CS1 Week 3
Empty

System properties, Linearization

Recap Last week

Quiz 1

What are block diagrams?

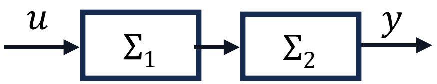
A. schematic for designing electrical circuits

B. simple graphical representation of a system’s inputs, outputs, and functions

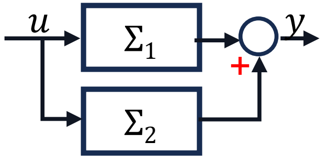
C. secret diagrams for Minecraft building blocks

Quiz 2

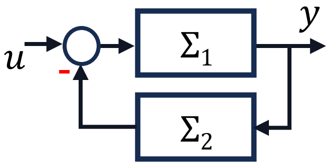
Which is the block diagram for feedback interconnection?



A



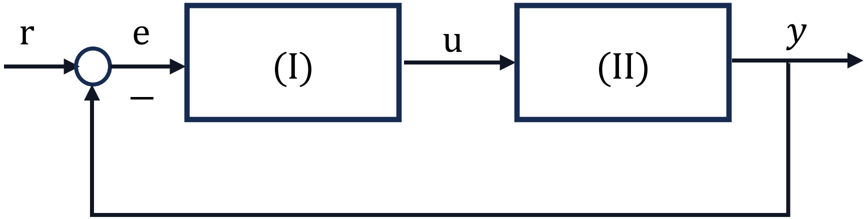
B



C

Quiz 3

What are (I) and (II) labeled as in an closed-loop system by convetion?



A. (I): Plant P; (II): Controller C

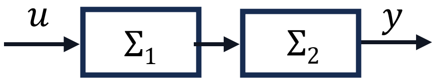
B. (I): Controller C; (II): Plant P

C. (I): System 1; (II): System 2

D. (I): Monitor; (II): Operator

Quiz 4

What is the transfer function $\Sigma_{u \to y}$?



A. Σ_1

B. Σ_2

C. $\Sigma_1 \Sigma_2$

D. $\Sigma_2 \Sigma_1$

Course Schedule

	Subject	Week
Modeling	Introduction, Control Architectures, Motivation	1
	Modeling, Model examples	2
	System properties, Linearization	3
Analysis	Analysis: Time response, Stability	4
	Transfer functions 1: Definition and properties	5
	Transfer functions 2: Poles and Zeros	6
	Proportional feedback control, Root Locus	7
	Time-Domain specifications, PID control, Computer implementation	8
	Frequency response, Bode plots	9
	The Nyquist condition, Time delays	10
Synthesis	Frequency-domain Specifications, Dynamic Compensation, Loop Shaping	11
	Time delays, Successive loop closure, Nonlinearities	12
	Describing functions	13
	Intro to Uncertainty and Robustness	14

Today

1. System Classification
2. Linearization

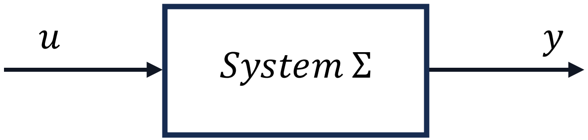
1. System Classification

Why System Classification?

- *System classification* = organizing systems into categories based on input–output behavior
- It provides a *framework* to describe and compare different types of systems
- Helps to predict how a system will respond under various conditions
- Clarifies if the system is *physically realizable* (e.g. causal vs. non-causal)

Input/Output Model

In CS1 we usually deal with SISO LTI (Single Input Single Output, Linear Time-Invariant systems)



A system Σ can be classified in the following ways:

1. **Linear** vs **Non-linear**
2. **Causal** vs **Non-causal**
3. **Static (memoryless)** vs **Dynamic**
4. **Time-invariant** vs **Time-variant**

Linearity

For a system to be **linear** two conditions have to be fullfilled:

- Additivity: $\Sigma(\mathrm{u}_1 + \mathrm{u}_2) = \Sigma\mathrm{u}_1 + \Sigma\mathrm{u}_2$
- Homogeneity: $\Sigma(\mathrm{ku}) = \mathrm{k}\Sigma\mathrm{u} \quad k \in \mathbb{R}$

Both can be summarized to:

$$\Sigma(\alpha u_1 + \beta u_2) = \alpha\Sigma u_1 + \beta\Sigma u_2 = \alpha y_1 + \beta y_2 \qquad \alpha, \beta \in \mathbb{R}$$

