Fall 2021 ME424 Modern Control and Estimation

Lecture Note 2
State Space Models

Prof. Wei Zhang
Department of Mechanical and Energy Engineering
SUSTech Institute of Robotics
Southern University of Science and Technology

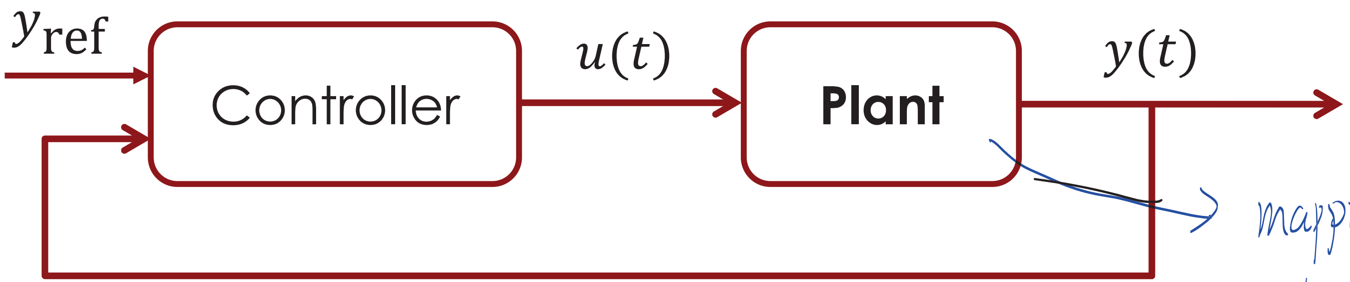
zhangw3@sustech.edu.cn
https://www.wzhanglab.site/

Outline

- **State space model: definition and examples**
- From continuous-time to discrete time model
- From nonlinear to linear model
- State space model $\leftrightarrow$ transfer function

State-space model based feedback control system:

- Goal: determine control input to achieve desired output



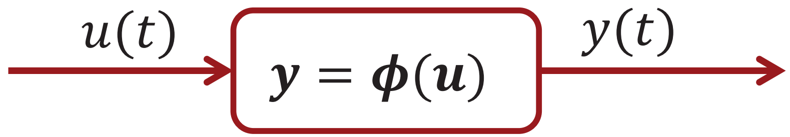
mapping input u to output y

Model ≠ Plant

- Controller design is based on plant model
  - Model is different from the actual plant
  - "all models are wrong, but some are useful"

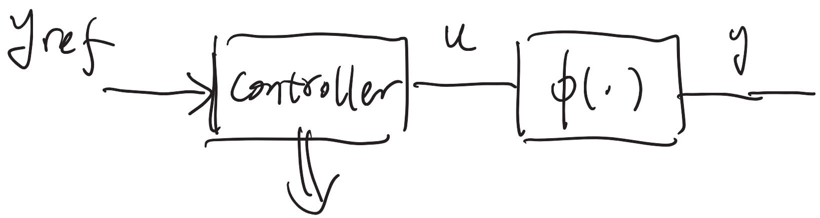
- Modeling approach:
  - First principle : physical "laws" This lecture.
  - Data driven (System ID) : Next lecture (use input/output data pairs to construct a model)

- **Static vs. Dynamic Systems**
  - Static system



$y = 2u$

  - $u(t)$ completely and immediately determines $y(t)$
  - Desired output y_{ref} can be perfectly tracked (in absence of disturbance) by open-loop plant inversion

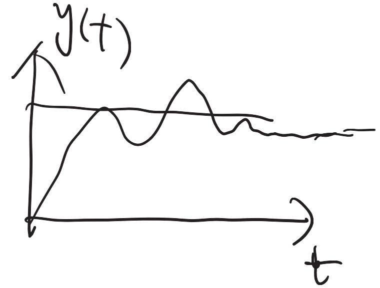


$\phi^{-1}(\cdot)$

$u = \phi^{-1}(y_{ref})$

$y = \phi(u) = \phi(\phi^{-1}(y_{ref})) = y_{ref}$

- **Static vs. Dynamic Systems**
  - **Dynamic system:** differential or difference equation



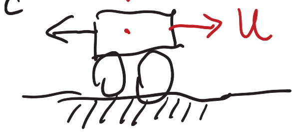
$$u(t) \longrightarrow \boxed{\mathrm{ODE}(y,u)} \longrightarrow y(t)$$

$\ddot{y} + 2\dot{y} = -e^{u}$

- $u(t)$ does not fully determines $y(t)$
- At time t_0, the output $y(t_0)$ does not fully captures the system "behavior"

friction

c p u force



$m\ddot{p} = u - c$

ODE

$\Rightarrow$ $u \longrightarrow \boxed{\text{Cart}} \longrightarrow y = p$

① output $y = p$ is not enough, because the same position, may have different velocity $\Rightarrow$ future behavior is different.

