ELEC 341: Systems and Control

Lecture 4

Modeling of electrical & mechanical systems

Modeling of electrical & mechanical systems

Modeling of Electrical & Mechanical Systems in Control Engineering:

- **Purpose of Modeling**:
  - To represent real-world systems mathematically for analysis and ***controller design***.
  - Enables ***prediction*** of system behavior under different conditions.
- **Mechanical System Modeling:**
  - Based on Newton's laws or other mechanical principles.
    - **Elements**: Mass (inertia), damper (viscous friction), spring (elasticity).
    - **Common Models**: Translational and rotational systems.
  - Represented using ***differential equations***.
- **Electrical System Modeling:**
  - Based on Kirchhoff's laws (KVL and KCL).
    - **Elements**: Resistor (R), inductor (L), capacitor (C), voltage and current sources.
  - Also modeled using ***differential equations***.

Modeling of electrical & mechanical systems

- **Electromechanical Systems:**
  - Combine both electrical and mechanical components in one system.
    - Examples include ***DC motors***, ***stepper motors***, and ***solenoids***.
  - Electrical input produces mechanical motion (or vice versa).
  - Modeling requires coupling of electrical and mechanical equations (e.g., torque-current and back EMF-speed relationships in motors).
  - Essential for applications like ***robotics***, ***automotive systems***, and ***mechatronics***.
- **Analogies Between Systems:**
  - Mechanical ↔ Electrical analogies help **unify analysis**.
  - Two common analogies: **Force-Voltage** and **Force-Current** analogies.
- **Transfer Function Representation:**
  - Systems are often represented in **Laplace domain** for analysis.
  - Transfer function relates input to output as a ratio of polynomials in '***s***'.
- **State-Space Modeling:**
  - An alternative to transfer functions, suitable for MIMO (multiple input multiple output) systems and time-domain analysis.
- **Importance in Control Engineering:**
  - ***Models are essential for system design, simulation, stability analysis, and controller synthesis***.

Water tank level control

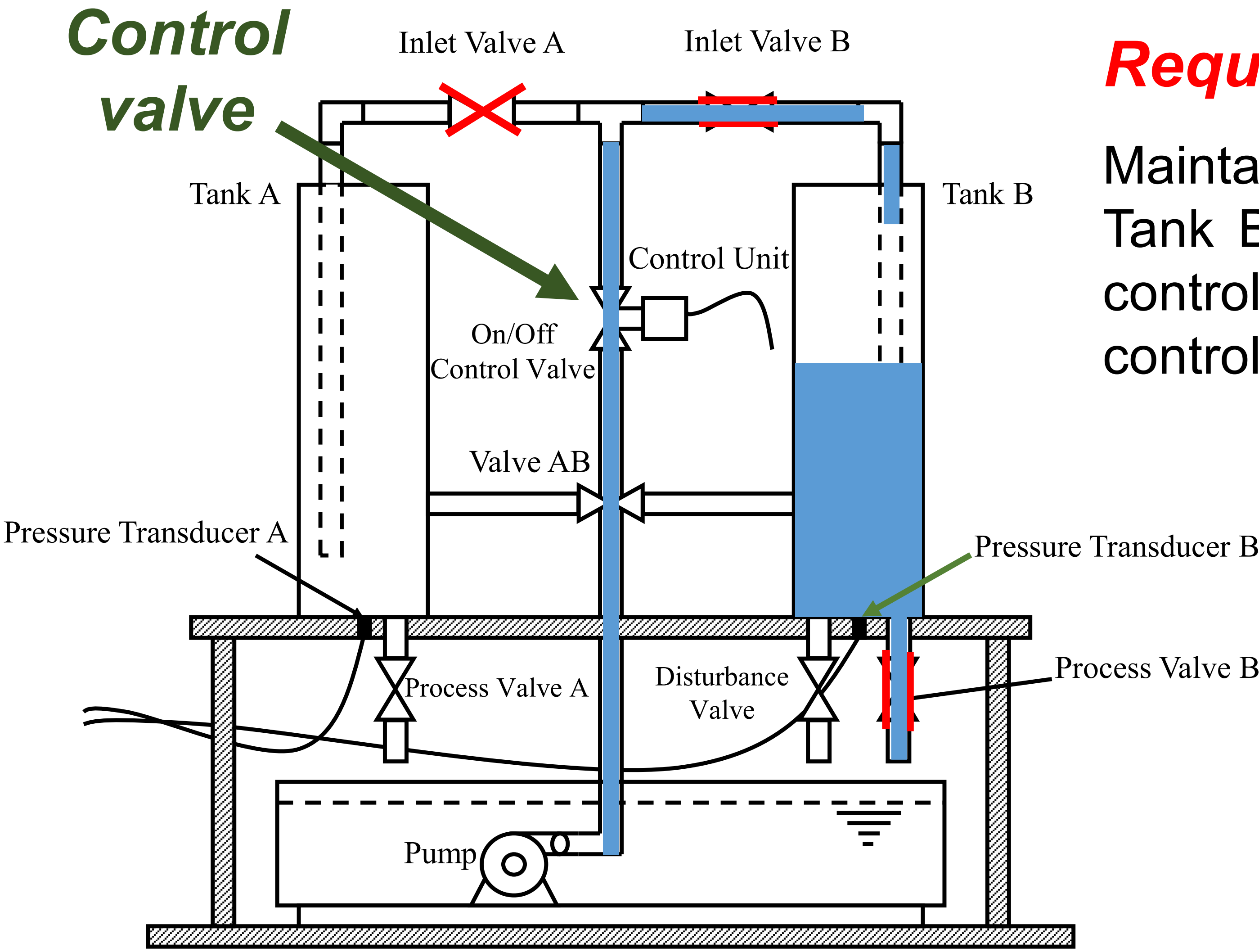
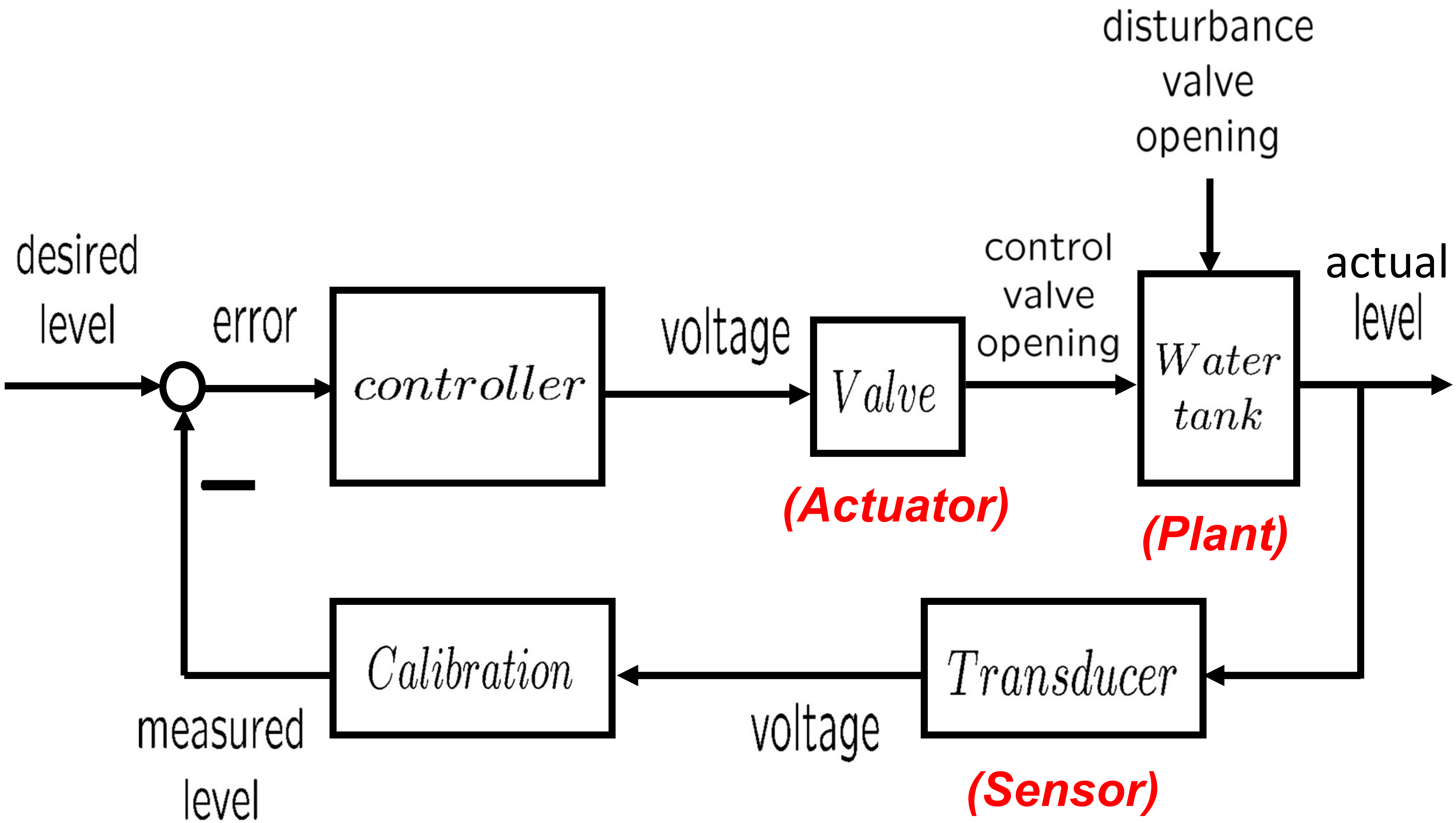


Figure: Schematic of the Tank Level Control Setup.

Requirement:

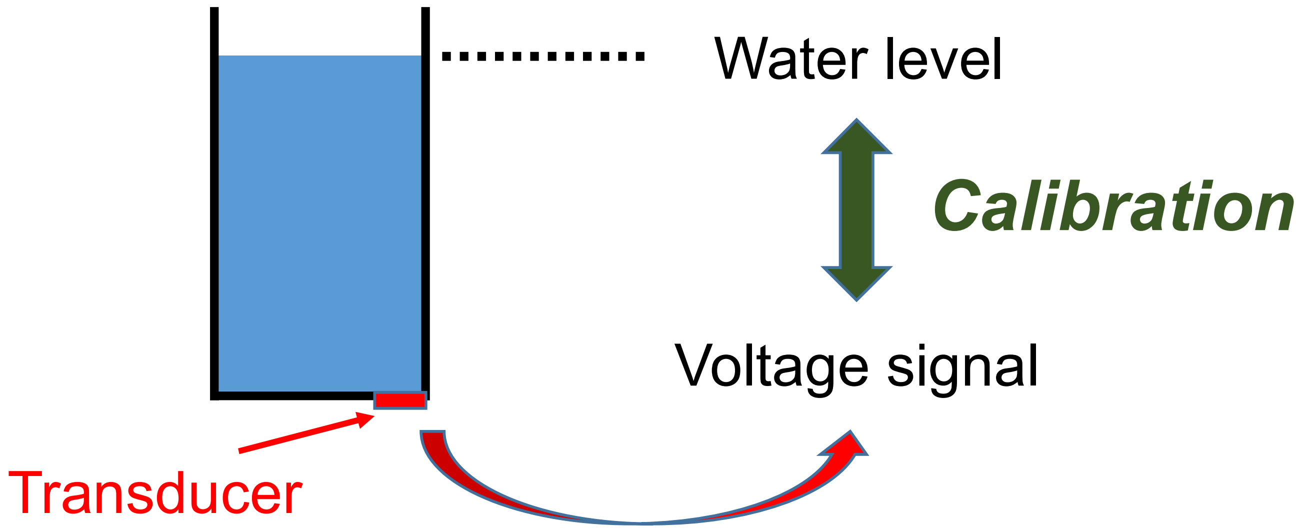
Maintain the liquid level of Tank B at a desired level by controlling the flow through control valve.

