CDS 101/110: Lecture 8.2
PID Control

November 16, 2016

Goals:

- Nyquist Example
- Introduce and review PID control.
- Show how to use “loop shaping” using PID to achieve a performance specification
- Discuss the use of integral feedback and anti-windup compensation

Reading:

- Åström and Murray, Feedback Systems 2-3, Sections 11.1 – 11.3

Nyquist Example (unstable system)

Consider $L(s) = P(s)C(s) = \frac{k}{s(s-1)}$

- Pole at the origin, and unstable pole at $s = -1$
- **Q:** Does unity gain negative feedback stabilize this system?
- **Q:** Does closed loop stability depend upon gain, k ?

Analysis of Closed Loop Poles

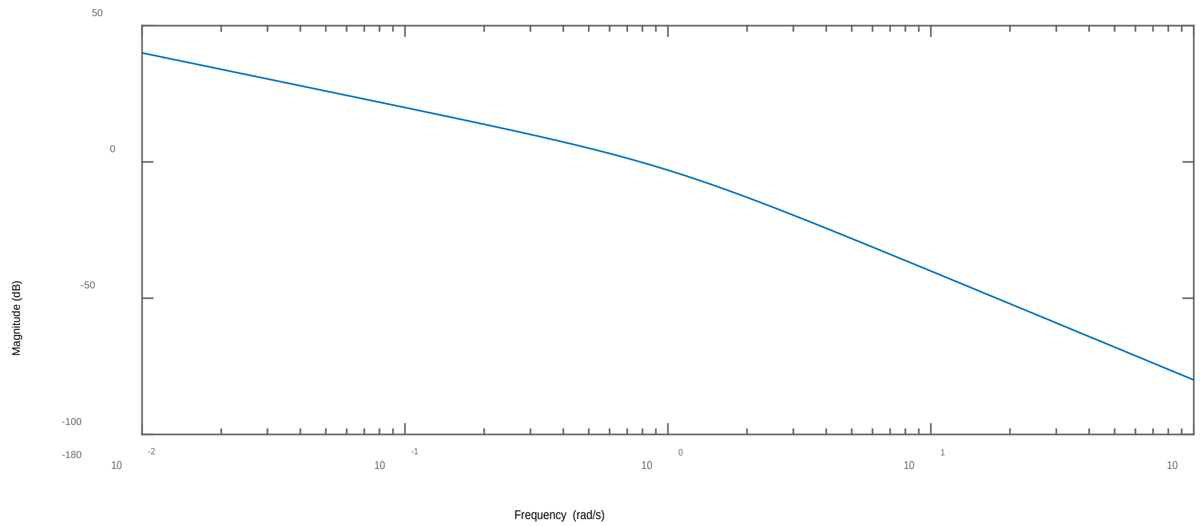
- $G_{yr}(s) = \frac{k}{s^2-s+k} \quad \Rightarrow \quad$ characteristic equation roots: $s = \frac{1}{2} \pm \frac{1}{2}\sqrt{1-4k}$
- Closed loop system is ***always*** unstable for any k

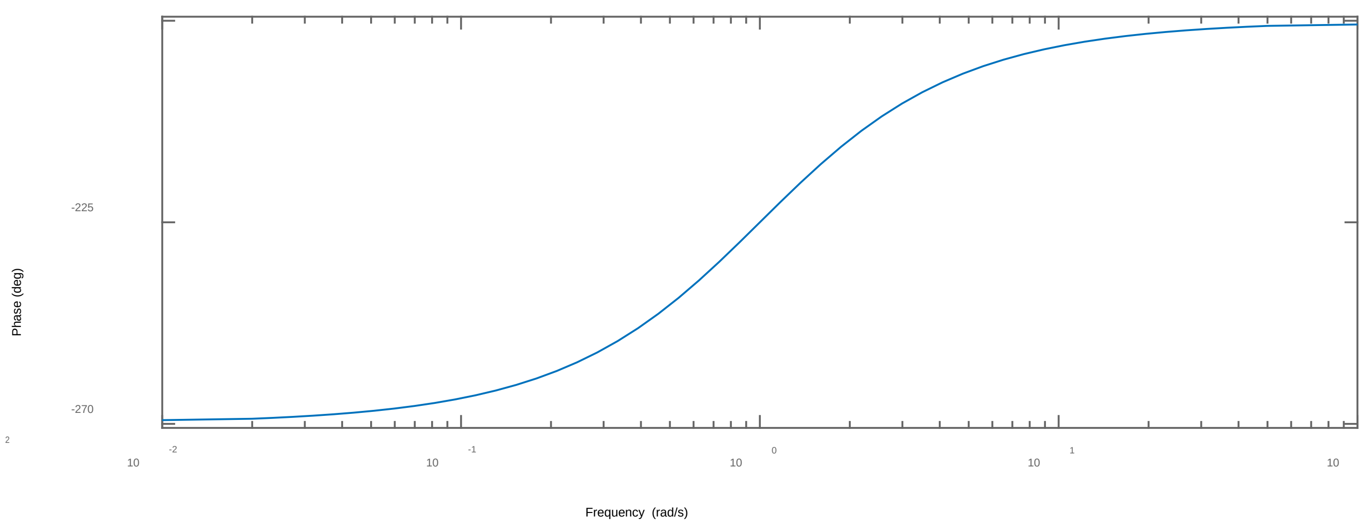
Nyquist Plot Analysis

- **Aside:** magnitude and phase (bode plot) of unstable pole:
  - Let $H(s) = \frac{1}{(s-a)}$. $H(i\omega) = \frac{1}{i\omega-a}\frac{-i\omega-a}{(-i\omega-a)} = \frac{-i\omega-a}{\omega^2+a^2}$
  - Magnitude: $|G(i\omega)| = \frac{\sqrt{\omega^2+a^2}}{\omega^2+a^2} = \frac{1}{\sqrt{\omega^2+a^2}}$
  - Phase: $\arg\left(G(i\omega)\right) = arctan\left(\frac{-\omega}{-a}\right) = \pm 180^o + \arctan\left(\frac{\omega}{a}\right)$

Nyquist Example (unstable system)

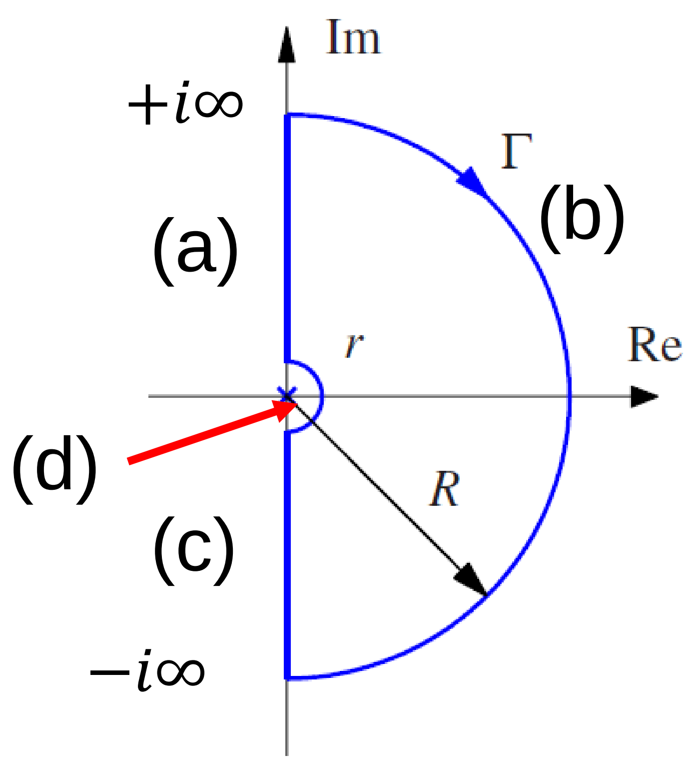
Bode Plots of Open Loop $L(s) = P(s)C(s) = \frac{k}{s(s-1)}$





Nyquist Contour and Plot

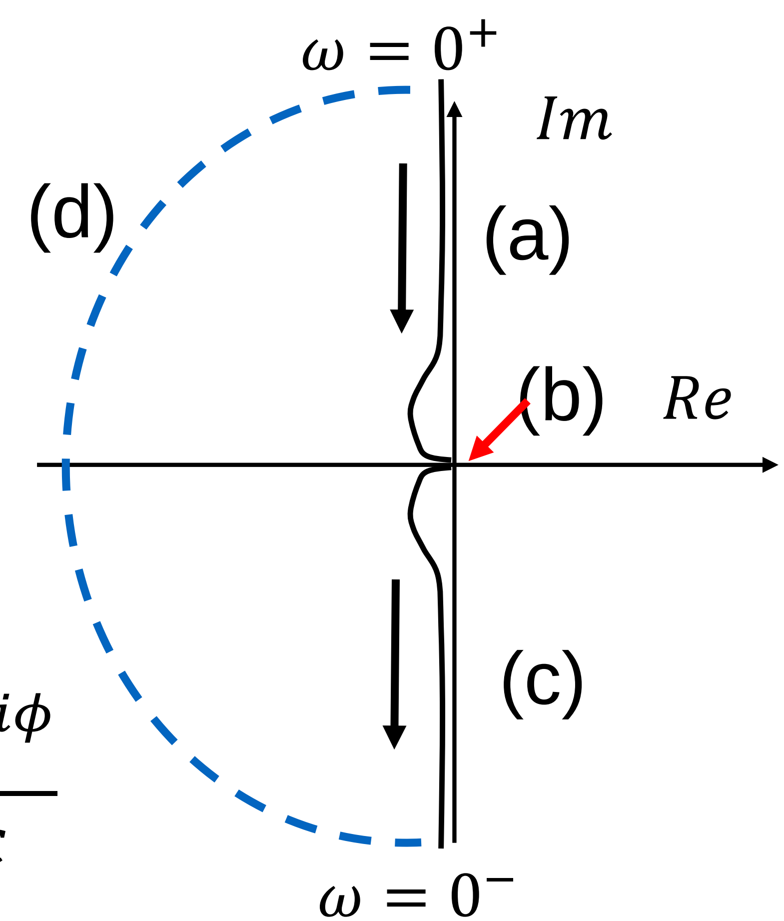
- Must account for pole on the $i\omega$ axis



$$a)\ \omega = 0^+ \rightarrow +\infty$$
$$b)\ \omega = +\infty \rightarrow -\infty$$
$$c)\ \omega = -\infty \rightarrow \omega = 0^-$$
$$d)\ \omega = 0^- \rightarrow \omega = 0^+$$

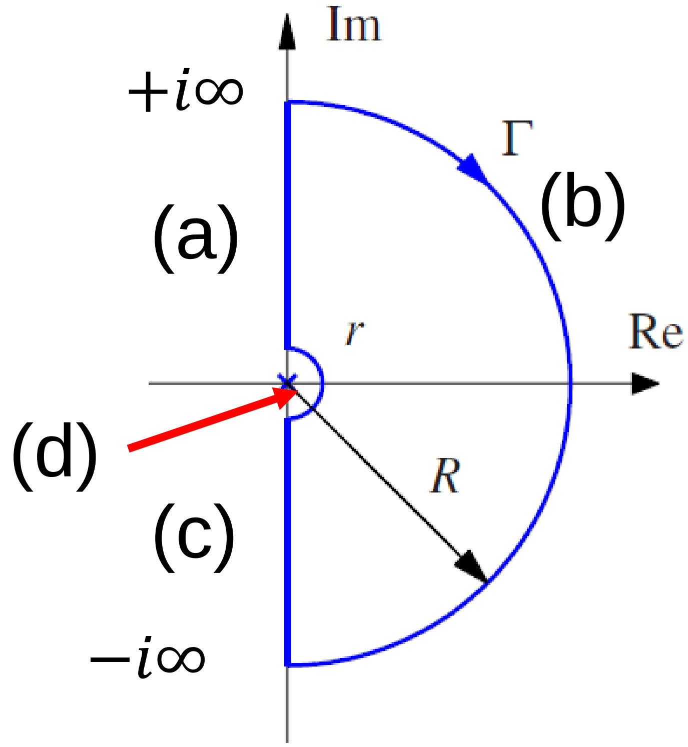
$\omega = \varepsilon e^{i\phi}$ for $[-90^o, 90^o]$

$$G\left(s = \varepsilon e^{i\phi}\right) \approx \frac{k}{-\varepsilon e^{i\phi}} = \frac{k e^{-i\phi}}{-\varepsilon}$$



Nyquist Example (unstable system)

Nyquist Contour and Plot

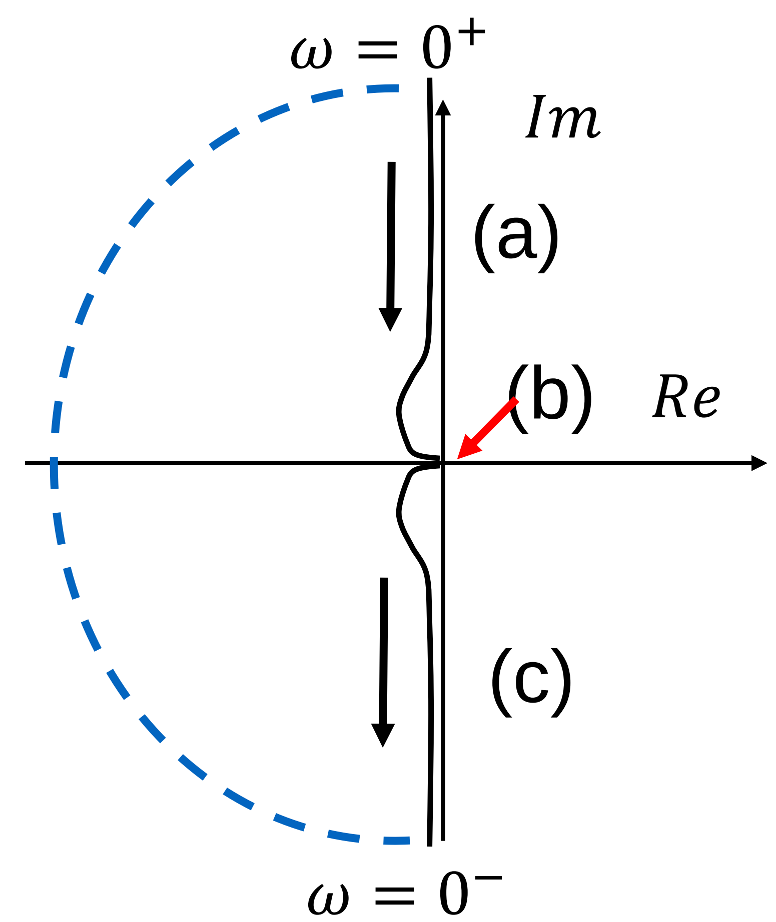


$(d)\ \omega = 0^- \rightarrow \omega = 0^+$

$\omega = \varepsilon e^{i\phi}$ for $[-90^o, 90^o]$

$$G\left(s = \varepsilon e^{i\phi}\right) \approx \frac{k}{-\varepsilon e^{i\phi}}$$

$$= \frac{ke^{-i\phi}}{-\varepsilon} = \frac{k}{\varepsilon}(-\cos\phi + i\sin\phi)$$