② we need $p(t_0)$ and velocity $\dot{p}(t_0)$

$=$ n^{th}-order ODE needs n-initial conditions

at time t_0: we need $\left.\begin{bmatrix} p(t_0) \\ \dot{p}(t_0) \end{bmatrix}\right\}$ state at time t_0

③ let $x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} p(t) \\ \dot{p}(t) \end{bmatrix}$

$\dot{x} = \begin{bmatrix} x_2(t) \\ u - c \end{bmatrix} = f(x, u)$

- **"State": info needed for future evolution,** it separates future from past
- **State** $x(t_0)$ at time t_0 and **input** $\boldsymbol{u(t)}$ over time $[t_o, t_f]$, **completely determines** the system behaviors

General continuous-time state space model

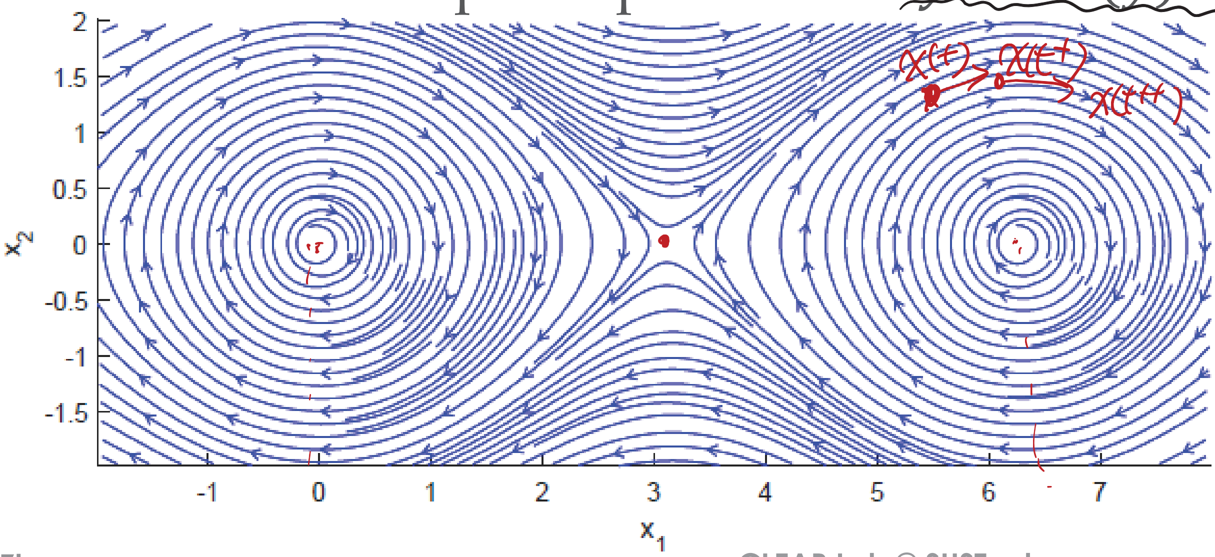
$$\begin{cases} \dot{x} = \underbrace{f(x,u)} \\ y = h(x,u) \end{cases}$$

only involve 1^{st}-order derivative

but in $\mathbb{R}^n$

- $x \in R^n$ state vector, $u \in R^m$ control input, $y \in R^p$ output,
- $f: R^n \times R^m \to R^n$: called **vector field** — specify "velocity" of the state vector
- $h: R^n \times R^m \to R^p$: output function — e.g. $u = 2x \Rightarrow \underbrace{f(x, 2x)}_{f_c(x)}$
- Called autonomous system if there is no control $f(x,u) = f(x)$
- For autonomous sys, $\hat{x} \in \boldsymbol{R^n}$ is called **equilibrium** if $\underbrace{f(\hat{x}) = 0}_{\text{velocity} = 0}$

Vector field example of pendulum: $\ddot{y} + \sin(y) = 0$



$x_1 = y$

$x_2 = \dot{y}$

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} x_2 \\ -\sin x_1 \end{bmatrix}}_{f(x)}$$

at time t, $x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \Rightarrow \underbrace{f(x(t))}_{\text{velocity}}$

- **General discrete-time state space model**

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ y(k) = h(x(k), u(k)) \end{cases}$$

  - $x \in R^n$ state vector, $u \in R^m$ control input, $y \in R^p$ output
  - $f: R^n \times R^m \to R^n$: state update equation
  - $h: R^n \times R^m \to R^p$: output function
  - Called autonomous system if there is no control $f(x,u) = f(x)$
  - For autonomous sys, $\hat{x} \in \boldsymbol{R^n}$ is called **equilibrium** if $\hat{x} = f(\hat{x})$

- Discrete-time system:
  - Some discrete-time system is obtained from continuous time model by sampling
  - Some systems naturally evolve in discrete time.

- **Linear Systems:** system is called linear if:

$x \in \mathbb{R}^n$

Continuous time $\quad \dot{x} = f(x, u) = Ax + Bu,$ $\quad u \in \mathbb{R}^m$

$\quad y = h(x, u) = Cx + Du,$ $\quad y \in \mathbb{R}^p$

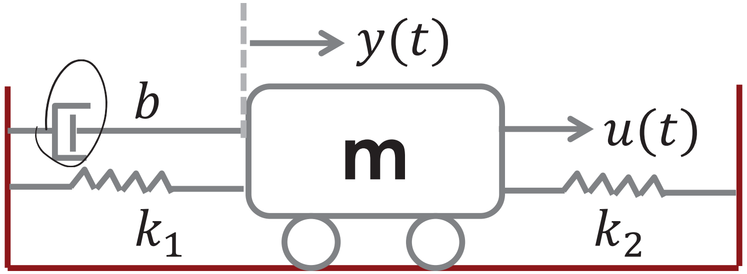
Discrete time $\quad x(k+1) = f(x(k), u(k)) = Ax(k) + Bu(k),$

