

State Space Modeling and Analysis

Basic Concepts related to State Space

- **State**

State Variables

The smallest set of variables $\{q_1, q_2, \dots, q_n\}$ such that the knowledge of these variables at time $t = t_0$, together with the knowledge of the input for $t \geq t_0$ completely determines the behavior (the values of the state variables) of the system for time $t \geq t_0$.

- **State Vector**

All the state variables $\{q_1, q_2, \dots, q_n\}$ can be looked on as components of state vector.

- **State Space**

A space whose coordinates consist of state variables is called a state space. Any state can be represented by a point in state space.

State Space Representation

Introduction:

- Two parts:
 - A set of first order ODEs that represents the derivative of each state variable x_i as an algebraic function of the set of state variables $\{x_i\}$ and the inputs $\{u_i\}$.

$$\begin{cases} \dot{q}_1 = f_1(x_1, x_2, x_3, \dots, x_n, u_1, u_2, u_3, \dots, u_m) \\ \dot{q}_2 = f_2(x_1, x_2, x_3, \dots, x_n, u_1, u_2, u_3, \dots, u_m) \\ \vdots \\ \dot{q}_n = f_n(x_1, x_2, x_3, \dots, x_n, u_1, u_2, u_3, \dots, u_m) \end{cases} \quad \dot{x} = \mathbf{A}x + \mathbf{B}u$$

- A set of equations that represents the output variables as algebraic functions of the set of state variables $\{x_i\}$ and the inputs $\{u_i\}$.

$$\begin{cases} y_1 = f_1(x_1, x_2, x_3, \dots, x_n, u_1, u_2, u_3, \dots, u_m) \\ y_2 = f_2(x_1, x_2, x_3, \dots, x_n, u_1, u_2, u_3, \dots, u_m) \\ \vdots \\ y_n = f_n(x_1, x_2, x_3, \dots, x_n, u_1, u_2, u_3, \dots, u_m) \end{cases} \quad y = \mathbf{C}x + \mathbf{D}u$$

Differential Equations and State Space Representation

$$\ddot{y}(t) + a_1 \dot{y}(t) + a_0 y(t) = b_0 v(t)$$

- A second-order differential equation requires two state variables:

$$x_1(t) = y(t) \quad x_2(t) = \dot{y}(t)$$

- We can reformulate the differential equation as a set of three equations:

$$\dot{x}_1(t) = x_2(t)$$

$$\dot{x}_2(t) = -a_0 x_1(t) - a_1 x_2(t) + b_0 v(t)$$

$$y(t) = x_1(t)$$

- We can write these in matrix form as:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ b_0 \end{bmatrix} v(t)$$

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

- This can be extended to an N^{th} -order differential equation of this type: $y^{(N)}(t) + \sum_{i=0}^{N-1} a_i y^{(i)}(t) = b_0 v(t)$

- The state variables are defined as: $x_i(t) = y^{(i-1)}(t), \quad i = 1, 2, \dots, N$

$$\begin{aligned} \dot{X} &= AX + Bu \\ y &= CX + Du \end{aligned}$$

Differential Equations and State Space Representation

- The resulting state equations are:

$$\dot{x}_1(t) = x_2(t)$$

$$\dot{x}_2(t) = x_3(t)$$

⋮

$$\dot{x}_{N-1}(t) = x_N(t)$$

$$\dot{x}_N(t) = -\sum_{i=0}^{N-1} a_i x_{i+1}(t) + b_0 v(t)$$

$$y(t) = x_1(t)$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{N-1} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ b_0 \end{bmatrix}$$

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

$$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}u$$

$$y = \mathbf{C}\mathbf{X} + \mathbf{D}u$$

- Next, consider a differential equation with a more complex forcing function:

$$\ddot{y}(t) + a_1 \dot{y}(t) + a_0 y(t) = b_1 \dot{v}(t) + b_0 v(t)$$

- The state model is:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v(t) \quad y(t) = [b_0 \quad b_1] \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

- We can verify this by expanding the matrix equation:

$$\dot{x}_1(t) = x_2(t)$$

$$\dot{x}_2(t) = -a_0 x_1(t) - a_1 x_2(t) + v(t)$$

$$y(t) = b_0 x_1(t) + b_1 x_2(t)$$

Differential Equations and State Space Representation

- To construct the original equation, differentiate the last equation:

$$\begin{aligned} \dot{y}(t) &= b_0 \dot{x}_1(t) + b_1 \dot{x}_2(t) \\ &= b_0 x_2(t) + b_1 [-a_0 x_1(t) - a_1 x_2(t) + v(t)] \\ &= -a_1 y(t) + (a_1 b_0 - a_0 b_1) x_1(t) + b_0 x_2(t) + b_1 v(t) \end{aligned}$$

- Differentiate the last equation again and substitute:
- Hence, given a general LTI system:

$$\begin{aligned} \ddot{y}(t) &= -a_1 \dot{y}(t) + (a_1 b_0 - a_0 b_1) \dot{x}_1(t) + b_0 \dot{x}_2(t) + b_1 \dot{v}(t) \\ &= -a_1 \dot{y}(t) + (a_1 b_0 - a_0 b_1) x_2(t) \\ &\quad + b_0 [-a_0 x_1(t) - a_1 x_2(t) + v(t)] + b_1 \dot{v}(t) \\ &= -a_1 \dot{y}(t) - b_0 a_0 x_1(t) - a_0 b_1 x_2(t) + b_0 v(t) + b_1 \dot{v}(t) \\ &= -a_1 \dot{y}(t) - a_0 (b_0 x_1(t) + b_1 x_2(t)) + b_0 v(t) + b_1 \dot{v}(t) \\ &= -a_1 \dot{y}(t) - a_0 y(t) + b_0 v(t) + b_1 \dot{v}(t) \end{aligned}$$

$$\ddot{y}(t) + a_1 \dot{y}(t) + a_0 y(t) = b_0 v(t) + b_1 \dot{v}(t)$$

$$y^{(N)}(t) + \sum_{i=0}^{N-1} a_i y^{(i)}(t) = \sum_{i=0}^{N-1} b_i v^{(i)}(t)$$

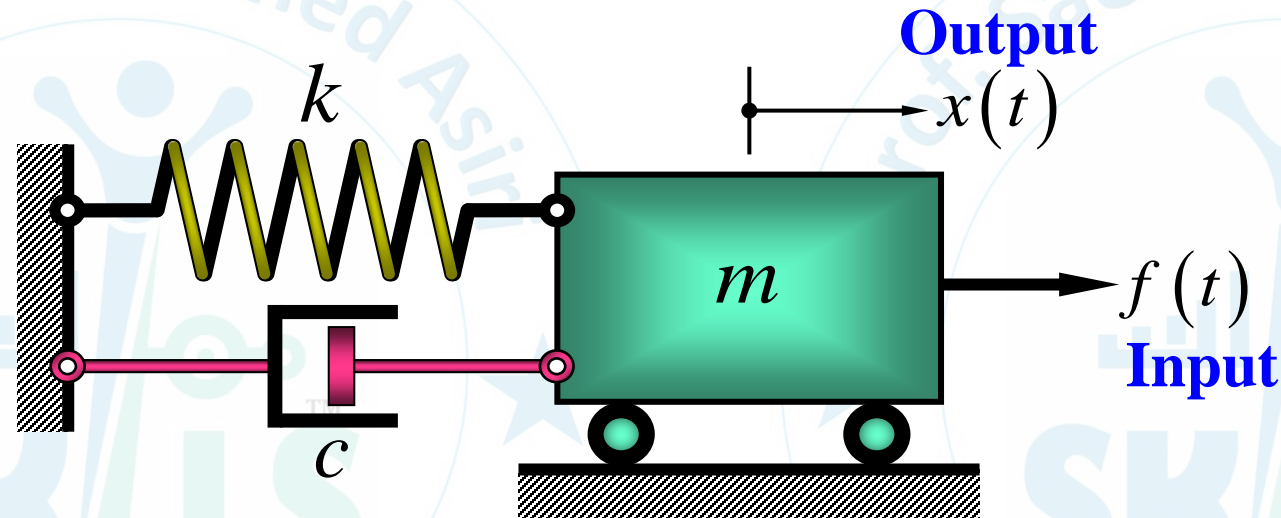
$$\begin{aligned} \dot{X} &= AX + Bu \\ y &= CX + Du \end{aligned}$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{N-1} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{C} = [b_0 \quad b_1 \quad b_2 \quad \dots \quad b_N] \quad \mathbf{D} = 0$$

State Space Representation

Introduction:



$$m \ddot{x} + c \dot{x} + k x = F$$

$F = \text{Control force}$

State Space Representation

Introduction:

Let $x_1 = x$, $x_2 = \dot{x}$

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}}_{\dot{\mathbf{X}}} = \underbrace{\begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{\mathbf{X}} + \underbrace{\begin{bmatrix} 0 \\ 1/m \end{bmatrix}}_{\mathbf{B}} \underbrace{F}_{u}$$

State Space Representation

Introduction:

$$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}u$$

where

$$\mathbf{X} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}, \quad \dot{\mathbf{X}} = \begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 1/m \end{bmatrix}, \quad u = F$$

State Space Representation

Introduction:

$$\begin{aligned}\dot{\mathbf{X}} &= \mathbf{A}\mathbf{X} + \mathbf{B}u \\ \mathbf{y} &= \mathbf{C}\mathbf{X} + \mathbf{D}u\end{aligned}$$

where

\mathbf{X} = state vector, \mathbf{y} = Output vector, u = control vector

\mathbf{A} = system matrix, \mathbf{B} = Input matrix

\mathbf{C} = Output matrix, \mathbf{D} = Feedforward Matrix

State Space Representation

Introduction:

