Control Systems And Their Components (EE391)

Lec. 3: Signal Flow Graphs & State Space Representation

Wed. Feb. 24th, 2016

Dr. Mohamed Hamdy Osman

Lecture Outline

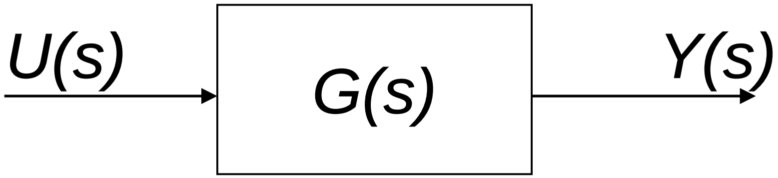
- Block Diagram Representations
- Signal Flow Graphs
- Introduction to State Space (SS) Equations

Block Diagram Representations

- The transfer function relationship

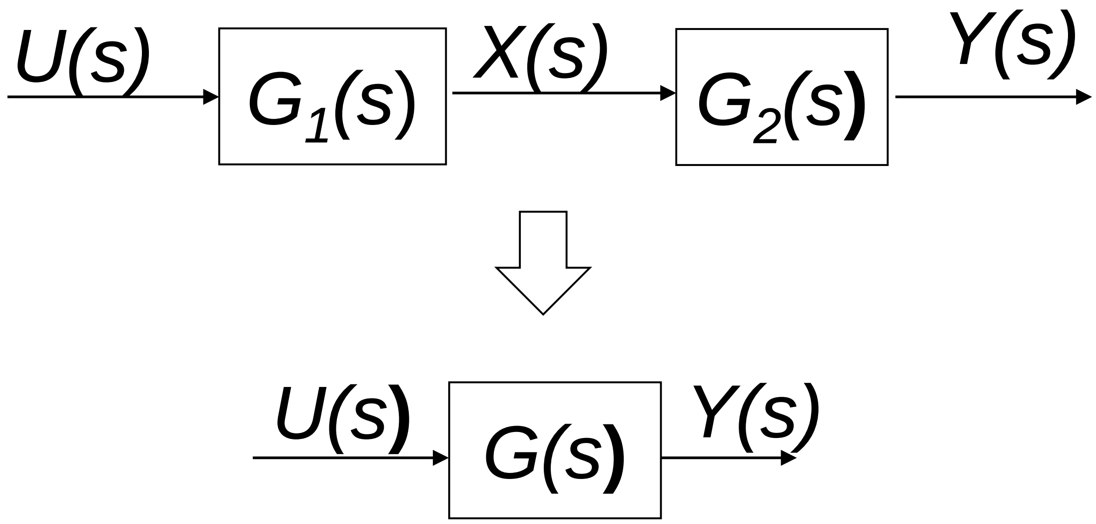
$$Y(s) = G(s)U(s)$$

can be graphically denoted through a block diagram.



Block Diagram Representations

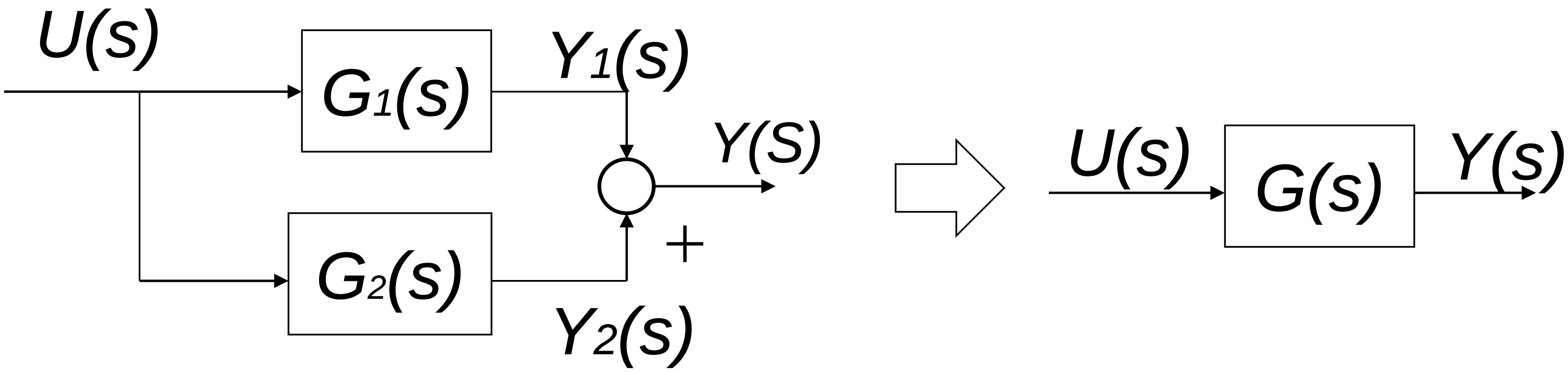
- Equivalent block diagram of two blocks in series (cascade)



$$G(s)=\frac{Y(s)}{U(s)}=\frac{X(s)}{U(s)}\cdot\frac{Y(s)}{X(s)}=G_1(s)\cdot G_2(s)$$

Block Diagram Representations

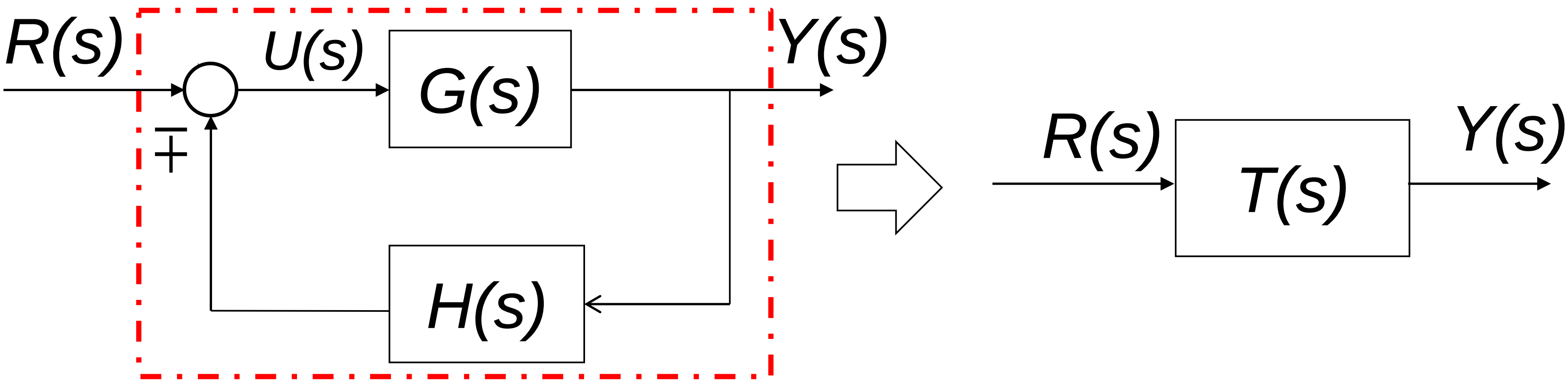
- Equivalent block diagram of two blocks in parallel



