a place of mind

ELEC 341: Systems and Control

Lecture 4

Modeling of electrical & mechanical systems

Water tank level control

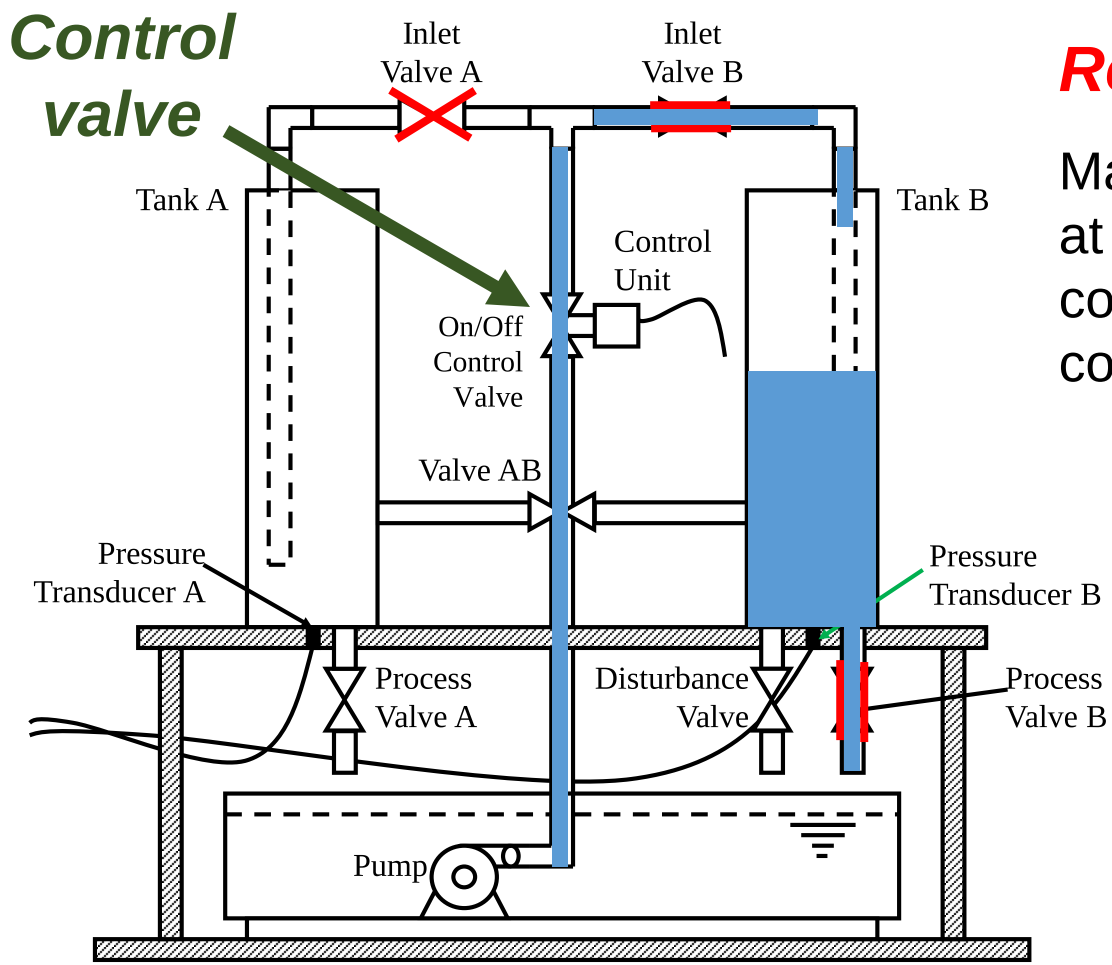
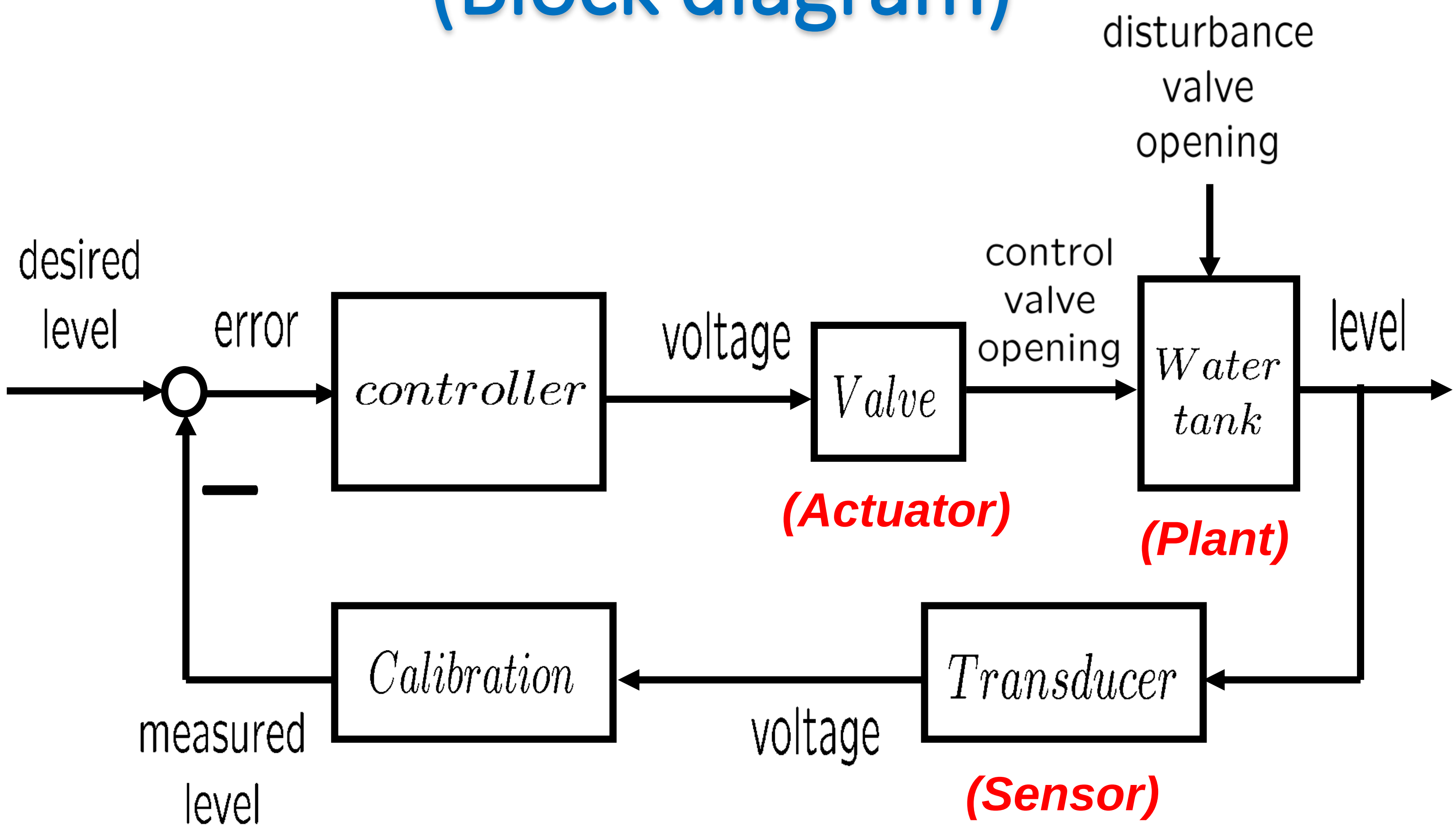


Figure: Schematic of the Tank Level Control Setup.

Requirement:

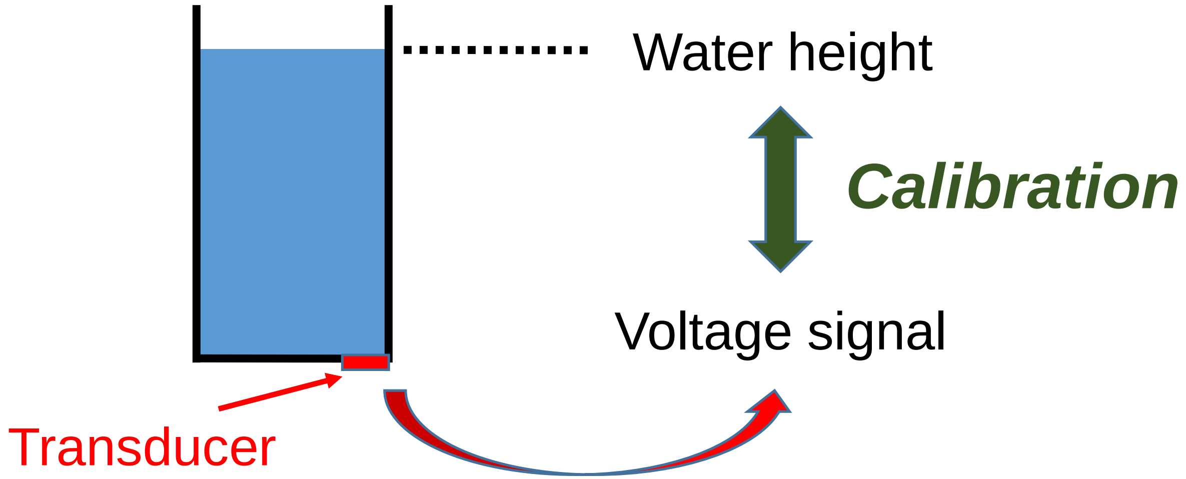
Maintain the level of Tank B at a desired level by controlling the flow through control valve.