This implies the idea of **superposition**. That means that, when a system is linear, we can:

1. Break down "complicated" input signal into simpler components $u = u_1 + u_2$
2. Compute the output for each simple input seperately $y_1 = \Sigma u_1\, ;\, y_2 = \Sigma u_2$
3. Sum all the outputs to obtain the reponse of the complicated input $y = y_1 + y_2$

Remember: **Differentiation** and **Integration** are **linear operations!**

Linear vs Non-linear: Examples

Examples of **linear** systems:

- Integrator: $y(t) = \int_{-\infty}^{t} u(\tau)\, d\tau$
- Derivative: $y(t) = \dot{u}(t)$
- Time shifts: $y(t) = u(t - \tau)$
- Time scaling: $y(t) = u(t^2)$

Examples of **non-linear** systems:

- $y(t) = u(t)^2$
- $y(t) = u(t) + a$

Quiz 1

Classify the system as linear or non-linear

$$y(t) = t^2 u(t-1)$$

A. linear

B. non-linear

Quiz 2

Classify the system as linear or non-linear

$$y(t) = \sin(\mathrm{u}(t))$$

A. linear

B. non-linear

Causality

A system is said to be **causal**, *iff* the future input doesn't affect the present output.

- output depends on the past and current inputs
- The time domain for a causal system is $(-\infty, t]$
- All practically realizable systems are causal. Otherwise you could predict the future 🔮

Examples of **causal** systems:
Output doesn't depend on future inputs
- $y(t) = u(t)$
- $y(t) = u(t - \tau),\ \forall \tau > 0$

Examples of **non-causal** systems:
- $y(t) = u(t - \tau),\ \forall \tau < 0$

Static vs Dynamic

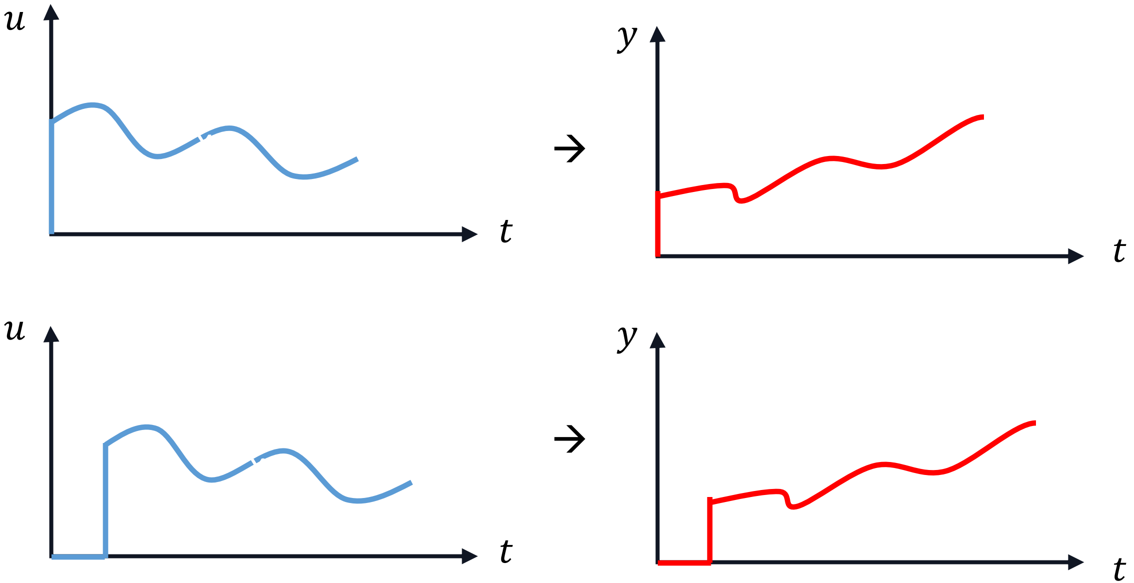
- A system is **static (memoryless)** if its output at any time depends only on the input at that same time (current input)
- If the output depends on past or future values of the input, the system is **dynamic**
- Systems described by ODEs are always dynamic!
- Static systems are usually desribed by algrebraic equations

Examples of **static** systems:
- $y(t) = 2^{-t+1}u(t)$

Examples of **dynamic** systems:
- Integrator: $y(t) = \int_{-\infty}^{t} u(\tau)\,d\tau$
- Derivative: $y(t) = \dot{u}(t)$
- Future/Past Inputs: $y(t) = u(t-\tau), \forall, \tau \neq 0$

Time invariant vs. Time-varying

A *time invariant* system will always have the same output to a certain input, independent of when the input is applied. Formally this means that we can shift the input in time and the output will also be shifted.