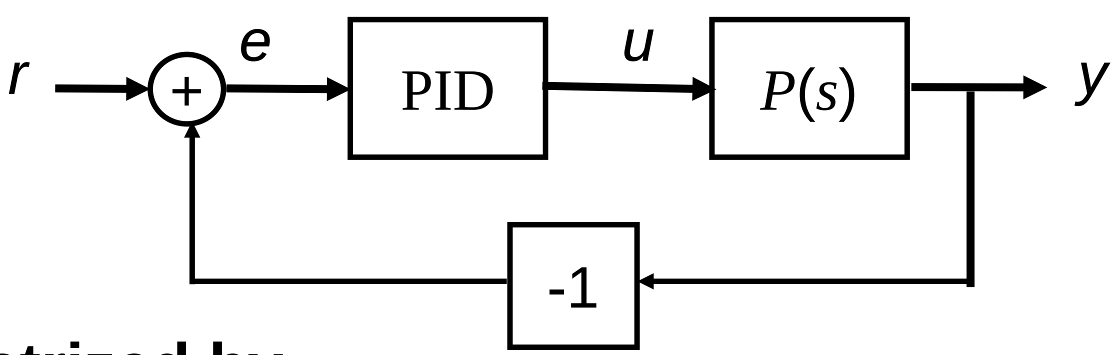


Accounting:

- One open loop pole in RHP: $P = 1$
- One clockwise encirclement of -1 point: $N = 1$
- $Z = N + P = 1 + 1 = 2 \quad \Rightarrow$ two unstable poles in closed loop system

Homework: show that $\frac{k_1(1+k_2 s)}{s(s-1)}$ is stable for $k_1 k_2 > 1$

Overview: PID control



$$u = k_p e + k_i \int_0^t e(\tau)\,d\tau + k_d \frac{de}{dt}$$
$$= k_p \left(e + \frac{1}{T_i} \int_0^t e(\tau)\,d\tau + T_d \frac{de}{dt} \right)$$

Parametrized by:

- k_p, the "proportional gain"
- k_i, the "integral gain"
- k_d, the "derivative gain"

Alternatively:

$$k_p, \quad T_i = \frac{k_p}{k_i}, \quad T_d = \frac{k_d}{k_p}$$

Intuition

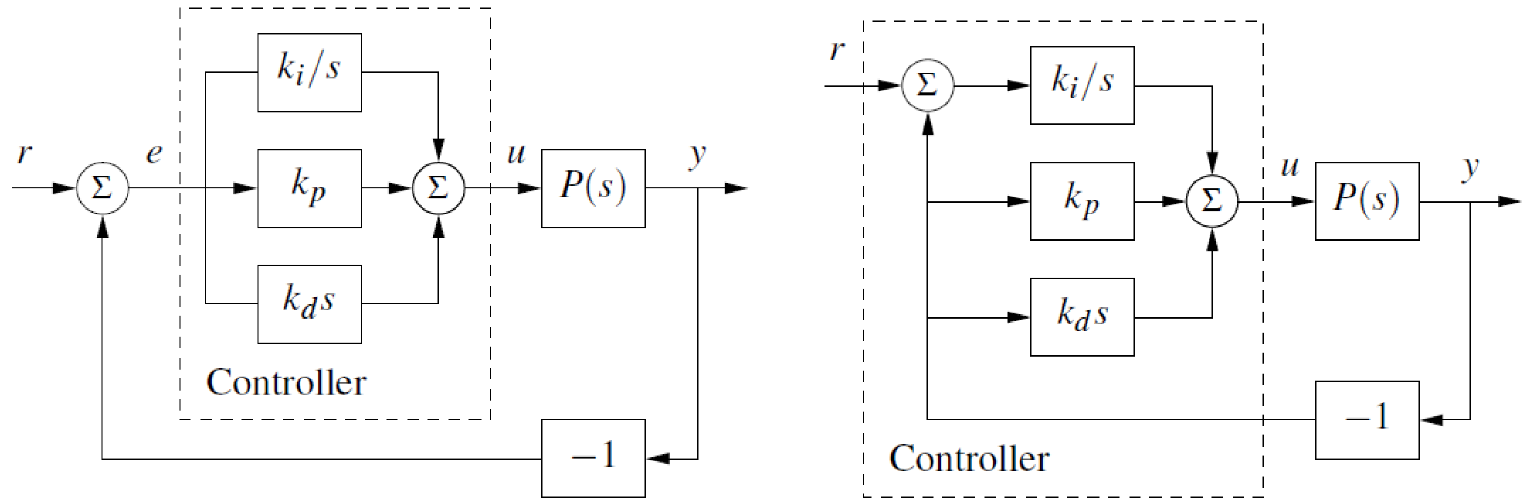
- Proportional term: provides inputs that correct for "current" errors
- Integral term: insures steady state error goes to zero
- Derivative term: provides "anticipation" of upcoming changes (also provides "damping")
- Controller specified in time domain, but can be analyzed in frequency domain

A bit of history on "three term control"

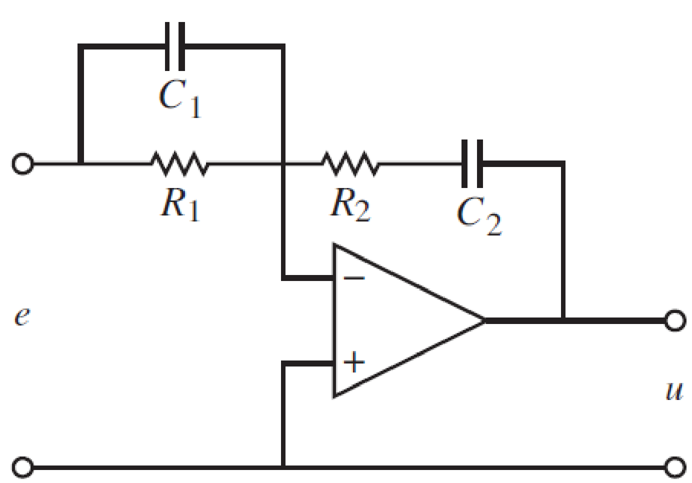
- First appeared in 1922 paper by Minorsky: "*Directional stability of automatically steered bodies*" under the name "three term control"

Utility of PID

- PID control is most common feedback structure in engineering systems
- For many systems, only need PI or PD (special case)
- Many tools for tuning PID loops and designing gains



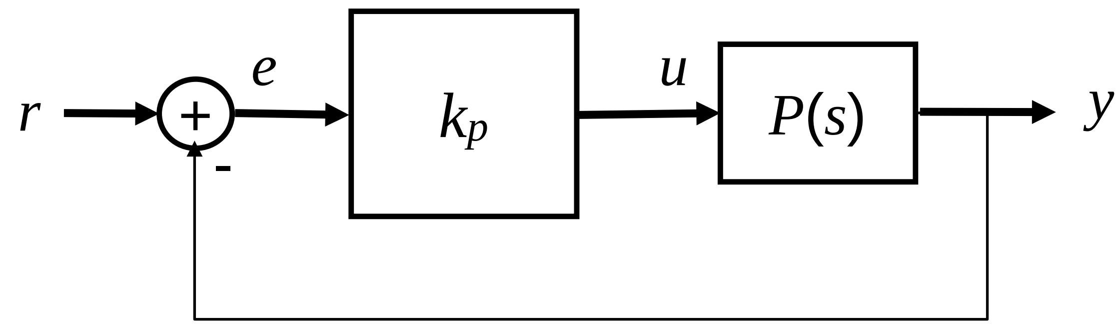
(a) PID using error feedback (b) PID using two degrees of freedom



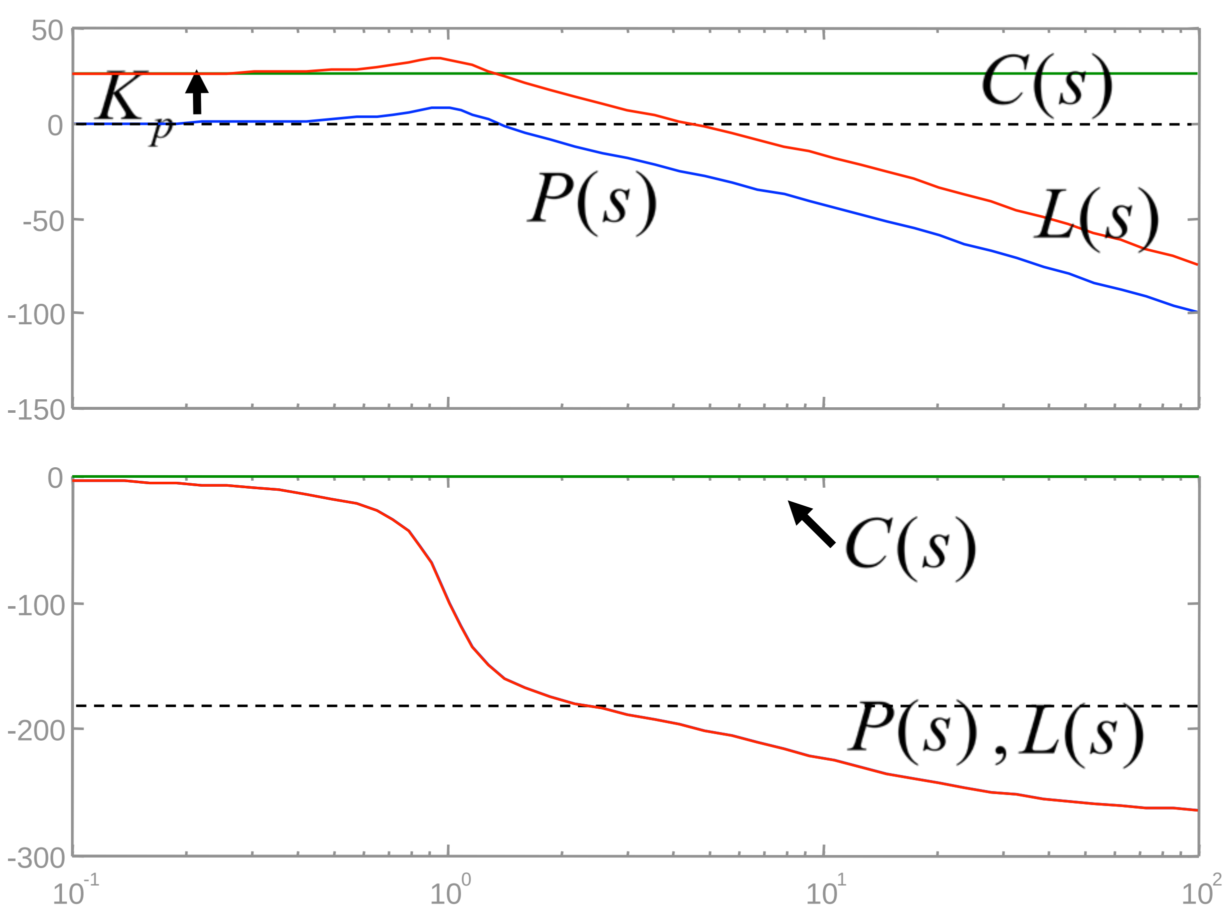
Proportional Feedback

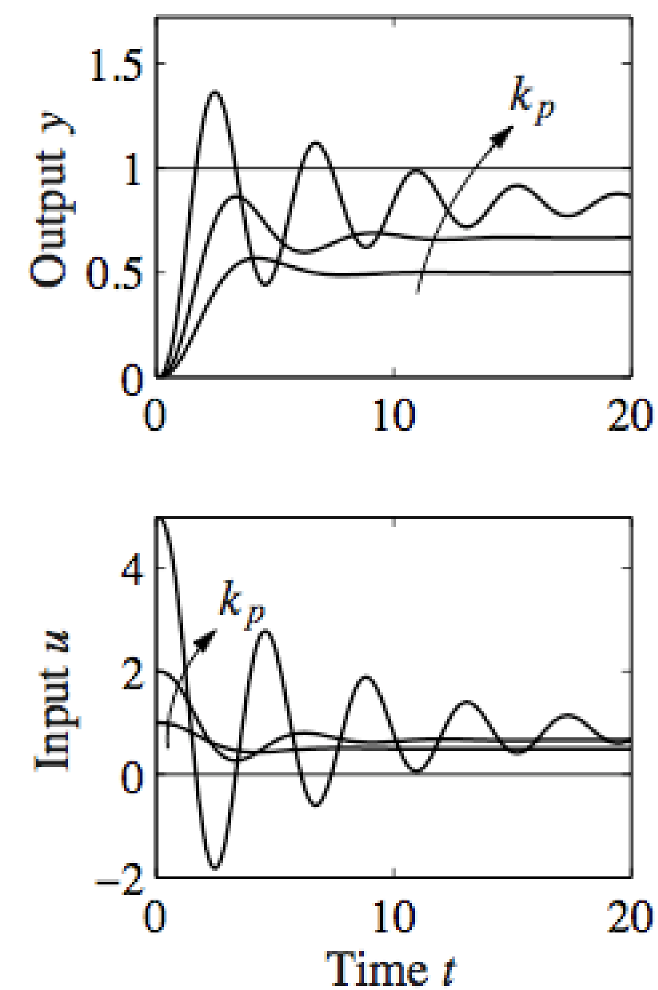
Simplest controller choice: $u = k_p e$

- Effect: lifts gain with no change in phase
- Good for plants with low phase up to desired bandwidth
- Bode: shift gain up by factor of kp
- Step response: better steady state error, but with decreasing stability



