期末复习

1. Basic Linear Algebra.
   - 1.1 Angles between vectors. (projection)
   - 1.2 Linearly independence
   - 1.3 Range. column space
   - 1.4 Rank
   - 1.5 Conservation of dim
   - 1.6 Null space.
   - 1.7 *diagonalize, Jodan form.
   - 1.8 * inverse

2. State Space models.
   - 2.1 Data Based modeling (LSE)
     - 2.1.1 estimate parameters of ARX model
   - 2.2 Transformation among models
     - 2.2.1 Transfer function
     - 2.2.2 State space model
     - 2.2.3 ARX model
   - 2.3 Least square problems.

3. State space System properties
   - 3.1 state space solution
   - 3.2 Internal stability & BIBO (External)
   - 3.3 K-step reachability
   - 3.4 controllability (def and test)
   - 3.5 Observability (def and test)
   - 3.6 similarity transform

4. Eigenvalue Assignment controller design.
   - 4.1 closed-loop dynamics under state feedback
   - 4.2 controllable canonical form
   - 4.3 eigenvalue assignment in controllable canonical form
   - 4.4 eigenvalue assignment for system not in canonical, how to assign eig

5. Observer design
   - 5.1 Observer structure

5.2 Error dynamics
5.3 Design observer gain matrix
5.4 Output feedback design.

6. Probability
6.1 probability and conditional probability
6.2 random Variables and random vectors
6.3 jointly distributed random vectors and conditional expectation
6.4 covariance matrix

7. Kalman filter
7.1 Minimum Mean squared estimation
7.2 Gaussian Random Vectors (condition)
7.3 Kalman filter Derivations
7.4 EKF
7.5 Kalman filter problems.
7.5.1 tracking
7.5.2 joint state and parameter estimation

8. Dynamic Programming and LQR.
8.1 general discrete-time optimal control problem
8.2 Dynamical programming
8.3 Linear Quadratic Regulator

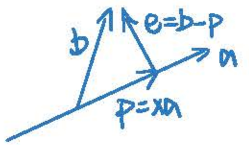
Reference:
HW2-7 SLN
Quiz 1, 2, 4 SLN
Project 1-3 SLN

1. Basic Linear Algebra.

1.1 Angles between vectors. (projection)

① Projection (MIT)

vector project to vector $P = \frac{aa^T}{a^Ta}$



$a^T(b-ax)=0$ (求 x)

$a^Tb - a^Tax = 0$

$x = \frac{a^Tb}{a^Ta}$

$\Rightarrow p = \underline{\frac{aa^T}{a^Ta}} b$ projection matrix

vector project to plane $P = A(A^TA)^{-1}A^T$

② Find the matrix that projects every point in the plane onto the line $x-2y=0$

direction vector of $x-2y=0$ is $[2,1]^T$

b is a point or vector $\vec{OB}$, $|\vec{OB}||v|\cos\theta = \vec{OB}\cdot\vec{v}$

$|\vec{OB}|\cos\theta \cdot \frac{v}{\|v\|} = \vec{v}^T\cdot\vec{OB}\cdot\frac{\vec{v}}{\|v\|^2} = \frac{\vec{v}\vec{v}^T}{\|v\|^2}\vec{OB}$

③ Find a set of vectors of the plane that perpendicular to the plane $x+y-z=3$ in $\mathbb{R}^3$.

intersect points, $(3,0,0)$, $(0,3,0)$, $(0,0,-3)$

$\Rightarrow l_1 = [-1,1,0]^T$ $l_2 = [1,0,1]^T$ $v = [1,1,-1]^T$

v is one of the set, the other could be arbitrary as long as it perpendicular to v

1.2 Linearly independence ⇐ base

A set of vectors (Vector space) $\triangleq V$, let it be the column space of A. $Ax=0$'s null space only have 0. s.t. $c_1v_1 + c_2v_2 + \cdots c_nv_n$ is 0 only if $c_1, c_2 \cdots c_n$ all equal 0.

1.3 Range. column space ...

1.4 Rank ...

1.5 Conservation of dim ...

1.6 Null space.

① find the null space

$$A = \begin{bmatrix} 1 & 2 & 2 & 2 \\ 2 & 4 & 6 & 8 \\ 3 & 6 & 8 & 10 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 & 2 & 2 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 2 & 4 \end{bmatrix} \rightarrow \begin{array}{c} \begin{bmatrix} 1 & \boxed{2} & 0 & \boxed{-2} \\ 0 & \boxed{0} & 1 & \boxed{2} \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ \begin{matrix} \uparrow & & \uparrow & \\ p_1 & & p_2 & \end{matrix} \end{array}$$

$$\text{Null space} = \begin{bmatrix} -2 & 2 \\ 0 & -2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

1.9* eigenvalues and eigenvetor ↓

1.7* diagonalize, Jodan form.

