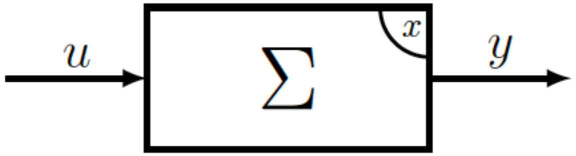
Control Systems I
Recitation 02

https://n.ethz.ch/~jgeurts/
jgeurts@ethz.ch

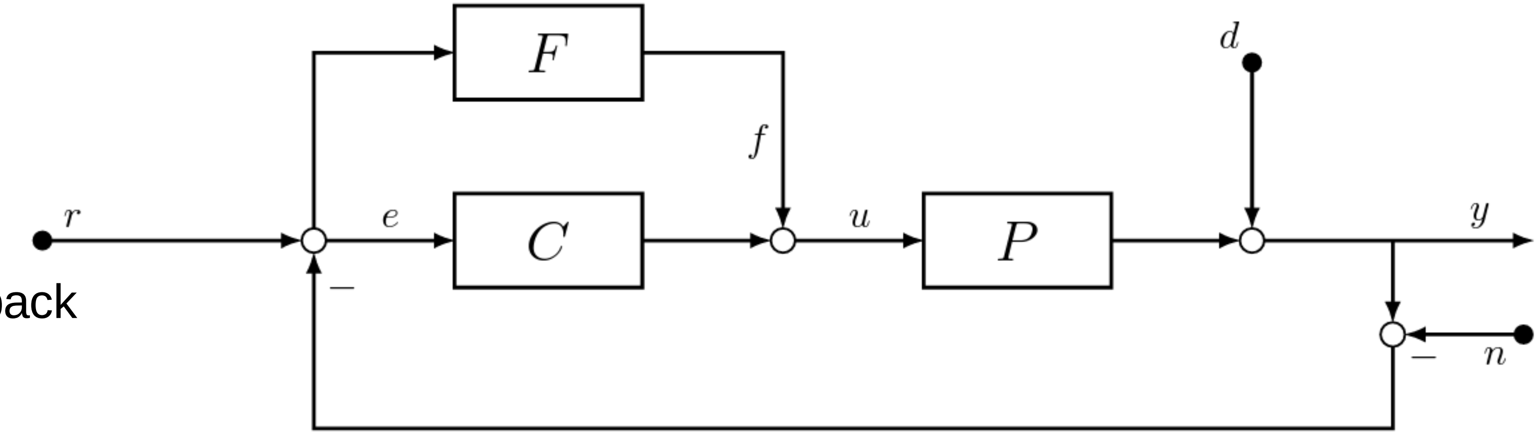
Last week

- What is a system:
  - A **system** in Control Systems is an **operator** which defines a relationship between an input, an internal state and an output.



  - **u, x and y** are in general **time variant** -> $u(t), x(t), y(t)$
  - **Differential Equations:** $\frac{d}{dt}x(t) = f\big(x(t), u(t)\big) \rightarrow y(t) = h\big(x(t), u(t)\big)$
  - The general description of a system looks as follows: $y(t) = \Sigma\big(u(t), x(t)\big)$
    - $u(t)$ being the input, $x(t)$ being the internal states of the system, $y(t)$ being the output, $\Sigma(t)$ being the system itself

Last week



- Control Structures:
  - Feedforward and Feedback
    - r = reference value
    - e = error signal
    - C = controller
    - F = feedforward
    - u = control signal
    - P = plant/system
    - d = external disturbances
    - n = measurement noise

Outline

- Modelling
  - What? Why? How?
  - Example
- Output Calculation
  - What? Why? How?
  - Example
- Control Intro – Benefits and Dangers
  - Example
- Block Diagrams
  - What? Why? How?
  - Example

System Modelling

What? and Why?

- What?
  - Mathematical description of the underlying physics (starting from a black-box)
    - ODE/PDE(CS1)
    - Could also just be a mathematical model from data or a Neural Network
- Why?
  - Understand the system properties
    - Stability, Controllability, Observability
  - Simulate the system
    - Use a physics simulator to simulate the system behaviour
    - Use simulations to test controller
  - Derive/Synthesize a Controller
- **All models are wrong, some are useful**
  - Know the limits of your model!!!

System Modelling

How?

- **1. Identify system boundaries**
- **2. Write down differential equations:**
  - **Difficult for now but will become clear later**
- **3. Formulate in standard form**
- 4. Normalize (next week)
- 5. Linearize (next week)

System Modelling

How?

