8/8 ME132 Discussion!

Bode Plots Review

- Relates the magnitude (amplification) and phase (time offset) to the frequency of the input signal

- Bandwidth Frequency → the frequency corresponding to a specific decibel drop in the magnitude.

- **(Key Idea for Transient Design)** Multiplying the transfer function by a constant gain K shifts the gain bode plot vertically, without other deformations!

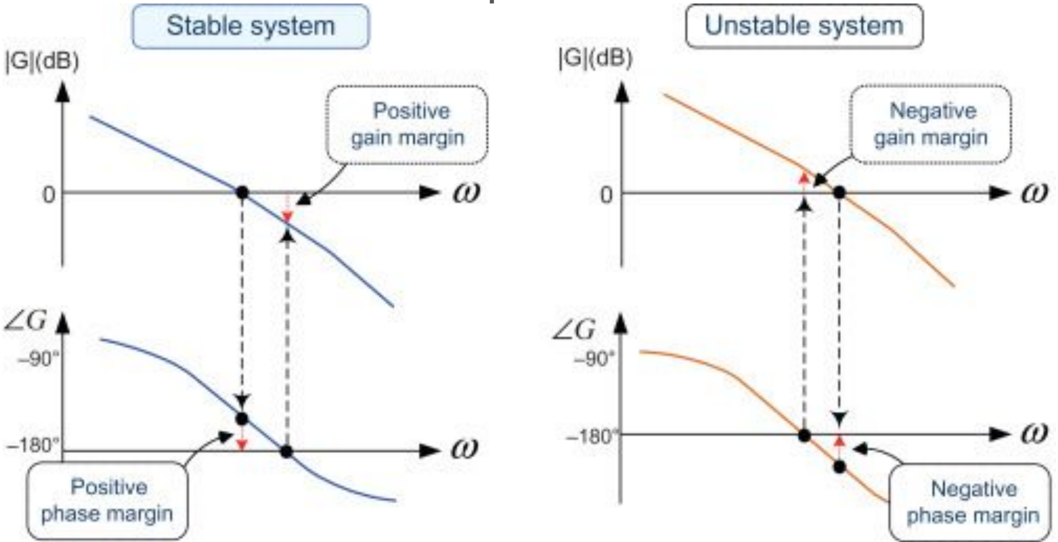
Bode Plot Transient Design

Frequency Response Design

↳ Type of design that cascades a compensator to the feedback loop to shape FR function!

↳ A way to design/improve stability and Transient Response Specifications

- Gain Margin Requirement
    - Find the phase crossover frequency ω.
    - Find the gain K to shift the gain margin at ω such that it is equal to our desired value.
        - Solve for the gain at the crossover frequency ω in terms of K and set it equal to our desired value



Bode Plot Transient Design

- Phase Margin Requirement
  - Choose K such that gain is 0dB (1) as $\omega \rightarrow 0$, and plot the bode plots for gain and phase.
  - Find ω on the bode plot that yields the desired phase margin (CD)
  - Adjust K such that the point B is now A on the gain plot (at ω, the gain is now 0db)

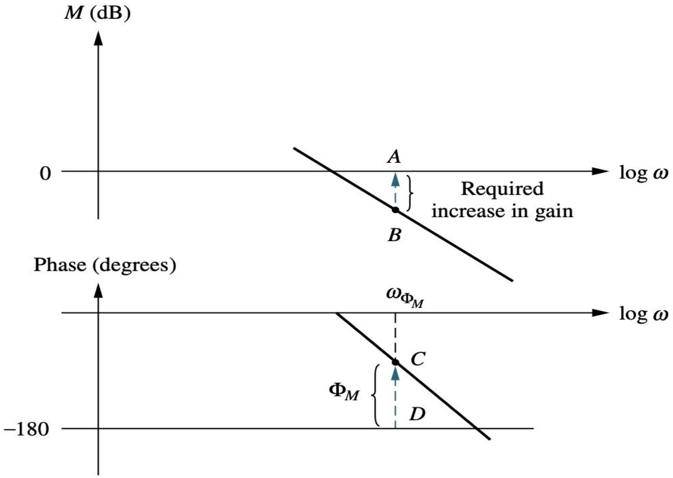
Design Procedure

① Draw Bode Plots for like $K=1$

② Use $\zeta = \frac{-\ln(\%OS/100)}{\sqrt{\pi^2 + \ln^2(\%OS/100)}}$ and $\Phi_M = \arctan\left(\frac{2\zeta}{\sqrt{-2\zeta^2 + \sqrt{1+4\zeta^4}}}\right)$

③ Find ω_{Φ_M} on Bode phase plot that gives desired PM

④ Change gain K by an amount to force magnitude curve to go through 0 dB at ω_{Φ_M}.