$\quad y(k) = h(x(k), u(k)) = Cx(k) + Du(k),$

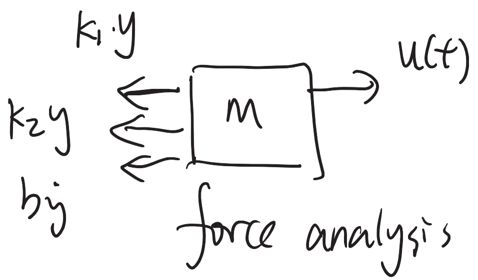
for some matrices A, B, C, D $\quad A \in \mathbb{R}^{n \times n}$ $\quad B \in \mathbb{R}^{n \times m}$ $\quad C \in \mathbb{R}^{p \times n}$, $\quad D \in \mathbb{R}^{p \times m}$

- **State-space modeling:**
  - Find the functions $f(\cdot,\cdot), h(\cdot,\cdot)$
  - Or find A, B, C, D matrices if the system is linear

Example 1: Consider spring-damper cart system with zero initial conditions (initially at y = 0 and not moving). No friction



- Differential equation model



$$\Rightarrow \quad m\ddot{y} = u - (k_1+k_2)\,y - b\dot{y}$$

$$\Rightarrow \quad m\ddot{y} + b\dot{y} + (k_1+k_2)\,y = u$$

- State space model of Example 1 (infinitely many)

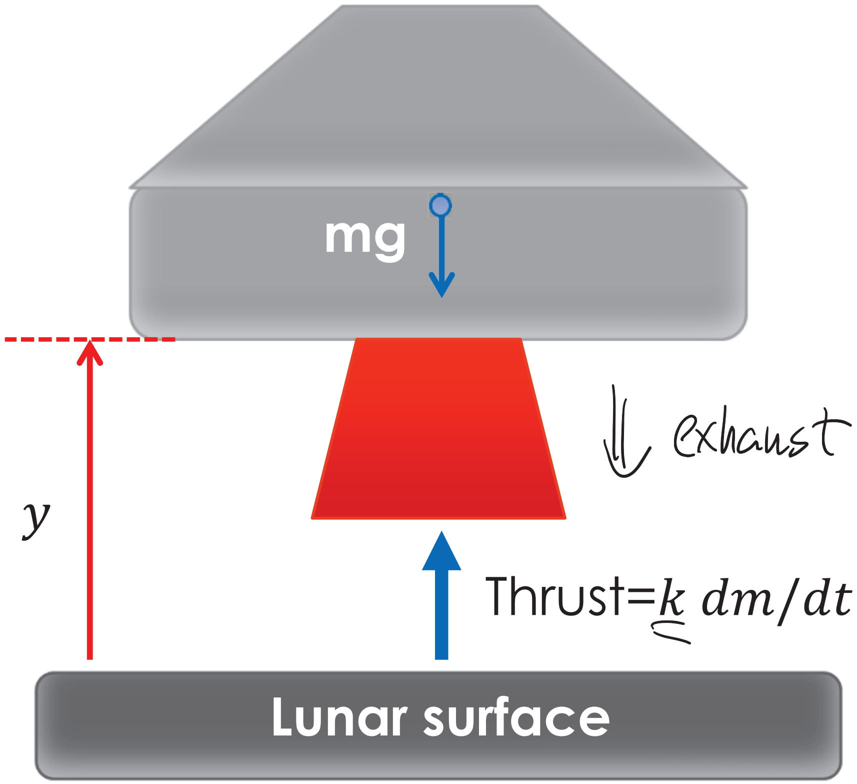
Let's define $\left\{\begin{array}{l} x_1(t) = y(t) \\ x_2(t) = \dot{y}(t) \end{array}\right\} \xRightarrow{x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}$ $\begin{array}{l} \dot{x} = f(x, u) \\ y = h(x, u) \end{array}$ find

$$\dot{x} = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ \frac{1}{m}\left(u - b x_2(t) - (k_1 + k_2) x_1(t)\right) \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} 0 & 1 \\ \frac{-(k_1+k_2)}{m} & \frac{-b}{m} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}}_{B} u$$

$$y = x_1 = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{0}_{D} \cdot u$$

- **Example 2**: soft landing of a lunar module, $u = \dfrac{dm}{dt}$



thrust $= k \cdot \dfrac{dm}{dt}$

m, mg

$$\begin{cases} m\ddot{y} = -mg + k \cdot u \\ \dot{m} = u \end{cases}$$

let $x_1 = y$, $x_2 = \dot{y}$, $x_3 = m$

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} x_2 \\ -g + \frac{k}{x_3} \cdot u \\ u \end{bmatrix} = f(x,u) = \begin{bmatrix} f_1(x,u) \\ f_2(x,u) \\ f_3(x,u) \end{bmatrix}$$

x_2

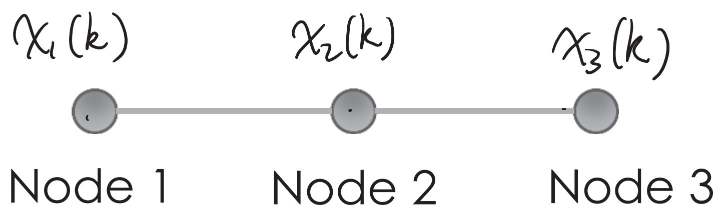
$f_3(x,u) = u$

$\neq Ax + Bu$

No (nonlinear)