$$G(s) = \frac{Y(s)}{U(s)} = \frac{Y_1(s) + Y_2(s)}{U(s)} = G_1(s) + G_2(s)$$

Block Diagram Representations

- Equivalent block diagram of a feedback system



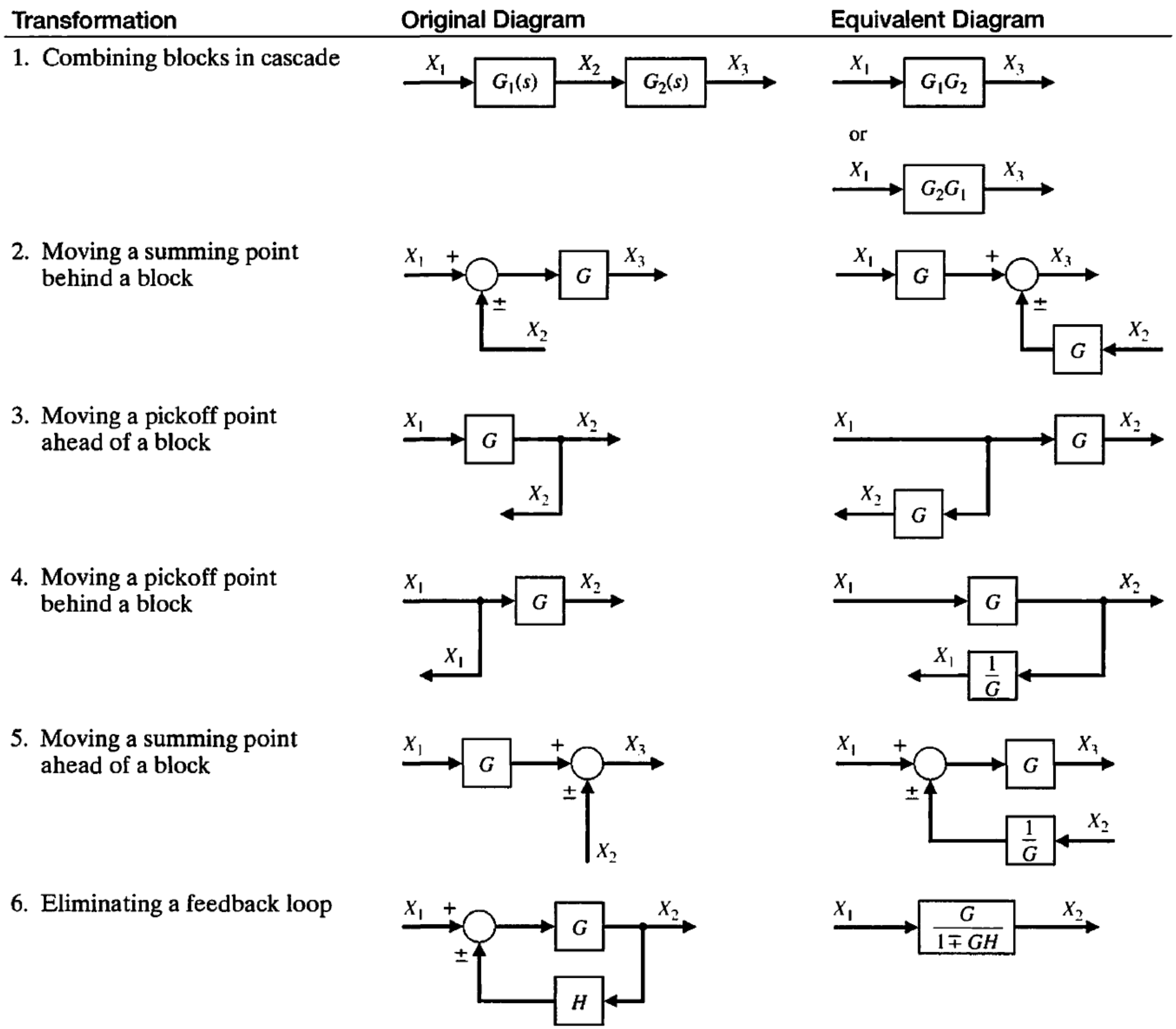
$$\begin{cases} Y(s) = U(s)G(s) \\ U(s) = R(s) - Y(s)H(s) \end{cases} \Rightarrow Y(s) = \left[R(s) - Y(s)H(s)\right]G(s)$$

$$\Downarrow$$

$$T(s) = \frac{G(s)}{1 \pm G(s)H(s)} = \frac{\text{gain of forward path}}{\text{1-loop gain}}$$

