Dynamic Systems Modelling

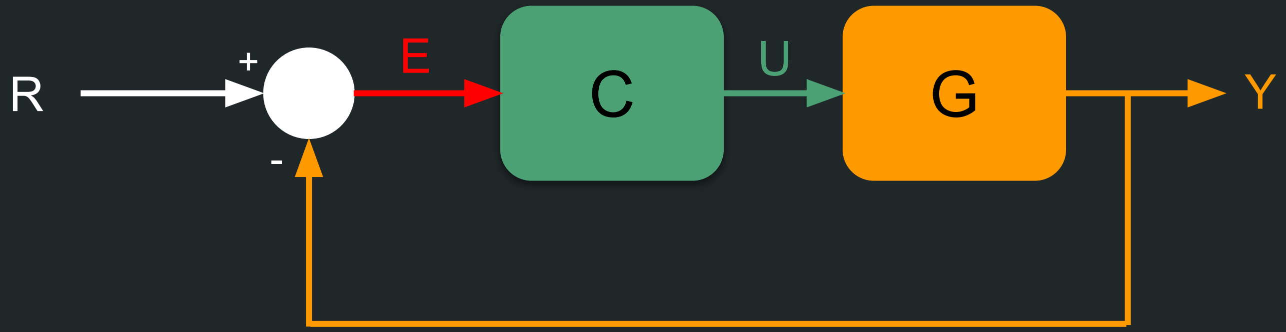
Renan G. Maidana

Porto Alegre, 2018

Previously...

- What is feedback?

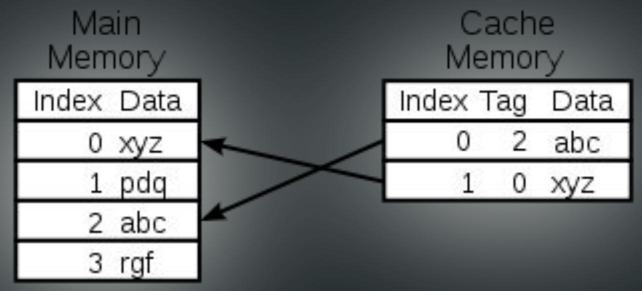
- What is control?



Previously...

- Practical examples
- Advantages/Disadvantages
- Applications (general and in Robotics)

- Challenge: Cache Hit Ratio



Today!

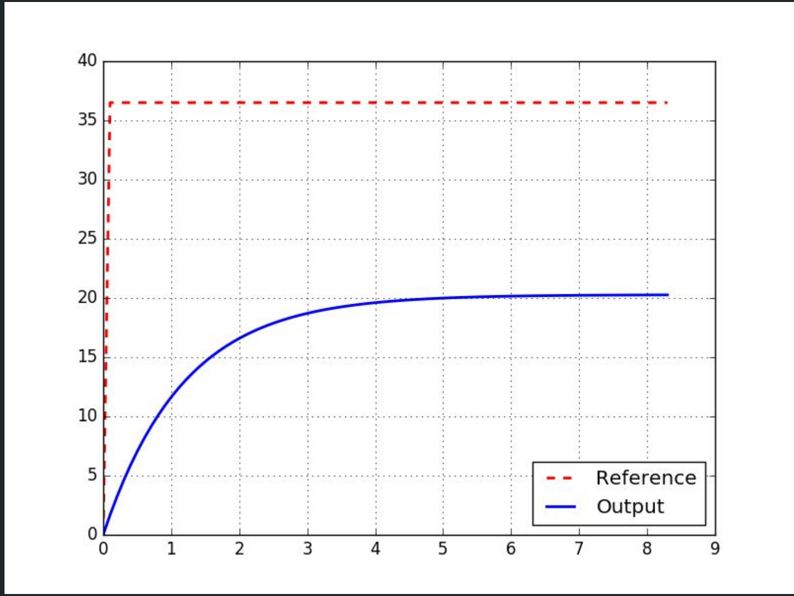
- First half: **Dynamic Systems Modelling**
  - Introduction
  - Differential Equations
  - Transfer Functions
  - Laplace and Z Transforms
  - Discretization
  - Systems Identification
  - State-spaces
  - Examples and Exercises

Today!

- Second half: **Time Domain Analysis**
  - Steady-state and Transient Response
  - Root-locus Analysis
  - Stability Criteria
  - Examples and Exercises

Introduction to Dynamic Systems Modelling

- A **dynamic model** is a mathematical representation of a system in which the response to a given input is **not immediate**



Introduction to Dynamic Systems Modelling

- In control theory, a model represents **the behavior of a dynamic system between its input and output**

Introduction to Dynamic Systems Modelling

- In control theory, a model represents **the behavior of a dynamic system between its input and output**

- In order to design and/or analyze control systems, the controlled process model (G) must be known

Introduction to Dynamic Systems Modelling

- How can we obtain these models?
  - **Differential Equations**
  - **Systems Identification**
  - Markov Chain/HMM
  - Artificial Neural Networks
  - Graphs
  - Petri nets
  - ...

Introduction to Dynamic Systems Modelling

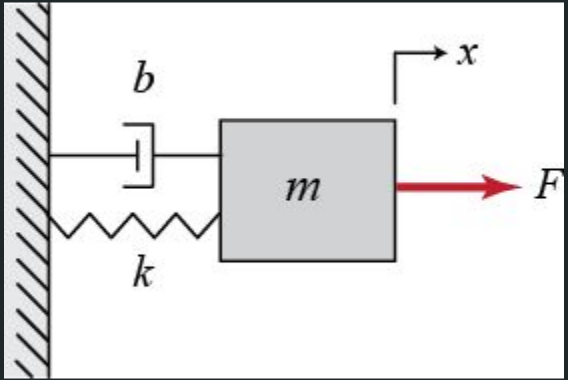
- **Differential equations** are perhaps the most popular way of modelling dynamic systems

- A differential equation typically represents the relationship between **rates of change** of the various elements in a system
  - In dynamic systems, an ordinary differential equation (ODE) is used to model the rate of change of the output w.r.t the input

$$a_n \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_1 \frac{dy}{dt} + a_0 y = u(t)$$

Differential Equations

- Example: **Spring-Mass-Damper system**

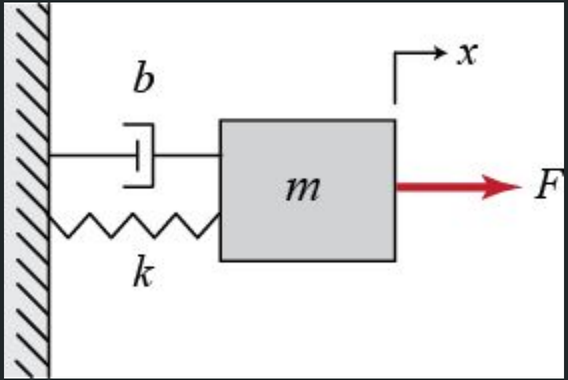


$$m\ddot{x} + b\dot{x} + kx = 0$$

- m = Object mass (kg)
- x = Object displacement (m)
- b = Damping ratio
- k = Spring constant (N/m)

Differential Equations

- Example: **Spring-Mass-Damper system**



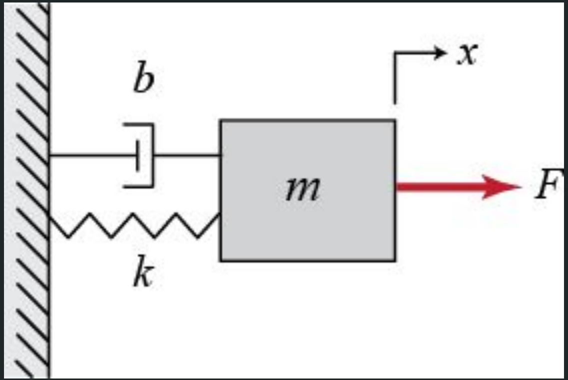
- m = Object mass (kg)
- x = Object displacement (m)
- b = Damping ratio
- k = Spring constant (N/m)

$$m\ddot{x} + b\dot{x} + kx = 0$$

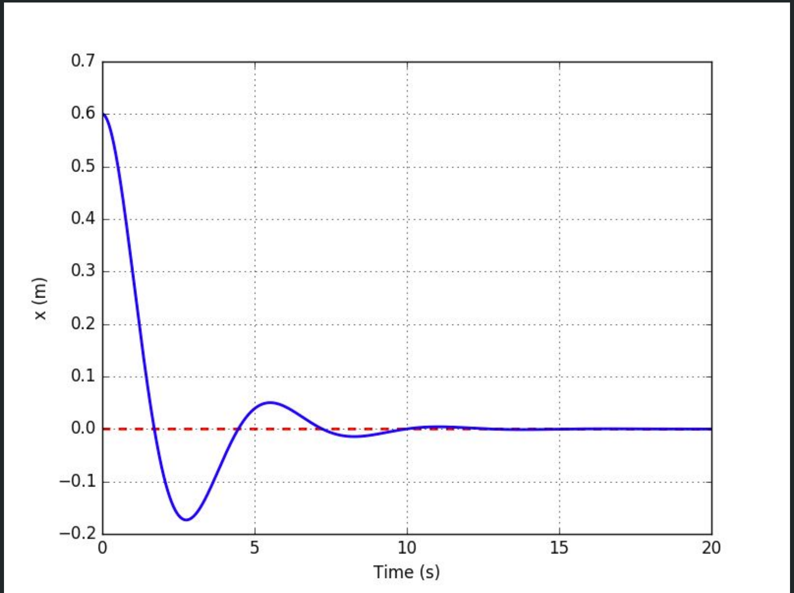
It is the sum of the equilibrated forces acting upon the system