Bode Plot Transient Design Example

4. Given the unity feedback system of Figure P11.1 with

$$G(s) = \frac{K(s+10)(s+15)}{s(s+2)(s+5)(s+20)}$$

do the following: [Section: 11.2]

a. Use frequency response methods to determine the value of gain, K, to yield a step response with a 20% overshoot. Make any required second-order approximations.

b. Use MATLAB or any other computer program to test your second-order approximation by simulating the system for your designed value of K. MATLAB ML

① Determine Required PM

$$\zeta = \frac{-\ln(0.2)}{\sqrt{\pi^2 + \ln^2(0.2)}} = 0.456$$

$$\Phi_M = \arctan\left(\frac{2\zeta}{\sqrt{-2\zeta^2 + \sqrt{1+4\zeta^4}}}\right)$$

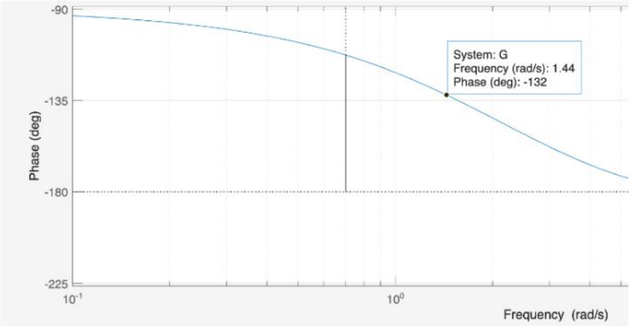
$$= \arctan\left(\frac{2(0.456)}{\sqrt{-2(0.456)^2 + \sqrt{1+4(0.456)^4}}}\right) = 48.15^\circ$$