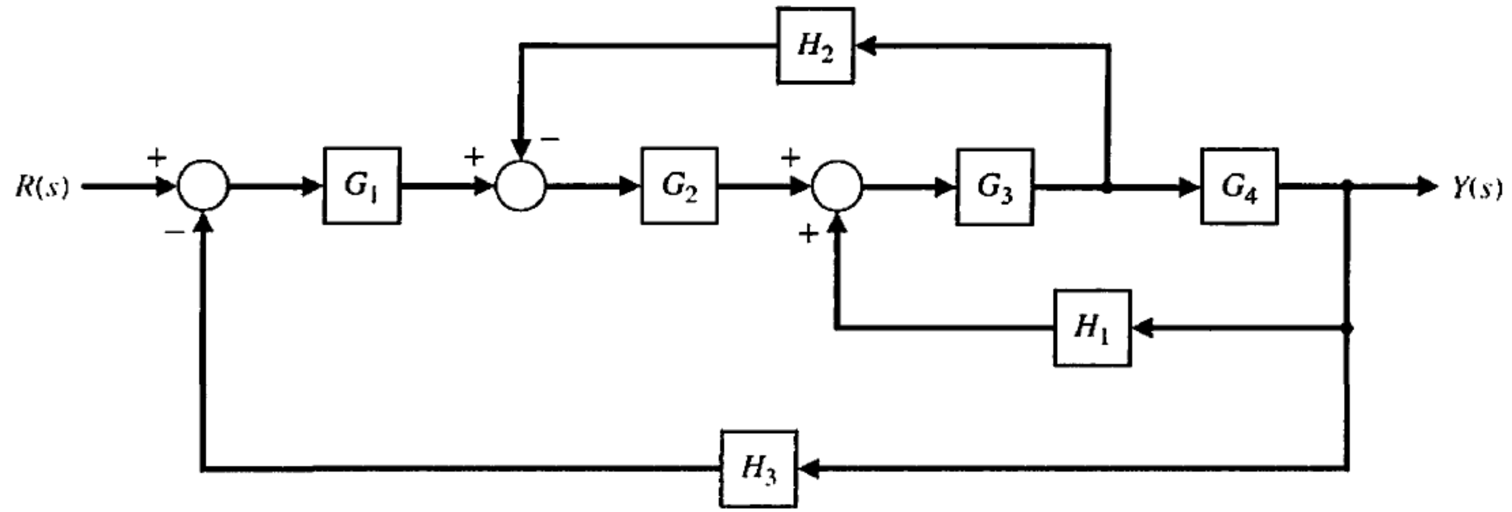
Block Diagram Representations

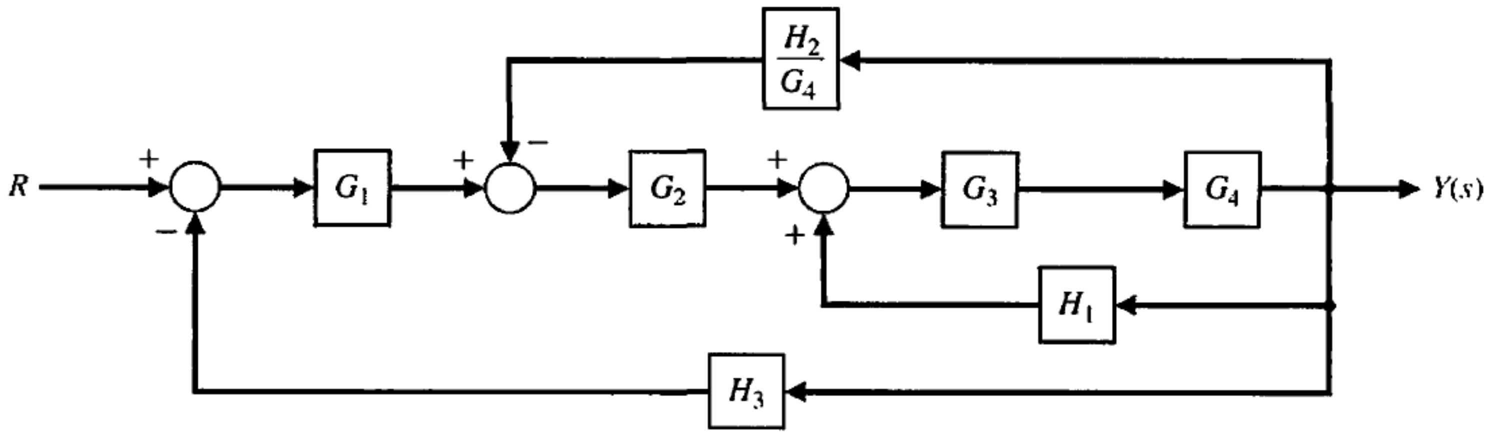
Summary



Block Diagram Representations

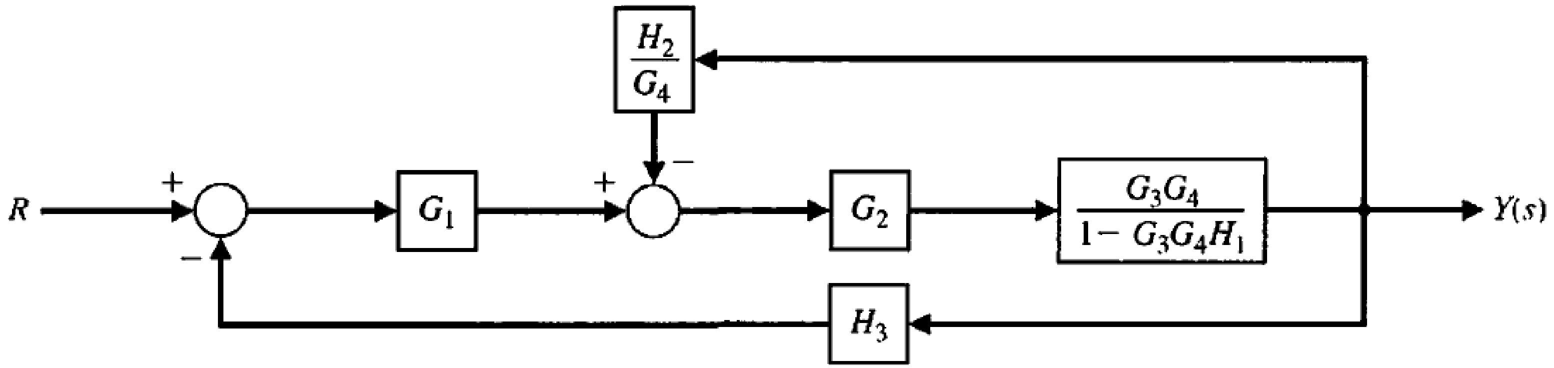
- Example

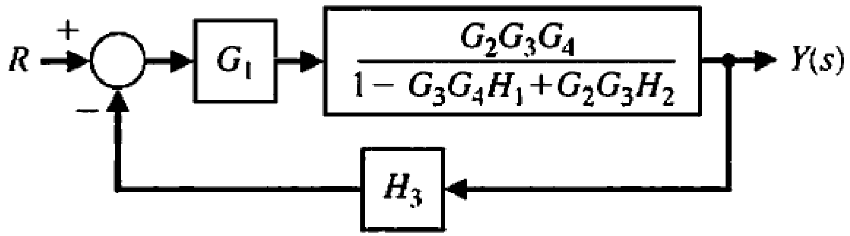


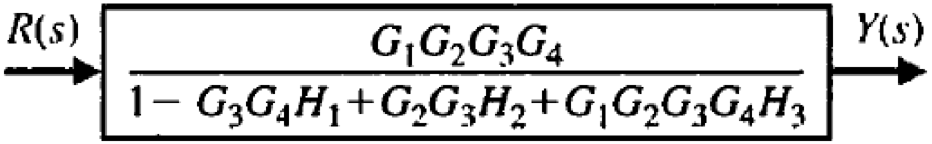


Block Diagram Representations

- Example (cont.)