Output $y = x$



$$y = CX + Du$$

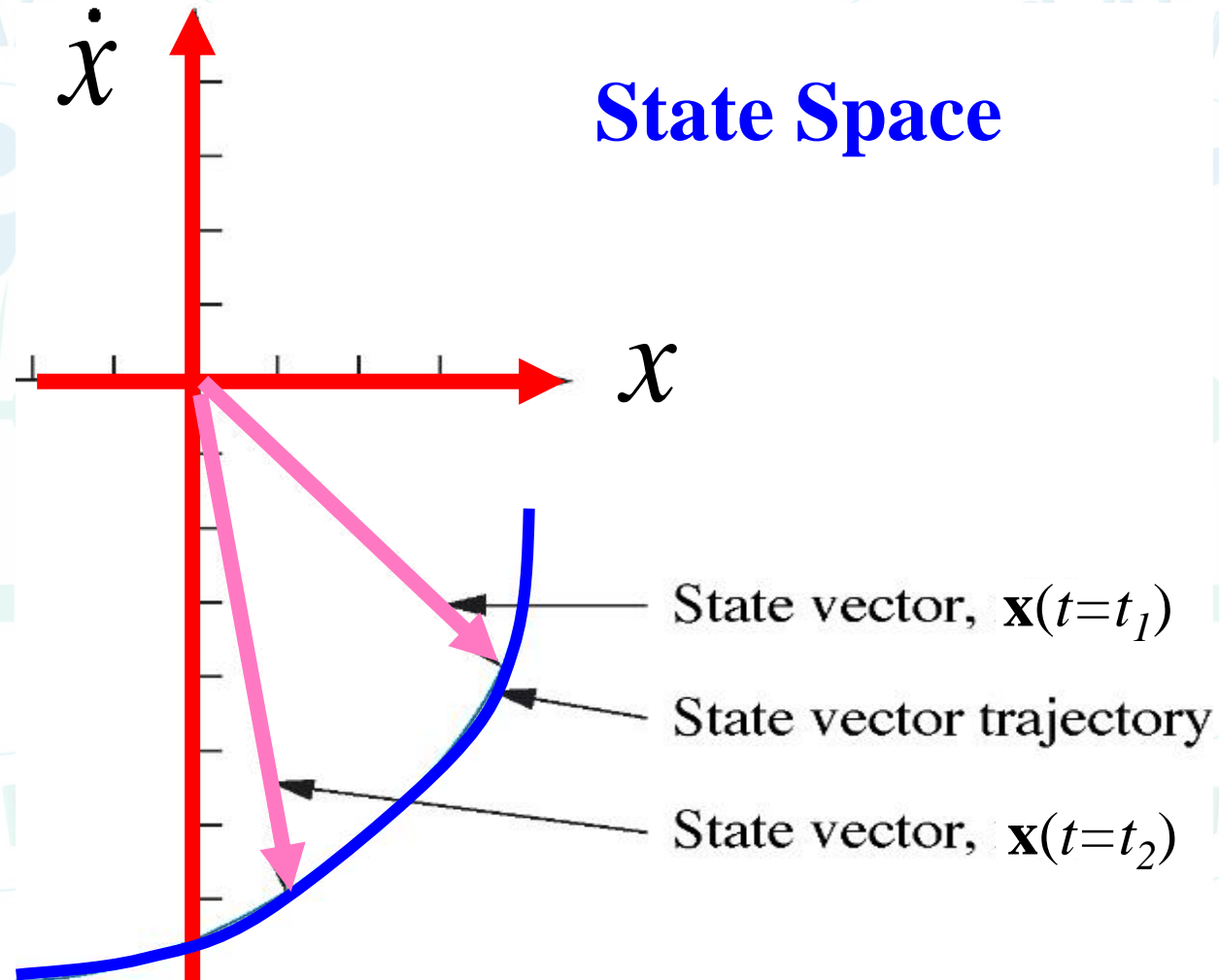
where

$$\mathbf{X} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}, \quad C = [1 \quad 0], \quad D = 0, \quad u = F$$

**Displacement
Sensor**

State Space Representation

Introduction:



State Space Representation

Introduction:

System Equation
$$L \frac{di}{dt} + Ri + \frac{1}{C} \int i dt = v$$

Output Equation
$$v_L = L \frac{di}{dt} = -Ri - \frac{1}{C} \int i dt + v$$

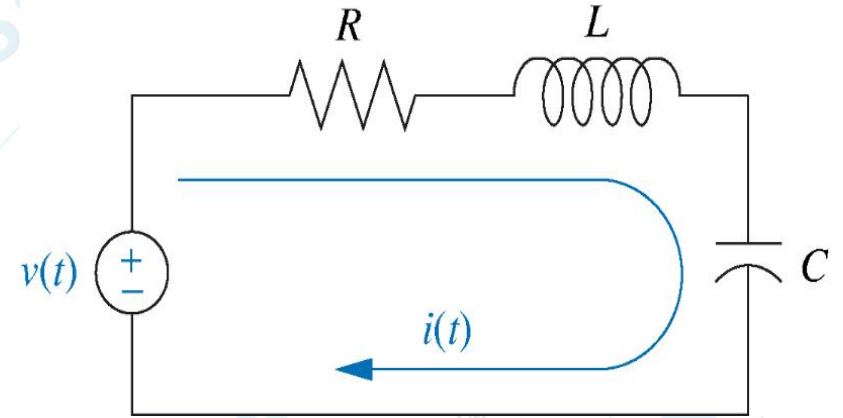
$$L \frac{di}{dt} + Ri + \frac{1}{C} \int i dt = v$$

Let

$$\frac{dq}{dt} = i \Rightarrow L \frac{di}{dt} + Ri + \frac{1}{C} q = v$$

$$L \frac{di}{dt} = -\frac{1}{C} q - Ri + v$$

$$\frac{di}{dt} = -\frac{1}{LC} q - \frac{R}{L} i + \frac{1}{L} v$$



State Space Representation

Introduction:

$$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}u$$

where $\mathbf{X} = \begin{bmatrix} q \\ i \end{bmatrix}$, $\dot{\mathbf{X}} = \begin{bmatrix} \frac{dq}{dt} \\ \frac{di}{dt} \end{bmatrix}$, $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -1/LC & -R/L \end{bmatrix}$,

$$\mathbf{B} = \begin{bmatrix} 0 \\ 1/L \end{bmatrix}, u = v$$

$$\frac{dq}{dt} = i$$

$$\frac{di}{dt} = -\frac{1}{LC}q - \frac{R}{L}i + \frac{1}{L}v$$

State Space Representation

Introduction:

Output Equation

$$v_L = L \frac{di}{dt} = -Ri - \frac{1}{C} \int i dt + v$$

$$v_L = -\frac{1}{C} q - Ri + v$$

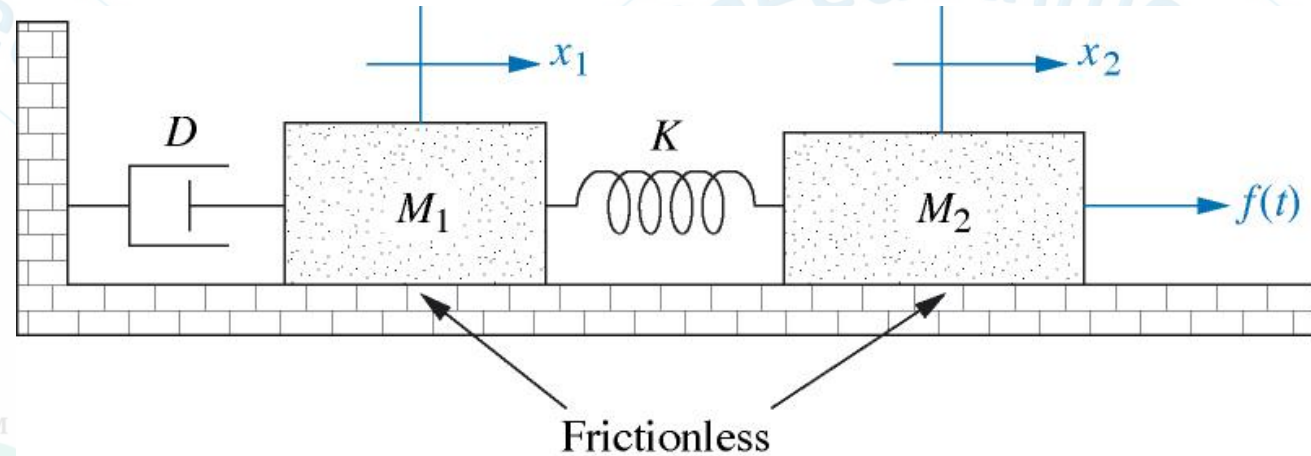
where

$$\mathbf{y} = \mathbf{C}\mathbf{X} + \mathbf{D}u$$

$$\mathbf{X} = \begin{bmatrix} q \\ i \end{bmatrix}, \mathbf{C} = [-1/C - R], \mathbf{D} = 1, u = v$$

State Space Representation

Example:



Equations of Motion

$$M_1 \ddot{x}_1 + D \dot{x}_1 + Kx_1 - Kx_2 = 0$$

$$-Kx_1 + M_2 \ddot{x}_2 + 0 \dot{x}_2 + Kx_2 = f$$

State Space Representation

Example:

Let $X = [x_1 \quad \dot{x}_1 \quad x_2 \quad \dot{x}_2]^T$

$$\begin{bmatrix} \dot{x}_1 \\ \ddot{x}_1 \\ \dot{x}_2 \\ \ddot{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ -K/M_1 & -D/M_1 & K/M_1 & 0 \\ 0 & 0 & 0 & 1 \\ K/M_2 & 0 & -K/M_2 & 0 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} x_1 \\ \dot{x}_1 \\ x_2 \\ \dot{x}_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/M_2 \end{bmatrix}}_{\mathbf{B}} f$$

State Space Representation

Example:

$$y = x_1 = [1 \quad 0 \quad 0 \quad 0] \begin{bmatrix} x_1 \\ \dot{x}_1 \\ x_2 \\ \dot{x}_2 \end{bmatrix} + 0 \times f$$

**Displacement
Sensor**

$$= C X + D u$$

where $C = [1 \quad 0 \quad 0 \quad 0]$ and $D = 0$

Transfer Function to State Space

System Equation

$$\frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \dots + a_1 \frac{dy}{dt} + a_0 y = b_0 u$$

Let $x_1 = y, x_2 = \frac{dy}{dt}, x_3 = \frac{d^2 y}{dt^2}, \dots, x_n = \frac{d^{n-1} y}{dt^{n-1}}$

