conditions \( \rightarrow \) **zero state response.**
Zero state response and zero input response of a LTI system

- The solution of a LTI system contains two components: the **zero state response** (in red) and the **zero input response** (in blue).

\[ v_C(t_0+t) = e^{\frac{-t}{RC}}v_C(t_0) + \frac{1}{RC} \int_{t_0}^{t_0+t} e^{\frac{(t_0+t-\tau)}{RC}}V(\tau)d\tau \]

- **Zero input response**: takes the past values of the states into account (here \( v_C(t_0) \)), but not the input (here \( V(t) \)).

- **Zeros state response**: takes the input into account, but not the past values of the states (zero state conditions).
Zero state response and zero input response of a LTI system

- The solution of a LTI system contains two components: the zero state response (in red) and the zero input response (in blue).

\[ v_C(t_0 + t) = e^{-\frac{t}{RC}} v_C(t_0) + \frac{1}{RC} \int_{t_0}^{t_0 + t} e^{-\frac{(t_0 + t - \tau)}{RC}} V(\tau) d\tau \]

- The impulse response only “gives access” to the zero state response. To have an accurate description of the system, we need the complete history of the input.

- How can we derive the zero-input response? This response depends on the past values of system states \( \text{states} \rightarrow \text{state-space representation}! \)
Outline

- Zero-state response and zero-input response

- Block diagram and state-space representation (discrete and continuous)

- Solutions of state-space equations: transition matrix and matrix exponential

- Computing the matrix exponential

- State-space transformation

- Input-output approach vs state-space approach
Limitations of input-output approach: \( y(\cdot) = \int_{-\infty}^{+\infty} u(\tau)h(\cdot - \tau) d\tau \)

(i) Zero state: does not handle non-zero initial conditions.
(ii) We have to know the complete history of the input.

We have to define “intermediate variables” that store the past condition of the system and therefore relevant past values of the input: the states.
State-space representation of LTI systems

- Limitations of input-output approach: \( y(\cdot) = \int_{-\infty}^{+\infty} u(\tau) h(\cdot - \tau) d\tau \)
  
  (i) Zero state: does not handle non-zero initial conditions.
  (ii) We have to know the complete history of the input.

We have to define “intermediate variables” that store the past condition of the system and therefore relevant past values of the input: the states.

- A LTI system can be represented as follows

\[
\begin{align*}
\dot{x} &= Ax + Bu \\
y &= Cx + Du
\end{align*}
\]

\[
\begin{align*}
\sigma^{-1}x &= Ax + Bu \\
y &= Cx + Du
\end{align*}
\]

where \( x \) is the state vector whose dimension gives the dimension of the system.
State-space representation of LTI systems

- A LTI system can be represented as follows

\[
\begin{align*}
\dot{x} &= Ax + Bu \\
y &= Cx + Du
\end{align*}
\]

where \( x \) is the state vector whose dimension gives the dimension of the system.

- In a single-input/single-output (SISO) system, we have (ex: 4 states)

\[
A = \begin{pmatrix} & & & \\ & & & \\ & & & \\ \end{pmatrix} \quad B = \begin{pmatrix} \cdot \\
\cdot \\
\cdot \\
\cdot \\ \cdot \\
\cdot \\
\cdot \\
\cdot \\ \cdot \\
\cdot \\
\cdot \\
\end{pmatrix} \quad C = (\cdot \cdot \cdot \cdot) \\
D = (\cdot)
\]
State-space representation of LTI systems

- A LTI system can be represented as follows

\[
\begin{align*}
\dot{x} &= Ax + Bu \\
y &= Cx + Du
\end{align*}
\]

where \( x \) is the state vector whose dimension gives the dimension of the system.