Water tank level control (Block diagram)



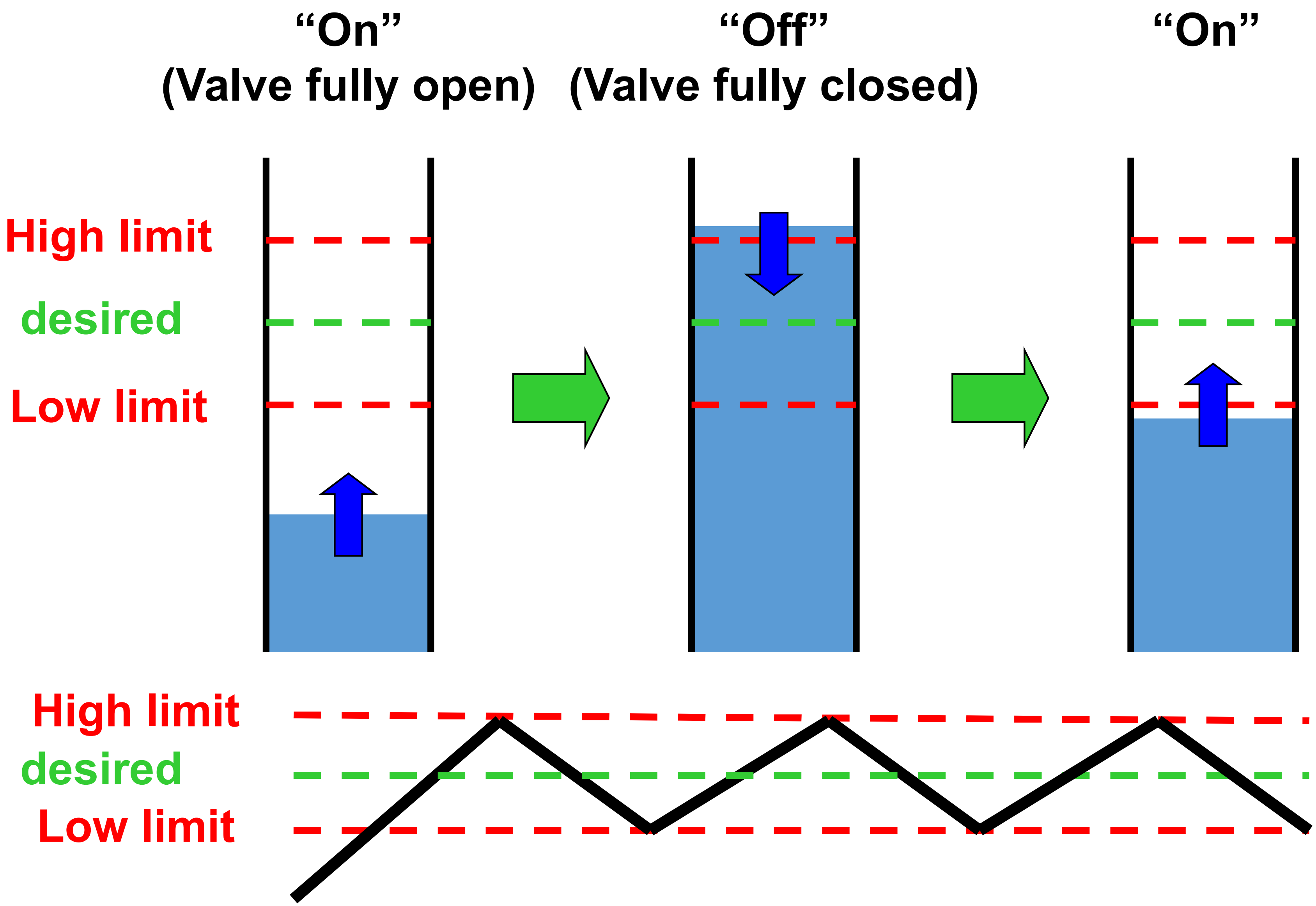
Two main tasks

- 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 using a proportional control algorithm instead.

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

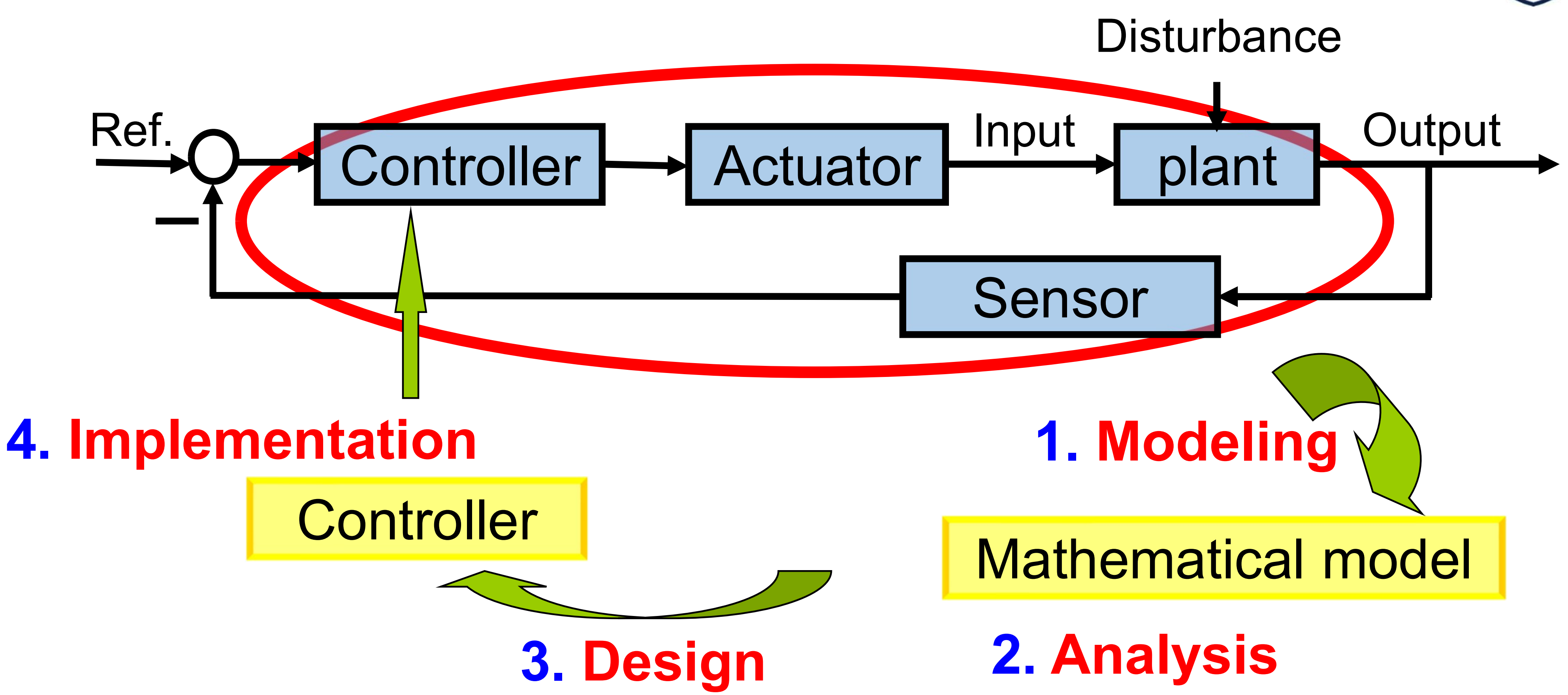
Frequency domain

PID & Lead-lag

Design examples

Matlab simulations

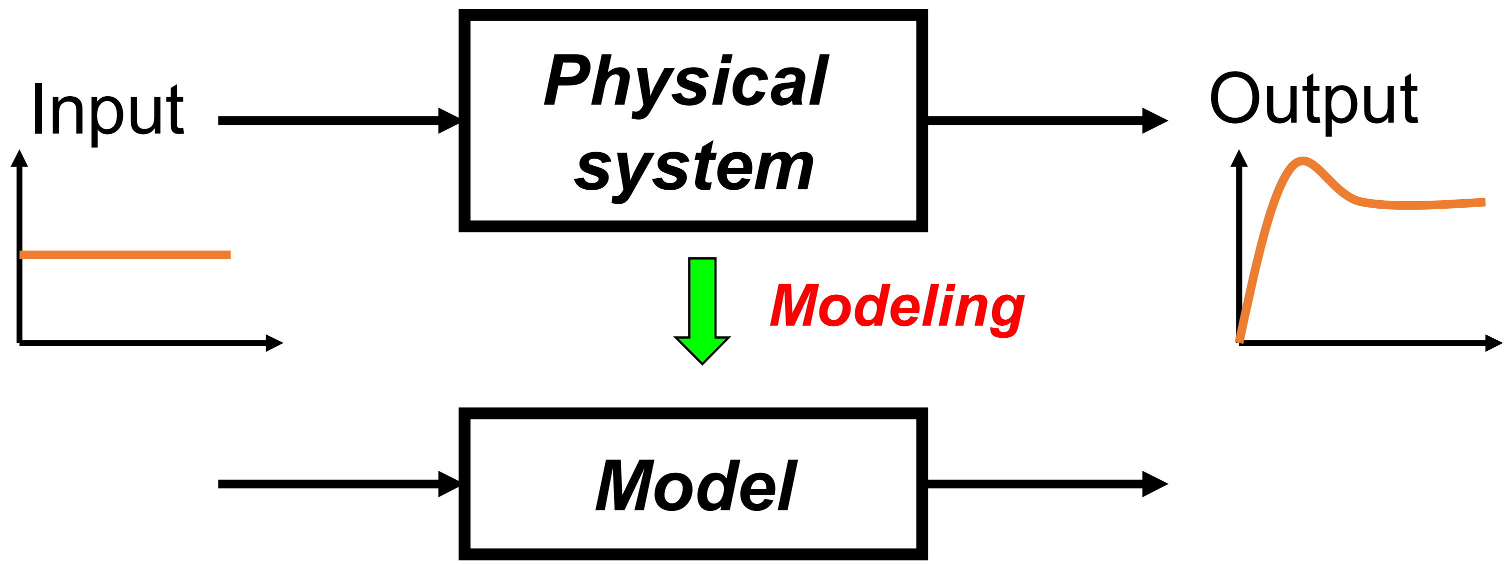
Controller design process (review)



- What is the “mathematical model”?
- What is the “transfer function”?
- How to do “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

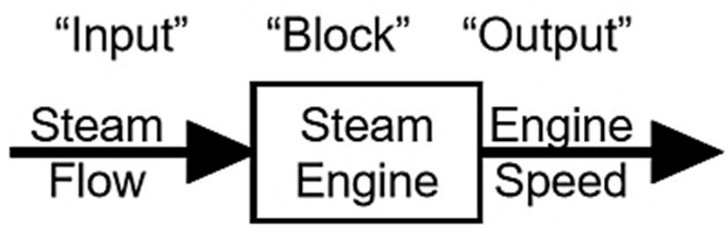
- Modeling is one of the most important and most difficult tasks in control system design.
- No mathematical model exactly represents a physical system.

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

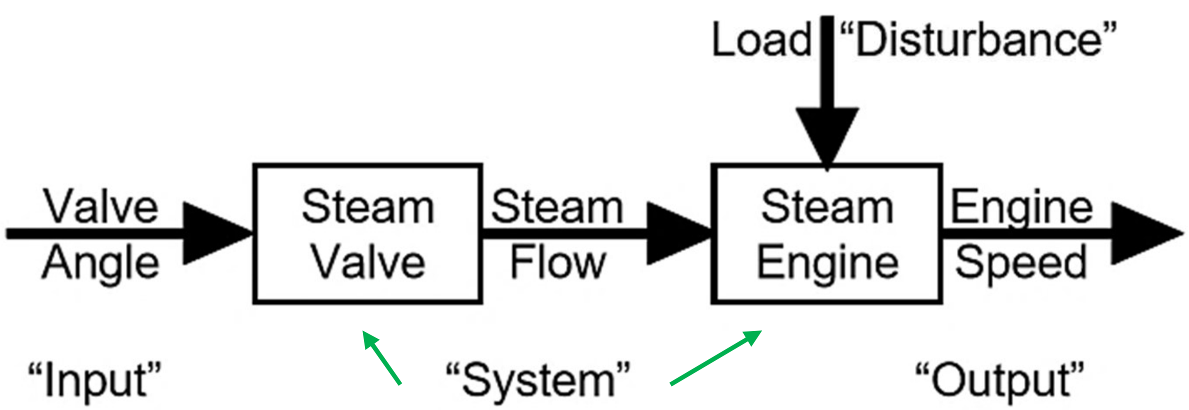
- Do not confuse math models with physical/engineering systems!