→ $\dot{x}_1 = \frac{dy}{dt}, \dot{x}_2 = \frac{d^2 y}{dt^2}, \dot{x}_3 = \frac{d^3 y}{dt^3}, \dots, \dot{x}_n = \frac{d^n y}{dt^n}$

or $\dot{x}_1 = x_2, \dot{x}_2 = x_3, \dots, \dot{x}_{n-1} = x_n,$

$$\dot{x}_n = -a_0 x_1 - a_1 x_2 - \dots - a_{n-1} x_n + b_0 u$$

Transfer Function to State Space

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ -a_0 & -a_1 & -a_2 & -a_3 & -a_4 & -a_5 & \dots & -a_{n-1} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_{n-1} \\ x_n \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \dots \\ 0 \\ b_0 \end{bmatrix}}_{\mathbf{B}} u$$

$$y = \underbrace{[1 \ 0 \ 0 \ \dots \ 0]}_{\mathbf{C}} \begin{bmatrix} x_1 & x_2 & x_3 & \dots & x_{n-1} & x_n \end{bmatrix}^T$$

Transfer Function to State Space

Example:

$$\frac{C}{R} = \frac{24}{s^3 + 9s^2 + 26s + 24}$$



$$\ddot{c} + 9\dot{c} + 26c = 24r$$

Let

$$x_1 = c, \quad x_2 = \dot{c}, \quad x_3 = \ddot{c}$$



$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \dot{x}_3 = -24x_1 - 26x_2 - 9x_3 + 24r, \\ y = c = x_1 \end{array} \right.$$

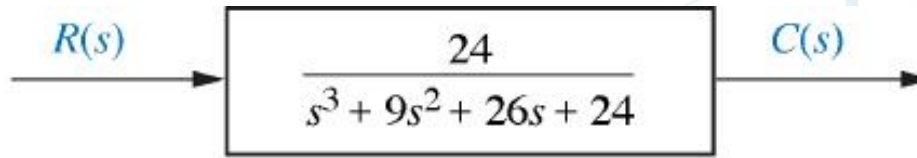
Transfer Function to State Space

Example:

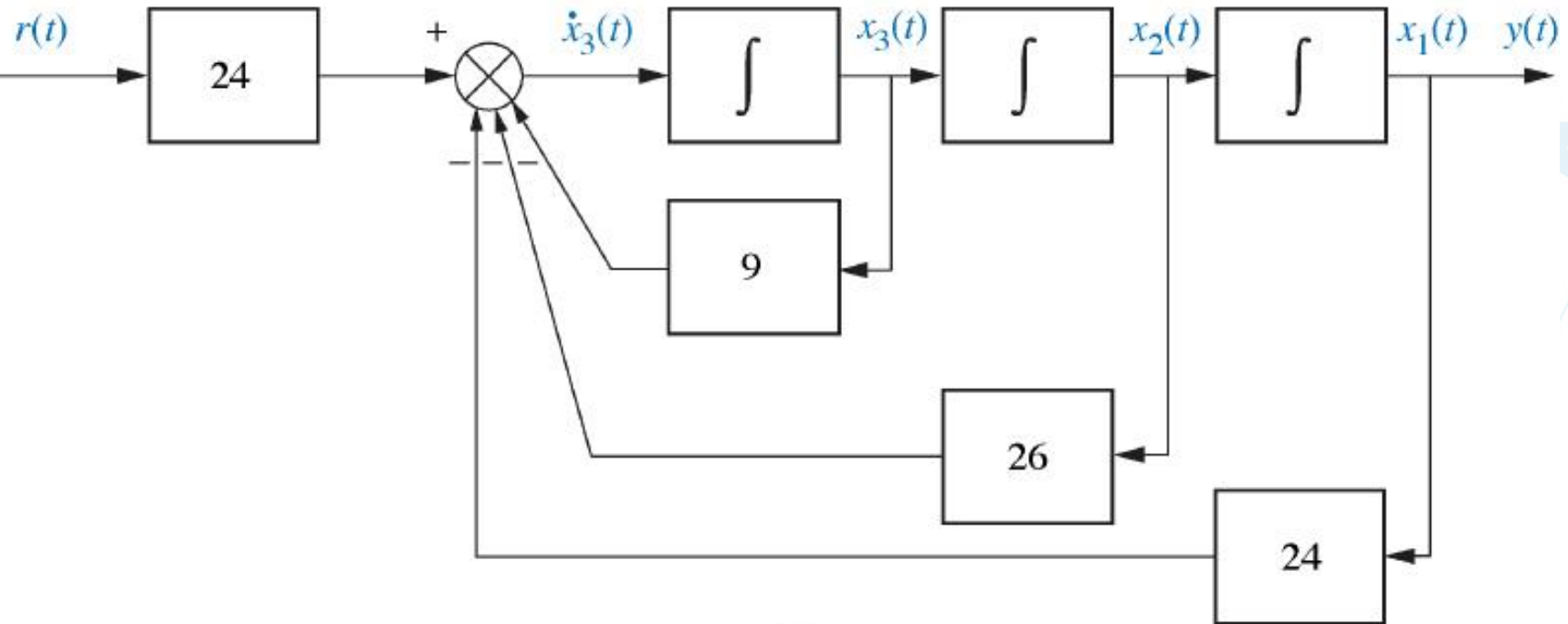
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -24 & -26 & -9 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 24 \end{bmatrix}}_{\mathbf{B}} u$$
$$y = \underbrace{[1 \ 0 \ 0]}_{\mathbf{C}} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Transfer Function to State Space

Example:



(a)



(b)

Transfer Function to State Space

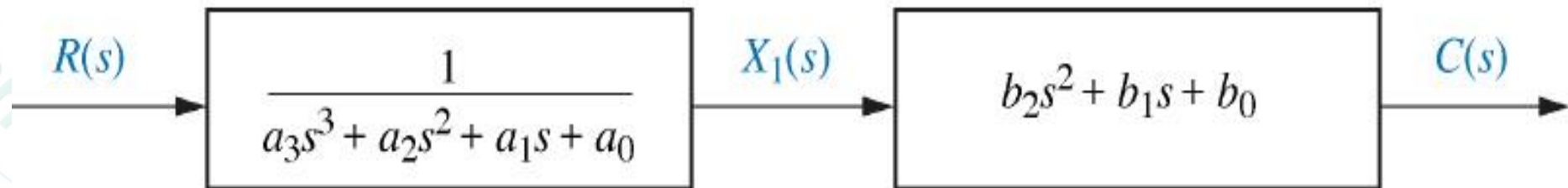
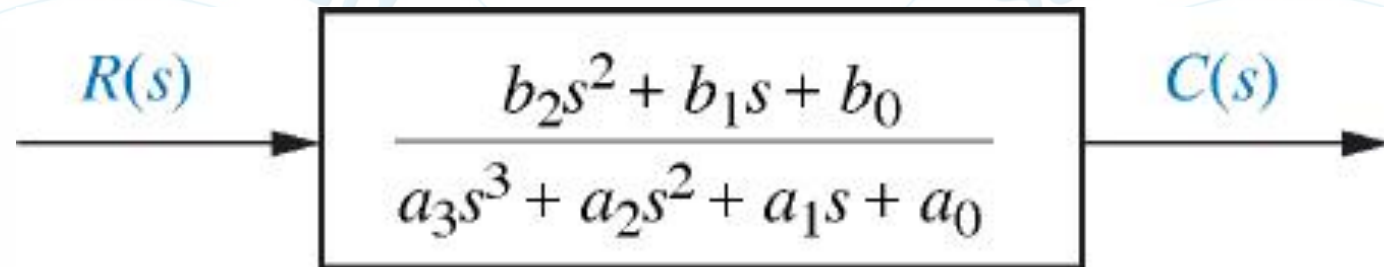
Example:

MATLAB

```
>> n=24;  
>> d=[1 9 26 24];  
>>  
[A,B,C,D]=tf2ss(n,d)
```

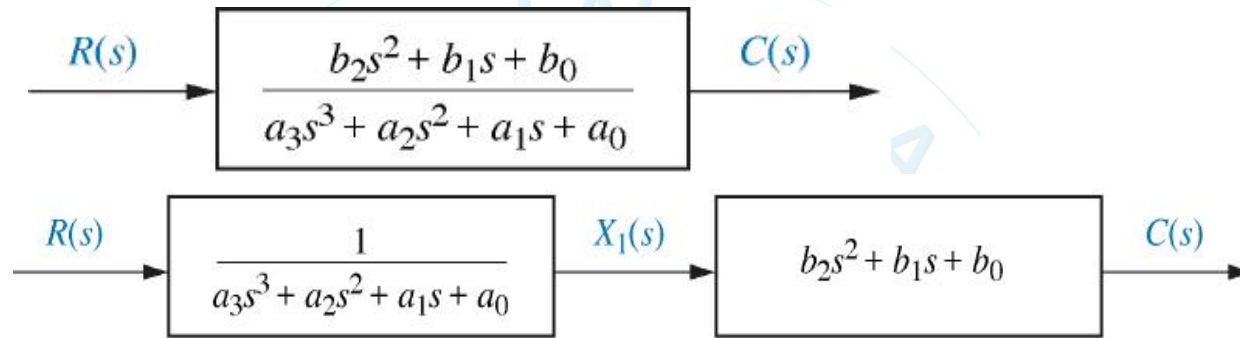
```
A =  
  -9  -26  -24  
   1   0   0  
   0   1   0  
  
B =  
   1  
   0  
   0  
  
C =   0   0  24  
  
D =   0
```