- In a multiple-inputs/multiple-outputs (MIMO) system (\( m \) inputs, \( p \) outputs, \( n \) states):

\[
\begin{align*}
A &= n \times n \\
B &= n \times m \\
C &= p \times n \\
D &= p \times m
\end{align*}
\]

Update function

Output function
Example: RC circuit

The history of the input $V(t)$ is **stored in the charges accumulated around the capacitor**. This energy storage constitutes the state of the system.

$$v_C(t_0+t) = e^{-\frac{t}{RC}}v_C(t_0) + \frac{1}{RC} \int_{t_0}^{t_0+t} e^{-\frac{(t_0+t-\tau)}{RC}} V(\tau) d\tau$$

If we now the value of the state at $t_0$ ($v_C(t_0)$), we can just integrate the input between $t_0$ and $t_0+t$. 
State-space representation of LTI systems

- Some LTI systems cannot be represented with a finite state-space!

- Example: delay \( y(t) = u(t - T) \)

  - At \( t = t_0 \), we need to know all the values of the input between \( t_0 - T \) and \( t_0 \).

  - In continuous time, it represents an infinite number of values to be stored, i.e. an infinite number of states.

  - In discrete time, a delay can be represented by a finite number of states!
State-space representation of LTI systems

□ What kind of LTI systems admit a state-space representation?

□ Continuous time:

\[
\sum_{k=0}^{N} a_k \frac{d^k y(t)}{dt^k} = \sum_{k=0}^{N} b_k \frac{d^k u(t)}{dt^k} \quad \text{yes if} \quad a_N \neq 0
\]

□ Discrete time:

\[
\sum_{k=0}^{N} a_k y[n - k] = \sum_{k=0}^{N} b_k u[n - k] \quad \text{yes if} \quad a_N \neq 0
\]
We will derive a **general structure** for the state-space representation of LTI systems in continuous time.

Let’s start with a first order system (i.e. one state):

\[
\dot{y}(t) + a_0 y(t) = b_0 u(t)
\]

This system can be done with **one integrator** (not a differentiator!!), **multipliers**, and **one adder** (or summer).

We will start by building a block diagram of the system using integrators, multipliers and adders.
Block diagram of a first order system in continuous time

Block diagram of the first order system \( \dot{y}(t) + a_0 y(t) = b_0 u(t) \)
Block diagram of a first order system in continuous time

\[ \dot{y}(t) + a_0 y(t) = b_0 u(t) \]
Block diagram of a first order system in continuous time

\[ \dot{y}(t) + a_0 y(t) = b_0 u(t) \]
Block diagram of a first order system in continuous time

Block diagram of the first order system $\dot{y}(t) + a_0y(t) = b_0u(t)$

- $u$ goes into $b_0$.
- $\dot{y}$ goes into $f$.
- $y$ goes back in $u$ for the next “update”.
- $a_0$ effect of the past value of $y$!
Block diagram of a first order system in continuous time

- Block diagram of the first order system: \( \dot{y}(t) + a_0y(t) = b_0u(t) \)

- Input-output approach using the block diagram:
  If \( y(0) = 0 \), we have
  \[
y(t) = \int_0^t (b_0u(\tau) - a_0y(\tau))d\tau, \quad t \geq 0
\]
Block diagram of a first order system in continuous time

\[ y(t) + a_0 y(t) = b_0 u(t) \]

Let's define a state \( x \) in which we store the history of the system (here, \( y = x \)):

\[ \dot{x} = -a_0 x + b_0 u \]
Block diagram of a first order system in continuous time

- Block diagram of the first order system: $\dot{y}(t) + a_0 y(t) = b_0 u(t)$

Now, we have $x(t) = x(t_0) + \int_{t_0}^{t} (b_0 u(\tau) - a_0 x(\tau)) d\tau$, $y(t) = x(t)$

The past of the system is stored in the integrator. States are therefore the outputs of the integrators.
The past of the system (or energy) is stored in the integrators. At \( t = t_0 + t_1 \), the integral contains the **history of the past values**.
State-space representation/block diagram in continuous time

- **Block diagram**: representation of the system using **integrators**.

- **State-space**: representation of the system using **states**, which are the **outputs of the integrators**!