Block diagram

- Communication tool for engineering systems
  - Composed of blocks with inputs and outputs



- Each block can be considered as a "system"
  - Output from one block becomes input to another



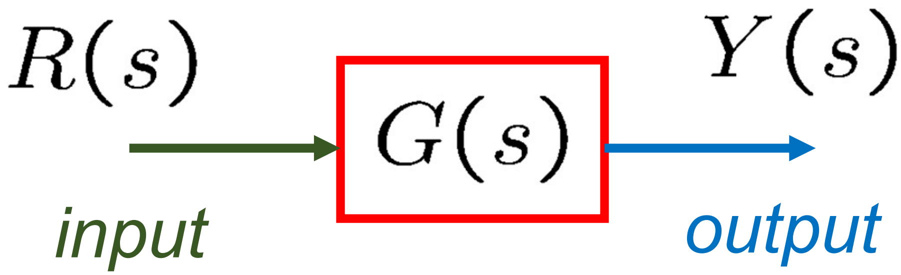
Transfer function

- A transfer function is defined by:

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

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

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



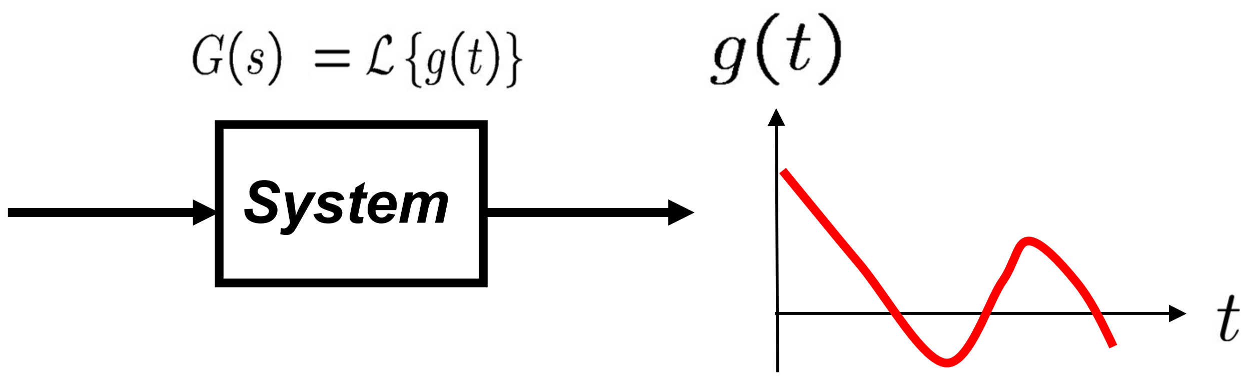
$$Y(s) = G(s)R(s)$$

- Transfer function is a generalization of "**gain**" concept.

Impulse response

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

$$\mathcal{L}: \quad r(t) = \delta(t) \quad \longrightarrow \quad R(s) = 1$$



- The output *g*(*t*) for the unit impulse input is called *unit impulse response*.

- Since $R(s)=1$, the system transfer function $G(s)$ can also be defined as the Laplace transform of impulse response, i.e., $Y(s)$:

$$Y(s) = G(s)R(s) \xrightarrow{R(s)=1} Y(s) = G(s)$$

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

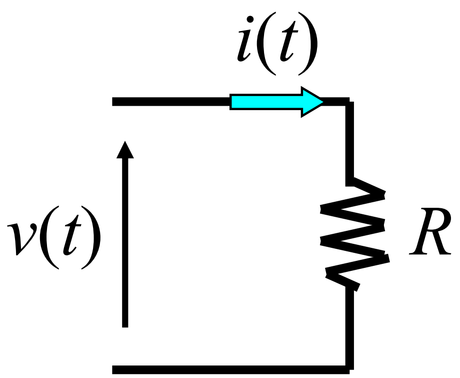
PID & Lead-lag

Design examples

Matlab simulations

Models of electrical elements

Resistance

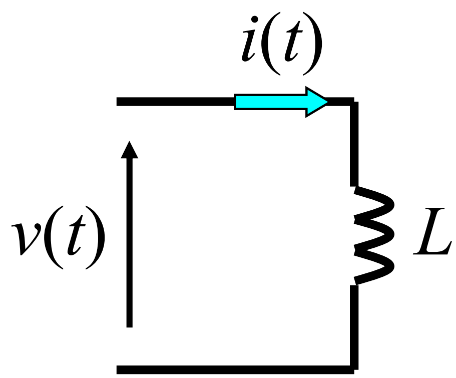


$$v(t) = Ri(t)$$

Laplace transform

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

Inductance

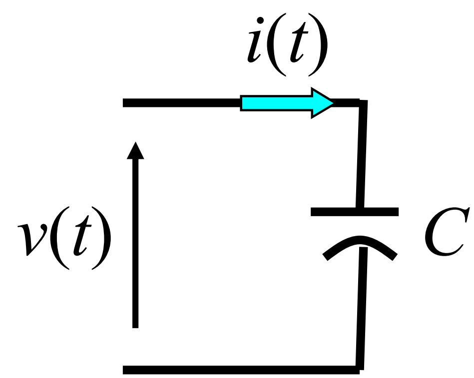


$$v(t) = L\frac{di(t)}{dt}$$

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

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

Capacitance



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

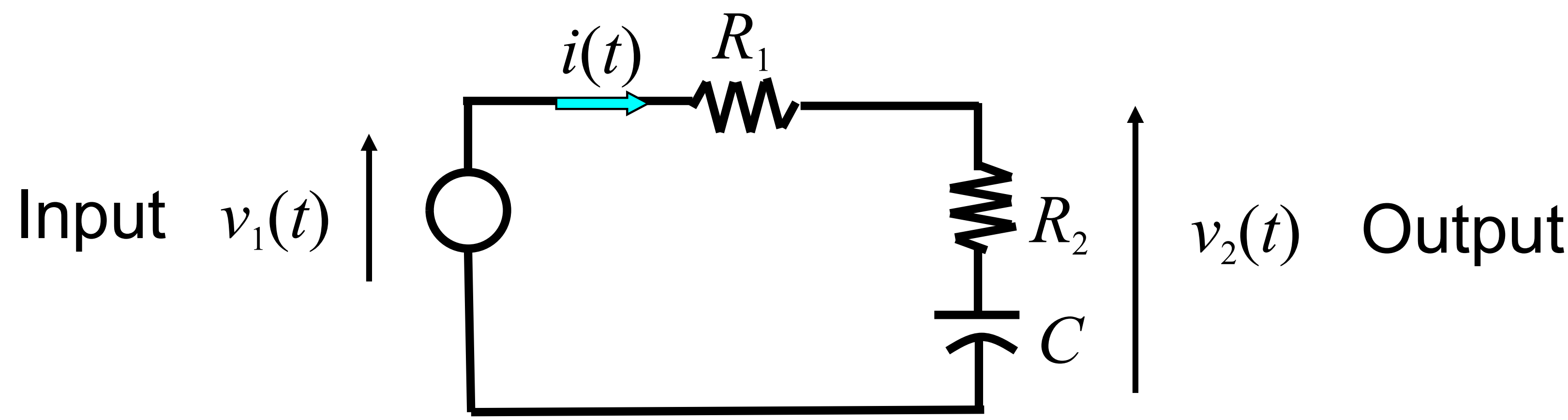
$$(v(0) = 0)$$

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

Impedance

Example 1: Modeling

Method 1: Conventional circuit analysis method



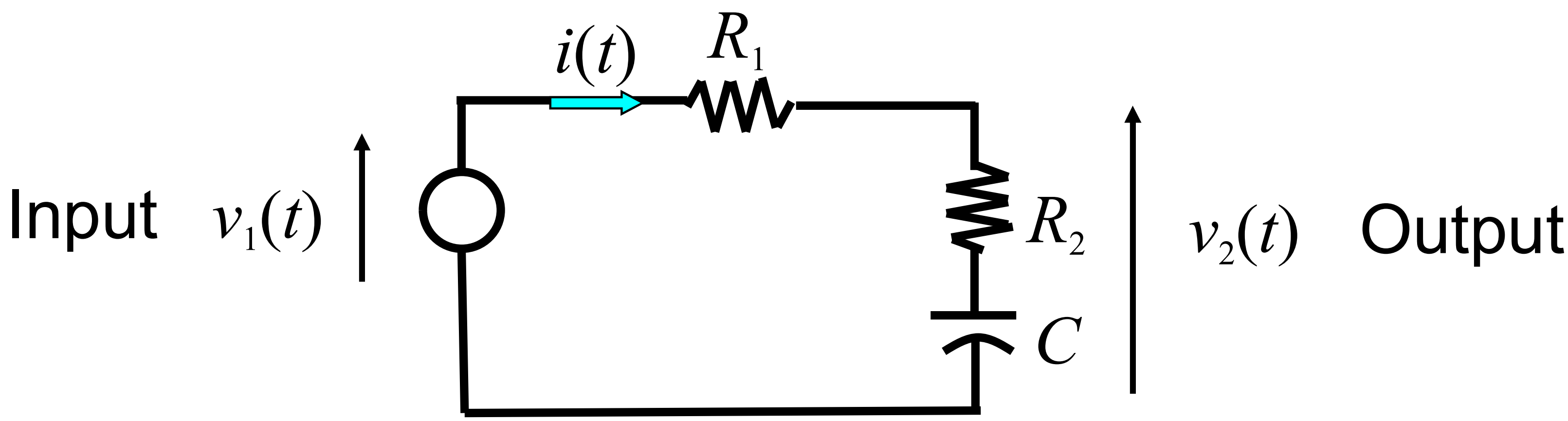
- Kirchhoff voltage law (with zero initial conditions),

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

- 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 $G(s)$:

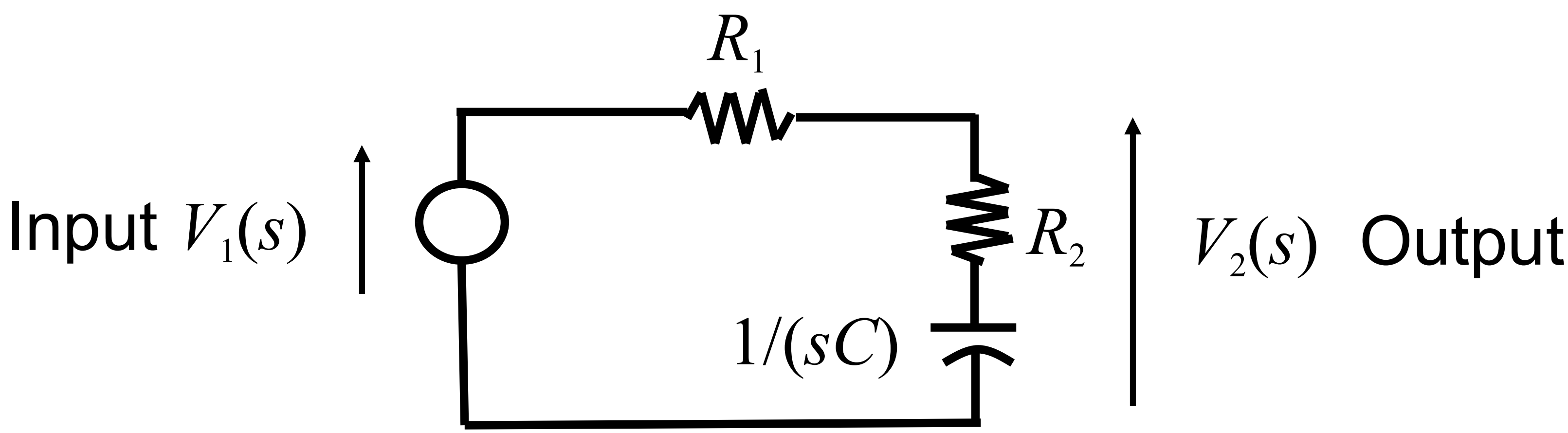
$$G(s) = \frac{V_2(s)}{V_1(s)} = \frac{R_2 + \frac{1}{sC}}{(R_1+R_2)+\frac{1}{sC}} \Rightarrow$$

$$\boxed{G(s) = \frac{R_2Cs+1}{(R_1+R_2)Cs+1}}$$ (first-order system)

Example 1 (cont'd)

Method 2: Impedance method

- **How to use impedance method?**
  - **Step 1:** Replace electrical elements with impedances.
  - **Step 2:** Deal with impedances as if they were ***resistances***.