Differential Equations

- Example: **Spring-Mass-Damper system**



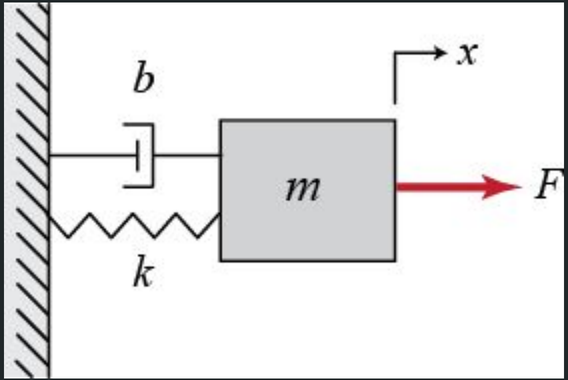
$$m\ddot{x} + b\dot{x} + kx = 0$$



m = 1 kg; b = 1; k = 1.5 N/m;
Initial conditions: x = 0.6

Differential Equations

- Example: **Spring-Mass-Damper system**

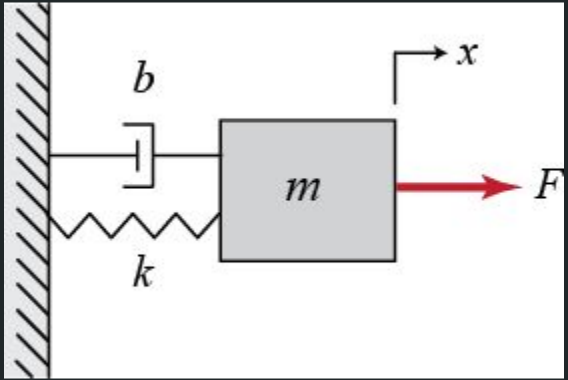


$$m\ddot{x} + b\dot{x} + kx = 0$$

- The SMD system as modelled here is known as an **autonomous system**, because it is unaffected by external influences

Differential Equations

- Example: **Spring-Mass-Damper system**



- The SMD system as modelled here is known as an **autonomous system**, because it is unaffected by external influences

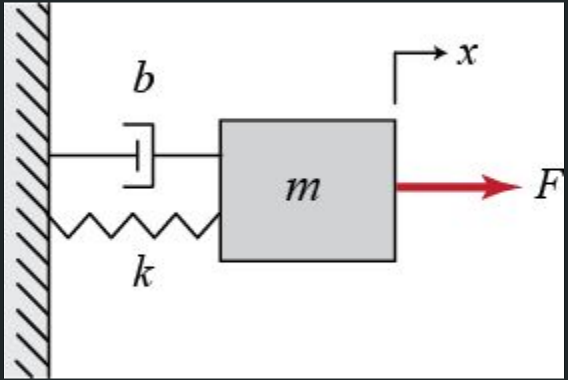
- In control theory, it is useful to model external disturbances or controlled forces

$$m\ddot{x} + b\dot{x} + kx = u$$

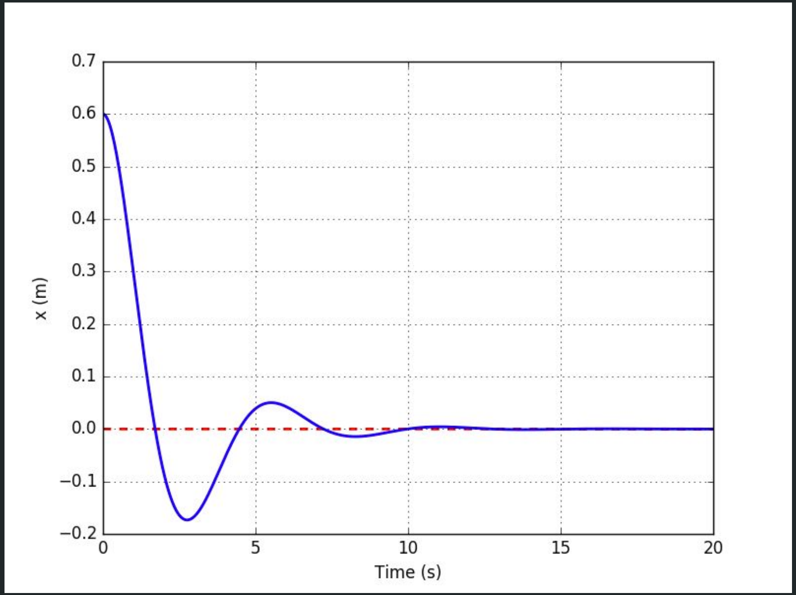
- This is known as a **forced** or **controlled** differential equation

Differential Equations

- Example: **Spring-Mass-Damper system**



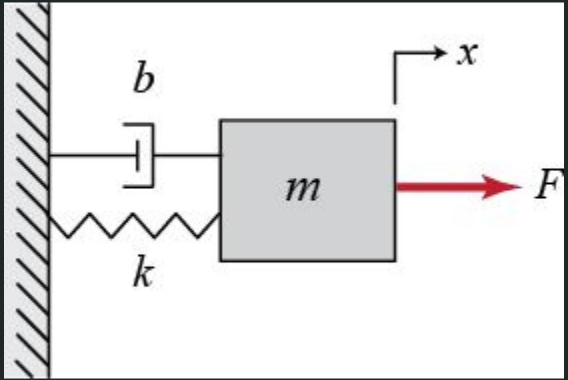
$$m\ddot{x} + b\dot{x} + kx = 0$$



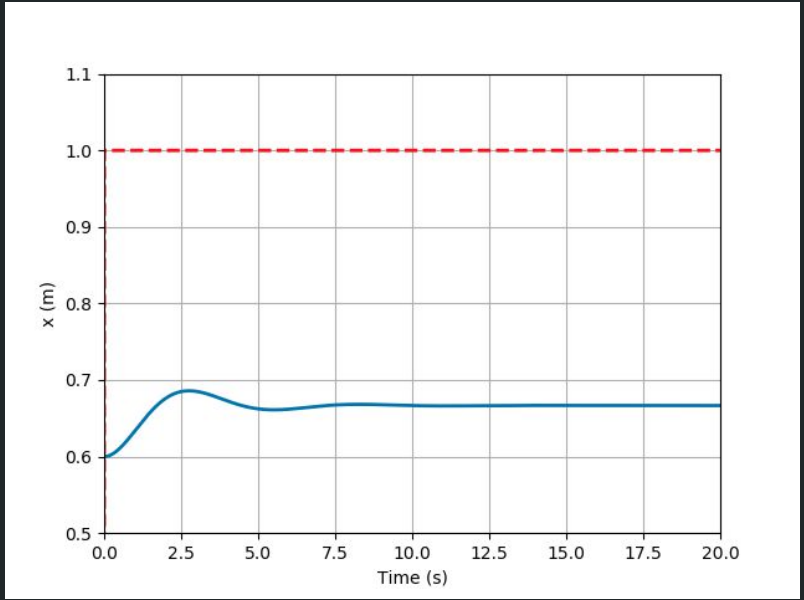
m = 1 kg; b = 1; k = 1.5 N/m;
Initial conditions: x = 0.6 m

Differential Equations

- Example: **Spring-Mass-Damper system**



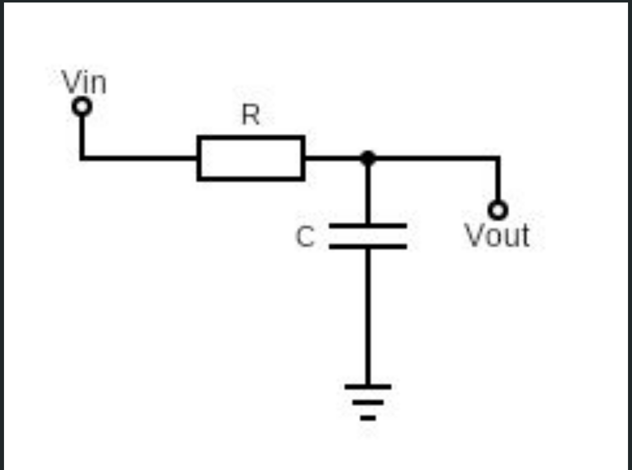
$$m\ddot{x} + b\dot{x} + kx = u$$



m = 1 kg; b = 1; k = 1.5 N/m;
Initial conditions: x = 0.6 m

Differential Equations

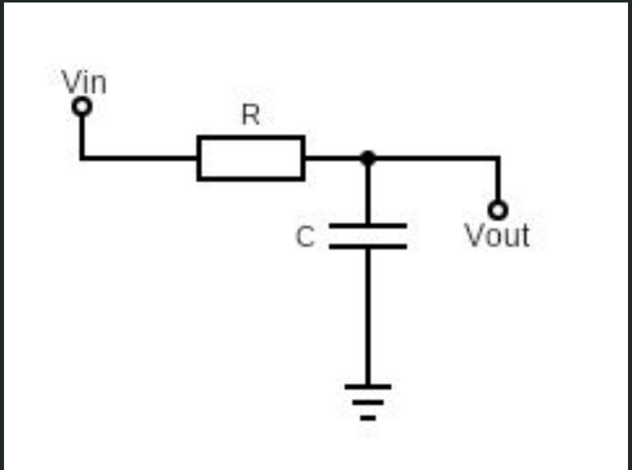
- Practice: **RC Low-pass filter**



- How can we model this system?
- Tip: Use Kirchoff's first law (node current)

