Systems Dynamics

Course ID: 267MI – Fall 2023

Thomas Parisini
Gianfranco Fenu

University of Trieste
Department of Engineering and Architecture

267MI –Fall 2023

Lecture 1

Generalities: Systems and Models

Lecture 1: Table of Contents

1. Generalities: Systems and Models

Systems Dynamics

Systems

Inputs ("causes")

$$u(t) = \begin{bmatrix} u_1(t) \\ \vdots \\ u_m(t) \end{bmatrix} \in \Re^m$$

Outputs ("effects")

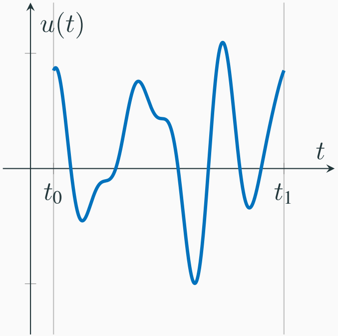
$$y(t) = \begin{bmatrix} y_1(t) \\ \vdots \\ y_m(t) \end{bmatrix} \in \Re^p$$

Definition of the "system" entity to be analysed $\Longrightarrow$ Physical laws, a priori knowledge, heuristic considerations, statistical evidence, etc. $\Longrightarrow$ *Mathematical models*: algebraic and/or differential and/or difference equations

Dynamic Systems Described by State Equations

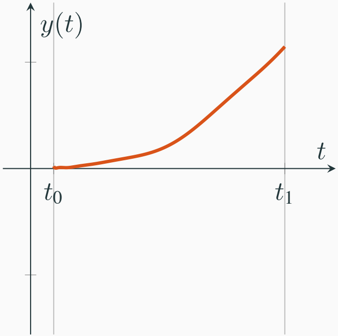
Recalling from the *Fundamentals in Control* course

What is the meaning of ”Dynamic”?



?

Can $y(t)$ be determined in a unique way?

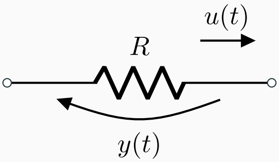


If the answer is **“NO”** → The system is a **dynamic system**.

Dynamic Systems Described by State Equations

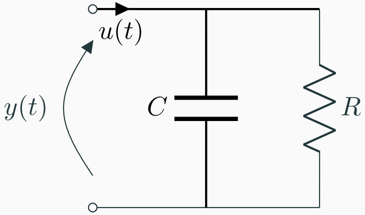
Dynamic Systems

Dynamic Systems: Examples



$$y(t) = R \cdot u(t)$$

The system is **NOT** dynamic

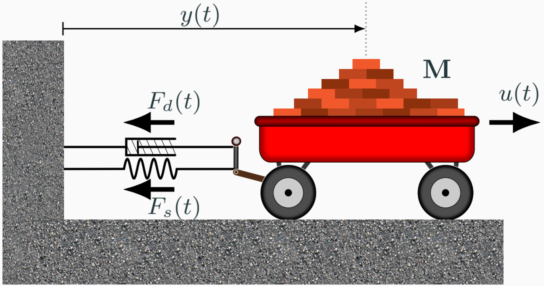


$$\left.\begin{array}{l} u(t),\ t \in [t_0, t_1] \\ y(t_0) \end{array}\right\}$$

$$\implies y(t),\ t \in [t_0, t_1]$$

The system is **dynamic**

Dynamic Systems: Examples



$$\left.\begin{array}{l} u(t),\ t \in [t_0, t_1] \\ y(t_0) \\ \dot{y}(t_0) \end{array}\right\} \implies y(t),\ t \in [t_0, t_1]$$

The system is **dynamic**

State Variables: a Qualitative Definition

State variables

Variables to be known at time $t = t_0$ in order to be able to determine the output $y(t),\ t \geq t_0$ from the knowledge of the input $u(t),\ t \geq t_0$:

$$x_i(t),\ i = 1, 2, \ldots, n \qquad \text{(state variables)}$$

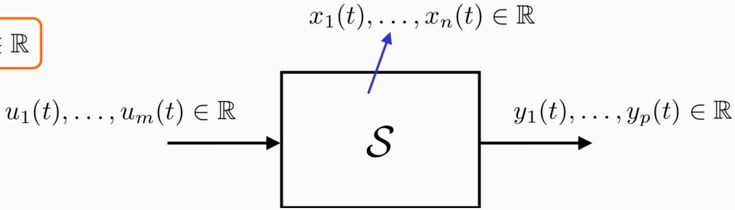
$\cdots$ In more **rigorous** terms $\Longrightarrow$

Dynamic Systems Described by State Equations

Continuous-time State Equations

Continuous-time State Equations

$\forall\, t \in \mathbb{R}$



State equations (dynamic)

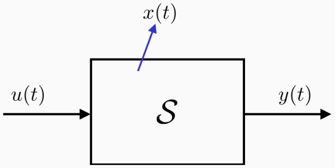
$$\begin{cases} \dot{x}_1(t) = f_1(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t), t) \\ \quad \vdots \\ \dot{x}_n(t) = f_n(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t), t) \end{cases}$$

Output equations (algebraic)

$$\begin{cases} y_1(t) = g_1(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t), t) \\ \quad \vdots \\ y_p(t) = g_p(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t), t) \end{cases}$$

Continuous-time State Equations (cont.)

$$u(t) = \begin{bmatrix} u_1(t) \\ \vdots \\ u_m(t) \end{bmatrix} \in \mathbb{R}^m, \quad y(t) = \begin{bmatrix} y_1(t) \\ \vdots \\ y_p(t) \end{bmatrix} \in \mathbb{R}^p$$



$$x(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{bmatrix} \in \mathbb{R}^n$$

$$f(x,u,t) = \begin{bmatrix} f_1(x,u,t) \\ \vdots \\ f_n(x,u,t) \end{bmatrix} \in \mathbb{R}^n$$

$$g(x,u,t) = \begin{bmatrix} g_1(x,u,t) \\ \vdots \\ g_p(x,u,t) \end{bmatrix} \in \mathbb{R}^p$$

Compact form

$$\begin{cases} \dot{x}(t) = f(x(t), u(t), t) \\ y(t) = g(x(t), u(t), t) \end{cases}$$

Linear Dynamic Systems

Consider the continuous-time dynamic system state-space representation:

$$\begin{cases} \dot{x}(t) = f(x(t), u(t), t) \\ y(t) = g(x(t), u(t), t) \end{cases}$$

This state-space equation describes a **linear system** if and only if the functions $f(\cdot)$ and $g(\cdot)$ are **linear with respect to their state and input vector arguments**:

$$\forall\, \alpha_1, \alpha_2 \in \mathbb{R},\, \forall\, x_1, x_2 \in \mathbb{R}^n,\, \forall\, u_1, u_2 \in \mathbb{R}^m \;:$$

$$f(\alpha_1 x_1 + \alpha_2 x_2, \alpha_1 u_1 + \alpha_2 u_2, t) = \alpha_1 f(x_1, u_1, t) + \alpha_2 f(x_2, u_2, t)$$

$$g(\alpha_1 x_1 + \alpha_2 x_2, \alpha_1 u_1 + \alpha_2 u_2, t) = \alpha_1 g(x_1, u_1, t) + \alpha_2 g(x_2, u_2, t)$$

Linear Dynamic Systems: Matrix Form

Consider the state-space representation:

$$\begin{cases} \dot{x}(t) = f(x(t), u(t), t) \\ y(t) = g(x(t), u(t), t) \end{cases}$$

and suppose that the linearity assumption holds. Then:

$$\begin{cases} f_1(x,u,t) = a_{11}(t)x_1 + \cdots + a_{1n}(t)x_n + b_{11}(t)u_1 + \cdots + b_{1m}(t)u_m \\ \vdots \\ f_n(x,u,t) = a_{n1}(t)x_1 + \cdots + a_{nn}(t)x_n + b_{n1}(t)u_1 + \cdots + b_{nm}(t)u_m \\ \\ y_1 = c_{11}(t)x_1 + \cdots + c_{1n}(t)x_n + d_{11}(t)u_1 + \cdots + d_{1m}(t)u_m \\ \vdots \\ y_p = c_{p1}(t)x_1 + \cdots + c_{pn}(t)x_n + d_{p1}(t)u_1 + \cdots + d_{pm}(t)u_m \end{cases}$$

where $a_{ij}(t), b_{ij}(t), c_{ij}(t), d_{ij}(t)$ are generic functions of the time instant t.