- **1. Identify system boundaries:**
  - Inputs: *Outside world things that affect the system and vary over time.*
    - Endogenous Inputs: Controlled by us -> usually control input u(t)
    - Exogeneous Inputs: Things we can't control -> usually disturbances d(t)
  - Output: *The measurements/observations we make of our system*
    - Measured outputs: Directly measurable (e.g. speed, position)
    - Performance outputs: Not directly measurable (e.g. fuel efficiency)
  - States: *Quantities defining the system ("memory") – time variant*
    - Often become apparent when setting up the ODE/PDE
    - For a car -> position and velocity
  - Parameters: *Quantities that define the system but are constant over time*
    - Mass of a car, length of a pendulum
- All these aspects can be chosen freely and need to be considered carefully
  - Car model, with or without air drag, with or without motor model?

System Modelling

How?

- **2. Write down differential equations:**
  - In general, we can use the **storage method**:
    - Define storage variables (energy, momentum, temperature, …)
    - Define inflows and outflows (heat flow, physical work, …)
    - Write down differential equation as:
      - $\frac{d}{dt}\text{storage} = \sum\text{infows} - \sum\text{outflows}$
  - In practice we can also directly use **physical laws**:
    - Newton: $\sum F = ma = m\ddot{x}, \rightarrow \sum T = I\ddot{\theta}$
    - Heat: $\dot{Q}_{ij} = c_{ij}\left(T_i - T_j\right), \rightarrow C\dot{T} = \sum\dot{Q}$ (explicit representation of the storage method)
  - Which to choose depends on the problem:
    - *Physical laws*: simple mechanical problems, electrical problems
    - *Storage methods*: complicated mechanical problems, thermodynamic problems

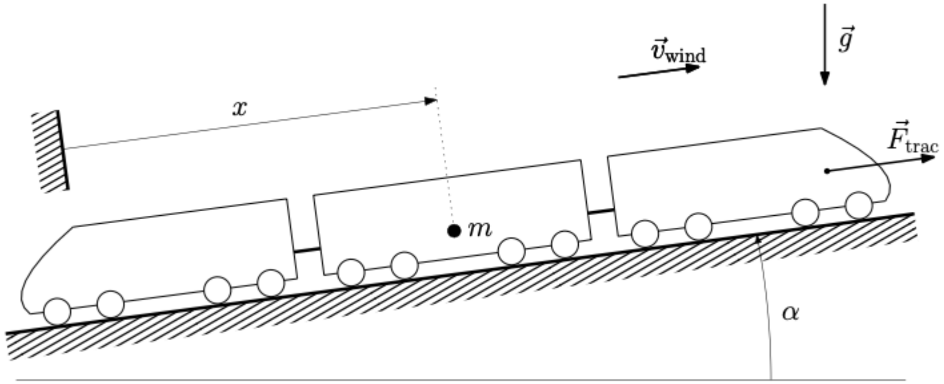
System Modelling

How?

- **3. Formulate in standard form:**
  - The ODE/PDE are of higher derivatives and not comparable to one-another
  - We want to have a standardized form:
    - $\frac{d}{dt}x(t) = f(x(t), u(t)) \rightarrow y(t) = h(x(t), u(t))$
    - Comparable form, unified mathematics, only first order ODE
  - When a higher derivative is present, define intermediate derivatives:
    - $\ddddot{z} = f(z(t), v(t))$
    - Define: $\mathbf{x} = \begin{bmatrix} x_1 = z \\ x_2 = \dot{z} \\ x_3 = \ddot{z} \\ x_4 = \dddot{z} \end{bmatrix} \Rightarrow \frac{d}{dt}\mathbf{x} = \begin{bmatrix} \dot{x}_1 = \dot{z} = x_2 \\ \dot{x}_2 = \ddot{z} = x_3 \\ \dot{x}_3 = \dddot{z} = x_4 \\ \dot{x}_4 = \ddddot{z} = f(x(t), u(t)) \end{bmatrix}$ we substitute v=u and z=x
    - $y(t) = h(x(t), u(t))$ usually straight forward