Water tank level control (Block diagram)



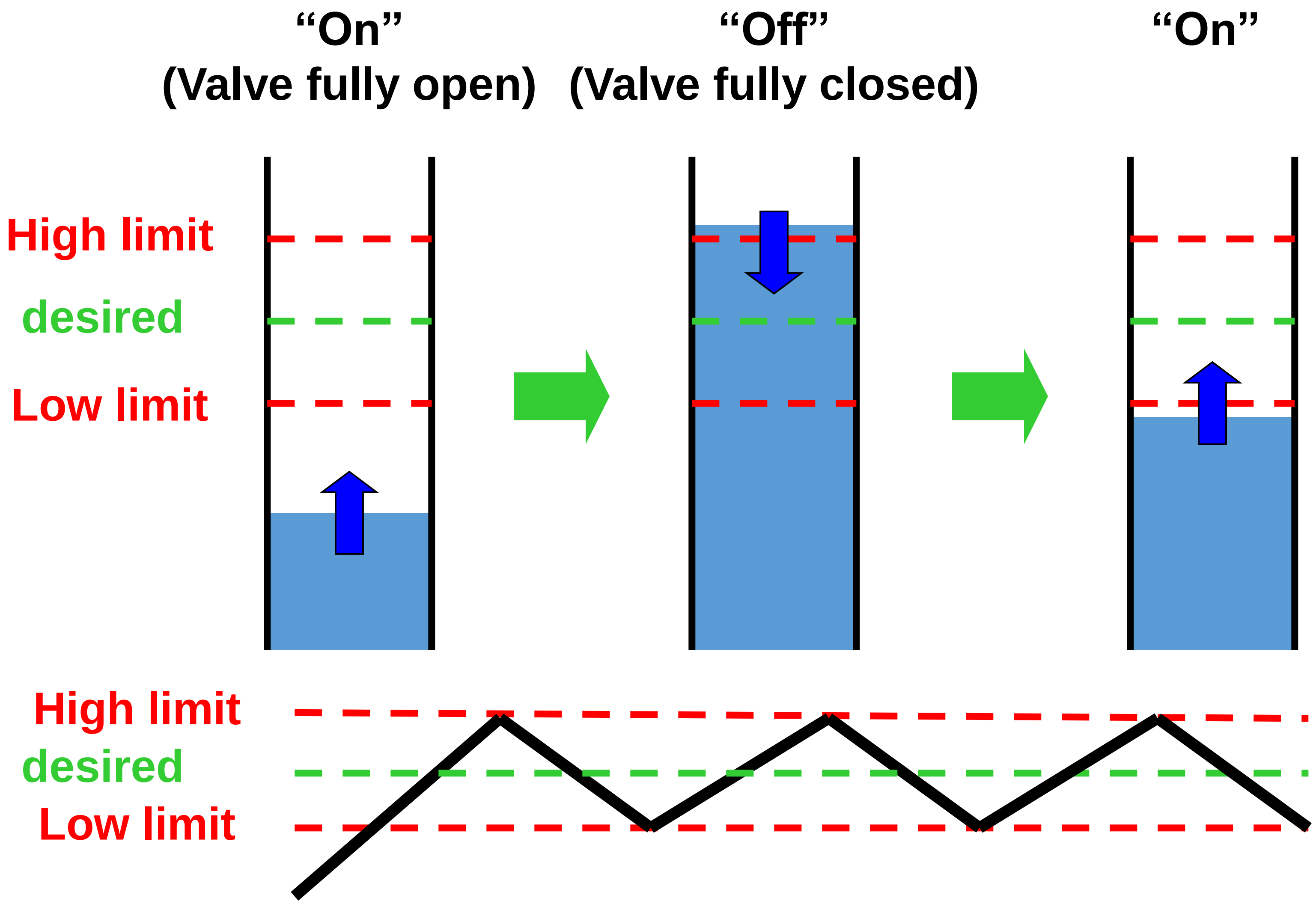
Two main tasks

- Calibration
  - Relate transducer output voltage to actual water level.



- Implementation of the Proportional or ON/OFF controller
  - Analyze the performance of the closed-loop system with a provided ON/OFF controller block.

ON/OFF (bang-bang) control



Remarks on ON/OFF control

- Simplest design control algorithm
- Oscillatory behavior
- Difficult to maintain the level at the desired level.
- Small difference between high and low limits causes the chattering (rapid switching) problem.
- Over-reaction (small change of water level may cause full action of valve). This can be avoided by a proportional control.

Course roadmap

Modeling

- Laplace transform
- Transfer function

Models for systems
- Electrical
- Electromechanical
- Mechanical

Linearization, delay

Analysis

Stability
- Routh-Hurwitz
- Nyquist

Time response
- Transient
- Steady state

Frequency response
- Bode plot

Design

Design specs

Root locus

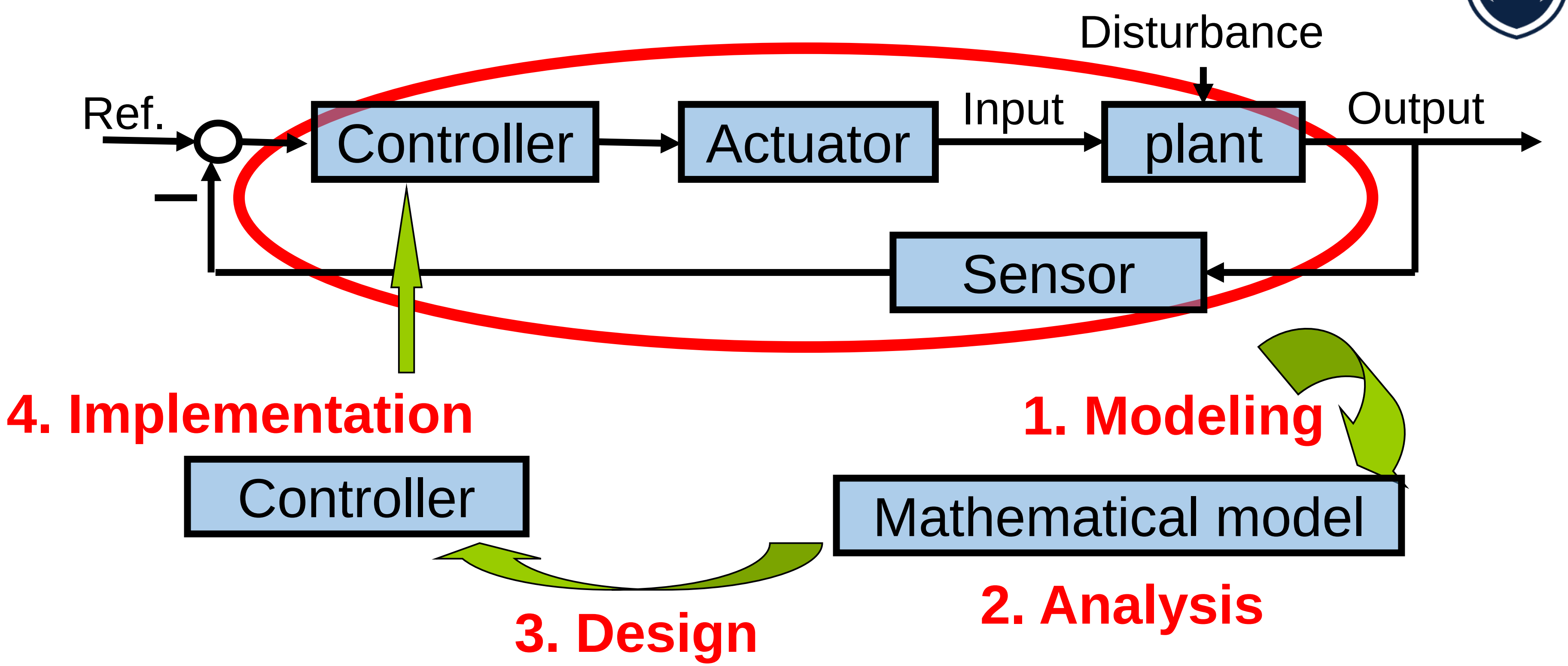
Frequency domain

PID & Lead-lag

Design examples

Matlab simulations

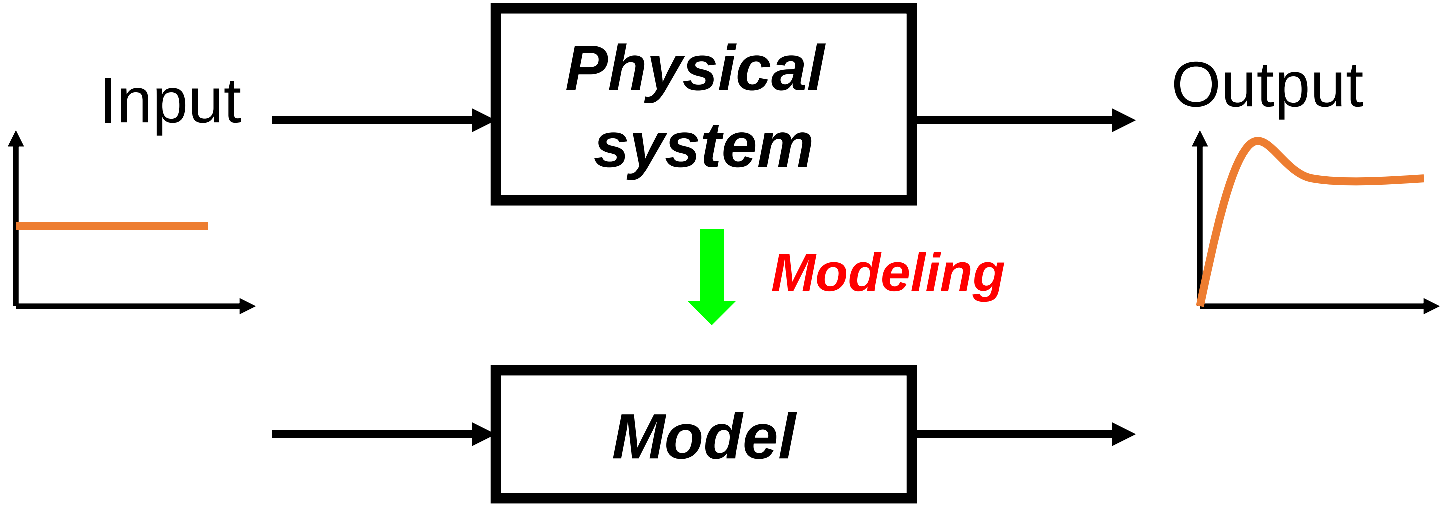
Controller design process (review)



- What is the "mathematical model"?
- Transfer function
- Modeling of electrical & mechanical systems

Mathematical model

- A mathematical model is a representation of the input-output (signal) relation of a physical system:



- A model is used for the **analysis** and **design** of control systems.

Important remarks on models

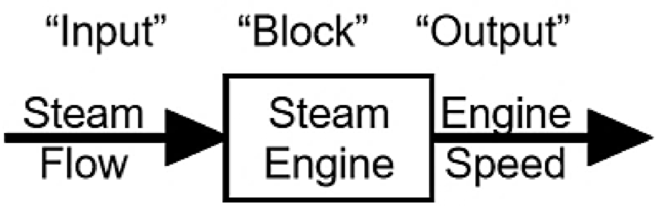
- Modeling is one of the most important and most difficult tasks in control system design.
- No mathematical model exactly represents a physical system.

$$\text{Math model} \neq \text{Physical system}$$
$$\text{Math model} \approx \text{Physical system}$$

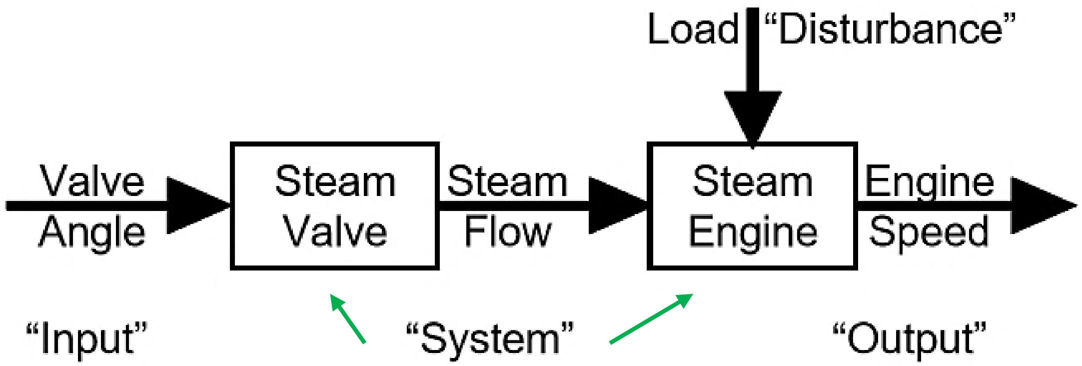
- Do not confuse models with physical systems!