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

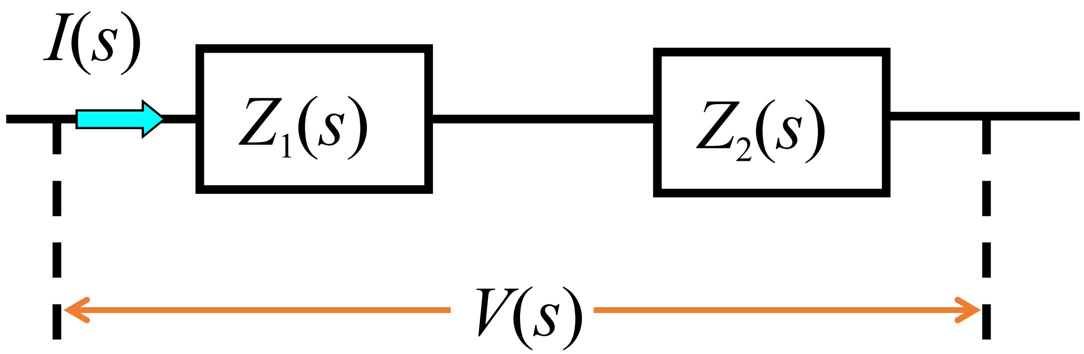
$$\boxed{G(s) = \frac{R_2 Cs + 1}{(R_1 + R_2)Cs + 1}}$$

Impedance computation

- Series connection

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

$Z_i(s)$: impedance

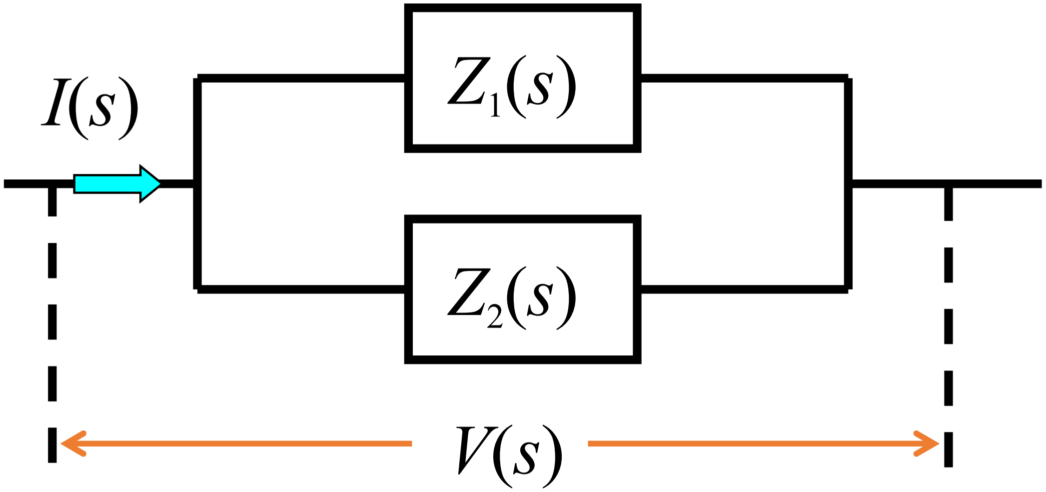


$$V(s) = Z(s)I(s)$$

- Parallel connection

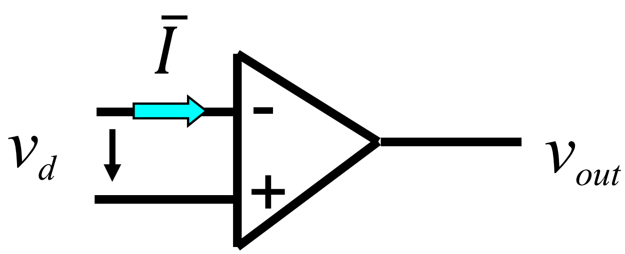
$$\boxed{\frac{1}{Z(s)} = \frac{1}{Z_1(s)} + \frac{1}{Z_2(s)}}$$

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



Operational amplifier (op-amp)

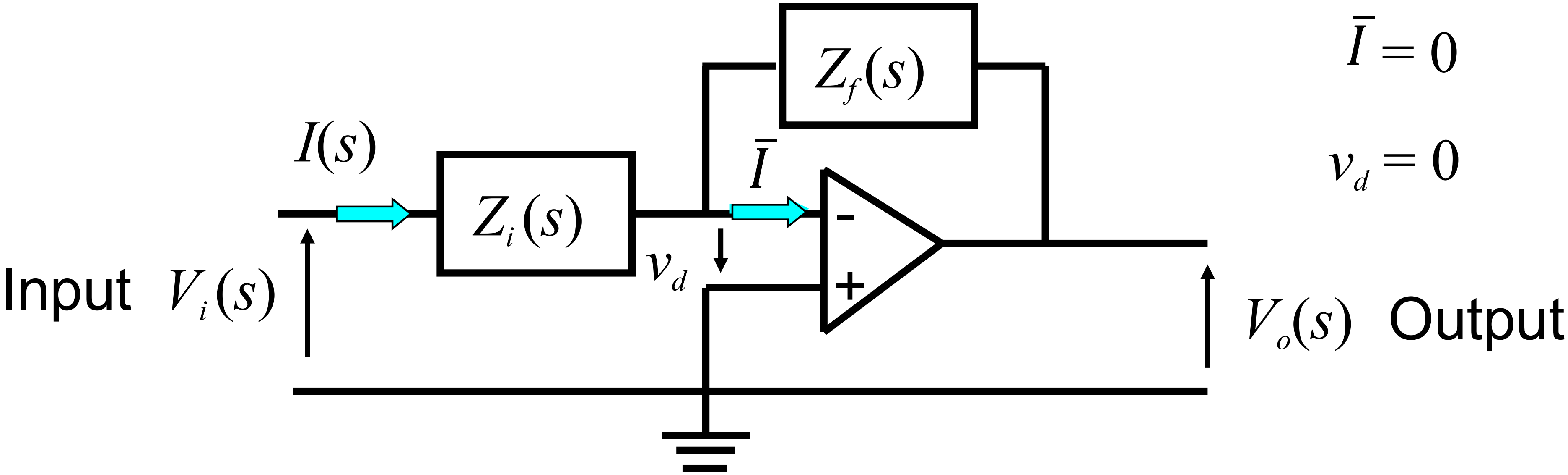
- Electronic voltage amplifier
- Basic building block of analog circuits
- Ideal op-amp (does not exist, but is a good approximation of reality):



$$\bar{I} = 0$$

$$v_d = 0$$

Example 2: Modeling of op-amp



$\bar{I} = 0$

$v_d = 0$

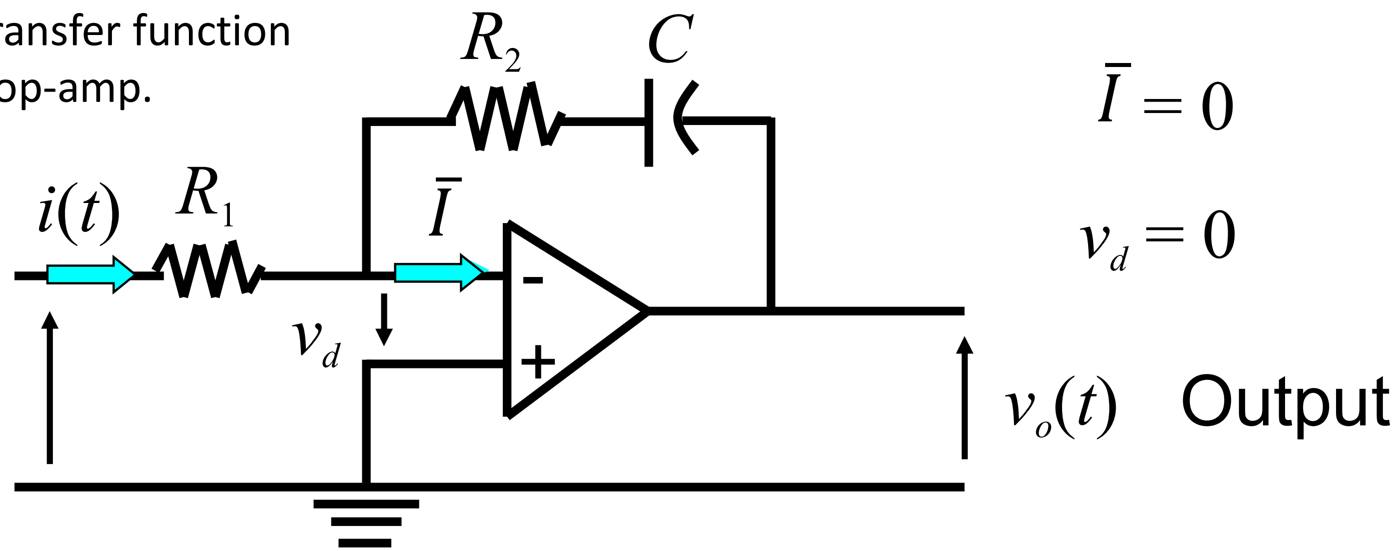
- Impedance $Z(s)$: $V(s) = Z(s)I(s)$
- Transfer function of the above op amp:

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

Example 2 (cont'd)

Goal: Let us find the transfer function $G(s)$ for the following op-amp.