$x(k+2) + x(k-10) + \cdots$

- **Example 3**: Sensor Network
  - Each iteration, exchange measurements with neighbors
  - The updated value is the average of its own value with the neighbors

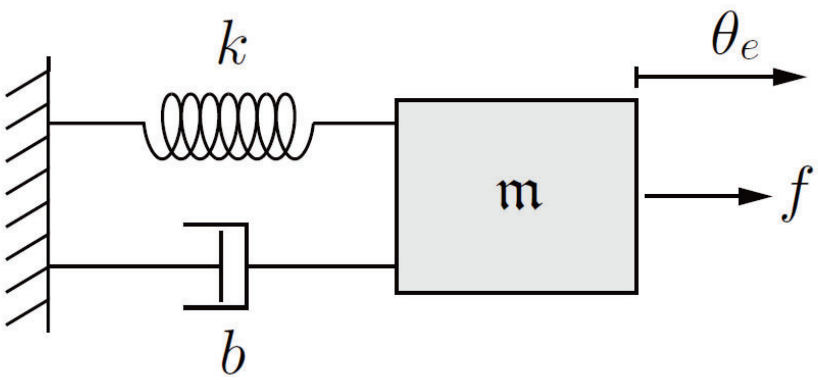


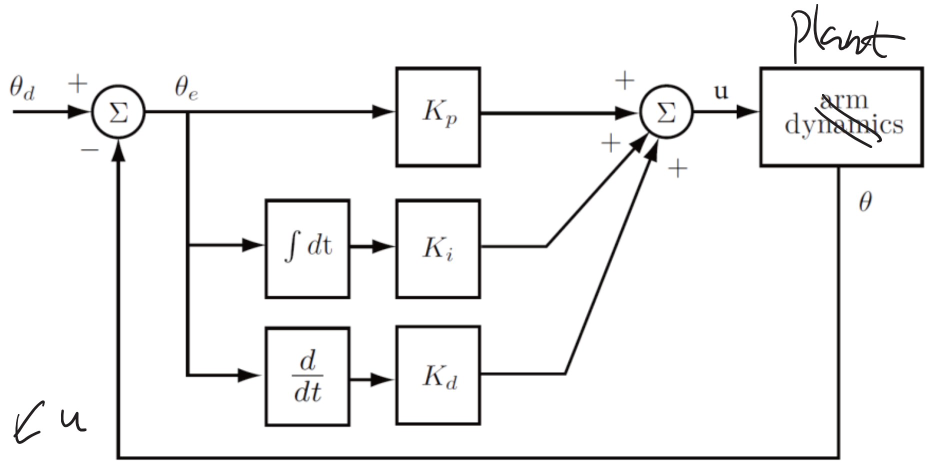
value stored at node i

Q: How does values stored in the nodes evolve over time?

$$x(k+1) = \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{1}{2}\left(x_1(k) + x_2(k)\right) \\ \frac{1}{3}\left(x_2(k) + x_1(k) + x_3(k)\right) \\ \frac{1}{2}\left(x_2(k) + x_3(k)\right) \end{bmatrix}}_{f(x(k),\, u(k))} = \underbrace{\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}}_{A} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix}$$

▪ **Example 4**: PID for spring-damper system





Dynamics: $\underbrace{m\ddot{\theta}_e + b\dot{\theta}_e + k\cdot\theta_e}_{u} = f$ $\Longleftrightarrow$ $\underbrace{\dot{x} = f(x,u), \quad u = g(x)}$

$\Rightarrow \dot{x} = f(x, g(x))$

using PID: $f = \underline{k_p\cdot\theta_e + k_i\cdot\int\theta_e dt + k_d\cdot\dot{\theta}_e}$

$\underbrace{\text{closed-Loop}}$ system: $m\ddot{\theta}_e + (b-k_d)\dot{\theta}_e + (k-k_p)\theta_e - k_i\int\theta_e dt = 0$

plug in control

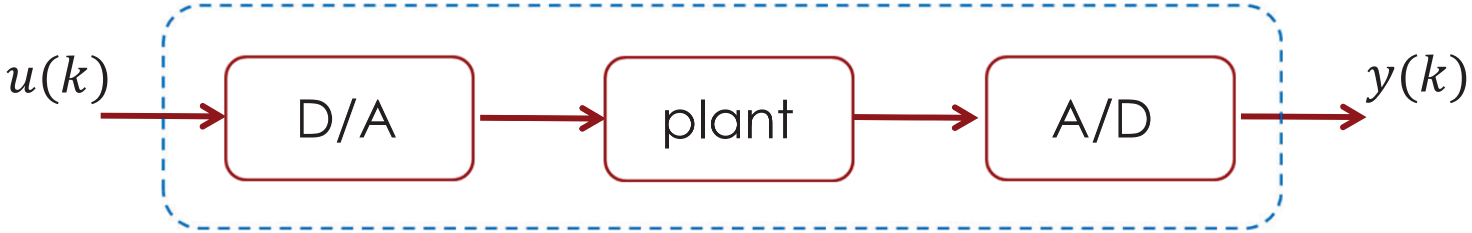
state-space model: $x_1 = \int\theta_e dt \quad x_2 = \theta_e, \quad x_3 = \dot{\theta}_e$