System Modelling

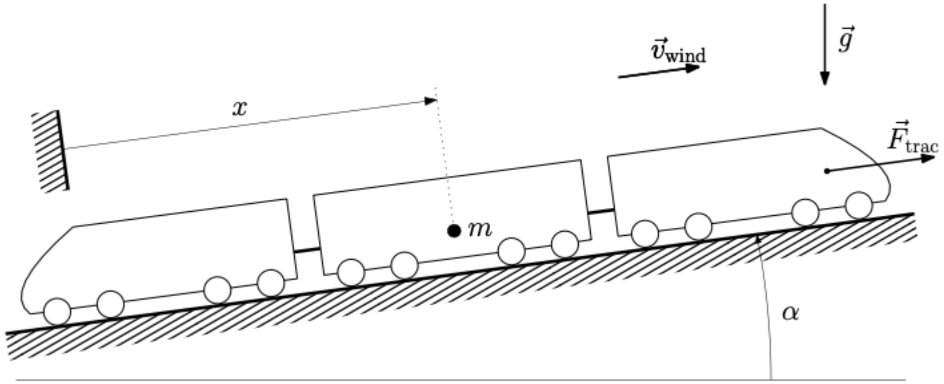
Example - Train Driving up a Hill ($v_{wind} = v_{wind}(t), \alpha = const$)



- **1. Identify System Boundaries:**
  - Inputs
    - Endogenous:
    - Exogeneous:
  - Output
    -
  - State
    - $x(t) = [\quad]$
  - Parameters
    -

System Modelling

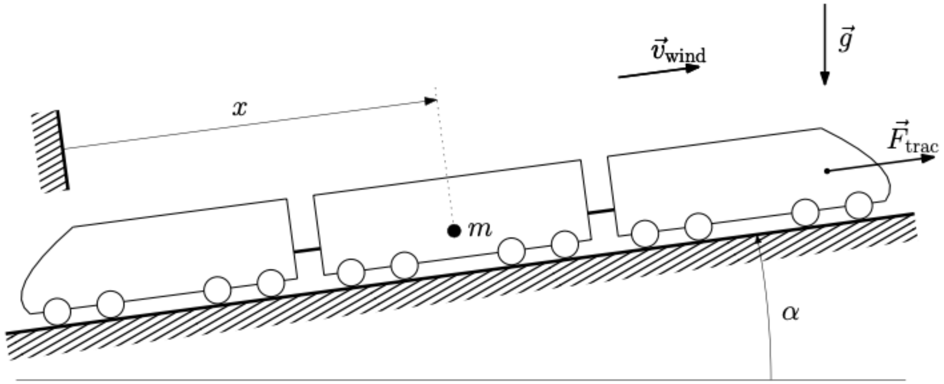
Example - Train Driving up a Hill ($v_{wind} = v_{wind}(t), \alpha = const$)



- **2. Write down differential equation:**
  - 1. Directly using Newton: (faster for simple systems)
    - $\sum F = ma =$
    - $\sum F =$
    - Drag: $F_{drag}(t) =$
    - Gravity: $F_{grav}(t) =$
    - → $=$

System Modelling

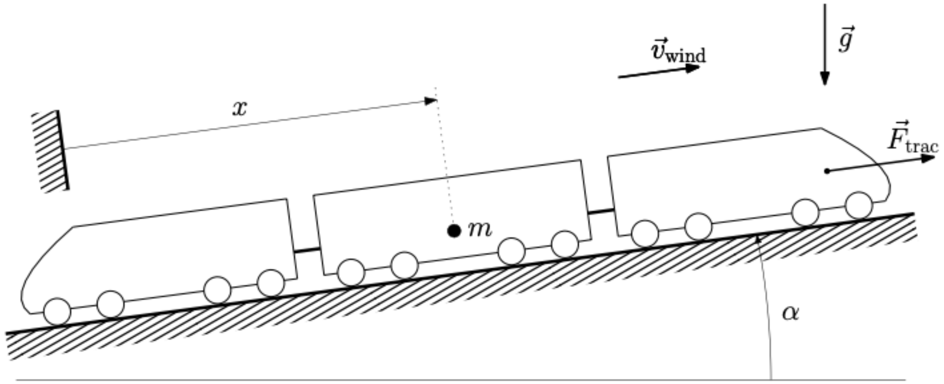
Example - Train Driving up a Hill ($v_{wind} = v_{wind}(t), \alpha = const$)



