2. Recital 04.10.24

Recap

Modeling:

We want to learn how to mathematically represent dynamic systems. All systems that we want to describe, can be represented by differential equations. We will generally use this standard form to describe dynamic systems.

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

It is a system of 1^{st} order ODEs (as seen in LinAlg II). This is called the state-space form, since we are observing how the state vector x changes.

The vector $x = \begin{pmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{pmatrix}$ contains all state variables of the system. The states describe how a system changes internally over time. It can be thought of as a memory, containing a summary of how the system behaved in the past. Given the internal states and the current input, we can uniquley predict any future behavior.

Block Diagrams:

Block diagrams are an effective way to visually show how different systems are connected. It is the standard way to illustrate the interconnection of different systems and control architectures.

The simplest example is:

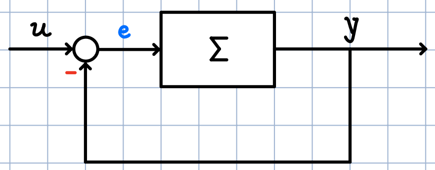
$$\xrightarrow{u} \boxed{\Sigma} \xrightarrow{y}$$

Here Σ maps an input u to an output y. We can write that:

$$\boxed{y = \Sigma u}$$

We can also look at a simple feedback loop with Σ being a scalar gain, meaning

$y = \Sigma e$ with $\Sigma \in \mathbb{R}$



$$\boxed{y = \frac{\Sigma}{1+\Sigma} u}$$

If we now choose Σ to be -1, the output y will go to infinity, independent of u.

That means that we have to be careful when using feedback, else our system might become unstable (i.e., $y \to \infty$).

> Remember course objectives:
>
> We're here → **Modeling**: represent real world systems with mathematical equations.
>
> **Analysis**: understand how a given system behaves; how the input affects the output, how feedback influences the system.
>
> **Synthesis**: Change the system, so that it behaves in a desirable way.

System Classification

Now we know how to describe physical systems with equations. With that we can also classify them in different ways. This classification is important for us, since we will only consider one specific type of system in this class. (So called linear time invariant or LTI systems)

But generally we classify system in these categories:

- → Linear vs. Nonlinear
- → Causal vs. Non-causal
- → Static (memoryless) vs. Dynamic
- → Time invariant vs. Time-varying

Linearity:

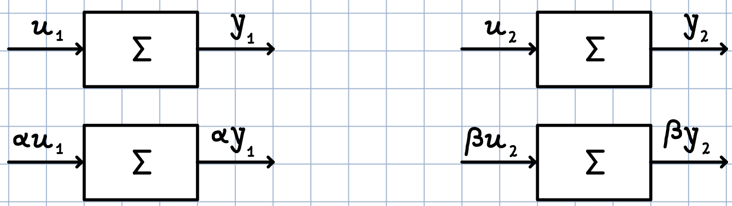
For a system to be linear two conditions have to be fullfilled.

- → Additivity: $\Sigma(u_1+u_2) = \Sigma u_1 + \Sigma u_2$
- → Homogeneity: $\Sigma k u = k \Sigma u \quad , \quad k \in \mathbb{R}$

Differentiation and integration are linear operations!

We can summarize both to:

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

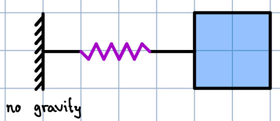


This implies the idea of superposition. That means that, when a system is linear, we can:

- → Break down "complicated" input signals into simpler components
  $u = u_1 + u_2$
- → Compute the output for each simple input seperatley
  $y_1 = \Sigma u_1 \ ; \ y_2 = \Sigma u_2$
- → Sum all of the simple outputs together to obtain the response to the complicated input
  $y = y_1 + y_2$

Examples:

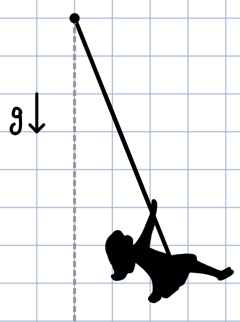
Linear:



$\Longleftrightarrow$

$$\dot{x}_1(t) = x_2(t)$$
$$\dot{x}_2(t) = -\frac{k}{m} x_1(t) + \frac{1}{m} u(t)$$
$$y(t) = x_2(t)$$

Nonlinear:



$\Longleftrightarrow$

$$\dot{x}_1(t) = x_2(t)$$
$$\dot{x}_2(t) = \frac{1}{ml^2}\left[-lmg \sin x_1(t) - c\, x_2(t) + l \cos x_1(t)\, u(t)\right]$$
$$y(t) = x_1(t)$$

Linear vs. Nonlinear:

		Linear	Nonlinear
1.	$y(t) = t^2 u(t-1)$		
2.	$y(t) = \sin(u(t))$		

Causality:

if and only if

A system is said to be causal, iff the future input does not affect the present output. All practically realizable systems are causal. Otherwise you could predict the future.

