- final exam in classroom: 10:30–12:20p Thu Dec 12
- notes on one (1) sheet of 8.5×11 in paper permitted (no other materials)

- comprehensive up through end of lecture Tue Dec 3
  - covers any concept/technique from lecture or homework

  * no programming

---

week 6: state and output feedback

goal: design stabilizing controllers and estimating observers

1°. state feedback [AMv2 Ch 7] [Nv7 Ch 12.2]

1!. stabilization

- given $\dot{x} = Ax + Bu$, $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$

  want $K \in \mathbb{R}^{p\times n}$ s.t. $\underbrace{\mathrm{Re}(\lambda(A-BK)) < 0}_{\text{all eigenvalues in left-half plane}}$

  i.e. closed-loop system $\dot{x} = Ax - BKx = (A-BK)x$ obtained w/ linear state feedback $u = -Kx$ is stable

- more generally, if we want the closed-loop system to have $\{\lambda_i\}_{i=1}^m$ as its set of eigenvalues then we want the characteristic polynomial

$$
\begin{aligned}
a(s) &= \det(sI - (A-BK)) \\
&= s^n + a_1 s^{n-1} + \cdots + a_{n-1}s^1 + a_n \\
&= (s-\lambda_1)(s-\lambda_2)\cdots(s-\lambda_n) \\
&= \prod_{i=1}^{n}(s-\lambda_i)
\end{aligned}
$$

~~1². integral feedback~~ not covered

2°. output feedback [AMv2 Ch 8] [Nv7 Ch 12.5]

2¹. observer design

- to estimate the state of an LTI system using only its output

$$\dot{x} = Ax + Bu \qquad y = Cx + Du$$

we'll construct another LTI system termed an <u>observer</u>:

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}) \quad \hat{y} = C\hat{x} + Du$$

→ the state $\hat{x}$ of this system is known to us
 – we implement a simulation of it

– to see why this works, consider
the dynamics of the error $e = x - \hat{x}$

$$\begin{aligned}\dot{e} &= \dot{x} - \dot{\hat{x}} \\ &= (Ax + Bu) - (A\hat{x} + Bu + L(y - \hat{y})) \\ &= Ax - A\hat{x} - L(Cx - C\hat{x}) \\ &= (A - LC)e\end{aligned}$$

* if $\text{Re}\,\lambda(A - LC) < 0$, then error dynamics are asymptotically stable, $\dot{e} \to 0$, which means observer state converges to real state: $\hat{x} \to x$!

2². closing the loop ⟶ see HW7 part (h.)

---

week 7: transfer functions

goal: frequency-domain tools and concepts for analysis & control

1°. frequency-domain modeling

1¹. transfer function of an LTI system [AMv2 Ch 6.3, 9.2] [Nv7 Ch 4.11]

- $x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)\,d\tau$

- $\underbrace{x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)\,d\tau}_{\text{convolution equation}}$

- $y(t) = Cx(t) + Du(t)$

$$= \underbrace{Ce^{At}x(0)}_{\underline{\text{homogeneous}}\ \text{response}} + \underbrace{\int_0^t Ce^{A(t-\tau)}Bu(\tau)\,d\tau + Du(t)}_{\underline{\text{particular}}\ \text{response}}$$

- if $u(t) = e^{st}$ then $y(t) = Ce^{At}\left(x(0) - (sI-A)^{-1}B\right)$

$$+ \left[C(sI-A)^{-1}B + D\right]e^{st}$$

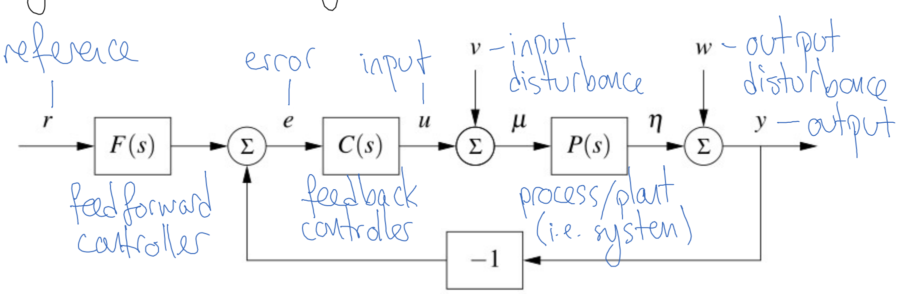
  - if A is stable, then

$$y \to \left(C(sI-A)^{-1}B + D\right)e^{st}$$

$$= \underbrace{G_{yu}(s)}_{\text{transfer function}}\, u$$

1.2 block diagrams [AMv2 Ch 9.4] [Nv7 Ch 5]