- Example: RLC circuit

  \[ v_L = L \frac{di_L}{dt} \implies i_L(t) = \frac{1}{L} \int_0^t v_L(\tau) d\tau \]

  \[ i_C = C \frac{dv_C}{dt} \implies v_C(t) = \frac{1}{C} \int_0^t i_C(\tau) d\tau \]

  the current in the inductor and the voltage across the capacitor are outputs of integrators!
A N-dimensional system in continuous time

Let’s first consider the case

$$\sum_{k=0}^{N} a_k y^{(k)} = b_0 u, \ a_N = 1$$

Which can be written

$$y^{(N)} = -\sum_{k=0}^{N-1} a_k y^{(k)} + b_0 u$$

(incorrect in the book)
Block diagram of a N-dimensional system in continuous time

\[ y^{(N)} = - \sum_{k=0}^{N-1} a_k y^{(k)} + b_0 u \]
Block diagram of a N-dimensional system in continuous time

\[ y^{(N)} = - \sum_{k=0}^{N-1} a_k y^{(k)} + b_0 u \]
Block diagram of a N-dimensional system in continuous time

\[ y^{(N)} = - \sum_{k=0}^{N-1} a_k y^{(k)} + b_0 u \]
Block diagram of a \( N \)-dimensional system in continuous time

\[
y^{(N)} = - \sum_{k=0}^{N-1} a_k y^{(k)} + b_0 u
\]
Block diagram of a N-dimensional system in continuous time

\[ y^{(N)} = - \sum_{k=0}^{N-1} a_k y^{(k)} + b_0 u \]

States are outputs of integrators:

\[ x_1 = y, \ x_2 = y^{(1)}, \ x_3 = y^{(3)}, \ldots, \ x_n = y^{(N-1)} \]
State-space representation of a N-dimensional system in continuous time

\[ y^{(N)} = - \sum_{k=0}^{N-1} a_k y^{(k)} + b_0 u \]

- States are outputs of integrators:

\[ x_1 = y, x_2 = y^{(1)}, x_3 = y^{(3)}, \ldots, x_n = y^{(N-1)} \]

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3 \\
\dot{x}_3 &= x_4 \\
& \vdots \\
\dot{x}_{n-1} &= x_n \\
\dot{x}_n &= -a_0 x_1 - a_1 x_2 - \cdots - a_{N-1} x_n + b_0 u \\
y &= x_n
\end{align*}
\]
A general N-dimensional system in continuous time

Let’s now consider the general case

\[ \sum_{k=0}^{N} a_k y^{(k)} = \sum_{k=0}^{N} b_k u^{(k)}, \quad a_N = 1 \]

We have to consider an intermediate signal \( \nu \) where

\[ \sum_{k=0}^{N} a_k \nu^{(k)} = u \quad \text{and} \quad y = \sum_{k=0}^{N} b_k \nu^{(k)} \]

Justification:

\[ P(D)y = Q(D)u \iff P(D)\nu = u \quad \text{and} \quad y = Q(D)\nu \]
Block diagram of a general N-dimensional system in continuous time

\[ \sum_{k=0}^{N} a_k \nu^{(k)} = u \quad \text{and} \quad y = \sum_{k=0}^{N} b_k \nu^{(k)} \]
Block diagram of a general N-dimensional system in continuous time

\[
\sum_{k=0}^{N} a_k v^{(k)} = u \quad \text{and} \quad y = \sum_{k=0}^{N} b_k v^{(k)}
\]
Block diagram of a general N-dimensional system in continuous time

\[ \sum_{k=0}^{N} a_k u^{(k)} = u \quad \text{and} \quad y = \sum_{k=0}^{N} b_k u^{(k)} \]
Block diagram of a general N-dimensional system in continuous time

\[
\sum_{k=0}^{N} a_k v^{(k)} = u \quad \text{and} \quad y = \sum_{k=0}^{N} b_k v^{(k)}
\]
Block diagram of a general N-dimensional system in continuous time

\[ \sum_{k=0}^{N} a_k v^{(k)} = u \quad \text{and} \quad \sum_{k=0}^{N} b_k v^{(k)} = y \]
State-space of a general N-dimensional system in continuous time

\[
\sum_{k=0}^{N} a_k u^{(k)} = u \quad \text{and} \quad y = \sum_{k=0}^{N} b_k u^{(k)}
\]