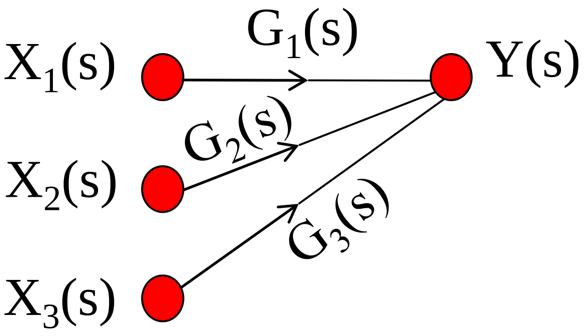




Signal Flow Graphs (SFG)

- Signal flow graphs is an alternative graphical representation of interconnected systems
- An SFG comprises:
  - Nodes: representing different signals
  - Branches: connecting between nodes having certain directions. These branches have gains representing the transfer functions of the systems acting on the signals represented by the starting nodes of the branches
- Example: The value of a signal represented by a node is the sum of all signals flowing into the node

$$Y(s) = G_1(s)X_1(s) + G_2(s)X_2(s) + G_3(s)X_3(s)$$

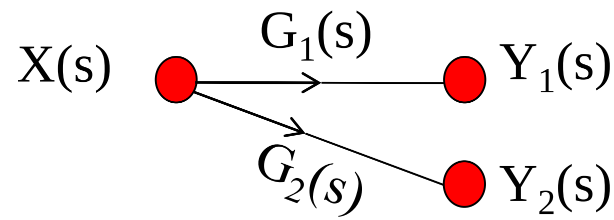


Signal Flow Graphs (SFG)

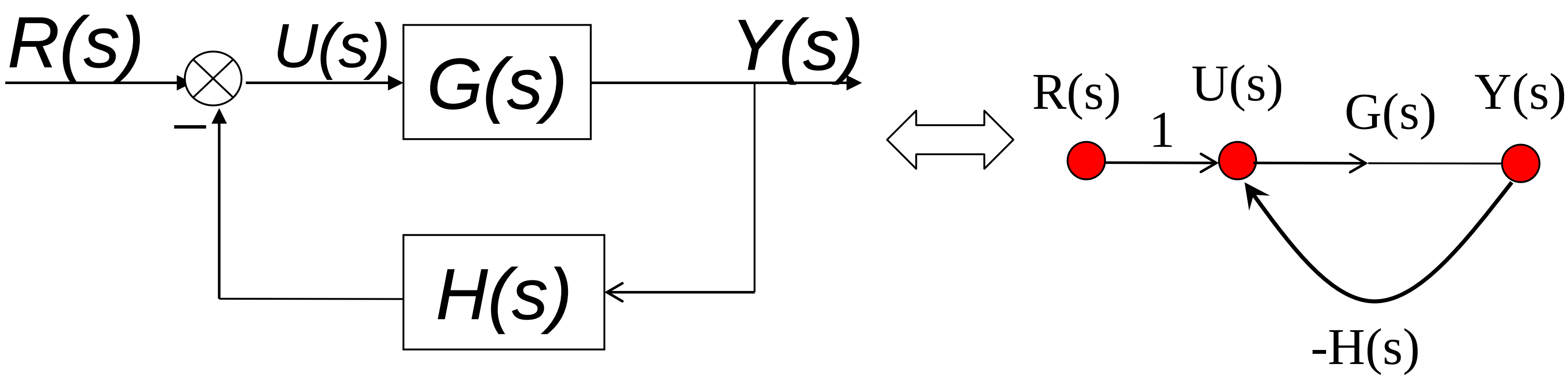
- Example: A signal is transmitted through all branches leaving node

$$Y_1(s) = G_1(s)X(s)$$

$$Y_2(s) = G_2(s)X(s)$$

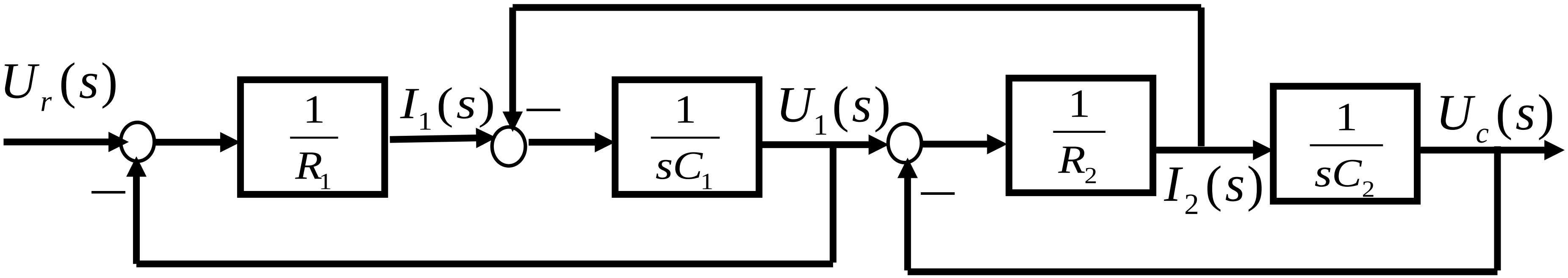


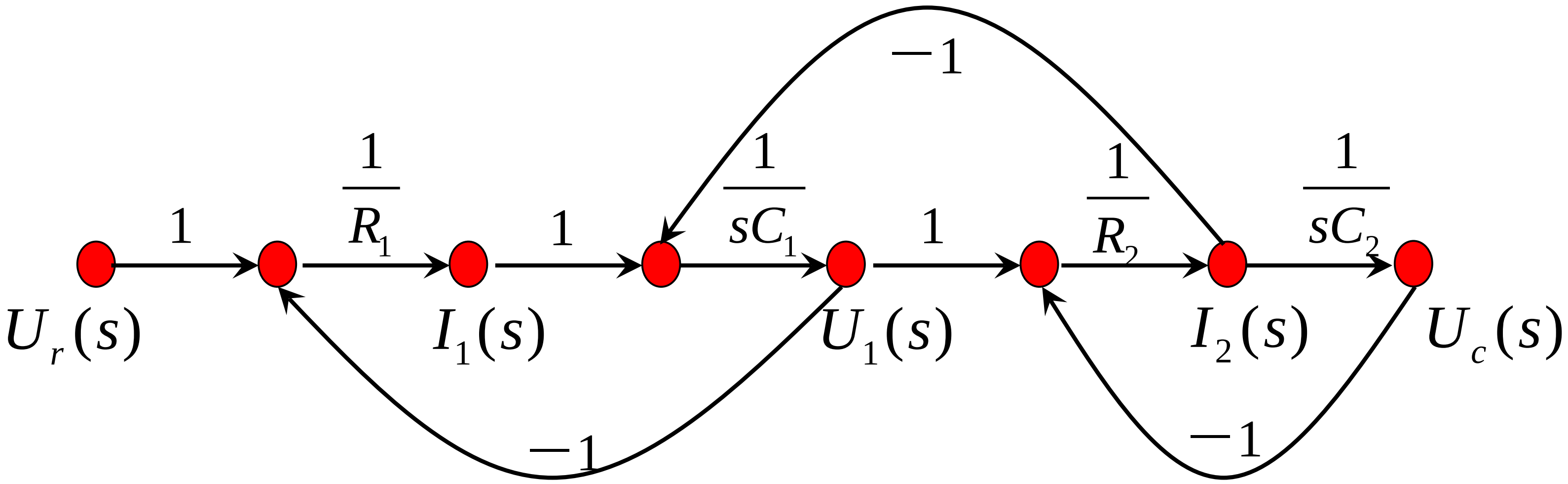
- Example: A block diagram is equivalent to a SFG



Signal Flow Graphs (SFG)