Block diagram

- Communication tool for Engineering Systems
  - Composed of Blocks with inputs and outputs



- Blocks Connect to form systems
  - Outputs of one block becomes input to another



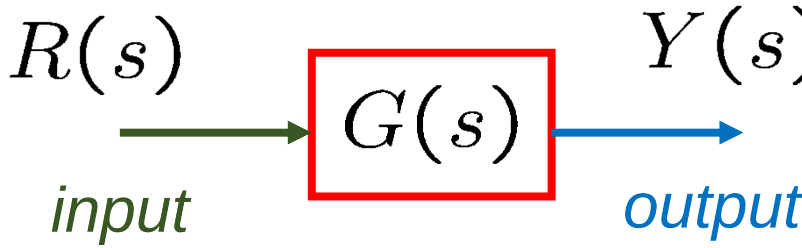
Transfer function

- A transfer function is defined by

$$G(s) = \frac{Y(s)}{R(s)}$$

Laplace transform of system output ← $Y(s)$

Laplace transform of system input ← $R(s)$



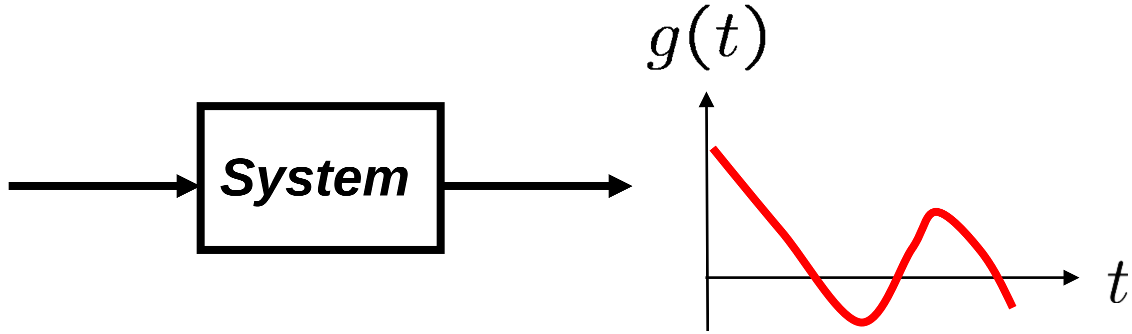
$$Y(s) = G(s)R(s)$$

- Transfer function is a generalization of “gain” concept.

Impulse response

- Suppose that $r(t)$ is the unit impulse function and system is at rest.

$$r(t) = \delta(t) \xrightarrow{\mathcal{L}} R(s) = 1$$



- The output $g(t)$ for the unit impulse input is called *unit impulse response*.
- Since $R(s)=1$, the transfer function can also be defined as the Laplace transform of impulse response:

$$G(s) = \mathcal{L}\{g(t)\}$$

Course roadmap

Modeling

Laplace transform

Transfer function

Models for systems
- Electrical
- Electromechanical
- Mechanical

Linearization, delay

Analysis

Stability
- Routh-Hurwitz
- Nyquist

Time response
- Transient
- Steady state

Frequency response
- Bode plot

Design

Design specs

Root locus

Frequency domain

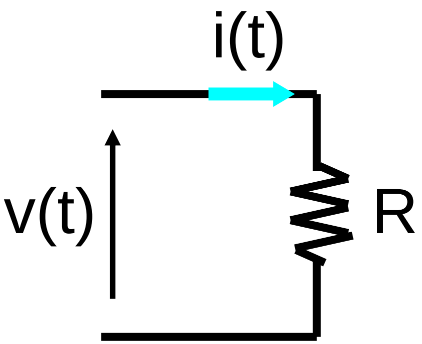
PID & Lead-lag

Design examples

Matlab simulations

Models of electrical elements

Resistance

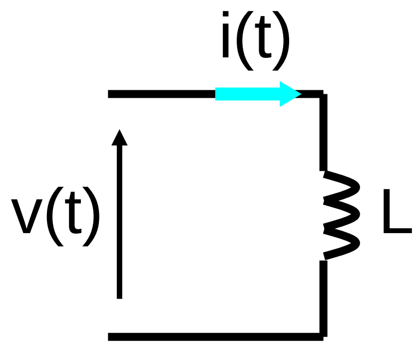


$$v(t) = Ri(t)$$

Laplace transform

$$\frac{V(s)}{I(s)} = \underline{R}$$

Inductance

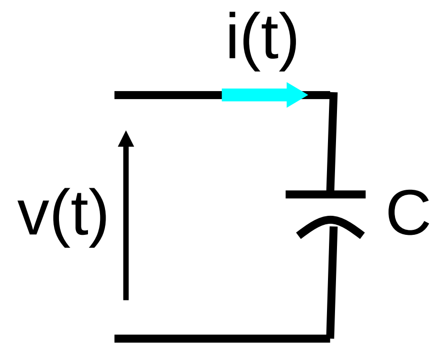


$$v(t) = L\frac{di(t)}{dt}$$

$$(i(0) = 0)$$

$$\frac{V(s)}{I(s)} = \underline{sL}$$

Capacitance



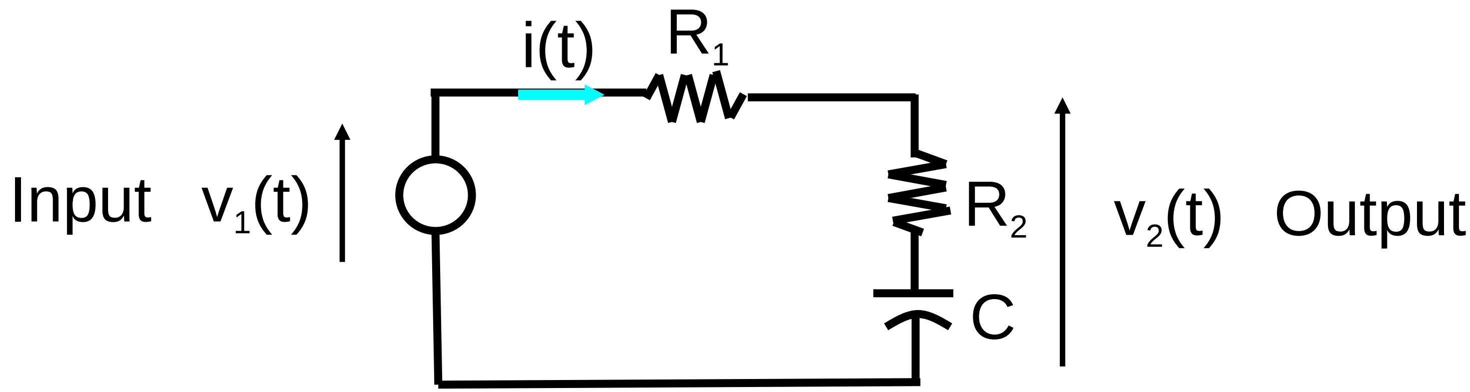
$$i(t) = C\frac{dv(t)}{dt}$$

$$(v(0) = 0)$$

$$\frac{V(s)}{I(s)} = \underline{\frac{1}{sC}}$$

Impedance

Example 1: Modeling



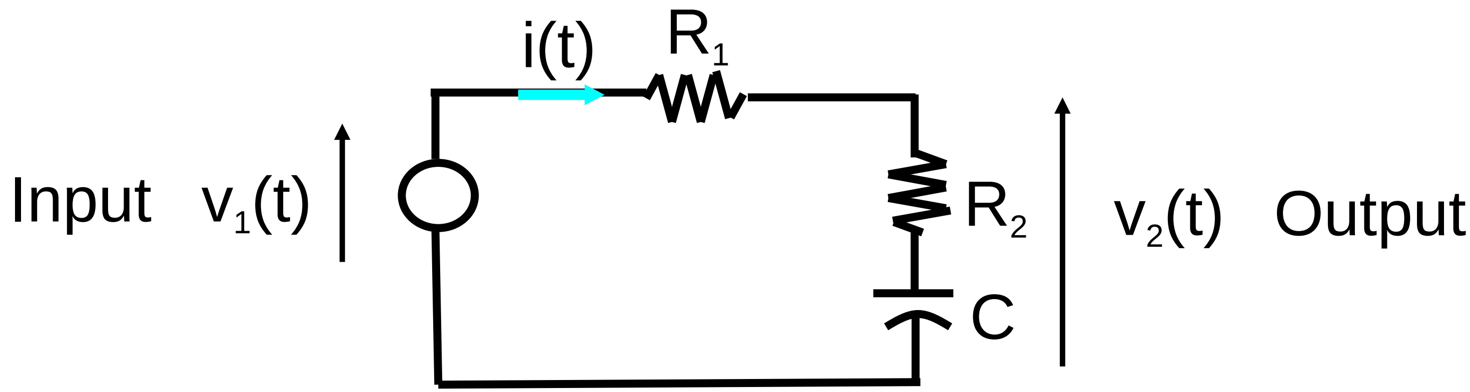
- Kirchhoff voltage law (with zero initial conditions)

$$
\begin{aligned}
v_1(t) &= (R_1 + R_2)i(t) + \frac{1}{C}\int_0^t i(\tau)d\tau \\
v_2(t) &= R_2 i(t) + \frac{1}{C}\int_0^t i(\tau)d\tau
\end{aligned}
$$

- By Laplace transform,

$$
\begin{aligned}
V_1(s) &= (R_1 + R_2)I(s) + \frac{1}{sC}I(s) \\
V_2(s) &= R_2 I(s) + \frac{1}{sC}I(s)
\end{aligned}
$$

Example 1 (cont'd)