① diagonalize $A = \begin{bmatrix} 6 & 2 & 4 \\ 2 & 3 & 2 \\ 4 & 2 & 6 \end{bmatrix}$

$$\lambda I - A = \begin{bmatrix} \lambda-6 & -2 & -4 \\ -2 & \lambda-3 & -2 \\ -4 & -2 & \lambda-6 \end{bmatrix} \rightarrow \begin{bmatrix} \boxed{\begin{matrix} \lambda-6 & -10 \\ -2 & \lambda-7 \end{matrix}} & \begin{matrix} -4 \\ -2 \end{matrix} \\ \begin{matrix} 0 & 0 \end{matrix} & \boxed{\lambda-2} \end{bmatrix}$$

$|\lambda I - A| = 0 \Rightarrow \lambda_1 = \lambda_2 = 2 \quad \lambda_3 = 11$

substitute to $\lambda I - A$

(1) $\lambda = 2$ $\begin{bmatrix} -4 & -2 & -4 \\ -2 & -1 & -2 \\ -4 & -2 & -4 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1/2 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

$$\text{Null space} = \begin{bmatrix} -1/2 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \rightarrow v_1 = \begin{bmatrix} -1 \\ 2 \\ 0 \end{bmatrix} \quad v_2 = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

(2) $\lambda = 11 \Rightarrow v_3 = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}$

$T = [v_1 \; v_2 \; v_3] \quad \Lambda = T^{-1} A T = P A P^{-1}$

$$P = T^{-1} = \frac{1}{9} \begin{bmatrix} -1 & -4 & 2 \\ 4 & -2 & 1 \\ -1 & 5 & 2 \end{bmatrix}$$

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

$|\lambda I - A| \rightarrow \quad \lambda_1 = \lambda_2 = 1 \quad \lambda_3 = \lambda_4 = -1$

Find Jordan Block:

(1) $\lambda = 1 \quad B_1 = I - A$

$\text{rank}(B_1) = 3 \quad \text{rank}(B_1^2) = 2 \quad \text{rank}(B_1^n) = 2 \; (n > 2)$

num of block (rk=1) $[n - rk(B_1)] - [rk(B_1) - rk(B_1^2)] = 0$

num of block (rk=2) $[rk(B_1) - rk(B_1^2)] - [rk(B_1^2) - rk(B_1^3)] = 1$

(2) $\lambda = -1 \Rightarrow$ num of block (rk=2) = 1

$$\Rightarrow \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

1.8* inverse matrix

① dim = 2 $= \begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$

② Adjugate matrix

$$\underbrace{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}}_{A} \xrightarrow[(-1)^{i+j}]{det} \begin{bmatrix} -3 & 6 & -3 \\ 6 & -12 & 6 \\ -3 & 6 & -3 \end{bmatrix} \xrightarrow{C^T} \underbrace{\begin{bmatrix} -3 & 6 & -3 \\ 6 & -12 & 6 \\ -3 & 3 & -3 \end{bmatrix}}_{adj(A)}$$

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

2. State space models.

2.1 Data Based modeling (LSE)

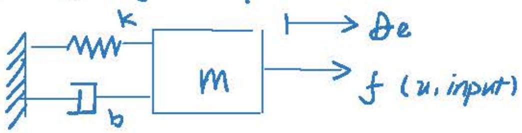
① continuous-time state model

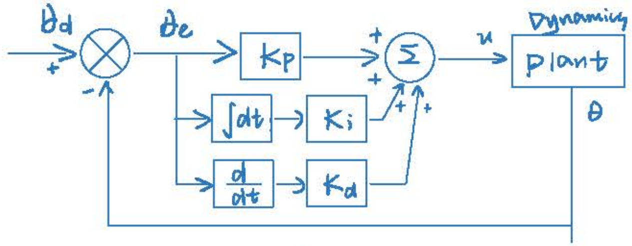
$$\begin{cases} \dot{x} = f(x,u) & \text{vector field} \\ y = h(x,u) \end{cases}$$

discrete-time state space model

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

③ PID spring damper.





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

(f is what to control)

PID: $u = K_p \theta_e + K_d \dot{\theta}_e + K_i \int \theta_e dt$

$0 = m\ddot{\theta}_e + (b - K_d)\dot{\theta}_e + (k - K_p)\theta_e - K_i \int \theta_e dt$

$$x = \begin{bmatrix} \int \theta_e dt \\ \theta_e \\ \dot{\theta}_e \end{bmatrix} \Rightarrow \dot{x} = f(x,u) = f(x, g(x))$$

⑬ Continuous to Discrete.