Example of **time invariant** system:

- $y(t) = u(t-1)u(t+2)$

Example of **time-varying** system:

- $y(t) = \cos(t)u(t)$

σ_τ is the time-shifting operator. The parameter τ indicates the amount of shift.

Time invariant vs. Time-varying formally

A system is time-invariant if:

$$\sigma_\tau y = \sigma_\tau \Sigma u = \Sigma \sigma_\tau u$$

or, equivalently if:

$$y(t-\tau) = (\Sigma \tilde{u})(t)$$

where $\tilde{u}(t) = u(t-\tau)$

Example: Is the following system **time invariant**?

$y(t) = u(t) + 3u(t-2)$

1. Input shift: $u_\tau(t) = u(t-\tau)$
2. Output with shifted input: $y_\tau(t) = u_\tau(t) + 3u_\tau(t-2) = u(t-\tau) + 3u(t-2-\tau)$
3. Shifted output: $y(t-\tau) = u(t-\tau) + 3u\big((t-\tau)-2\big) = u(t-\tau) + 3u(t-2-\tau)$

Output with shifted input is the same as shifted output → system is **time invariant**

Exam Question W18

$$y(t) = n^3 t^3 u(t), n \in R$$

Mark all correct statements.

A. Causal

B. Linear

C. Memoryless/Static

D. Time-invariant

2. Linearization

State-Space Model

- Recall: standard form, this will be called state-space model

$$\dot{x}(t) = f\big(x(t), u(t)\big)$$
$$y(t) = f\big(x(t), u(t)\big)$$

One characteristic of **LTI systems**, is that we can write the state space model in the form:

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

where A,B,C, and D are *constant matrices/vectors*.

- Reminder: In a minimal realization, the system order equals the dimension of the state vector $x(t)$

State-Space Model

- For LTI systems the state-space model can be written as:

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

in minimal realization

- Where A,B,C,D are generally matrices
- The order of the system is given by the dimension of the state vector $x(t)$
- Higher order ODEs can be converted in the following way (LinAlg):

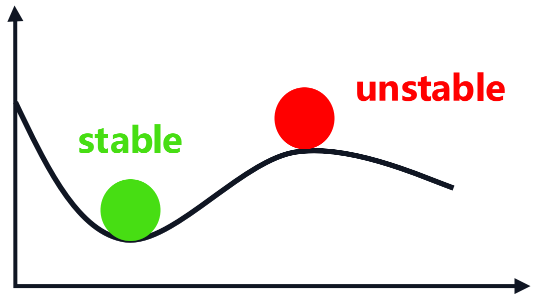
$$\theta^{(n)} = f\left(\theta^{(n-1)}, \theta^{(n-2)}, \dots, \theta\right) \rightarrow x(t) = \begin{pmatrix} \theta \\ \dots \\ \theta^{(n-1)} \end{pmatrix} = \begin{pmatrix} x_1 \\ \dots \\ x_n \end{pmatrix} \rightarrow f(x_1, x_2, \dots, xn)$$

Equilibrium Points

- Given a non-linear system:

$$\begin{aligned} \dot{x}(t) &= f\big(x(t), u(t)\big) \\ y(t) &= f\big(x(t), u(t)\big) \end{aligned}$$

- To apply the LTI state-space notation, we need to *linearize* the system around equilibrium points
- A point (x_e, ue) is an equilibrium point if $\dot{x} = f(x_e, u_e) = 0$



Linearization Procedure

1. Find equilibrium points (i.e. states where the system doesn't change)

   Solve $\dot{x}(t) = \begin{pmatrix} x_1 \\ x_2 \\ \vdots\ x_n \end{pmatrix} = 0$ to find equilibrium points (x_e, ue)

