CDS 101/110: Lecture 8.2
PID Control

November 18, 2016

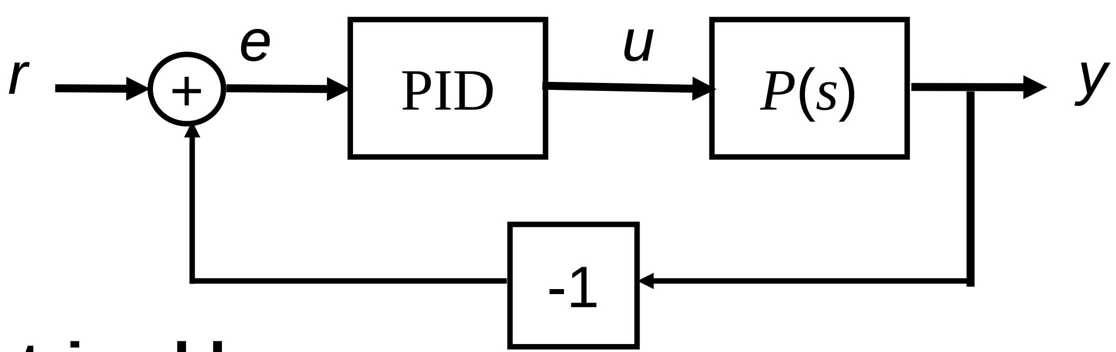
Goals:

- Review PID control.
- Example of PID “loop shaping”
- Discuss the use of integral feedback and anti-windup compensation
- Aside: time delay in feedback systems.

Reading:

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

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:

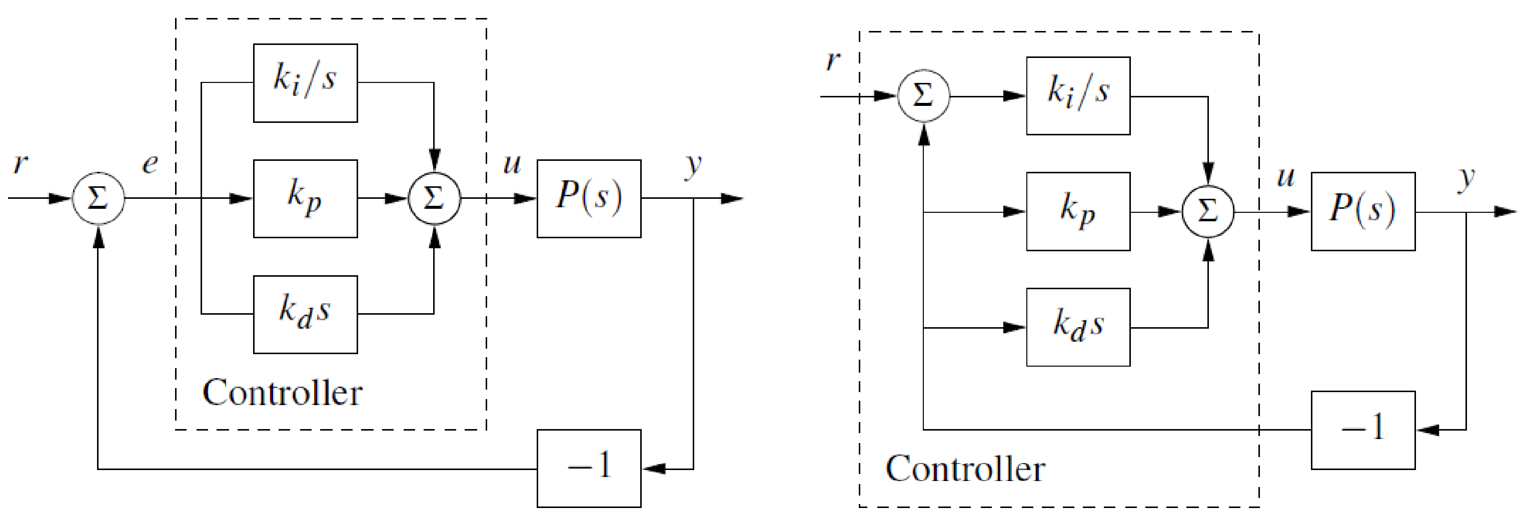
- k_p, the "proportional gain"
- k_i, the "integral gain"
- k_d, the "derivative gain"

Alternatively:

