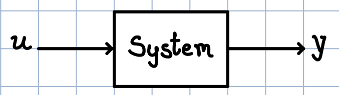
1. Recital 20.09.24

Modeling

We want to learn how to mathematically represent dynamic systems. Specifically we want to write down equations that express the output as a function of the input, and some internal parameters.



Important: All models are wrong, but some are useful.

Inputs can be:

→ Endogenous: can be manipulated by the designer, e.g. control inputs

→ Exogenous: generated by the environment and can't be controlled, e.g. disturbances

they encompass everything that affects the system over time.

The outputs can be classified as:

→ Measured outputs: what we can measure (sensors), e.g. speed of car

→ Performance outputs: not directly measurable, but we want to control, e.g. avg. fuel consumpt.

they encompass everything that we observe about the system over time.

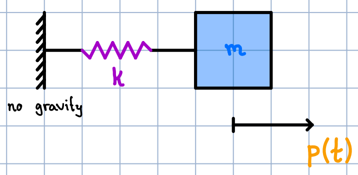
Internal parameters are system specific and do not change over time.

All systems that we want to describe, can be represented by differential equations.

That means that the way the system is changing, is related to the current state

how the system changes = f(current state)

Example:



Newtons second law : $F = m \cdot a$

For this system : $-$ $=$

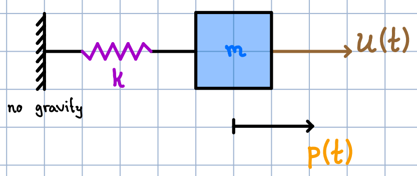
Hookes Law:

$F_{spring} = kx \rightarrow$ Displacement
↳ Spring const.
↓ Restoring force

$\rightarrow$ $=$ $-$ — 2^{nd} order ODE

Since k and m are constant, we can observe that a change of $\ddot{p}(t)$ must be accompanied by a change in $p(t)$.

We can also apply an external force $u(t)$ to the system.



Now Newtons second law yields : $-$ $=$

$\rightarrow$ $=$ $+$ — 2^{nd} order ODE

Now the input force $u(t)$ also influences how the system changes.

If we take closer look at the 2^{nd} order ODE we can regocnize that it can be re-written as a system of 1^{st} order ODEs (see Linear Algebra II).

With the substitution:

$$x_1(t) = p(t)$$

$$x_2(t) = \dot{p}(t)$$

we obtain:

$$\dot{x}_1(t) = x_2(t)$$

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

$\Longleftrightarrow$

which can also be re-written in matrix form.

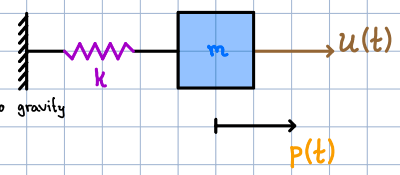
$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{m} \end{pmatrix} u(t)$$

This is something we already know and can solve. The only thing left to do is defining what we want to measure in our system, i.e. what the output is. Here we can measure, for example, the velocity of the mass.

The output $y(t)$ is then equal to $\dot{p}(t)$ and therefore $x_2(t)$. Now we can write everything together as:

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

$$y(t) = \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$$



This system of equations now represents the original mass-spring system.

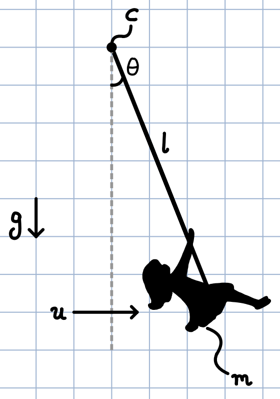
The vector $x = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$ contains all state variables of the system. The states describe how a system changes internally over time. It can be thought of as a memory, containing a summary of how the system behaved in the past. Given the internal states and the current input, we can uniquley predict any future behavior. We will introduce a formal definition next week!

We can now generalize this to a standard form that we will generally use to describe dynamic systems.

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

This is called the state-space form, since we are observing how the state vector x changes.