The update function is the same: states are outputs of integrators:

\[
x_1 = y, \quad x_2 = y^{(1)}, \quad x_3 = y^{(3)}, \ldots, \quad x_n = y^{(N-1)}
\]

\[
\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3 \\
\dot{x}_3 &= x_4 \\
&\vdots \\
\dot{x}_{n-1} &= x_n \\
\dot{x}_n &= -a_0 x_1 - a_1 x_2 - \cdots - a_{N-1} x_n + u
\end{aligned}
\]
State-space of a general N-dimensional system in continuous time

\[ \sum_{k=0}^{N} a_k \nu^{(k)} = u \quad \text{and} \quad y = \sum_{k=0}^{N} b_k \nu^{(k)} \]

The **output function** is more complex:

\[ x_1 = y, \quad x_2 = y^{(1)}, \quad x_3 = y^{(3)}, \ldots, \quad x_n = y^{(N-1)} \]

\[ y = b_0 x_1 + b_1 x_2 + b_2 x_3 + \cdots + b_{N-1} x_n \]

\[ + b_N (u - a_0 x_1 - a_1 x_2 - \cdots - a_{N-1} x_n) \]
State-space of a general N-dimensional system in continuous time

\[ \sum_{k=0}^{N} a_k \nu^{(k)} = u \quad \text{and} \quad y = \sum_{k=0}^{N} b_k \nu^{(k)} \]

It gives the following state-space representation:

\[ x_1 = y, \quad x_2 = y^{(1)}, \quad x_3 = y^{(3)}, \ldots, \quad x_n = y^{(N-1)} \]

\[
A = \begin{bmatrix}
0 & 1 & 0 & \cdots & 0 \\
\vdots & \ddots & \ddots & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & 1 \\
-a_0 & -a_1 & -a_2 & \cdots & -a_{N-1}
\end{bmatrix}, \quad B = \begin{bmatrix}
0 \\
0 \\
\vdots \\
0 \\
1
\end{bmatrix}, \quad C = \begin{bmatrix}
b_0 - b_N a_0 & b_1 - b_N a_1 & \cdots & b_{N-1} - b_N a_{N-1}
\end{bmatrix}, \quad D = b_N.
State-space representation of LTI systems in discrete time

- We will derive a **general structure** for the state-space representation of LTI systems in discrete time.

- Let’s start with a first order system (i.e. one state):

  \[ y[n] + a_1 y[n - 1] = b_0 u[n] \]

- This system can be done with **one delay, multipliers**, and **one adder** (or summer).

- We will start by building a block diagram of the system using delays, multipliers and adders.
Block diagram of a first order system in discrete time

□ Block diagram of the first order system \( y[n] + a_1 y[n - 1] = b_0 u[n] \)
Block diagram of a first order system in discrete time.

\[ y[n] + a_1 y[n - 1] = b_0 u[n] \]
Block diagram of a first order system in discrete time

\[ y[n] + a_1 y[n - 1] = b_0 u[n] \]
Block diagram of a first order system in discrete time

- Block diagram of the first order system: \( y[n] + a_1 y[n - 1] = b_0 u[n] \)

- States are outputs of delays: 
  \[
  x[n] = y[n - 1] \\
  x[n + 1] = -a_1 x[n] + b_0 u[n]
  \]
Let’s now consider the general case

\[ \sum_{k=0}^{N} a_k y[n - k] = \sum_{k=0}^{N} b_k u[n - k], \quad a_0 = 1 \]

We have to consider an intermediate signal \( \nu \) where

\[ \sum_{k=0}^{N} a_k \nu[n - k] = u[n] \quad \text{and} \quad y[n] = \sum_{k=0}^{N} b_k \nu[n - k] \]