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

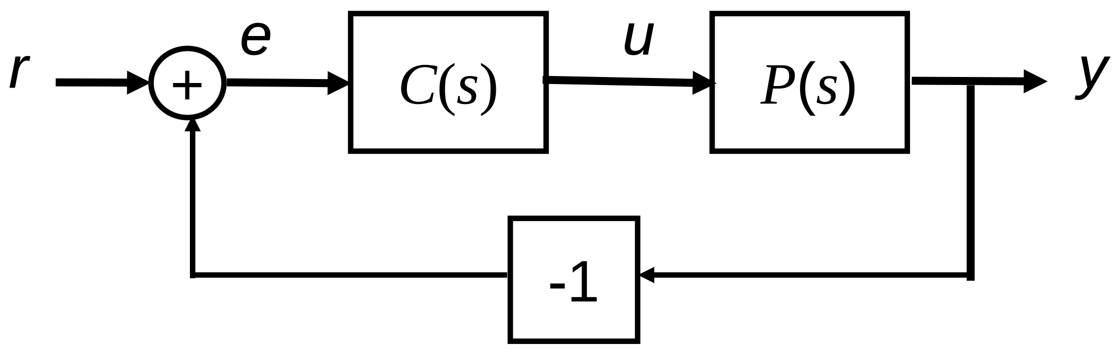
Utility of PID

- 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 + Integral + Derivative (PID)



$$C(s) = k_p + k_i \frac{1}{s} + k_d s$$

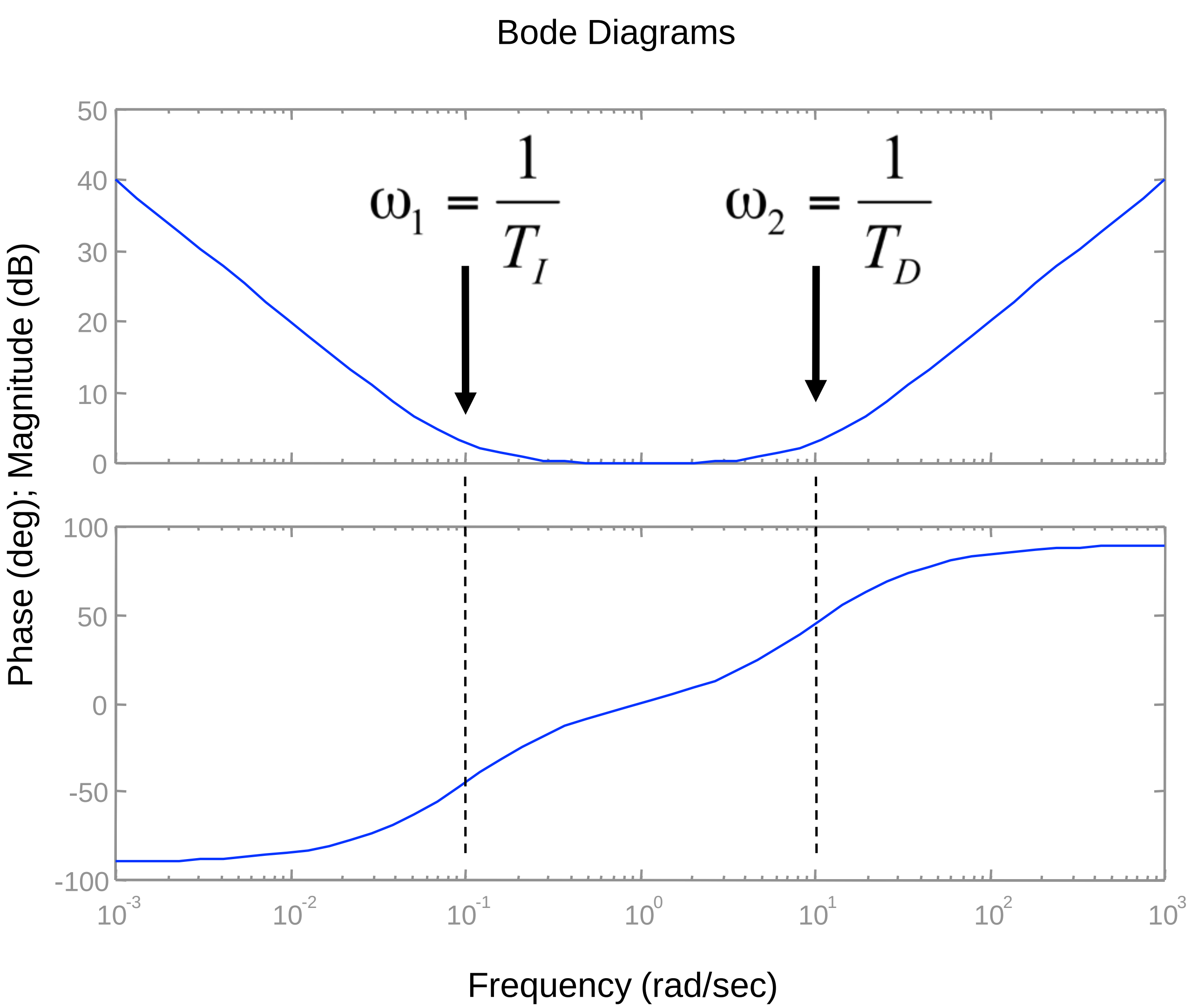
$$= k_p \left(1 + \frac{1}{T_i s} + T_d s\right)$$