- general feedback diagram



$$e = \underbrace{\frac{F}{1+PC}}_{G_{er}}r - \underbrace{\frac{P}{1+PC}}_{G_{ev}}v - \underbrace{\frac{1}{1+PC}}_{G_{ew}}w$$

1.3 poles & zeros [AMv2 Ch 9.5] [Nv7 Ch 4.2, 4.10]

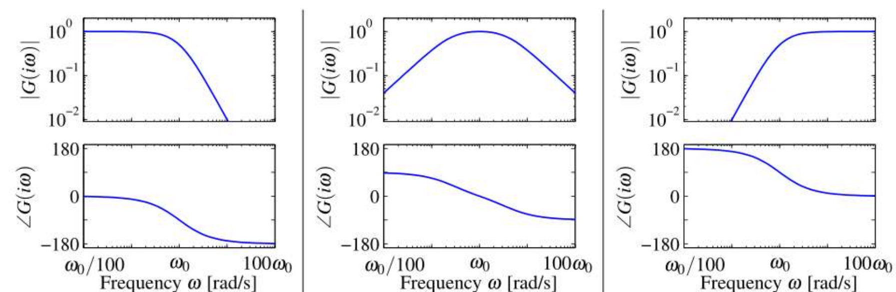
$G(s) = \frac{b(s)}{a(s)}$ — roots of a are termed $\underline{\text{poles}}$,

roots of b are termed zeros
- difference between order of
a & that of b termed relative degree;
G is proper if ≥ 0, strictly proper if > 0

14. Bode plot

[AMv2 Ch 9.6] [Nv7 Ch 10.1]

input $e^{j\omega t}$ yields
output $G(s)e^{j\omega t} = |G(j\omega)| e^{j\omega(t + \angle G(j\omega))}$
it's useful to visualize
gain $|G(j\omega)|$ and phase $\angle G(j\omega)$
as functions of frequency ω:



$G(s) = \dfrac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$ (a) Low-pass filter

$G(s) = \dfrac{2\zeta\omega_0 s}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$ (b) Band-pass filter

$G(s) = \dfrac{s^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$ (c) High-pass filter

Figure 9.17: Bode plots for low-pass, band-pass, and high-pass filters. The upper plots are the gain curves and the lower plots are the phase curves. Each system passes frequencies in a different range and attenuates frequencies outside of that range.

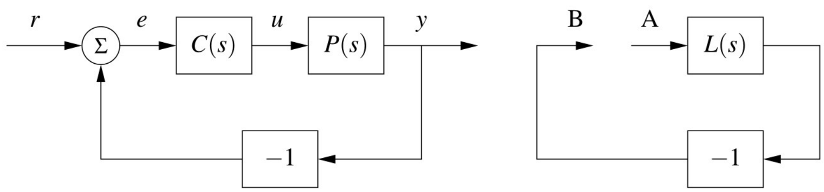
---

week 8: frequency domain

goal: tools for analysis
using transfer functions,
Nyquist / Bode plots

1°. frequency domain analysis

1'. Nyquist stability criterion [AMv2 Ch 10.1, 10.2] [Nv7 Ch 10.3]



(a) Closed loop system (b) Open loop system

Figure 10.1: The loop transfer function. The stability of the feedback system (a) can be determined by tracing signals around the loop. Letting $L = PC$ represent the loop transfer function, we break the loop in (b) and ask whether a signal injected at the point A has the same magnitude and phase when it reaches point B.