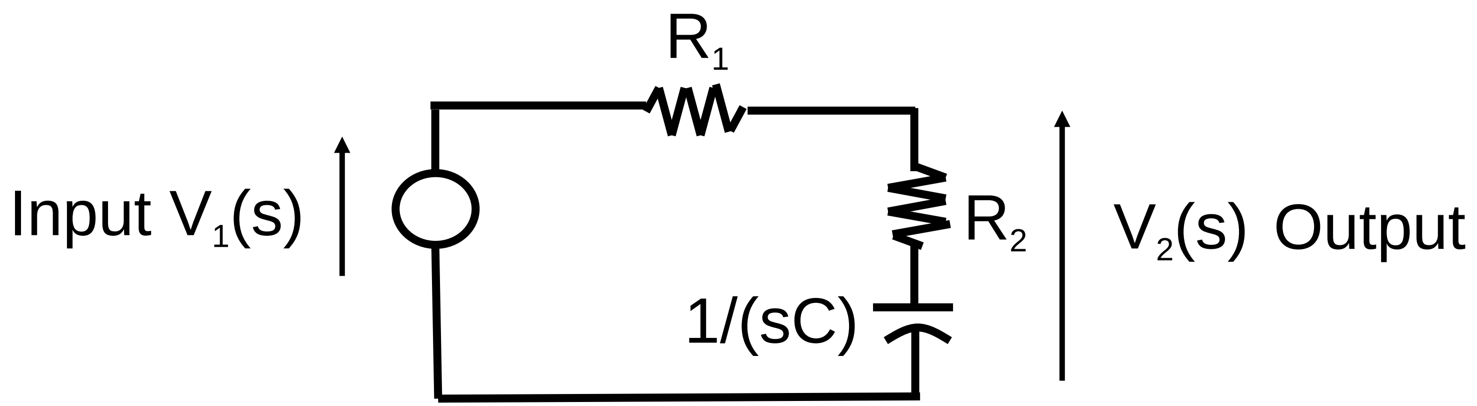
- Transfer function

$$G(s) = \frac{V_2(s)}{V_1(s)} = \frac{R_2 + \frac{1}{sC}}{(R_1+R_2) + \frac{1}{sC}}$$

$$= \frac{R_2Cs+1}{(R_1+R_2)Cs+1} \quad \text{(first-order system)}$$

Example 1 (cont'd)

- **Impedance method**
  - **Step 1:** Replace electrical elements with impedances.
  - **Step 2:** Deal with impedances as if they were resistances.



$$G(s) = \frac{V_2(s)}{V_1(s)} = \frac{\text{(Impedance for output)}}{\text{(Total impedance)}} = \frac{R_2 + \frac{1}{sC}}{R_1 + R_2 + \frac{1}{sC}}$$

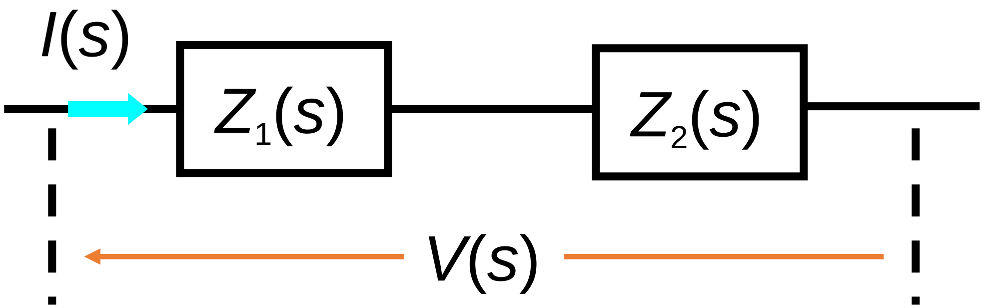
$$= \frac{R_2Cs+1}{(R_1+R_2)Cs+1}$$

Impedance computation

$Z_i(s)$: impedance

- Series connection

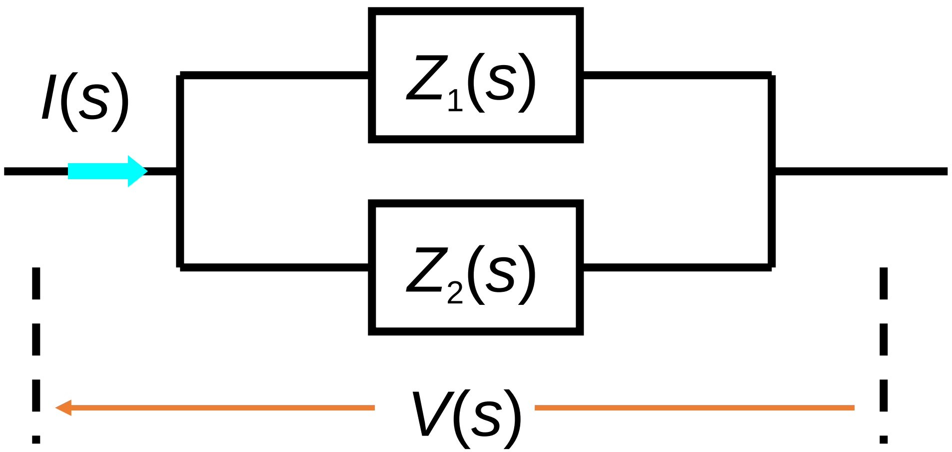
$$Z(s) = Z_1(s) + Z_2(s)$$



- Parallel connection

$$Z(s) = \frac{Z_1(s)Z_2(s)}{Z_1(s) + Z_2(s)}$$

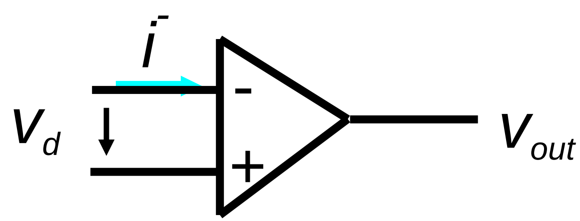
$$\frac{1}{Z(s)} = \frac{1}{Z_1(s)} + \frac{1}{Z_2(s)}$$



Operational amplifier (op-amp)

a place of mind

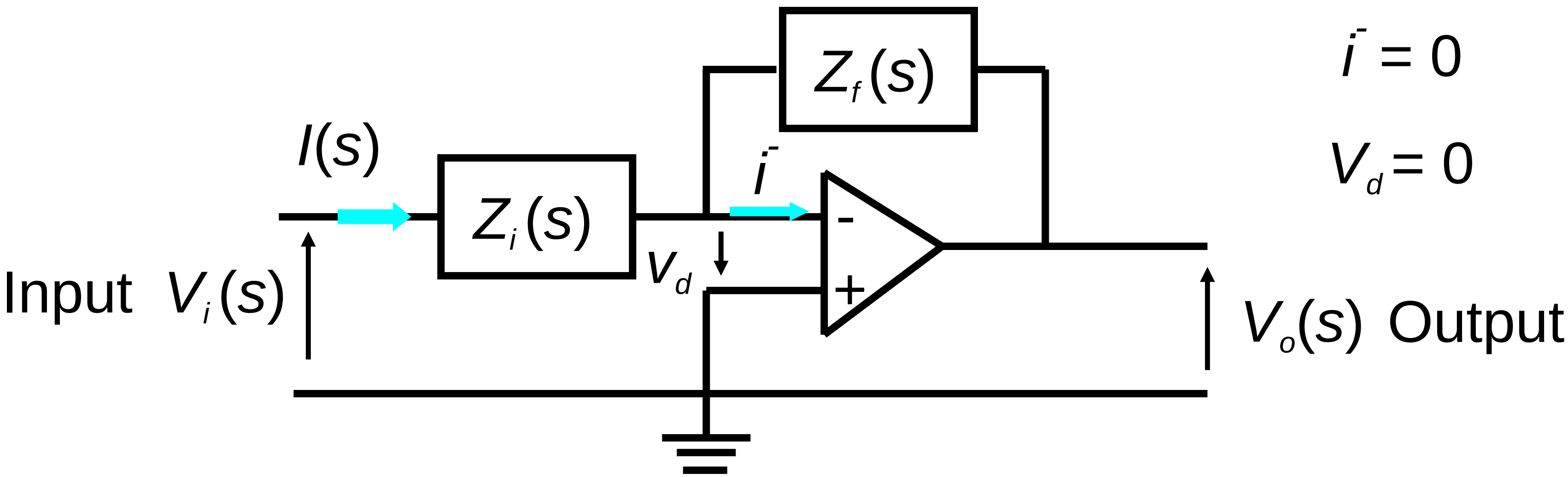
- Electronic voltage amplifier
- Basic building block of analog circuits
- Ideal op-amp (does not exist, but is a good approximation of reality):



$i^- = 0$

$V_d = 0$

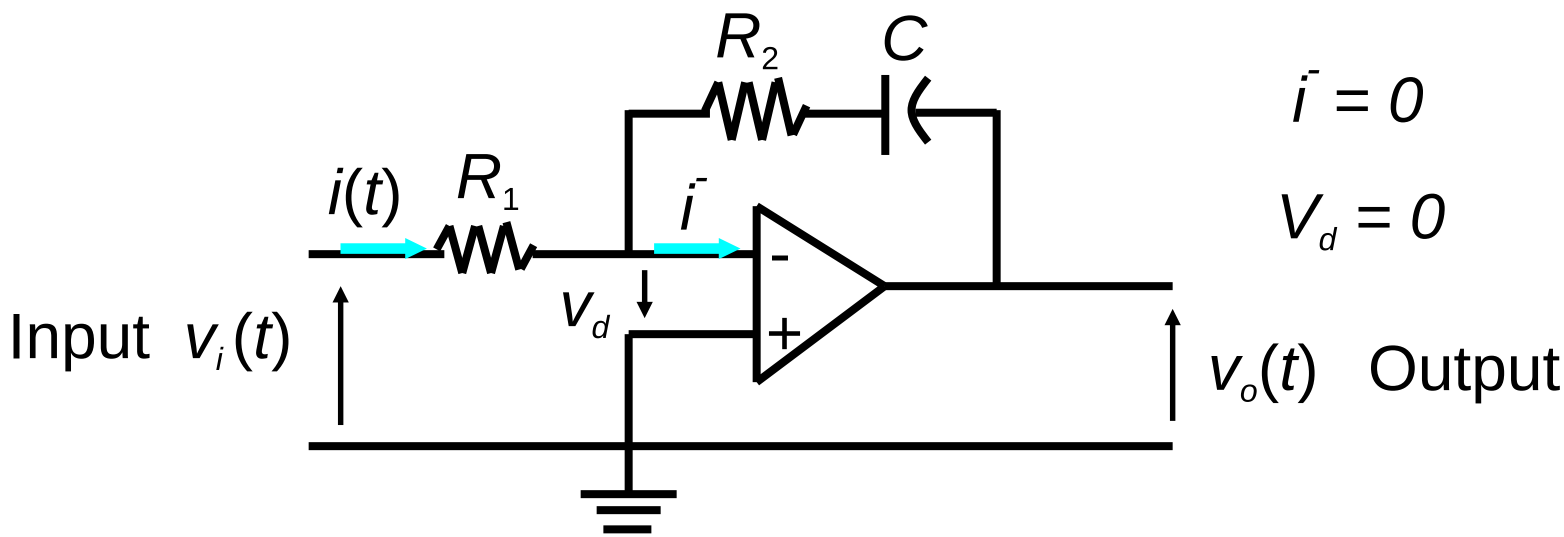
Example 2: Modeling of op-amp



- Impedance $Z(s)$: $V(s) = Z(s)I(s)$
- Transfer function of the above op amp:

$$G(s) = \frac{V_o(s)}{V_i(s)} = \frac{-Z_f(s)I(s)}{Z_i(s)I(s)} = -\frac{Z_f(s)}{Z_i(s)}$$