Linear Dynamic Systems: Matrix Form (cont.)

Letting:

$$A(t) := \begin{bmatrix} a_{11}(t) & \cdots & a_{1n}(t) \\ \vdots & \ddots & \vdots \\ a_{n1}(t) & \cdots & a_{nn}(t) \end{bmatrix} ; \quad B(t) := \begin{bmatrix} b_{11}(t) & \cdots & b_{1m}(t) \\ \vdots & \vdots & \vdots \\ b_{n1}(t) & \cdots & b_{nm}(t) \end{bmatrix}$$

$$C(t) := \begin{bmatrix} c_{11}(t) & \cdots & c_{1n}(t) \\ \vdots & \ddots & \vdots \\ c_{p1}(t) & \cdots & c_{pn}(t) \end{bmatrix} ; \quad D(t) := \begin{bmatrix} d_{11}(t) & \cdots & d_{1m}(t) \\ \vdots & \vdots & \vdots \\ d_{p1}(t) & \cdots & d_{pm}(t) \end{bmatrix}$$

$$x(t) := \begin{bmatrix} x_1(t) & \cdots & x_n(t) \end{bmatrix}^T ; \; u(t) := \begin{bmatrix} u_1(t) & \cdots & u_m(t) \end{bmatrix}^T ; \; y(t) := \begin{bmatrix} y_1(t) & \cdots & y_p(t) \end{bmatrix}^T$$

One gets:

$$\begin{cases} \dot{x}(t) = A(t)x(t) + B(t)u(t) \\ \\ y(t) = C(t)x(t) + D(t)u(t) \end{cases}$$

Time-Invariant Linear Dynamic Systems

In the **time-invariant** scenario, the matrices $A(t), B(t), C(t), D(t)$ do not depend on the time-index k, that is are **constant** matrices A, B, C, D:

$$A := \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \; ; \quad B := \begin{bmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & \vdots & \vdots \\ b_{n1} & \cdots & b_{nm} \end{bmatrix}$$

$$C := \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{p1} & \cdots & c_{pn} \end{bmatrix} \; ; \quad D := \begin{bmatrix} d_{11} & \cdots & d_{1m} \\ \vdots & \vdots & \vdots \\ d_{p1} & \cdots & d_{pm} \end{bmatrix}$$

and thus:

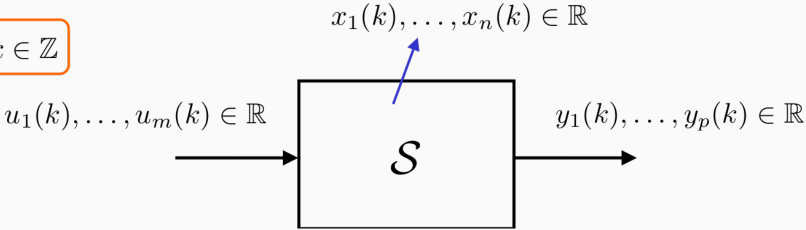
$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ \\ y(t) = Cx(t) + Du(t) \end{cases}$$

Dynamic Systems Described by State Equations

Discrete-time State Equations

Discrete-time State Equations

$\forall\, k \in \mathbb{Z}$



State equations (dynamic)

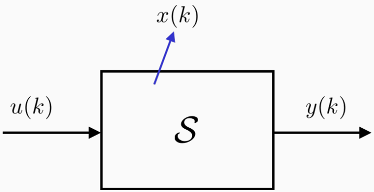
$$\begin{cases} x_1(k+1) = f_1(x_1(k), \ldots, x_n(k), u_1(k), \ldots, u_m(k), k) \\ \quad \vdots \\ x_n(k+1) = f_n(x_1(k), \ldots, x_n(k), u_1(k), \ldots, u_m(k), k) \end{cases}$$

Output equations (algebraic)

$$\begin{cases} y_1(k) = g_1(x_1(k), \ldots, x_n(k), u_1(k), \ldots, u_m(k), k) \\ \quad \vdots \\ y_p(k) = g_p(x_1(k), \ldots, x_n(k), u_1(k), \ldots, u_m(k), k) \end{cases}$$

Discrete-time State Equations (cont.)

$$u(k) = \begin{bmatrix} u_1(k) \\ \vdots \\ u_m(k) \end{bmatrix} \in \mathbb{R}^m, \quad y(k) = \begin{bmatrix} y_1(k) \\ \vdots \\ y_p(k) \end{bmatrix} \in \mathbb{R}^p$$



$$x(k) = \begin{bmatrix} x_1(k) \\ \vdots \\ x_n(k) \end{bmatrix} \in \mathbb{R}^n$$

$$f(x,u,k) = \begin{bmatrix} f_1(x,u,k) \\ \vdots \\ f_n(x,u,k) \end{bmatrix} \in \mathbb{R}^n$$

$$g(x,u,k) = \begin{bmatrix} g_1(x,u,k) \\ \vdots \\ g_p(x,u,k) \end{bmatrix} \in \mathbb{R}^p$$

Compact form

$$\begin{cases} x(k+1) = f(x(k), u(k), k) \\ y(k) = g(x(k), u(k), k) \end{cases}$$

Linear Dynamic Systems

Consider the discrete-time dynamic system state-space representation:

$$\begin{cases} x(k+1) = f(x(k), u(k), k) \\ y(k) = g(x(k), u(k), k) \end{cases}$$

This state-space equation describes a **linear system** if and only if the functions $f(\cdot)$ and $g(\cdot)$ are **linear with respect to their state and input vector arguments**:

$$\forall\, \alpha_1, \alpha_2 \in \mathbb{R},\ \forall\, x_1, x_2 \in \mathbb{R}^n,\ \forall\, u_1, u_2 \in \mathbb{R}^m \ :$$

$$\begin{aligned} f(\alpha_1 x_1 + \alpha_2 x_2, \alpha_1 u_1 + \alpha_2 u_2, k) &= \alpha_1 f(x_1, u_1, k) + \alpha_2 f(x_2, u_2, k) \\ g(\alpha_1 x_1 + \alpha_2 x_2, \alpha_1 u_1 + \alpha_2 u_2, k) &= \alpha_1 g(x_1, u_1, k) + \alpha_2 g(x_2, u_2, k) \end{aligned}$$

Linear Dynamic Systems: Matrix Form

Consider the state-space representation:

$$\begin{cases} x(k+1) = f(x(k), u(k), k) \\ y(k) = g(x(k), u(k), k) \end{cases}$$

and suppose that the linearity assumption holds. Then:

$$\begin{cases} f_1(x, u, k) = a_{11}(k)x_1 + \cdots + a_{1n}(k)x_n + b_{11}(k)u_1 + \cdots + b_{1m}(k)u_m \\ \vdots \\ f_n(x, u, k) = a_{n1}(k)x_1 + \cdots + a_{nn}(k)x_n + b_{n1}(k)u_1 + \cdots + b_{nm}(k)u_m \\ \\ y_1 = c_{11}(k)x_1 + \cdots + c_{1n}(k)x_n + d_{11}(k)u_1 + \cdots + d_{1m}(k)u_m \\ \vdots \\ y_p = c_{p1}(k)x_1 + \cdots + c_{pn}(k)x_n + d_{p1}(k)u_1 + \cdots + d_{pm}(k)u_m \end{cases}$$

where $a_{ij}(k), b_{ij}(k), c_{ij}(k), d_{ij}(k)$ are generic functions of the discrete-time index k.