<u>loop transfer function</u>

$L(s) = P(s)\,C(s)$

<u>Nyquist stability criterion</u>

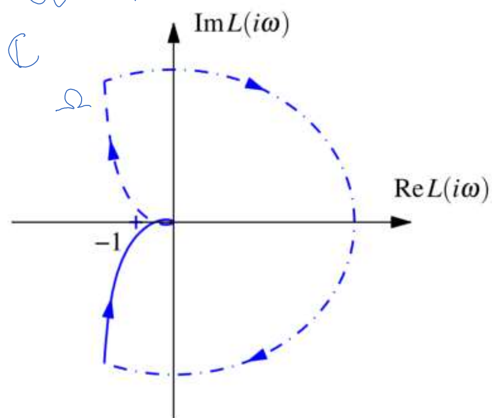
if L has no poles in the RHP,

then: $\frac{L}{1+L} = \frac{PC}{1+PC}$ is stable

$\Longleftrightarrow$ Ω does not encircle $-1 \in \mathbb{C}$

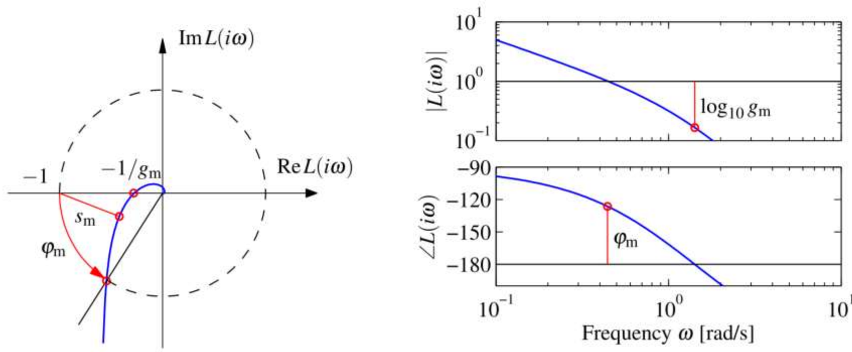
" $\{L(j\omega) : -\infty < \omega < \infty\} \subset \mathbb{C}$

– Nyquist plot:



– since Ω doesn't encircle $-1 \in \mathbb{C}$, the Nyquist stability criterion implies $\frac{L}{1+L}$ is (asymptotically) stable

1°. stability margins [AMv2 Ch 10.3] [Nv7 Ch 10.7]



(a) Nyquist plot (b) Bode plot

Figure 10.11: Stability margins for a third-order loop transfer function $L(s)$. The Nyquist plot (a) shows the stability margin, s_m, the gain margin g_m, and the phase margin φ_m. The stability margin s_m is the shortest distance to the critical point -1. The gain margin corresponds to the smallest increase in gain that creates an encirclement, and the phase margin is the smallest change in phase that creates an encirclement. The Bode plot (b) shows the gain and phase margins.

- stability margin s_m = distance from Ω to $-1 \in \mathbb{C}$
- gain margin $g_m = \begin{pmatrix}\text{distance from } \Omega \text{ to } -1 \in \mathbb{C} \\ \text{restricted to real axis}\end{pmatrix}^{-1}$
- phase margin φ_m = distance from Ω to $-1 \in \mathbb{C}$ restricted to rotation of Ω