Input $v_i(t)$



- Using the formula in previous slide,

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

$$G(s) = \frac{V_o(s)}{V_i(s)} = \frac{-(R_2 + \frac{1}{sC})}{R_1} = \boxed{-\frac{R_2Cs + 1}{R_1Cs}}$$

(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

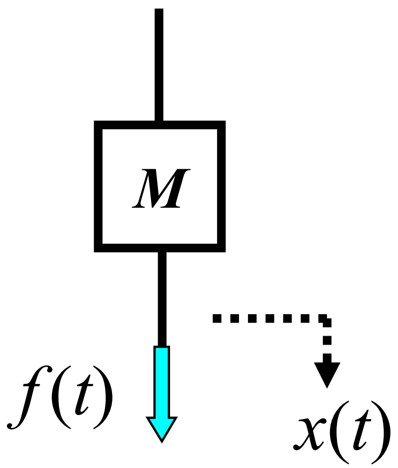
PID & Lead-lag

Design examples

Matlab simulations

Translational mechanical elements

Mass

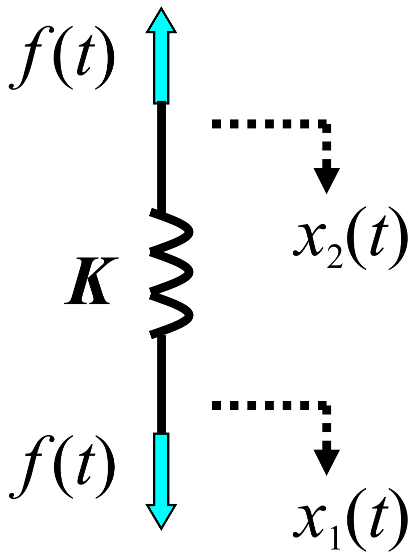


$$f(t) = Mx''(t)$$

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

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

Spring

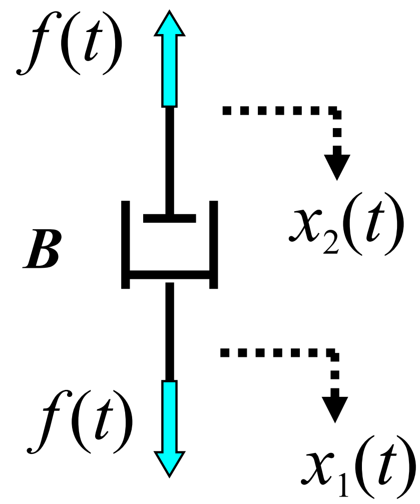


$$f(t) = K(x_1(t) - x_2(t))$$

$$\Downarrow$$

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

Damper



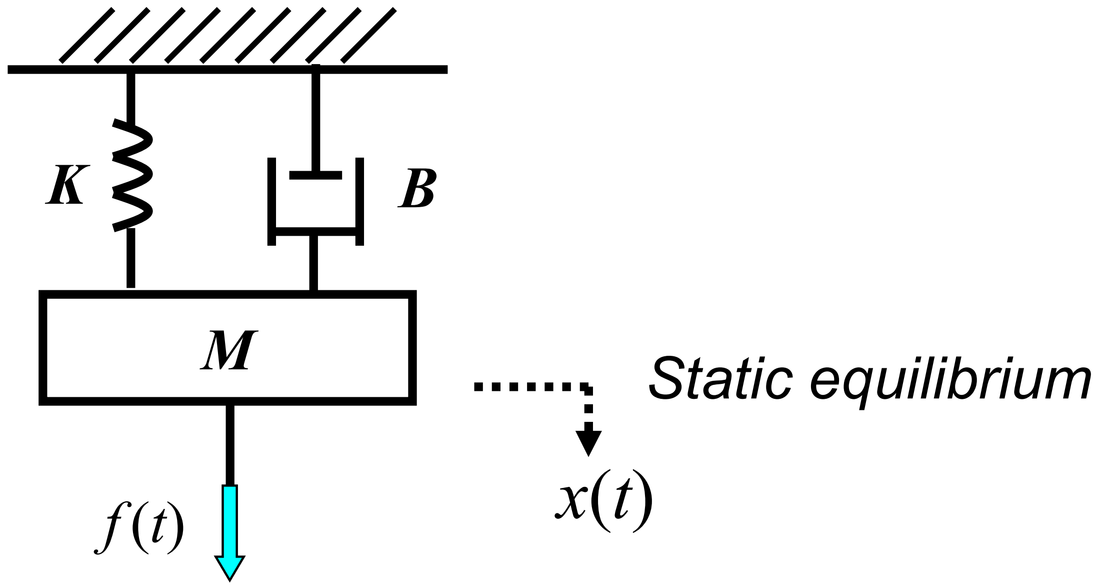
$$f(t) = B(x_1'(t) - x_2'(t))$$

$$\Downarrow \begin{pmatrix} x_1(0) = 0 \\ x_2(0) = 0 \end{pmatrix}$$

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

Note: The above equations are mostly conceptual. In practice, the elements are connected to other elements and because of that, we will use a ***systematic convention*** to be explained later on in order to tackle mechanical problems.

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),

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

How to write the governing differential equations for translational mechanical elements

Step 1:
Always put the mass of the system times $\boldsymbol{a}$ or $\ddot{\boldsymbol{x}}$ on the left-hand side of the equation and everything else on the right. Below I will describe the remaining terms which will all be on the right-hand side of the equation.

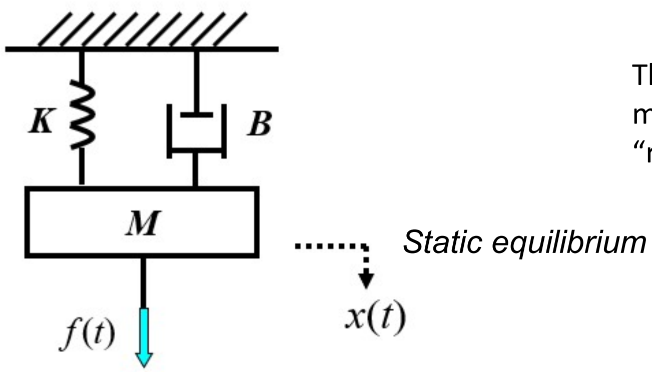
Step 2:
Draw free body diagram for each mass. Add ***artificial coordinates*** for $\boldsymbol{x}$'s on the ***masses*** and on the ***stationary walls***. The direction of $\boldsymbol{x}$'s should be in the same direction of the given $\boldsymbol{f(t)}$. The direction of forces (except for $\boldsymbol{f(t)}$, which is given) should always be ***away from the object***. For the stationary walls, use $\boldsymbol{x_i = 0}$.

Step 3:
- For "***K***" elements, use: $\boldsymbol{(-K)(x_{\text{left}} - x_{\text{right}})}$.
- For "***B***" elements, use: $\boldsymbol{(-B)(x'_{\text{left}} - x'_{\text{right}})}$.
- For "***M***" elements, use: $\boldsymbol{M\ddot{x}_M}$
  - Here, the subscript $\boldsymbol{M}$ on $\ddot{\boldsymbol{x}}_{\boldsymbol{M}}$ refers to the $\boldsymbol{x}$-coordinate which is placed ***on the mass***.

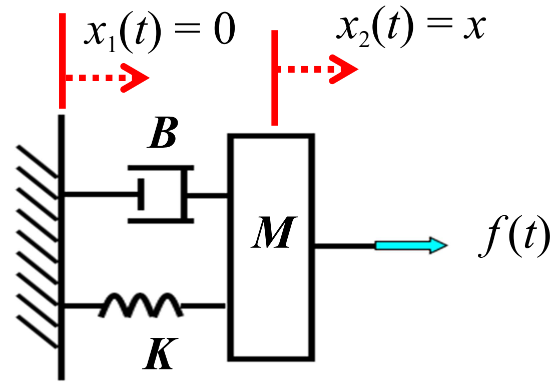
Important Note:
To find $\boldsymbol{x_{\text{left}}}$, we look at the left side of the spring to find the relevant $\boldsymbol{x}$. For $\boldsymbol{x'_{\text{left}}}$, we look at the left side of the dashpot. For $\boldsymbol{x_{\text{right}}}$ or $\boldsymbol{x'_{\text{right}}}$, we look at the right side of the element.

How to write the governing differential equations for translational mechanical elements (cont'd)

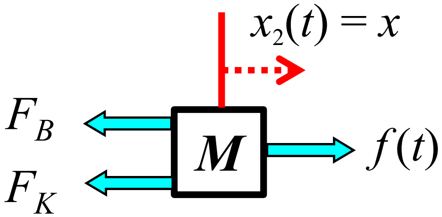
Let us find the governing differential equation in Example 3: $Mx''(t) = f(t) - Bx'(t) - Kx(t)$



The diagram can be rotated to make the concept of "left" and "right" easier to apply.



Free body diagram for mass $\boldsymbol{M}$:

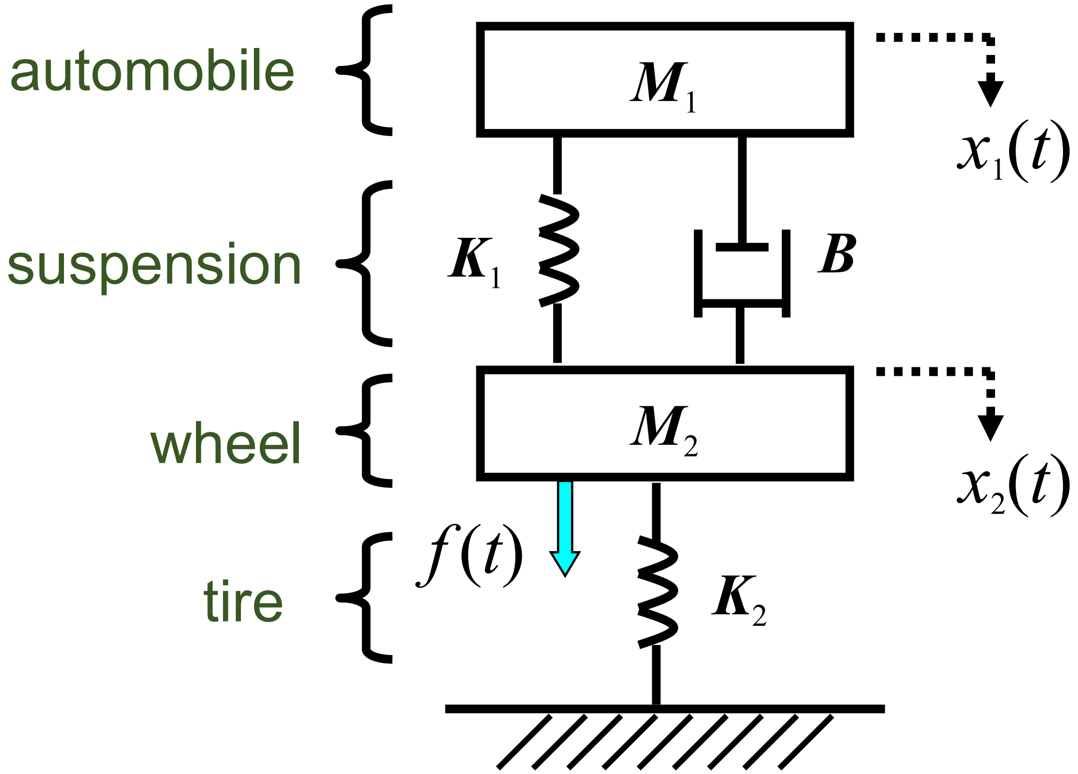


$$M\ddot{x}_2 = f(t) - F_B - F_K \xrightarrow{x_2 = x}$$

$$M\ddot{x} = f(t) - \{(-B)(\cancelto{0}{\dot{x}_1} - \dot{x}_2)\} - \{(-K)(\cancelto{0}{x_1} - x_2)\} \xrightarrow[\dot{x}_2 = \dot{x}]{x_2 = x}$$

$$\boxed{M\ddot{x}(t) = f(t) - B\dot{x}(t) - Kx(t)}$$

Example 4: Automobile suspension system



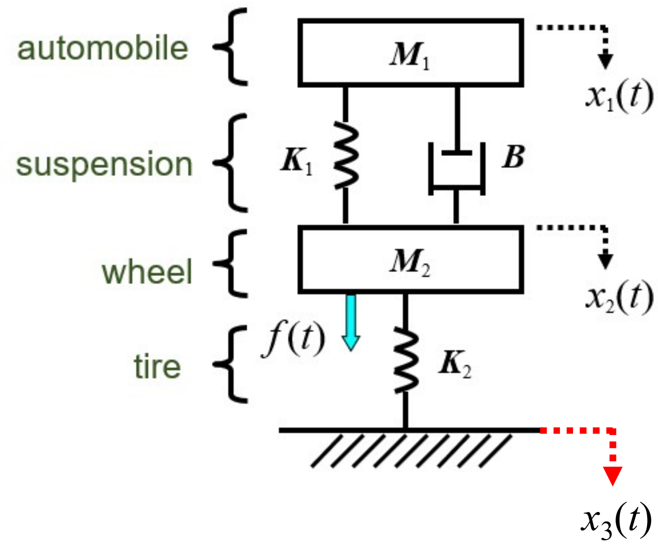
- Equations of motion by Newton's 2nd law

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

Example 4: Automobile suspension system

Always:

$$F_k = -k(x_{left} - x_{right})$$
$$F_B = -B(\dot{x}_{left} - \dot{x}_{right})$$