We can rewrite the first equation:

\[ \nu[n] = - \sum_{k=1}^{N} a_k \nu[n - k] + u[n] \]
State-space of a general N-dimensional system in discrete time

\[
\nu[n] = -\sum_{k=1}^{N} a_k \nu[n - k] + u[n] \quad \text{and} \quad y[n] = \sum_{k=0}^{N} b_k \nu[n - k]
\]

- States are **output of delays** (and \(x_1[n + 1] = x_2[n]\))

\[
x_1[n] = \nu[n - N], \quad x_2[n] = \nu[n - N + 1], \ldots, \quad x_N[n] = \nu[n - 1]
\]

\[
\begin{align*}
  x_1[n+1] &= x_2[n] \\
  x_2[n+1] &= x_3[n] \\
  x_3[n+1] &= x_4[n] \\
  &\vdots \\
  x_N[n+1] &= -a_N x_1[n] - a_{N-1} x_2[n] - \cdots - a_1 x_N[n]
\end{align*}
\]
State-space of a general N-dimensional system in discrete time

\[ \nu[n] = - \sum_{k=1}^{N} a_k \nu[n - k] + u[n] \quad \text{and} \quad y[n] = \sum_{k=0}^{N} b_k \nu[n - k] \]

- States are **output of delays** (and \( x_1[n + 1] = x_2[n] \))

\[ x_1[n] = \nu[n - N], \ x_2[n] = \nu[n - N + 1], \ldots, x_N[n] = \nu[n - 1] \]

\[
\begin{cases}
  x_1[n + 1] = x_2[n] \\
  x_2[n + 1] = x_3[n] \\
  x_3[n + 1] = x_4[n] \\
  \vdots \\
  x_N[n + 1] = -a_N x_1[n] - a_{N-1} x_2[n] - \cdots - a_1 x_N[n]
\end{cases}
\]

\[ y[n] = b_N x_1 + b_{N-1} x_2 + \cdots + b_1 x_N \]
\[ + b_0 (-a_N x_1 - a_{N-1} x_2 - \cdots - a_1 x_N) \]
State-space of a general N-dimensional system in discrete time

\[ \nu[n] = -\sum_{k=1}^{N} a_k \nu[n - k] + u[n] \quad \text{and} \quad y[n] = \sum_{k=0}^{N} b_k \nu[n - k] \]

- States are **output of delays** (and \( x_1[n + 1] = x_2[n] \))

\[ x_1[n] = \nu[n - N], x_2[n] = \nu[n - N + 1], \ldots, x_N[n] = \nu[n - 1] \]

\[
A = \begin{bmatrix}
0 & 1 & 0 & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & 0 \\
\vdots & & \ddots & \ddots & 0 \\
0 & \cdots & & 0 & 1 \\
-a_N & -a_{N-1} & -a_{N-2} & \cdots & -a_1
\end{bmatrix}, \quad \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix},
\]

\[
C = \begin{bmatrix} b_N - b_0 a_N \\ b_{N-1} - b_0 a_{N-1} \end{bmatrix}, \quad D = b_0.
\]
Block diagram of a general N-dimensional system in discrete time

\[ \nu[n] = - \sum_{k=1}^{N} a_k \nu[n-k] + u[n] \quad \text{and} \quad y[n] = \sum_{k=0}^{N} b_k \nu[n-k] \]
Outline

- Zero-state response and zero-input response
- Block diagram and state-space representation (discrete and continuous)
- Solutions of state-space equations: transition matrix and matrix exponential
- Computing the matrix exponential
- State-space transformation
- Input-output approach vs state-space approach
Solutions of state-space equations: discrete case