$k_p > 0$

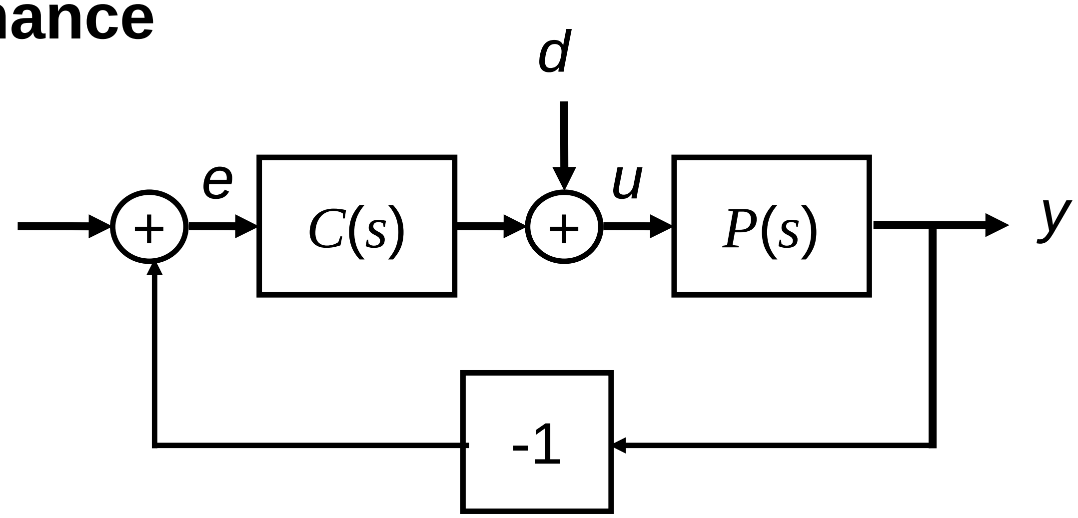




Steady State error removed by feedforward: $u = k_p e + u_{ff}$

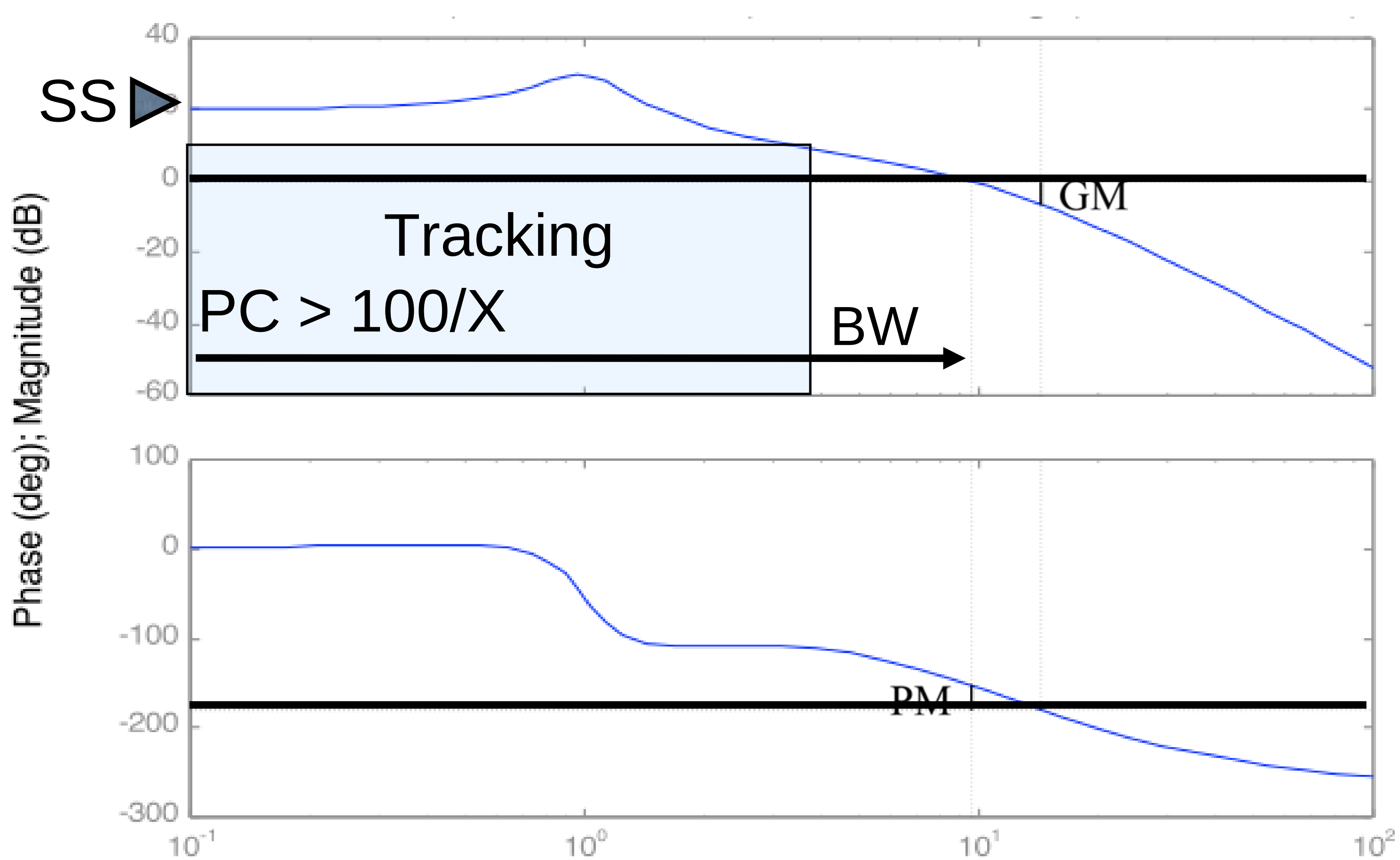
Frequency Domain Performance Specifications

Specify bounds on the loop transfer function to guarantee desired performance



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

$$H_{er} = \frac{1}{1+L} \qquad H_{yr} = \frac{L}{1+L}$$



- Steady state error:

$$H_{er}(0) = 1/\big(1 + L(0)\big) \approx 1/L(0)$$

⇒ zero frequency (“DC”) gain

- Bandwidth: assuming ~90° phase margin

$$\frac{L}{1+L}(j\omega_c) \approx \left|\frac{1}{1+j}\right| = \frac{1}{\sqrt{2}}$$

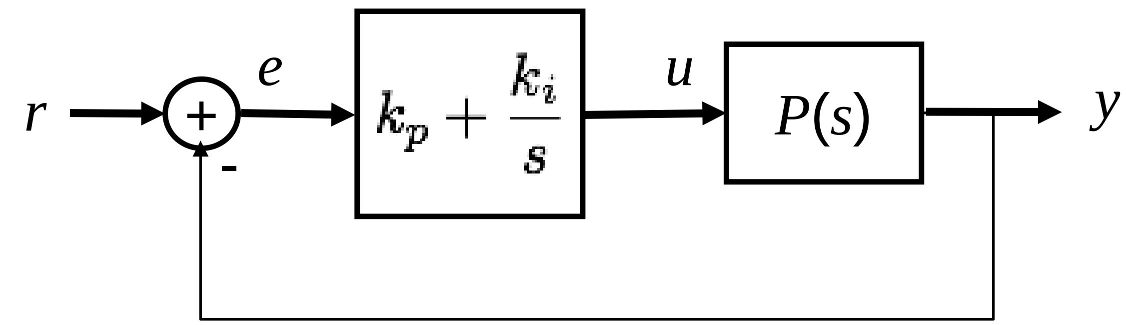
⇒ sets crossover freq

- Tracking: X% error up to frequency ω_t ⇒ determines gain bound ($1 + PC > 100/X$)

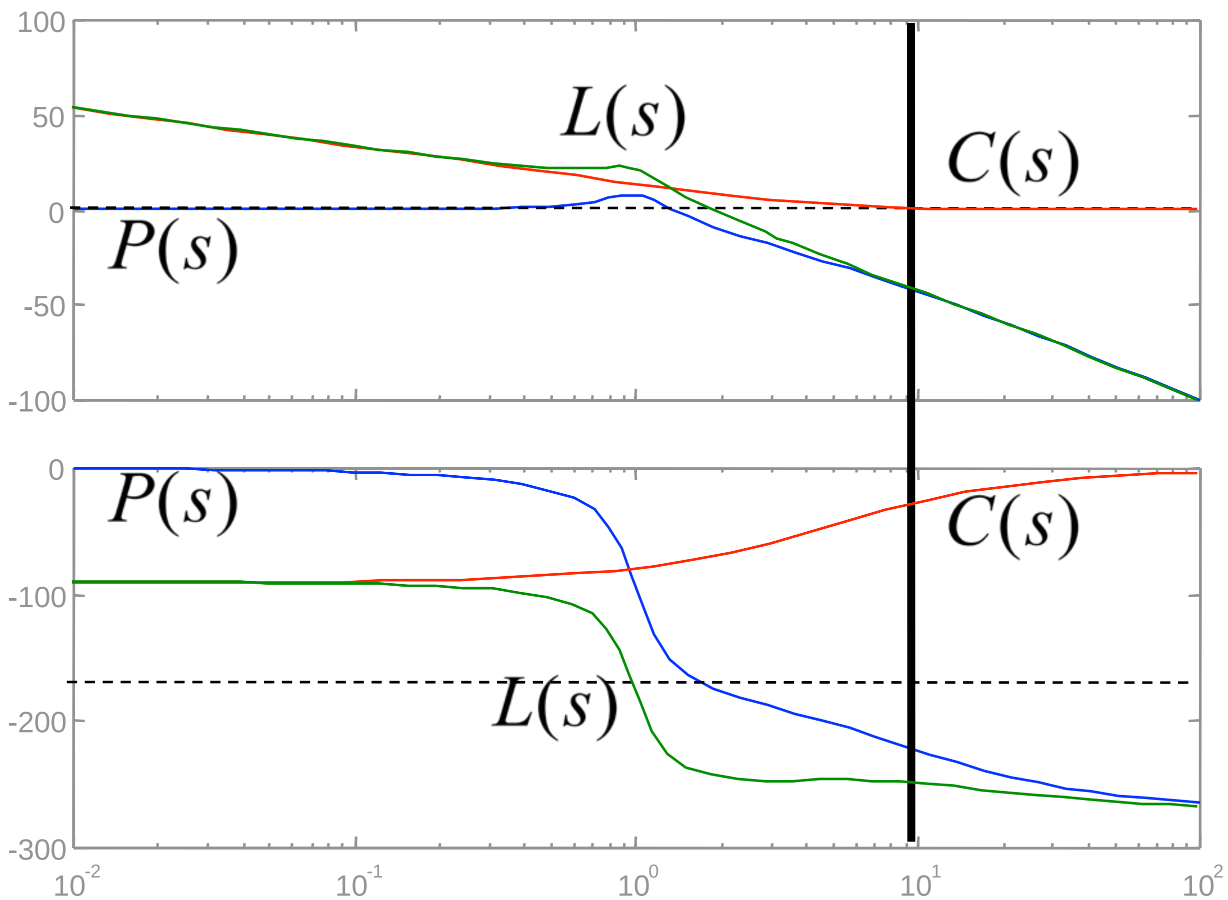
Proportional + Integral Compensation

Use to eliminate steady state error

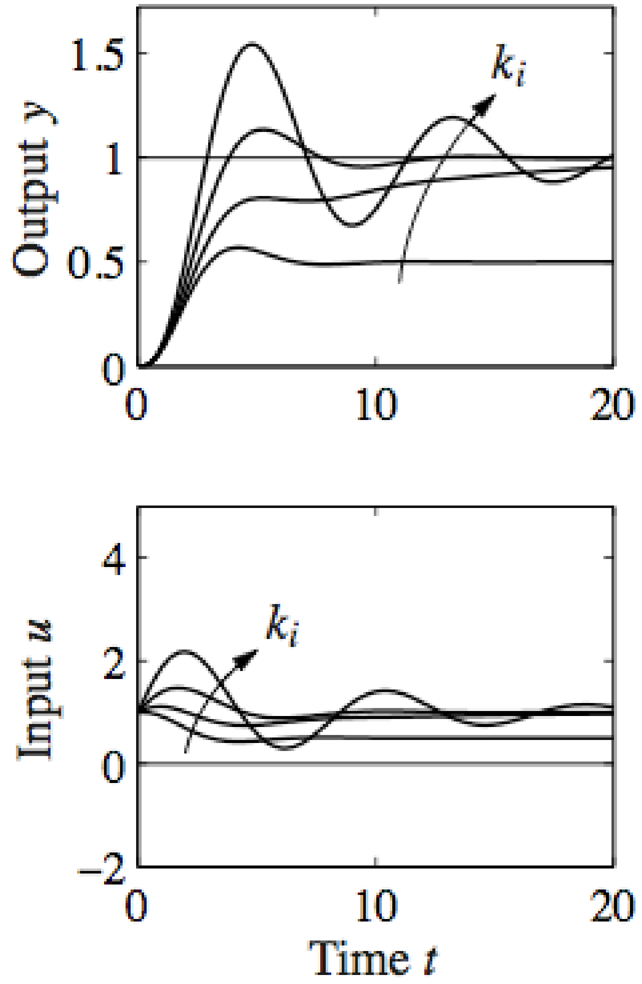
- Effect: lifts gain at low frequency
- Gives zero steady state error
- Handles modeling error
- Bode: infinite SS gain + phase lag
- Step response: zero steady state error, with smaller settling time, but more overshoot