Differential Equations

- Practice: **RC Low-pass filter**



$$\frac{V_i - V_o}{R} = C\dot{V}_o$$

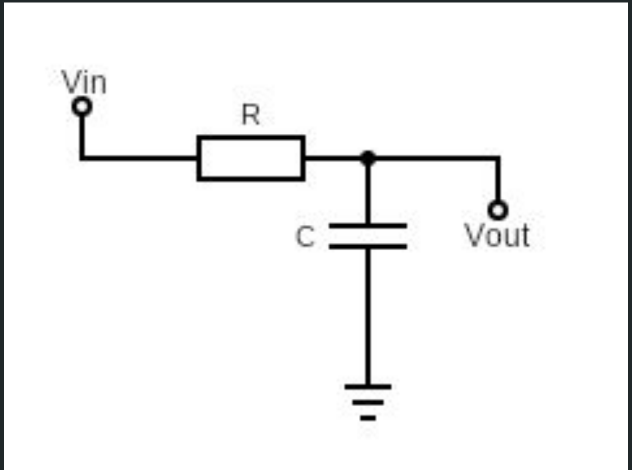
$$V_i - V_o = RC \times \dot{V}_o$$

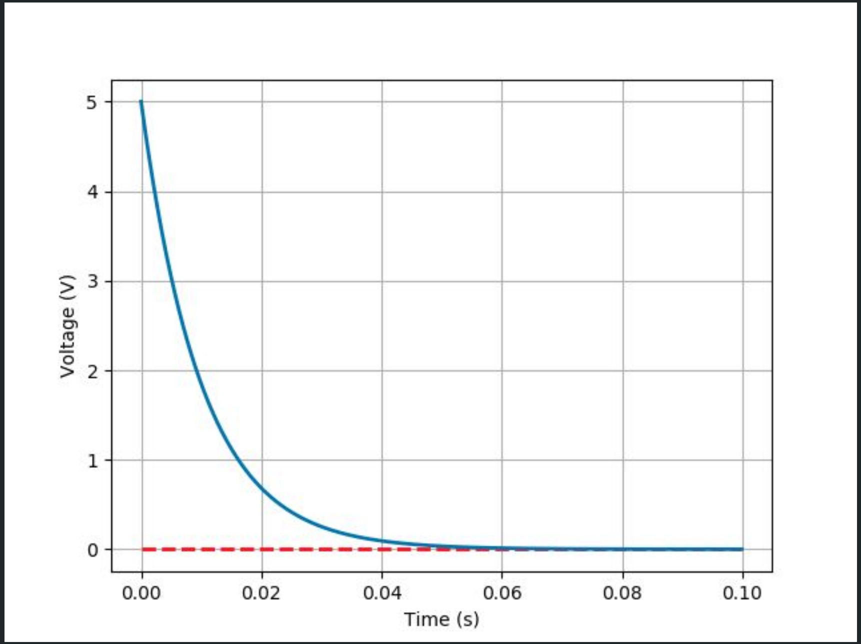
$$RC \times \dot{V}_o + V_o = V_i$$

Forced Differential Equation

Differential Equations

- Practice: **RC Low-pass filter**

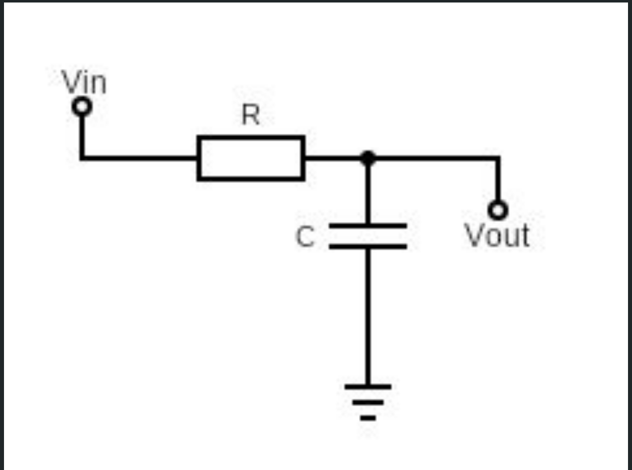


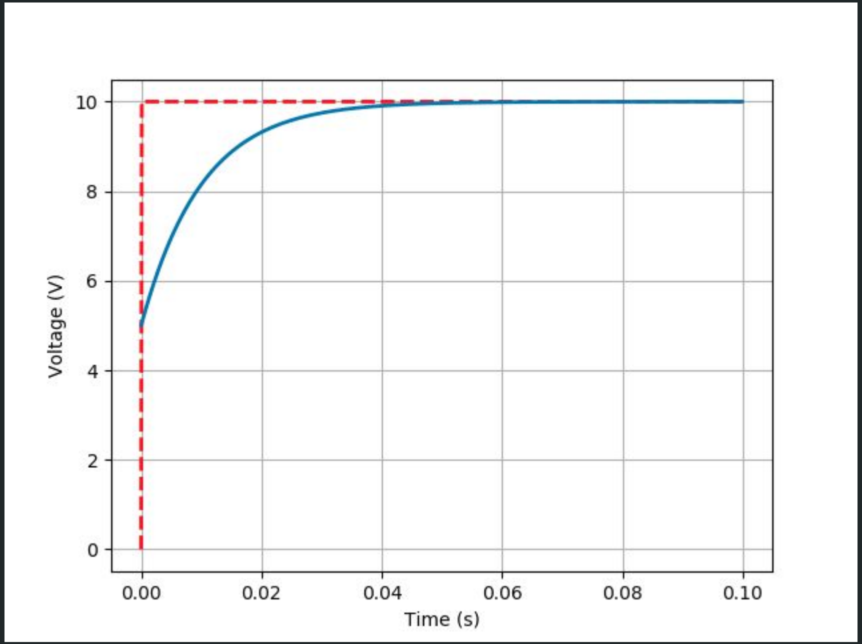


R = 10 kΩ; C = 1 μF;
Initial conditions: Vout = Vc = 5 V

Differential Equations

- Practice: **RC Low-pass filter**

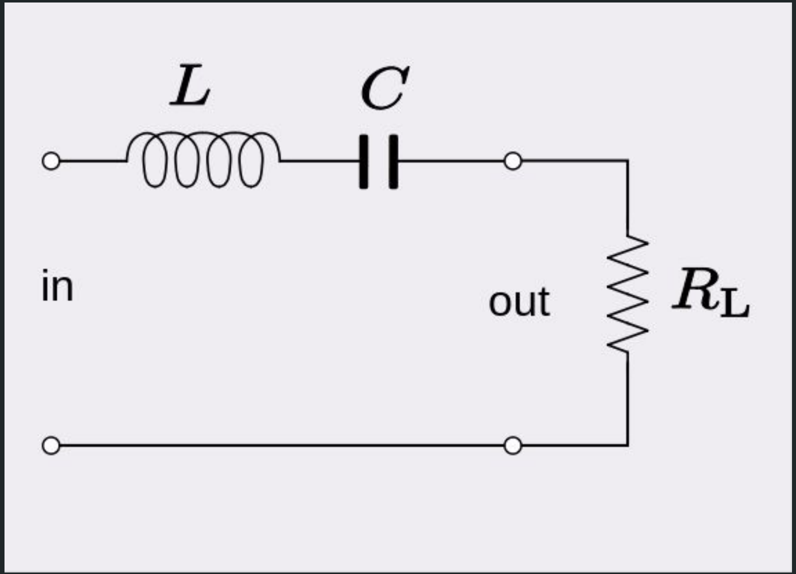




R = 10 kΩ; C = 1 μF;
Initial conditions: Vout = Vc = 5 V

Differential Equations

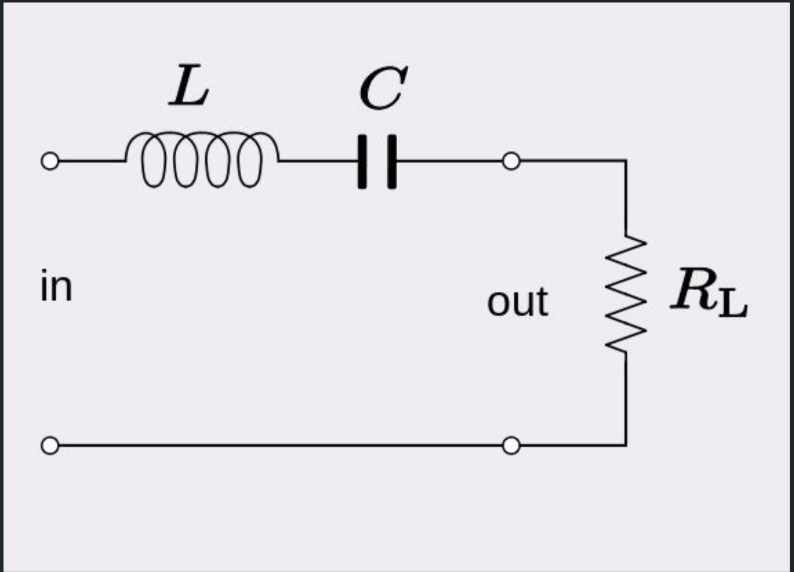
- Practice: **RLC low-pass filter**



- What are the autonomous and forced equations?