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

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

If Δt is too large $\Rightarrow$ unstable

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

$$x(k+1) = (I + A\Delta t)\, x(k) + u(k)$$

(HW 3, 4) (Proj 2 4)

⑭ Nonlinear to Linear.

linearize around $(\hat{x}, \hat{u})$

$$\begin{cases} \Delta x = x - \hat{x} \\ \Delta u = u - \hat{u} \\ \Delta y = y - h(\hat{x}, \hat{u}) \end{cases} \qquad f(\hat{x}, \hat{u}) - \hat{x}$$

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

Jacobian:

$$\frac{\partial f}{\partial z} = \begin{bmatrix} \frac{\partial f_1}{\partial z_1} & \frac{\partial f_1}{\partial z_2} & \frac{\partial f_1}{\partial z_3} \\ \frac{\partial f_2}{\partial z_1} & \frac{\partial f_2}{\partial z_2} & \frac{\partial f_2}{\partial z_3} \end{bmatrix}$$ (care about the order)

$$df = \frac{\partial f}{\partial z} dz = \begin{bmatrix} \frac{\partial f_1}{\partial z_1} & \frac{\partial f_1}{\partial z_2} & \frac{\partial f_1}{\partial z_3} \\ \frac{\partial f_2}{\partial z_1} & \frac{\partial f_2}{\partial z_2} & \frac{\partial f_2}{\partial z_3} \end{bmatrix} \begin{bmatrix} dz_1 \\ dz_2 \\ dz_3 \end{bmatrix}$$

e.g. $f(z) = \begin{bmatrix} 2z_1 + e^{z_2} \\ \log(z_3) + \frac{1}{z_2} \end{bmatrix} \quad \hat{z} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$

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

$$f(z) = f(\hat{z}) + \left(\frac{\partial f}{\partial z}(z)\Big|_{z=\hat{z}}\right)\Delta z + \text{H.O.T}$$

(given Δz, find $f(z)$)

$$f(x,u) \approx f(\hat{x}, \hat{u}) + \underbrace{\left(\frac{\partial f(x,u)}{\partial x}\Big|_{x=\hat{x}, u=\hat{u}}\right)}_{\hat{A}}(x - \hat{x}) + \underbrace{\left(\frac{\partial f(x,u)}{\partial u}\Big|_{x=\hat{x}, u=\hat{u}}\right)}_{\hat{B}}(u - \hat{u})$$

Could also be written as:

$$x = f(x, u) = f(\hat{x}, \hat{u}) + \left[\frac{\partial f}{\partial x}\ \frac{\partial f}{\partial u}\right]\begin{bmatrix}\Delta x \\ \Delta u\end{bmatrix}$$

$$\Delta x_{k+1} \triangleq x_{k+1} - \hat{x}$$
$$= f(x_k, u_k) - \hat{x}$$
$$= \hat{A}\Delta x_k + \hat{B}\Delta u_k + \cancel{f(\hat{x},\hat{u}) - \hat{x}}$$

is zero at equilibrium

$$\Delta y \triangleq \underbrace{\frac{\partial h}{\partial x}}_{\hat{C}}\Delta x_k + \underbrace{\frac{\partial h}{\partial u}}_{\hat{D}}\Delta u_k$$

(Proj 2 3)

(HW3 4)

(Lecture Note 2 23 example)

2.4 Least square problems.

① Linear LSE Problem

Solution: $\theta = (X^T X)^{-1} X^T b$

② Curve fitting (HW3, 5)

$y = be^{ax}$ (LN3 15) $y = \alpha_0 + \alpha_1 x + \alpha_2 x^2 \ldots$ (LN3 16)

2.1.1 estimate parameters of ARX model

2.2.3 ARX model

Auto Regression:

$$y(k) + \alpha_1 y(k-1) + \cdots + \alpha_p y(k-p)$$
$$= \beta_0 u(k) + \beta_1 u(k-1) + \cdots + \beta_q u(k-q) + \underbrace{v(k)}_{\text{noise}}$$

Model parameter: $\theta = [\alpha_1, \ldots, \alpha_p, \beta_0, \beta_1, \ldots, \beta_q]^T$

given θ

One-step predictor:

$$\hat{y}(k|\theta) = -\alpha_1 y(k-1) - \cdots - \alpha_p y(k-p)$$
$$+ \beta_0 u(k) + \beta_1 u(k-1) + \cdots + \beta_q u(k-q)$$

$$y\left\{\begin{bmatrix} y(k_0) \\ \vdots \\ y(m)\end{bmatrix}\right. = \underbrace{\begin{bmatrix}\phi^T(k_0) \\ \vdots \\ \phi^T(m)\end{bmatrix}}_{H}\theta + \begin{bmatrix} v(k_0) \\ \vdots \\ v(m)\end{bmatrix}$$