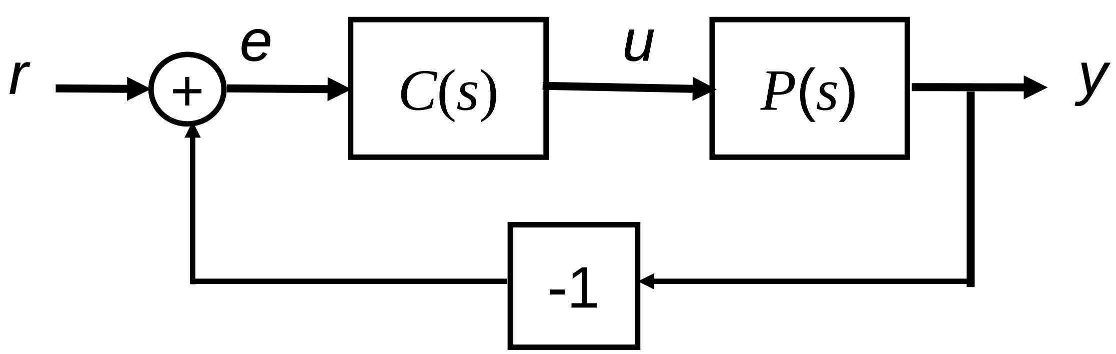
$$k_p > 0, \quad k_i > 0$$



$\omega_z = K_I / K_p$

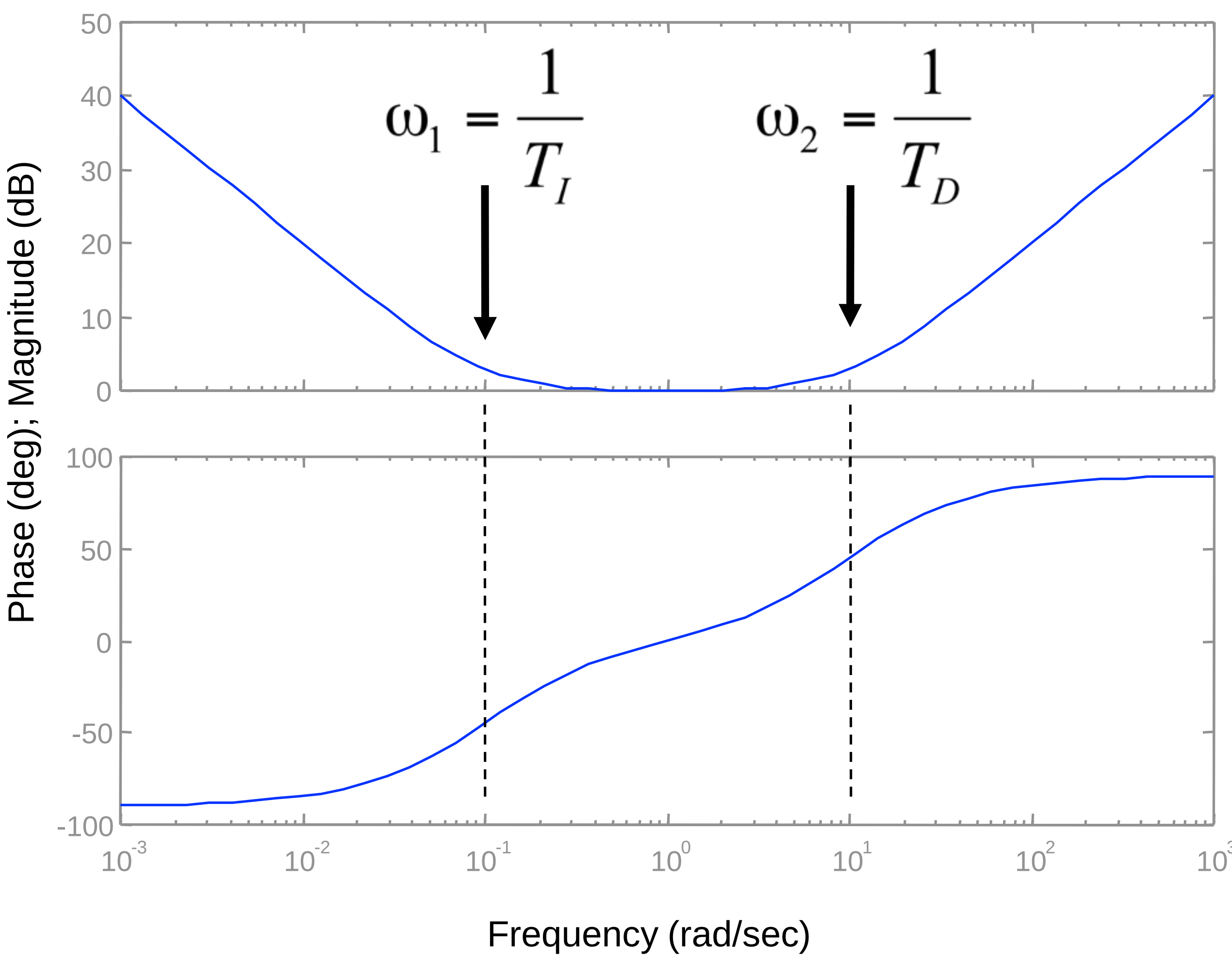


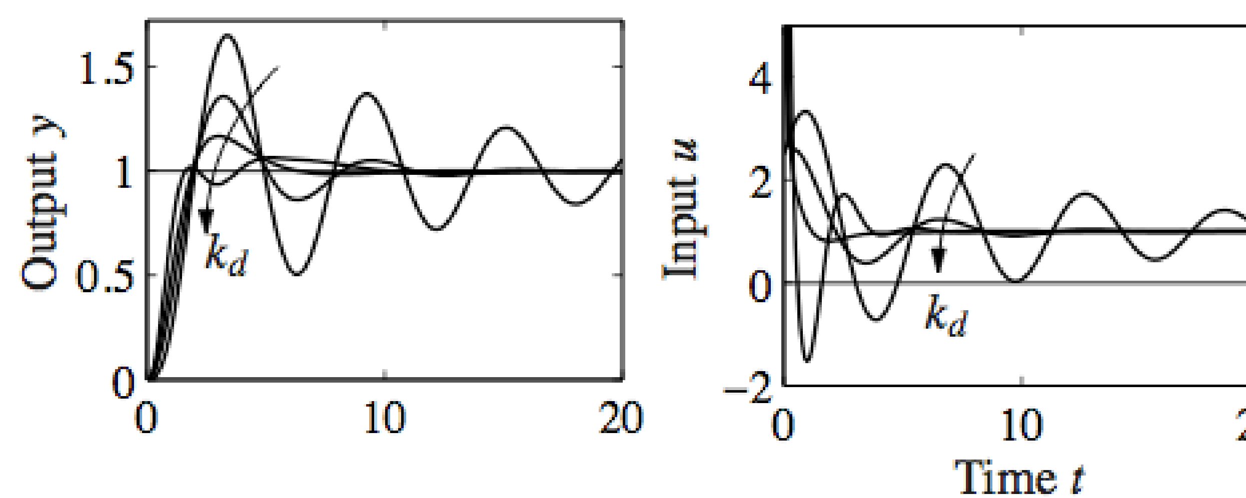
Proportional + Integral + Derivative (PID)



$$C(s) = k_p + k_i \frac{1}{s} + k_d s$$
$$= k(1 + \frac{1}{T_i s} + T_d s)$$
$$= \frac{kT_d}{T_i} \frac{(s + 1/T_i)(s + 1/T_d)}{s}$$

Bode Diagrams





Derivative Action:

- $u = k_p e + k_d \dot{e} = k_p \left(e + T_d \frac{de}{dt} \right) = k_p e_P$
- e_p is 1st –order (linearized) prediction error at time $t + T_d$
- T_d is the *derivative time constant*

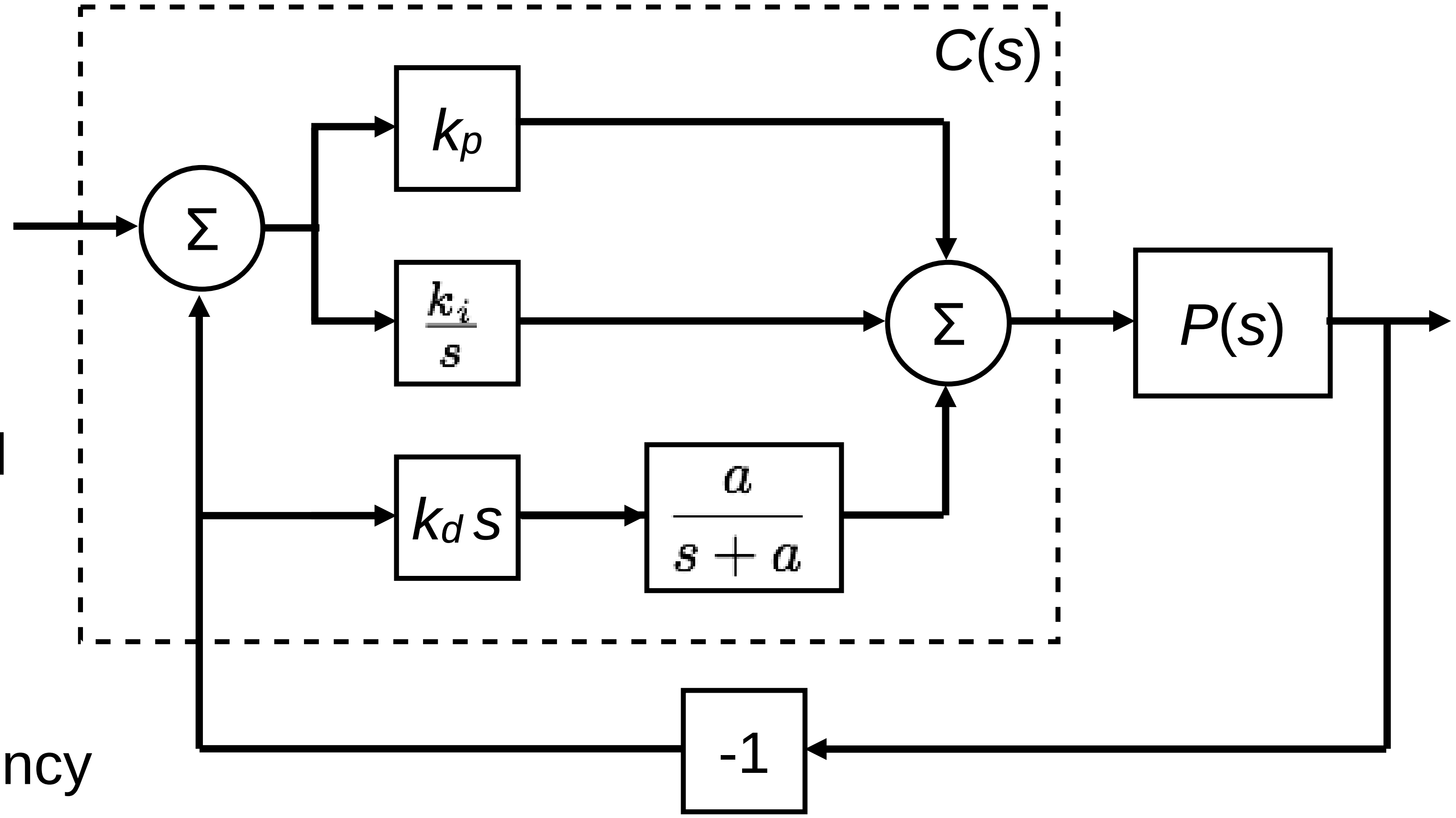
Implementing Derivative Action

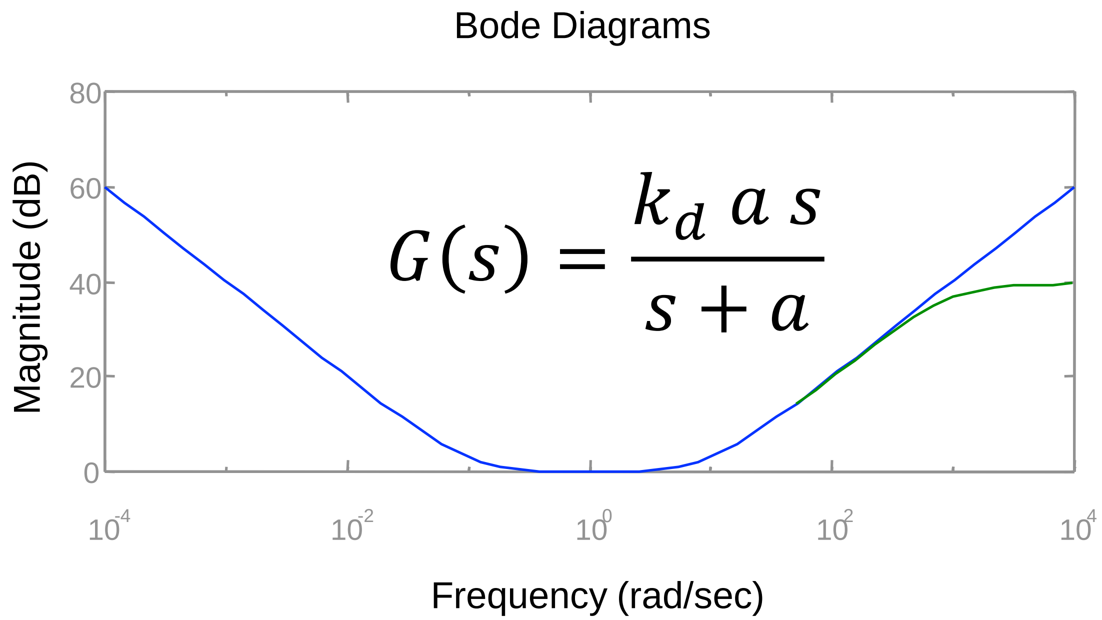
Problems with derivatives

- High frequency noise amplified by derivative term
- Step inputs in reference can cause large inputs