T. Fn. with numerator Poly. to S-S



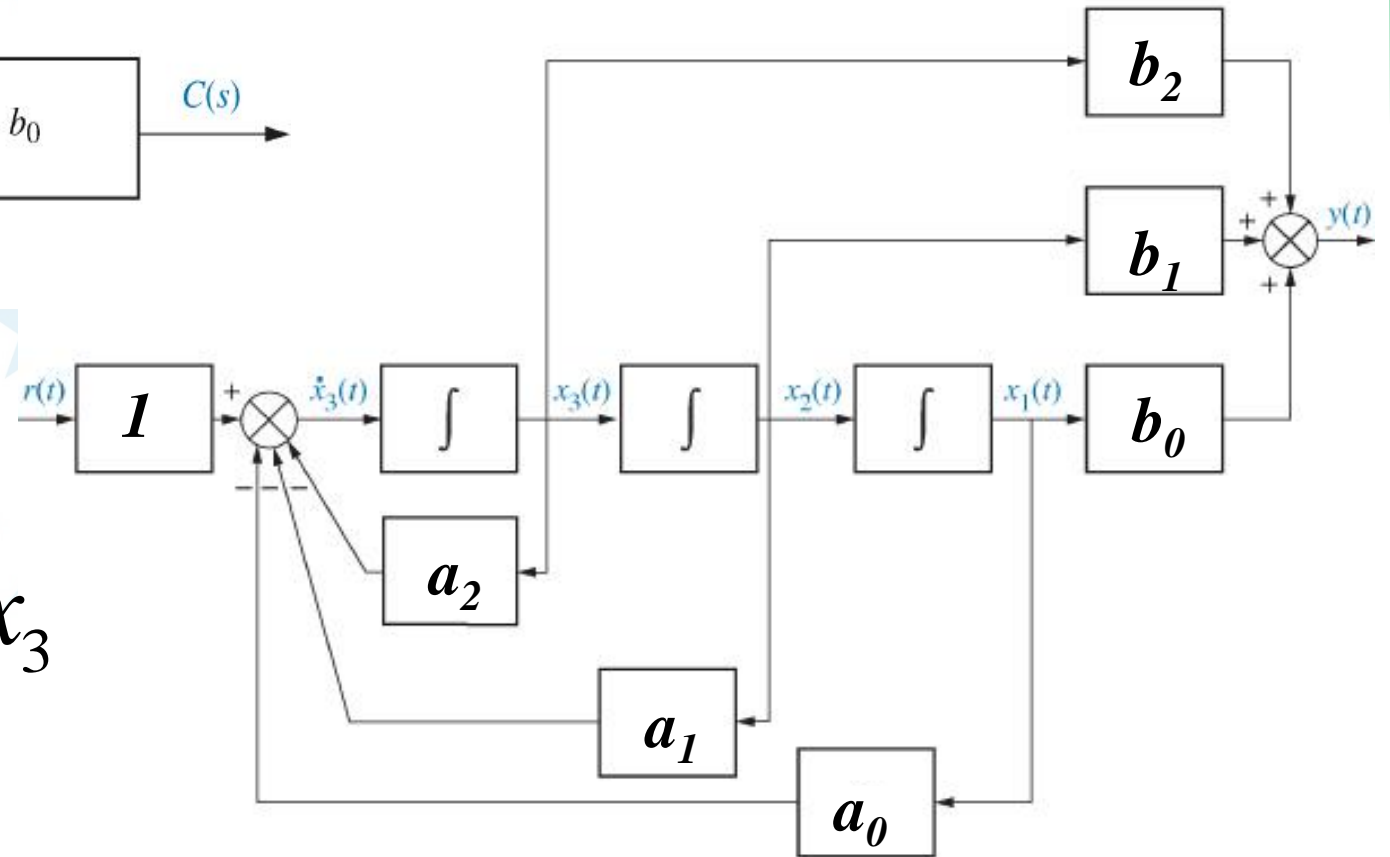
Internal variables:
 $X_2(s), X_3(s)$

T. Fn. with numerator Poly. to S-S

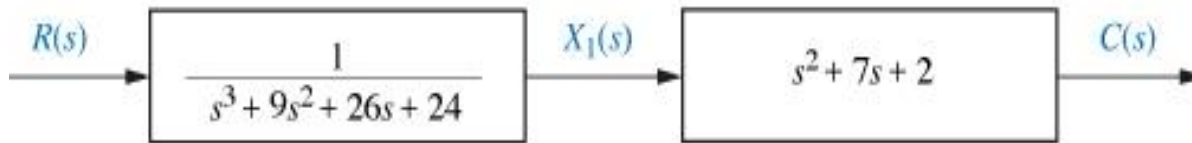
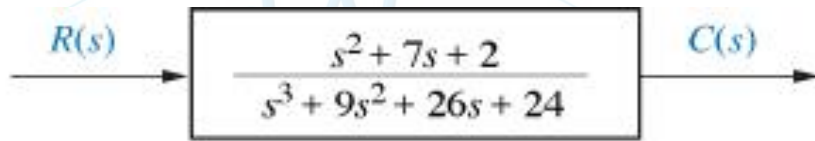


Internal variables:
 $X_2(s), X_3(s)$

$$x_1 = x_1, \dot{x}_1 = x_2, \ddot{x}_1 = x_3$$

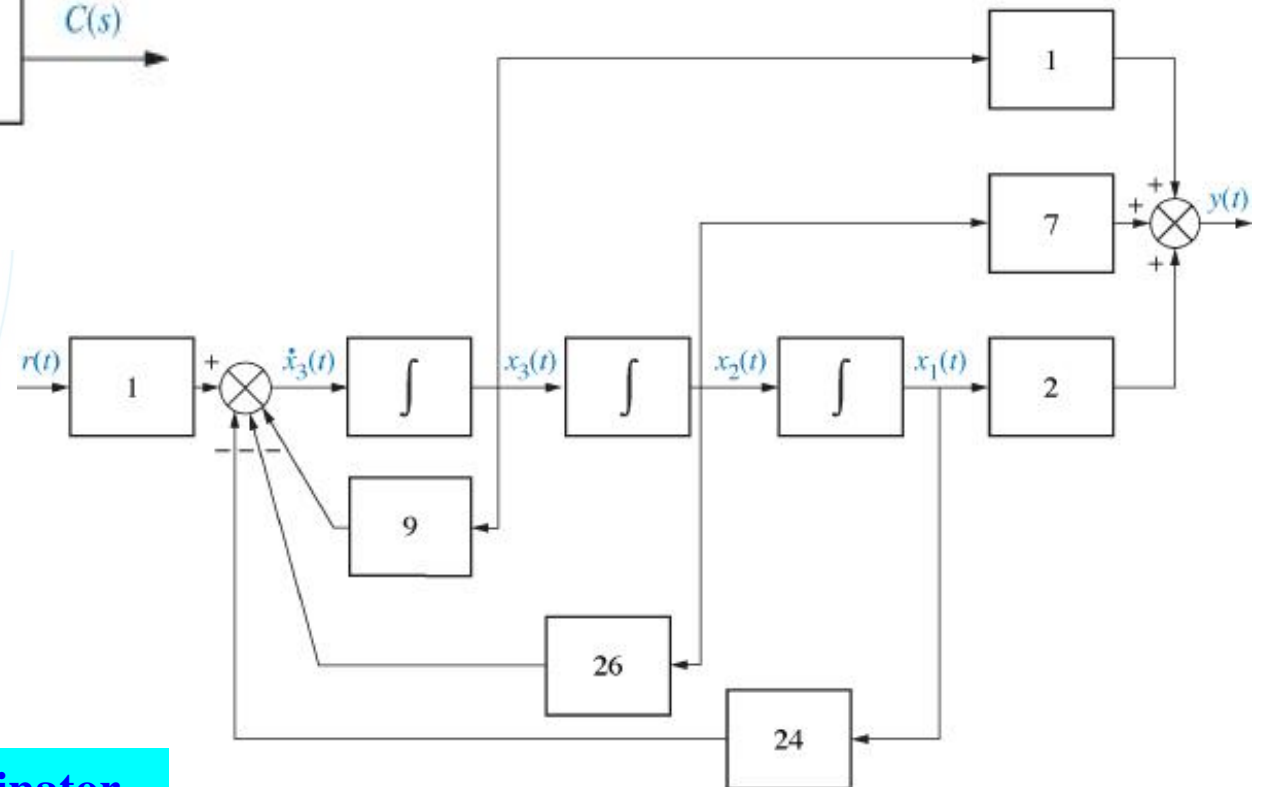


T. Fn. with numerator Poly. to S-S



Internal variables:
 $X_2(s), X_3(s)$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -24 & -26 & -9 \end{bmatrix}}_A \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_B u$$



Same as Transfer function with polynomial denominator

T. Fn. with numerator Poly. to S-S

But $C(s) = (b_2s^2 + b_1s + b_0) X_1(s) = (s^2 + 7s + 2) X_1(s)$

→ $c = \ddot{x}_1 + 7\dot{x}_1 + 2x_1$

Let $x_1 = x_1, \dot{x}_1 = x_2, \ddot{x}_1 = x_3$

→ $y = c = x_3 + 7x_2 + 2x_1 = b_2x_3 + b_1x_2 + b_0x_1$

or $y = [b_0 \ b_1 \ b_2][x_1 \ x_2 \ x_3]^T = \underbrace{[2 \ 7 \ 1]}_C [x_1 \ x_2 \ x_3]^T$

T. Fn. with numerator Poly. to S-S

```
>> n=[1 7 2];  
>> d=[1 9 26 24];  
>> [A,B,C,D]=tf2ss(n,d)
```

A =
-9 -26 -24
1 0 0
0 1 0

B =
1
0
0

C = 1 7 2

D = 0

T. Fn. with numerator Poly. to S-S

SUMMARY

- Denominator of Tr. Fn. (**Poles**) affects the A matrix
- Numerator of Tr. Fn. (**Zeros**) affects the C matrix
- Numerator **gain** affects B matrix

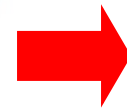
State-Space to Transfer Function

State-Space Model

$$\begin{aligned}\dot{\mathbf{X}} &= \mathbf{A}\mathbf{X} + \mathbf{B}u, \\ \mathbf{y} &= \mathbf{C}\mathbf{X} + \mathbf{D}u\end{aligned}$$

Laplace Transform

$$\begin{aligned}s\mathbf{X} &= \mathbf{A}\mathbf{X} + \mathbf{B}u, \\ \mathbf{y} &= \mathbf{C}\mathbf{X} + \mathbf{D}u\end{aligned}$$



$$\mathbf{X} = (s\mathbf{I} - \mathbf{A})^{-1} \mathbf{B}u$$

or

$$\mathbf{Y}(s) = \left[\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1} \mathbf{B} + \mathbf{D} \right] u$$