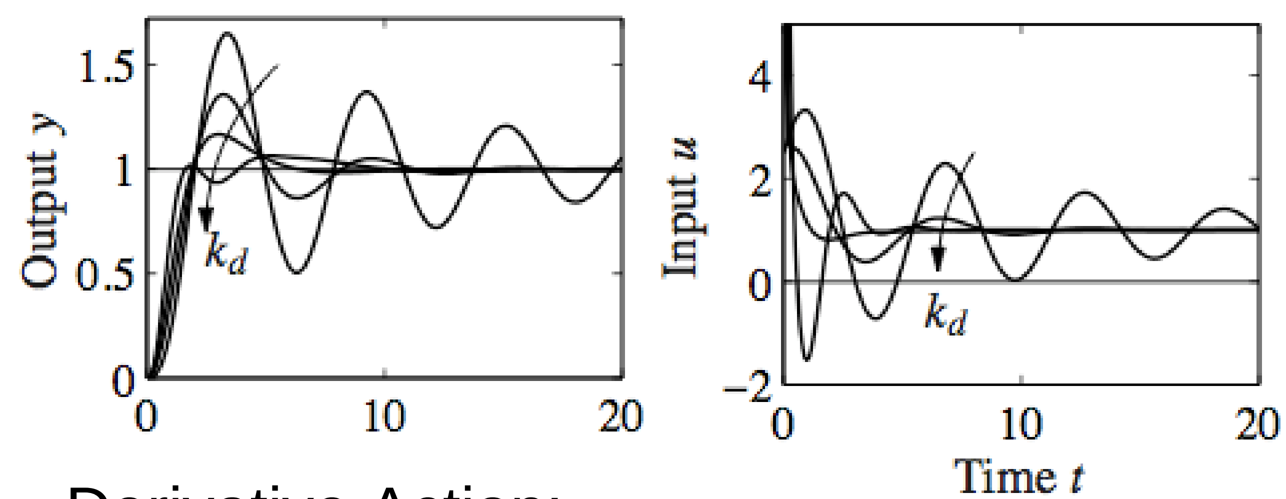
$$= \frac{k_p}{T_i}$$

$$= \frac{k_p T_d}{T_i} \frac{(s + 1/T_i)(s + 1/T_d)}{s}$$

$$T_i = \frac{k_p}{k_i}$$

$$T_d = \frac{k_d}{k_p}$$

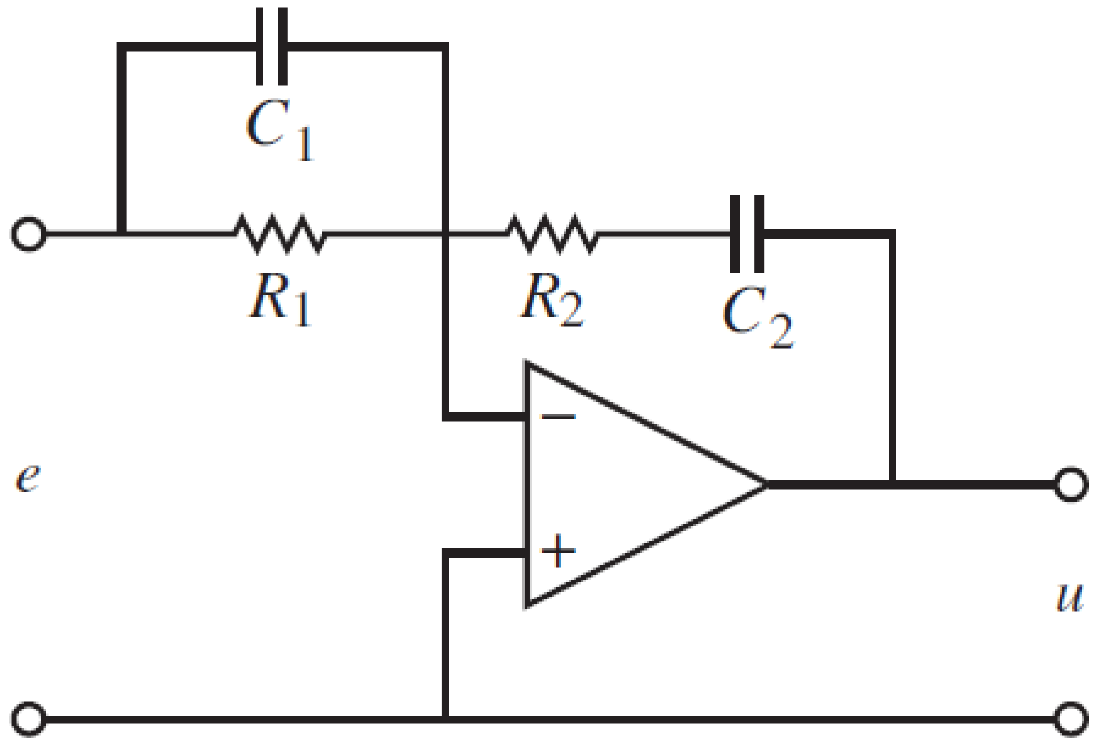




Derivative Action:

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

PID Controllers are easy to implement



Galil “Controller Board”

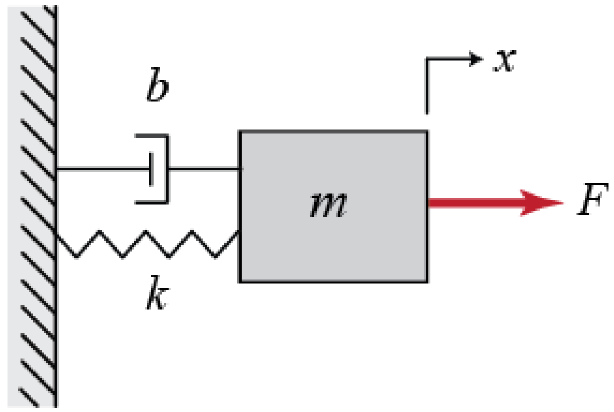
$$u = -\frac{Z_2}{Z_1} e = -\frac{R_2}{R_1} \frac{(1 + R_1 C_1 s)(1 + R_2 C_2 s)}{R_2 C_2 s} e.$$