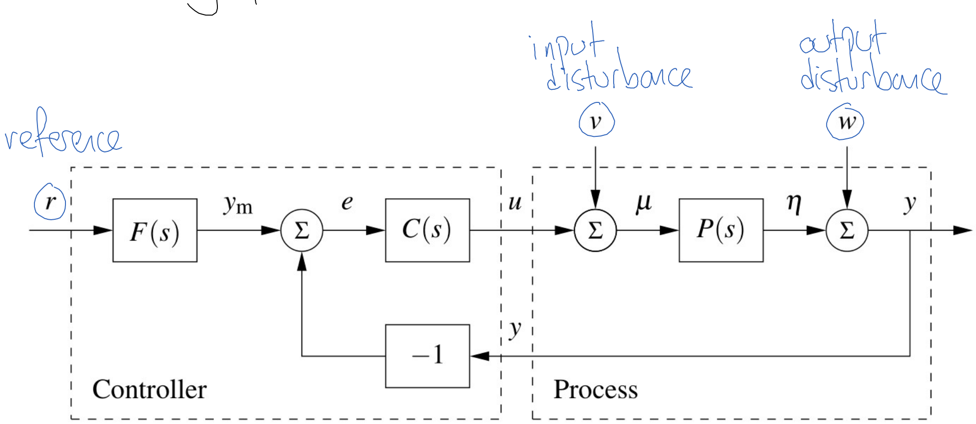
---

week 9: frequency domain

goal: tools for analysis using transfer functions, root locus plots

1°. frequency domain analysis

1'. sensitivity functions [AMv2 ch 12.1, 12.2] [Nv7 not covered]



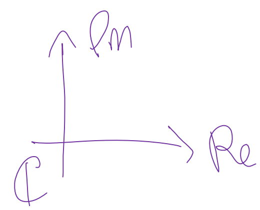
y	u	e	μ	η

y	u	e	μ	η	
$\frac{PCF}{1+PC}$	$\frac{CF}{1+PC}$	$\frac{F}{1+PC}$	$\frac{CF}{1+PC}$	$\frac{PCF}{1+PC}$	r
$\frac{P}{1+PC}$	$\frac{-PC}{1+PC}$	$\frac{-P}{1+PC}$	$\frac{1}{1+PC}$	$\frac{P}{1+PC}$	v
$\frac{1}{1+PC}$	$\frac{-C}{1+PC}$	$\frac{-1}{1+PC}$	$\frac{-C}{1+PC}$	$\frac{-PC}{1+PC}$	w

$S = \frac{1}{1+PC}$ <u>sensitivity</u> $T = \frac{PC}{1+PC}$ <u>complementary sensitivity</u>

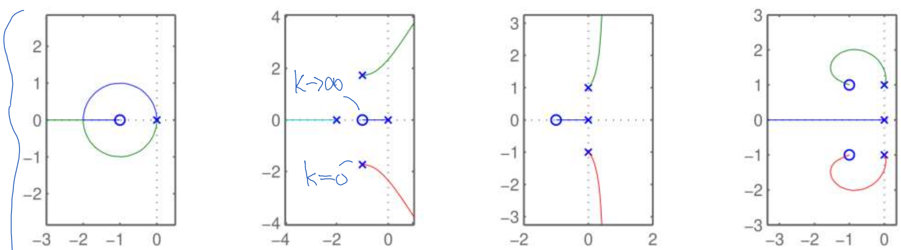
↳ note: $S + T = \frac{1+PC}{1+PC} = 1$, thus name makes sense

1°. root locus [AMv2 Ch 12.5] [Nv7 Ch 9]



$$P_a(s) = k\frac{s+1}{s^2}, \qquad P_b(s) = k\frac{s+1}{s(s+2)(s^2+2s+4)},$$
$$P_c(s) = k\frac{s+1}{s(s^2+1)}, \qquad P_d(s) = k\frac{s^2+2s+2}{s(s^2+1)}. \tag{12.18}$$

start at "x", tend to "o"



(a) $P_a(s)$ (b) $P_b(s)$ (c) $P_c(s)$ (d) $P_d(s)$

"root locus diagram"

Figure 12.18: Examples of root loci for processes with the transfer functions $P_a(s)$, $P_b(s)$, $P_c(s)$, and $P_d(s)$ given by equation (12.18).

→ which of these systems can be stabilized by proportional feedback? (can the gain be arbitrarily large?)