Example 2 (cont'd)



- By the formula in previous two slides,

$$G(s) = \frac{V_o(s)}{V_i(s)} = \frac{-(R_2 + \frac{1}{sC})}{R_1} = -\frac{R_2 C s + 1}{R_1 C s}$$

(first-order system)

Course roadmap

Modeling

Laplace transform

Transfer function

Models for systems
- Electrical
- Electromechanical
- Mechanical

Linearization, delay

Analysis

Stability
- Routh-Hurwitz
- Nyquist

Time response
- Transient
- Steady state

Frequency response
- Bode plot

Design

Design specs

Root locus

Frequency domain

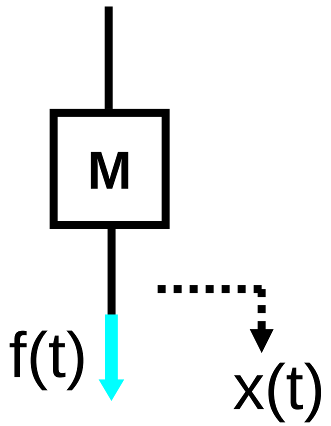
PID & Lead-lag

Design examples

Matlab simulations

Translational mechanical elements

Mass

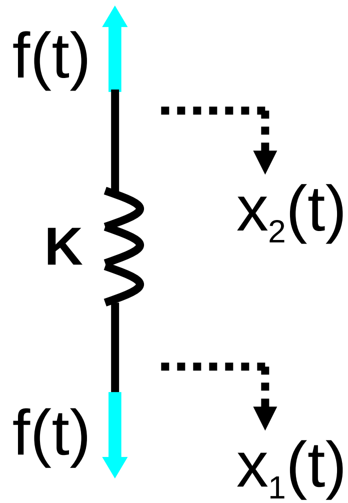


$$f(t) = Mx''(t)$$

$$\Downarrow \left(\begin{array}{l} x(0) = 0 \\ \dot{x}(0) = 0 \end{array} \right)$$

$$F(s) = Ms^2 X(s)$$

Spring

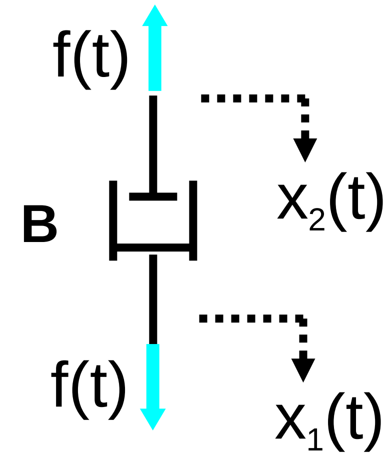


$$f(t) = K(x_1(t) - x_2(t))$$

$$\Downarrow$$

$$F(s) = K(X_1(s) - X_2(s))$$

Damper

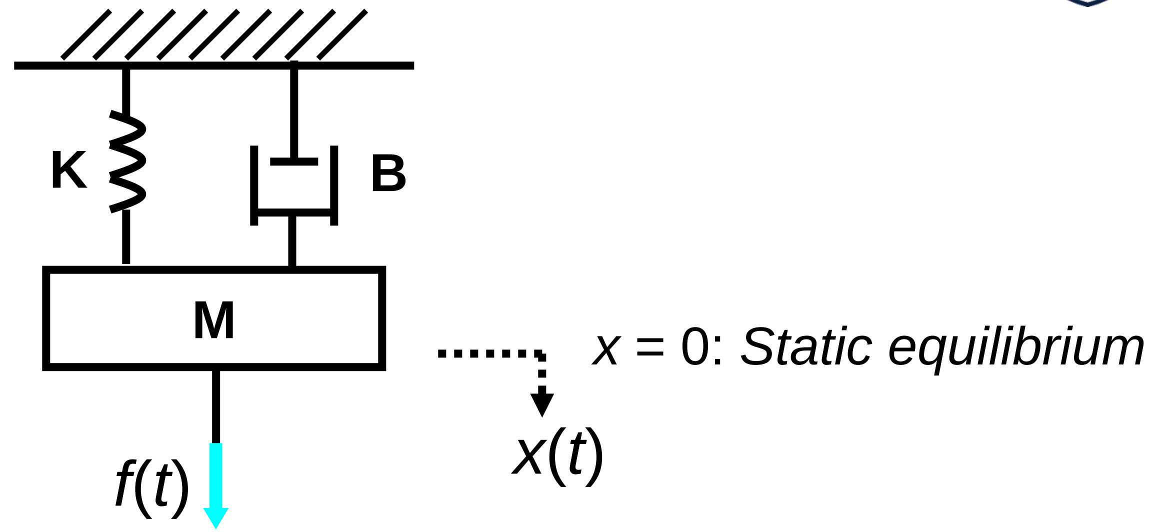


$$f(t) = B(x_1'(t) - x_2'(t))$$

$$\Downarrow \left(\begin{array}{l} x_1(0) = 0 \\ x_2(0) = 0 \end{array} \right)$$

$$F(s) = Bs(X_1(s) - X_2(s))$$

Example 3: Mass-spring-damper system



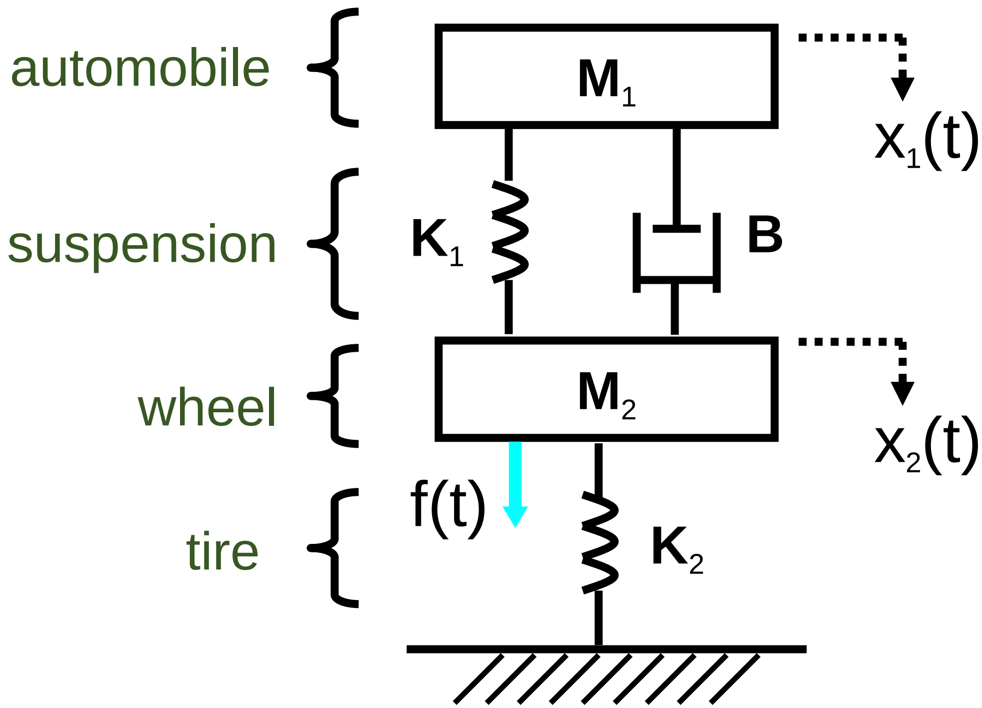
- Equation of motion by Newton's 2nd law

$$Mx''(t) = f(t) - Bx'(t) - Kx(t)$$

- By Laplace transform (with zero initial conditions),

$$\frac{X(s)}{F(s)} = \frac{1}{Ms^2 + Bs + K} \quad \textit{(2nd order system)}$$

Example 4: Automobile suspension system



- Equations of motion by Newton's 2nd law

$$\begin{cases} M_1 x_1''(t) &= -B(x_1'(t) - x_2'(t)) - K_1(x_1(t) - x_2(t)) \\ M_2 x_2''(t) &= f(t) - B(x_2'(t) - x_1'(t)) - K_1(x_2(t) - x_1(t)) - K_2 x_2(t) \end{cases}$$

Example 4 (cont'd)

$$\begin{cases} M_1 x_1''(t) &= -B(x_1'(t) - x_2'(t)) - K_1(x_1(t) - x_2(t)) \\ M_2 x_2''(t) &= f(t) - B(x_2'(t) - x_1'(t)) - K_1(x_2(t) - x_1(t)) - K_2 x_2(t) \end{cases}$$

Laplace transform with zero ICs

$$\begin{cases} M_1 s^2 X_1(s) &= -B(sX_1(s) - sX_2(s)) - K_1(X_1(s) - X_2(s)) \\ M_2 s^2 X_2(s) &= F(s) - B(sX_2(s) - sX_1(s)) - K_1(X_2(s) - X_1(s)) - K_2 X_2(s) \end{cases}$$

$$\begin{cases} X_1(s) &= \underbrace{\frac{\overbrace{Bs + K_1}}{M_1 s^2 + Bs + K_1}}_{G_1(s)} X_2(s) \\ X_2(s) &= \underbrace{\frac{\overbrace{1}}{M_2 s^2 + Bs + K_1 + K_2}}_{G_2(s)} F(s) + \underbrace{\frac{\overbrace{Bs + K_1}}{M_2 s^2 + Bs + K_1 + K_2}}_{G_3(s)} X_1(s) \end{cases}$$

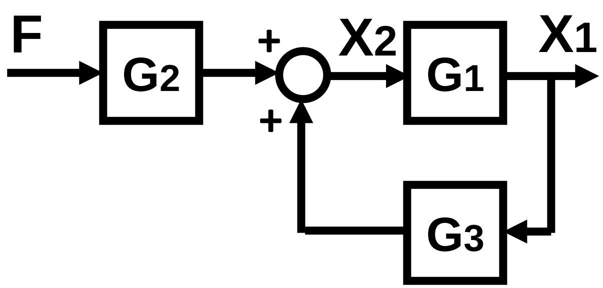
Make transfer functions so that

$$\deg(\text{den}) \geq \deg(\text{num})$$

Note: To save space, from now on, I will not show the top curly bracket.