$$\dot{x} = \begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \\ \frac{k_i}{m}x_1 + \frac{k_p-k}{m}x_2 + \frac{k_d-b}{m}x_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{k_i}{m} & \frac{k_p-k}{m} & \frac{k_d-b}{m}\end{bmatrix}}_{A}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}$$

Outline

- State space model: definition and examples
- **From continuous-time to discrete time model**
- From nonlinear to linear model
- State space model ↔ transfer function

From continuous time to discrete time model



- Approximate differential equation with difference equation
  - Euler forward rule:

From calculus: $\dot{g}(t) = \lim_{\Delta t \to 0} \frac{g(t+\Delta t) - g(t)}{\Delta t}$

For small enough Δt: $\dot{g}(t) \approx \frac{g(t+\Delta t) - g(t)}{\Delta t}$

$$g(t+\Delta t) \approx g(t) + \dot{g}(t)\Delta t$$

From continuous-time to discrete-time model

- General nonlinear case:

$f(z, p)$

$f(x, u)$ $\quad$ $f(x(k), u(k))$

Given $\begin{cases} \dot{x} = f(x, u) \\ y = h(x, u) \end{cases}$

$0 \quad \Delta t \quad 2\Delta t \quad 3\Delta t \quad \cdots \quad t$

Define: $x(k) \triangleq x(k \cdot \Delta t)$, $u(k) \triangleq u(k\Delta t)$ $\quad y(k) \triangleq y(k \cdot \Delta t)$ ...

$$x(k+1) = \underbrace{x((k+1)\Delta t)} \approx \underbrace{x(k \Delta t)} + \underbrace{\dot{x}(k \cdot \Delta t)} \cdot \Delta t$$

$$= \boxed{x(k) + f(x(k), \underbrace{u(k)) \cdot \Delta t}}$$

$$f_d(x(k), u(k))$$

$$y(k) = \underbrace{h(x(k), u(k))}$$

$$= h_d(x(k), u(k)).$$

$$\begin{cases} x(k+1) = f_d(x(k), u(k)) \\ y(k) = h_d(x(k), u(k)) \end{cases}$$

From continuous-time to discrete-time model

- Linear case:

CT: $\boxed{\begin{aligned} \dot{x} &= \underbrace{A_c x + B_c u}_{}, \leftarrow f(x,u) \\ y &= C_c x + D_c u, \leftarrow h(x,u) \end{aligned}}$

using the previous nonlinear result

$$x(k+1) = x(k) + \underbrace{\left(A_c \cdot x(k) + B_c \cdot u(k)\right)} \cdot \Delta t$$

$$= \underbrace{(I + A_c \cdot \Delta t)}_{A_d} x(k) + \underbrace{B_c \cdot \Delta t}_{B_d} \cdot u(k)$$

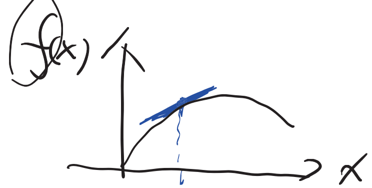
$$y(k) = \underbrace{C_c}_{C_d} \cdot x(k) + \underbrace{D_c}_{D_d} u(k)$$

$\Rightarrow$ DT: $\quad x(k+1) = A_d x(k) + B_d u(k)$

$\Delta t \qquad\quad y(k) = C_d x(k) + D_d u(k)$

Outline

- State space model: definition and examples
- From continuous-time to discrete time model
- **From nonlinear to linear model**
- State space model ↔ transfer function



$f(x) = f(x_0) + f(x_0)'(x - x_0)$

$\Delta f \approx f(x_0)' \Delta x \quad + \text{H.o.T.}$

From nonlinear to linear

- Given model: $x(k+1) = f(x(k), u(k))$, $y(k) = h(x(k), u(k))$ and operating point: $(\hat{x}, \hat{u})$
- Goal: find a linearized model around $(\hat{x}, \hat{u})$

Define: $\Delta x = x - \hat{x}$, $\Delta u = u - \hat{u}$, $\Delta y = y - h(\hat{x}, \hat{u})$

$$\Delta x(k+1) \approx \hat{A} \Delta x(k) + \hat{B} \Delta u(k) + C$$

nonzero if $(\hat{x}, \hat{u})$ is not equilibrium

- Jacobian matrix of multivariable function $f: R^n \to R^m$

$f: \mathbb{R}^3 \to \mathbb{R}^2$, $\begin{bmatrix} f_1(z_1, z_2, z_3) \\ f_2(z_1, z_2, z_3) \end{bmatrix} \in \mathbb{R}^2$, $f(z) \Longrightarrow$ Jacobian