	Causal	Non-causal
$y(t) = u(t-a),\ \forall a < 0$		
$y(t) = u(t-\tau),\ \forall \tau > 0$		
$y(t) = \cos(3t+1)u(t-1)$		

Static vs. Dynamic

An input-output system Σ is static or memoryless if for all t, $y(t^*)$ is only a function of $u(t^*)$.

In other words: the present output depends only on the present input and not on past or future inputs.

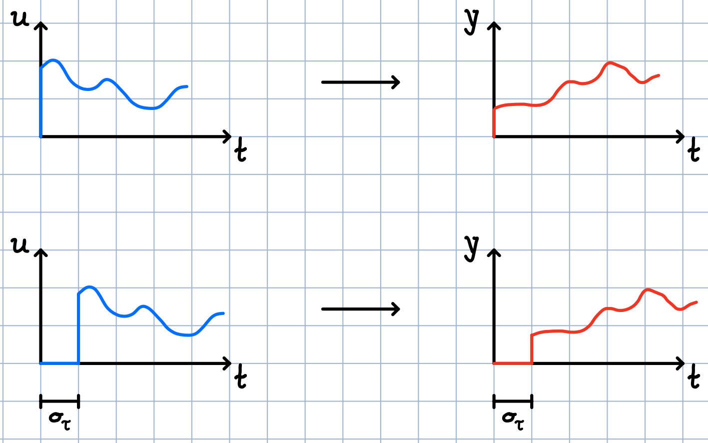
Systems described by ODEs are always dynamic. Static systems are usually described by algebraic equations. You can think of systems from Mech I as static, and those from Mech III as dynamic.

	Static	Dynamic
$y(t) = 2^{-(t+1)}u(t)$		
$y(t) = \int_{-\infty}^{t} u(\tau)\,d\tau$		
$y(t) = \dot{u}(t)$		

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.



	Variant	Invariant
$y(t) = u(t-1)\,u(t+2)$		
$y(t) = \cos(t)\,u(t)$		

But what kind of systems do we care about?

- → Linear
- → Time invariant
- → Causal
- → Single input, Single output

LTI SISO Systems

very restrictive class of systems

many systems can be well approximated by LTI SISO systems

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

$$\dot{x}(t) = A x(t) + B u(t)$$

$$y(t) = C x(t) + D u(t)$$

where A, B, C, and D are constant matrices or vectors. One example of such system is the mass-spring system from before. The state space form can be re-written in matrix form.

$$\dot{x}_1(t) = x_2(t)$$
$$\dot{x}_2(t) = -\frac{k}{m} x_1(t) + \frac{1}{m} u(t) \quad \Longleftrightarrow \quad \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{m} \end{pmatrix} u(t)$$
$$y(t) = x_2(t) \qquad\qquad y(t) = \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$$

In this example A, B, C, and D are defined as follows:

$$A = \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ \frac{1}{m} \end{pmatrix}, \quad C = \begin{pmatrix} 0 & 1 \end{pmatrix}, \quad D = 0$$

The vector $x = \begin{pmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{pmatrix}$ is the state of the system. It contains the information needed, together with the current input, to uniquely predict future outputs.

The dimension n of the state vector $x \in \mathbb{R}^n$ is equal to the dimension of the system. The choice of a systems state is not unique. Every different choice of states is called realization (there are infinitley many realizations). If we choose a realization with the smallest possible state vector, it is called minimal realization. A static system has a zero-dimensional state vector.

Let's look at the other example from before, the swing:

$$\dot{x}_1(t) = x_2(t)$$
$$\dot{x}_2(t) = \frac{1}{ml^2}\left[-lmg \sin x_1(t) - c\, x_2(t) + l \cos x_1(t)\, u(t)\right]$$
$$y(t) = x_1(t)$$

How do we write this in the matrix form, i.e. what are A, B, C, and D ?

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} ? \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} ? \end{pmatrix} u(t)$$
$$y(t) = \begin{pmatrix} ? \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$$

The system is nonlinear. Therefore we can't directley write it in the form of a LTI system. But we can modify the system and approximate it as a LTI system. (We will do this a lot since most systems are not LTI.

Linearization:

When linearizing, we take some finite-dimensional, time-invariant, causal nonlinear system and approximate it as a LTI system. This approximation works very well and let's us use control systems designed for LTI systems even on nonlinear systems.