② Find frequency ω_{Φ_M}

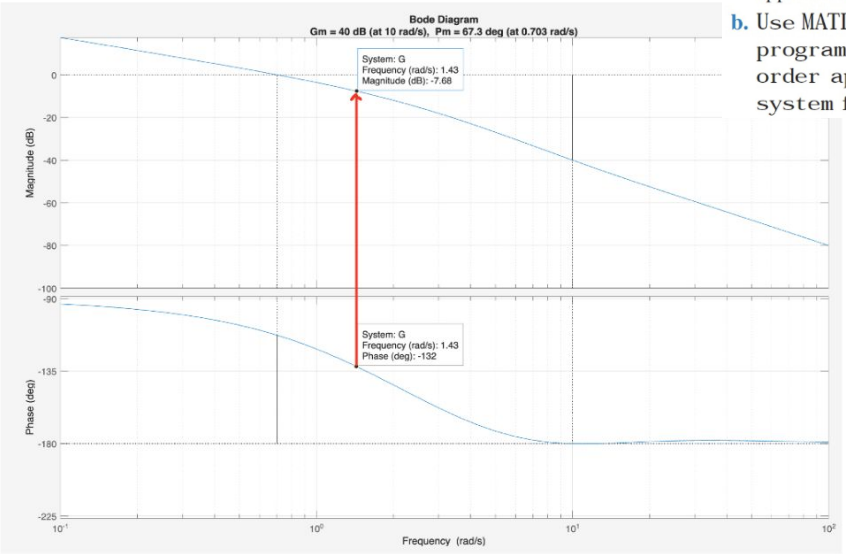
$$\phi_{req} = \Phi_M - 180^\circ$$

$$= 48.15^\circ - 180^\circ = -131.85^\circ \approx -132^\circ$$

$$\Rightarrow \omega_{\Phi_M} = 1.44 \text{ rad/s}$$



③ Solve for Gain



$\omega_{\Phi_M} = 1.44$ rad/s corresponds to a magnitude of -7.43 dB

$$20\log_{10}(K) = 7.43$$

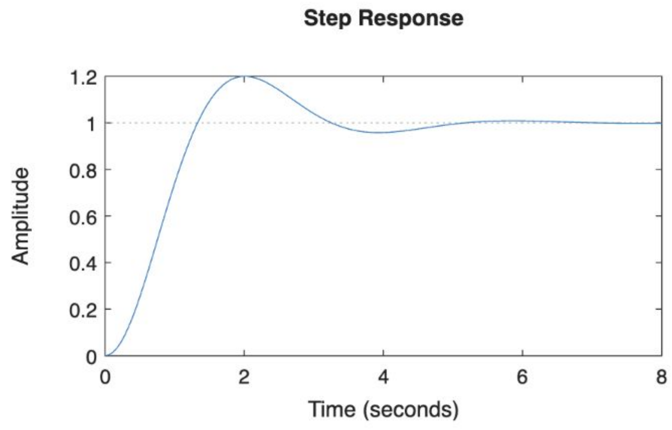
$$\log_{10} K = \frac{7.43}{20} \Rightarrow K = 10^{\frac{7.43}{20}}$$

$$\boxed{K = 2.35}$$

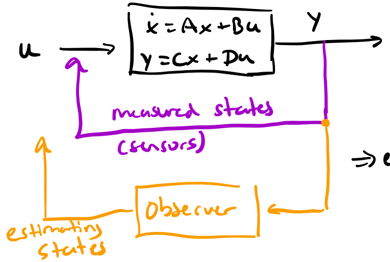
Bode Plot Transient Design Example

```
zeta = sqrt(log(0.2)^2/(pi^2+(log(0.2))^2))
s = tf('s')
pm = rad2deg(atan(2*zeta/(sqrt(-2*zeta^2+sqrt(1+4*zeta^4)))))
G= ((s+10)*(s+15))/(s*(s+2)*(s+5)*(s+20))

[Gm, Pm, wcg, wcp] = margin(G)
[gain, phase, w] = bode(G);%list of gain and phase and corresponding frequencies
phase = squeeze(phase);

wdes = interp1(phase, w, -180+pm) %linearly interpolate between points to find desired phase!

currgain = norm(evalfr(G, wdes*i)) %evaluate current gain at that point

desK = 1/currgain %we want gain at 0dB = 1 at this point, so desired K is 1/current

T_cl = feedback(desK*G, 1)
stepinfo(T_cl)
step(T_cl) %its goated!!
```

```
ans = struct with fields:
         RiseTime: 0.8851
    TransientTime: 4.6434
     SettlingTime: 4.6434
      SettlingMin: 0.9017
      SettlingMax: 1.1997
        Overshoot: 19.9744
       Undershoot: 0
             Peak: 1.1997
         PeakTime: 1.9817
```



Observers & Controllers



Say you only want to use 1 sensor for position, but the critical states in state vector are velocity + acceleration. $\Rightarrow$ estimate v and a using observer!

- Observer → We don't know the actual state of the system x(t).
  - How can we best estimate it using our internal model and measurements? $\Rightarrow$ allows us to reduce hardware complexity in system

i.e. Luenberger Observer: $\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})$ ← observer gain! error $e = x - \hat{x} \Rightarrow \dot{e} = (A - LC)e$ choose λ via pole-placement to make $\hat{x} \to x$ fast!

- Controller → how do we best choose an input u(t)?
  - How can we design this to meet our desired requirements?

gain on reference to reduce ss error.

↳ controller computes: $u = \underbrace{-Kx}_{\text{feedback}} + K_r r(t)$

Pole placement: we choose closed loop poles for each of these! These will define our observers and controllers.

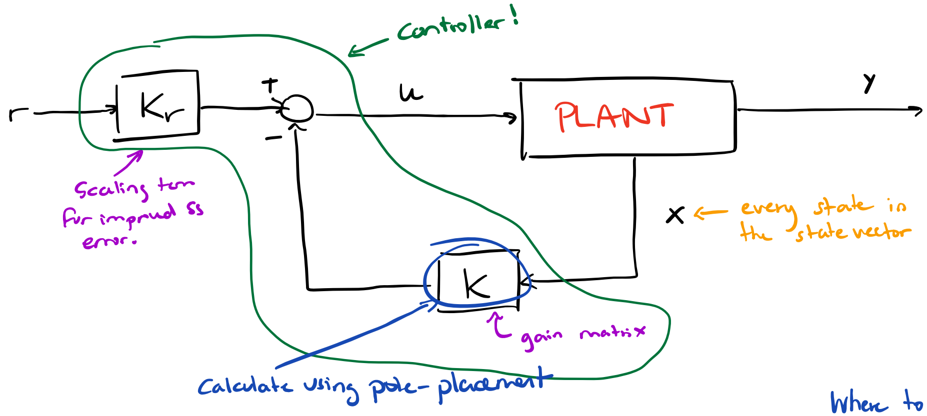
- What if we don't know CL poles?
  - LQR and Kalman Filters (Not in scope)