- Example: A block diagram is equivalent to a SFG





Signal Flow Graphs (SFG)

❑ Mason's Gain Formula

$$G(s)=\frac{Y(s)}{U(s)}=\frac{1}{\Delta}\sum_{k=1}^{N}G_k\Delta_k$$

N = total number of forward paths between output *Y(s)* and input *U(s)*

G_k = path gain of the *k*th forward path (product of branches traversed along the forward path)

Δ = $1-\sum($ all individual loop gains)

$+\sum($ gain products of all possible two loops that do not touch)

$-\sum($ gain products of all possible three loops that do not touch)

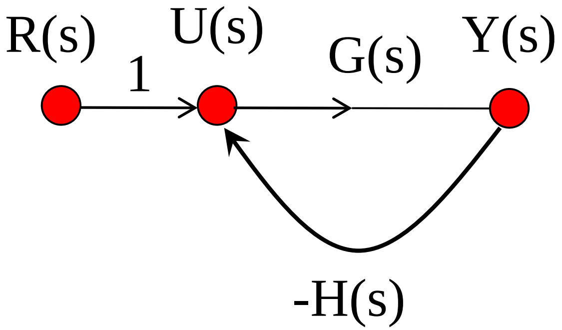
$+\cdots$

Δ_k = value of Δ for that part of the block diagram that does not touch the *k*th forward path.

Signal Flow Graphs (SFG)

- <u>Example:</u> Use Mason's formula to find overall closed loop transfer func of negative feedback sys

$$G(s)=\frac{Y(s)}{R(s)}=\frac{1}{\Delta}\sum_{k=1}^{N}G_k\Delta_k$$



1 forward path and 1 loop

$$\Delta=1-\left[G(s)\cdot -H(s)\right]=1+G(s)H(s)$$

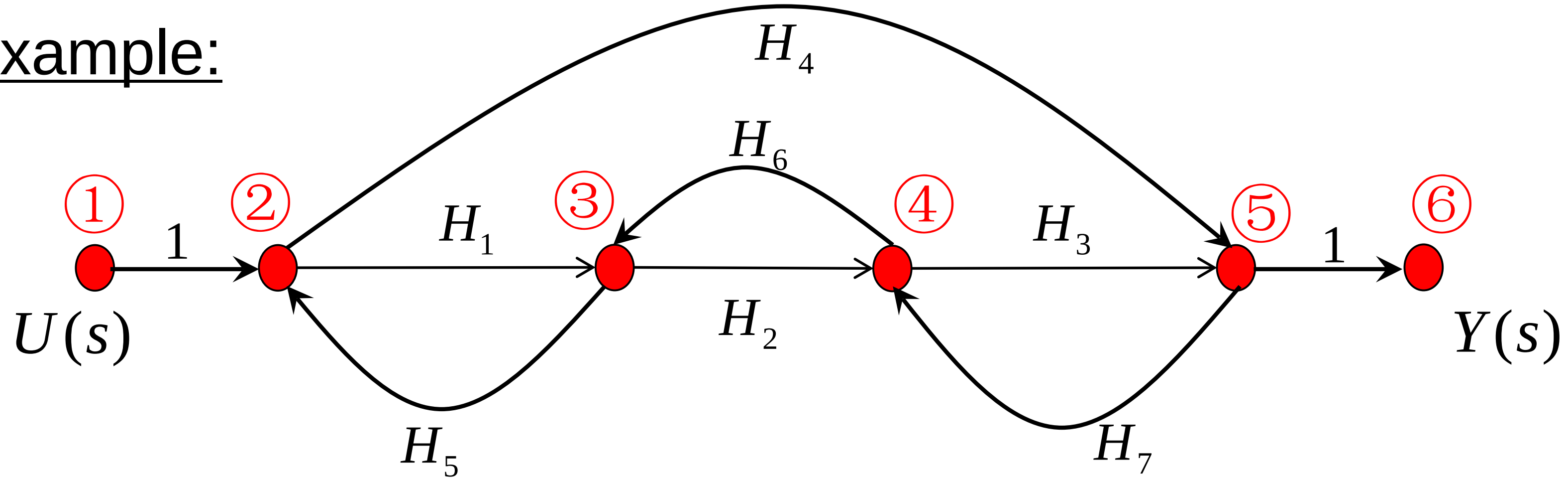
$$G_1=1\cdot G(s)=G(s)$$

$$\Delta_1=1$$

$$\therefore\frac{Y(s)}{R(s)}=\frac{G(s)}{1+G(s)H(s)}$$

Signal Flow Graphs (SFG)

- Example:



Forward path	Path gain
123456	$G_1 = H_1 H_2 H_3$
1256	$G_2 = H_4$

Loop path	Path gain
232	$l_1 = H_1 H_5$
343	$l_2 = H_2 H_6$
454	$l_3 = H_3 H_7$
25432	$l_4 = H_4 H_7 H_6 H_5$

$$\Delta = 1 - (l_1 + l_2 + l_3 + l_4) + (l_1 l_3)$$

$$\Delta_1 = 1 - 0$$

$$\Delta_2 = 1 - H_2 H_6$$

Signal Flow Graphs (SFG)

- <u>Example:</u>

Forward path	Path gain
123456	$G_1 = H_1 H_2 H_3$
1256	$G_2 = H_4$

Loop path	Path gain
232	$l_1 = H_1 H_5$
343	$l_2 = H_2 H_6$
454	$l_3 = H_3 H_7$
25432	$l_4 = H_4 H_7 H_6 H_5$

$$\Delta = 1 - \left(l_1 + l_2 + l_3 + l_4\right) + (l_1 l_3)$$

$$\Delta_1 = 1 - 0$$

$$\Delta_2 = 1 - H_2 H_6$$

$$G(s) = \frac{Y(s)}{U(s)} = \sum_{k=1}^{2} \frac{G_k \Delta_k}{\Delta}$$

$$= \frac{H_1 H_2 H_3 + H_4 \left(1 - H_2 H_6\right)}{1 - H_1 H_5 - H_2 H_6 - H_3 H_7 - H_4 H_7 H_6 H_5 + H_1 H_5 H_3 H}$$

Introduction to State Space Representation

Classical Control

- SISO
  (Single Input Single Output)
- Low order ODEs
- Time-invariant
- Transfer function based approaches (Root-Locus and Frequency domain design approaches)
- Continuous, analog
- Before 80s

Modern Control

- MIMO
  (Multiple Input Multiple Output)
- High order ODEs -> System of linear 1^{st} order DEs
- Time-invariant and time variant
- State space approach
- Discrete, digital
- 80s and after

- The difference between classical control and modern control originates from the different modeling approach used by each control
- The modeling approach used by modern control enables it to have new features and ability to control much more complicated systems compared to classical control