Linear Dynamic Systems: Matrix Form (cont.)

Letting:

$$A(k) := \begin{bmatrix} a_{11}(k) & \cdots & a_{1n}(k) \\ \vdots & \ddots & \vdots \\ a_{n1}(k) & \cdots & a_{nn}(k) \end{bmatrix} \; ; \quad B(k) := \begin{bmatrix} b_{11}(k) & \cdots & b_{1m}(k) \\ \vdots & \vdots & \vdots \\ b_{n1}(k) & \cdots & b_{nm}(k) \end{bmatrix}$$

$$C(k) := \begin{bmatrix} c_{11}(k) & \cdots & c_{1n}(k) \\ \vdots & \ddots & \vdots \\ c_{p1}(k) & \cdots & c_{pn}(k) \end{bmatrix} \; ; \quad D(k) := \begin{bmatrix} d_{11}(k) & \cdots & d_{1m}(k) \\ \vdots & \vdots & \vdots \\ d_{p1}(k) & \cdots & d_{pm}(k) \end{bmatrix}$$

$$x(k) := \begin{bmatrix} x_1(k) & \cdots & x_n(k) \end{bmatrix}^T \; ; \; u(k) := \begin{bmatrix} u_1(k) & \cdots & u_m(k) \end{bmatrix}^T \; ; \; y(k) := \begin{bmatrix} y_1(k) & \cdots & y_p(k) \end{bmatrix}^T$$

One gets:

$$\begin{cases} x(k+1) = A(k)x(k) + B(k)u(k) \\ \\ y(k) = C(k)x(k) + D(k)u(k) \end{cases}$$

Time-Invariant Linear Dynamic Systems

In the **time-invariant** scenario, the matrices $A(k), B(k), C(k), D(k)$ do not depend on the time-index k, that is are **constant** matrices A, B, C, D:

$$A := \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \; ; \quad B := \begin{bmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & \vdots & \vdots \\ b_{n1} & \cdots & b_{nm} \end{bmatrix}$$

$$C := \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{p1} & \cdots & c_{pn} \end{bmatrix} \; ; \quad D := \begin{bmatrix} d_{11} & \cdots & d_{1m} \\ \vdots & \vdots & \vdots \\ d_{p1} & \cdots & d_{pm} \end{bmatrix}$$

and thus:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ \\ y(k) = Cx(k) + Du(k) \end{cases}$$

Dynamic Systems Described by State Equations

An Example

Matlab live script

Do we obtain a valid discrete-time representation of a continuous-time dynamical system for whatever possible choice of the sampling time?

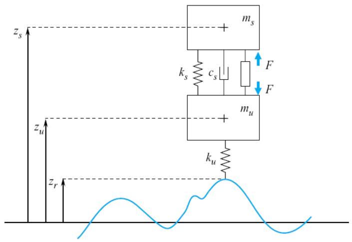
A **Matlab live script** is available, illustrating what are the effects of sampling on continuous-time dynamical systems.
Steps to retrieve the live script:

- Download as a ZIP archive the whole contents of the folder named "**L1_Sampling_Effects_LTI_Systems**," available in the "**Class Materials**" file area of the MS Teams course team.
- Uncompress the archive into a preferred folder and add the chosen folder and subfolders to the Matlab path.
- Open the live script using the Matlab command:

```
open('sampling_effects_LTI_systems.mlx');
```

An Example: Continuous-Time Model of a Car Suspension

From a real vehicle ...

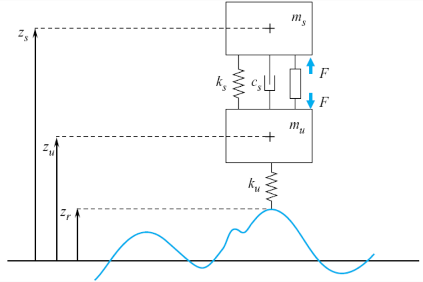


to a simplified *quarter-car model*

quarter-car model hypotheses

- vehicle as assembly of four decoupled parts
- each part consists of
  - the *sprung mass*: a quarter of the vehicle mass, supported by a suspension actuator, placed between the vehicle and the tyre
  - the *unsprung mass*: the wheel/tyre sub-assembly
- the model allows only for vertical motion: the vehicle is moving forward with an almost constant speed

Continuous-Time Model of a Car Suspension (cont.)



- inputs:
  - ground vertical position vs. the steady-state
  - active actuator force
- outputs:
  - sprung mass vertical acceleration
  - contact force between tyre and ground

- state variables:
  - vertical positions of sprung and unsprung masses vs. the corresponding steady-state values
  - vertical speeds of masses

$$
\begin{cases}
x_1(t) = z_s(t) - \bar{z}_s \\
x_2(t) = z_u(t) - \bar{z}_u \\
x_3(t) = \dot{x}_1(t) \\
x_4(t) = \dot{x}_2(t) \\
u_1(t) = z_r(t) - \bar{z}_r \\
u_2(t) = F(t) \\
y_1(t) = \ddot{x}_1 \\
y_2(t) = k_u\left(x_2(t) - u_1(t)\right)
\end{cases}
$$

Continuous-Time Model of a Car Suspension (cont.)

$$
\begin{cases}
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \dfrac{k_s}{m_s} & \dfrac{k_s}{m_s} & -\dfrac{c_s}{m_s} & \dfrac{c_s}{m_s} \\ \dfrac{k_s}{m_u} & -\dfrac{k_s + k_u}{m_u} & \dfrac{c_s}{m_u} & -\dfrac{c_s}{m_u} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \dfrac{1}{m_s} \\ \dfrac{k_s}{m_u} & -\dfrac{1}{m_u} \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\
\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} -\dfrac{k_s}{m_s} & \dfrac{k_s}{m_s} & -\dfrac{c_s}{m_s} & \dfrac{c_s}{m_s} \\ 0 & k_u & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & \dfrac{1}{m_s} \\ -k_u & 0 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}
\end{cases}
$$

Continuous-Time Car Suspension: an Example

Assuming

$$m_s = 400.0 \text{ kg} \qquad m_u = 50.0 \text{ kg} \qquad c_s = 2.0\,10^3 \text{ N s m}^{-1}$$

$$k_s = 2.0\,10^4 \text{ N m}^{-1} \qquad k_u = 2.5\,10^5 \text{ N m}^{-1}$$

the car suspension model becomes

$$\begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 1.0 \\ -50.0 & 50.0 & -5.0 & 5.0 \\ 400.0 & -5400.0 & 40.0 & -40.0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 2.5\,10^{-3} \\ 5.0\,10^3 & -2.0\,10^{-2} \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\ \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} -50.0 & 50.0 & -5.0 & 5.0 \\ 0 & 2.5\,10^5 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 2.5\,10^{-3} \\ -2.5\,10^5 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \end{cases}$$

Sampled-Time Car Suspension Models

Let's get a **sampled-time** description of the same dynamic system:

- How does the sampled-time description correlate with the continuous-time model?
- What happens if we increase or decrease the sampling rate? Does the sampled-time model change with the sampling time?
- Does the sampled-time model describe the behaviour of the continuous-time dynamic system for **any possible choice** of the sampling time value?

Sampled-Time Car Suspension Models (cont.)