M1:

$$M_1\ddot{x}_1 = F_B + F_{k1} = -B(\dot{x}_1 - \dot{x}_2) + (-k_1)(x_1 - x_2)$$
$$\boxed{M_1\ddot{x}_1 = -B(\dot{x}_1 - \dot{x}_2) - k_1(x_1 - x_2)}$$

M2:

$$M_2\ddot{x}_2 = -F_{k1} - F_B + F_{k2} + f(t)$$
$$M_2\ddot{x}_2 = -(-k_1)(x_1 - x_2) - (-B)(\dot{x}_1 - \dot{x}_2) + (-k_2)(x_2 - x_3) + f(t)$$
$$\boxed{M_2\ddot{x}_2 = f(t) - B(\dot{x}_2 - \dot{x}_1) - k_1(x_2 - x_1) - k_2 x_2}$$

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

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

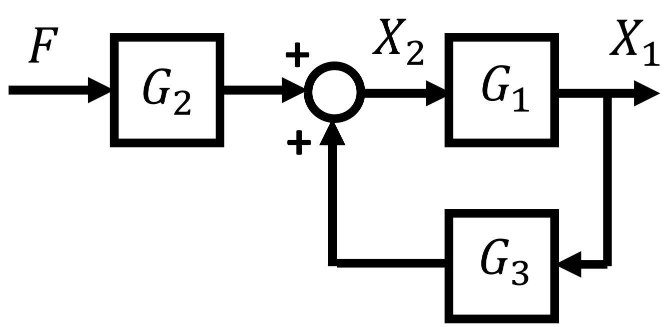
$$\begin{cases} M_1 s^2 X_1(s) &= -B(sX_1(s) - sX_2(s)) - K_1(X_1(s) - X_2(s)) \\ M_2 s^2 X_2(s) &= F(s) - B(sX_2(s) - sX_1(s)) - K_1(X_2(s) - X_1(s)) - K_2 X_2(s) \end{cases}$$

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

Make transfer functions so that
$\deg(\text{den}) \geq \deg(\text{num})$

Example 4 (cont’d)

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



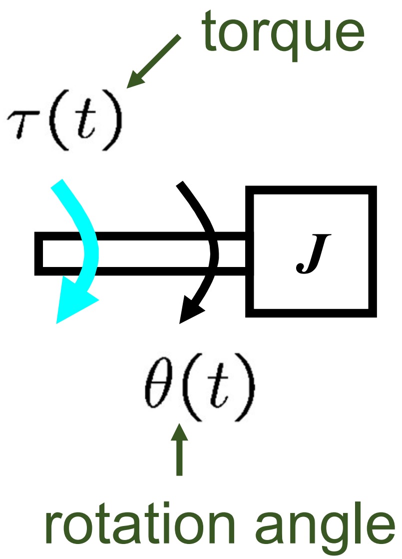
Block diagram

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

We will study how to derive this transfer function in the next lecture using a more systematic method.

Rotational mechanical elements

Moment of inertia

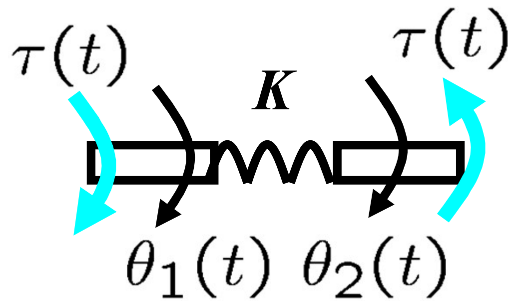


$$\tau(t) = J\theta''(t)$$

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

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

Rotational spring

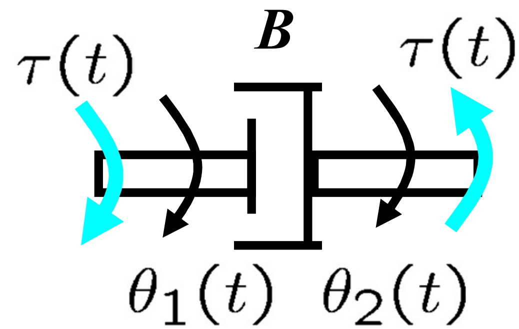


$$\tau(t) = K(\theta_1(t) - \theta_2(t))$$

$$\Downarrow$$

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

Friction

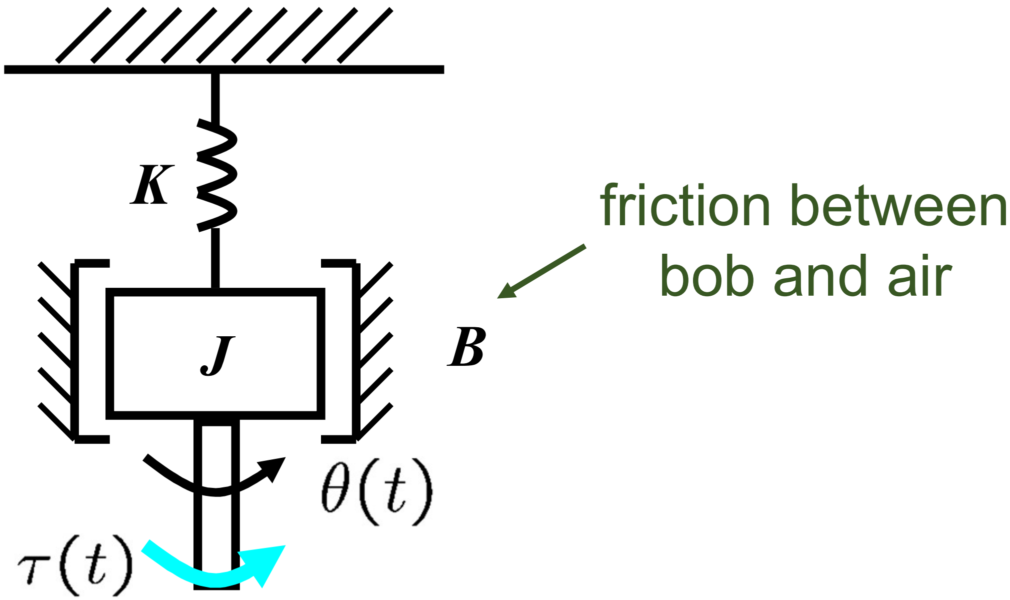


$$\tau(t) = B(\theta_1'(t) - \theta_2'(t))$$

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

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

Example 5: Torsional pendulum system



- Equation of motion by Newton's law,

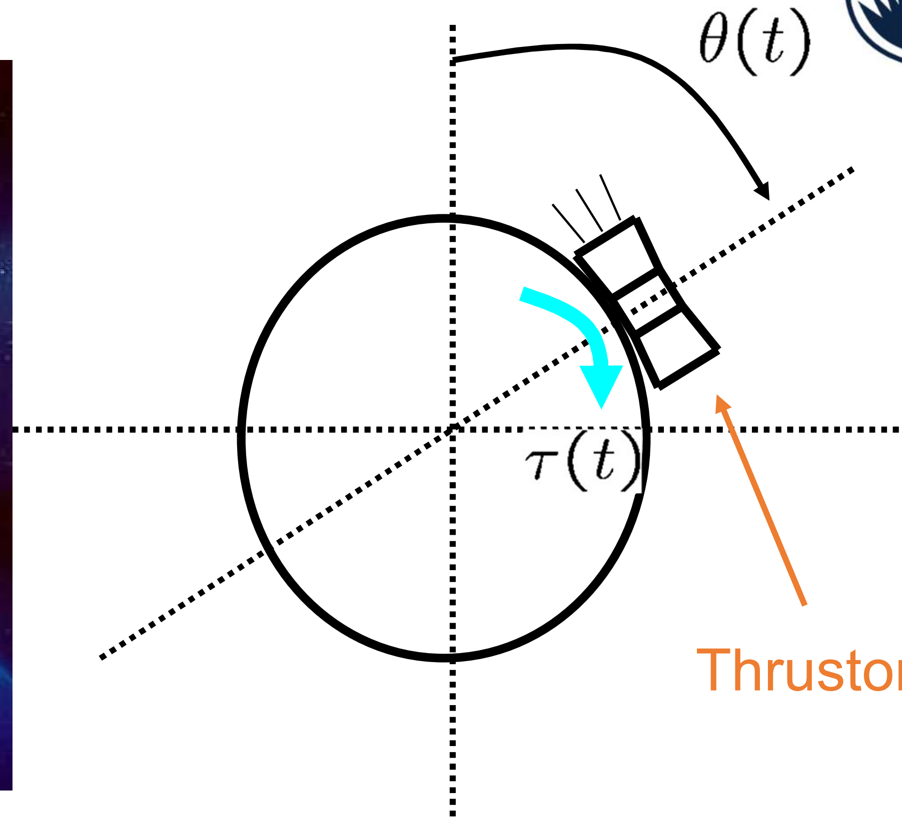
$$J\theta''(t) = \tau(t) - B\theta'(t) - K\theta(t)$$

- By Laplace transform (with zero ICs),

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

(2nd 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.
- **First Approach:** The first approach is known as the **classical approach,** or **frequency-domain approach**.
  - 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 of interconnected subsystems.
- The primary **disadvantage** of the classical approach is its limited applicability: It can mostly 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

- With the advent of space exploration, the demands on control systems expanded significantly, providing strong motivation for adopting the second approach.

- **Second Approach:** The second approach, **state-space approach** (also referred to as the **modern approach**, 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.
- Additionally, many systems do not have just a single input and a single output (**SISO**).
  - Multiple-input, multiple-output systems (**MIMO**) 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 (a great **advantage**), it is not as intuitive as the classical approach (a **disadvantage**).
  - 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 graphical presentation of data rapidly yields the physical interpretation.

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

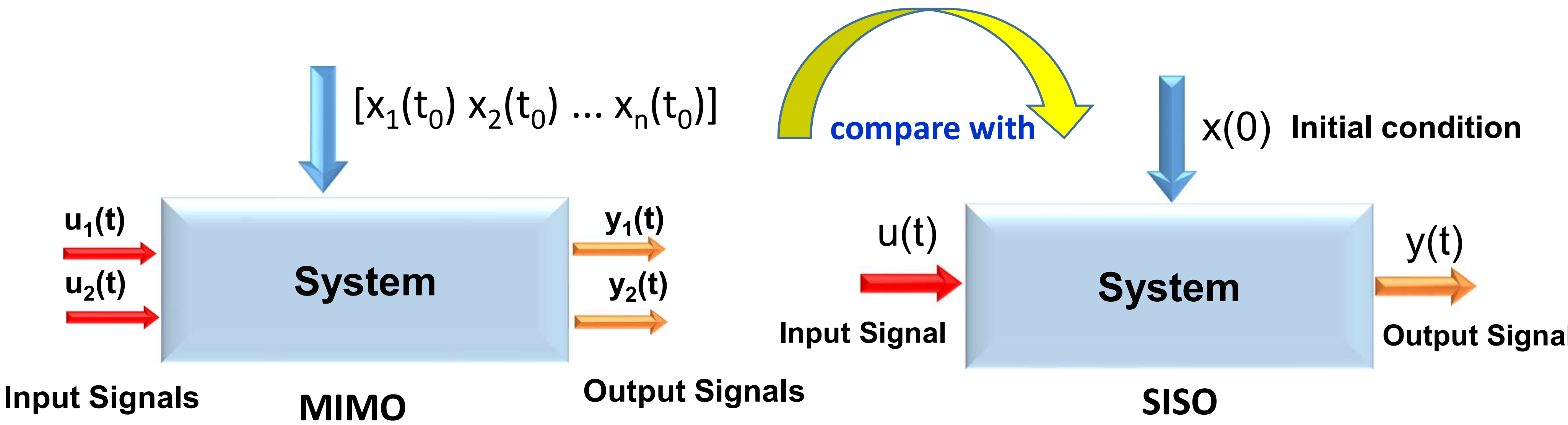
State Variables of a Dynamic System:

- 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

- Put differently, 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** (i.e., **input 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 $[x_1\ x_2\ ...\ x_n]$ for the system shown in the figure is a set such that knowledge of the initial values of the state variables at the initial time t_0 , i.e., $[x_1(t_0)\ x_2(t_0)\ ...\ x_n(t_0)]$, 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 ... x_n]. These first-order differential equations can be written in general form as:

$$
\begin{aligned}
\dot{x}_1 &= a_{11}x_1 + a_{12}x_2 + \ldots a_{1n}x_n + b_{11}u_1 + \cdots b_{1m}u_m \\
\dot{x}_2 &= a_{21}x_1 + a_{22}x_2 + \ldots a_{2n}x_n + b_{21}u_1 + \cdots b_{2m}u_m \\
&\vdots \\
\dot{x}_n &= a_{n1}x_1 + a_{n2}x_2 + \ldots a_{nn}x_n + b_{n1}u_1 + \cdots b_{nm}u_m
\end{aligned}
$$

State-Space Modeling

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

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

$\boldsymbol{n}$: number of state variables, $\boldsymbol{m}$: 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

- If the vector of input signals is defined as $\mathbf{u}$, then the system can be represented by the compact notation of the state variable differential equation as:

$$\dot{\mathrm{x}} = \mathrm{A\,x} + \mathrm{B\,u} \qquad \textbf{(state equation)}$$

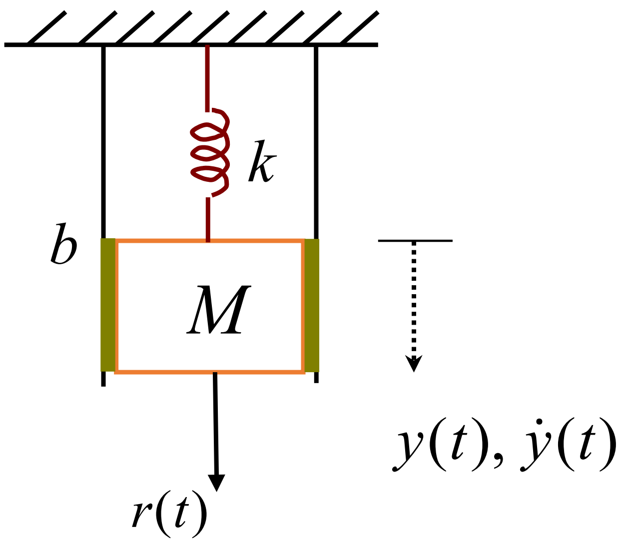
- This differential equation is also commonly called the **state equation**. The matrix $\mathbf{A}$ (the **system matrix**) is an $\boldsymbol{n}\times\boldsymbol{n}$ square matrix, and $\mathbf{B}$ (the **input matrix**) is an $\boldsymbol{n}\times\boldsymbol{m}$ matrix. The state differential equation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

$$\mathrm{y} = \mathrm{C\,x} + \mathrm{D\,u} \qquad \textbf{(output equation)}$$

- Where $\mathbf{y}$ is the set of output signals expressed in column vector form, $\mathbf{C}$ is the **output matrix**, and $\mathbf{D}$ is the **feed-forward matrix**. The **state-space representation** (or **state-variable representation**) is comprised of the state equation and the output equation. The state-space representation is sometimes called **dynamical equation**.

Example 6: State-space representation

Find the state-space representation of the following system:



By Newton's Law:

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

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

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

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

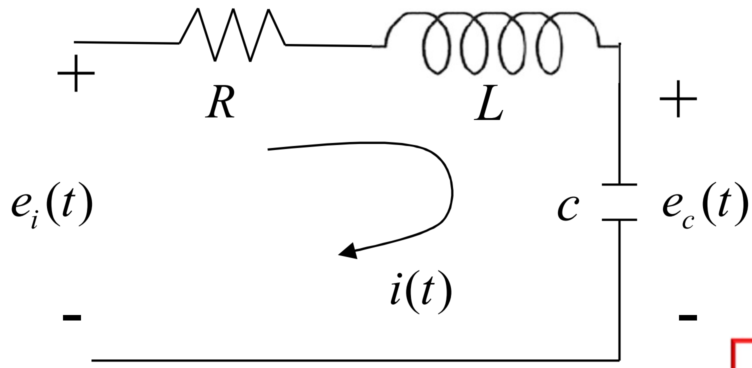
$$\Rightarrow \begin{cases} \dot{x}_1 = (0)x_1 + (1)x_2 + (0)u \\ \dot{x}_2 = -\dfrac{k}{M}x_1 - \dfrac{b}{M}x_2 + \dfrac{1}{M}u \end{cases} \quad ; \quad y = x_1 \rightarrow \quad y = (1)x_1 + (0)x_2 + (0)u$$

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

Example 7: State-space representation

Find the state-space representation of the following system:

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



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

(a)

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

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

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

(b)

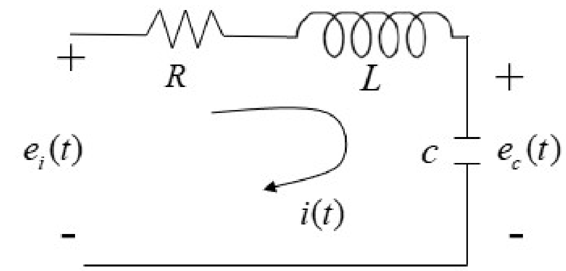
let $\begin{cases} x_1(t) = e_c(t) \\ x_2(t) = i(t) \\ y_1(t) = e_c(t) \\ y_2(t) = e_R(t) = Ri \end{cases}$

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

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

Example 7: State-space representation

(a)



State-Space Representation of an RLC Circuit

We begin with the voltage equation for the RLC series circuit:

$$Ri + L\frac{di}{dt} + \frac{1}{C}\int_0^t i(\tau)\,d\tau = e_i \tag{1}$$

Define the state variables as:

$$x_1 = i, \quad x_2 = \int_0^t i(\tau)\,d\tau \tag{2}$$

Taking the derivative of the state variables:

$$\dot{x}_1 = \frac{di}{dt}, \quad \dot{x}_2 = i = x_1$$

Example 7: State-space representation

a place of mind

From equation (1):

$$Ri + L\frac{di}{dt} + \frac{1}{C}x_2 = e_i \Rightarrow L\frac{di}{dt} = e_i - Rx_1 - \frac{1}{C}x_2$$

$$\frac{di}{dt} = \frac{1}{L}e_i - \frac{R}{L}x_1 - \frac{1}{LC}x_2$$

Substituting into the state-space form:

$$\dot{x}_1 = -\frac{R}{L}x_1 - \frac{1}{LC}x_2 + \frac{1}{L}e_i$$

$$\dot{x}_2 = (1)\cdot x_1 + (0)\cdot x_1 + (0)\cdot e_i$$

Writing the system in matrix form:

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

This is in the standard state-space form:

$$\dot{x} = Ax + Bu$$

Example 7: State-space representation

Output Equation

Assuming the output y is the current $i = x_1$, we define:

$$y = Cx + Du, \quad \text{where} \quad C = [1 \;\; 0], \quad D = [0]$$

Thus:

$$\boxed{y = [1 \;\; 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0] e_i}$$

Obtain transfer function from state-space representation

Dynamical equation → 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

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

$$Y(s) = CX(s) + DU(s)$$

$$assume \quad x(0) = 0$$

$$X(s) = (sI - A)^{-1} BU(s)$$

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

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.

Reminder for calculation of inverse of a matrix

a place of mind UBC

Minors and Cofactors:

- A **minor** is defined as the determinant of a square matrix, shown by "**A**", that is formed when a row and a column is deleted from a square matrix. The minors are based on the columns and rows that are deleted. Let $\mathbf{M}_{ij}$ be the $\mathbf{(}\boldsymbol{n}\mathbf{-1)\times(}\boldsymbol{n}\mathbf{-1)}$ matrix obtained by deleting the ith row and jth column. So, we will have: **minor of** $\boldsymbol{a_{ij}} = \mathbf{det(M}_{ij}\mathbf{)}$.
- **Co-factors** are the number you get when you eliminate the row and column of a designated element in a matrix, which is just a grid in the form of a square or a rectangle. The co-factor is always preceded by a negative (-) or a positive (+) sign, depending on whether the number is in a + or – position.

Cofactor Formula:

- Let **A** be any matrix of order $\boldsymbol{n\times n}$ and $\mathbf{M}_{ij}$ be the $\mathbf{(}\boldsymbol{n}\mathbf{-1)\times(}\boldsymbol{n}\mathbf{-1)}$ matrix obtained by deleting the ith row and jth column. **Here,** $\mathbf{det(M}_{ij}\mathbf{)}$ **is the minor of** $\boldsymbol{a_{ij}}$. The **cofactor** $\mathbf{C}_{ij}$ of $\boldsymbol{a_{ij}}$ can be found using the formula:

$$\mathbf{C}_{ij} = (-1)^{i+j}\,\mathbf{det}(\mathbf{M}_{ij})$$

- Thus, cofactor is always represented with +ve (positive) or -ve (negative) signs.

Reminder for calculation of inverse of a matrix

- For a 2x2 matrix the inverse is:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \Rightarrow A^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

- In other words, swap the positions of a and d, put negatives in front of b and c, and divide everything by ad – bc .
- In general, ***the inverse of a matrix*** is obtained as follows:

$$\mathbf{A}^{-1} = \frac{\mathbf{adj}\ (\mathbf{A})}{|\mathbf{A}|}$$

where,

$\mathbf{adj}\ (\mathbf{A}) = \mathbf{C}^T$ **=** transpose of matrix of cofactors
$|\mathbf{A}|$ = the determinant of **A**

Note: The **transpose of a matrix** is found by interchanging its rows into columns or columns into rows.

Example 8: Calculation of inverse of a matrix

Find the inverse matrix of the given 3 by 3 matrix: $A = \begin{bmatrix} 1 & 2 & -1 \\ 2 & 1 & 2 \\ -1 & 2 & 1 \end{bmatrix}$

Solution:

Cofactor matrix is:

$$\begin{bmatrix} \begin{vmatrix} 1 & 2 \\ 2 & 1 \end{vmatrix} & -\begin{vmatrix} 2 & 2 \\ -1 & 1 \end{vmatrix} & \begin{vmatrix} 2 & 1 \\ -1 & 2 \end{vmatrix} \\ -\begin{vmatrix} 2 & -1 \\ 2 & 1 \end{vmatrix} & \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix} & -\begin{vmatrix} 1 & 2 \\ -1 & 2 \end{vmatrix} \\ \begin{vmatrix} 2 & -1 \\ 1 & 2 \end{vmatrix} & -\begin{vmatrix} 1 & -1 \\ 2 & 2 \end{vmatrix} & \begin{vmatrix} 1 & 2 \\ 2 & 1 \end{vmatrix} \end{bmatrix} \Rightarrow$$

So the cofactor matrix = $\begin{bmatrix} 1-4 & -(2+2) & 4+1 \\ -(2+2) & 1-1 & -(2+2) \\ 4+1 & -(2+2) & 1-4 \end{bmatrix}$

$$C = \begin{bmatrix} -3 & -4 & 5 \\ -4 & 0 & -4 \\ 5 & -4 & -3 \end{bmatrix} \Rightarrow$$

By transposing the cofactor matrix, we get the adjoint matrix. $\Rightarrow$

So adj A = $\begin{bmatrix} -3 & -4 & 5 \\ -4 & 0 & -4 \\ 5 & -4 & -3 \end{bmatrix} = C^T.$

$A = \begin{bmatrix} 1 & 2 & -1 \\ 2 & 1 & 2 \\ -1 & 2 & 1 \end{bmatrix}$. Let us use the first row to find the determinant.

det A = 1 (cofactor of 1) + 2 (cofactor of 2) + (-1) cofactor of (-1)

= 1(-3) + 2(-4) + (-1)5

= -3 - 8 - 5

= -16

A^{-1} = (adj A)/(det A). i.e., divide every element of adj A by det A.

$$\Rightarrow \text{Then } A^{-1} = \begin{bmatrix} -3/-16 & -4/-16 & 5/-16 \\ -4/-16 & 0/-16 & -4/-16 \\ 5/-16 & -4/-16 & -3/-16 \end{bmatrix}$$

$$\boxed{A^{-1} = \begin{bmatrix} 3/16 & 1/4 & -5/16 \\ 1/4 & 0 & 1/4 \\ -5/16 & 1/4 & 3/16 \end{bmatrix}.}$$

Example 9: Converting 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.