$\boxed{k_0 \text{ start from } \max(p, q) + 1}$

① find best estimate a, b. $G(z) = \frac{z^2 + b}{z^3 + az}$

Given dataset $\{u_i, y_i\}$ $i = 1, 2, \cdots, 20$

$$Y(z) = G(z)U(z)$$
$$(1 + az^{-2})Y(z) = (z^{-1} + bz^{-3})U(z)$$
$$y(k) + a\,y(k-2) = u(k-1) + b\,u(k-3)$$

$$k_0 = \max(p, q) + 1 = 3 + 1 = 4$$

$$y(4) - u(3) = \begin{bmatrix} -y(2) & u(1) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} y(4) - u(3) \\ y(5) - u(4) \\ \vdots \\ y(20) - u(19) \end{bmatrix} = \begin{bmatrix} -y(2) & u(1) \\ -y(3) & u(2) \\ \vdots & \vdots \\ -y(18) & u(17) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

$$y = H\theta$$

$$\hat{\theta}_{LS} = (H^T H)^{-1} H^T y$$

(Quiz 1) 将参数全部放右侧

2.2 Transformation among models

2.2.1 Transfer function

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

SISO system:

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

Write the above transfer function in controllable canonical form. (multiple sln)

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ -a_0 & -a_1 & \cdots & & -a_{n-1} \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} b_0 - a_0 b_n & b_1 - a_1 b_n & \cdots & b_{n-1} - a_{n-1} b_n \end{bmatrix}$$

$$D = [b_n]$$

one solution

A more common form is:

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ -a_0 & -a_1 & \cdots & & -a_{n-1} \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} b_0 & b_1 & \cdots & b_{n-2} & b_{n-1} \end{bmatrix}$$

$$D = [0]$$

another solution

① Example in (LN3 28)

② (HW4 5) (a) (b)

3. State space System properties

3.1 state space solution

$$x(k+1) = A(k)\,x(k) + B(k)\,u(k)$$
$$y(k) = C(k)\,x(k) + D(k)\,u(k)$$

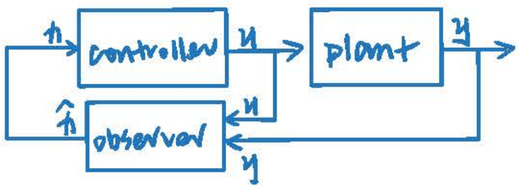
state trajectory: $x(0) = \hat{x}$

$$x(k) = A^k \hat{x} + \sum_{j=0}^{k-1} A^{k-j-1} B u(j)$$

e.g. $x(2) = A(A\hat{x} + Bu(0)) + Bu(1)$
$$= A^2\hat{x} + [ABu(0) + Bu(1)]$$

$$y(k) = \underbrace{CA^k\hat{x}}_{\text{zero input res}} + \underbrace{\sum_{j=0}^{k-1} CA^{k-1-j} Bu(j) + Du(k)}_{\text{zero state res}}$$

$\downarrow$
output trajectory



3.2 Internal stability & BIBO (External)

	definition	test
Internal (Asym)	$u_k \equiv 0$, $\|x_k\| \to 0 \;\; \forall x_0$	$\Rightarrow$ $\|\mathrm{eig}(A)\| < 1$
		$A^k \xrightarrow[\text{Jordan form}]{\text{Diagonalize}} A$
		$\Downarrow \quad \not\Uparrow$
External (BIBO)	Bounded $\|u_k\|$ $\Rightarrow$ Bounded $\|y_k\|$	$\Rightarrow$ poles of $H(z) < 1$
		$H(z) = C(zI - A)^{-1}B + D$

poles 是 eig 的一部分
有的可能分子分母消去了

(HW4 1) (HW4 2)

3.3 K-step reachability

Given (A, B), final state x_f, initial x_0

$$x(k) - A^k x_0 = [B \;\; AB \;\cdots\; A^{k-1}B] \begin{bmatrix} u(k-1) \\ u(k-2) \\ \vdots \\ u(0) \end{bmatrix}$$

that is to solve $x_f - A^k x_0 = M_k u_k$.

(Quiz 2) (if $x_f - A^k x_0 \in \mathrm{Col}(M_k)$)

3.4 controllability (def and test)

Controllability:

definition	test
Any $x_0 \to$ Any x_K in finite time $\Rightarrow$	$rank(M_c) = n$

$$M_c = [B \ AB \ A^2B \cdots A^{k-1}B]$$ (HW4 3)

3.5 Observability (def and test)

Observability:

definition	test
$\{u_k, y_k\} \to x_0$ determine $\Rightarrow$	$rank(M_o) = n$

$$M_o = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{k-1} \end{bmatrix} \qquad \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(k-1) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{k-1} \end{bmatrix} x_0$$

(Assume $u(k)=0$)

There is also K-step observable; like (HW4 4(b))

observable (A, C) $\Leftrightarrow$ controllable (A^T, C^T)

3.6 similarity transform

$A \Leftrightarrow PAP^{-1}$ $\quad \hat{M}_c = P^{-1}M_c$ (LN4 28)

4. Eigenvalue Assignment controller design

4.1 closed-loop dynamics under state feedback

state feedback controller = mapping

$$u_k = \mu(x_k)$$

closed loop system: $\begin{cases} x_{k+1} = f(x_k, \mu(x_k)) \triangleq f_{cl}(x_k) \\ y_k = h(x_k, \mu(x_k)) \triangleq h_{cl}(x_k) \end{cases}$

4.2 controllable canonical form ...