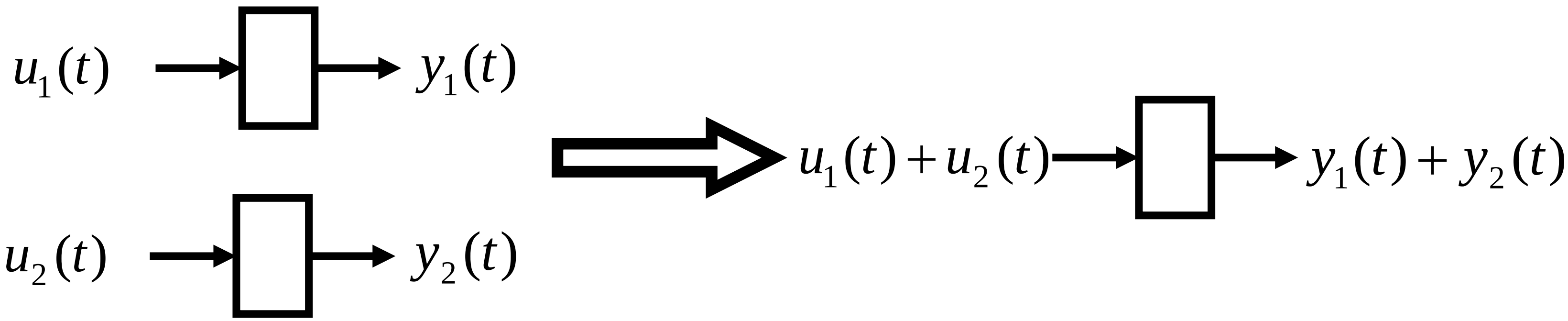
Refresher of System Classifications

- Systems are classified based on:
  - *The number of inputs and outputs*: single-input single-output (SISO), multi-input multi-output (MIMO), MISO, SIMO
  - *Existence of memory*: if the present output depends on the present input only, then the system is said to be memoryless (or static), otherwise it has memory (e.g. purely resistive circuit)
  - *Causality*: a system is called causal if the output depends only on the present and past inputs and independent of the future inputs
  - *Dimensionality*: the dimension of system can be finite (lumped) or infinite (distributed)
  - *Linearity*: a linear system satisfies the properties of superposition and homogeneity
  - *Time-Invariance*: a time-invariant (TI) system has static characteristics that do not change with time.

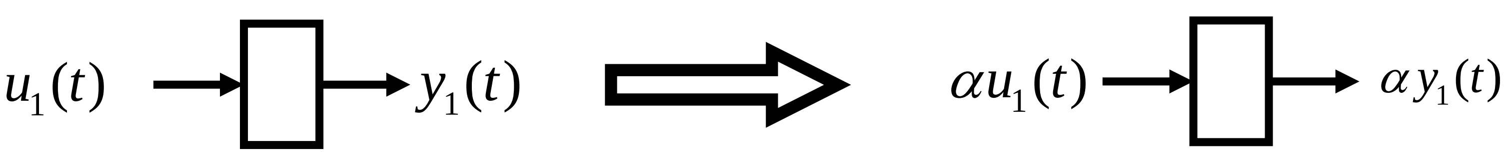
Refresher of System Classifications

- Linearity:

- A system is said to be linear if it satisfies the following two properties of superposition and homogeneity
  - Superposition



  - Homogeneity



Refresher of System Classifications

- Linearity:

Example: Check the linearity of the following system (governed by ODE).

$$u(t) \longrightarrow \boxed{y''(t)+2y'(t)+y(t)=u'(t)+3u(t)} \longrightarrow y(t)$$

Let $y_1''(t)+2y_1'(t)+y_1(t)=u_1'(t)+3u_1(t)$

$y_2''(t)+2y_2'(t)+y_2(t)=u_2'(t)+3u_2(t)$

Then $[\alpha u_1(t)+\beta u_2(t)]'+3[\alpha u_1(t)+\beta u_2(t)]$

$$=\alpha u_1'(t)+\beta u_2'(t)+\alpha 3u_1(t)+\beta 3u_2(t)$$

$$=\alpha[u_1'(t)+3u_1(t)]+\beta[u_2'(t)+3u_2(t)]$$

$$=\alpha[y_1''(t)+2y_1'(t)+y_1(t)]+\beta[y_2''(t)+2y_2'(t)+y_2(t)]$$

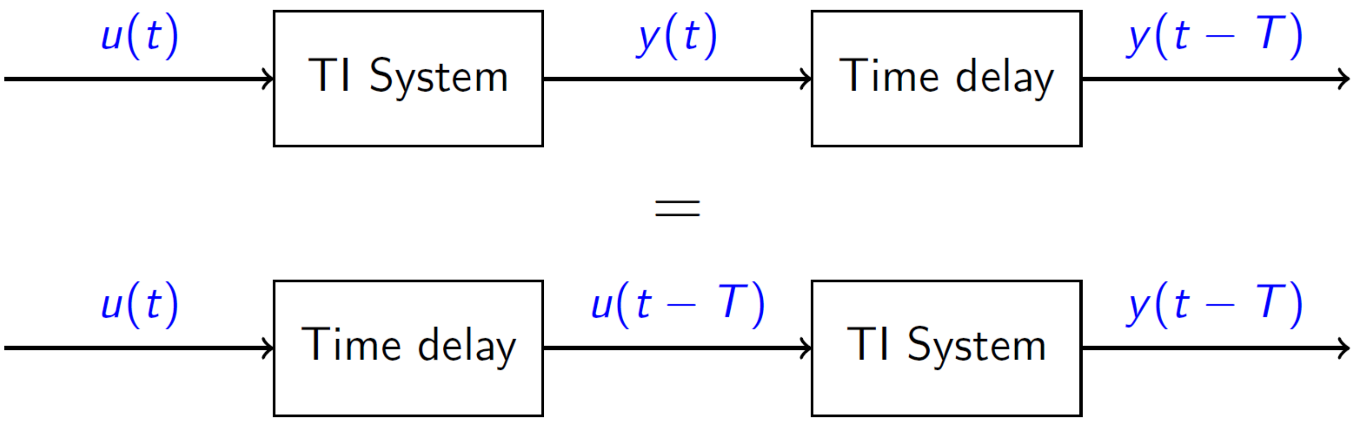
$$=[\alpha y_1(t)+\beta y_2(t)]''+2[\alpha y_1(t)+\beta y_2(t)]'+[\alpha y_1(t)+\beta y_2(t)]$$

→ The system is **linear**

Refresher of System Classifications

- Time Invariance:

- A system is said to be time-invariant if it commutes with time delays. In other words, the output to a time delayed input is the same as the time delayed output to the original input



- A system time-invariant if its parameters do not change over time.