Using *400 samples per second (SPS)* as sampling rate

$$
\left\{
\begin{array}{l}
\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \end{bmatrix} =
\begin{bmatrix}
9.9985 \cdot 10^{-1} & 8.6921 \cdot 10^{-5} & 2.4848 \cdot 10^{-3} & 1.5139 \cdot 10^{-5} \\
1.2010 \cdot 10^{-3} & 9.8372 \cdot 10^{-1} & 1.2111 \cdot 10^{-4} & 2.3662 \cdot 10^{-3} \\
-1.1819 \cdot 10^{-1} & 4.2490 \cdot 10^{-2} & 9.8803 \cdot 10^{-1} & 1.1905 \cdot 10^{-2} \\
9.4043 \cdot 10^{-1} & -1.2771 \cdot 10^{-1} & 9.5244 \cdot 10^{-2} & 8.8968 \cdot 10^{-1}
\end{bmatrix} \cdot
\begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} \\
\qquad + \begin{bmatrix}
6.3604 \cdot 10^{-5} & 7.5262 \cdot 10^{-9} \\
1.5076 \cdot 10^{-2} & -6.0051 \cdot 10^{-8} \\
7.5696 \cdot 10^{-2} & 5.9093 \cdot 10^{-6} \\
1.1831 \cdot 10^{+1} & -4.7021 \cdot 10^{-5}
\end{bmatrix} \cdot
\begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \\
\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} =
\begin{bmatrix}
-50.0 & 50.0 & -5.0 & 5.0 \\
0 & 2.5 \cdot 10^{+5} & 0 & 0
\end{bmatrix} \cdot
\begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k \end{bmatrix} +
\begin{bmatrix}
0 & 2.5 \cdot 10^{-3} \\
-2.5 \cdot 10^{+5} & 0
\end{bmatrix} \cdot
\begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}
\end{array}
\right.
$$

Sampled-Time Car Suspension Models (cont.)

Instead, using *15 samples per second (SPS)* as sampling rate

$$
\begin{cases}
\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \end{bmatrix} = \begin{bmatrix} 9.2495 \cdot 10^{-1} & -2.5315 \cdot 10^{-1} & 5.7487 \cdot 10^{-2} & 1.2779 \cdot 10^{-3} \\ 6.0514 \cdot 10^{-2} & -1.4515 \cdot 10^{-1} & 1.0223 \cdot 10^{-2} & -3.6015 \cdot 10^{-3} \\ -2.3632 & -4.0261 & 6.8863 \cdot 10^{-1} & -1.6833 \cdot 10^{-2} \\ -1.9518 & 1.9959 \cdot 10^{+1} & -1.3466 \cdot 10^{-1} & 5.0026 \cdot 10^{-2} \end{bmatrix} \cdot \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} \\
\qquad + \begin{bmatrix} 3.2821 \cdot 10^{-1} & 3.7527 \cdot 10^{-6} \\ 1.0846 & -3.0257 \cdot 10^{-6} \\ 6.3893 & 1.1816 \cdot 10^{-4} \\ -1.8008 \cdot 10^{+1} & 9.7588 \cdot 10^{-5} \end{bmatrix} \cdot \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \\
\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} -50.0 & 50.0 & -5.0 & 5.0 \\ 0 & 2.5 \cdot 10^{+5} & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k \end{bmatrix} + \begin{bmatrix} 0 & 2.5 \cdot 10^{-3} \\ -2.5 \cdot 10^{+5} & 0 \end{bmatrix} \cdot \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}
\end{cases}
$$

Sampled-Time Car Suspension Models (cont.)

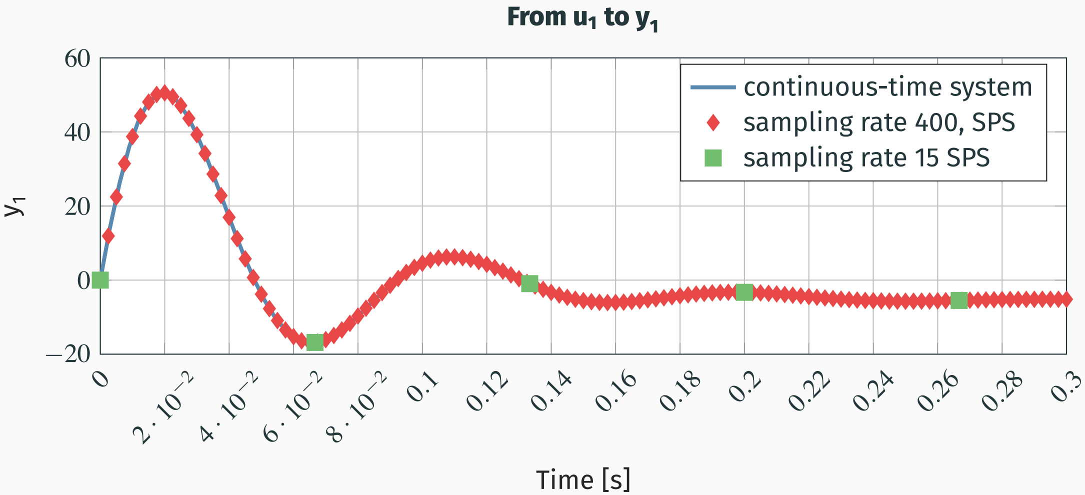


Figure 10: Step responses comparison: from u_1 to y_1

Sampled-Time Car Suspension Models (cont.)

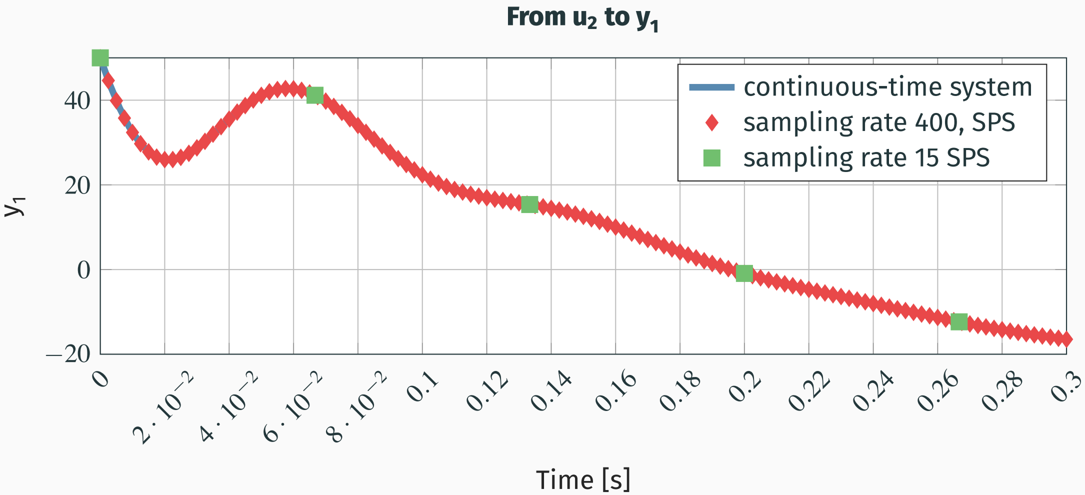


Figure 11: Step responses comparison: from u_2 to y_1

Sampled-Time Car Suspension Models (cont.)

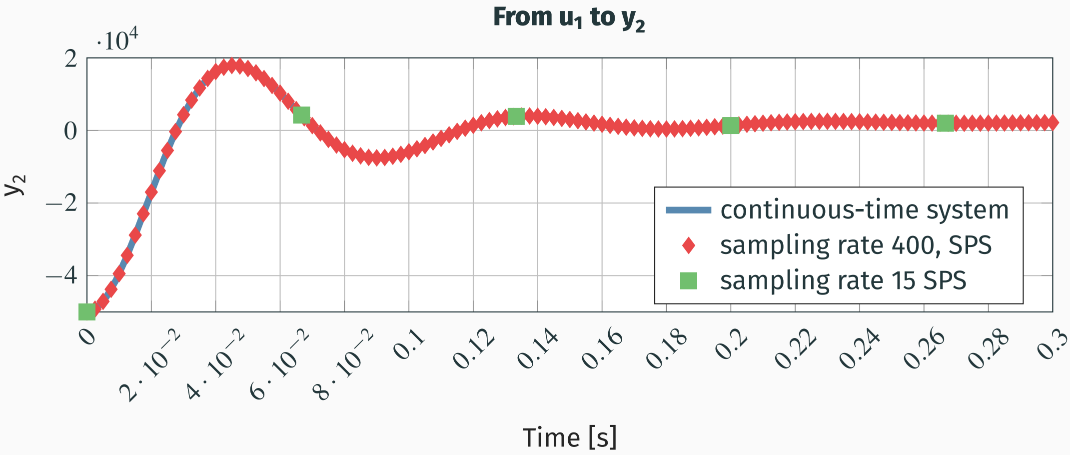


Figure 12: Step responses comparison: from u_1 to y_2

Sampled-Time Car Suspension Models (cont.)