4.3 eigenvalue assignment in controllable canonical form

Linear controller: $u_k = -\underbrace{K}_{\text{feedback gain}} x_k$

$\Rightarrow$ for a linear closed loop system

$$x_{k+1} = (A - BK) x_k$$

Use "eigenvalue assignment" to design gain

$\Leftrightarrow$ Find K s.t. $eig(A-BK) = eig_{desired}$

Steps:

(1) If system (A, B) is controllable

(2) Find $\bar{K} = [K_1\ K_2]$ s.t. $eig(A - BK) = eig_{desired}$

(3) Check new $\bar{A}_{cl}$

(1) Example (LN5 13)

$$\bar{A} = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix} \quad \bar{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \lambda^*_{1,2} = \pm 0.5$$

step 1: $\Delta_{desired}(\lambda) = (\lambda - 0.5)(\lambda + 0.5) = \lambda^2 - 0.25$

$\Rightarrow \alpha_0^* = -0.25 \quad \alpha_1^* = 0$

step 2: $\alpha_0 = 1 \quad \alpha_1 = 2 \quad$ find $\bar{K} = [K_1\ K_2]$

$A - BK \Rightarrow$ new $\alpha_i^* \Rightarrow -\alpha_i - K_{i+1} = -\alpha_i^*$

$\Rightarrow K_{i+1} = \alpha_i^* - \alpha_i$

$K_1 = -0.25 - 1 = -1.25 \qquad K_2 = 0 - 2 = -2$

$K = [-1.25\ \ -2]$

step 3: $\bar{A}_{cl} = \bar{A} - \bar{B}\bar{K} = \begin{bmatrix} 0 & 1 \\ 0.25 & 0 \end{bmatrix} \Rightarrow \Delta_{cl} = \lambda^2 - 0.25$

$\Rightarrow \lambda_{cl} = \pm 0.5$

4.4 eigenvalue assignment for system not in canonical, how to assign eig

(1) General SISO system (A, B)

Under similarity transform:

$$\begin{cases} x(k) = P\bar{x}(k) \\ \bar{x}(k+1) = \bar{A}\bar{x}(k) + \bar{B}u(k) \\ \bar{A} = P^{-1}AP \qquad \bar{B} = P^{-1}B \\ \bar{M}_c = P^{-1}M_c \\ eig(A) = eig(\bar{A}) \\ \Delta_A(\lambda) = \Delta_{\bar{A}}(\lambda) \end{cases}$$

How to find P

$P^{-1} = \bar{M}_c M_c^{-1} \Rightarrow P = M_c \bar{M}_c^{-1}$

Step 5: (LN5 15)

(1) similarity transform, find P

$\Delta_A(\lambda) = \lambda^n + \alpha_{n-1}\lambda^{n-1} + \cdots + \alpha_1\lambda + \alpha_0$

$$\Delta_{\bar{A}}(\lambda) = \Delta_A(\lambda) \longrightarrow \bar{A} = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \vdots & \cdots & 1 \\ -\alpha_0 & -\alpha_1 & \cdots & -\alpha_{n-1} \end{pmatrix} \quad \bar{B} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

$\bar{M}_c \leftarrow (\bar{A}, \bar{B})$ $\quad M_c \leftarrow (A, B)$

$P = M_c \bar{M}_c^{-1} \longrightarrow$ found P

(2) Find $\bar{K}$ to assign $eig_{desired}$

(3) compute $K = \bar{K}P^{-1}$, found K

(4) compute $P^{-1}(A-BK)P$ to check

$\bar{A} - \bar{B}\bar{K} = P^{-1}AP - P^{-1}BKP = P^{-1}(A-BK)P$

(Project 2 b)

If B has multiple input, use 1 col (LNJ 17)

---

how to assign eig (LNJ 19)

Suppose we wand $T_s \le 5s$ $PO \le 35\%$

$$\begin{cases} \frac{4}{\zeta\omega_n} \le 5 \\ 100 e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} \le 35 \end{cases} \Rightarrow \begin{cases} \zeta\omega_n \ge 0.8 \\ \zeta \ge 0.32 \end{cases}$$

choose $\zeta = 0.5$ $\quad \zeta\omega_n = 1 \Rightarrow \omega_n = 2$.

$p_{1,2} = -1 \pm j\sqrt{3} \quad \Rightarrow T = 0.03$, DT: $z_{1,2} = e^{(-1 \pm \sqrt{3}j) 0.03}$

(LNJ 20)

5. Observer design

5.1 Observer structure

state vector is not available

because (A, B) is actually plugin (?)