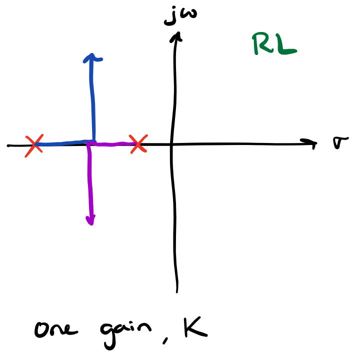
optimal

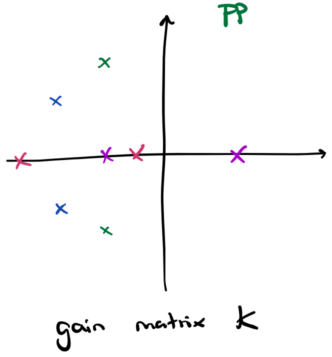
↳ KF is optimal linear estimator. If measurements + disturbances are stochastic!

Pole Placement Intuition

Pole placement = Fancy Root Locus!







$u = r \cdot K_r - Kx$

sub into SEE to close the loop

$\dot{x} = Ax + B(rK_r - Kx)$

$\dot{x} = Ax - BKx + BrK_r$

$\dot{x} = \underbrace{(A - BK)}_{\text{Closed loop A matrix}}x + BrK_r$

Where to place poles/eigenvalues?

↳ if you have high order system, keep 2 poles closer to jω axis than other i.e. 10 times further right than other poles

↳ These are dominant poles (slower)

↳ If you move to far to the left, from original OL poles, you might not have the ability to move it. ⇒ higher gain.

Feedback Controller Design (pole placement)

We are given a state space system whose poles are in undesirable positions.

- The poles of matrix A are in unfavorable positions
- Feedback controller → $u = -\mathbf{K}x$

CL A matrix

$$\implies \mathbf{A}x + \mathbf{B}u = \mathbf{A}x + \mathbf{B}(-\mathbf{K}x) = (\mathbf{A} - \mathbf{B}\mathbf{K})x$$

- We want to design K such that the poles of **(A-BK)** are at our desired positions!

Procedure

1. Find the desired characteristic polynomial from desired poles.
2. Let $\mathbf{K} = [k_1 \;\; k_2 \;\; k_3]$
3. Solve for the characteristic equation of $(\mathbf{A} - \mathbf{B}\mathbf{K})$ in terms of $k_1 \;\; k_2 \;\; k_3$ and s.
4. Match coefficients compared to the desired characteristic polynomial.

Pole Placement Example

6. Given the following open-loop plant: [Section: 12.2]

$$G(s) = \frac{100(s+2)(s+25)}{(s+1)(s+3)(s+5)}$$

design a controller to yield 10% overshoot with a peak time of 0.5 second. Use the controller canonical form for state-variable feedback.

$$G(s) = \frac{100(s+2)(s+25)}{(s+1)(s+3)(s+5)}$$

① Rewrite in CCF

$$G(s) = \frac{100(s^2+27s+50)}{s^3+9s^2+23s+15}$$

$$= \frac{100s^2+2700s+5000}{s^3+9s^2+23s+15} \sim H(s) = \frac{\beta_{n-1}s^{n-1}+\cdots+\beta_1 s+\beta_0}{s^n+\alpha_{n-1}s^{n-1}+\cdots+\alpha_1 s+\alpha_0}+d$$

$$\Sigma = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{cccccc|c} 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 \\ -\alpha_0 & -\alpha_1 & -\alpha_2 & \cdots & -\alpha_{n-2} & -\alpha_{n-1} & 1 \\ \hline \beta_0 & \beta_1 & \beta_2 & \cdots & \beta_{n-2} & \beta_{n-1} & d \end{array}\right]$$

we have $n=3$, then by pattern matching

$$\Sigma = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{ccc|c} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -15 & -23 & -9 & 1 \\ \hline 5000 & 2700 & 100 & 0 \end{array}\right]$$

just use tf2ss to save yourself!