Refresher of System Classifications

- Causality

- A system is causal if the output at time t_0 depends only on the input up to and including time t_0
- A causal system has memory
- A causal system has the following condition on its impulse response

$$h(t) = 0 \qquad \forall t < 0$$

State Space Equations

- We know that for a causal system, in order to compute the output at a given time t_0, we need to know the input signal over $(-\infty, t_0] \rightarrow$ "A lot of information!!"
- The central question is: "Is there a more manageable way to express the entire memory or history of the past inputs to the system?"

State Variables

They are the set of variables $x_1(t), x_2(t), x_3(t), \ldots x_n(t)$ such that the knowledge of these variables at time t_1 together with the input u between times t_1 and t_2 is sufficient to uniquely evaluate: a) the output at time t_2, and b) the updated state variables at t_2

- In other words, these state variables is the concise way to summarize the entire history of past inputs to the system, i.e. knowledge of present values of the state variables as well as present and future inputs is sufficient to obtain future output
- The state of the system is usually a vector **x**[t] in an *n*-dimensional space $\mathbb{R}^n$

State Space Equations

- There choice of state variables of a system is not unique, i.e. there are infinite choices (realizations) of the same system
- What is key is the minimum number of state variables needed to fully represent or realize the system which is called “minimal realizations”

Dimension of a system

It is the minimum number of state variables sufficient to describe the system state

- Throughout the course, We deal only with finite dimensional systems
- Mostly systems with lumped elements are finite dimensional whereas systems with distributed component models are infinite dimensional (e.g. transmission lines)

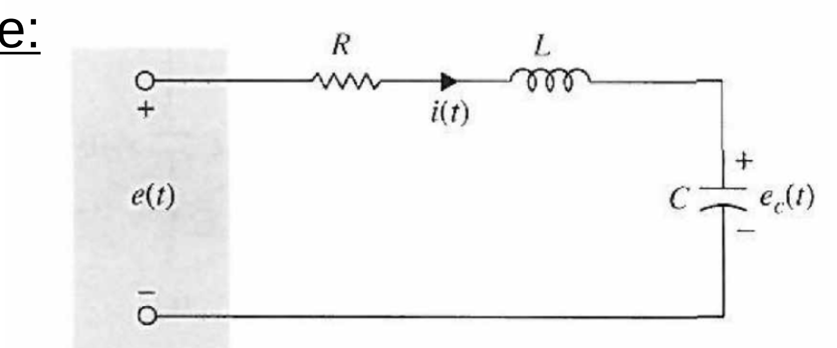
State Space Equations

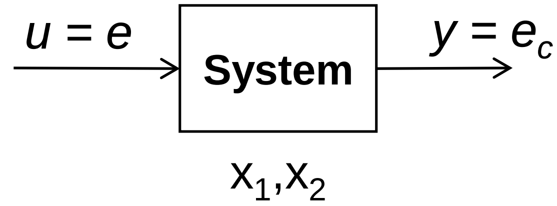
The key idea in state space equations is that you break the n^{th} order Diff eq. that represents a system into n 1st order Diff. equations

- Since the entire SS formulation involve matrices that describe the system behavior, it is naturally suited for multiple-input multiple-output (MIMO) systems compared to transfer function models
- Matrices mean **a lot of Linear Algebra!!!!**
- Let's start with an example showing how an 2nd order Diff. eq. is broken into two 1st order Diff. Eqs giving us the SS model before getting into the general SS formulation

State Space Equations

Example:





$$e(t) = Ri(t) + L\frac{di}{dt} + e_c(t) \quad \Leftarrow \quad i(t) = C\frac{de_c(t)}{dt}$$

$$e(t) = RC\frac{de_c(t)}{dt} + LC\frac{d^2e_c(t)}{dt^2} + e_c(t)$$

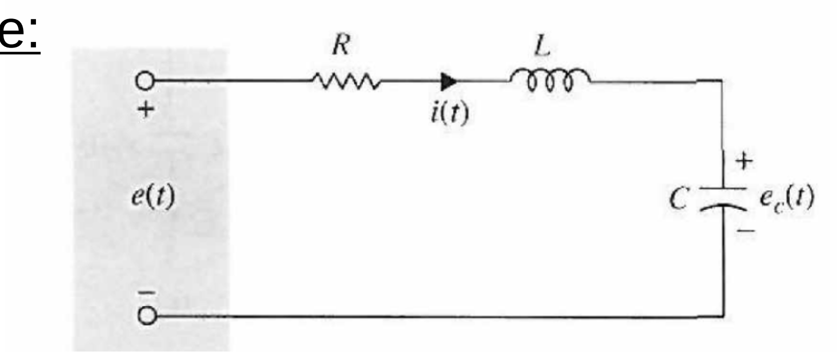
$$\boxed{LC\ddot{e}_c + RC\dot{e}_c + e_c = e}$$

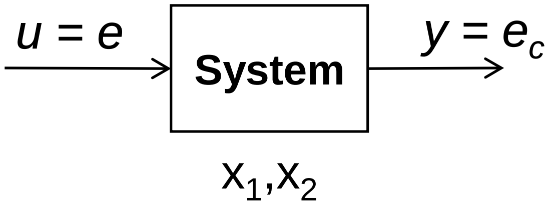
Let us take a possible state variable assignment

$$\boxed{x_1 = e_c \quad and \quad x_2 = \dot{e}_c}$$

State Space Equations

Example:





$$LC\ddot{e}_c + RC\dot{e}_c + e_c = e$$

Let us take a possible state variable assignment

$$x_1 = e_c \quad and \quad x_2 = \dot{e}_c$$

Then

$$\dot{x}_1 = \dot{e}_c = x_2$$

$$\dot{x}_2 = \ddot{e}_c = -\frac{R}{L}\dot{e}_c - \frac{1}{LC}e_c + \frac{1}{LC}e$$

$$= -\frac{R}{L}x_2 - \frac{1}{LC}x_1 + \frac{1}{LC}u$$

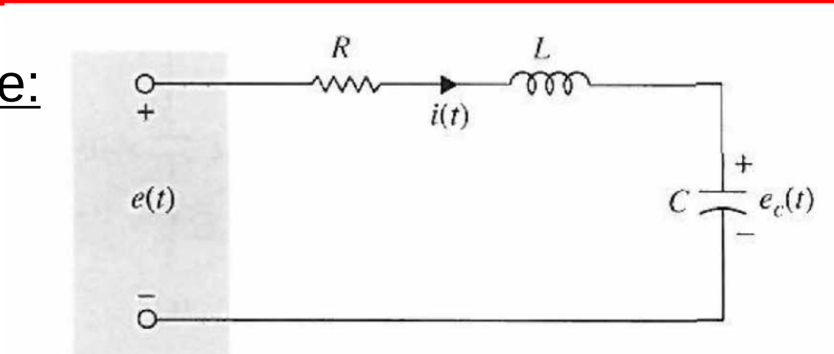
$$\dot{x}_1 = x_2$$

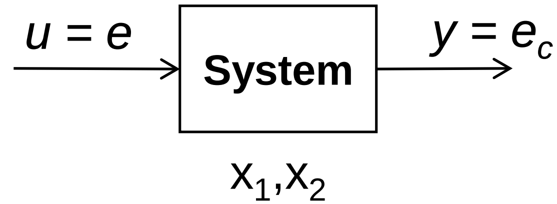
$$\dot{x}_2 = -\frac{R}{L}x_2 - \frac{1}{LC}x_1 + \frac{1}{LC}u$$

System of two 1st order Diff. Eqs.