Solution: modified PID control

- Use high frequency ***rolloff*** in derivative term
  - 1st-order filter gives finite gain at high frequency
  - use higher order filter if needed
- Don't feed reference signal through derivative block
  - Useful when reference has unwanted high frequency content
  - Better solution: reference shaping via two DOF design (F(s) block)
- Many other variations (see text + refs)





$$G(s) = \frac{k_d\, a\, s}{s + a}$$

Choosing PID gains ("tuning")

First order system: $P(s) = \frac{b}{s+a}$

- PI controller has char. poly: $s^2 + (a + bk_p)s + bk_i$
- Closed loop poles can be set arbitrarily:

Second Order System: $P(s) = \frac{\omega_0^2}{s^2+2\zeta\omega_0 s+\omega_0^2}$

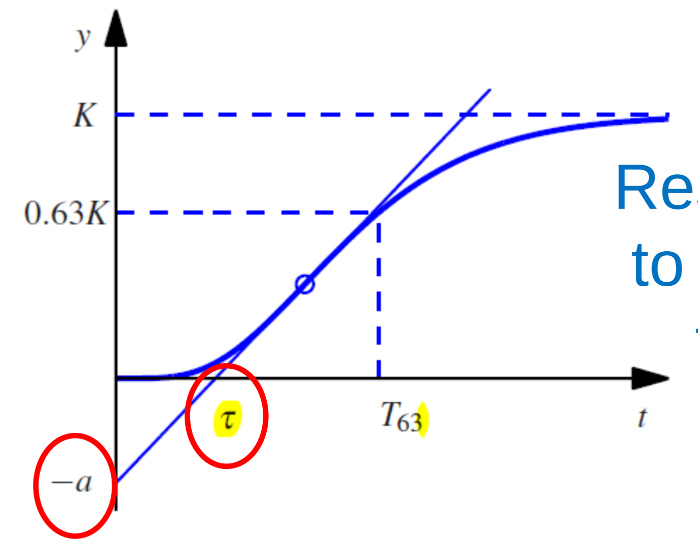
- PID controller allows closed loop poles to be set arbitrarily

Higher Order Systems:

- Use PID to controller a "reduced order" (simplified system)
- Use PID "knobs" to set performance for "dominant" modes

Zeigler-Nichols step response method

- Design PID gains based on step response
- Measure maximum slope + intercept
- Works OK for many plants (but underdamped)
- Maybe useful way to get a first cut controller, especially for higher order, or unknown order

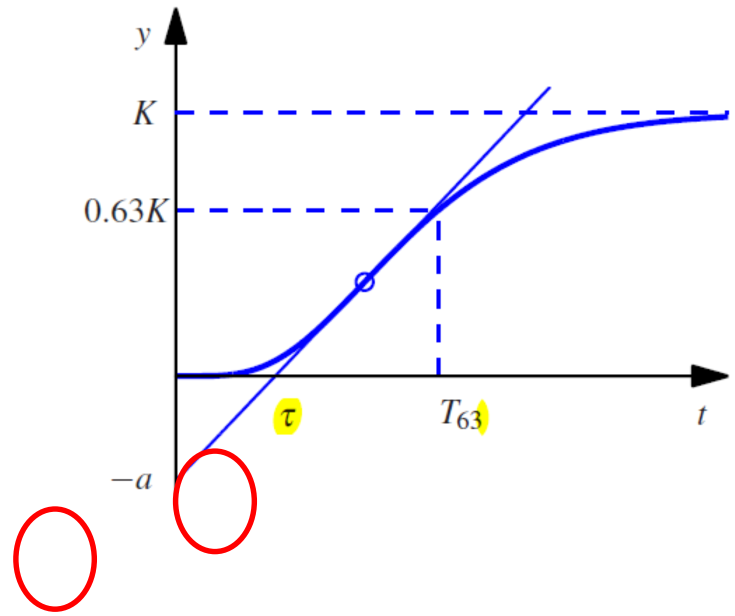


Response to "bump test"

PID "Tuning"

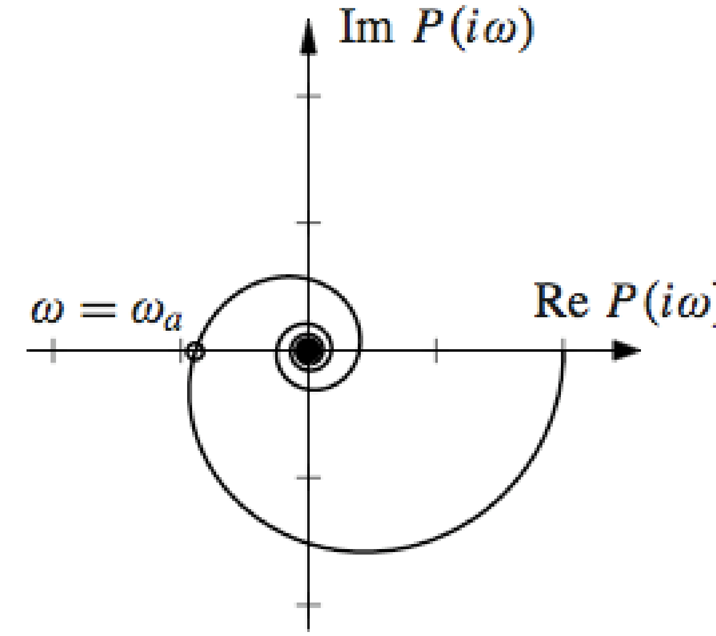
Type	k_p	T_i	T_d
P	1/a		
PI	0.9/a	3τ	
PID	1.2/a	2τ	0.5τ

(a) Step response method



Ziegler-Nichols frequency response method

- Increase proportional gain (with zero derivative and integral gain) until system goes unstable $\rightarrow k_c$
- Use critical gain and frequency as parameters
- Based on Nyquist plot



Variations

- Modified formulas (see text) give better response
- Relay feedback: provides automated way to obtain critical gain, frequency

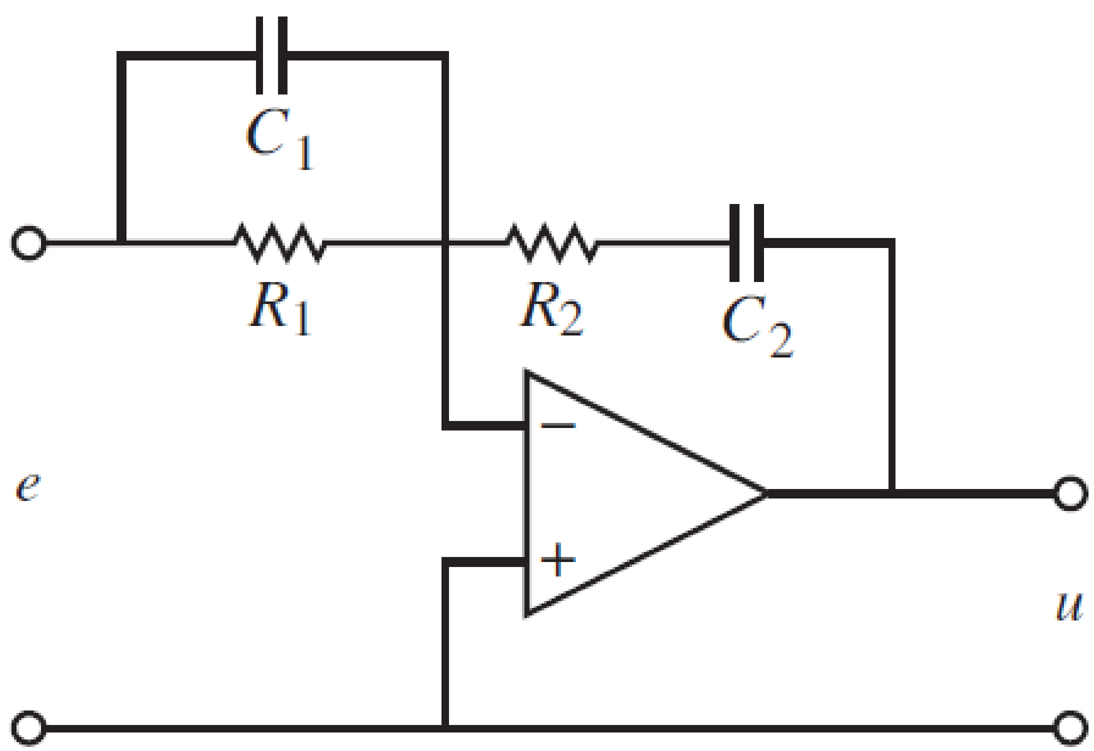
Type	k_p	T_i	T_d
P	$0.5k_c$		
PI	$0.4k_c$	$0.8T_c$	
PID	$0.6k_c$	$0.5T_c$	$0.125T_c$

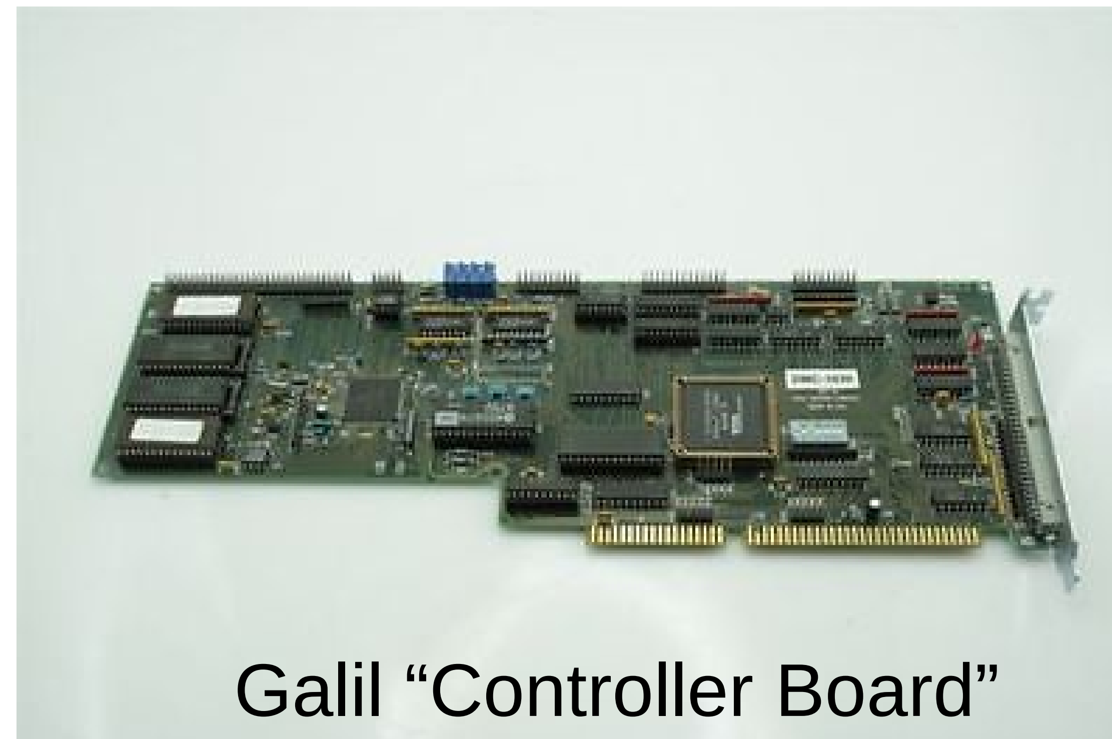
(b) Frequency response method

$$k_p = \frac{0.15\tau + 0.35T}{K\tau} \quad \left(\frac{0.9T}{K\tau}\right), \qquad k_i = \frac{0.46\tau + 0.02T}{K\tau^2} \quad \left(\frac{0.3T}{K\tau^2}\right),$$

$$k_p = 0.22k_c - \frac{0.07}{K} \quad \left(0.4k_c\right), \qquad k_i = \frac{0.16k_c}{T_c} + \frac{0.62}{KT_c} \quad \left(\frac{0.5k_c}{T_c}\right).$$

PID Controllers are easy to implement





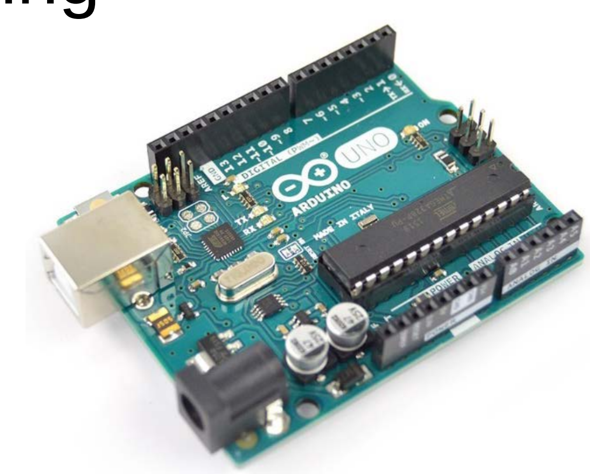
Galil "Controller Board"

$$u = -\frac{Z_2}{Z_1} e = -\frac{R_2}{R_1} \frac{(1 + R_1 C_1 s)(1 + R_2 C_2 s)}{R_2 C_2 s} e.$$