State-Space to Transfer Function

Example

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -2 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix} u$$

$$y = [1 \quad 0 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

State-Space to Transfer Function

Example

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

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

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

State-Space to Transfer Function

Example

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

$$\mathbf{B} = [10 \ 0 \ 0]^T, \mathbf{C} = [1 \ 0 \ 0], \mathbf{D} = 0$$

$$\mathbf{Y}(s) = \left[\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1} \mathbf{B} + \mathbf{D} \right] u = \frac{10(s^2 + 3s + 2)}{(s^3 + 3s^2 + 2s + 1)} u$$

Transfer Function

State-Space to Transfer Function

Example

Method 1

```
>> syms s
>> A=[0 1 0;0 0 1;-1 -2, -3];
>> B=[10 0 0]';
>> C=[1 0 0];
>> AS=s*eye(3)-A;
>> ASI=inv(AS);
>> G=C*ASI*B
```

$$G = 10*(s^2+3*s+2)/(s^3+3*s^2+2*s+1)$$

State-Space to Transfer Function

Example

Method 2

```
>> A=[0 1 0;0 0 1;-1 -2, -3];
```

```
>> B=[10 0 0]';
```

```
>> C=[1 0 0];
```

```
>> D=0;
```

```
>> [n,d]=ss2tf(A,B,C,D)
```

```
n =      0  10.0000  30.0000  20.0000
```

```
d =   1.0000  3.0000  2.0000  1.0000
```

```
>> G=tf(n,d)
```

Transfer function:
$$\frac{10 s^2 + 30 s + 20}{s^3 + 3 s^2 + 2 s + 1}$$

State Space Representation

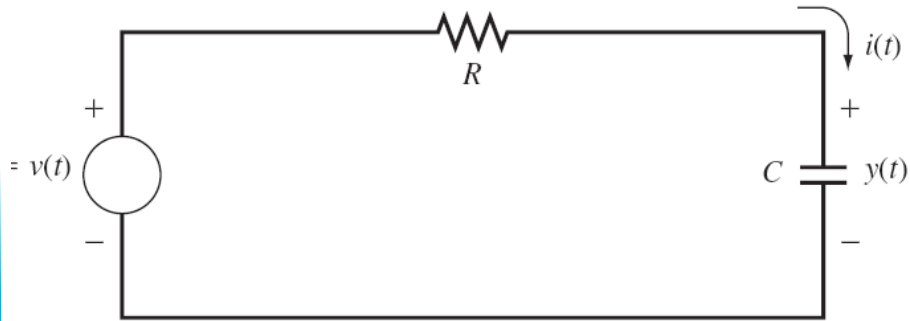
Obtaining State Space Representation

- Identify State Variables
 - Rule of Thumb:
 - Nth order ODE requires N state variables.
 - Position and velocity of inertia elements are natural state variables for translational mechanical systems.
- Eliminate all algebraic equations written in the modeling process.
- Express the resulting differential equations in terms of state variables and inputs in coupled first order ODEs.
- Express the output variables as algebraic functions of the state variables and inputs.
- For linear systems, put the equations in matrix form.

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A} \cdot \underbrace{\mathbf{x}}_{\text{State Vector}} + \mathbf{B} \cdot \underbrace{\mathbf{u}}_{\text{Input Vector}} \\ \mathbf{y} &= \mathbf{C} \cdot \mathbf{x} + \mathbf{D} \cdot \mathbf{u} \\ \text{Output Vector}\end{aligned}$$

State Space Representation

Electrical system Example:



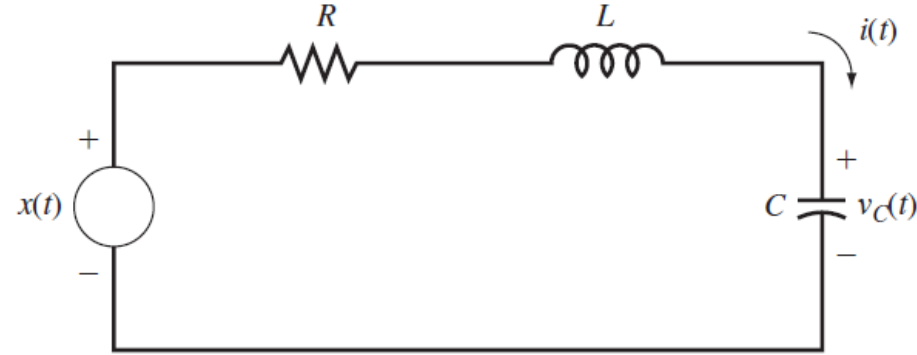
$$\frac{dy(t)}{dt} + \frac{1}{RC} y(t) = \frac{1}{RC} v(t)$$

$$\mathbf{A} = [-a_0] = \left[-\frac{1}{RC} \right]$$

$$\mathbf{B} = [b_0] = \left[\frac{1}{RC} \right] \quad \mathbf{C} = [1] \quad \mathbf{D} = 0$$

$$\begin{bmatrix} \dot{x}_1(t) \end{bmatrix} = \begin{bmatrix} -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} x_1(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{RC} \end{bmatrix} v(t)$$

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



$$\frac{d^2 y(t)}{dt^2} + \left(\frac{R}{L} \right) \frac{dy(t)}{dt} + \frac{1}{LC} y(t) = \frac{1}{LC} v(t)$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1/LC & -R/L \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ b_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1/LC \end{bmatrix} \quad \mathbf{C} = [1 \quad 0] \quad \mathbf{D} = 0$$

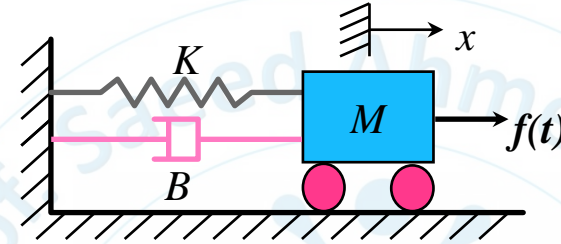
$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1/LC & -R/L \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1/LC \end{bmatrix} v(t)$$

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

State Space Representation

Mechanical system Example:

$$\text{EOM: } M\ddot{x} + B\dot{x} + Kx = f(t)$$



Q: What information about the mass do we need to know to be able to solve for $x(t)$ for $t \geq t_0$?

Input: $f(t)$, $t \geq t_0$

Initial Conditions (ICs): $x(t_0)$ $q_1 = x(t)$
 $\dot{x}(t_0)$ $q_2 = \dot{x}(t)$

Rule of Thumb

Number of state variables = Sum of orders of EOMs

State Space Representation

Mechanical system Example:

EOM $M \ddot{x} + B \dot{x} + K x = f(t)$

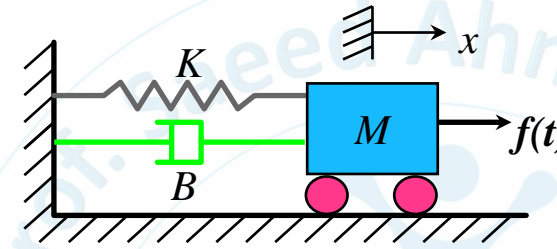
State Variables: $q_1 = x,$
 $q_2 = \dot{x}$

Outputs: $y_1 = x,$ $y_2 = -B \dot{x}$

State Space Representation:

$$\begin{cases} \dot{q}_1 = \dot{x} = q_2 \\ \dot{q}_2 = \ddot{x} = \frac{1}{M} (-B q_2 - K q_1) + \frac{1}{M} f(t) \end{cases} \quad \text{State equation}$$

$$\begin{cases} y_1 = x = q_1 \\ y_2 = -B \dot{x} = -B q_2 \end{cases} \quad \text{Output equation}$$



Matrix Form

$$\underbrace{\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{K}{M} & -\frac{B}{M} \end{bmatrix}}_A \underbrace{\begin{bmatrix} q_1 \\ q_2 \end{bmatrix}}_x + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{M} \end{bmatrix}}_B \underbrace{[f]}_u$$

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}}_y = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & -B \end{bmatrix}}_C \cdot \underbrace{\begin{bmatrix} q_1 \\ q_2 \end{bmatrix}}_x + \underbrace{0}_{D_{2 \times 1}} \cdot u$$

State Space Representation

Exercise

Represent the 2 DOF suspension system in a state space representation.

Let the system output be the relative position of mass M_1 with respect to M_2 .

$$M_1 \ddot{x}_1 + B_1 \dot{x}_1 - B_1 \dot{x}_2 + K_1 x_1 - K_1 x_2 = 0$$

$$M_2 \ddot{x}_2 - B_1 \dot{x}_1 + B_1 \dot{x}_2 - K_1 x_1 + (K_1 + K_2)x_2 = K_2 x_p$$

$$q_1 = x_1, \quad q_2 = \dot{x}_1, \quad q_3 = x_2, \quad q_4 = \dot{x}_2$$

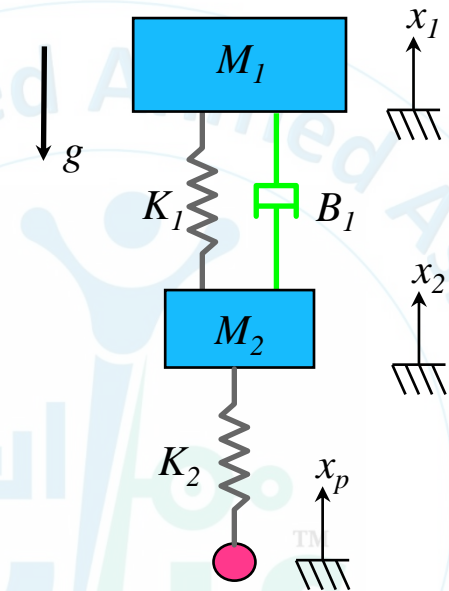
State Variables:

Output: $y = x_1 - x_2$ Input: x_p

State Space Representation:

$$\underbrace{\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{K_1}{M_1} & -\frac{B_1}{M_1} & \frac{K_1}{M_1} & \frac{B_1}{M_1} \\ 0 & 0 & 0 & 1 \\ \frac{K_1}{M_2} & \frac{B_1}{M_2} & -\frac{K_1 + K_2}{M_2} & -\frac{B_1}{M_2} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{K_2}{M_2} \end{bmatrix}}_{\mathbf{B}} \underbrace{x_p}_{\mathbf{u}}$$

$$y = \underbrace{[1 \quad 0 \quad -1 \quad 0]}_{\mathbf{C}} \cdot \mathbf{x} + \underbrace{0}_{\mathbf{D}} \cdot u$$



Input/Output Representation

Input/Output Model

Uses one n th order ODE to represent the relationship between the input variable, $u(t)$, and the output variable, $y(t)$, of a system.