$$\underset{m \times n}{\frac{\partial f}{\partial z}} \triangleq \left[\frac{\partial f_i}{\partial z_j}\right] , \quad \left[\frac{\partial f}{\partial z}\right]_{ij} = \frac{\partial f_i}{\partial z_j}$$

$$\frac{\partial f}{\partial z} = \begin{bmatrix} \frac{\partial f_1}{\partial z_1} & \frac{\partial f_1}{\partial z_2} & \frac{\partial f_1}{\partial z_3} \\ \frac{\partial f_2}{\partial z_1} & \frac{\partial f_2}{\partial z_2} & \frac{\partial f_2}{\partial z_3} \end{bmatrix}$$

$$df = \frac{\partial f}{\partial z} dz = \begin{bmatrix} \frac{\partial f_1}{\partial z_1} & \frac{\partial f_1}{\partial z_2} & \\ \frac{\partial f_2}{\partial z_1} & & \end{bmatrix} \begin{bmatrix} dz_1 \\ dz_2 \\ dz_3 \end{bmatrix}$$

$f: \mathbb{R}^3 \to \mathbb{R}^2$

$f_1(z)$

- Example of Jacobian matrix: $f(z) = \begin{bmatrix} 2z_1 + e^{z_2} \\ \log(z_3) + \frac{1}{z_2} \end{bmatrix}$, $\hat{z} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$

$$\frac{\partial f}{\partial z}(z) = \begin{bmatrix} 2 & e^{z_2} & 0 \\ 0 & \frac{-1}{z_2^2} & \frac{1}{z_3} \end{bmatrix}\Bigg|_{z=\hat{z}} = \begin{bmatrix} 2 & e^2 & 0 \\ 0 & \frac{-1}{4} & \frac{1}{3} \end{bmatrix}$$

- Taylor expansion of multivariate function
  - General expression: $f(z) = f(\hat{z}) + \left(\frac{\partial f}{\partial z}(z)\Big|_{z=\hat{z}}\right)\Delta z$+H.O.T

$f: \mathbb{R}^n \to \mathbb{R}^m$

- Linearization around $(\hat{x}, \hat{u})$ using Taylor expansion:

$x \in \mathbb{R}^n$
$u \in \mathbb{R}^m$
$f: \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$

$$f(x,u) \approx f(\hat{x},\hat{u}) + \overbrace{\left(\frac{\partial f(x,u)}{\partial x}\bigg|_{x=\hat{x},u=\hat{u}}\right)}^{\hat{A}} \cdot \overbrace{(x-\hat{x})}^{\Delta x} + \overbrace{\left(\frac{\partial f(x,u)}{\partial u}\bigg|_{x=\hat{x},u=\hat{u}}\right)}^{\hat{B}} \cdot \overbrace{(u-\hat{u})}^{\Delta u} + HOT$$

$$= \hat{A}\cdot\Delta x + \hat{B}\cdot\Delta u + f(\hat{x},\hat{u})$$

$$z = \begin{bmatrix} x \\ u \end{bmatrix} \in \mathbb{R}^{n+m}, \quad \hat{z} = \begin{bmatrix} \hat{x} \\ \hat{u} \end{bmatrix}, \quad f(x,u) = f(z) = f(\hat{x},\hat{u}) + \underbrace{\left[\frac{\partial f}{\partial x} \;\vdots\; \frac{\partial f}{\partial u}\right]}\begin{bmatrix} \Delta x \\ \Delta u \end{bmatrix}$$

$$\frac{\partial f}{\partial z} = \begin{bmatrix} \frac{\partial f_1}{\partial z_1} & \frac{\partial f_1}{\partial z_2} & \cdots & \frac{\partial f_1}{\partial z_n} & \frac{\partial f_1}{\partial z_{n+1}} & \cdots & \frac{\partial f_1}{\partial z_{n+m}} \\ \frac{\partial f_2}{\partial z_1} & & & & & & \\ \vdots & & & & & & \end{bmatrix}$$

$$\underbrace{\qquad\qquad}_{\frac{\partial f}{\partial x}} \qquad \underbrace{\qquad\qquad}_{\frac{\partial f}{\partial u}}$$

Apply to state space model.

Define: $\Delta x_k = x_k - \hat{x}$ $\quad \Delta u_k = u_k - \hat{u}$

$\Delta x_{k+1} \triangleq x_{k+1} - \hat{x}$

$$\Rightarrow \Delta x_{k+1} \triangleq x_{k+1} - \hat{x} = f(x_k, u_k) - \hat{x} \approx f(\hat{x},\hat{u}) + \underbrace{\left(\frac{\partial f}{\partial x}\bigg|_{x=\hat{x},u=\hat{u}}\right)}_{\hat{A}} \cdot \Delta x_k + \underbrace{\left(\frac{\partial f}{\partial u}\bigg|_{x=\hat{x},u=\hat{u}}\right)}_{\hat{B}} \Delta u_k \; - \hat{x}$$

$$= \hat{A}\Delta x_k + \hat{B}\Delta u_k - \hat{x}$$

$\Rightarrow \Delta x_{k+1} = \hat{A}\,\Delta x_k + \hat{B}\,\Delta u_k + (\underline{f(\hat{x},\hat{u}) - \hat{x}})$ is zero if $\hat{x} = f(\hat{x},\hat{u})$

$$h(x,u) \approx h(\hat{x},\hat{u}) + \underbrace{\left(\left.\frac{\partial h(x,u)}{\partial x}\right|_{x=\hat{x},u=\hat{u}}\right)}_{\hat{C}} \cdot \underbrace{(x-\hat{x})}_{\Delta x} + \underbrace{\left(\left.\frac{\partial h(x,u)}{\partial u}\right|_{x=\hat{x},u=\hat{u}}\right)}_{\widehat{D}} \cdot \underbrace{(u-\hat{u})}_{\Delta u}$$