Differential Equations

- Practice: **RLC low-pass filter**

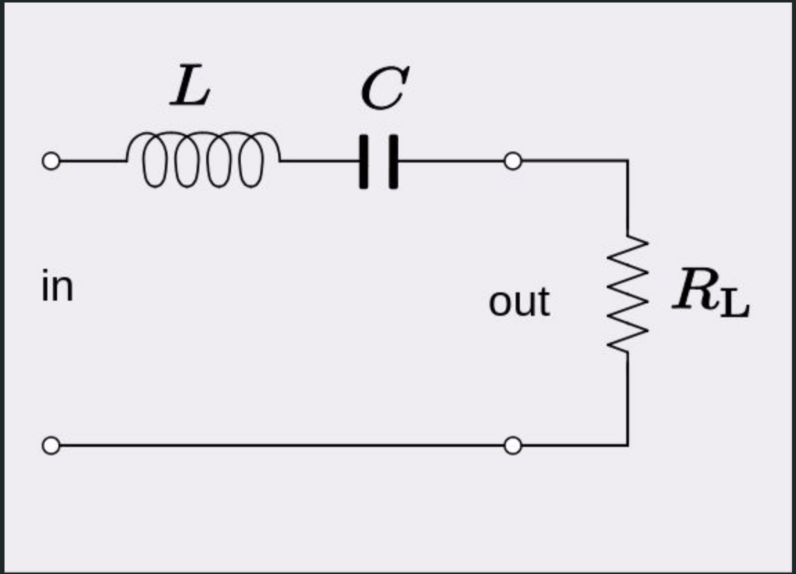


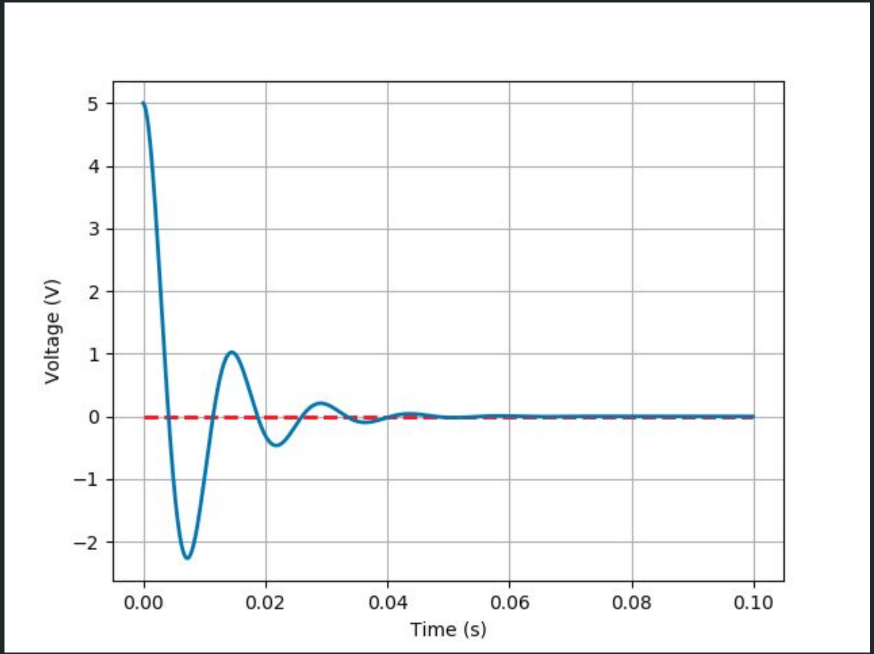
$$LC\ddot{V}_o + RC\dot{V}_o + V_o = V_i$$

- **Second-order** forced differential equation
  - When Vi = 0, it becomes autonomous

Differential Equations

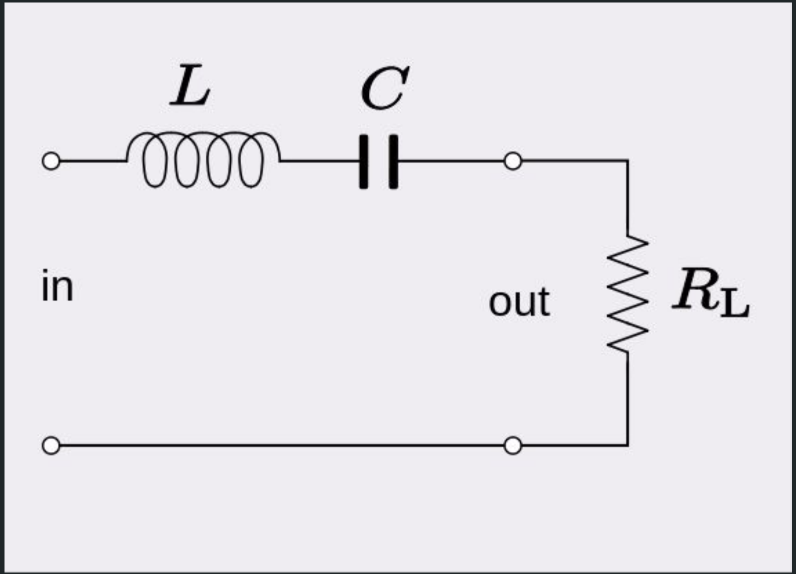
- Practice: **RLC low-pass filter**

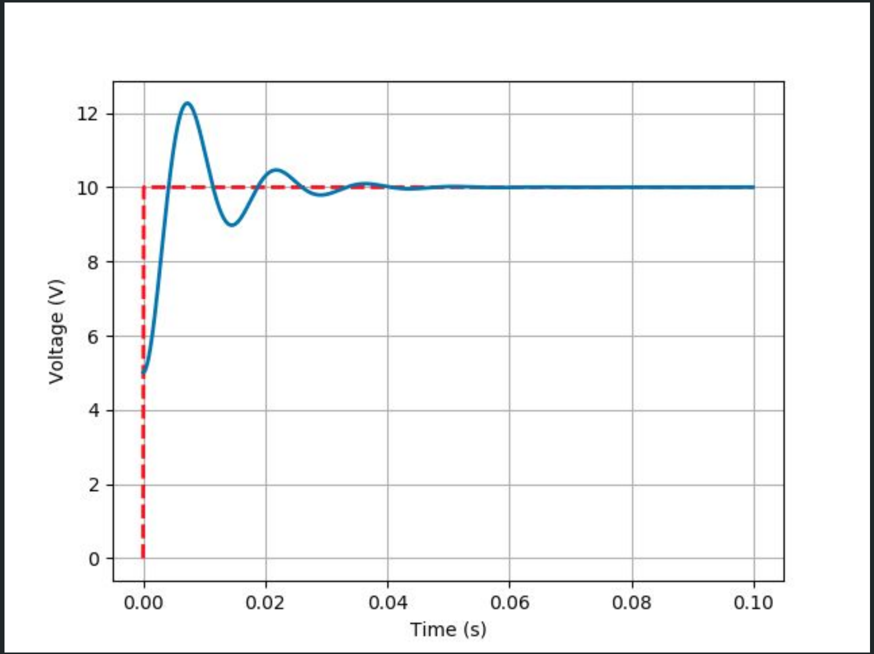




Differential Equations

- Practice: **RLC low-pass filter**





Differential Equations

- The solution to an ordinary differential equation is the **combination of independent variables** (e.g, time) which satisfies the ODE's condition

$$RC\dot{V}_o + V_o = V_i \longrightarrow RCdV_o + V_o dt = V_i dt$$

[Integral on all terms]

$$V_o(t + RC) = V_i t \longrightarrow V_o(t) = \frac{V_i \times t}{t + RC} + C_1$$

General Solution

Differential Equations

- Given any **initial conditions**, an ODE may have countless valid **particular solutions**

$$\text{If } V_o(0) = 5\ V \quad \therefore \quad 5 = \frac{V_i \times 0}{0 + RC} + C_1 \quad \therefore \quad C_1 = 5$$

$$V_o(t) = \frac{V_i \times t}{t + RC} + 5$$

Particular Solution

Transfer Functions

- A disadvantage of modelling dynamic systems with differential equations is **the need to know the system's initial conditions, which may not be constant**

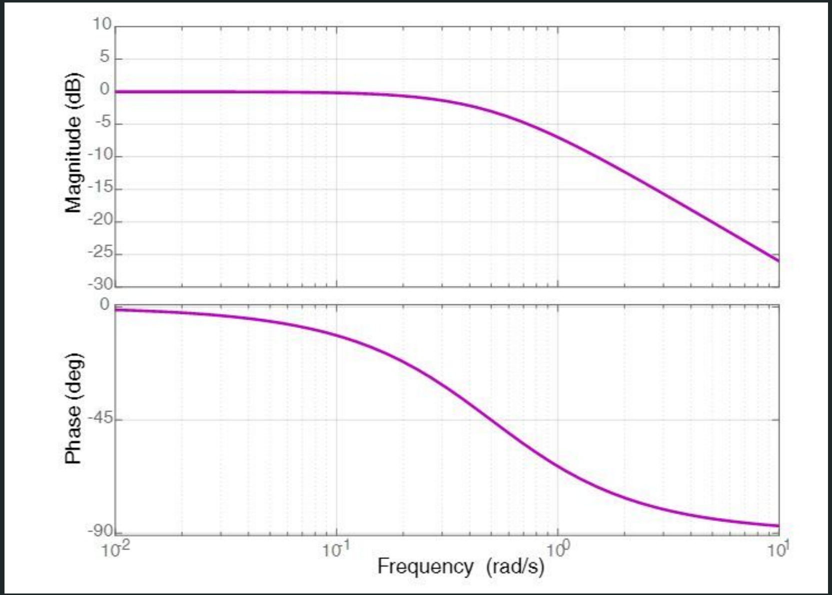
- Instead of using a time-domain model, we can look at the system from a **frequency domain** perspective

- A **Transfer Function** (TF) represents the input/output dynamics of a system through a generalized complex frequency $\boldsymbol{s}$ (= $j\omega$), rather than the time $\boldsymbol{t}$