$\Rightarrow$ generate estimate according to known system dynamics (directly use sensor data to compute would enlarge noise)

5.2 Error dynamics

$$\hat{x}(k+1) = \underbrace{A\hat{x}(k) + Bu(k)}_{\text{estimate at } k} + \underbrace{L[y(k) - C\hat{x}(k) - Du(k)]}_{\text{"suprise" correction term}}$$

L actually similar to the design of kalman gain

state estimate error: $e(k) = x(k) - \hat{x}(k)$

error dynamics: $e(k+1) = (A-LC)e(k)$

$$\begin{aligned} e(k+1) &= x(k+1) - \hat{x}(k+1) \\ &= Ax(k) + Bu(k) - \hat{x}(k+1) = \dots \\ &= A(x(k) - \hat{x}(k)) - LC(x(k) - \hat{x}(k)) \\ &= (A-LC)e(k) \end{aligned}$$

(LNJ 23)

5.3 Design observer gain matrix

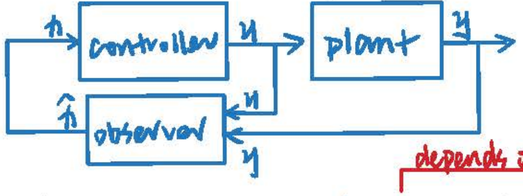
According to Duality Theorem

(A, C) observable $\Leftrightarrow$ (A^T, C^T) controllable

$\Rightarrow$ Design $eig(A^T - C^T\tilde{K}) = eig_{desired}$

$L = \tilde{K}^T$ (Proj2 8)

5.4 Output feedback design.



depends on

Design K, L to let $(A-BK)$, $(A-LC)$ meet the desired eigenvalues

$eig(A-CL)$ need to be faster than $eig(A-BK)$

Closed loop dynamics: (LN5 28)

$\begin{bmatrix} x(k) \\ e(k) \end{bmatrix} \Rightarrow$ joint state vector $\begin{bmatrix} x(k+1) \\ e(k+1) \end{bmatrix} = \begin{bmatrix} A-BK & BK \\ 0 & A-CL \end{bmatrix} \begin{bmatrix} x(k) \\ e(k) \end{bmatrix}$

$eig\left(\begin{bmatrix} A-BK & BK \\ 0 & A-CL \end{bmatrix}\right) = eig(A-BK)\, eig(A-CL)$

$\Rightarrow$ prove that K, L can be designed seperately

6. Probability

6.1 probability and conditional probability

① Probability Space $(\Omega, \mathcal{F}, P)$

event is a subset of sampling space Ω

event space $\mathcal{F}$ is all the subset of Ω

$P: \mathcal{F} \to [0,1]$ mapping $\mathcal{F}$ to $[0,1]$

(HW5 1)

② Conditional prob (LN7(1) 11)

(HW5 2) (HW5 3)

6.2 random variables and random vectors

6.3 jointly distributed random vectors and conditional expectation

① Expectation properties

$E(AX+BY) = AE(X) + BE(Y)$

$$E(X) = \begin{bmatrix} E(X_1) \\ \vdots \\ E(X_n) \end{bmatrix} \qquad E(X) \triangleq \int_{R^n} x f(x)\,dx$$
$$E(X) \triangleq \sum_x x \cdot \text{Prob}(X=x)$$

② jointly distributed

$$(X, Y) \sim f_{XY}(x, y)$$
$$p(x, y) = \int_A f_{XY}(x,y)\,dx\,dy$$

marginal density: (LN 7(1) 19)

$X \sim f_X(x)$ $\qquad$ $Y \sim f_Y(y)$

$$f_X(x) = \int_{R^m} f_{XY}(x,y)\,dy \qquad f_Y(y) = \int_{R^n} f_{XY}(x,y)\,dx$$

③ conditional density (LN 7(1) 20)

$$P(A|B) = \frac{P(A\cap B)}{P(B)} \Rightarrow p_{X|Y}(x=i \mid Y=j) = \frac{p_{XY}(x=i, Y=j)}{\underbrace{\sum_i p_{XY}(x=i, Y=j)}_{\text{Prob}(Y=j)}}$$

$$f_{X|Y}(x|y) = \frac{f_{XY}(x,y)}{f_Y(y)} \qquad f_X(x) = \int_{R^m} f_{X|Y}(x|y) f_Y(y)\,dy$$

(If X is independent of Y, $X \perp Y$, $f_{XY}(x,y) = f_X(x) f_Y(y)$)

④ conditional expectation

$$E(X|Y=y) \triangleq \int_{R^n} x f_{X|Y}(x,y)\,dx$$
$$E(X|Y=y) \triangleq \sum_i i \cdot \text{Prob}(x=i \mid Y=y)$$

$$E(X) = \sum_y E(X|Y=y) \cdot p_Y(Y=y)$$
$$E(g(X,Y)) = \sum_y E(g(X,Y)|Y=y) \cdot p_Y(Y=y)$$

(LN 7(1) 21-25)