For linear time-invariant (LTI) systems, it can be represented by :

where

$$a_n y^{(n)} + \dots + a_2 \ddot{y} + a_1 \dot{y} + a_0 y = b_m u^{(m)} + \dots + b_2 \ddot{u} + b_1 \dot{u} + b_0 u(t)$$

- To solve an input/output differential equation, we need to know

Input:

$$u(t), \quad t \geq 0$$

Initial Conditions (ICs):

$$y(0), \quad \dot{y}(0), \quad \dots, \quad \underbrace{y^{(n-1)}(0)}_n$$

- To obtain I/O models:
 - Identify input/output variables.
 - Derive equations of motion.
 - Combine equations of motion by eliminating all variables except the input and output variables and their derivatives.

Input/Output Representation

Example

Vibration Absorber

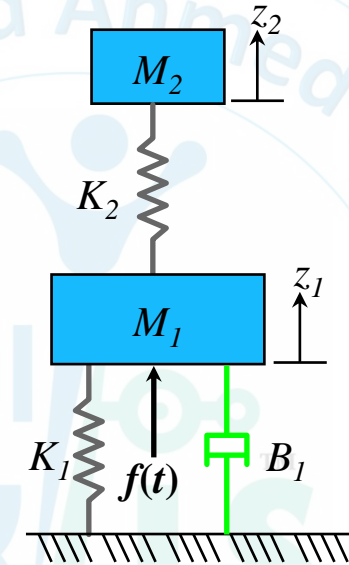
EOM:

$$\begin{aligned} M_1 \ddot{z}_1 + B_1 \dot{z}_1 + (K_1 + K_2)z_1 - K_2 z_2 &= f(t) \\ M_2 \ddot{z}_2 + K_2 z_2 - K_2 z_1 &= 0 \end{aligned}$$

- Find input/output representation between input $f(t)$ and output z_2 .

- Need to eliminate z_1 and its time derivative of all orders

$$\begin{aligned} z_1 &= \frac{1}{K_2} \{M_2 \ddot{z}_2 + K_2 z_2\} \\ \dot{z}_1 &= \frac{1}{K_2} \{M_2 \dot{z}_2^{(3)} + K_2 \dot{z}_2\}, \\ \ddot{z}_1 &= \frac{1}{K_2} \{M_2 \dot{z}_2^{(4)} + K_2 \ddot{z}_2\} \end{aligned}$$



$$z_2^{(4)} + \frac{B_1}{M_1} z_2^{(3)} + \frac{M_1 K_2 + M_2 (K_1 + K_2)}{M_1 M_2} \ddot{z}_2 + \frac{B_1 K_2}{M_1 M_2} \dot{z}_2 + \frac{K_1 K_2}{M_1 M_2} z_2 = \frac{K_2}{M_1 M_2} f$$

Input/Output Models vs State-Space Models

- **State Space Models:**
 - consider the internal behavior of a system
 - can easily incorporate complicated output variables
 - have significant computation advantage for computer simulation
 - can represent multi-input multi-output (MIMO) systems and nonlinear systems
- **Input/Output Models:**
 - are conceptually simple
 - are easily converted to frequency domain transfer functions that are more intuitive to practicing engineers
 - are difficult to solve in the time domain (solution: Laplace transformation)

Conversion between different models

Converting From	Converting to	Matlab function
Transfer Function	Zero-pole-gain	$[z, p, k] = \text{tf2zp}(\text{num}, \text{den})$
Transfer Function	State Space	$[A, B, C, D] = \text{tf2ss}(\text{num}, \text{den})$
Zero-pole-gain	Transfer Function	$[\text{num}, \text{den}] = \text{zp2tf}(z, p, k)$
Zero-pole-gain	State Space	$[A, B, C, D] = \text{zp2ss}(z, p, k)$
State Space	Transfer Function	$[\text{num}, \text{den}] = \text{ss2tf}(A, B, C, D)$
State Space	Zero-pole-gain	$[z, p, k] = \text{ss2zp}(A, B, C, D)$

State Space Equations

Dynamic Equations

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

Observation Equations

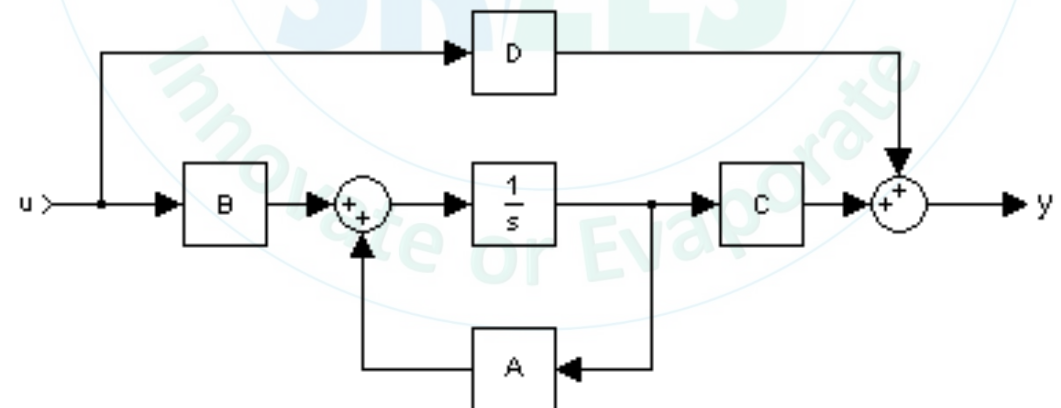
$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$$

\mathbf{x} is the state vector of the system, $n \times 1$;
 n is the order of the system;
 \mathbf{u} is the input vector, $m \times 1$;
 \mathbf{y} is the output vector, $p \times 1$;
 \mathbf{A} is the system (or coefficient) matrix, $n \times n$;
 \mathbf{B} is the input (or driving) matrix, $n \times m$;
 \mathbf{C} is the observation matrix, $p \times n$;
 \mathbf{D} is the feedforward matrix, $p \times m$.

MATLAB function:

```

SYS = SS (A, B, C, D
SYS = SS (A, B, C, D, T)
SYS = SS
SYS = SS (D)
SYS =
SS (A, B, C, D, LTISYS)
SYS = SS (SYS)
  
```



State Space Model to Transfer Function Model

$$s\mathbf{x}(s) - \mathbf{x}(0) = A \mathbf{x}(s) + B \mathbf{u}(s)$$

Solving for $\mathbf{x}(s)$,

$$\mathbf{x}(s) = \text{inv}(sI-A) \mathbf{x}(0) + \text{inv}(sI-A) B \mathbf{u}(s).$$

So,

$$\begin{aligned} \mathbf{y}(s) &= C \mathbf{x}(s) + D \mathbf{u}(s) \\ &= C \text{inv}(sI-A) B \mathbf{u}(s) + D \mathbf{u}(s) \text{ for } \mathbf{x}(0) = \mathbf{0}. \\ &= [C \text{inv}(sI-A) B + D] \mathbf{u}(s) \\ &= \{ [C \text{adj}(sI-A) B + D \det(sI-A)] / \det(sI-A) \} \mathbf{u}(s) \end{aligned}$$

$\det(sI-A) = 0$ is the *characteristic equation* of the system and $[C \text{adj}(sI-A) B + D \det(sI-A)]$ is the *pxm matrix of system zeros*

MATLAB function:
`[NUM,DEN] =`
`SS2TF(A,B,C,D,iu)`

State Space Model to Transfer Function Model

State equations as functions of time

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

$$y = Cx + Du$$

In the s-domain

$$sX - X_0 = AX + BU$$

$$X(sI - A) = BU + X_0$$

$$X = (sI - A)^{-1}BU + (sI - A)^{-1}X_0$$

$$Y = CX + DU$$

$$Y = C((sI - A)^{-1}BU + (sI - A)^{-1}X_0) + DU$$

$$Y = (C(sI - A)^{-1}B + D)U + C(sI - A)^{-1}X_0$$

Assuming the system starts at rest,

$$Y = (C(sI - A)^{-1}B + D)U$$

$$\frac{Y}{U} = (C(sI - A)^{-1}B + D) \quad (\text{the transfer function})$$



State Space Modeling

Example:



State Space Modeling

Example:

State-space representations are, in general, not unique. One system can be represented in several possible ways. For example, consider the following systems:

a. $\dot{x} = -5x + 3u$
 $y = 7x$

b. $\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -5 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 3 \\ 1 \end{bmatrix} u$
 $y = [7 \ 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

c. $\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -5 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 3 \\ 0 \end{bmatrix} u$
 $y = [7 \ 3] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

Show that these systems will result in the same transfer function. We will explore this phenomenon

a.

$$G(s) = C(sI - A)^{-1}B = 7(s + 5)^{-1}3 = \frac{21}{(s + 5)}$$

Pole at s=-5

1 state variable & one pole

b.

$$G(s) = C(sI - A)^{-1}B = [7 \ 0] \begin{bmatrix} s+5 & 0 \\ 0 & s+1 \end{bmatrix}^{-1} \begin{bmatrix} 3 \\ 1 \end{bmatrix} = [7 \ 0] \begin{bmatrix} \frac{1}{s+5} & 0 \\ 0 & \frac{1}{s+1} \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{7}{s+5} & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix} = \frac{21}{s+5}$$

c.

$$G(s) = C(sI - A)^{-1}B = [7 \ 3] \begin{bmatrix} s+5 & 0 \\ 0 & s+1 \end{bmatrix}^{-1} \begin{bmatrix} 3 \\ 0 \end{bmatrix} = [7 \ 3] \begin{bmatrix} \frac{1}{s+5} & 0 \\ 0 & \frac{1}{s+1} \end{bmatrix} \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

$$= [7 \ 0] \begin{bmatrix} \frac{3}{s+5} \\ 0 \end{bmatrix} = \frac{21}{s+5}$$

2 state variables & one pole

but, two eigenvalues



>> A=[-5 0; 0 -1];
 >> eig(A)

ans = -5 -1

Controllability and Observability

$$A = \begin{bmatrix} 0 & 6 & -5 \\ 1 & 0 & 2 \\ 3 & 2 & 4 \end{bmatrix} \text{----- (1)}$$

$$B = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} \text{----- (2)}$$

$$C = [1 \quad 2 \quad 3] \text{----- (3)}$$

Since A is a 3x3 matrix, so n=3.