Transfer Functions

- Typically, TFs are used in Electrical Engineering to model complex behaviors in circuits, and to analyze their frequency response

- With transfer functions, we can:
  - Represent voltage/current instant gain
  - Analyze a system's gain/phase margin
  - Analyze a system's stability and steady-state response
  - Etc…



Laplace and Z Transforms

- From a continuous-time differential equation, we can obtain a **continuous-time transfer function** by applying **Laplace Transforms**

$$F(s) = \int_0^\infty f(t)e^{-st}dt$$

$$\mathcal{L}\{\frac{d^n f(t)}{dt^n}\} = s^n F(s)$$

Laplace and Z Transforms

- From a continuous-time differential equation, we can obtain a **continuous-time transfer function** by applying **Laplace Transform**

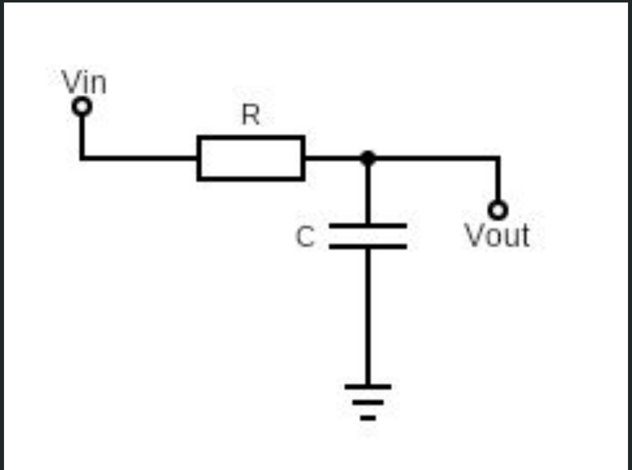
$$F(s) = \int_0^\infty f(t)e^{-st}dt \qquad \mathcal{L}\{\frac{d^n f(t)}{dt^n}\} = s^n F(s)$$

- From a discrete-time differential equation, we can obtain a **discrete-time transfer function** by applying **Z Transforms**

$$\mathcal{Z}\{f[n]\} = \sum_{n=0}^{\infty} f[n]z^{-n}$$

Transfer Func. (cont.)

- Continuous-time example: **RC Low-pass filter**



$$RC\dot{V}_o + V_o = V_i$$

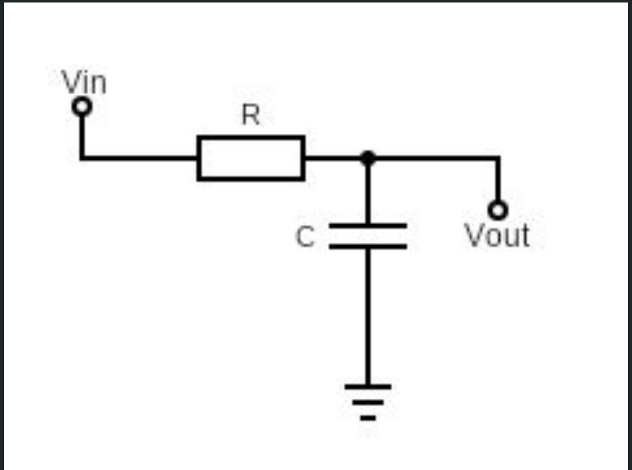
$$RC\dot{y} + y = u$$

$$RC\mathcal{L}\{\dot{y}\} + \mathcal{L}\{y\} = \mathcal{L}\{u\}$$

$$Y(s)(RCs + 1) = U(s) \quad \therefore \quad \frac{Y(s)}{U(s)} = \frac{1/RC}{s + 1/RC}$$

Transfer Func. (cont.)

- Continuous-time example: **RC Low-pass filter**



$$RC\dot{V}_o + V_o = V_i$$

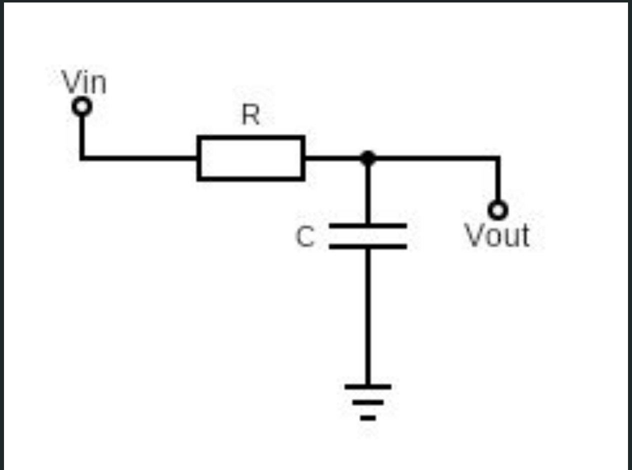
$$RC\dot{y} + y = u$$

$$RC\mathcal{L}\{\dot{y}\} + \mathcal{L}\{y\} = \mathcal{L}\{u\}$$

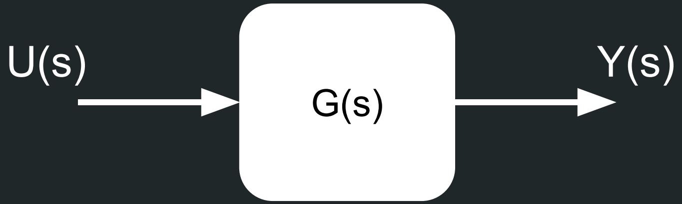
$$Y(s)(RCs + 1) = U(s) \quad \therefore \quad \boxed{\frac{Y(s)}{U(s)} = \frac{1/RC}{s + 1/RC}}$$

Transfer Func. (cont.)

- Continuous-time example: **RC Low-pass filter**



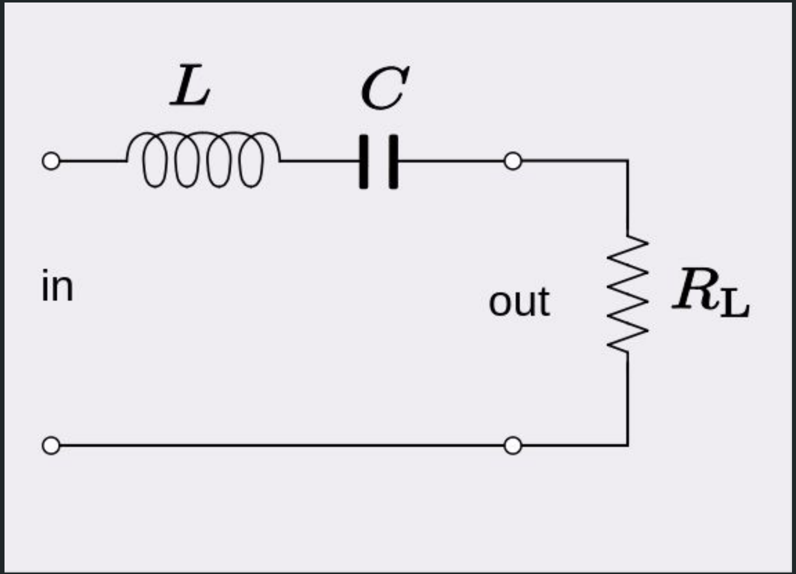
$$\frac{Y(s)}{U(s)} = \frac{1/RC}{s + 1/RC}$$



- What about discrete-time?

Transfer Func. (cont.)

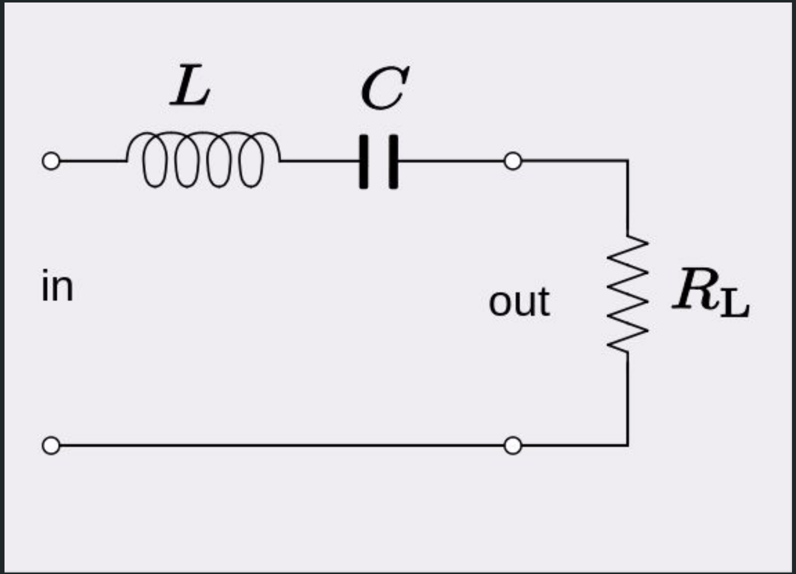
- Practice: **RLC low-pass filter**



$$LC\ddot{V}_o + RC\dot{V}_o + V_o = V_i$$

Transfer Func. (cont.)