Example 4 (cont'd)

$$\begin{cases} X_1(s) = \underbrace{\dfrac{Bs+K_1}{M_1s^2+Bs+K_1}}_{G_1(s)} X_2(s) \\ X_2(s) = \underbrace{\dfrac{1}{M_2s^2+Bs+K_1+K_2}}_{G_2(s)} F(s) + \underbrace{\dfrac{Bs+K_1}{M_2s^2+Bs+K_1+K_2}}_{G_3(s)} X_1(s) \end{cases}$$



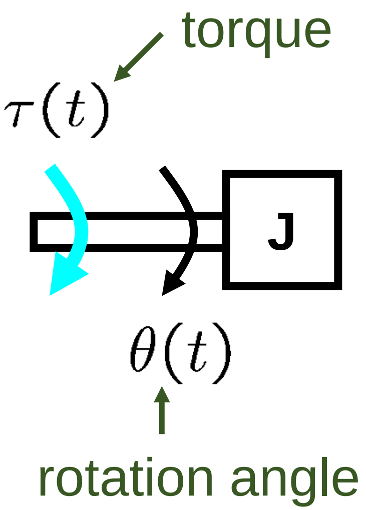
Block diagram

$$\frac{X_1(s)}{F(s)} = \frac{G_1(s)G_2(s)}{1-G_1(s)G_3(s)}$$

We will study how to derive this transfer function in the next lecture.

Rotational mechanical elements

Moment of inertia

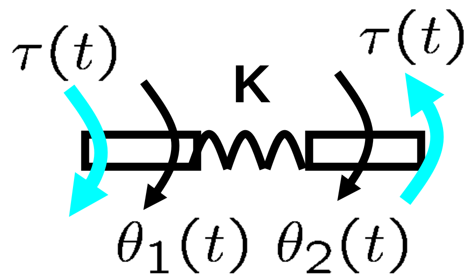


$$\tau(t) = J\theta''(t)$$

$$\Downarrow \quad \begin{pmatrix} \theta(0) = 0 \\ \dot{\theta}(0) = 0 \end{pmatrix}$$

$$T(s) = Js^2\Theta(s)$$

Rotational spring

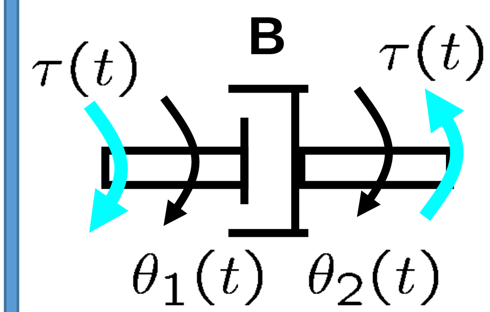


$$\tau(t) = K(\theta_1(t) - \theta_2(t))$$

$$\Downarrow$$

$$T(s) = K(\Theta_1(s) - \Theta_2(s))$$

Friction

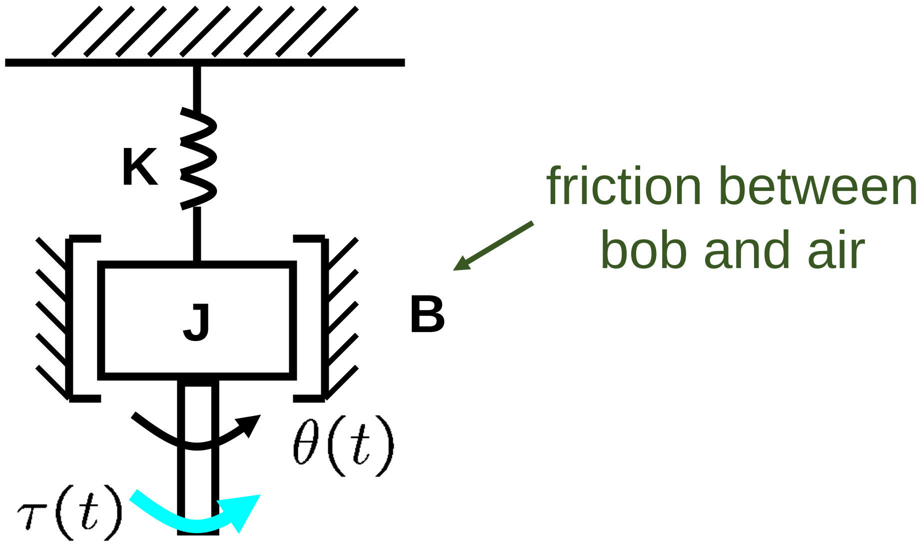


$$\tau(t) = B(\theta_1'(t) - \theta_2'(t))$$

$$\Downarrow \quad \begin{pmatrix} \theta_1(0) = 0 \\ \theta_2(0) = 0 \end{pmatrix}$$

$$T(s) = Bs(\Theta_1(s) - \Theta_2(s))$$

Example 5: Torsional pendulum system



- Equation of motion by Newton's law

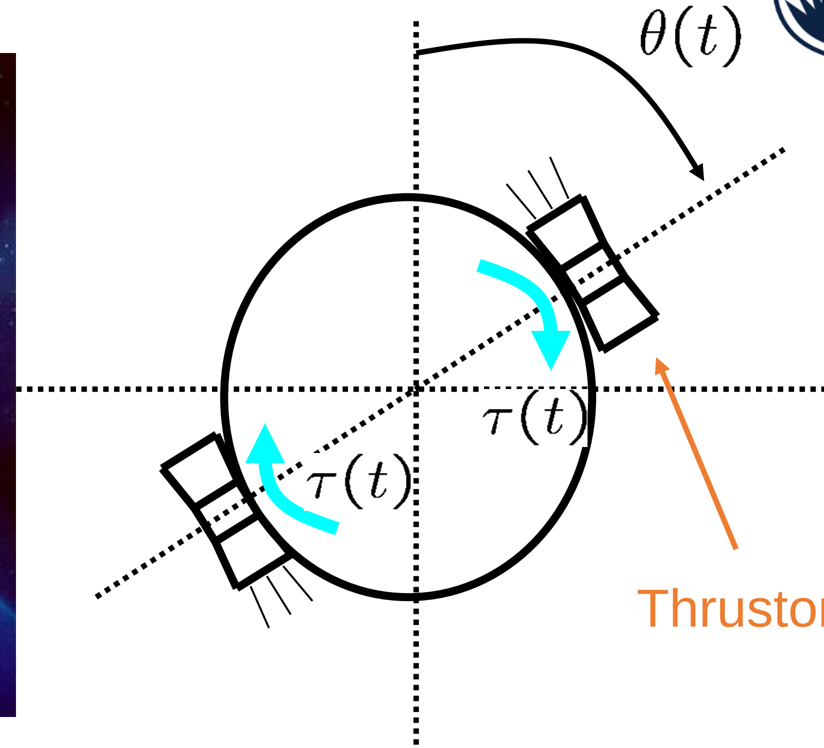
$$J\theta''(t) = \tau(t) - B\theta'(t) - K\theta(t)$$

- By Laplace transform (with zero ICs),

$$G(s) = \frac{\Theta(s)}{T(s)} = \frac{1}{Js^2 + Bs + K}$$

(2^{nd} order system)

Rigid satellite



- Broadcasting
- Weather forecast
- Communication
- GPS, etc.

$$\tau(t) = J\ddot{\theta}(t)$$

$$\Rightarrow \quad G(s) = \frac{\Theta(s)}{T(s)} = \frac{1}{Js^2}$$

Double integrator

State Space Modeling

- Two approaches are available for the analysis and design of feedback control systems.
- The first is known as the **classical,** or **frequency-domain**, technique.
- This approach is based on converting a system's differential equation to a transfer function, thus generating a mathematical model of the system that algebraically relates a representation of the output to a representation of the input.
- Replacing a differential equation with an algebraic equation not only simplifies the representation of individual subsystems but also simplifies modeling interconnected subsystems.
- The primary disadvantage of the classical approach is its limited applicability: It can be applied only to linear, time-invariant systems or systems that can be approximated as such.
- A major advantage of frequency-domain techniques is that they rapidly provide stability and transient response information.
- Thus, we can immediately see the effects of varying system parameters until an acceptable design is met.

State Space Modeling

a place of mind UBC

- With the arrival of space exploration, requirements for control systems increased in scope.
- The **state-space** approach (also referred to as the **modern**, or **time-domain**, approach) is a **unified method** for modeling, analyzing, and designing a wide range of systems.
- Time-varying systems, (for example, missiles with varying fuel levels or lift in an aircraft flying through a wide range of altitudes) can be represented in state space.
- Many systems do not have just a single input and a single output. Multiple-input, multiple-output systems can be compactly represented in state space with a model similar in form and complexity to that used for single-input, single-output systems.
- The state space approach is also attractive because of the availability of numerous state-space software packages for the personal computer.
- While the state-space approach can be applied to a wide range of systems, it is not as intuitive as the classical approach.
- The designer has to engage in several calculations before the physical interpretation of the model is apparent, whereas in classical control a few quick calculations or a graphic presentation of data rapidly yields the physical interpretation.

a place of mind UBC

State Space Modeling

1. We select a particular *subset* of all possible system variables and call the variables in this subset *state variables*.
2. For an nth-order system, we write *n simultaneous, first-order differential equations* in terms of the state variables. We call this system of simultaneous differential equations *state equations*.
3. If we know the initial condition of all of the state variables at t_0 as well as the system input for $t \geq t_0$, we can solve the simultaneous differential equations for the state variables for $t \geq t_0$.
4. We *algebraically* combine the state variables with the system's input and find all of the other system variables for $t \geq t_0$. We call this algebraic equation the *output equation*.
5. We consider the state equations and the output equations a viable representation of the system. We call this representation of the system a *state-space representation*.