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

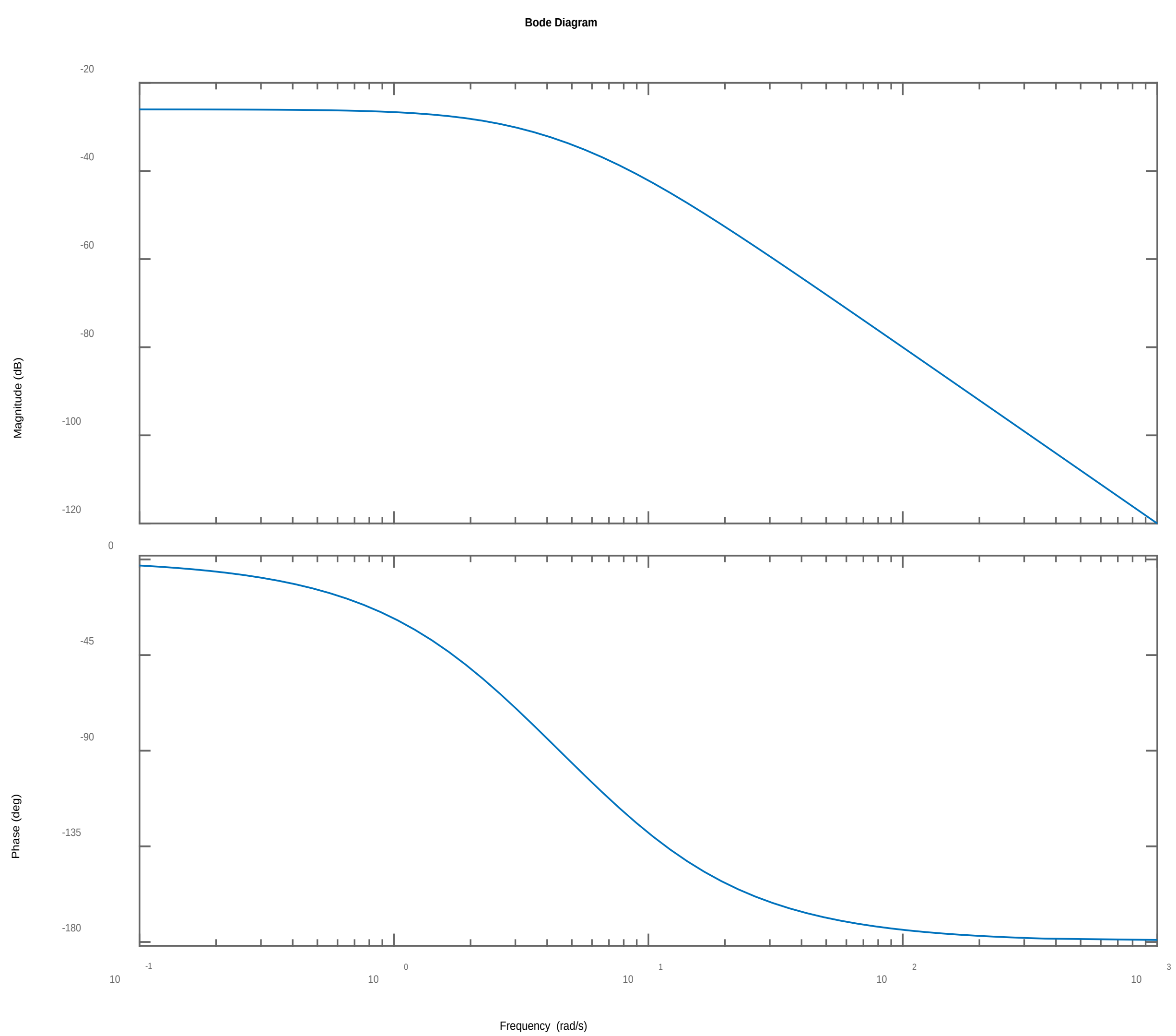
Built in Discrete Time PID

Built in “gain tuning” procedures

Example: Mass-Spring-Damper