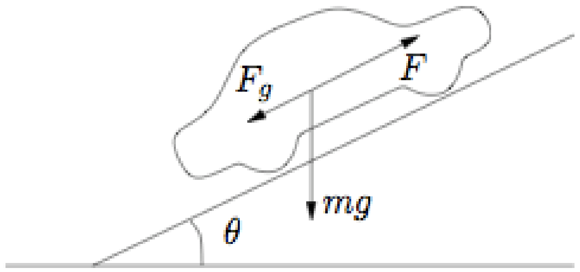
$$k_p = \frac{R_1 C_1 + R_2 C_2}{R_1 C_2}, \qquad T_i = R_1 C_1 + R_2 C_2, \qquad T_d = \frac{R_1 R_2 C_1 C_2}{R_1 C_1 + R_2 C_2}.$$

Built in Discrete Time PID

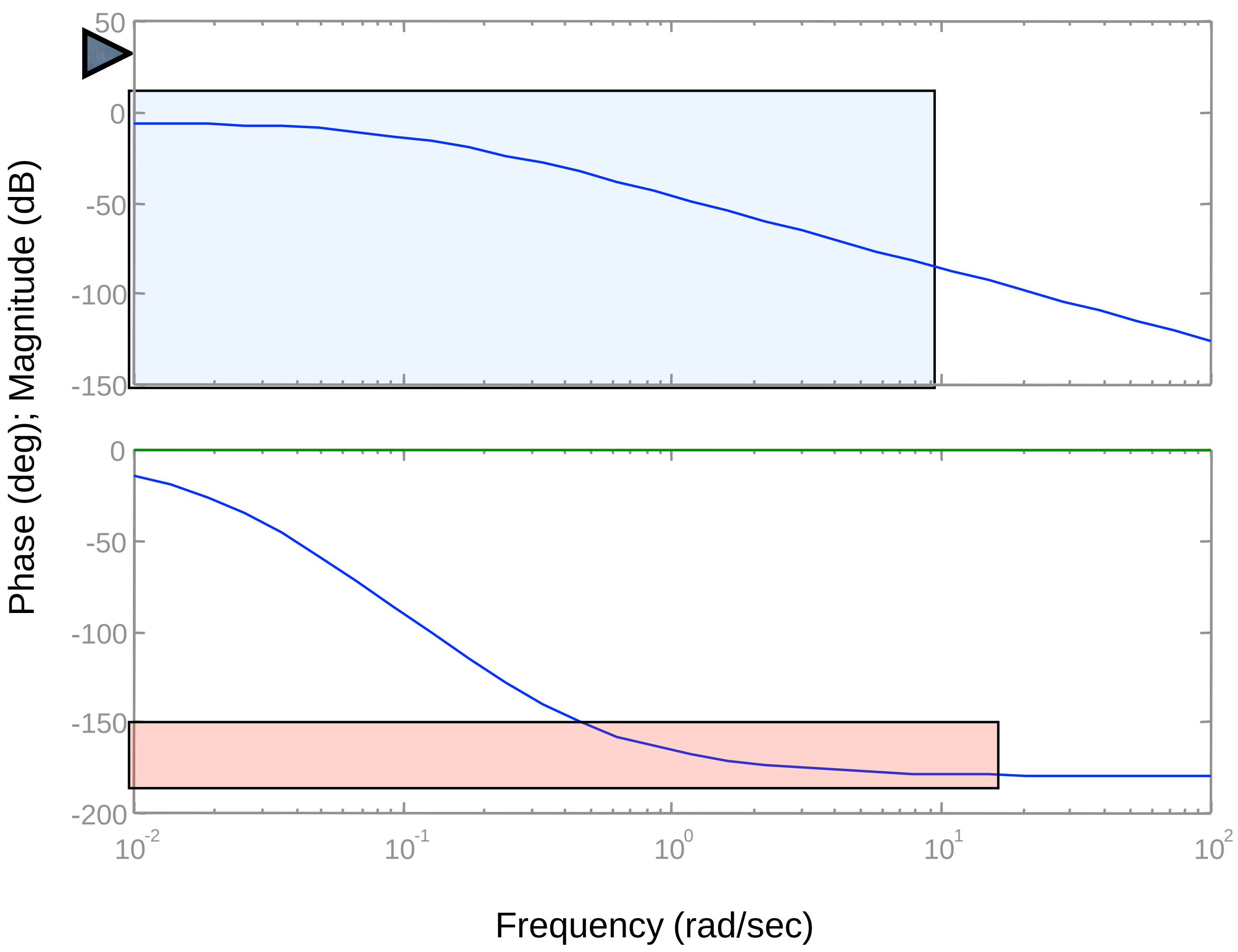
Built in "gain tuning" procedures



Example: Cruise Control using PID - Specification



$$P(s) = \frac{1/m}{s + b/m} \times \frac{r}{s+a}$$



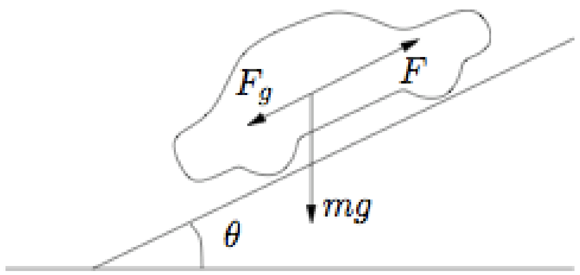
Performance Specification

- ≤ 1% steady state error
  - Zero frequency gain > 100
- ≤ 10% tracking error up to 10 rad/sec
  - Gain > 10 from 0-10 rad/sec
- ≥ 45˚ phase margin
  - Gives good relative stability
  - Provides robustness to uncertainty

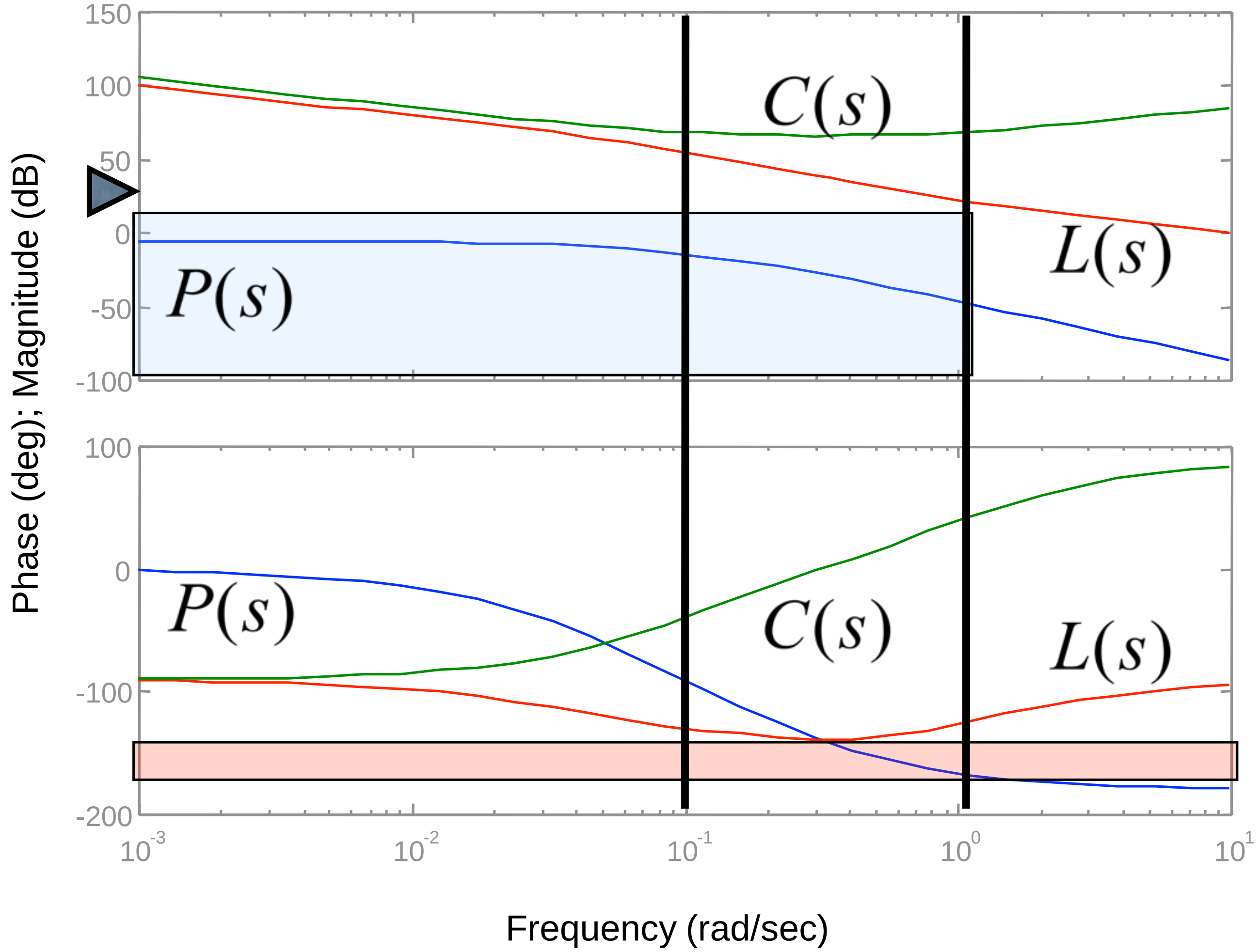
Observations

- Purely proportional gain won't work: to get gain above desired level will not leave adequate phase margin
- Need to increase the phase from ~0.5 to 2 rad/sec and increase gain as well

Example: Cruise Control using PID - Design



$$P(s) = \frac{1/m}{s + b/m} \times \frac{r}{s+a}$$



Approach

- Use integral gain to make steady state error small (zero, in fact)
- Use derivative action to increase phase lead in the cross over region
- Use proportional gain to give desired bandwidth

Controller

- Ti = 1/0.1; Td = 1/1; k = 2000

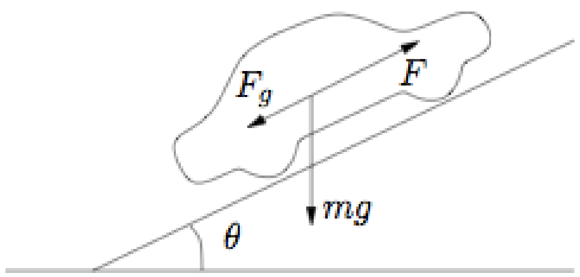
$$C(s) = 2000\frac{s^2 + 1.1s + 0.1}{s}$$

$$= 2200 + \frac{200}{s} + 2000s$$

Closed loop system

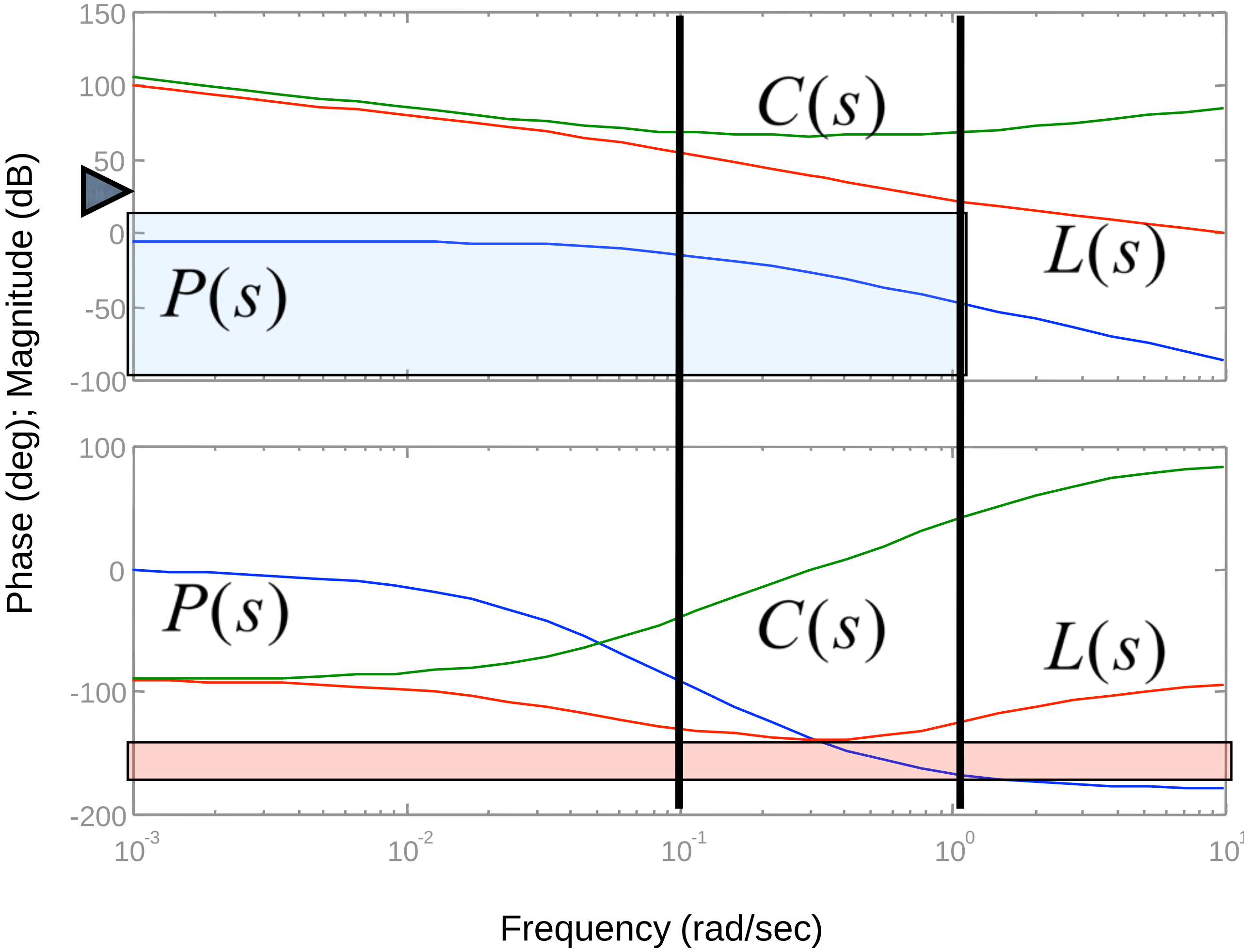
- Very high steady state gain
- Adequate tracking @ 1 rad/sec
- ~80° phase margin
- Verify with Nyquist