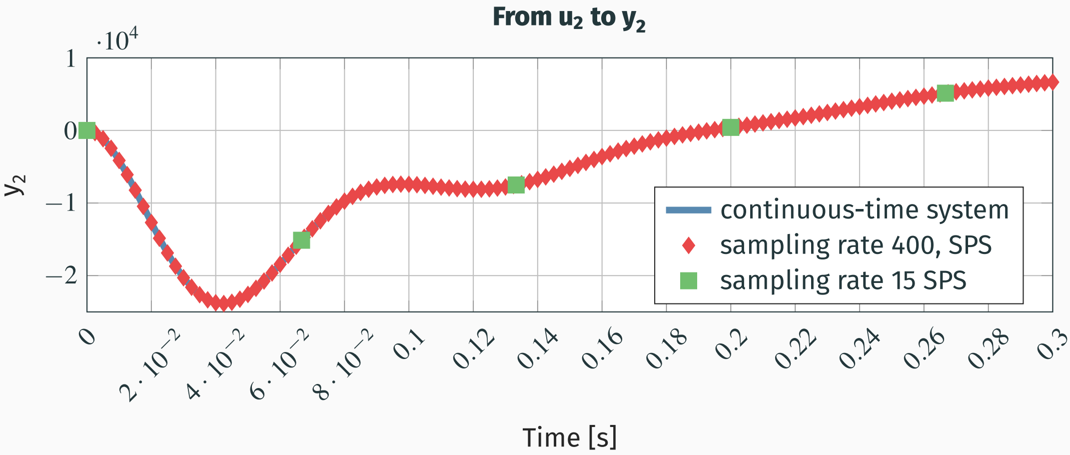


Figure 13: Step responses comparison: from u_2 to y_2

Sampled-Time Car Suspension Description (cont.)

Remarks

- by selecting **different sampling rates** we obtained **different representations** of the same continuous-time dynamic system
- **sampling** may **heavily distort the information**, giving a completely wrong discrete-time representation of the original continuous-time system: indeed the model obtained using *one sample per second* as the sampling rate is wrong!

Dynamic Systems Described by State Equations

More Definitions and Properties

More Definitions and Properties

- **Time-invariant Dynamic Systems**

$$\begin{cases} \dot{x}(t) = f(x(t), u(t), \cancel{t}) \\ y(t) = g(x(t), u(t), \cancel{t}) \end{cases} \Longrightarrow \begin{cases} \dot{x}(t) = f(x(t), u(t)) \\ y(t) = g(x(t), u(t)) \end{cases}$$

$$\begin{cases} x(k+1) = f(x(k), u(k), \cancel{k}) \\ y(k) = g(x(k), u(k), \cancel{k}) \end{cases} \Longrightarrow \begin{cases} x(k+1) = f(x(k), u(k)) \\ y(k) = g(x(k), u(k)) \end{cases}$$

- **Strictly Proper Dynamic Systems**

$$\begin{cases} \dot{x}(t) = f(x(t), u(t), t) \\ y(t) = g(x(t), \cancel{u(t)}, t) \end{cases} \Longrightarrow \begin{cases} \dot{x}(t) = f(x(t), u(t), t) \\ y(t) = g(x(t), t) \end{cases}$$

$$\begin{cases} x(k+1) = f(x(k), u(k), k) \\ y(k) = g(x(k), \cancel{u(k)}, k) \end{cases} \Longrightarrow \begin{cases} x(k+1) = f(x(k), u(k), k) \\ y(k) = g(x(k), k) \end{cases}$$

More Definitions and Properties (cont.)

- **Forced and Free Dynamic Systems**

$$\begin{cases} \dot{x}(t) = f(x(t), \cancel{u(t)}, t) \\ y(t) = g(x(t), \cancel{u(t)}, t) \end{cases} \Longrightarrow \begin{cases} \dot{x}(t) = f(x(t), t) \\ y(t) = g(x(t), t) \end{cases}$$

$$\begin{cases} x(k+1) = f(x(k), \cancel{u(k)}, k) \\ y(k) = g(x(k), \cancel{u(k)}, k) \end{cases} \Longrightarrow \begin{cases} x(k+1) = f(x(k), k) \\ y(k) = g(x(k), k) \end{cases}$$

It is worth noting that in case the input function $u(t), \forall\, t$ or input sequence $u(k), \forall\, k$ are **known beforehand**, the dynamic system can be re-written as a free one:

$$\begin{cases} \dot{x}(t) = f(x(t), u(t), t) = \widetilde{f}(x(t), t) \\ y(t) = g(x(t), u(t), t) = \widetilde{g}(x(t), t) \end{cases}$$

$$\begin{cases} x(k+1) = f(x(k), u(k), k) = \widetilde{f}(x(k), k) \\ y(k) = g(x(k), u(k), k) = \widetilde{g}(x(k), k) \end{cases}$$

More Definitions and Properties (cont.)

- **Free Movement**

$$\begin{aligned}&\dot{x}(t) = f(x(t), u(t), t)\\ &y(t) = g(x(t), u(t), t)\\ &\text{with:}\\ &\quad x(t_0) = x_0\,;\;\; u(t) = 0\,,\;\; \forall\, t\end{aligned} \quad\Longrightarrow\quad \begin{gathered}\{\,(x_l(t), t),\; t \in [t_0, t_1]\,\}\\ \text{free movement}\end{gathered}$$

$$\begin{aligned}&x(k+1) = f(x(k), u(k), k)\\ &y(k) = g(x(k), u(k), k)\\ &\text{with:}\\ &\quad x(k_0) = x_0\,;\;\; u(k) = 0\,,\;\; \forall\, k\end{aligned} \quad\Longrightarrow\quad \begin{gathered}\{\,(x_l(k), k),\; k \in [k_0, k_1]\,\}\\ \text{free movement}\end{gathered}$$

More Definitions and Properties (cont.)

- **Forced Movement**

$$\begin{aligned} &\dot{x}(t) = f(x(t), u(t), t) \\ &y(t) = g(x(t), u(t), t) \\ &\text{with:} \\ &\quad x(t_0) = 0 \end{aligned} \implies \begin{gathered} \{\, (x_f(t), t),\ t \in [t_0, t_1] \,\} \\ \text{forced movement} \end{gathered}$$

$$\begin{aligned} &x(k+1) = f(x(k), u(k), k) \\ &y(k) = g(x(k), u(k), k) \\ &\text{with:} \\ &\quad x(k_0) = 0 \end{aligned} \implies \begin{gathered} \{\, (x_f(k), k),\ k \in [k_0, k_1] \,\} \\ \text{forced movement} \end{gathered}$$

Dynamic Systems Described by State Equations

Discrete-time Systems

Discrete-time Systems

Consider:

$$\begin{aligned} x(k+1) &= f(x(k), u(k), k) \\ y(k) &= g(x(k), u(k), k) \end{aligned} \quad , \qquad k > k_0,\ x(k_0) = x_0$$