- **2. Write down differential equation:**
  - 2. Storage Method: (easier for complex systems)
    - Storage variable: Energy (potential and kinetic)
      - $\text{storage} = \frac{1}{2} m\dot{x}^2(t) + mg\sin(\alpha)\,x(t)$
    - Inflows: Traction power
      - $IN(t) = F_{trac}(t)\,\dot{x}(t)$
    - Outflows: Wind resistance loss
      - $OUT(t) = F_{drag}(t)\,\dot{x}(t)$
    - Equation: $\frac{d}{dt} storage = \sum infows - \sum outflows$
      - $\frac{d}{dt}\left[\frac{1}{2} m\dot{x}^2(t) + mg\sin(\alpha)\,x(t)\right] = F_{trac}(t)\dot{x}(t) - F_{drag}(t)\,\dot{x}(t)$
      - $m\ddot{x}(t)\,\dot{x}(t) + mg\sin(\alpha)\,\dot{x}(t) = F_{trac}(t)\dot{x}(t) - F_{drag}(t)\,\dot{x}(t)$
      - $\rightarrow \mathbf{m\,\ddot{x}(t) = F_{trac}(t) - F_{drag}(t) - \sin(\alpha)\,mg}$

System Modelling

Example - Train Driving up a Hill ($v_{wind} = v_{wind}(t), \alpha = const$)



- **3. Formulate in standard form :** $\frac{d}{dt}x(t) = f(x(t), u(t))$, $y(t) = h(x(t), u(t))$
  - We have a second derivative:
    - $=$
    - $=$
  - Define: $=$ $=$ $\rightarrow \mathbf{x(t)} = [\quad], u(t) =$ $, d(t) =$
    - $\rightarrow \dot{\mathbf{x}}(\mathbf{t}) = [\quad] = [\quad] = [\quad]$
    - $\frac{d}{dt}\mathbf{x(t)} = [\quad] = f(\mathbf{x}(t), u(t))$
    - $y(t) =$

Output Calculation

What?, Why? and How?

- What?
  - Given an input $u(t)$ and x_0 we want to know the output $y(t)$ of the system
- Why?
  - Simulations
  - See how the system behaves for different inputs -> controller synthesis
- How?
  - Almost impossible to find a closed form solution $y(t) = g(u(t), x_0)\ \forall\ u(t), x_0$
  - Rely on discretization and approximations:
    - $\dot{x} = f\big(x(t), u(t)\big) \approx \frac{x(t+\Delta t) - x(t)}{\Delta t}$
    - $\mathbf{x(t + \Delta t) \approx x(t) + \Delta t \cdot f\big(x(t), u(t)\big)}$
    - Solve this at every time step to get a simulation
- Example together with controller intro

Control Intro

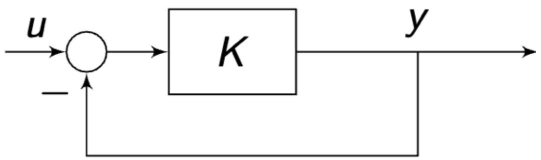
Benefits and Dangers

- Let’s say we have a model --> What’s next?
  - Setup simulation
  - Analyse system
  - Synthesize Controller
  - Notebook 1 demo
- Benefits:
  - Stabilize a system
  - Improve performance
  - Reach desired state
- Dangers:
  - Can break your physical system
  - Dangerous for people

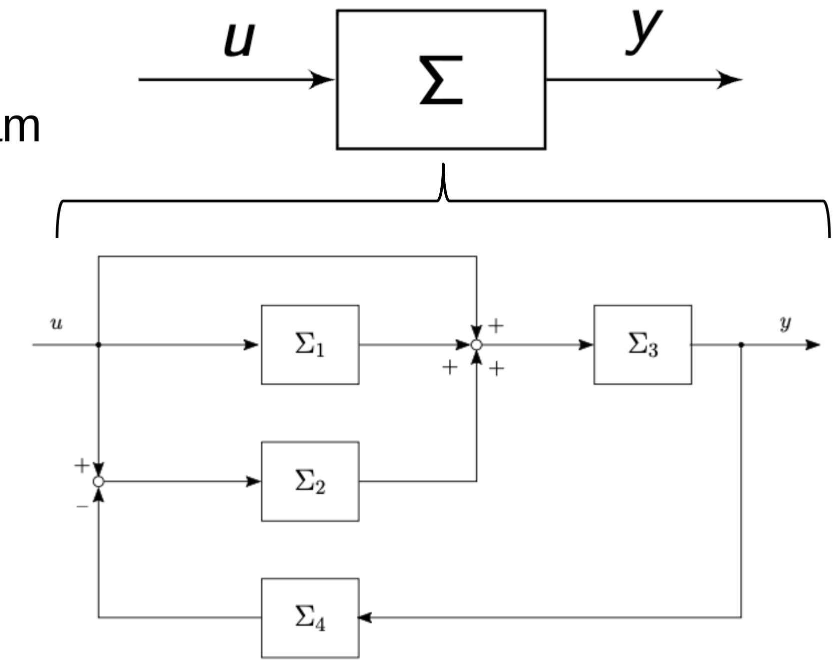
Block Diagrams and Block Diagram Algebra

What? and Why?