- Practice: **RLC low-pass filter**

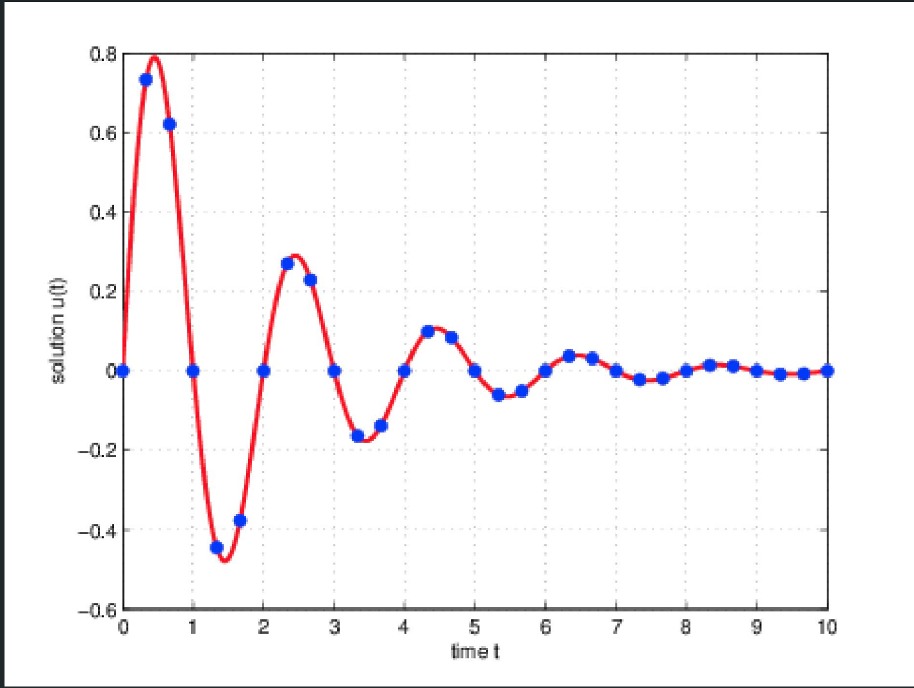


$$LC\ddot{V}_o + RC\dot{V}_o + V_o = V_i$$

$$\frac{Y(s)}{U(s)} = \frac{\frac{1}{LC}}{s^2 + \frac{R}{L}s + \frac{1}{LC}}$$

Discretization

- To obtain a system's discrete-time TF, we can **discretize** the modelled ODE



Discretization

- A discrete-time system can be represented by a **difference equation**, a discrete approximation of a differential equation

$$\dot{y} + \frac{y}{RC} = \frac{u}{RC} \qquad \dot{y} \approx \frac{y[k] - y[k-1]}{T}$$

Discretization

- A discrete-time system can be represented by a **difference equation**, a discrete approximation of a differential equation

$$\dot{y} + \frac{y}{RC} = \frac{u}{RC} \qquad \dot{y} \approx \frac{y[k] - y[k-1]}{T}$$

$$y[k] = \frac{1}{1 + \frac{T}{RC}} \left(\frac{T}{RC} u[k] + y[k-1] \right)$$

Discretization

- Finally, the Z-transform gives us the discrete-time transfer function

$$y[k] = \frac{1}{1 + \frac{T}{RC}} \left(\frac{T}{RC} u[k] + y[k-1] \right)$$

$$\frac{Y(z)}{U(z)} = G(z) = \frac{\frac{\alpha}{1+\alpha} z^2}{z^2 - \frac{1}{1+\alpha}}, \quad \text{where } \alpha = \frac{T}{RC}$$

Discretization

Euler-Backward

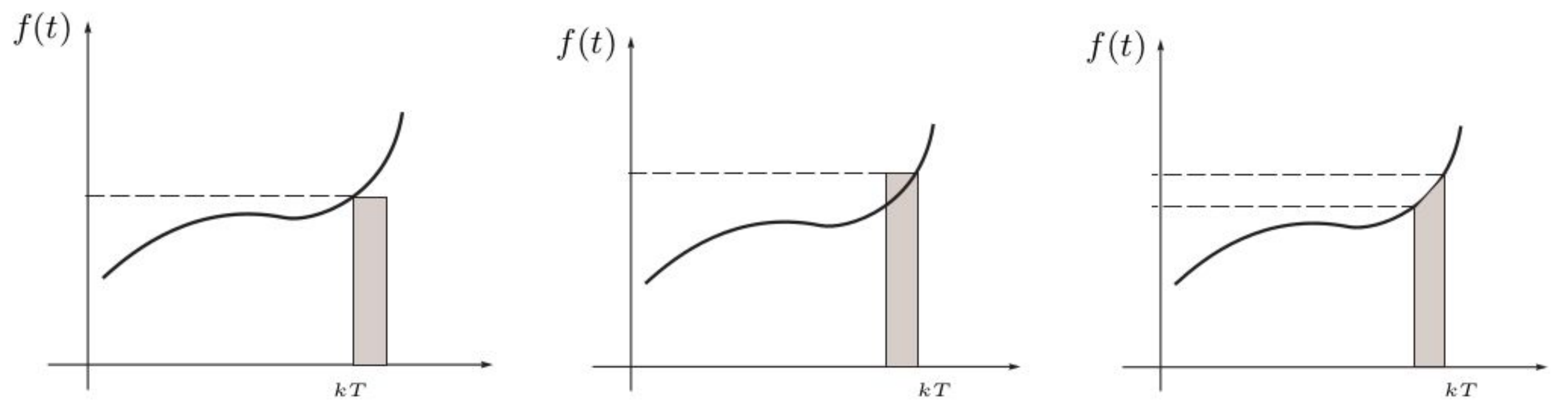
$$s \approx \frac{z-1}{Tz}$$

Euler-Forward

$$s \approx \frac{z-1}{T}$$

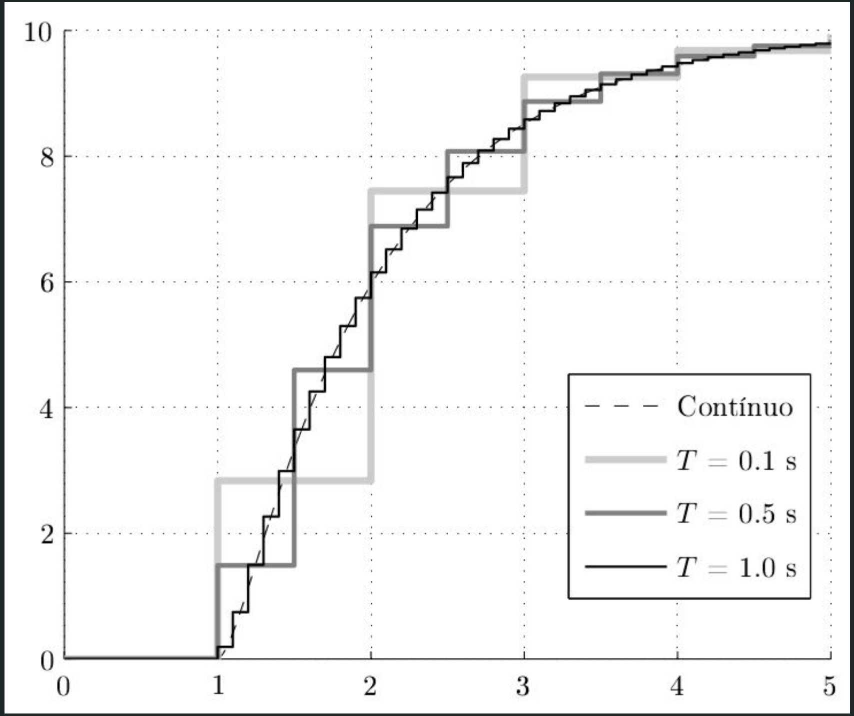
Tustin

$$s \approx \frac{2(z-1)}{T(z+1)}$$



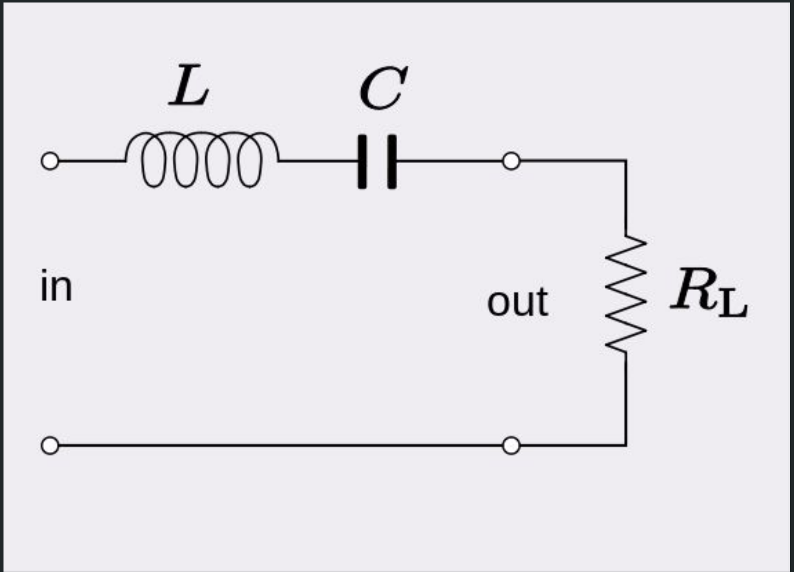
Discretization

- Effects of sampling time in discretization



Discretization

- Practice: **Discretization of the RLC low-pass filter**



$$LC\ddot{V}_o + RC\dot{V}_o + V_o = V_i$$

Discretization

- Practice: **Discretization of the RLC low-pass filter**

$$LC\ddot{V}_o + RC\dot{V}_o + V_o = V_i$$

$$y[k] = \frac{1}{T^2 + RCT + LC}(T^2u[k] + (2 + RCT)y[k-1] - y[k-2])$$

Systems Identification

- For complex systems, modelling transfer functions through differential equations can be challenging

- If we stimulate a system with a known input and record its output, it is possible to **approximate the transfer function through some regression**
  - Normal equation