---

week 10: proportional-integral-derivative

goal: techniques for analysis, design, & implementation of the most ubiquitous control architecture

1°. essentials of feedback control [AMv2 Ch 11] [Nv7 Ch 9.4]

1: a simple controller

P – present error

I – (integral of) past error

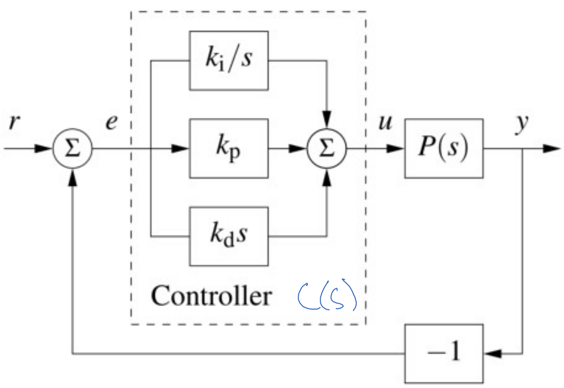
D – (prediction of) future error (using linear extrapolation)

– as a transfer function:

$$C(s) = k_P + \frac{k_I}{s} + k_D s$$

– as a function of time:

$$u(t) = k_P e(t) + k_I \int_0^t e(\tau)\,d\tau + k_D \frac{d}{dt} e(t)$$



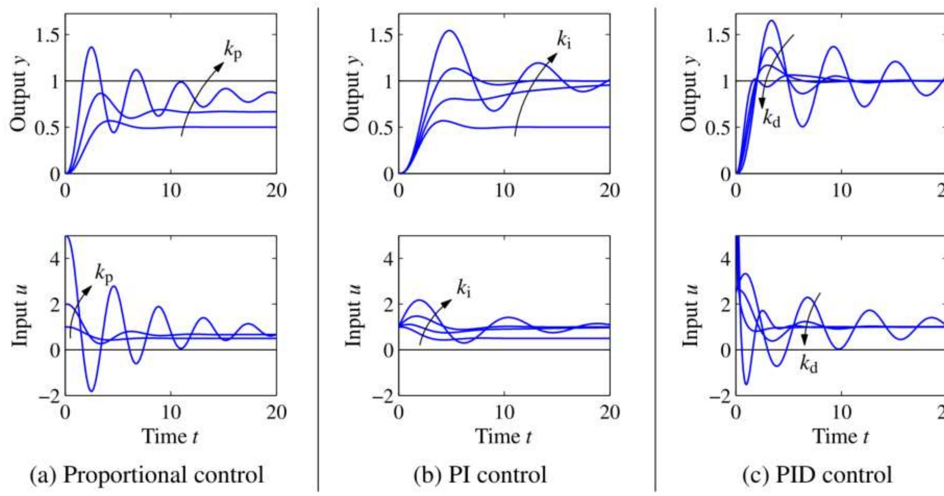


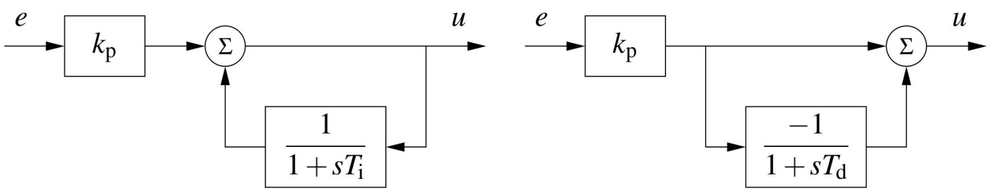
Figure 11.2: Responses to step changes in the reference value for a system with a proportional controller (a), PI controller (b) and PID controller (c). The process has the transfer function $P(s) = 1/(s+1)^3$, the proportional controller has parameters $k_p = 1$, 2 and 5, the PI controller has parameters $k_p = 1$, $k_i = 0$, 0.2, 0.5, and 1, and the PID controller has parameters $k_p = 2.5$, $k_i = 1.5$ and $k_d = 0$, 1, 2, and 4.

1^2: implementation issues

$u(t) = k_P e(t)$ proportional

$+ k_I \int_0^t e(\tau)\,d\tau$ integral ⟶ unbounded ("windup")

$+ k_D \frac{d}{dt} e(t)$ derivative ⟶ amplifies noise



(a) Integral action (automatic reset)

(b) Derivative action

Figure 11.3: Implementation of integral and derivative action. The block diagram in (a) shows how integral action is implemented using *positive feedback* with a first-order system, sometimes called automatic reset. The block diagram in (b) shows how derivative action can be implemented by taking differences between a static system and a first-order system.

→ compute transfer function G_{ue} for (a), (b)