- What?
  - Graphical representation how system interact with each other
  - Mathematical Formulation of **Transfer Functions** a Block Diagram
    - $y(t) = \Sigma \cdot u(t)$
- Why?
  - Simulink uses them
  - Can break down complex control systems
  - Graphical sometimes easier to grasp
  - Solving Algebraic Loops → one system
    - Algebraic Loop:

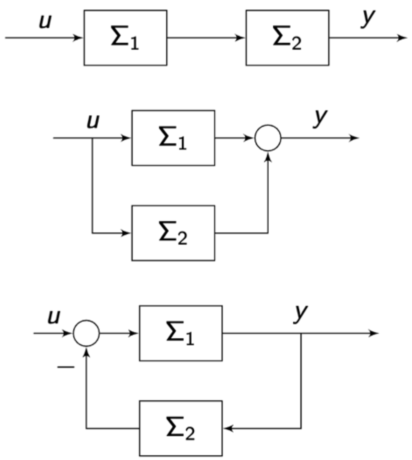


- When the input of a system depends on the output of the system (bad)



Block Diagrams and Block Diagram Algebra

How?

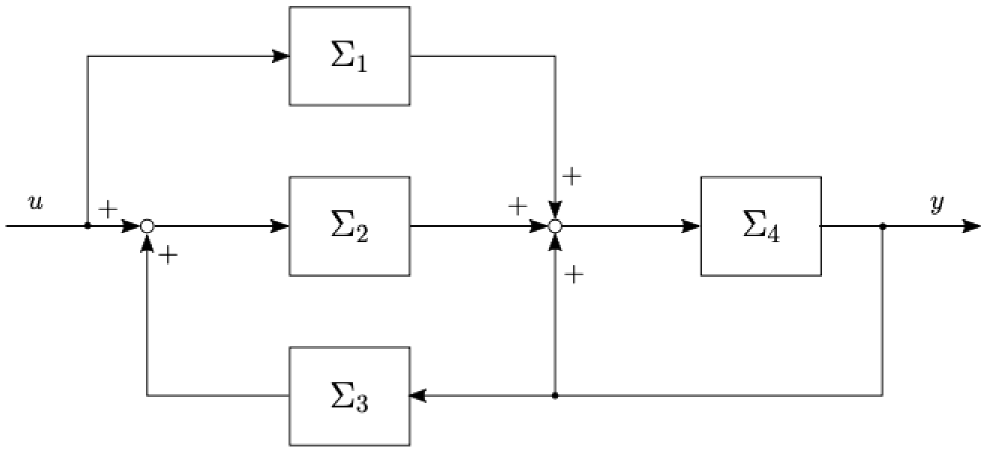


- How?
  - *1. Pick and Place different interconnections*
    - Serial: $\Sigma_2\Sigma_1$ (order matters for MIMO systems!)
    - Parallel: $\Sigma_1 + \Sigma_2$
    - Negative Feedback: $(\mathrm{I} + \Sigma_1\Sigma_2)^{-1}\Sigma_1$
    - Often not possible or very hard
  - **2. Start from the end and work yourself to the start:**
    - 1. Define every output of a block as function of its immediate input
      - $\mathrm{y_i} = \Sigma_\mathrm{i}\ \mathrm{u_i}$
    - 2. Define each input as function of other systems output (if applicable)
      - $\mathrm{u_i} = \mathrm{f}(\mathrm{y_j}) \forall\ \mathrm{j}$
    - 3. Solve system of equation
      - Best to start from the back and work your way to the beginning

Block Diagrams and Block Diagram Algebra

Example

- Derive the Transfer Function:
  - 1. Define outputs:
    - $=$
      $=$
      $=$
      $=$
      $=$
  - 2. Define inputs:
    - $=$
      $=$
      $=$
      $=$



- 3. Solve system of equations (starting from the back)
  -