Systems Identification

- To do this, we must first “guess” an approximation with the same (unknown) order of our system

$$\frac{Y(s)}{U(s)} = \frac{w_n^2}{s^2 + 2\xi w_n s + w_n^2}$$

Systems Identification

- Then we discretize it

$$\frac{Y(s)}{U(s)} = \frac{w_n^2}{s^2 + 2\xi w_n s + w_n^2}$$

$$y[k] = \frac{T^2 w_n^2}{T^2 wn^2 + 2T\xi w_n + 1} u[k] + \frac{2(1 + T\xi w_n)}{T^2 wn^2 + 2T\xi w_n + 1} y[k-1] - \frac{1}{T^2 wn^2 + 2T\xi w_n + 1} y[k-2]$$

$$y[k] = \theta_1 u[k] + \theta_2 y[k-1] + \theta_3 y[k-2]$$

Systems Identification

- Then we discretize it

$$\frac{Y(s)}{U(s)} = \frac{w_n^2}{s^2 + 2\xi w_n s + w_n^2}$$

$$y[k] = \boxed{\frac{T^2 w_n^2}{T^2 wn^2 + 2T\xi w_n + 1}} u[k] + \boxed{\frac{2(1 + T\xi w_n)}{T^2 wn^2 + 2T\xi w_n + 1}} y[k-1] - \boxed{\frac{1}{T^2 wn^2 + 2T\xi w_n + 1}} y[k-2]$$

$$y[k] = \boxed{\theta_1} u[k] + \boxed{\theta_2} y[k-1] + \boxed{\theta_3} y[k-2]$$

Systems Identification

- Then we represent it as a matrix multiplication

$$\begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ \vdots \\ y[n] \end{bmatrix} = \begin{bmatrix} u[0] & y[-1] & y[-2] \\ u[1] & y[0] & y[-1] \\ u[2] & y[1] & y[0] \\ \vdots & \vdots & \vdots \\ u[n] & y[n-1] & y[n-2] \end{bmatrix} \times \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

$$Y = \psi\theta$$

Systems Identification

- Then we solve for θ

$$Y = \psi\theta \qquad \theta = (\psi^T\psi)^{-1}\psi^T Y$$

Systems Identification

- Then we solve for θ

$$Y = \psi\theta$$

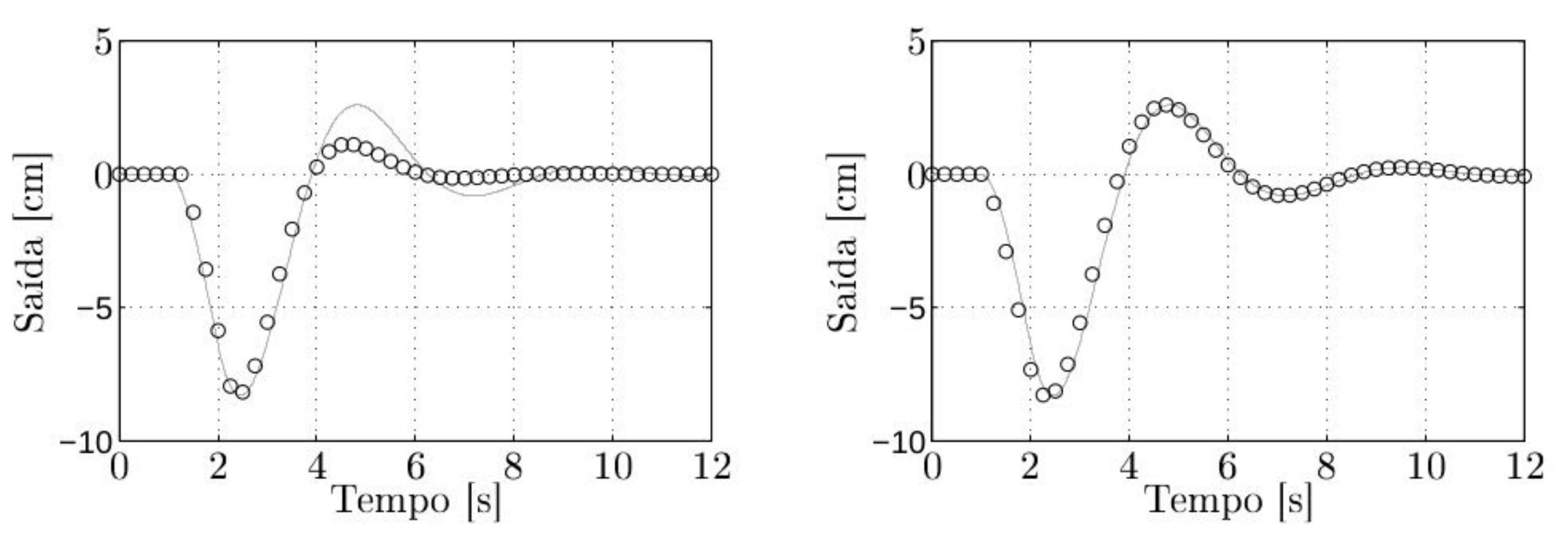
Regressors' Matrix

$$\theta = (\psi^T \psi)^{-1} \psi^T Y$$

Normal Equation

Systems Identification

- Effect of the approximation order on the output



$$\theta_1 y[k] + \theta_2 y[k-1] + \theta_3 u[k-1]$$

$$\theta_1 y[k] + \theta_2 y[k-1] + \theta_3 u[k] + \theta_4 u[k-1]$$

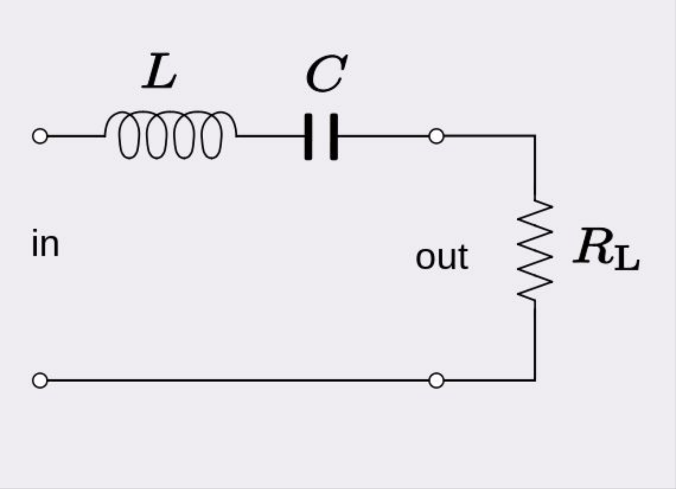
Systems Identification

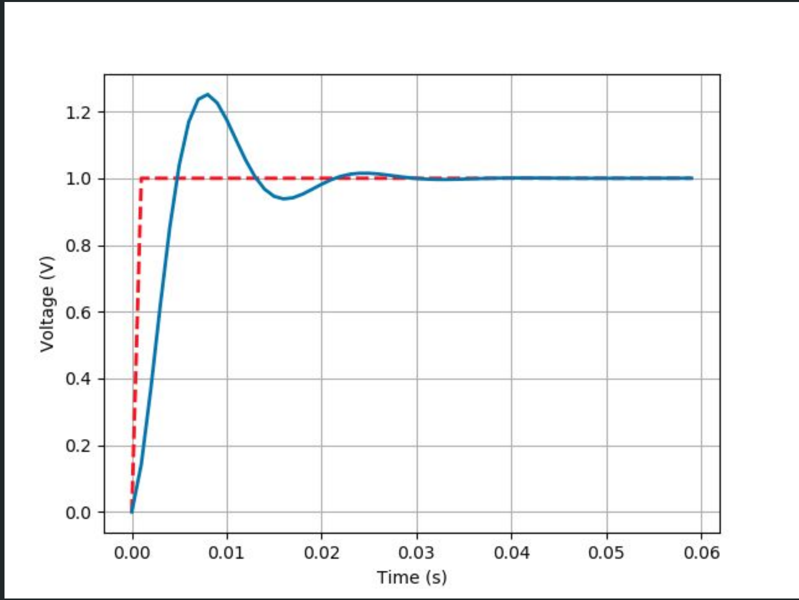
- **Beware!** While higher-orders provide better approximations, modelling them becomes increasingly complex

- What you are approximating in higher orders may not really be your system, but something else with a similar behavior

Systems Identification

- Example: **Identifying the RLC filter model**





R = ?
L = ?
C = ?