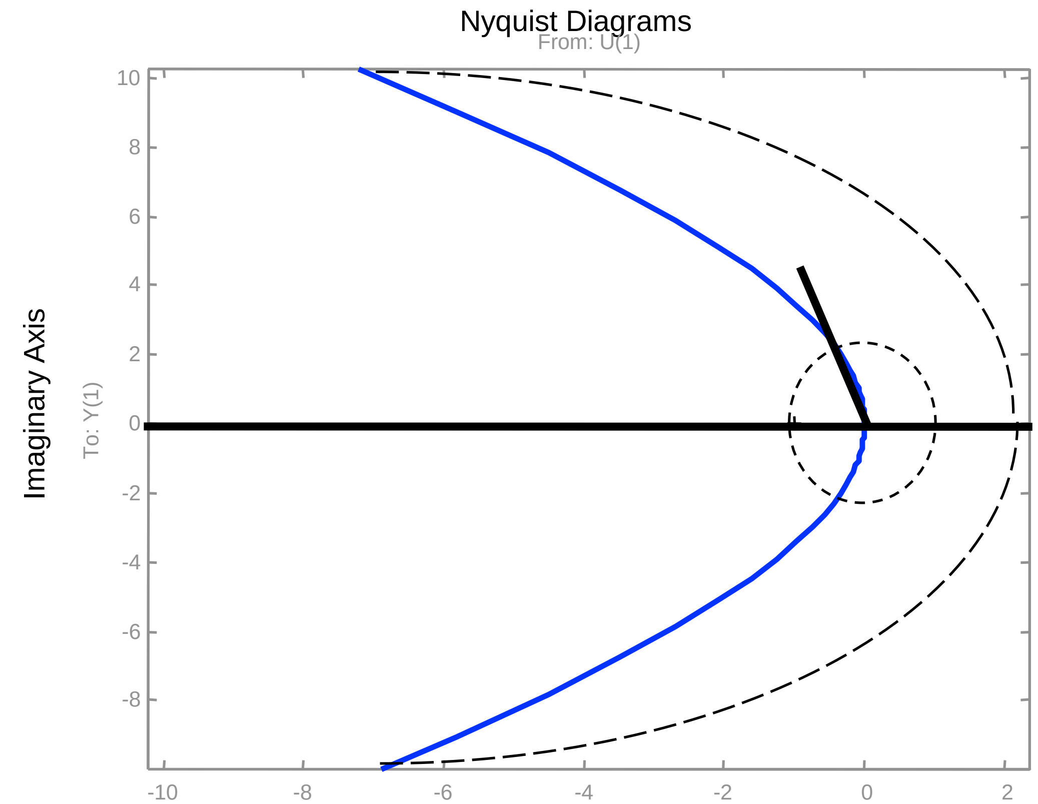
Example: Cruise Control using PID - Verification

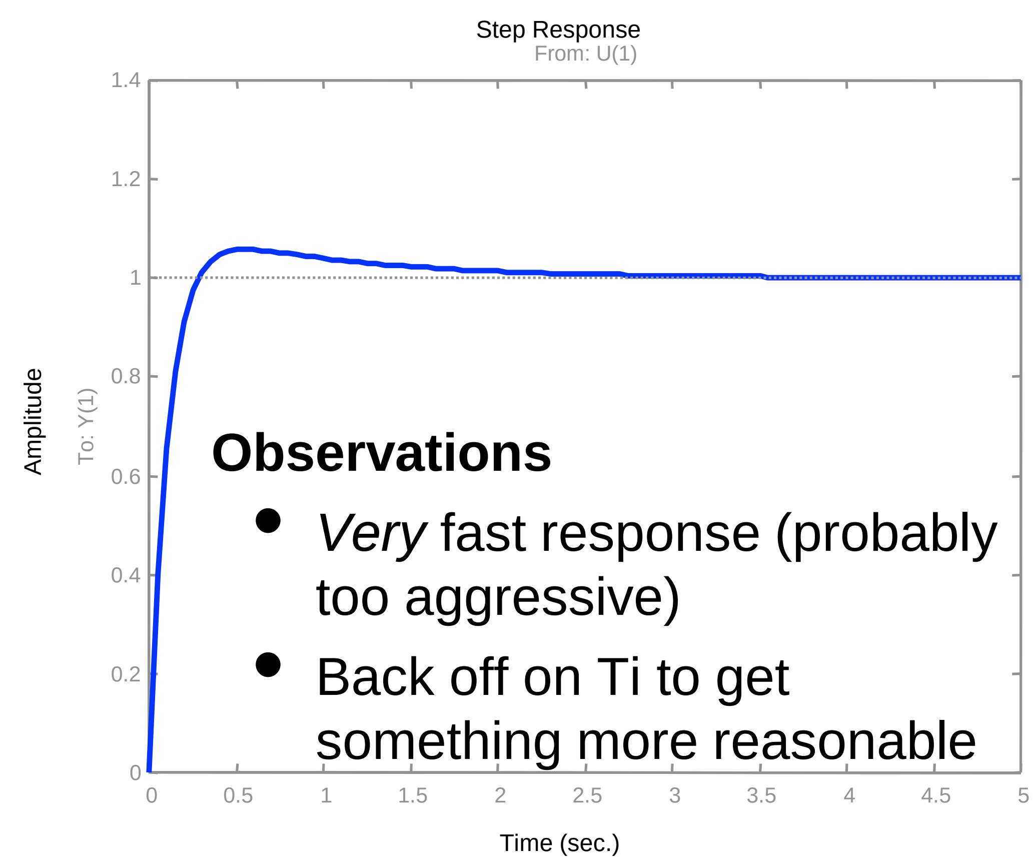


$$P(s) = \frac{1/m}{s + b/m} \times \frac{r}{s+a}$$

$$C(s) = 2000 \frac{s^2 + 1.1s + 0.1}{s}$$



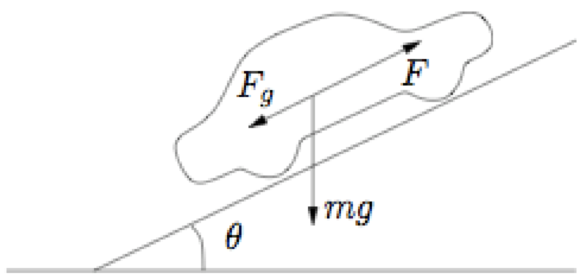




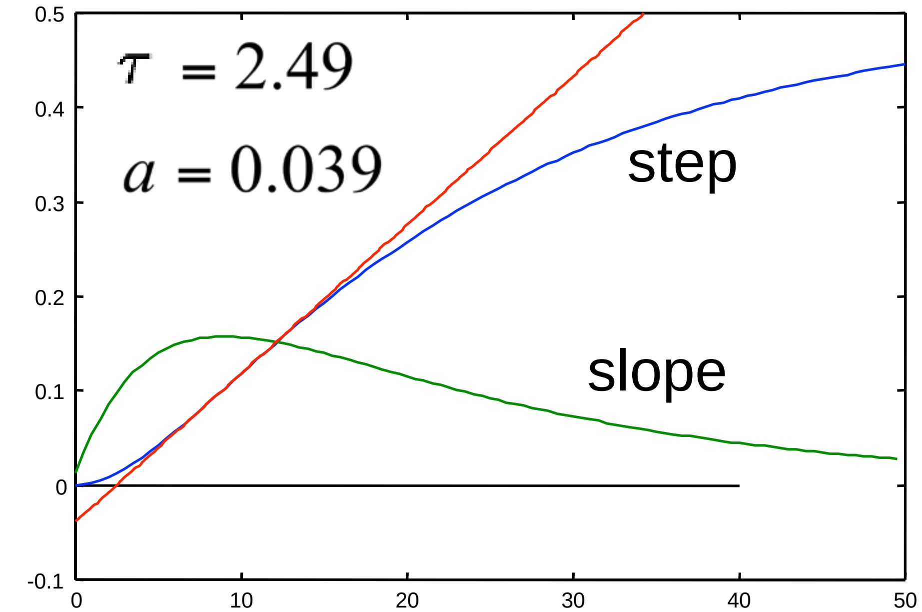
Observations

- *Very* fast response (probably too aggressive)
- Back off on Ti to get something more reasonable

Example: PID cruise control

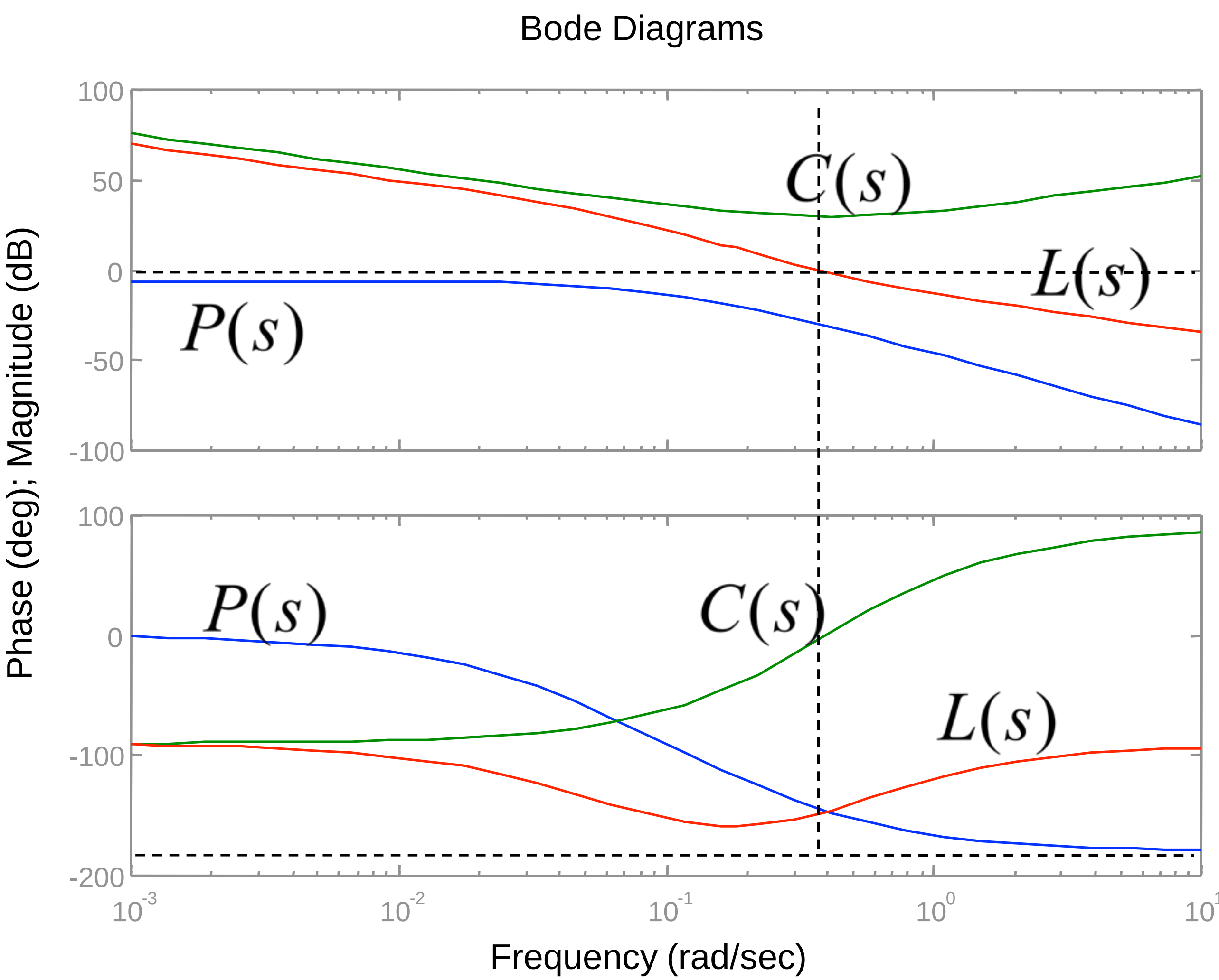


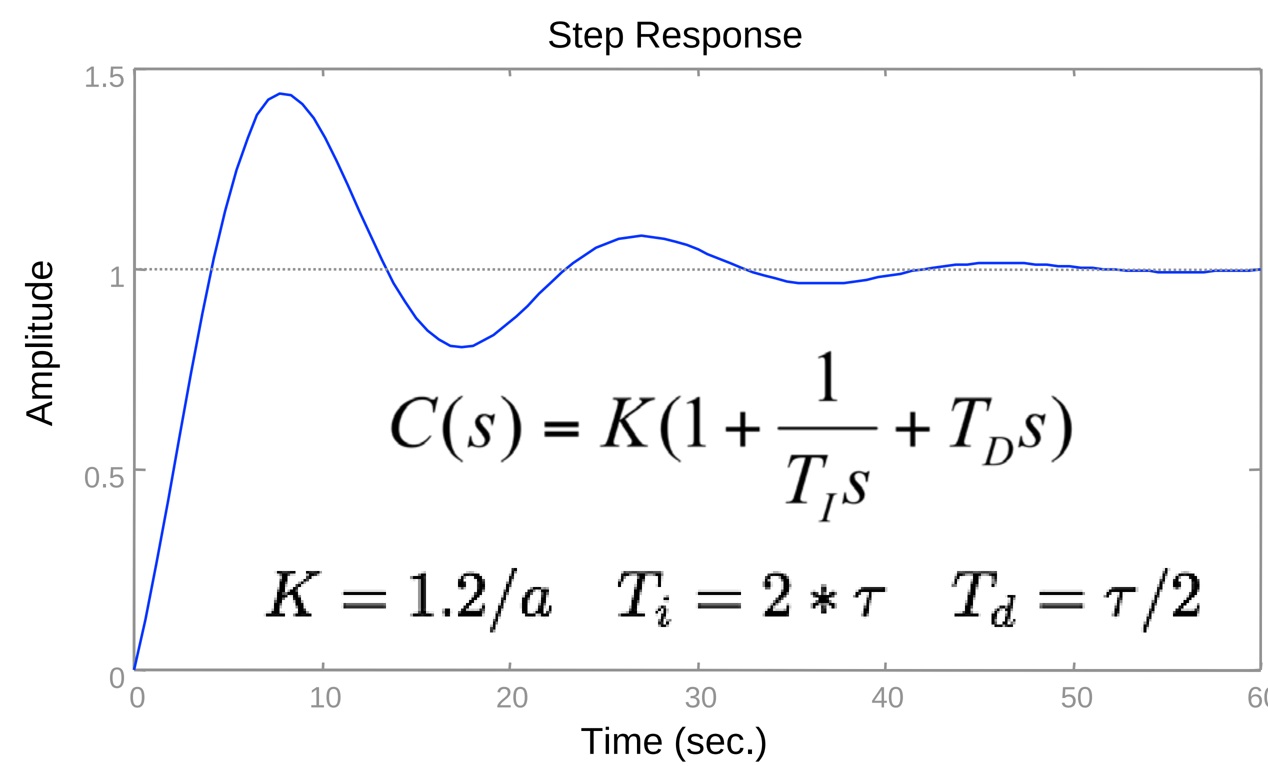
$$P(s) = \frac{1/m}{s + b/m} \times \frac{r}{s+a}$$