Clearly, by iterating the state equations:

$$\begin{aligned}
x(k_0) &= x_0 \\
x(k_0+1) &= f(x(k_0), u(k_0), k_0) \\
x(k_0+2) &= f(x(k_0+1), u(k_0+1), k_0+1) \\
&= f(f(x(k_0), u(k_0), k_0), u(k_0+1), k_0+1) \\
x(k_0+3) &= f(x(k_0+2), u(k_0+2), k_0+2) \\
&= f(f(f(x(k_0), u(k_0), k_0), u(k_0+1), k_0+1), u(k_0+2), k_0+2)
\end{aligned}$$

and so on. Hence, the **state transition function** has the form

$$x(k) = \varphi(k, k_0, x_0, \{u(k_0), \ldots, u(k-1)\})$$

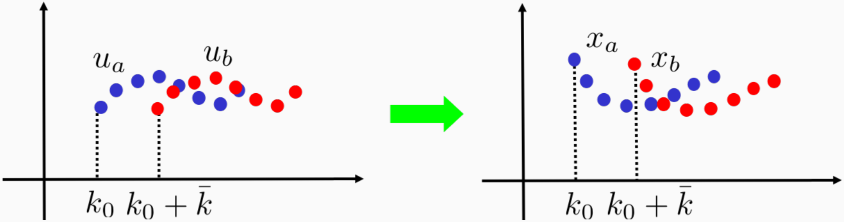
thus enhancing the **causality property**.

Time-invariant Discrete-time Systems

$$\begin{aligned} x(k+1) &= f(x(k), u(k)) \\ y(k) &= g(x(k), u(k)) \end{aligned} \quad , x(k_0) = x_0, \; u_a(k) = u(k), \; k \in \{k_0, \ldots, k_1\}$$

yields the state sequence $x_a(k), \; k \in \{k_0, \ldots, k_1\}$. Let's shift the initial time by $\bar{k}$ and the input sequence as well:

$$\begin{aligned} & x(k_0 + \bar{k}) = x_0 \\ & u_b(k) = u_a(k - \bar{k}), \\ & \quad k \in \{k_0 + \bar{k}, \ldots, k_1 + \bar{k}\} \end{aligned} \quad \Longrightarrow \quad \begin{aligned} & x_b(k) = x_a(k - \bar{k}), \\ & \quad k \in \{k_0 + \bar{k}, \ldots, k_1 + \bar{k}\} \end{aligned}$$



Conventionally, we set $k_0 = 0$.

Equilibrium Analysis: Equilibrium States and Outputs

- A state $\bar{x} \in \mathbb{R}^n$ is an **equilibrium state** if $\forall\, k_0$, $\exists\, \{\bar{u}(k) \in \mathbb{R}^m, k \geq k_0\}$ such that

$$\begin{array}{l} x(k_0) = \bar{x} \\ u(k) = \bar{u}(k),\, \forall\, k \geq k_0 \end{array} \implies x(k) = \bar{x},\, \forall\, k > k_0$$

- An output $\bar{y} \in \mathbb{R}^p$ is an **equilibrium output** if $\forall\, k_0$, $\exists\, \{\bar{u}(k) \in \mathbb{R}^m, k \geq k_0\}$ such that

$$\begin{array}{l} x(k_0) = \bar{x} \\ u(k) = \bar{u}(k),\, \forall\, k \geq k_0 \end{array} \implies y(k) = \bar{y},\, \forall\, k > k_0$$

In general:

- The input sequence $\{\bar{u}(k) \in \mathbb{R}^m, k \geq k_0\}$ depends on the initial time k_0
- The fact that the state is of equilibrium does **not** imply that the corresponding output coincides with an equilibrium output

Equilibrium Analysis in the Time-invariant Case

In the time-invariant case, **all equilibrium states** can be determined by imposing **constant** input sequences.

A state $\bar{x} \in \mathbb{R}^n$ is an equilibrium state if $\exists\, \bar{u} \in \mathbb{R}^m$ such that

$$\begin{array}{l} x(k_0) = \bar{x} \\ u(k) = \bar{u},\ \forall\, k \geq k_0 \end{array} \implies x(k) = \bar{x},\ \forall\, k > k_0$$

All equilibrium states $\bar{x} \in \mathbb{R}^n$ can thus be obtained by finding all solutions of the algebraic equation

$$\bar{x} = f(\bar{x}, \bar{u})\,, \quad \forall\, \bar{u} \in \mathbb{R}^m$$

The following sets are also introduced:

$$\begin{aligned} \bar{X}_{\bar{u}} &= \{\bar{x} \in \mathbb{R}^n :\ \bar{x} = f(\bar{x}, \bar{u})\} \\ \bar{X} &= \{\bar{x} \in \mathbb{R}^n :\ \exists\, \bar{u} \in \mathbb{R}^m \text{ such that } \bar{x} = f(\bar{x}, \bar{u})\} \end{aligned}$$

Dynamic Systems Described by State Equations

State Space Description: Criteria and Examples

But ... How to determine a state space description?

Recall:

State variables

Variables to be known at time $t = t_0$ in order to be able to determine the output $y(t),\ t \geq t_0$ from the knowledge of the input $u(t),\ t \geq t_0$:

$$x_i(t),\ i = 1, 2, \ldots, n \qquad \text{(state variables)}$$

State Space Descriptions (cont.)

A "physical" criterion

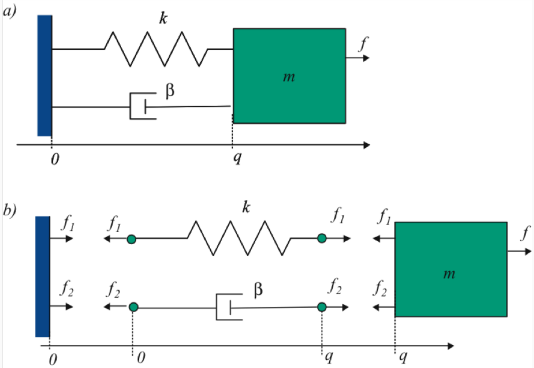
State variables can be defined as entities associated with storage of mass, energy, etc. ...

For example:

- **Passive electrical systems**: voltages on capacitors, currents on inductors
- **Translational mechanical systems**: linear displacements and velocities of each independent mass
- **Rotational mechanical systems**: angular displacements and velocities of each independent inertial rotating mass
- **Hydraulic systems**: pressure or level of fluids in tanks
- **Thermal systems**: temperatures
- ...

State Space Descriptions: Example 1 (continuous-time)

A mechanical system

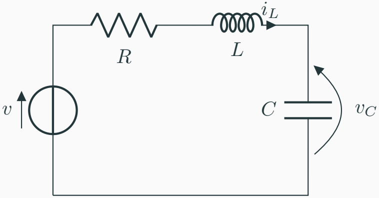


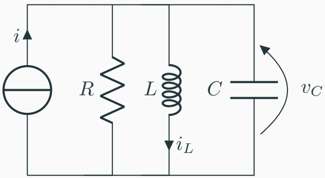
$$m\ddot{q} + \beta\dot{q} + kq = f$$

$$\begin{aligned} x_1 &:= q \\ x_2 &:= \dot{q} \end{aligned} \quad \Longrightarrow \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}; \quad \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \ddot{q} = -\dfrac{k}{m}x_1 - \dfrac{\beta}{m}x_2 + \dfrac{1}{m}f \end{cases}$$

State Space Descriptions: Example 2 (continuous-time)

Electrical systems





$$L\frac{di_L}{dt} = v - Ri_L - v_C$$

$$C\frac{dv_C}{dt} = i_L$$

$$\begin{cases} \dot{x}_1 = -\frac{R}{L}x_1 - \frac{1}{L}x_2 + \frac{1}{L}v \\ \dot{x}_2 = \frac{1}{C}x_1 \end{cases}$$

$$x_1 := i_L \; ; \;\; x_2 := v_C$$

$$C\frac{dv_C}{dt} = i - \frac{1}{R}v_C - i_L$$

$$L\frac{di_L}{dt} = v_C$$

$$\begin{cases} \dot{x}_1 = \frac{1}{L}x_2 \\ \dot{x}_2 = -\frac{1}{C}x_1 - \frac{1}{RC}x_2 + \frac{1}{C}iv \end{cases}$$