State Space Modeling

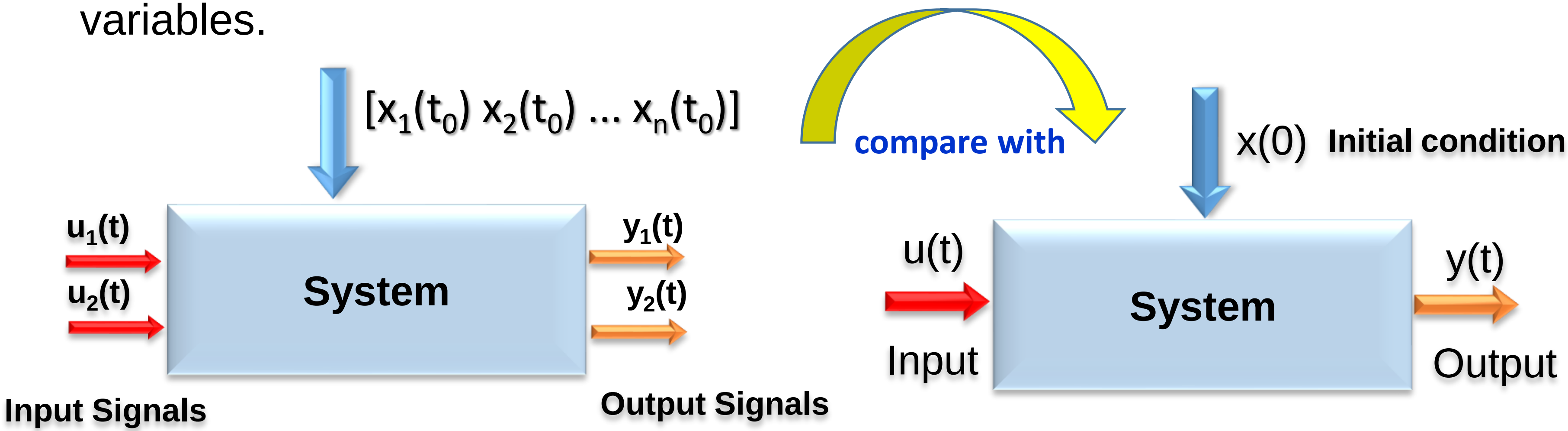
State Variables of a Dynamic System:

- The time-domain analysis and design of control systems utilizes the concept of the state of a system.
- The state of a system is a set of variables such that the knowledge of these variables and the input functions will, with the equations describing the dynamics, provide the future state and output of the system.
- For a dynamic system, the state of a system is described in terms of a set of state variables

$$[x_1(t) \quad x_2(t) \quad \ldots \quad x_n(t)]$$

State Space Modeling

- The state variables are those variables that determine the future behavior of a system when the present state of the system and the excitation signals are known.
- Consider the system shown below, where $y_1(t)$ and $y_2(t)$ are the output signals and $u_1(t)$ and $u_2(t)$ are the input signals. A set of state variables $[x_1\ x_2\ \ldots\ x_n]$ for the system shown in the figure is a set such that knowledge of the initial values of the state variables $[x_1(t_0)\ x_2(t_0)\ \ldots\ x_n(t_0)]$ at the initial time t_0, and of the input signals $u_1(t)$ and $u_2(t)$ for $t \geq t_0$, suffices to determine the future values of the outputs and state variables.



State Space Modeling

State Differential Equation:

- The state of a system is described by the set of **first-order** differential equations written in terms of the state variables $[x_1 \ x_2 \ \ldots \ x_n]$. These first-order differential equations can be written in general form as:

$$\dot{x}_1 = a_{11}x_1 + a_{12}x_2 + \ldots a_{1n}x_n + b_{11}u_1 + \cdots b_{1m}u_m$$

$$\dot{x}_2 = a_{21}x_1 + a_{22}x_2 + \ldots a_{2n}x_n + b_{21}u_1 + \cdots b_{2m}u_m$$

$$\vdots$$

$$\dot{x}_n = a_{n1}x_1 + a_{n2}x_2 + \ldots a_{nn}x_n + b_{n1}u_1 + \cdots b_{nm}u_m$$

State Space Modeling

a place of mind

- Thus, this set of simultaneous differential equations can be written in matrix form as follows:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \cdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & \cdots & \vdots \\ b_{n1} & \cdots & b_{nm} \end{bmatrix}\begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix}$$

n: number of state variables, *m*: number of inputs.

- The column matrix consisting of the state variables is called the **state vector** and is written as:

$$\mathrm{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

State Space Modeling

- The vector of input signals is defined as u. Then the system can be represented by the compact notation of the state differential equation as:

$$\dot{x} = A\,x + B\,u$$

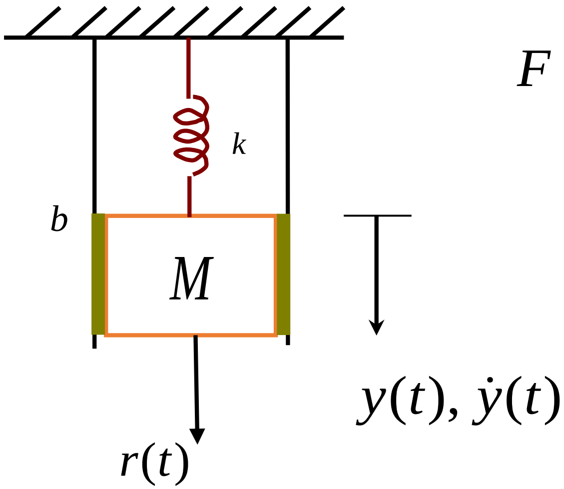
- This differential equation is also commonly called the **state equation**. The matrix **A** (the system matrix) is an $n \times n$ square matrix, and **B** (the input matrix) is an $n \times m$ matrix. The state differential equation relates the rate of change of the state of the system to the state of the system (i.e., x) and the input signals (i.e., u). In general, the outputs of a linear system can be related to the state variables and the input signals by the **output equation**:

$$y = C\,x + D\,u$$

- Where **y** is the set of output signals expressed in column vector form, **C** is the output matrix, and **D** is the feed-forward matrix. The **state-space representation** (or **state-variable representation**) is comprised of the state variable differential equation and the output equation.

Example 6

By Newton's Law



$$F = M\ddot{y} \rightarrow r - ky - b\dot{y} = M\ddot{y} \quad \Rightarrow M\ddot{y} + b\dot{y} + ky = r$$

let $\boxed{x_1 = y, \;\; x_2 = \dot{y}}$

$$\Rightarrow \begin{cases} \dot{x}_1 = \dot{y} = x_2 \\ \dot{x}_2 = \ddot{y} = -\dfrac{b}{M}\dot{y} - \dfrac{k}{M}y + \dfrac{1}{M}r \end{cases}$$

$$= -\frac{b}{M}x_2 - \frac{k}{M}x_1 + \frac{1}{M}u \qquad (u = r)$$

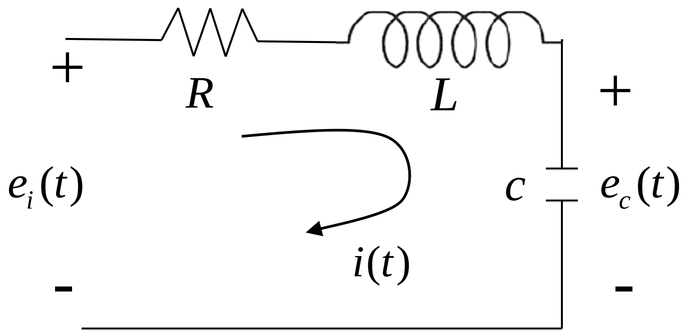
$$\Rightarrow \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\dfrac{k}{M}x_1 - \dfrac{b}{M}x_2 + \dfrac{1}{M}u \end{cases}$$

$$\dot{\mathrm{x}} = \mathrm{A\,x + B\,u}$$
$$\mathrm{y = C\,x + D\,u}$$

$$\Rightarrow \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\dfrac{k}{M} & -\dfrac{b}{M} \end{bmatrix}}_{A} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ \dfrac{1}{M} \end{bmatrix}}_{B} \cdot u \qquad y = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{0}_{D} \cdot u$$

Example 7

Remark: The choice of states is not unique and one can also have multiple outputs.



$$Ri(t) + L\frac{di(t)}{dt} + \frac{1}{c}\int_0^t i(t)dt = e_i(t)$$

(a)

let $\begin{cases} x_1(t) = i(t) \\ x_2(t) = \int i(t)dt \\ y(t) = i(t) \end{cases}$ $\Rightarrow$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{1}{LC} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} e_i(t)$$

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

(b)

let $\begin{cases} x_1(t) = e_c(t) \\ x_2(t) = i(t) \\ y_1(t) = e_c(t) \\ y_2(t) = e_R(t) = Ri \end{cases}$ $\Rightarrow$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{C} \\ -\frac{1}{LC} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} e_i(t)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Obtain transfer function from the state equation

Dynamical equation $\Rightarrow$ Transfer function

$$\left.\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned}\right\}$$ Dynamical equation

Laplace transform

$$sX(s) - x(0) = AX(s) + BU(s)$$
$$Y(s) = CX(s) + DU(s)$$

$assume \quad x(0) = 0$

$$X(s) = (sI - A)^{-1} BU(s)$$
$$Y(s) = [C(sI - A)^{-1} B + D]U(s) \rightarrow$$

Transfer function

$$\boxed{T(s) = \frac{Y(s)}{U(s)} = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}}$$

Transfer function

A, B, C, D, and I are all matrices.

Example 8: (State-space representation to transfer function)

Given the system defined by the following equations, find the transfer function, $T(s) = Y(s)/U(s)$ where $U(s)$ is the input and $Y(s)$ is the output.

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -2 & -3 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}\mathbf{x}$$

Solution:

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

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$$

Compare with the above two equations

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -2 & -3 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{D} = 0$$

$$(s\mathbf{I} - \mathbf{A}) = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -2 & -3 \end{bmatrix} = \begin{bmatrix} s & -1 & 0 \\ 0 & s & -1 \\ 1 & 2 & s+3 \end{bmatrix}$$

$$(s\mathbf{I} - \mathbf{A})^{-1} = \frac{\operatorname{adj}(s\mathbf{I} - \mathbf{A})}{\det(s\mathbf{I} - \mathbf{A})} = \frac{\begin{bmatrix} (s^2+3s+2) & s+3 & 1 \\ -1 & s(s+3) & s \\ -s & -(2s+1) & s^2 \end{bmatrix}}{s^3 + 3s^2 + 2s + 1}$$

$$\boxed{T(s) = \frac{Y(s)}{U(s)} = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}}$$

$$T(s) = \frac{10(s^2 + 3s + 2)}{s^3 + 3s^2 + 2s + 1}$$

Signal Flow Graphs (Introduction)

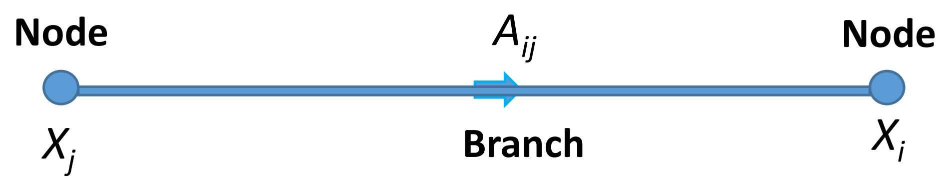
- Signal-flow graphs are an alternative to block diagrams.
- Signal flow graphs are a pictorial representation of the simultaneous equations describing a system.
- These graphs display the transmission of signals through the system, as does the block diagrams.
- Unlike block diagrams, which consist of blocks, signals, summing junctions, and pickoff points, a signal-flow graph consists only of branches, which represent systems, and nodes, which represent signals.