Example: Swing (from the lecture)

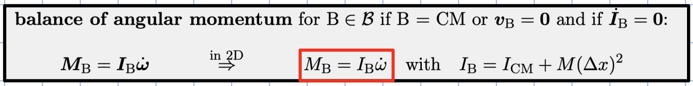


→ Notation

θ : Swing angle
m : Point mass
l : Pendulum length
g : Gravity
c : Friction coefficent
u : Horizontal push force

→ Given:

balance of angular momentum (basically $F = ma$ for rotations)



balance of angular momentum for $\mathrm{B} \in \mathcal{B}$ if $\mathrm{B} = \mathrm{CM}$ or $\boldsymbol{v}_\mathrm{B} = \boldsymbol{0}$ and if $\dot{\boldsymbol{I}}_\mathrm{B} = \boldsymbol{0}$:

$$\boldsymbol{M}_\mathrm{B} = \boldsymbol{I}_\mathrm{B}\dot{\boldsymbol{\omega}} \quad \overset{\text{in 2D}}{\Rightarrow} \quad \boxed{M_\mathrm{B} = I_\mathrm{B}\dot{\omega}} \quad \text{with} \quad I_\mathrm{B} = I_\mathrm{CM} + M(\Delta x)^2$$

(Mech. III summary)

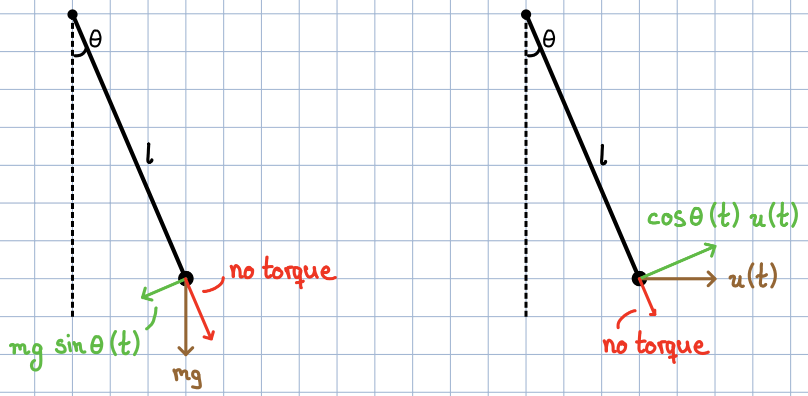
$$M = I\dot{\omega}$$

$$\updownarrow$$

$$\text{gravitytorque} + \text{frictiontorque} + \text{pushtorque} = J\ddot{\theta}$$ (from lecture slides)

The formula includes the sum of all torques acting on the system, the moment of inertia, and the angular acceleration. In our case the values are:

$$I = J = ml^2 \quad , \qquad M = \underset{\underset{-lmg\,\sin\theta(t)}{\downarrow}}{\text{gravitytorque}} + \underset{\underset{-c\dot{\theta}(t)}{\downarrow}}{\text{frictiontorque}} + \underset{\underset{l\cos\theta(t)\,u(t)}{\downarrow}}{\text{pushtorque}}$$



So our ODE becomes:

$$ml^2\ddot{\theta} = -lmg\,\sin\theta(t) - c\dot{\theta}(t) + l\cos\theta(t)\,u(t)$$

and we define the output to be:

$$y(t) = \theta(t)$$

Now we set: and obtain:

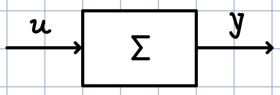
$$\begin{aligned} x_1(t) &= \theta(t) \\ x_2(t) &= \dot{\theta}(t) \end{aligned} \quad \Longrightarrow \quad \begin{aligned} \dot{x}_1(t) &= x_2(t) \\ ml^2\dot{x}_2(t) &= -lmg\,\sin x_1(t) - c\,x_2(t) + l\cos x_1(t)\,u(t) \\ y(t) &= x_1(t) \end{aligned}$$

which again is a system of 1st order ODEs in the state space representation. In the comming weeks we will further investigate state space representation.

Block diagrms

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.