State Space Descriptions: Example 3 (discrete-time)

Student dynamics: 3-years undergraduate course

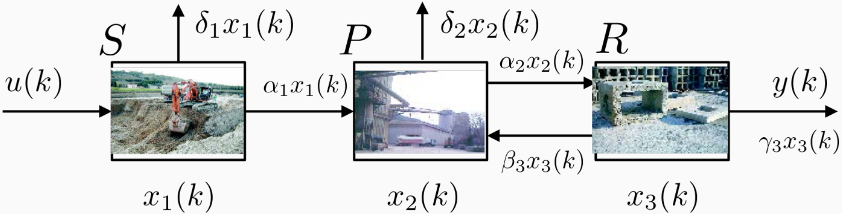
- percentages of students promoted, repeaters, and dropouts are roughly constant
- direct enrolment in 2nd and 3rd academic year is not allowed
- students cannot enrol for more than 3 years

$$\begin{cases} x_1(k+1) = \beta_1 x_1(k) + u(k) \\ x_2(k+1) = \alpha_1 x_1(k) + \beta_2 x_2(k) \\ x_3(k+1) = \alpha_2 x_2(k) + \beta_3 x_3(k) \\ y(k) = \alpha_3 x_3(k) \end{cases}$$

- $x_i(k)$: number of students enrolled in year i at year k, $i = 1, 2, 3$
- $u(k)$: number of freshmen at year k
- $y(k)$: number of graduates at year k
- α_i: promotion rate during year i, $\alpha_i \in [0, 1]$
- β_i: failure rate during year i, $\beta_i \in [0, 1]$
- γ_i: dropout rate during year i, $\gamma_i = 1 - \alpha_i - \beta_i \geq 0$

State Space Descriptions: Example 4 (discrete-time)

Supply chain



- S purchases the quantity $u(k)$ of raw material at each month k
- A fraction δ_1 of raw material is discarded, a fraction α_1 is shipped to producer P
- A fraction α_2 of product is sold by P to retailer R, a fraction δ_2 is discarded
- Retailer R returns a fraction β_3 of defective products every month, and sells a fraction γ_3 to customers

State Space Descriptions: Example 4 (discrete-time) (cont.)

$$
\begin{cases}
x_1(k+1) = (1-\alpha_1-\delta_1)x_1(k) + u(k) \\
x_2(k+1) = \alpha_1 x_1(k) + (1-\alpha_2-\delta_2)x_2(k) \\
\qquad\qquad\qquad +\beta_3 x_3(k) \\
x_3(k+1) = \alpha_2 x_2(k) + (1-\beta_3-\gamma_3)x_3(k) \\
y(k) = \gamma_3 x_3(k)
\end{cases}
$$

- k: month counter
- $x_1(k)$: raw material in S
- $x_2(k)$: products in P
- $x_3(k)$: products in R
- $y(k)$: products sold to customers

State Space Descriptions (cont.)

A ”mathematical” criterion

- **Continuous-time case.** An input-out differential equation model of the system is available:

$$\frac{\mathrm{d}^n y}{\mathrm{d}t^n} = \varphi\left(\frac{\mathrm{d}^{n-1} y}{\mathrm{d}t^{n-1}}, \ldots, \frac{\mathrm{d}y}{\mathrm{d}t}, y, u, t\right)$$

- **Discrete-time case.** An input-out difference equation model of the system is available:

$$y(k+n) = \varphi\left(y(k+n-1), y(k+n-2), \ldots, y(k), u(k), k\right)$$

Suitable state variables – without necessarily a physical meaning – are **defined** to represent ”mathematically” the differential equation or the difference equation models of the dynamic system

State Space Descriptions (cont.)

Continuous-time case:

$$\frac{\mathrm{d}^n y}{\mathrm{d}t^n} = \varphi\left(\frac{\mathrm{d}^{n-1} y}{\mathrm{d}t^{n-1}}, \dots, \frac{\mathrm{d}y}{\mathrm{d}t}, y, u, t\right)$$

Letting:

$$\begin{cases} x_1(t) := y(t) \\ x_2(t) := \dfrac{\mathrm{d}y}{\mathrm{d}t} \\ \vdots \\ x_n(t) := \dfrac{\mathrm{d}^n y}{\mathrm{d}t^n} \end{cases} \implies x := \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$$

one gets:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \vdots \\ \dot{x}_n = \varphi(x, u, t) \\ y = x_1 \end{cases}$$

State Space Descriptions (cont.)

Discrete-time case:

$$y(k+n) = \varphi\left(y(k+n-1), y(k+n-2), \ldots, y(k), u(k), k\right)$$

Letting:

$$\begin{cases} x_1(k) := y(k) \\ x_2(k) := y(k+1) \\ \vdots \\ x_n(k) := y(k+n-1) \end{cases} \implies \quad x := \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix}$$

one gets:

$$\begin{cases} x_1(k+1) = x_2(k) \\ x_2(k+1) = x_3(k) \\ \vdots \\ x_n(k+1) = \varphi(x, u, k) \\ y(k) = x_1(k) \end{cases}$$

Example (discrete-time):

$$w(k) - 3w(k-1) + 2w(k-2) - w(k-3) = 6u(k)$$

Letting:

$$\begin{cases} x_1(k) := w(k-3) \\ x_2(k) := w(k-2) \\ x_3(k) := w(k-1) \end{cases} \implies x := \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

one gets:

$$\begin{cases} x_1(k+1) = x_2(k) \\ x_2(k+1) = x_3(k) \\ x_3(k+1) = 3x_3(k) - 2x_2(k) + x_1(k) + 6u(k) \\ y(k) = 3x_3(k) - 2x_2(k) + x_1(k) + 6u(k) \end{cases}$$

State Space Descriptions (cont.)

The state space description is not unique

- The fact that physical and non-physical approaches can be followed to describe the **same dynamic system** in state-space form clearly reveals the **non-uniqueness** of this representation
- Later on some more details will be given concerning **equivalent** state space descriptions

State Space Descriptions (cont.)

Matlab live script

Given a state-space description for a dynamical system, how to implement it in Matlab/Simulink? How to tune the model, run it, and retrieve the resulting state and output movements?

A **Matlab live script** is available, illustrating how to implement a state space description for a dynamical system.
Steps to retrieve the live script:

- Download as a ZIP archive the whole contents of the folder named "**L1_StateSpaceDescriptionExamples,**" available in the "**Class Materials**" file area of the MS Teams course team. and uncompress it in a preferred folder.
- Add the chosen folder and subfolders to the Matlab path.
- Open the live script using the Matlab command:

```
open('StateSpaceDescriptionExamples.mlx');
```

Linear Dynamic Systems

Linear Dynamic Systems

Consider the discrete-time dynamic system state-space representation:

$$\begin{cases} x(k+1) = f(x(k), u(k), k) \\ y(k) = g(x(k), u(k), k) \end{cases}$$

This state-space equation describes a **linear system** if and only if the functions $f(\cdot)$ and $g(\cdot)$ are **linear with respect to their state and input vector arguments**:

$$\forall\, \alpha_1, \alpha_2 \in \mathbb{R},\ \forall\, x_1, x_2 \in \mathbb{R}^n,\ \forall\, u_1, u_2 \in \mathbb{R}^m\ :$$

$$\begin{aligned} f(\alpha_1 x_1 + \alpha_2 x_2, \alpha_1 u_1 + \alpha_2 u_2, k) &= \alpha_1 f(x_1, u_1, k) + \alpha_2 f(x_2, u_2, k) \\ g(\alpha_1 x_1 + \alpha_2 x_2, \alpha_1 u_1 + \alpha_2 u_2, k) &= \alpha_1 g(x_1, u_1, k) + \alpha_2 g(x_2, u_2, k) \end{aligned}$$