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

$$y = [1 \quad 0 \quad 0]\mathbf{x}$$

Solution:

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

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

Compare with the above two equations $\longrightarrow$

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

$$\mathbf{C} = [1 \quad 0 \quad 0]$$

$$\mathbf{D} = 0$$

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

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

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

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

Signal Flow Graphs (Introduction)

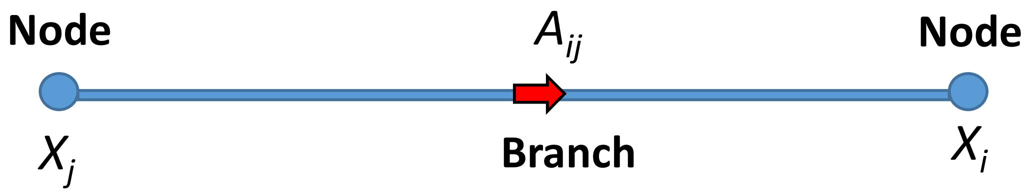
- Signal-flow graphs are an alternative to block diagrams.
  - Block diagram approach will be discussed in details later in the next lecture.
- **Signal flow graphs** are a pictorial representation of the simultaneous equations describing a system.
- These graphs display the transmission of signals through the system, as does the block diagrams.
- Unlike block diagrams, which consist of blocks, signals, summing junctions, etc., 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:

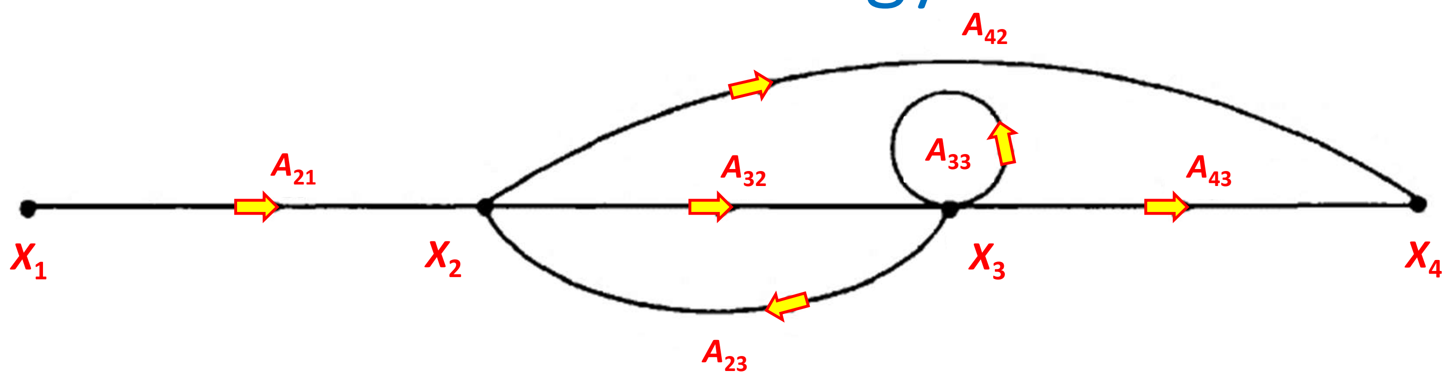
$$X_i = A_{ij} X_j$$

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



- Every variable (i.e., ***signal***) in a signal flow graph is designated by a **Node**.
- Every ***transmission function*** in a signal flow graph is designated 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**, i.e., X_1, contains only the outgoing branches.
- An **output node** or **sink**, i.e., X_4, contains only the incoming branches.
- 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 **branch gain** is the transmission function of the branch.
- 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 $A_{21}A_{32}A_{43}$. This is because the transmission function is a **multiplicative operator**.
- 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 10:
Converting feedback system block diagram into a signal flow graph

- **Step 1:** Draw the signal nodes for the system. The signal nodes for the given system are shown in **Figure (a)**.
- **Step 2:** Interconnect the signal nodes with system branches. The interconnection of the nodes with branches that represent the systems is shown in **Figure (b)**.

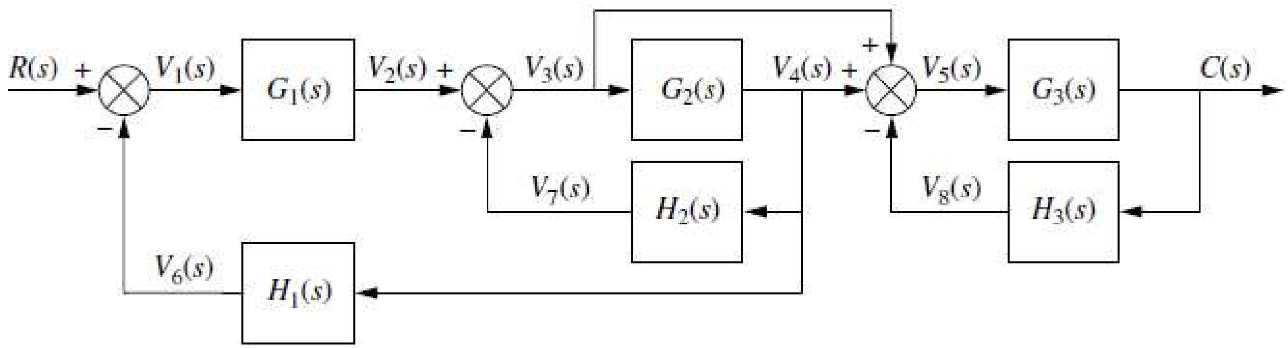


Figure (a)

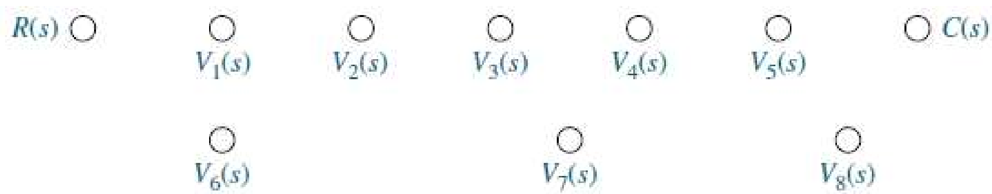
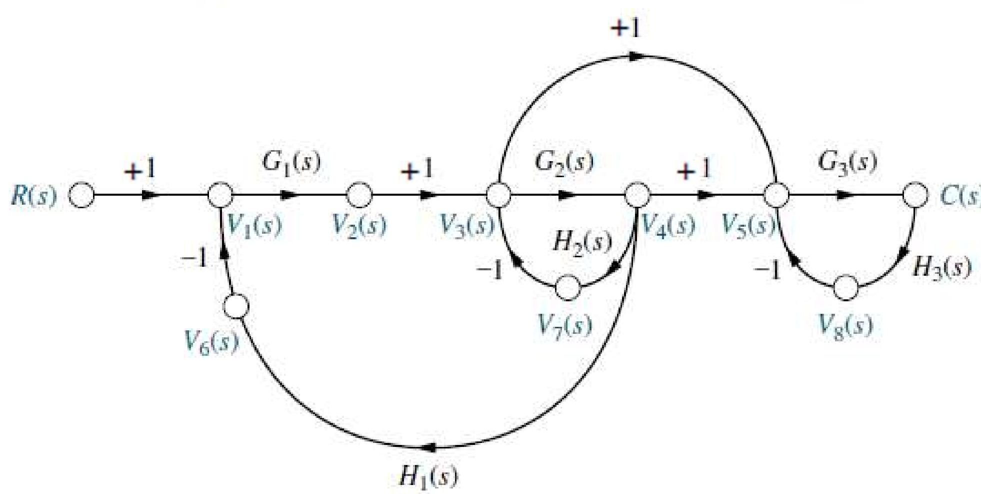


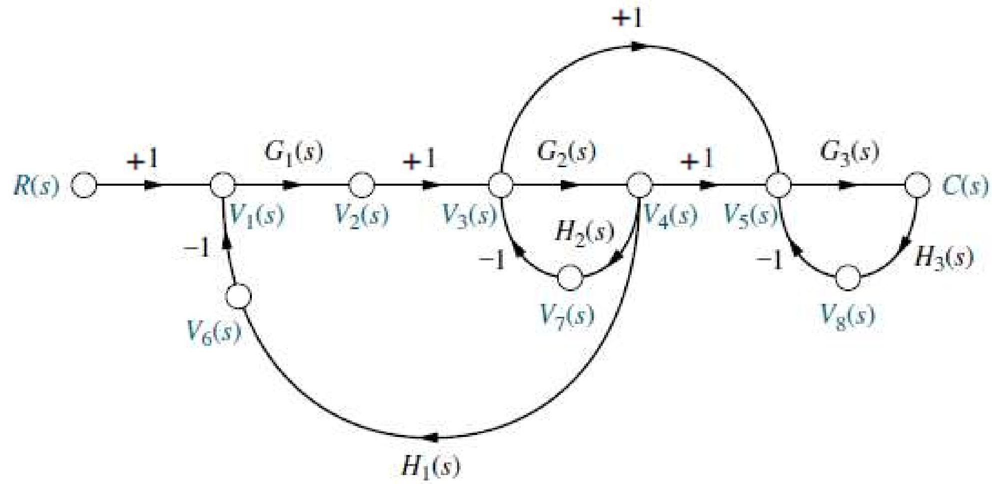
Figure (b)



Example 10 (cont'd)

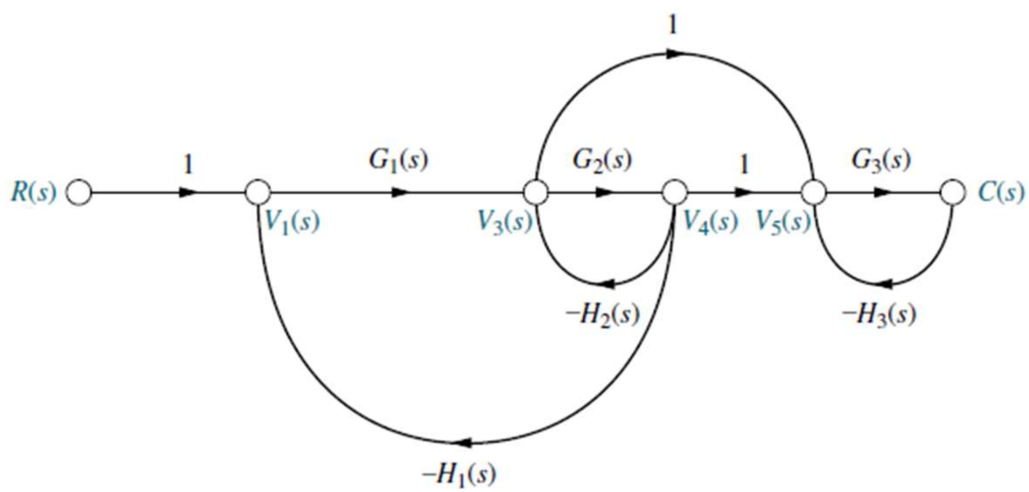
- **Step 3:** Simplify the signal-flow graph to the one shown in **Figure (c)** by eliminating signal nodes that have a *single flow in* and a *single flow out*, such as $V_2(s)$, $V_6(s)$, $V_7(s)$, and $V_8(s)$. Make sure that you multiply the transmission functions before and after these particular signal nodes.

Figure (b)



(from previous slide)

Figure (c)



Mason's Rule (an alternative to block diagram)

a place of mind UBC

- 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 quite cumbersome and less straightforward.

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

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