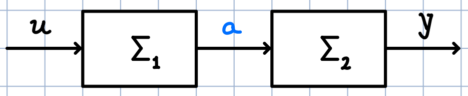
Lets begin with the simple case given by:



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

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

We can also have two systems, one after the other, like so:



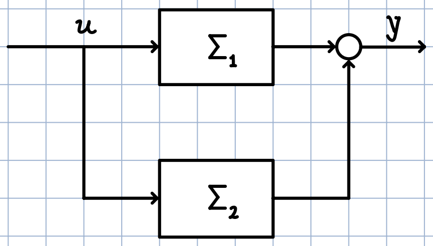
To help us find the input-output relation we can define an intermediate signal a, and analyze both blocks seperatley:

1. $y = \Sigma_2 a$

2. $a = \Sigma_1 u$

⟶ combining both results in: $\boxed{y = \Sigma_2 \Sigma_1 u}$

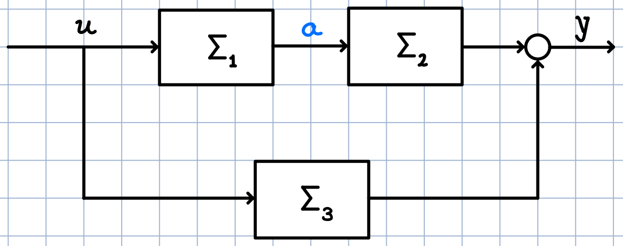
We can also have two systems in parallel:



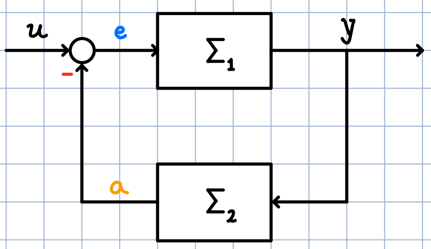
$$y = \Sigma_1 u + \Sigma_2 u$$

$$\boxed{y = (\Sigma_1 + \Sigma_2) u}$$

If we combine both we obtain



We can also introduce (negative) feedback:



We follow the same recipe

$y = \Sigma_1 e$

$e = u - a$

$a = \Sigma_2 y$

$e = u - \Sigma_2 y$

$y = \Sigma_1 (u - \Sigma_2 y)$

so $y = \Sigma_1 (u - \Sigma_2 y)$, rearanging we obtain:

$$y = \Sigma_1 u - \Sigma_1 \Sigma_2 y$$

$$y + \Sigma_1 \Sigma_2 y = \Sigma_1 u$$

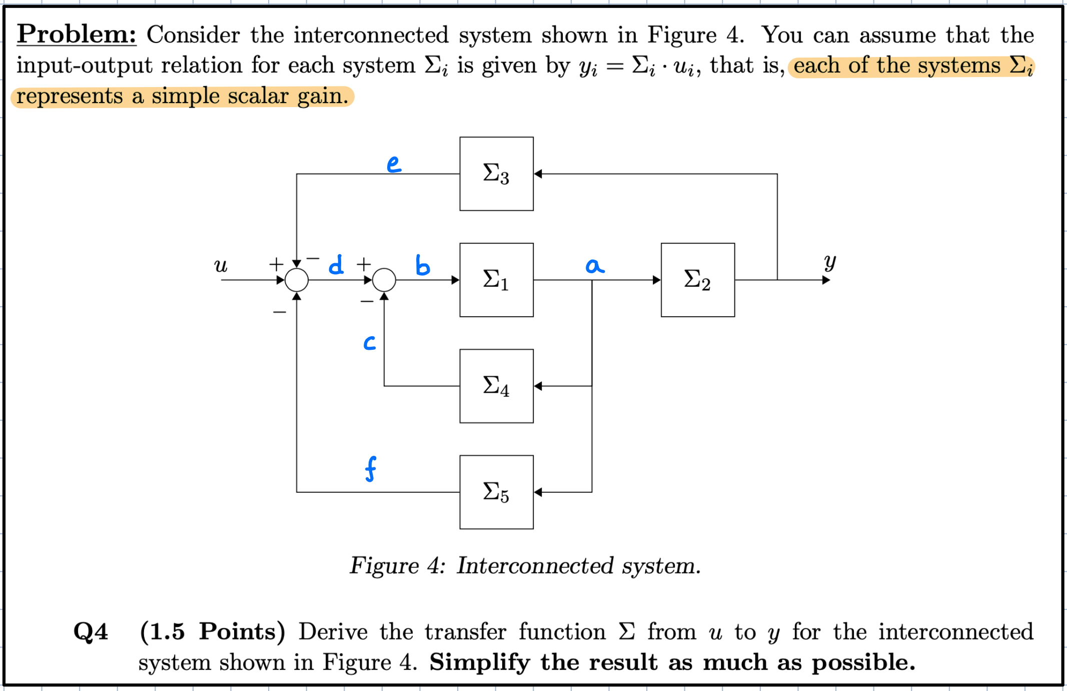
$$(1 + \Sigma_1 \Sigma_2) y = \Sigma_1 u$$