Linear Dynamic Systems: Matrix Form

Consider the state-space representation:

$$\begin{cases} x(k+1) = f(x(k), u(k), k) \\ y(k) = g(x(k), u(k), k) \end{cases}$$

and suppose that the linearity assumption holds. Then:

$$\begin{cases} f_1(x,u,k) = a_{11}(k)x_1 + \cdots + a_{1n}(k)x_n + b_{11}(k)u_1 + \cdots + b_{1m}(k)u_m \\ \vdots \\ f_n(x,u,k) = a_{n1}(k)x_1 + \cdots + a_{nn}(k)x_n + b_{n1}(k)u_1 + \cdots + b_{nm}(k)u_m \\ \\ y_1 = c_{11}(k)x_1 + \cdots + c_{1n}(k)x_n + d_{11}(k)u_1 + \cdots + d_{1m}(k)u_m \\ \vdots \\ y_p = c_{p1}(k)x_1 + \cdots + c_{pn}(k)x_n + d_{p1}(k)u_1 + \cdots + d_{pm}(k)u_m \end{cases}$$

where $a_{ij}(k), b_{ij}(k), c_{ij}(k), d_{ij}(k)$ are generic functions of the discrete-time index k.

Linear Dynamic Systems: Matrix Form (cont.)

Letting:

$$A(k) := \begin{bmatrix} a_{11}(k) & \cdots & a_{1n}(k) \\ \vdots & \ddots & \vdots \\ a_{n1}(k) & \cdots & a_{nn}(k) \end{bmatrix} ; \quad B(k) := \begin{bmatrix} b_{11}(k) & \cdots & b_{1m}(k) \\ \vdots & \vdots & \vdots \\ b_{n1}(k) & \cdots & b_{nm}(k) \end{bmatrix}$$

$$C(k) := \begin{bmatrix} c_{11}(k) & \cdots & c_{1n}(k) \\ \vdots & \ddots & \vdots \\ c_{p1}(k) & \cdots & c_{pn}(k) \end{bmatrix} ; \quad D(k) := \begin{bmatrix} d_{11}(k) & \cdots & d_{1m}(k) \\ \vdots & \vdots & \vdots \\ d_{p1}(k) & \cdots & d_{pm}(k) \end{bmatrix}$$

$$x(k) := \begin{bmatrix} x_1(k) & \cdots & x_n(k) \end{bmatrix}^T ; u(k) := \begin{bmatrix} u_1(k) & \cdots & u_m(k) \end{bmatrix}^T ; y(k) := \begin{bmatrix} y_1(k) & \cdots & y_p(k) \end{bmatrix}^T$$

One gets:

$$\begin{cases} x(k+1) = A(k)x(k) + B(k)u(k) \\ y(k) = C(k)x(k) + D(k)u(k) \end{cases}$$

Linear Dynamic Systems

Time-Invariant Linear Dynamic Systems

Time-Invariant Linear Dynamic Systems

In the **time-invariant** scenario, the matrices $A(k), B(k), C(k), D(k)$ do not depend on the time-index k, that is are **constant** matrices A, B, C, D:

$$A := \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \; ; \quad B := \begin{bmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & \vdots & \vdots \\ b_{n1} & \cdots & b_{nm} \end{bmatrix}$$

$$C := \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{p1} & \cdots & c_{pn} \end{bmatrix} \; ; \quad D := \begin{bmatrix} d_{11} & \cdots & d_{1m} \\ \vdots & \vdots & \vdots \\ d_{p1} & \cdots & d_{pm} \end{bmatrix}$$

and thus:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ \\ y(k) = Cx(k) + Du(k) \end{cases}$$

Time-Invariant Linear Dynamic Systems: Equilibrium States

Consider a linear time-invariant dynamic system:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases}$$

and consider a constant input sequence $u(k) = \bar{u},\ k \geq 0$. Hence, one has to solve the following equation for x:

$$x = Ax + B\bar{u} \implies (I - A)x = B\bar{u}$$

The following two cases have to be considered:

- $\det(I - A) \neq 0$
- $\det(I - A) = 0$

Time-Invariant Linear Dynamic Systems: Equilibrium States

- $\det(I - A) \neq 0$. In this case, one gets:

$$\bar{x} = (I - A)^{-1} B\bar{u} \quad \Longrightarrow \quad \bar{x} \quad \text{is} \quad \textbf{unique} \ \forall \, \bar{u} \in \mathbb{R}^m$$

Accordingly, the equilibrium output is given by:

$$\bar{y} = C\bar{x} + D\bar{u} = \left[C(I - A)^{-1} B + D \right] \bar{u}$$

Matrix $\left[C(I - A)^{-1} B + D \right]$ is defined as **static gain**.

- $\det(I - A) = 0$. In this case, two different situations may occur:
  - $\exists\, \infty$ equilibrium states $\bar{x}$, $\exists\, \infty$ equilibrium outputs $\bar{y}$
  - $\nexists$ equilibrium states $\bar{x}$, $\nexists$ equilibrium outputs $\bar{y}$

Time-Invariant Linear Dynamic Systems: Equilibrium States (cont.)

Matlab live script

How can we determine the equilibrium states for a discrete-time dynamical LTI system in Matlab?

A **Matlab live script** is available, illustrating how to cope with all the possible cases (there is either a single equilibrium state, or there are infinitely many, or none at all).

Steps to retrieve the live script:

- Download as a ZIP archive the whole contents of the folder named "**L1_EqulibriumState_LTI_Systems**," available in the "**Class Materials**" file area of the MS Teams course team, and uncompress it in a preferred folder.
- Add the chosen folder and subfolders to the Matlab path.
- Open the live script using the Matlab command:

```
open('equilibriumStatesLTIsys.mlx');
```

Linear Dynamic Systems

Time-Invariant Linear Dynamic Systems: Equivalent State-Space Representations

Equivalent State-Space Representations: LTI

Consider the discrete-time linear time-invariant (LTI) dynamic system state-space representation:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases}$$

Let $\hat{x} := T^{-1}x$, where $T \in \mathbb{R}^{n \times n}$ is a generic non-singular $n \times n$ matrix ($\det(T) \neq 0$).
Then, the equivalent state-space description is given by:

$$\begin{cases} \hat{x}(k+1) = T^{-1}x(k+1) = T^{-1}AT\hat{x}(k) + T^{-1}Bu(k) = \hat{A}\hat{x}(k) + \hat{B}u(k) \\ y(k) = CT\hat{x}(k) + Du(k) = \hat{C}\hat{x}(k) + Du(k) \end{cases}$$

Hence:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases} \iff \begin{cases} \hat{x}(k+1) = \hat{A}\hat{x}(k) + \hat{B}u(k) \\ y(k) = \hat{C}\hat{x}(k) + Du(k) \end{cases}$$

Equivalent State-Space Representations: LTI (cont.)

Matlab live script

What does it mean in practice equivalent state-space representation?
How do we apply a state variable linear transformation in Matlab?

A **Matlab live script** is available, illustrating how to deal with state transformation for LTI systems and what it means to have an equivalent state-space representation for a given LTI system regarding state and output movements.
Steps to retrieve the live script:

- Download as a ZIP archive the whole contents of the folder named "**L1_LTI_EquivStateSpaceForm**," in the "**Class Materials**" file area of the MS Teams course team and uncompress it in a preferred folder.
- Add the chosen folder and subfolders to the Matlab path.
- Open the live script using the Matlab command:

```
open('LTI_Systems_EquivalentStateSpaceRepresentation.mlx');
```

267MI –Fall 2023

Lecture 1

Generalities: Systems and Models

END