Step input (u = 1V $\forall$t)
T = 0.001 s

Systems Identification

- Example: **Identifying the RLC filter model**

$$\frac{Y(s)}{U(s)} = \frac{a}{s+b}$$

$$y[k] = \theta_1 u[k] + \theta_2 y[k-1]$$

First-order

$$\theta = \begin{bmatrix} 0.2477 \\ 0.7612 \end{bmatrix}$$

Systems Identification

- Example: **Identifying the RLC filter model**

$$\frac{Y(s)}{U(s)} = \frac{a}{s+b}$$

$$y[k] = \theta_1 u[k] + \theta_2 y[k-1]$$

First-order

$$\theta = \begin{bmatrix} 0.2477 \\ 0.7612 \end{bmatrix}$$

Second-order

$$\frac{Y(s)}{U(s)} = \frac{w_n^2}{s^2 + 2\xi w_n s + w_n^2}$$

$$y[k] = \theta_1 u[k] + \theta_2 y[k-1] + \theta_3 y[k-2]$$

Systems Identification

- Example: **Identifying the RLC filter model**

$$\frac{Y(s)}{U(s)} = \frac{a}{s+b}$$

$$y[k] = \theta_1 u[k] + \theta_2 y[k-1]$$

First-order

$$\theta = \begin{bmatrix} 0.2477 \\ 0.7612 \end{bmatrix}$$

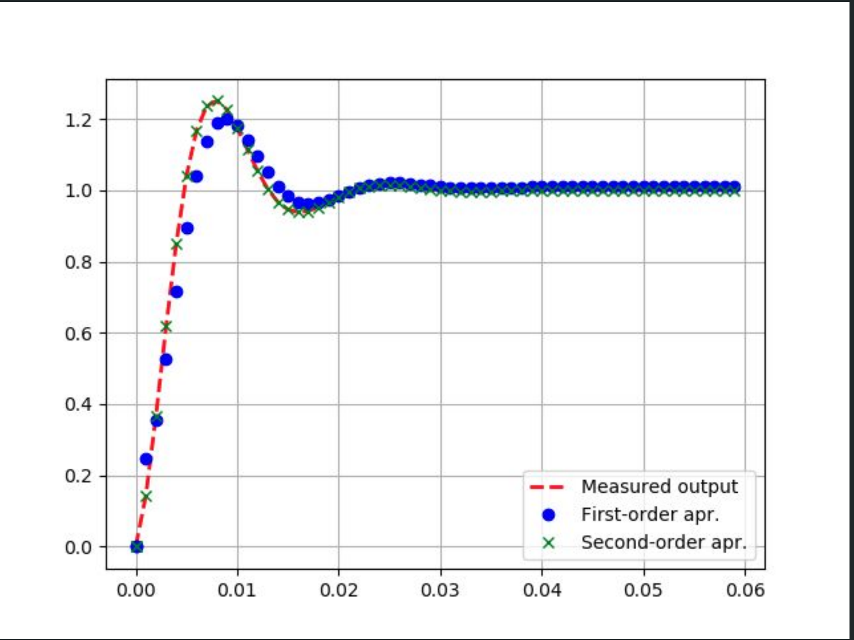
$$\theta = \begin{bmatrix} 0.1428 \\ 1.5714 \\ -0.7143 \end{bmatrix}$$

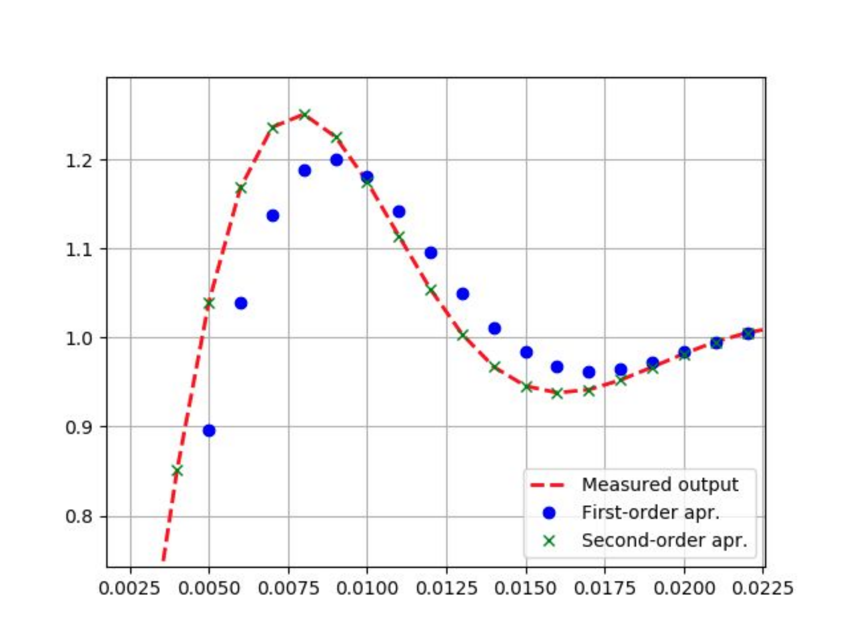
Second-order

$$\frac{Y(s)}{U(s)} = \frac{w_n^2}{s^2 + 2\xi w_n s + w_n^2}$$

$$y[k] = \theta_1 u[k] + \theta_2 y[k-1] + \theta_3 y[k-2]$$

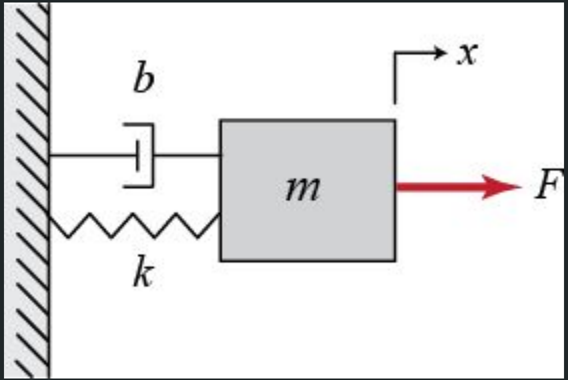
Systems Identification





Systems Identification

- Challenge: **Spring-Mass-Damper system**

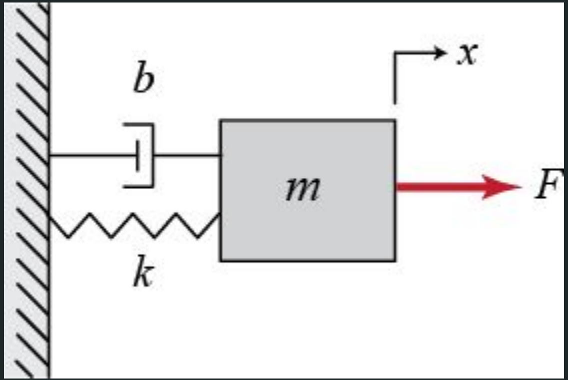


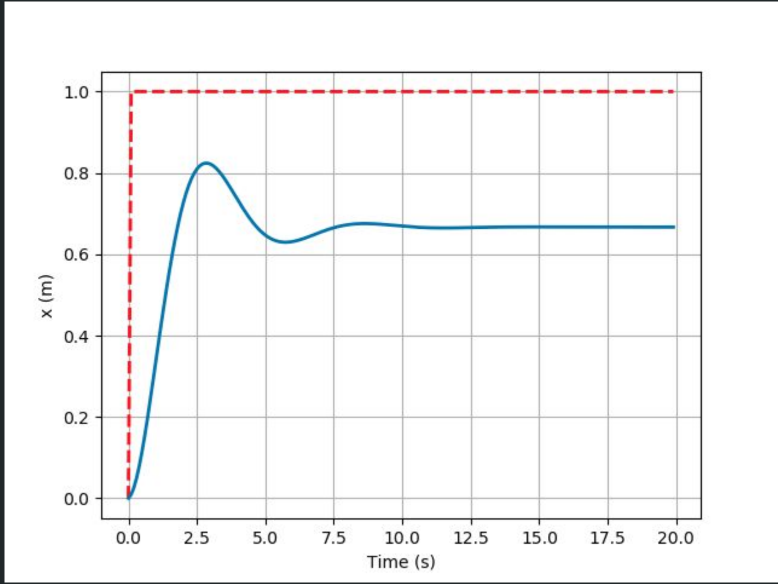
$$m\ddot{x} + b\dot{x} + kx = 0$$

1) **Find the system's transfer function, G(s)**
2) **Discretize G(s)**
3) **Use your discretized model to approximate G(s) through identification**
    a) Tip: Use matlab, octave, python...
4) **Define the previously unknown mass, damping coefficient (b) and spring constant (k)**

Systems Identification

- Challenge: **Spring-Mass-Damper system**





m = ?; b = ?; k = ?

Sampling time T = 0.1 s

- **Data available in Github! (https://github.com/imr-pucrs/didactic_resources)**

State-Spaces

- In modern control theory, we typically use **state-spaces** to represent dynamic systems
    - Robust, optimal and predictive control;
    - Kalman Filters;
    - Etc;

- In a simple interpretation, they are **a set of equations which describe how a system changes in time**, relative to the **states** of the input and output

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

State-Spaces

- Example: **Spring-Mass-Damper system**

$$m\ddot{y} + b\dot{y} + ky = u$$

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \overset{A}{\begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \overset{B}{\begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}} u$$

$$y = \underset{C}{\begin{bmatrix} 1 & 0 \end{bmatrix}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + Du$$

State-Spaces

- To convert from transfer functions to state-spaces, we can use the **canonical forms**

Controllable Form

$$\frac{Y(s)}{U(s)} = \frac{b_0 s^n + b_1 s^{n-1} + \ldots + b_{n-1}s + b_n}{s^n + a_1 s^{n-1} + \ldots + a_{n-1}s + a_n}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \ldots & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} b_n - a_n b_0 & b_{n-1} - a_{n-1} b_0 & \ldots & b_1 - a_1 b_0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + b_0 u$$

State-Spaces

- To convert from transfer functions to state-spaces, we can use the **canonical forms**

$$\frac{Y(s)}{U(s)} = \frac{b_0 s^n + b_1 s^{n-1} + \ldots + b_{n-1}s + b_n}{s^n + a_1 s^{n-1} + \ldots + a_{n-1}s + a_n}$$

Observable Form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 0 & \ldots & 0 & -a_n \\ 1 & 0 & \ldots & 0 & -a_{n-1} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} b_n - a_n b_0 \\ b_{n-1} - a_{n-1} b_0 \\ \vdots \\ b_1 - a_1 b_0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 0 & 0 & \ldots & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + b_0 u$$

$$\begin{aligned} A_{obs} &= A^T_{cont} \\ B_{obs} &= C^T_{cont} \\ C_{obs} &= B^T_{cont} \\ D_{obs} &= D_{cont} \end{aligned}$$

Coffee Break

Relax, take a deep breath