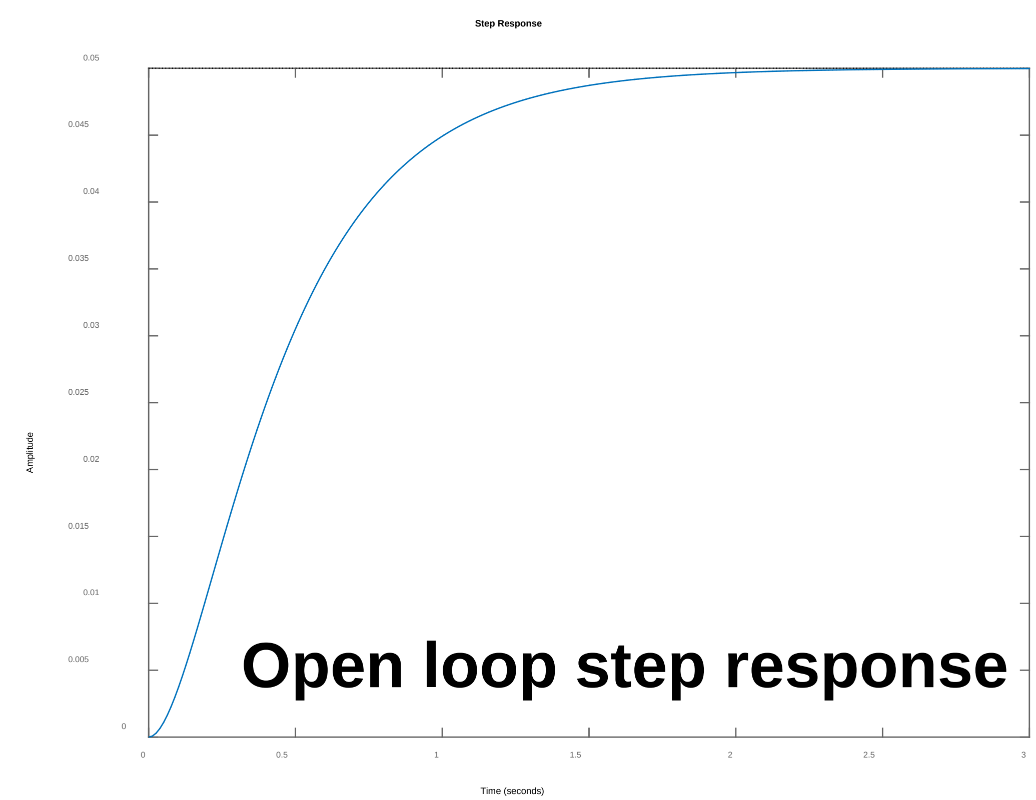


$$m\ddot{x} + b\dot{x} + \mathrm{kx} = \mathrm{F} \quad \rightarrow \quad \frac{X(s)}{F(s)} = \frac{1}{ms^2 + bs + k}$$