A. Update function: $x[k + 1] = Ax[k] + Bu[k]$, $x[0] = x_0$

$$x[1] = Ax[0] + Bu[0]$$
Solutions of state-space equations: discrete case

A. Update function: \( x[k + 1] = Ax[k] + Bu[k], \ x[0] = x_0 \)

\[
\begin{align*}
x[1] &= Ax[0] + Bu[0] \\
     &= A(Ax[0] + Bu[0]) + Bu[1] \\
     &= A^2x[0] + ABu[0] + Bu[1]
\end{align*}
\]
Solutions of state-space equations: discrete case

A. Update function: \( x[k + 1] = Ax[k] + Bu[k] \), \( x[0] = x_0 \)

\[
x[1] = Ax[0] + Bu[0]
\]

\[
= A(Ax[0] + Bu[0]) + Bu[1]
= A^2 x[0] + ABu[0] + Bu[1]
\]

\[
= A(A^2 x[0] + ABu[0] + Bu[1]) + Bu[2]
\]
Solutions of state-space equations: discrete case

A. Update function: \( x[k+1] = Ax[k] + Bu[k] \), \( x[0] = x_0 \)

\[
\begin{align*}
  x[1] &= Ax[0] + Bu[0] \\
        &= A(Ax[0] + Bu[0]) + Bu[1] \\
        &= A^2x[0] + ABu[0] + Bu[1] \\
        &= A(A^2x[0] + ABu[0] + Bu[1]) + Bu[2] \\
        \vdots \\
  x[n] &= A^nx[0] + \sum_{k=0}^{n-1} A^{n-1-k}Bu[k], \quad n > 0
\end{align*}
\]
Solutions of state-space equations: discrete case

□ **A.** Update function: \( x[k + 1] = Ax[k] + Bu[k] \), \( x[0] = x_0 \)

\[
x[n] = A^n x[0] + \sum_{k=0}^{n-1} A^{n-1-k} Bu[k], \quad n > 0
\]

□ \( A^n = AAA \cdots A \) is the **transition matrix** of the system \( x[k + 1] = Ax[k] \):

\[
x[n_2] = A^{n_2-n_1} x[n_1], \quad n_2 > n_1
\]

□ If you know the solution at any time \( n_0 \), the transition matrix gives you the solution at any **future time** \( n > n_0 \).
Solutions of state-space equations: discrete case

- **B.** Output function: \( y[k] = Cx[k] + Du[k] \)

Using \( x[n] = A^n x[0] + \sum_{k=0}^{n-1} A^{n-1-k} Bu[k], \ n > 0 \) we have

\[
y[n] = CA^n x[0] + \sum_{k=0}^{n-1} CA^{n-1-k} Bu[k] + Du[n]
\]

We can derive the **impulse response** by imposing the zero state conditions:
\( x[0] = 0, \ u[k] = 0 \ \forall k < 0 \)

\[
y[n] = \sum_{k=0}^{n-1} CA^{n-1-k} Bu[k] + Du[n] = \sum_{k \in \mathbb{Z}} u[k] h[n - k]
\]
We can derive the **impulse response** by imposing the zero state conditions:

\[ x[0] = 0, \ u[k] = 0 \ \forall k < 0 \]

\[
y[n] = \sum_{k=0}^{n-1} CA^{n-1-k} Bu[k] + Du[n]
\]

\[ = \sum_{k \in \mathbb{Z}} u[k] h[n - k] \]

It gives the impulse response

\[ h[n] = CA^{n-1}[n-1] + D\delta[n] \]

Indeed, **blue** implies that \( h[n - k] = 0 \ \forall k > n - 1 \) (causality),
**red** implies that \( u[k] = 0 \ \forall k < 0 \) (zero state conditions).
Outline (lectures #4 and #5)

- Zero-state response and zero-input response
- Block diagram and state-space representation (discrete and continuous)
- Solutions of state-space equations: transition matrix and matrix exponential
- Computing the matrix exponential
- State-space transformation
- Input-output approach vs state-space approach
Outline

- Zero-state response and zero-input response
- Block diagram and state-space representation (discrete and continuous)
- Solutions of state-space equations: transition matrix and matrix exponential
- Computing the matrix exponential
- State-space transformation
- Input-output approach vs state-space approach
Solutions of state-space equations: continuous case

- **A. Update function:** $\dot{x} = Ax + Bu$.