Fundamentals of Signal Flow Graphs

- Consider a simple equation below and let us draw its signal flow graph:

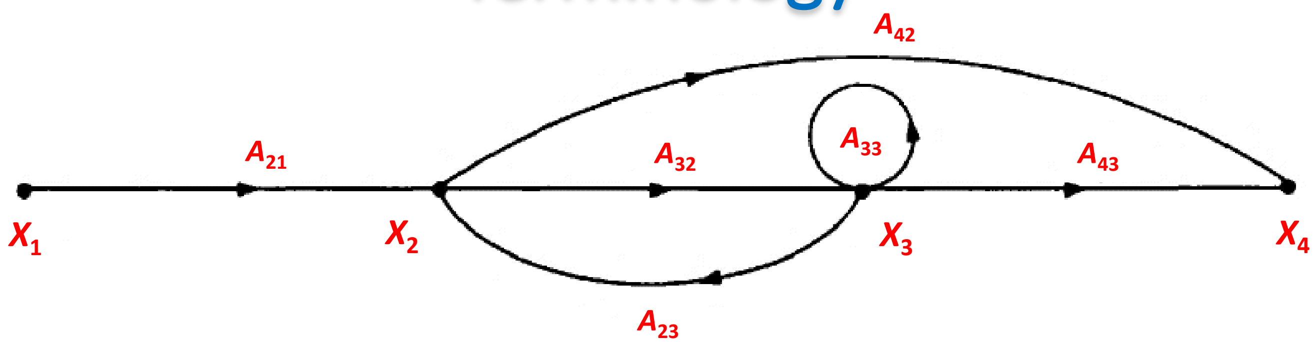
$$X_i = A_{ij}X_j$$

- The signal flow graph of the equation is shown below:



- Every variable in a signal flow graph is designed by a **Node**.
- Every transmission function in a signal flow graph is designed by a **Branch**.
- Branches are always **unidirectional**.
- The arrow in the branch denotes the **direction** of the signal flow.
- The variables X_i and X_j are represented by a small dot or circle called a **Node.**
- The **transmission function** A_{ij} is represented by a line with an arrow and placed on the line (i.e., on the branch).
- The node X_j is called **input node** and node X_i is called **output node**.

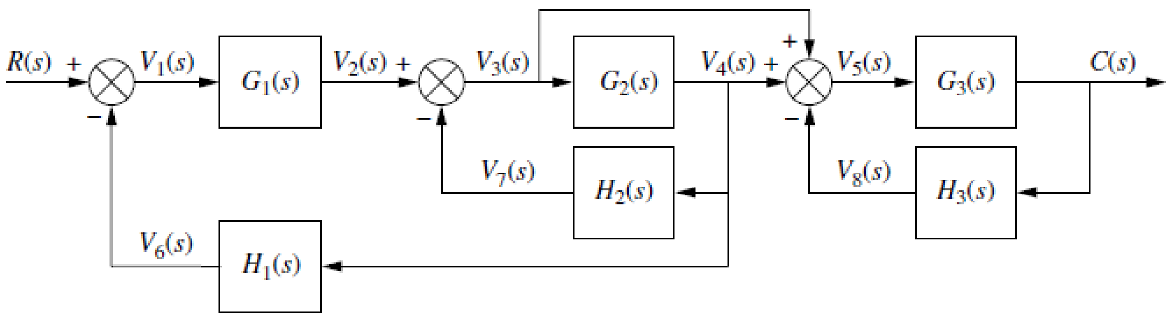
Terminology



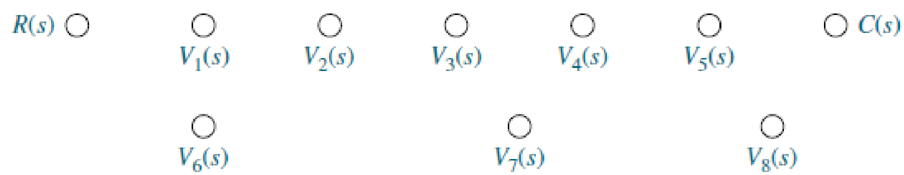
- An **input node** or **source** contains only the outgoing branches, i.e., X_1.
- An **output node** or **sink** contains only the incoming branches, i.e., X_4.
- A **path** is a continuous, unidirectional succession of branches along which no node is passed more than once, i.e., X_1 to X_2 to X_3 to X_4, also X_2 to X_3 to X_4, and X_1 to X_2 to X_4, are all paths.
- A **forward path** is a path from the input node to the output node, i.e., X_1 to X_2 to X_3 to X_4, and X_1 to X_2 to X_4, are forward paths.
- A **feedback path** or feedback loop is a path which originates and terminates on the same node, i.e., X_2 to X_3 and back to X_2 is a feedback path.
- A **self-loop** is a feedback loop consisting of a single branch, e.g., A_{33} is a self loop.
- The **gain** of a branch is the transmission function of that branch when the transmission function is a multiplicative operator, e.g., A_{33}.
- The **path gain** is the product of branch gains encountered in traversing a path, e.g., X_1 to X_2 to X_3 to X_4 is $A_{21}A_{32}A_{43}$.
- The **loop gain** is the product of the branch gains of the loop, e.g., the loop gain of the feedback loop from X_2 to X_3 and back to X_2 is $A_{32}A_{23}$.

Example 9:
Converting Feedback System Block Diagram into a Signal Flow Graph

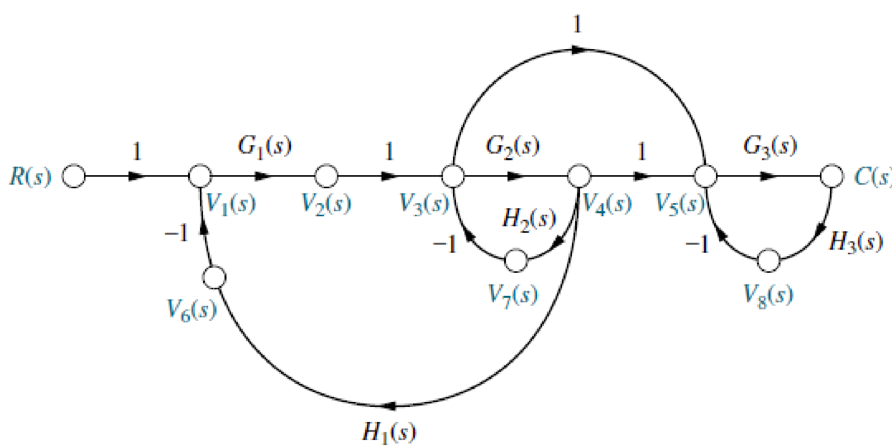
- First thing is to draw the signal nodes for the system. The signal nodes for the given system are shown in Figure **(a)**.
- Next thing is to interconnect the signal nodes with system branches. The interconnection of the nodes with branches that represent the subsystem is shown in Figure **(b)**.



(a)



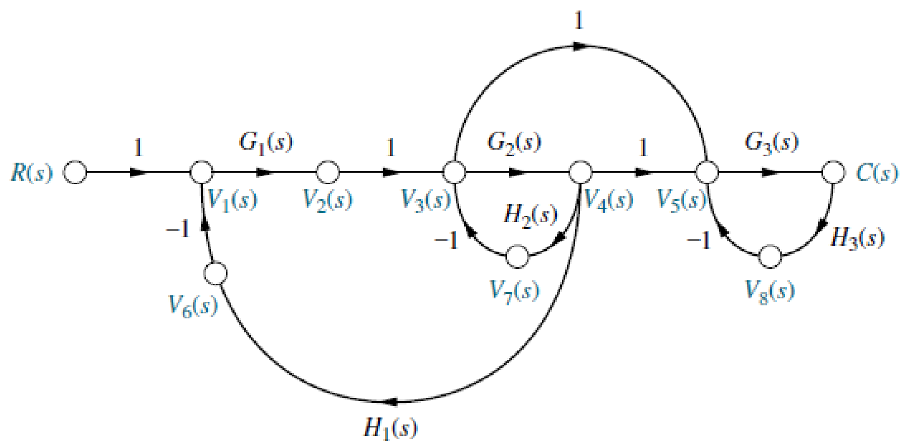
(b)



Example 9 (cont'd)

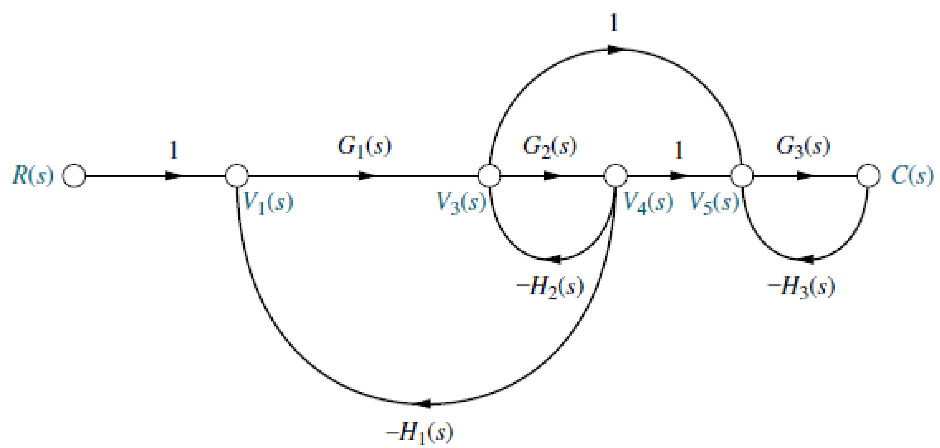
- If desired, simplify the signal-flow graph to the one shown in Figure **(c)** by eliminating signal nodes that have a single flow in and a single flow out, such as $V_2(s)$, $V_6(s)$, $V_7(s)$, and $V_8(s)$.

(b)



(from previous slide)

(c)



Mason's Rule (an alternative to block diagram)

- As will be shown later, the block diagram reduction technique requires successive application of fundamental relationships in order to arrive at the system transfer function.

- On the other hand, Mason's rule for reducing a signal-flow graph to a single transfer function requires the application of one formula. However, the use of the rule is somehow cumbersome and less straightforward.

- In this course, we will be using Block Diagram approach and Mason's Rule will not be covered.

a place of mind

Summary

- Modeling
  - Modeling is an important task!
  - Transfer function
  - Modeling of electrical & mechanical systems
  - State-space modeling
  - Signal flow graph
- Next
  - Modeling of electromechanical systems