Ziegler-Nichols design for cruise controller

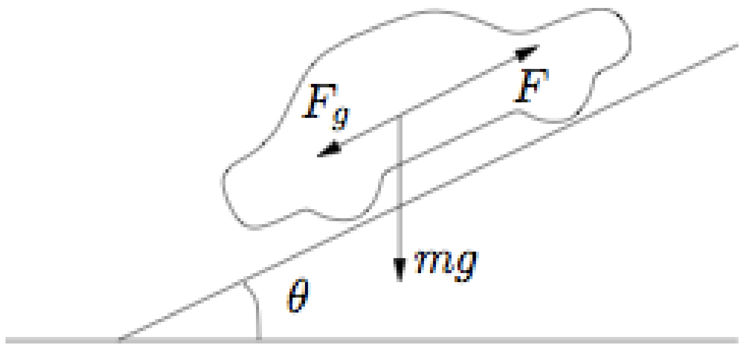
- Plot step response, extract τ and a, compute gains

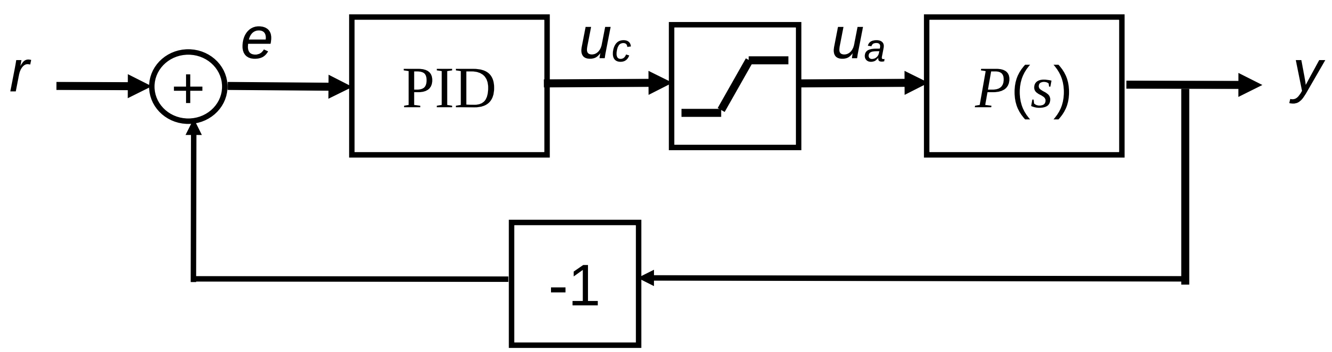




• Result: *sluggish* ® increase loop gain + more phase margine (shift zero)

Windup and Anti-Windup Compensation



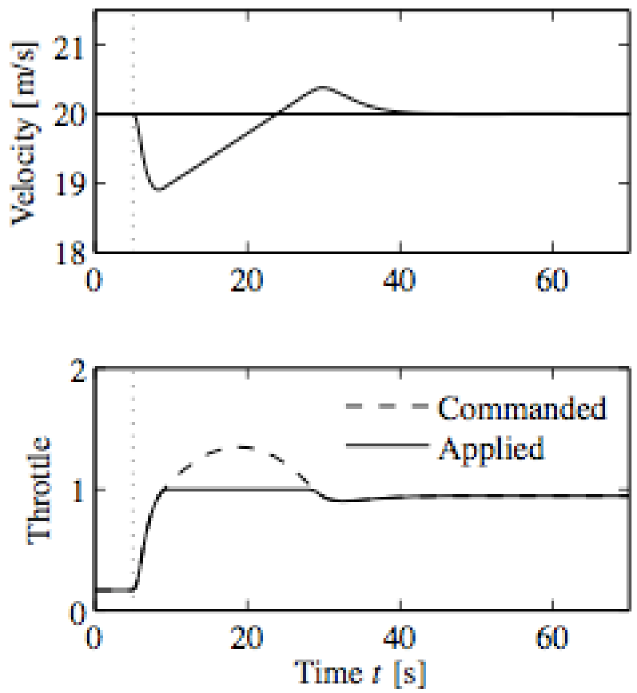


Problem

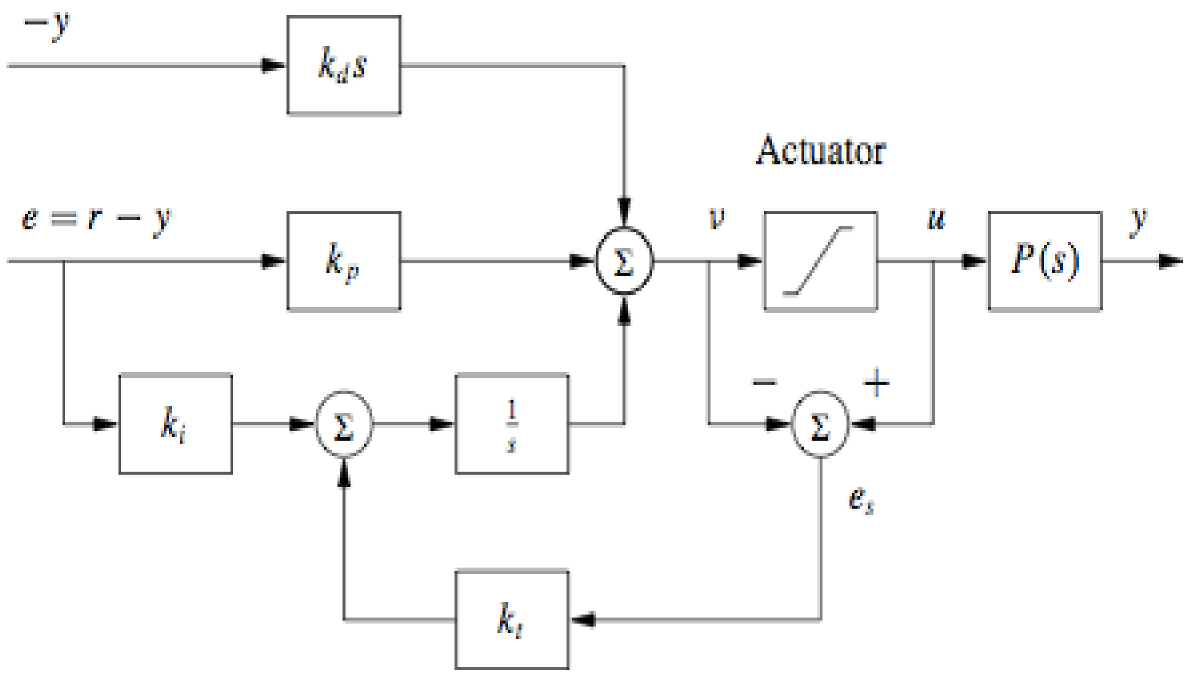
- Limited magnitude input (saturation)
- Integrator "winds up" => overshoot

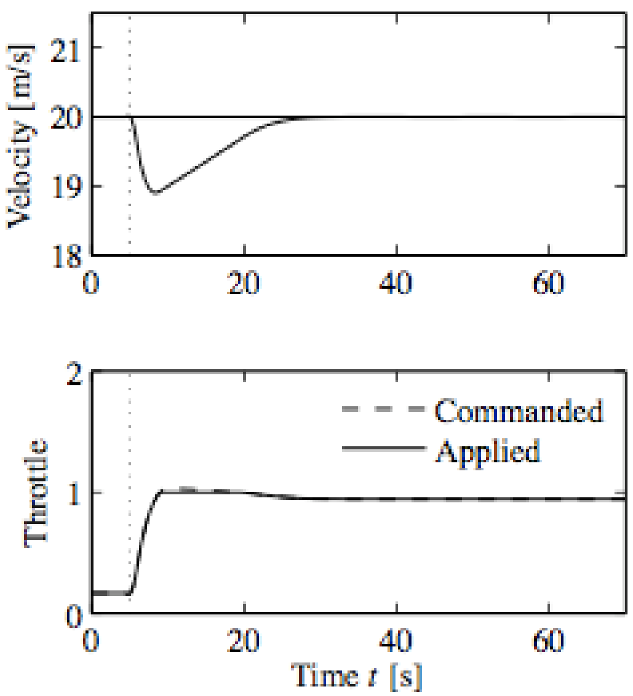
Solution

- Compare commanded input to actual
- Subtract off difference from integrator



(a) Windup





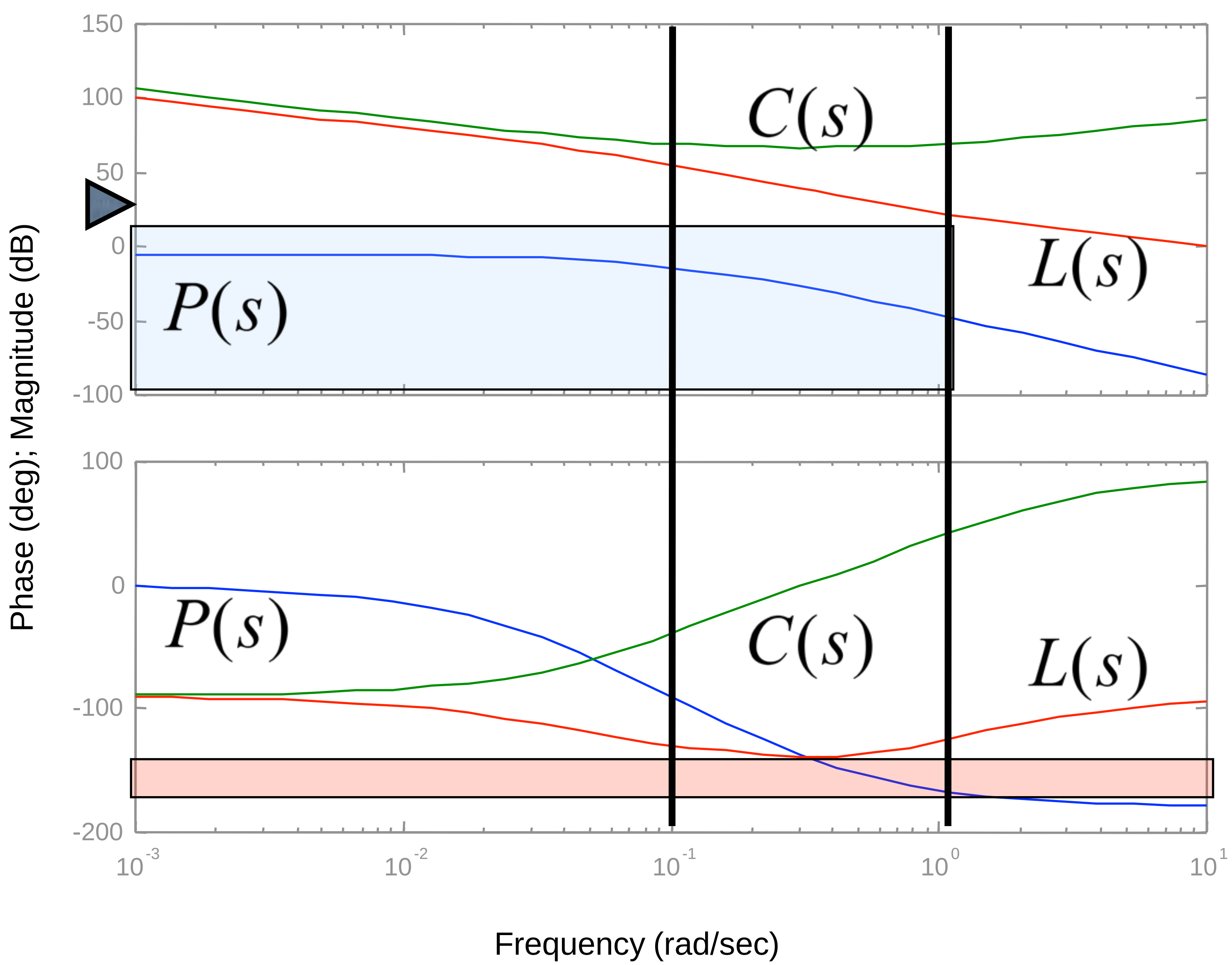
(b) Anti-windup

Summary: Frequency Domain Design using PID

Loop Shaping for Stability & Performance

- Steady state error, bandwidth, tracking

$$H_{ue}(s) = K_p + K_I \times \frac{1}{s} + K_D s$$



Main ideas

- Performance specs give bounds on loop transfer function
- Use controller to shape response
- Gain/phase relationships constrain design approach
- Standard compensators: proportional, PI, PID