- Let's start with the autonomous system $\dot{x} = Ax$, $x(0) = x_0$ (zero-input).

  - If $A = a$ (scalar, i.e. one-dimensional case), we have $\dot{x} = ax$, $x(0) = x_0$, which gives the solution (zero-input response):
    
    $$x(t) = x_0 e^{at}$$

- We want to **extend this solution** to the general, N-dimensional case!

  $$x(t) = x_0 e^{At}$$
Solutions of state-space equations: the matrix exponential

- We want to extend the 1D solution to the general, N-dimensional case.

\[ x(t) = x_0 e^{at} \quad \rightarrow \quad x(t) = x_0 e^{At} \]

- We therefore need to define a matrix where \( e^{At} \) has the same Taylor series as \( e^{at} \): the **matrix exponential**.

- Reminder: **Taylor series** represent a function as an infinite sum of terms calculated from the values of the function derivatives at one point.

\[
f(x) = f(a) + \frac{f'(a)}{1!} (x - a)^1 + \frac{f''(a)}{2!} (x - a)^2 + \frac{f'''(a)}{3!} (x - a)^3 + \cdots
\]

\[
= \sum_{n=0}^{\infty} \frac{f^n(a)}{n!} (x - a)^n
\]
Solutions of state-space equations: the matrix exponential

- We want to extend the 1D solution to the general, N-dimensional case.
  \[ x(t) = x_0 e^{at} \rightarrow x(t) = x_0 e^{At} \]

- We therefore need to define a matrix where \( e^{At} \) has the same Taylor series as \( e^{at} \): the **matrix exponential**.

- Here, we use the values of \( e^{at}/e^{At} \) at \( t = 0 \), which gives
  \[
  e^{at} = 1 + \frac{(at)^1}{1!} + \frac{(at)^2}{2!} + \frac{(at)^3}{3!} + \cdots
  \]
  and therefore
  \[
  e^{At} = I + \frac{(At)^1}{1!} + \frac{(At)^2}{2!} + \frac{(At)^3}{3!} + \cdots
  \]
Solutions of state-space equations: the matrix exponential

- If the matrix exponential has the same Taylor expansion than the scalar exponential, it ensures that

\[
\frac{de^{At}}{dt} = Ae^{At}
\]

- Proof:

\[
\frac{de^{At}}{dt} = \frac{d}{dt} \sum_{k=0}^{\infty} \frac{A^k t^k}{k!} = \sum_{k=1}^{\infty} k \frac{A^k t^{k-1}}{k!}
\]

if we use the change of variable \( l = k - 1 \), it yields

\[
\frac{de^{At}}{dt} = \sum_{l=0}^{\infty} (l + 1) \frac{A^{l+1} t^l}{(l + 1)!} = A \sum_{l=0}^{\infty} \frac{A^l t^l}{l!} = Ae^{At}
\]
Solutions of state-space equations: the matrix exponential

The system $\dot{x} = Ax$, $x(0) = x_0$ has the solution

$$x(t) = x_0 e^{At}$$

where $A$ is the **matrix exponential** and $e^{At}$ is the **transition matrix**:

$$x(t_2) = e^{A(t_2-t_1)}x(t_1), \quad t_2 > t_1$$

Reminder: in the discrete case, we had

$$x[n_2] = A^{n_2-n_1}x[n_1], \quad n_2 > n_1$$
Solutions of state-space equations

Now, we will try to find the complete solution of the system

\[ \dot{x} = Ax + Bu, \quad x(0) = x_0 \]

We can define \( z(t) = x(t)e^{-At} \) such that

\[
\dot{z}(t) = -Ae^{-At}x(t) + e^{-At}(Ax(t) + Bu(t)) = e^{-At}Bu(t)
\]