The main idea is to pick an equilibrium point of the system and then make a linear approximation arround that equilibrium point. For a system modeled with ODEs $\dot{x}(t) = f(x(t), u(t))$, we can define an equilibrium point (x_e, u_e) to be at

$$f(x_e, u_e) = 0$$

Let's find some equilibrium points of the swing example. We will look at equilibria where $u_e = 0$. Intuitevley, where would those be?

Let's take a closer look:

$u_e = 0$, $f(x_e, 0) = 0$

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = \frac{1}{ml^2}\left[-lmg \sin x_1(t) - c\, x_2(t) + l \cos x_1(t)\, u(t)\right] \\ y(t) = x_1(t) \end{cases}$$

$\dot{x}_2(t) = \qquad = 0 \implies x_1 = \qquad$ or $x_1 =$

So for $u_e = 0$ there are two equilibrium points:

$x_e = (\ , \ )$, $u_e =$

$x_e = (\ , \ )$, $u_e =$

This also aligns with expectations since the first eq. is just pointing down, and the second is pointing vertically up.

In order to linearize arround an equilibrium we use the Jacobian Linearization procedure. Where we do a Taylor series expansion arround (x_e, u_e) of the nonlinear system's dynamic. The linearized LTI system matracies are then given by:

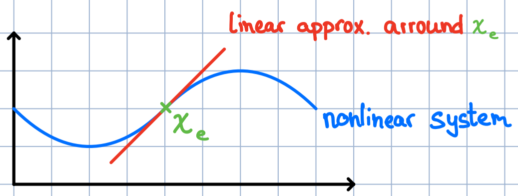
$$A = \frac{\partial f(x,u)}{\partial x}\bigg|_{(x_e,u_e)} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}\right|_{(x_e,u_e)}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

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

$$C = \frac{\partial h(x,u)}{\partial x}\bigg|_{(x_e,u_e)} = \left.\begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_p}{\partial x_1} & \cdots & \frac{\partial h_p}{\partial x_n} \end{bmatrix}\right|_{(x_e,u_e)} \in \mathbb{R}^{p\times n}$$

$$D = \frac{\partial h(x,u)}{\partial u}\bigg|_{(x_e,u_e)} = \left.\begin{bmatrix} \frac{\partial h_1}{\partial u_1} & \cdots & \frac{\partial h_1}{\partial u_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_p}{\partial u_1} & \cdots & \frac{\partial h_p}{\partial u_m} \end{bmatrix}\right|_{(x_e,u_e)} \in \mathbb{R}^{p\times m}$$

Graphically you can think of this approach as follows:



Lets try it on our swing example:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = \frac{1}{ml^2}\left[-lmg \sin x_1(t) - c\, x_2(t) + l \cos x_1(t)\, u(t)\right] \\ y(t) = x_1(t) \end{cases}$$

with equilibrium points at: $x_e = (0,0)$, $u_e = 0$

$x_e = (\pi,0)$, $u_e = 0$

Computing A:

$$A = \left.\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}\right|_{(x_e,u_e)} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_n} \end{bmatrix}\right|_{(x_e,u_e)}$$

$$= \left.\begin{bmatrix} \frac{\partial}{\partial x_1}\left[x_2(t)\right] & \frac{\partial}{\partial x_2}\left[x_2(t)\right] \\ \frac{\partial}{\partial x_1}\left[\frac{1}{ml^2}\left[-lmg \sin x_1(t) - c\, x_2(t) + l\cos x_1(t)\, u(t)\right]\right] & \frac{\partial}{\partial x_2}\left[\frac{1}{ml^2}\left[-lmg \sin x_1(t) - c\, x_2(t) + l\cos x_1(t)\, u(t)\right]\right] \end{bmatrix}\right|_{(x_e,u_e)}$$

$$A = \begin{bmatrix} \quad & \quad \end{bmatrix}$$

If we do the same for B, C, and D we get:

$$B = \begin{bmatrix} 0 \\ \frac{1}{ml} \end{bmatrix} \quad , \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad , \quad D = 0$$

Important: we linearized the system arround $x_e = (0,0)$, $u_e = 0$! If we linearize near the other equilibrium point $x_e = (\pi, 0)$, $u_e = 0$ we will get a different state space representation:

$$A = \begin{bmatrix} 0 & 1 \\ +\frac{g}{l} & -\frac{c}{ml^2} \end{bmatrix} \quad , \quad B = \begin{bmatrix} 0 \\ \frac{1}{ml} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad , \quad D = 0$$

This LTI approximation works well if we stay close to the equilibrium point. By choosing the right controller we can also ensure it works well withe real nonlinear system.