② Solve for A-BK

$$K = [K_1 \quad K_2 \quad K_3]$$

$$A-BK = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -(15+K_1) & -(23+K_2) & -(9+K_3) \end{bmatrix}$$

③ Find Characteristic Equation

$$\det(sI-(A-BK)) = 0$$

$$\det\left(\begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -(15+K_1) & -(23+K_2) & -(9+K_3) \end{bmatrix}\right) = 0$$

$$\det\begin{pmatrix} s & 1 & 0 \\ 0 & s & 1 \\ -(15+K_1) & -(23+K_2) & s+(9+K_3) \end{pmatrix} = 0$$

$$\Rightarrow s^3 + (9+K_3)s^2 + (23+K_2)s + (15+K_1) = 0$$

④ Determine Equivalent Characteristic Equation

↳ Desired: $\%OS = 10\%$

$T_p = 0.5$

$$\zeta = \frac{-\ln(0.1)}{\sqrt{\pi^2+\ln^2(0.1)}} = 0.59$$

$$T_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} = 0.5$$

↳ rearrange for $\omega_n \Rightarrow \omega_n = 7.78$ rad/s

Pole Placement Example

6. Given the following open-loop plant: [Section: 12.2]

$$G(s) = \frac{100(s+2)(s+25)}{(s+1)(s+3)(s+5)}$$

design a controller to yield 10% overshoot with a peak time of 0.5 second. Use the controller canonical form for state-variable feedback.

$10\%\ PO \Rightarrow \zeta$

$$10\%\ PO = 100\% \times e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}$$

$$\Rightarrow \zeta = \sqrt{\frac{(\ln 0.1)^2}{\pi^2 + (\ln 0.1)^2}} = 0.5912$$

$$T_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} = 0.5 \Rightarrow \omega_n = \frac{\pi}{(0.5)\sqrt{1-\zeta^2}} = 7.7901$$

desired 2nd order poles!

$$-\zeta\omega_n \pm \omega_n\sqrt{1-\zeta^2}\,j = -4.6052 \pm 6.2832i = s_1, s_2$$

*key insight, note that there is a zero at −2 hindering our 2nd order approx!

∴ rather than place the 3rd pole further away, cancel the zero at −2!

$\Rightarrow s_3 = -2$

desired char poly:

$(s+s_1)(s+s_2)(s+s_3)$

$= s^3 + 11.2103s^2 + 79.10672s + 121.3720$

Convert Given Form to State Space:

$$G = \frac{100s^2 + 2700s + 5000}{s^3 + 9s^2 + 23s + 15}$$

CCF
$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -15 & -23 & -9 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$C = [5000 \;\; 2700 \;\; 100] \quad D = 0$$

MATLAB:

k=place(A, B, p)

done!

By "Hand":

Let $K = [k_1 \;\; k_2 \;\; k_3]$

$$A-BK = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -15 & -23 & -9 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ k_1 & k_2 & k_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -15-k_1 & -23-k_2 & -9-k_3 \end{bmatrix}$$

Char. Poly = $\det(sI-(A-BK)) =$

$s^3 + (k_3+9)s^2 + (k_2+23)s + (k_1+15)$

$k_3 + 9 = 11.2103 \qquad k_2 + 23 = 79.1067 \qquad k_1 + 15 = 121.372$

$$\boxed{k_3 = 2.21 \qquad k_2 = 56.1067 \qquad k_1 = 106.372}$$

Pole Placement Example (MATLAB)

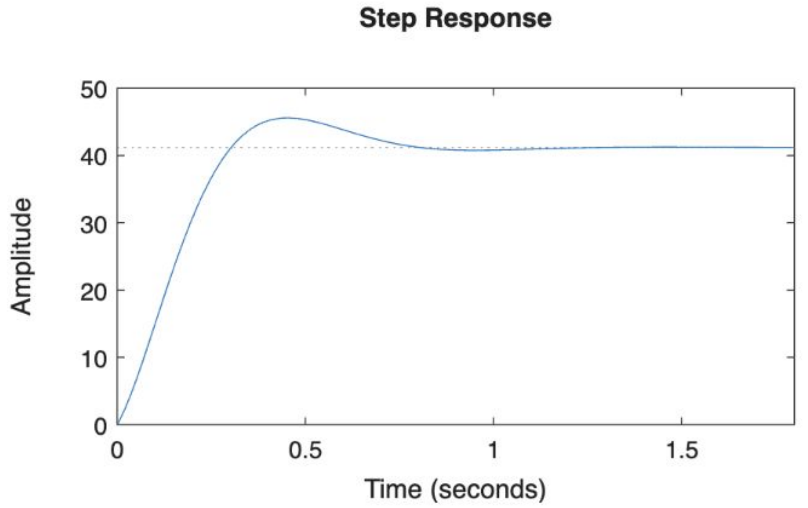
6. Given the following open-loop plant: [Section: 12.2]