Define: $\Delta y_k = y_k - \hat{y} = y_k - h(\hat{x},\hat{u}) = \underline{h(x_k,u_k)} - \underline{h(\hat{x},\hat{u})}$

$$\approx \cancel{h(\hat{x},\hat{u})} + \underbrace{\frac{\partial h}{\partial x}}_{\hat{C}}\Delta x_k + \underbrace{\frac{\partial h}{\partial u}}_{\hat{D}}\Delta u_k - \cancel{h(\hat{x},\hat{u})}$$

$$\Delta y := y - h(\hat{x},\hat{u}) \approx \hat{C}\cdot\Delta x + \widehat{D}\cdot\Delta u$$

- **Example:**

$$\underbrace{\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix}}_{x(k+1)} = \underbrace{\begin{bmatrix} \sin(x_2(k)) + \cos(u_2(k)) \\ x_1(k)x_2(k) + u_1u_2(k) \end{bmatrix}}_{f(x_k, u_k)} \begin{matrix} \leftarrow f_1(x,u) \\ \leftarrow f_2 \end{matrix} \quad x \in \mathbb{R}^2, \; u \in \mathbb{R}^2$$

$$f: \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2$$

$$y(k) = \underbrace{\cos(x_2(k)) + 2x_1(k)}_{h(x_k, u_k)} \qquad \hat{x} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \hat{u} = \begin{bmatrix} 0 \\ \frac{\pi}{2} \end{bmatrix}$$

$$\hat{A} = \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}, u=\hat{u}} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}\right|_{\hat{x},\hat{u}} = \left.\begin{bmatrix} 0 & \cos(x_2(k)) \\ x_2(k) & x_1(k) \end{bmatrix}\right|_{\hat{x},\hat{u}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$\hat{B} = \left.\frac{\partial f}{\partial u}\right|_{\hat{x},\hat{u}} = \left.\begin{bmatrix} 0 & -\sin(u_2(k)) \\ u_2(k) & u_1(k) \end{bmatrix}\right|_{\hat{x},\hat{u}} = \begin{bmatrix} 0 & -1 \\ \frac{\pi}{2} & 0 \end{bmatrix}$$

$$\hat{C} = \left.\frac{\partial h}{\partial x}\right|_{\hat{x},\hat{u}} = \left.\begin{bmatrix} \frac{\partial h}{\partial x_1} & \frac{\partial h}{\partial x_2} \end{bmatrix}\right|_{\hat{x},\hat{u}} = \left.\begin{bmatrix} 2 & -\sin x_2(k) \end{bmatrix}\right|_{\hat{x},\hat{u}} = \begin{bmatrix} 2 & 0 \end{bmatrix}$$

$$\hat{D} = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

Outline

- State space model: definition and examples
- From continuous-time to discrete time model
- From nonlinear to linear model
- **State space model ↔ transfer function**

- **From state space to transfer function**

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases}$$

$x \in \mathbb{R}^n$

$u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$

  - Let $X(z), U(Z), Y(z)$ be the z-transforms of $x(k), u(k), y(k)$
  - z-transform: $\boxed{X(z) \triangleq \sum_{k=0}^{\infty} x(k) z^{-k}}$ $\Leftarrow$ $X: \mathbb{C}$

tim: $u(k) \longrightarrow \boxed{\text{sys}} \longrightarrow y(k)$: Time domain sys is modeled as diff equ.

"Frequency" domain: sys is modeled as transfer func.

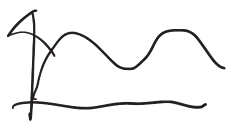
$\Big\downarrow$ z $\Big\downarrow$ z transform

freq

z: $U(z) \longrightarrow \boxed{\text{sys}} \longrightarrow Y(z)$ $H(z) = \dfrac{Y(z)}{U(z)}$ transfer func.

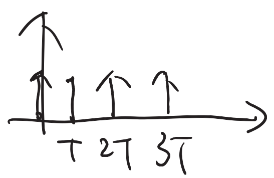
Recall:

C.T signal $x(t)$



$\otimes \Rightarrow$

$x_d(t)$



$$x_d(t) = \sum_{k=0}^{\infty} x(kT) \cdot \delta(t - kT)$$

$$= \sum_{k=0}^{\infty} x[k] \, \delta(t - kT)$$

$$\mathcal{L}[x_d(t)] = \sum_{k=0}^{\infty} x[k] \, e^{-skT}$$

- **From state space to transfer function**

① $x(k+1) = Ax(k) + Bu(k)$

② $y(k) = Cx(k) + Du(k)$

- Recall: if $x(k) \leftrightarrow X(z)$, then $x(k+1) \leftrightarrow zX(z) - zx(0)$