The solution of \( z(t) \) for any input \( u(t) \) is therefore

\[
z(t) = z(0) + \int_0^t e^{-A\tau} Bu(\tau) d\tau
\]
Solutions of state-space equations: zero-input and zero-state responses

- The solution of $z(t)$ for any input $u(t)$ is

$$z(t) = z(0) + \int_0^t e^{-At} Bu(\tau) d\tau$$

- We can use the change of variable $x(t) = z(t)e^{At}$, which gives

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)} Bu(\tau) d\tau$$

  zero input        zero state
Solutions of state-space equations: impulse response

- General solution of the system:

\[ y(t) = Cx + Du \]
\[ = Ce^{At}x_0 + C \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau + Du(t) \]

- To extract the impulse response, we impose the **zero-state conditions**, which gives

\[ h(t) = Ce^{At}B\mathbb{I}(t) + D\delta(t) \]
Outline

- Zero-state response and zero-input response

- Block diagram and state-space representation (discrete and continuous)

- Solutions of state-space equations: transition matrix and matrix exponential

- Computing the matrix exponential

- State-space transformation

- Input-output approach vs state-space approach
Computing the matrix exponential: the Jordan form

- The solution of the system involves $A$ only if $A$ is the matrix exponential, i.e. it has to respect the properties described earlier.

- It means that to find the solution of the state-space equations, the matrix $A$ has to be into a specific form, called the Jordan form (see textbook).

- This implies that we have to change the variables of the state-space representation, i.e. make a state-space transformation (same to reach the canonical form).
Outline

☐ Zero-state response and zero-input response

☐ Block diagram and state-space representation (discrete and continuous)

☐ Solutions of state-space equations: transition matrix and matrix exponential

☐ Computing the matrix exponential

☐ State-space transformation

☐ Input-output approach vs state-space approach
We already saw that the state-space representation of a system is not unique.

Example: in an RC circuit, the state relating to the accumulation of energy in the capacitance can be either the charges accumulated on each plate, or the voltage gradient between the plates, among others.

A state-space transformation is performed by imposing a change of variable $x = Tz$, which gives

$$\begin{align*}
\dot{z} &= T^{-1}ATz + T^{-1}Bu \\
y &= CTz + Du
\end{align*}$$

where $A, B, C, D$ are the matrix of the system
State-space equivalence

Two state space representations

\[
\begin{align*}
\dot{x} &= A_1 x + B_1 u \\
y &= C_1 x + D_1 u
\end{align*}
\quad 
\begin{align*}
\dot{x} &= A_2 x + B_2 u \\
y &= C_2 x + D_2 u
\end{align*}
\]

are equivalent if there exists a matrix \( T \) such that

\[
A_2 = T^{-1} A_1 T, \quad B_2 = T^{-1} B_1, \quad C_2 = C_1 T, \quad D_2 = D_1
\]

This is for instance very useful to put a system into the **canonical form** (analysis and design of closed-loop systems).
Outline

- Zero-state response and zero-input response
- Block diagram and state-space representation (discrete and continuous)
- Solutions of state-space equations: transition matrix and matrix exponential
- Computing the matrix exponential
- State-space transformation
- Input-output approach vs state-space approach
Input-output approach vs state-space approach

- Input-output approach: we describe the system using the relationship

\[ y = S(u) \]

This approach “only” gives access to the **zero-state response**.

- State-space approach: we describe the system using the system of equations

\[
\begin{align*}
\dot{x} &= Ax + Bu \\
y &=Cx + Du
\end{align*}
\]

This approach gives access to a complete solution, but is **more complex**, especially if we consider a high-dimensional system.
Input-output approach vs state-space approach

- Input-output approach: we describe the system using the relationship

\[ y = S(u) \]

This approach “only” gives access to the **zero-state response**.

- So far, we have decomposed the input signals into a sum of pulses, which showed that the system could be characterized using the **impulse response**.

- In the second part of the course, we will decompose the input signals into a sum of (co)sines of different frequencies, which will give us access to the **frequency response** of a system:

  from the **time domain** to the **frequency domain**.