6.4 covariance matrix

$$\text{Cov}(X, Y) = E\left((X - E(X))(Y - E(Y))^T\right)$$

meaning: correlation, if $\text{cov}(X, Y) = 0$, uncorrelated

$$\text{cov}(X, Y) = \begin{bmatrix} \text{cov}(X_1, Y_1) & \text{cov}(X_1, Y_2) & \cdots & \text{cov}(X_1, Y_m) \\ \text{cov}(X_2, Y_1) & \text{cov}(X_2, Y_2) & \cdots & \text{cov}(X_2, Y_m) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(X_n, Y_1) & \text{cov}(X_n, Y_2) & \cdots & \text{cov}(X_n, Y_m) \end{bmatrix}$$

properties:

$$\text{Cov}(X+a, Y+b) = \text{Cov}(X, Y)$$
$$\text{Cov}(X, Y) = \text{Cov}(Y, X)^T$$

$Cov(X_1+X_2, Y) = Cov(X_1, Y) + Cov(X_2, Y)$

$Cov(AX, BY) = A\,Cov(X, Y)\,B^T$

$Cov(X, Y) = 0$ if $X \perp Y$

$Cov(X) \triangleq Cov(X, X)$ is positive semidefinite

$Cov(AX+BY) = A\Sigma_X A^T + A\Sigma_{XY}B^T + B\Sigma_{XY}^T A^T + B\Sigma_Y B^T$

(LN 7 (1) 32)

7. Kalman filter

7.1 Minimum Mean squared estimation

An estimator is a function that maps each measurement $Y=y$ to an estimate $\hat{x}$

We use MSE (mean-squared error) to judge the performance of an estimator $E(\|\phi(Y)-X\|^2)$

$$y \longrightarrow \underset{\phi(y)}{\boxed{\text{estimator}}} \longrightarrow \hat{x}$$

Example (LN 7 (2) 5) (LN 7 (2) 17-18)

$\Rightarrow$ MMSE is an estimator that minimize MSE

$E(\|\phi(Y)-X\|^2 \mid Y=y)$ minimize when

$\hat{x}_{MMSE} = \phi_{MMSE}(y) = E(X \mid Y=y)$

7.2 Gaussian Random Vectors (condition)

Calculate $E(X \mid Y=y)$ for jointly Gaussian random vector $\longrightarrow$ simple form ~~($\int x f_{X|Y}(x|y)dx$)~~

$X \sim N(\mu, \sigma)$

1D Gau pdf: $f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

nD Gau pdf:

$$f_X(x) = \frac{1}{(2\pi)^{\frac{n}{2}}(\det(\Sigma))^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\}$$

σ^2 and Σ are covariance or variance, not standard deviation σ (Qui 8.4 2)

... is also Gaussian (LN 7 (2) 11-12)

$X \sim N(\mu, \Sigma) \longrightarrow Z = AX+b \sim N(A\mu+b, A\Sigma A^T)$

Conditional distribution of X given $Y=y$

$X \mid Y=y \sim N(\mu_{X|Y=y}, \Sigma_{X|Y=y})$

where

$$\begin{cases} \mu_{x|y=y} = \mu_x + \Sigma_{xy}\Sigma_y^{-1}\underbrace{(y-\mu_y)}_{\text{surprise term}} \\ \Sigma_{x|y=y} = \Sigma_x - \Sigma_{xy}\Sigma_y^{-1}\Sigma_{yx} \end{cases}$$

(Quiz 4 3) (LN7 (2) 15-16)

7.3 Kalman Filter Derivations

$$\begin{cases} x_{k+1} = A_k x_k + B_k u_k + w_k \\ y_k = C_k x_k + D_k u_k + v_k \end{cases}$$

$w_k \sim N(0, Q_k)$ process noise (tuned)

$v_k \sim N(0, R_k)$ measurement noise

Prediction

Use prior knowledge to predict k state: $\underbrace{\hat{x}_{k|k-1}}_{\text{prior knowledge of k state}} = A_{k-1}\underbrace{\hat{x}_{k-1}}_{\text{posterior of k-1 state}} + B_{k-1}u_{k-1}$

the covariance of the prior prediction result: $\underbrace{P_{k|k-1}}_{\text{prior covariance of } \hat{x}_{k|k-1}} = A_{k-1}\underbrace{P_{k-1}}_{\text{posterior covariance of } \hat{x}_{k-1}}A_{k-1}^T + Q_{k-1}$

Measurement Update

Compute Kalman gain:

$$K_k = P_{k|k-1}C_k^T(C_kP_{k|k-1}C_k^T + R_k)^{-1}$$

$$= \frac{P_{k|k-1}C_k^T}{C_kP_{k|k-1}C_k^T + R_k} \quad (R_k \text{ is known})$$

$P_{k|k-1}C_k^T \longrightarrow cov(x_{k+1}, y_{k+1} | Y_k)$

$\underbrace{C_kP_{k|k-1}C_k^T}_{\text{covariance of prediction}} \longrightarrow cov(y_{k+1}|Y_k)$ vs. $\underbrace{R_k}_{\text{covariance of measure}}$