Let's define: $z \triangleq e^{sT}$

$$\Rightarrow \mathcal{L}[x_d(t)] = \sum_{k=0}^{\infty} x[k] z^{-k}$$

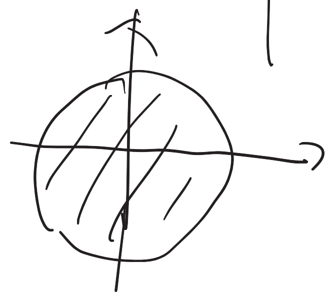
"s"-domain

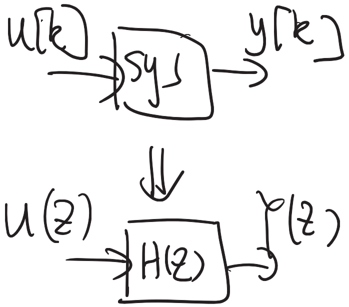
$z = e^{sT}$ $\Downarrow$

$$\boxed{s = \sigma + j\omega, \quad \sigma < 0}$$

$\Downarrow$

$$|z = e^{\sigma + j\omega}| < 1$$





- Scalar case: $H(z) = \dfrac{Y(z)}{U(z)}$ (zero-state respone) r

- MIMO:

Multi-input-multi-output

e.g.: $u \in \mathbb{R}^2$, $y \in \mathbb{R}^3$, $y = \begin{bmatrix} y_1[k] \\ y_2[k] \\ y_3[k] \end{bmatrix} \Rightarrow H(z) = \begin{bmatrix} H_{11}(z) & H_{12}(z) \\ H_{21}(z) & H_{22}(z) \\ H_{31}(z) & H_{32}(z) \end{bmatrix}$

$$Y(z) = \begin{bmatrix} Y_1(z) \\ Y_2(z) \\ Y_3(z) \end{bmatrix}$$

$$Y_2(z) = H_{21}(z) U_1(z) + H_{22}(z) U_2(z)$$

$$x(t) \overset{\mathcal{L}}{\longleftrightarrow} X(s)$$

$$\dot{x}(t) \longleftrightarrow sX(s) - x(0)$$

$$x[k] \overset{z\text{-trf}}{\longleftrightarrow} X(z) = \sum_{k=0}^{\infty} x[k] z^{-k} = x[0] + z^{-1}x[1] + \cdots$$

$$x[k+1] \longleftrightarrow \hat{X}(z) = \sum_{k=0}^{\infty} x[k+1] z^{-k}$$

$$= x[1] + x[2] z^{-1} + \cdots$$

$$= zX(z) - zx[0]$$

Apply z-transform to (1)

$$zX(z) - zx[0] = AX(z) + BU(z)$$

$$(zI - A)X(z) = BU(z) + zx[0]$$

$$X(z) = (zI-A)^{-1}BU(z) + (zI-A)^{-1}zx[0]$$

Apply z-transform to (2)

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

$$= \underbrace{[C(zI-A)^{-1}B + D]U(z)}_{\text{zero-state response}} + \underbrace{C(zI-A)^{-1}zx[0]}_{\text{zero-input respones}}$$

$$\boxed{H(z) = C(zI-A)^{-1}B + D}$$

$$\underset{\text{zero-state}}{Y(z)} = H(z)U(z)$$

- **From transfer function to state space model**
  - **Realization problem**: given transfer function $H(z)$, find (A, B, C, D)

such that $H(z) = C(zI-A)^{-1}B + D$

- Single-input-single-output system:

$$H(z) = \frac{b_n z^n + b_{n-1} z^{n-1} + \cdots + b_1 z + b_0}{z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0}$$

leading coefficient $=1$

- Procedure:
  - First write the transfer function in the above canonical form
  - One possible realization is:

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 \cdots & 0 \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & & \cdots & -a_{n-1} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} b_0 - a_0 b_n & b_1 - a_1 b_n & \cdots & b_{n-1} - a_{n-1} b_n \end{bmatrix}$$

$D = b_n$

V.F.Y. $\quad C(zI-A)^{-1}B + D = H(z)$

- **Example**: $y(k+1) + 3y(k-2) = 2u(k-1)$ $\Leftarrow$
  - First find transfer function: Find state-space model.

Apply z-transform: (Assume zero-I.C.)

$y(k) \Leftrightarrow Y(z)$, $u(k) \Leftrightarrow U(z)$

$zY(z) + 3 \cdot z^{-2} Y(z) = 2z^{-1} U(z)$

$\Rightarrow Y(z) = \dfrac{2z^{-1}}{z + 3 \cdot z^{-2}} U(z)$

$\dfrac{2z^{-1}}{z + 3 \cdot z^{-2}} = \dfrac{2z}{z^3+3} = \dfrac{2z}{z^3 + 0 \cdot z^2 + 0 \cdot z + 3}$

$\Rightarrow n=3$, $a_0 = 3$, $a_1 = a_2 = 0$, $b_0 = 0$, $b_1 = 2$, $b_2 = b_3 = 0$

$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -3 & 0 & 0 \end{bmatrix}$ $B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

$C = \begin{bmatrix} 0 & 2 & 0 \end{bmatrix}$