Controllability: The necessary and sufficient condition for controllability is

$$Q_C = [B \quad AB \quad \text{-----} \quad A^{n-1}B]$$

$$Q_C = [B \quad AB \quad A^2B] \text{----- (4)}$$

From equation (2),

$$B = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} \text{----- (5)}$$

$$A \cdot B = \begin{bmatrix} 0 & 6 & -5 \\ 1 & 0 & 2 \\ 3 & 2 & 4 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$$

$$A \cdot B = \begin{bmatrix} -4 \\ 4 \\ 10 \end{bmatrix} \text{----- (6)}$$

$$A^2B = A \cdot (AB) = \begin{bmatrix} 0 & 6 & -5 \\ 1 & 0 & 2 \\ 3 & 2 & 4 \end{bmatrix} \begin{bmatrix} -4 \\ 4 \\ 10 \end{bmatrix}$$

$$A^2B = \begin{bmatrix} -26 \\ 16 \\ 36 \end{bmatrix} \text{----- (7)}$$

Put equation (5), (6), (7) in equation (4),

$$Q_C = \begin{bmatrix} 0 & -4 & -26 \\ 1 & 4 & 16 \\ 2 & 10 & 36 \end{bmatrix}$$

Now, find the determinant of Q_C

$$|Q_C| = \begin{vmatrix} 0 & -4 & -26 \\ 1 & 4 & 16 \\ 2 & 10 & 36 \end{vmatrix}$$

$$|Q_C| = -36$$

$$\therefore |Q_C| \neq 0$$

Since the determinant of Q_C is non-zero, therefore of $Q_C = n = 3$. Hence, the system is completely controllable.



Controllability and Observability

Observability:

Given,

$$B = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$$

$$B' = [0 \quad 1 \quad 2]$$

$$C = [1 \quad 2 \quad 3]$$

$$C' = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 6 & -5 \\ 1 & 0 & 2 \\ 3 & 2 & 4 \end{bmatrix}$$

$$A' = \begin{bmatrix} 0 & 1 & 3 \\ 6 & 0 & 2 \\ -5 & 2 & 4 \end{bmatrix}$$

The necessary and sufficient condition for observability,

$$Q_C = [C^T \quad A^T C^T \quad \dots \quad (A^T)^{n-1} C^T]$$

For n=3,

$$Q_C = [C^T \quad A^T C^T \quad (A^T)^2 C^T] \quad \text{--- (8)}$$

$$C^T = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \quad \text{--- (9)}$$

$$A^T C^T = \begin{bmatrix} 0 & 1 & 3 \\ 6 & 0 & 2 \\ -5 & 2 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$$A^T C^T = \begin{bmatrix} 11 \\ 12 \\ 11 \end{bmatrix} \quad \text{--- (10)}$$

$$(A^T)^2 C^T = A^T \cdot (A^T C^T) = \begin{bmatrix} 0 & 1 & 3 \\ 6 & 0 & 2 \\ -5 & 2 & 4 \end{bmatrix} \begin{bmatrix} 11 \\ 12 \\ 11 \end{bmatrix}$$

$$(A^T)^2 C^T = \begin{bmatrix} 45 \\ 88 \\ 13 \end{bmatrix} \quad \text{--- (11)}$$

Put equations (9), (10), (11) in equation (8),

$$Q_C = \begin{bmatrix} 1 & 11 & 45 \\ 2 & 12 & 88 \\ 3 & 11 & 13 \end{bmatrix}$$

Now, find the determinant of Q_C

$$|Q_C| = \begin{vmatrix} 1 & 11 & 45 \\ 2 & 12 & 88 \\ 3 & 11 & 13 \end{vmatrix}$$

$$|Q_C| = 1176$$

Since, $|Q_C| \neq 0$, the rank of Q_C is n=3.

∴ The system is completely observable.



END

Controller Canonical Form

Consider the transfer function

$$\frac{y(s)}{u(s)} = \sum_{i=0}^{n-1} b_i s^i / \sum_{j=0}^n a_j s^{n-j}, a_0 = 1.$$

Define

$$y = \sum_i b_i x_{c_i}, \text{ and } \dot{x}_{c_i} = x_{c_{i-1}}, i = 2, \dots, n$$

Structure of the state space model

$$\dot{\mathbf{x}}_c = \mathbf{A}_c \mathbf{x}_c + \mathbf{B}_c \mathbf{u}$$

$$\mathbf{A}_c = \begin{bmatrix} -a_1 & -a_2 & -a_3 & \dots & -a_{n-1} & -a_n \\ \mathbf{I}_{(n-1) \times (n-1)} & \dots & \dots & \dots & \mathbf{0}_{(n-1) \times 1} \end{bmatrix}_{(n \times n)}$$

$$\mathbf{B}_c = \begin{bmatrix} 1 \\ \vdots \\ 0 \end{bmatrix}_{(n \times 1)}, \mathbf{C}_c = [b_1 \quad \dots \quad b_n], \mathbf{D}_c = \mathbf{0}.$$

Controller Canonical Form

Controller canonical form is particularly useful:

- ❑ The coefficients of the numerator and denominator
- ❑ Polynomials of the transfer function appear directly in the state variable model.
- ❑ All other elements are either *0* or *1*.
- ❑ The *state variable model can be written by inspection*, and vice versa. This is used in MATLAB to compute the state space model from the transfer function model with the function **tf2ss**.
- ❑ *It replicates itself if state variable feedback is used.*

Controller Canonical Form

Consider a state variable feedback control law.

$$u(t) = -\mathbf{K}\mathbf{x}_c(t), \mathbf{K} = [k_1 \dots \dots k_n]$$

$$\dot{\mathbf{x}}_c = \mathbf{A}_c \mathbf{x}_c - \mathbf{B}_c \mathbf{K} \mathbf{x}_c = [\mathbf{A}_c - \mathbf{B}_c \mathbf{K}] \mathbf{x}_c$$

$$[\mathbf{A}_c - \mathbf{B}_c \mathbf{K}] = \begin{bmatrix} -(a_1 + k_1) & \dots & \dots & -(a_n + k_n) \\ \mathbf{I}_{(n-1) \times (n-1)} & \dots & \dots & \mathbf{0}_{(n-1) \times 1} \end{bmatrix}$$

Thus, the closed-loop characteristic equation is:

$$s^n + (a_1 + k_1)s^{n-1} + (a_2 + k_2)s^{n-2} + \dots + (a_n + k_n) = 0$$

The feedback coefficients determine the closed-loop poles.

State Space Representations

Example 1.1: Conversion from Transfer-Function Model to State-Space Model

$$\frac{Y(s)}{U(s)} = \frac{s}{s^3 + 14s^2 + 56s + 160}$$

MATLAB code:

```
n=[1 0]; d=[1 14 56 160];
```

```
[Ac,Bc,Cc,Dc]=tf2ss(n,d) %Controller canonical form
```

```
system=tf(n,d)
```

```
[a,b,c,d]=ssdata(system) % Not controller canonical form
```

State Space Representations

Example 1:

$$s^3 + 14s^2 + 56s + 160$$

$$A_c = \begin{bmatrix} -14 & -56 & -160 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$B_c = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$C_c = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

$$D_c = \begin{bmatrix} 0 \end{bmatrix}$$

$$a = \begin{bmatrix} -14 & -7 & -5 \\ 8 & 0 & 0 \\ 0 & 4 & 0 \end{bmatrix}$$

$$b = \begin{bmatrix} 0.2500 \\ 0 \\ 0 \end{bmatrix}$$

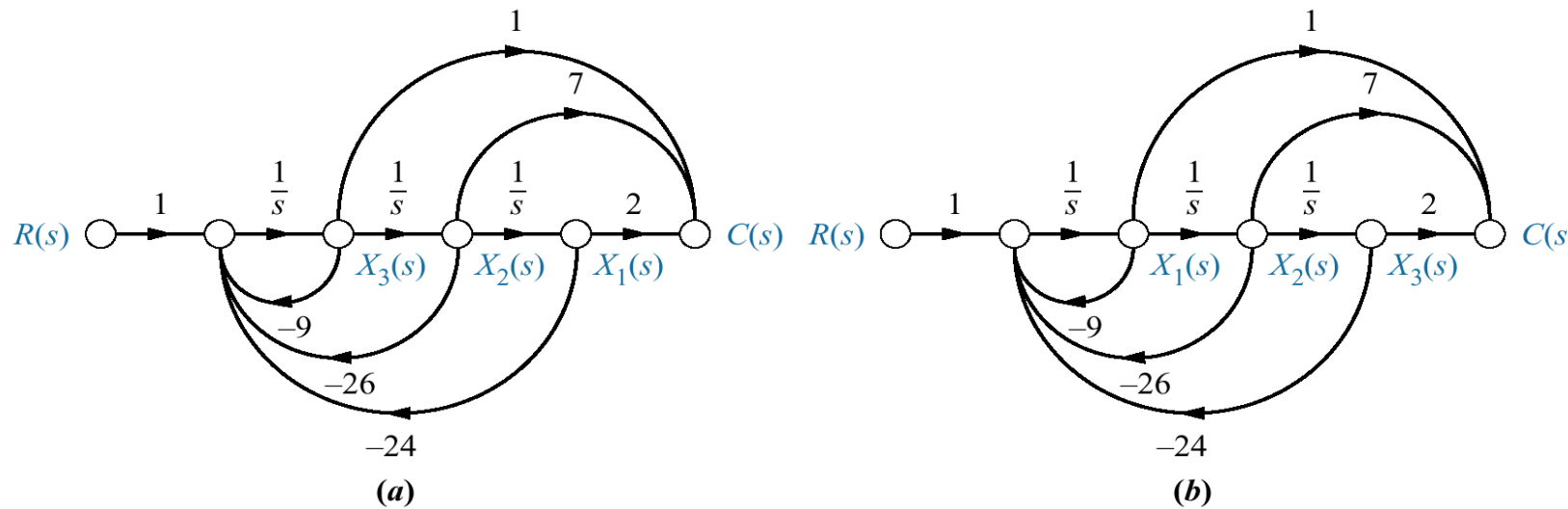
$$c = \begin{bmatrix} 0 & 0.5000 & 0 \end{bmatrix}$$

$$d = \begin{bmatrix} 0 \end{bmatrix}$$

Similarity between phase-variable & controller canonical forms

Signal-flow graphs for obtaining forms for

$$G(s) = C(s)/R(s) = (s^2 + 7s + 2)/(s^3 + 9s^2 + 26s + 24)$$



a. phase-variable form;

b. controller canonical form

Modal Canonical Form

For non-repeated roots

$$y(s) / u(s) = \sum_{i=1}^n r_i / (s - p_i) \quad \text{for } p_i \neq p_j \quad i \neq j$$