2. Use definitions of A, B, C and D. Plug in $x = x_e$ and $u = u_e$

$$A = \left.\frac{\partial f(x,u)}{\partial x}\right|_{(x_e,u_e)} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}\right|_{(x_e,u_e)} \in \mathbb{R}^{n\times n}$$

$$C = \left.\frac{\partial g(x,u)}{\partial x}\right|_{(x_e,u_e)} = \left.\begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_n}{\partial x_1} & \cdots & \frac{\partial g_n}{\partial x_n} \end{bmatrix}\right|_{(x_e,u_e)} \in \mathbb{R}^{p\times n}$$

$$B = \left.\frac{\partial f(x,u)}{\partial u}\right|_{(x_e,u_e)} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_m} \end{bmatrix}\right|_{(x_e,u_e)} \in \mathbb{R}^{n\times m}$$

$$D = \left.\frac{\partial g(x,u)}{\partial u}\right|_{(x_e,u_e)} = \left.\begin{bmatrix} \frac{\partial g_1}{\partial u_1} & \cdots & \frac{\partial g_1}{\partial u_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_n}{\partial u_1} & \cdots & \frac{\partial g_n}{\partial u_m} \end{bmatrix}\right|_{(x_e,u_e)} \in \mathbb{R}^{p\times m}$$

Linearization Example

Find equilibrium points: Solve $\dot{x}(t) = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = 0$

Aufgabe: Gegeben ist das folgende nichtlineare System mit Zustandsvektor $x(t)$ und Eingang $u(t)$.
Die Systemgleichungen lauten:

$$\dot{x}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} u(t) + 5x_1(t) \cdot x_2(t) + 1 \\ u(t) \cdot x_1(t) \end{bmatrix} .$$

Der Systemausgang ist gegeben durch

$$y(t) = x_1(t) \cdot u(t) + 2x_2(t) .$$

F14 **(1.5 Punkte)** Berechnen Sie den reellen Gleichgewichtspunkt u_e, $x_{1,e}$, $x_{2,e}$, y_e für einen Eingang $\boxed{u_e \neq 0, \text{ für den } y_e = 2}$ gilt.

$u_e =$

$x_{1,e} =$

$x_{2,e} =$

Control Systems 1 Exam Spring 2021

ETH zürich IDSC

Linearization Example

Bringen Sie das System in die Zustandsraumdarstellung und nutzen Sie die Variablen $x_e = [x_{1,e}, x_{2,e}]^\top$ und u_e um die nötigen Matrizen **symbolisch** auszuwerten.

F15 **(1 Punkt)**

$A =$

F16 **(0.5 Punkte)**

$b =$

F17 **(0.5 Punkte)**

$c =$ $\quad d =$

$$A = \left.\frac{\partial f(x,u)}{\partial x}\right|_{(x_e,u_e)} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}\right|_{(x_e,u_e)} \in \mathbb{R}^{n\times n} \qquad C = \left.\frac{\partial g(x,u)}{\partial x}\right|_{(x_e,u_e)} = \left.\begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_n}{\partial x_1} & \cdots & \frac{\partial g_n}{\partial x_n} \end{bmatrix}\right|_{(x_e,u_e)} \in \mathbb{R}^{p\times n}$$

$$B = \left.\frac{\partial f(x,u)}{\partial u}\right|_{(x_e,u_e)} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_m} \end{bmatrix}\right|_{(x_e,u_e)} \in \mathbb{R}^{n\times m} \qquad D = \left.\frac{\partial g(x,u)}{\partial u}\right|_{(x_e,u_e)} = \left.\begin{bmatrix} \frac{\partial g_1}{\partial u_1} & \cdots & \frac{\partial g_1}{\partial u_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_n}{\partial u_1} & \cdots & \frac{\partial g_n}{\partial u_m} \end{bmatrix}\right|_{(x_e,u_e)} \in \mathbb{R}^{p\times m}$$

$$\dot{x}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} u(t) + 5x_1(t)\cdot x_2(t) + 1 \\ u(t)\cdot x_1(t) \end{bmatrix}$$

$$y(t) = x_1(t)\cdot u(t) + 2x_2(t)$$

Control Systems 1 Exam Spring 2021

Questions?

Feedback?

Too fast? Too slow? Less theory, more exercises?

I would appreciate your feedback. Please let me know.

https://docs.google.com/forms/d/e/1FAIpQLSdHl0kjWo63aNzDkAV0cnmQadCAj5L0D7v7aSh0BK7BBdEgpA/viewform?usp=header