$$G(s) = \frac{100(s+2)(s+25)}{(s+1)(s+3)(s+5)}$$

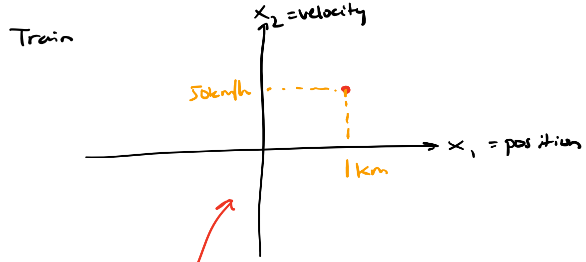
design a controller to yield 10% overshoot with a peak time of 0.5 second. Use the controller canonical form for state-variable feedback.

```
%specs
zeta = sqrt((log(0.1)^2)/(pi^2+(log(0.1)^2))) %zeta from PO
Tp = 0.5
wn = pi/(Tp*sqrt(1-zeta^2))

%desired poles based on spec
s1 =-zeta*wn + sqrt(1-zeta^2)*wn*i
s2 = -zeta*wn - sqrt(1-zeta^2)*wn*i
s3 = -2

%construct state space
s=tf('s')
G = 100*(s+2)*(s+25)/((s+1)*(s+3)*(s+5))
num = G.Numerator{1}
den = G.Denominator{1}
A = [0 1 0; 0 0 1; -15 -23 -9];
B = [0; 0; 1]
C = [5000 2700 100]
D = 0

K = place(A, B, [s1, s2, s3]) %find K using place()

[numcl, dencl] = ss2tf(A-B*K, B, C, D) %feedback system!
Gcl = tf(numcl, dencl)

stepinfo(Gcl)
step(Gcl)
```

```
ans = struct with fields:
        RiseTime: 0.2205
   TransientTime: 0.7173
    SettlingTime: 0.7173
     SettlingMin: 37.6734
     SettlingMax: 45.5726
       Overshoot: 10.6247
      Undershoot: 0
            Peak: 45.5726
        PeakTime: 0.4500
```



Controllability/Observability



Controllability (Reachability)

there exists control signals which allow the system to reach any state in a finite amt. of time!

- Can we steer the system from any initial state to any final state?
- defined by the Controllability Matrix (Doesn't need to maintain state!)

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

Observability

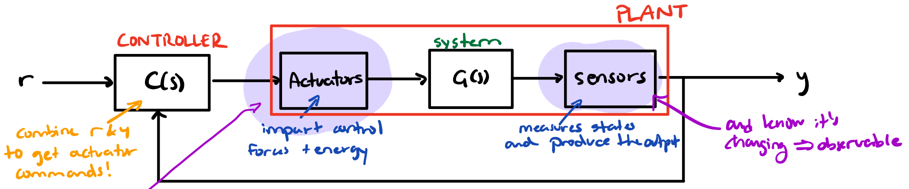
critical states (i.e. states in state vector)

- Given any measurement y(t), can we know the exact true state x(t)? (All states can be known from outputs of a system i.e. use appropriate selection of sensors + sensor locations)
- defined by the Observability Matrix

↳ impractical to know every state! i.e. temp of train is an irrelevant state useless for controlling $\vec{x}$ and $\vec{v}$

$$\mathbf{O} = [\mathbf{C}\ \mathbf{CA}\ \mathbf{CA}^2 \cdots\ \mathbf{CA}^{n-1}]^\top$$

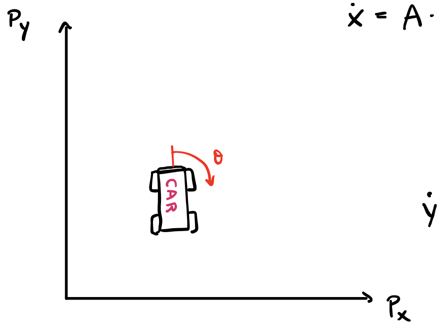
Controllability/Observability Intuition



[controller design will fail if your sys. doesn't have appropriate actuators that can affect the right parts of the sys OR not the right sensors to measure the right states]

you need to be able to influence the system ⇒ controllable!