Update estimate and error convariance:

$$\underbrace{\hat{x}_k}_{\text{posterior}} = \underbrace{\hat{x}_{k|k-1}}_{\text{prior}} + K_k\underbrace{(y_k - C_k\hat{x}_{k|k-1} - D_ku_k)}_{\text{surprise}}$$

$$\Downarrow$$

$$P_k = (I - K_kC_k)P_{k|k-1}$$

A 1-D example: (LN7 (2) 34)

7.4 EKF (LN7 (B) 7)

EKF Prediction step: $F_k \triangleq \frac{\partial f}{\partial x}\Big|_{\hat{x}_k, u_k}$

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1}, u_{k-1})$$

$$P_{k|k-1} = F_{k-1} P_{k-1} F_{k-1}^T + Q_{k-1}$$

EKF Measurement Update: $H_k = \frac{\partial h}{\partial x}\Big|_{\hat{x}_{k|k-1}, u_k}$

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1}$$

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k (y_k - h(\hat{x}_{k|k-1}, u_k))$$

$$P_k = (I - K_k H_k) P_{k|k-1}$$

7.5 Kalman filter problems

7.5.1 tracking

example of tracking (LN7 (B) 11)

example of joint state and paramter estimation

(LN7 (B) 14)

(Project 3) K_k is between 0 and C^{-1}

8. Dynamic Programming and LQR

8.1 general discrete-time optimal control problem

8.2 Dynamical programming

$$x \rightarrow \boxed{\mu_k^*(x_k)} \rightarrow u$$

Encode objective into "Cost Function + constraint"

$$\underbrace{J_N(z, u)}_{\text{cost function}} = \sum_{k=0}^{N-1} \{ l(x_k, u_k), g(x_N) \}$$

s.t. ©
$$\begin{cases} x_{k+1} = f(x_k, u_k) \rightarrow \text{dynamics} \\ x_k \in X \rightarrow \text{robot not fall down, state ©...} \\ u_k \in U(x_k) \rightarrow \text{state dependent constraint} \end{cases}$$

Shortest path problem

DP: $V_j(z) \triangleq \min_{\substack{u_0 \cdots u_{j-1} \\ \text{s.t. ©}}} J_j(z, u)$

Start from 0 horizon: $V_0(z) = g(z)$

From j to $j+1$: $V_{j+1}(z) = \min_{u \in U(z)} \{ l(z, u), \underbrace{V_j(f(z, u))}_{\text{future state}} \}$

(for all z, value iteration)

$\downarrow$

curse of dim for $z \in \mathbb{R}^n$ (n too large)

$$\mu_{j+1}(z) = \arg\min_{u \in U(z)} \{ l(z, u), V_j(f(z, u)) \}$$

8.4 Linear Quadratic Regulator

Exception / special case of former DP:

$$J_N(z,u) = \Big\{\underbrace{\sum_{k=0}^{N-1}\left(x_k^T Q x_k + u_k^T R u_k\right)}_{\text{running cost}}\Big\} + \underbrace{x_N^T Q_f x_N}_{\text{terminal cost}}$$

s.t. ©
$$\begin{cases} x_{k+1} = A x_k + B u_k & \text{(dynamics)} \\ x_k \in \mathbb{R}^n \quad u_k \in \mathbb{R}^m & \text{(no constraint)} \end{cases}$$

$V_j(z) = z^T P_j z \quad \mu_j(z) = -K_j z$ (optimal !)

start from $P_0 = Q_f \qquad P_{j+1} = \underbrace{\rho(P_j)}_{\text{Riccati mapping (map } j \text{ to } j+1)}$

N-horizon: $x_0 = z \xrightarrow[\mu_N^*(x_0)]{u_0^*} x_1^* \xrightarrow[\mu_{N-1}^*(x_1^*)]{u_1^*} x_2^* \longrightarrow \cdots$

control law def: when you have N-1 steps left you are at state x_1^*, you optimal control should be u_1^*; If add a noise to $x_1^* \longrightarrow x_1' = x_1^* + \delta$, the optimal control becomes $\hat{u}_1^* = \mu_{N-1}^*(x_1')$

① example of shortest path (1):

g			

Problem

Running cost: 1

terminal cost: $\begin{cases} 0 & g \\ \infty & \text{otherwise} \end{cases}$

0	∞	∞	∞
∞	∞	∞	∞
∞	∞	∞	∞
0	∞	∞	∞

→

0	1	∞	∞
1	∞	∞	∞
1	∞	∞	∞
0	1	∞	∞

→

0	1	2	∞
1	2	∞	∞
1	2	∞	∞
0	1	2	∞

↓

0	1	2	3
1	2	3	∞
1	2	3	∞
0	1	2	3

←

0	1	2	3
1	2	3	4
1	2	3	4
0	1	2	3

←

0	1	2	3
1	2	3	4
1	2	3	4
0	1	2	3

② (HW7 1,2)