$$\boxed{y = (1 + \Sigma_1 \Sigma_2)^{-1} \Sigma_1 u} \quad \text{or} \quad \boxed{y = \frac{\Sigma_1 u}{1 + \Sigma_1 \Sigma_2}}$$

only if Σ_1 & Σ_2 are scalars!

Exam Problem:

HS 22:



Problem: Consider the interconnected system shown in Figure 4. You can assume that the input-output relation for each system Σ_i is given by $y_i = \Sigma_i \cdot u_i$, that is, each of the systems Σ_i represents a simple scalar gain.

e
Σ_3
u
+
−
d
+
b
Σ_1
a
Σ_2
y
−
−
c
Σ_4
f
Σ_5

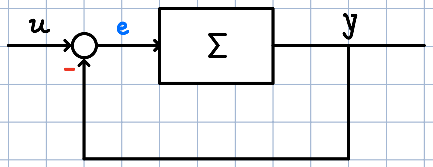
Figure 4: Interconnected system.

Q4 **(1.5 Points)** Derive the transfer function Σ from u to y for the interconnected system shown in Figure 4. **Simplify the result as much as possible.**

Important: We assume every Σ_i to be a scalar!

Let's do it step-by-step:

Lets go back one step and look at a simple feedback loop with Σ being a scalar gain, meaning $y = \Sigma e$ with $\Sigma \in \mathbb{R}$



The input-output relation is given as

$$y = \Sigma e$$

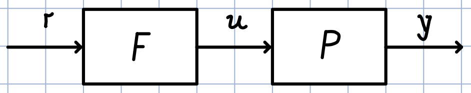
$$e = u - y$$

$$y = \Sigma(u-y) \iff (1+\Sigma)y = \Sigma u \iff y = \frac{\Sigma}{1+\Sigma} u$$

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., $y \to \infty$).

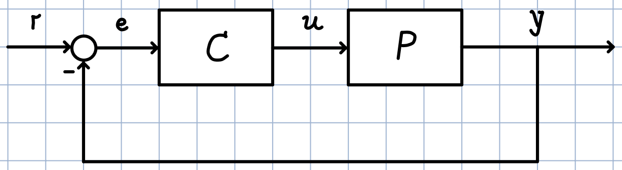
With the help of block diagrams we can also vizualize some basic control architectures.

Feed-forward :



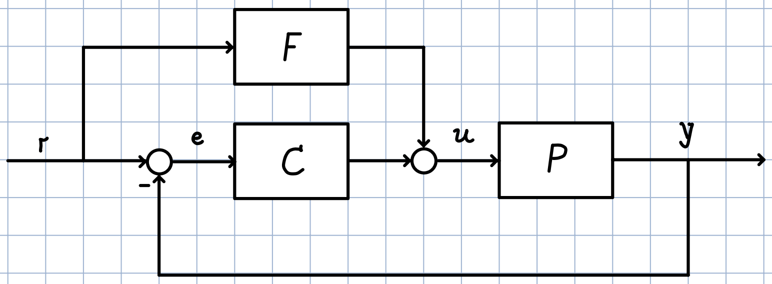
requires precise knowledge of plant

Feedback :



can handle disturbances and uncertainties, but can introduce instability

To degrees of freedom :



good transient behavior and good tracking of fast changing references.