∴ Cont. + Obs are conditions of how system interacts with actuators + sensors!



$$\dot{x} = A \cdot \begin{bmatrix} P_x \\ P_y \\ \dot{P}_x \\ \dot{P}_y \\ \theta \\ \dot{\theta} \end{bmatrix} + B \,[\text{steering} \;\; \text{pedals}]$$

$$\dot{y} = C \cdot \begin{bmatrix} P_x \\ P_y \\ \dot{P}_x \\ \dot{P}_y \\ \theta \\ \dot{\theta} \end{bmatrix} + D \,[\text{steering} \;\; \text{pedals}]$$

① Close your eyes!

⇒ eliminates C matrix so you don't know any states, but you can still apply gas and steer!

② Remove steering wheel + pedals (ice patch)

⇒ eliminates B matrix so not controllable but you can see the speedometer and track position by looking outside

∴ If any one critical state is uncontrollable / unobservable then the entire system is.

Controllability Example

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u = \begin{bmatrix} -2 & -1 & -3 \\ 0 & -2 & 1 \\ -7 & -8 & -9 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} u$$

$$y = \mathbf{C}\mathbf{x} = [\,4 \quad 6 \quad 8\,]\,\mathbf{x}$$

$$C = [\,B \;\; AB \;\; A^2B \;\cdots\; A^{n-1}B\,]$$

since $A \in \mathbb{R}^{3\times 3}$, $n = 3$

$$C = [\,B \;\; AB \;\; A^2B\,]$$

$$= \begin{bmatrix} 2 & -11 & 142 \\ 1 & 0 & -40 \\ 2 & -40 & 437 \end{bmatrix}$$

$$AB = \begin{bmatrix} -2 & -1 & -3 \\ 0 & -2 & 1 \\ -7 & -8 & -9 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} -11 \\ 0 \\ -40 \end{bmatrix}$$

$$A^2B = \begin{bmatrix} -2 & -1 & -3 \\ 0 & -2 & 1 \\ -7 & -8 & -9 \end{bmatrix} \begin{bmatrix} -11 \\ 0 \\ -40 \end{bmatrix} = \begin{bmatrix} 142 \\ -40 \\ 437 \end{bmatrix}$$

$\text{rank}(C) = 3 \Rightarrow$ full rank $\therefore$ fully controllable

MATLAB : `C= ctrb(A, B); rank(C)`

Observability Example

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u = \begin{bmatrix} -2 & -1 & -3 \\ 0 & -2 & 1 \\ -7 & -8 & -9 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} u$$

$$y = \mathbf{C}\mathbf{x} = [4 \quad 6 \quad 8]\,\mathbf{x}$$

$$O = [C \quad CA \quad CA^2 \cdots CA^{n-1}]^T$$

since $A \in \mathbb{R}^{3\times3}$, $n = 3$

$$O = [C \; CA \; CA^2]^T$$

$$= \begin{bmatrix} 4 & 6 & 8 \\ -64 & -80 & -78 \\ 674 & 848 & 814 \end{bmatrix}$$

rank$(O) = 3 \Rightarrow$ full rank $\therefore$ fully observable

$$CA = [4 \quad 6 \quad 8]\begin{bmatrix} -2 & -1 & -3 \\ 0 & -2 & 1 \\ -7 & -8 & -9 \end{bmatrix}$$

$$= [-64 \quad -80 \quad -78]$$

$$CA^2 = [-64 \quad -80 \quad -78]\begin{bmatrix} -2 & -1 & -3 \\ 0 & -2 & 1 \\ -7 & -8 & -9 \end{bmatrix}$$

$$= [674 \quad 848 \quad 814]$$

MATLAB: O = obsv(A, C); rank(O)

Helpful MATLAB tools!

- `ctrb(A, B)`: generates controllability matrix for a system
- `place(A, B, p)`: calculates K matrix given A and B, such that poles are placed at desired poles p
- `obsv(A, C)`: generates observability matrix for a system

Final Review!