State Space Equations

Example:





$$x_1 = e_c \quad and \quad x_2 = \dot{e}_c$$

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -\frac{R}{L}x_2 - \frac{1}{LC}x_1 + \frac{1}{LC}u$$

In matrix form

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

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

- System of two coupled 1st order Diff. Eqs. describing the dynamic behavior of state variables
- Another equation to calculate the output in terms of the state variables

State Space Equations

- General case: For an n^{th} order Diff. Eq. of SISO system

$$\frac{d^n y(t)}{dt^n} + a_1 \frac{d^{n-1} y(t)}{dt^{n-1}} + a_2 \frac{d^{n-2} y(t)}{dt^{n-2}} + \ldots + a_n y(t) = u(t)$$

Using the following state variables assignments (*n* state variables)

$$x_1(t) = y(t), \quad x_2(t) = \frac{dy(t)}{dt}, \quad x_3(t) = \frac{d^2 y(t)}{dt^2}, \quad x_n(t) = \frac{d^{n-1} y(t)}{dt^{n-1}}$$

Then $\quad \dot{x}_1 = x_2, \quad \dot{x}_2 = x_3, \quad \dot{x}_3 = x_4$

$$\dot{x}_n = \frac{d^n y(t)}{dt^n}$$

$$= -a_1 \frac{d^{n-1} y(t)}{dt^{n-1}} - a_2 \frac{d^{n-2} y(t)}{dt^{n-2}} - \ldots - a_n y(t) + u(t)$$

State Space Equations

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = x_3, \quad \dot{x}_3 = x_4$$

$$\dot{x}_n = \frac{d^n y(t)}{dt^n} = -a_1 \frac{d^{n-1} y(t)}{dt^{n-1}} - a_2 \frac{d^{n-2} y(t)}{dt^{n-2}} - \ldots - a_n y(t) + u(t)$$

$$= -a_1 x_n - a_2 x_{n-1} - \ldots - a_n x_1 + u(t)$$

In Matrix form

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \vdots \\ \dot{x}_n(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -a_n & -a_{n-1} & \cdots & -a_3 & -a_2 & -a_1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ \vdots \\ x_n(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} + 0 \cdot u(t)$$

State Space Equations

For an *n* dimensional system with *p* inputs and *m* outputs

$$\underset{n\times 1}{\dot{\mathbf{x}}(t)} = \underset{n\times n}{\mathbf{A}}\,\underset{n\times 1}{\mathbf{x}(t)} + \underset{n\times p}{\mathbf{B}}\,\underset{p\times 1}{\mathbf{u}(t)}$$

$$\underset{m\times 1}{\mathbf{y}(t)} = \underset{m\times n}{\mathbf{C}}\,\underset{n\times 1}{\mathbf{x}(t)} + \underset{m\times p}{\mathbf{D}}\,\underset{p\times 1}{\mathbf{u}(t)}$$

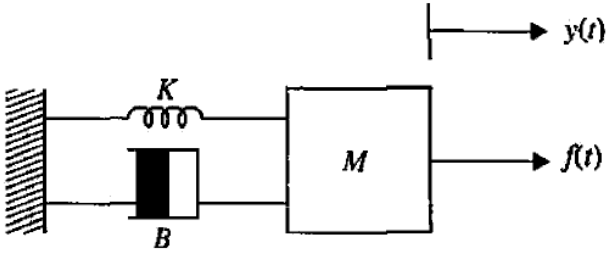
- $t \in \mathbb{R}$ denotes time
- $\mathbf{x} \in \mathbb{R}^n$ denotes the state vector
- $\mathbf{u} \in \mathbb{R}^p$ denotes the input vector
- $\mathbf{y} \in \mathbb{R}^m$ denotes the output vector

- $\mathbf{A} \in \mathbb{R}^{n\times n}$ denotes the system dynamic matrix
- $\mathbf{B} \in \mathbb{R}^{n\times p}$ denotes the input matrix
- $\mathbf{C} \in \mathbb{R}^{m\times n}$ denotes the output or sensor matrix
- $\mathbf{D} \in \mathbb{R}^{m\times p}$ denotes the feedthrough matrix

- For LTI systems, the matrices A,B,C and D are all constant, i.e. not f(*t*)
- For time variant systems → A(*t*), B(*t*), C(*t*), D(*t*)

State Space Equations

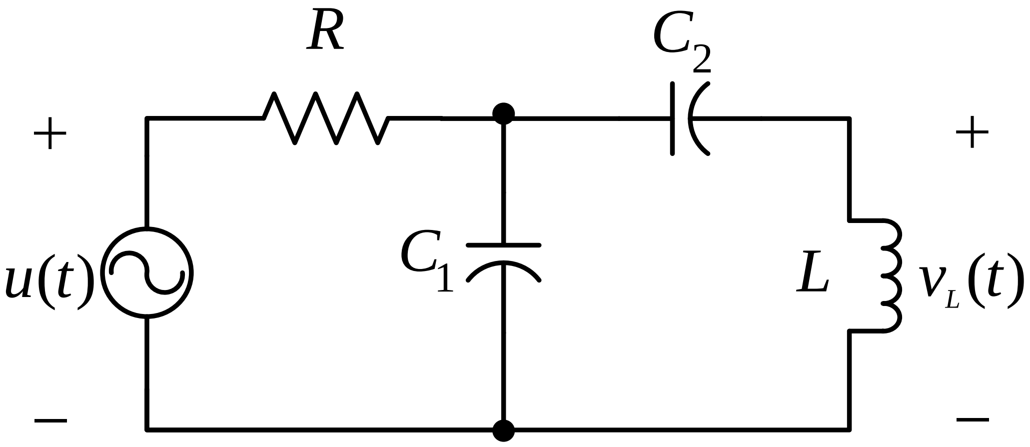
Problem 1: Find the SS formulation for the mass-spring-damper system with input *f* (applied force) and output *y* (displacement)



$$M\ddot{y} + B\dot{y} + Ky = f$$

State Space Equations

Problem 2: Find the SS formulation for the following circuit with input *u* (input voltage) and output v_L (inductor voltage)



State Space Equations

<u>Problem 3</u>: Find the SS formulation for the following system whose i/o relationship is given by the following Diff. Eq.

$$\dddot{y} + 2\ddot{y} + 3\dot{y} + y = \ddot{u} - \dot{u} + u$$