Let $x_{m_i}(s) / u(s) = r_i / (s - p_i)$

So,

$$\dot{x}_{m_i}(t) = p_i x_{m_i}(t) + r_i u(t)$$

$$y(t) = \sum_{i=1}^n x_{m_i}(t) = [1 \dots \dots 1] \mathbf{x}$$

MATLAB function:

`[R,P,K] = RESIDUE(B,A)`

`[B,A] = RESIDUE(R,P,K)`

Modal Canonical Form - Vector-Matrix Format

The state variables in this format are *uncoupled*. they are called the *modes*. In vector-matrix notation,

$$\dot{\mathbf{x}}_{\mathbf{m}}(t) = \begin{bmatrix} p_1 & 0 & \cdots & 0 \\ 0 & p_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & p_n \end{bmatrix} \mathbf{x}_{\mathbf{m}}(t) + \begin{bmatrix} r_1 \\ \vdots \\ \vdots \\ r_n \end{bmatrix} u(t)$$

and

$$y = [1 \dots 1] \mathbf{x}_{\mathbf{m}}$$

Definition of Controllability

- ❑ If an input can be found that takes *every* state variable from an *arbitrary initial state* to a *desired final state* in a *finite amount of time*, the system is said to be *controllable*;
- ❑ otherwise, the system is *uncontrollable*.
- ❑ The *controllability* of a system in *modal canonical form* can be determined by *inspection*.
- ❑ A system is *controllable* iff *all of its modes* can be affected by the control (input) *variable(s)*.

Consider a scalar system

$$\dot{x} = px + ru$$

$$x(T) = e^{p(T-t_0)} x(t_0) + \int_{t_0}^T e^{p(T-\tau)} ru(\tau) d\tau$$

For $u = \text{constant} = U$,

$$U = \frac{x(T) - e^{p(T-t_0)} x(t_0)}{r \int_{t_0}^T e^{p(T-\tau)} d\tau}$$

Thus, the scalar system is *controllable* iff r is not zero and the controllability of each mode can be assessed by inspection of the modal canonical form.

Transforming State Equations

$$\mathbf{x} = \mathbf{Tz}$$

$$\text{then } \dot{\mathbf{x}} = \mathbf{T}\dot{\mathbf{z}} = \mathbf{ATz} + \mathbf{Bu}$$

$$\text{or } \dot{\mathbf{z}} = \mathbf{T}^{-1}\mathbf{ATz} + \mathbf{T}^{-1}\mathbf{Bu} = \mathbf{Fz} + \mathbf{Gu}$$

$$\text{and } \mathbf{y} = \mathbf{CTz} + \mathbf{Du} = \mathbf{Hz} + \mathbf{Ju}.$$

So that the state space models are related by the matrix transformations:

$$\mathbf{F} = \mathbf{T}^{-1}\mathbf{AT}$$

$$\mathbf{G} = \mathbf{T}^{-1}\mathbf{B}$$

$$\mathbf{H} = \mathbf{CT}$$

$$\mathbf{J} = \mathbf{D}.$$

Transforming to Control Canonical Form

Suppose we wanted to transform an arbitrary state variable description $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ to the control canonical form. Is it possible?

From the general transformation equations,

$$\mathbf{A}_c \mathbf{T} = \mathbf{T} \mathbf{A}.$$

Considering the three-dimensional case and letting:

$$\mathbf{T} = [\mathbf{t}_1 \quad \mathbf{t}_2 \quad \mathbf{t}_3]^T,$$

Transforming to Control Canonical Form

$$\begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{t}_1^T \\ \mathbf{t}_2^T \\ \mathbf{t}_3^T \end{bmatrix} = \begin{bmatrix} \mathbf{t}_1^T \\ \mathbf{t}_2^T \\ \mathbf{t}_3^T \end{bmatrix} \mathbf{A}$$

From the last row, $\mathbf{t}_2^T = \mathbf{t}_3^T \mathbf{A}$.

From the second row, $\mathbf{t}_1^T = \mathbf{t}_2^T \mathbf{A}$.

Also, since $\mathbf{B}_c = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \mathbf{T} \mathbf{B} = \begin{bmatrix} \mathbf{t}_1^T \\ \mathbf{t}_2^T \\ \mathbf{t}_3^T \end{bmatrix} \mathbf{B}$,

$$1 = \mathbf{t}_1^T \mathbf{B} = \mathbf{t}_2^T \mathbf{A} \mathbf{B} = \mathbf{t}_3^T \mathbf{A}^2 \mathbf{B}$$

$$0 = \mathbf{t}_2^T \mathbf{B} = \mathbf{t}_3^T \mathbf{A} \mathbf{B}$$

$$0 = \mathbf{t}_3^T \mathbf{B}$$

or

$$\mathbf{t}_3^T [\mathbf{B}, \mathbf{A} \mathbf{B}, \mathbf{A}^2 \mathbf{B}] = [0 \quad 0 \quad 1].$$

Transforming to Control Canonical Form

Solving for \mathbf{t}_3^T ,

$$\mathbf{t}_3^T = [0 \quad 0 \quad 1][\mathbf{B} \quad \mathbf{A}\mathbf{B} \quad \mathbf{A}^2\mathbf{B}]^{-1}$$

From which we can solve for \mathbf{t}_2^T and \mathbf{t}_1^T from the equation derived above if, and only if, $[\mathbf{B} \quad \mathbf{A}\mathbf{B} \quad \mathbf{A}^2\mathbf{B}]^{-1}$ exists!

General Case: In general, we define the controllability matrix, \mathbf{C} , as:

$$\mathbf{C} = [\mathbf{B} \quad \mathbf{A}\mathbf{B} \quad \mathbf{A}^2\mathbf{B} \quad \dots \quad \mathbf{A}^{n-1}\mathbf{B}]$$

$$\mathbf{t}_n^T = [0 \quad \dots \quad 0 \quad 1]\mathbf{C}^{-1}$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{t}_n^T \mathbf{A}^{n-1} \\ \mathbf{t}_n^T \mathbf{A}^{n-2} \\ \vdots \\ \mathbf{t}_n^T \end{bmatrix}$$

MATLAB function:

`CO = CTRB (A, B)`

`CO = CTRB (SYS)`

Controllability - an alternate viewpoint

- We have just seen that an *arbitrary* state space representation can be transformed to the *control canonical* form *iff* the *controllability matrix is nonsingular*.
- Once in controller canonical form the closed-loop poles can be arbitrarily placed using state variable feedback.
- Thus, *a system is controllable iff its closed-loop poles can be arbitrarily selected using state variable feedback*.
- *All viewpoints of controllability are compatible*.

Forming the Controllability Matrix

Example

For the canonical state-space model defined in Example 2

$$\mathbf{CO} = \text{ctrb}(\mathbf{Ac}, \mathbf{Bc})$$

$\mathbf{CO} =$

$$\begin{bmatrix} 1 & -14 & 140 \\ 0 & 1 & -14 \\ 0 & 0 & 1 \end{bmatrix}$$

Now we can check its determinant to make sure its inverse exists:

$$\det(\mathbf{CO})$$

$$\text{ans} = 1$$

Therefore, the system is controllable.

An Uncontrollable System

Example 3: Let $\mathbf{a}=[1 \ 1;0 \ 1]$; $\mathbf{b}=[1;0]$; $\mathbf{CO}=\text{ctrb}(\mathbf{a},\mathbf{b})$; $\det(\mathbf{CO})$

ans = 0

Since the controllability matrix is singular the system is uncontrollable. Note the transfer function model has pole-zero cancellation at $s = -1$. This is symptomatic of an uncontrollable mode.

$[\text{zeros},\text{poles}]=\text{ss2zp}(\mathbf{a},\mathbf{b},[1 \ 1],0)$

zeros =

1.0000

poles =

1

1

Transforming to Modal Canonical Form

$$\dot{\mathbf{z}} = \mathbf{V}^{-1} \dot{\mathbf{x}} = \mathbf{V}^{-1} \mathbf{A} \mathbf{x} + \mathbf{V} \mathbf{B} \mathbf{u} = \mathbf{V}^{-1} \mathbf{A} \mathbf{V} \mathbf{z} + \mathbf{V}^{-1} \mathbf{B} \mathbf{u} = \mathbf{A}_m \mathbf{z} + \mathbf{B}_m \mathbf{u}.$$

So, $\mathbf{A}_m = \mathbf{V}^{-1} \mathbf{A} \mathbf{V}$ or $\mathbf{V} \mathbf{A}_m = \mathbf{A} \mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \end{bmatrix} \begin{bmatrix} p_1 & 0 & \cdots & 0 \\ 0 & p_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & p_n \end{bmatrix}.$

Thus, $\mathbf{A} \mathbf{v}_i = p_i \mathbf{v}_i$ for $i = 1 \cdots n$ **Eigenvalue Problem**

where

$\{p_i\}$ are the eigenvalues of \mathbf{A} , i.e., $\det(\mathbf{A} - p_i \mathbf{I}) = 0$; and

$\{\mathbf{v}_i\}$ are the eigenvectors of \mathbf{A} .

MATLAB function:

`E = EIG(X)`

`[V,D] = EIG(X)`