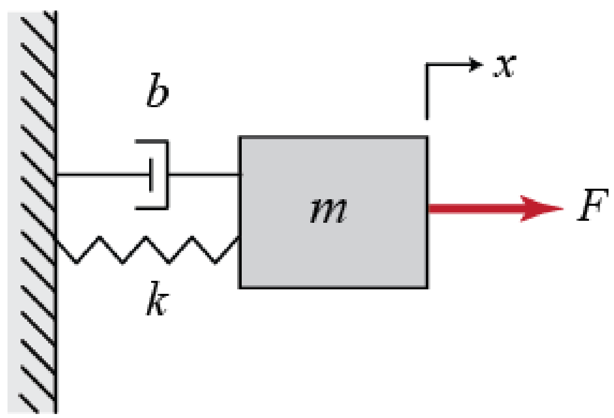
Example Parameters

- $m = 1$ kg
- $b = 10\ N\frac{sec}{m}$
- $k = 20\frac{\mathrm{N}}{\mathrm{m}}$
- $P(s) = \frac{1}{s^2+10s+20}$



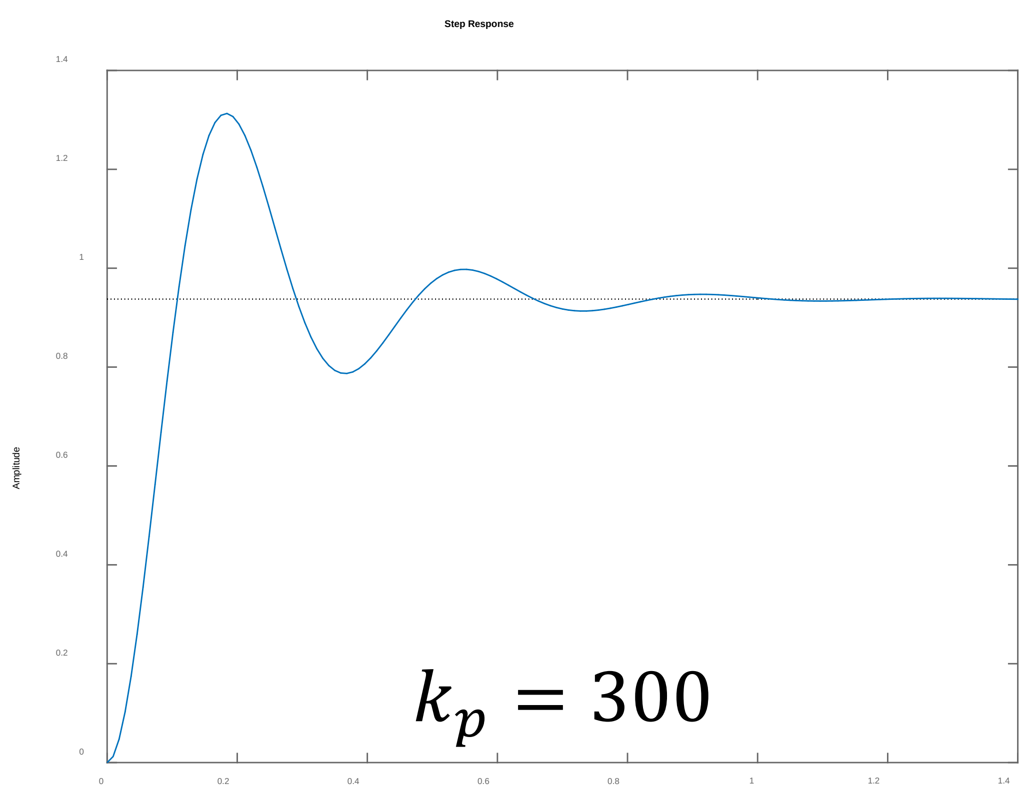
Open loop step response

Example: Mass-Spring-Damper (2)



Proportional Control: $F = k_p e$:

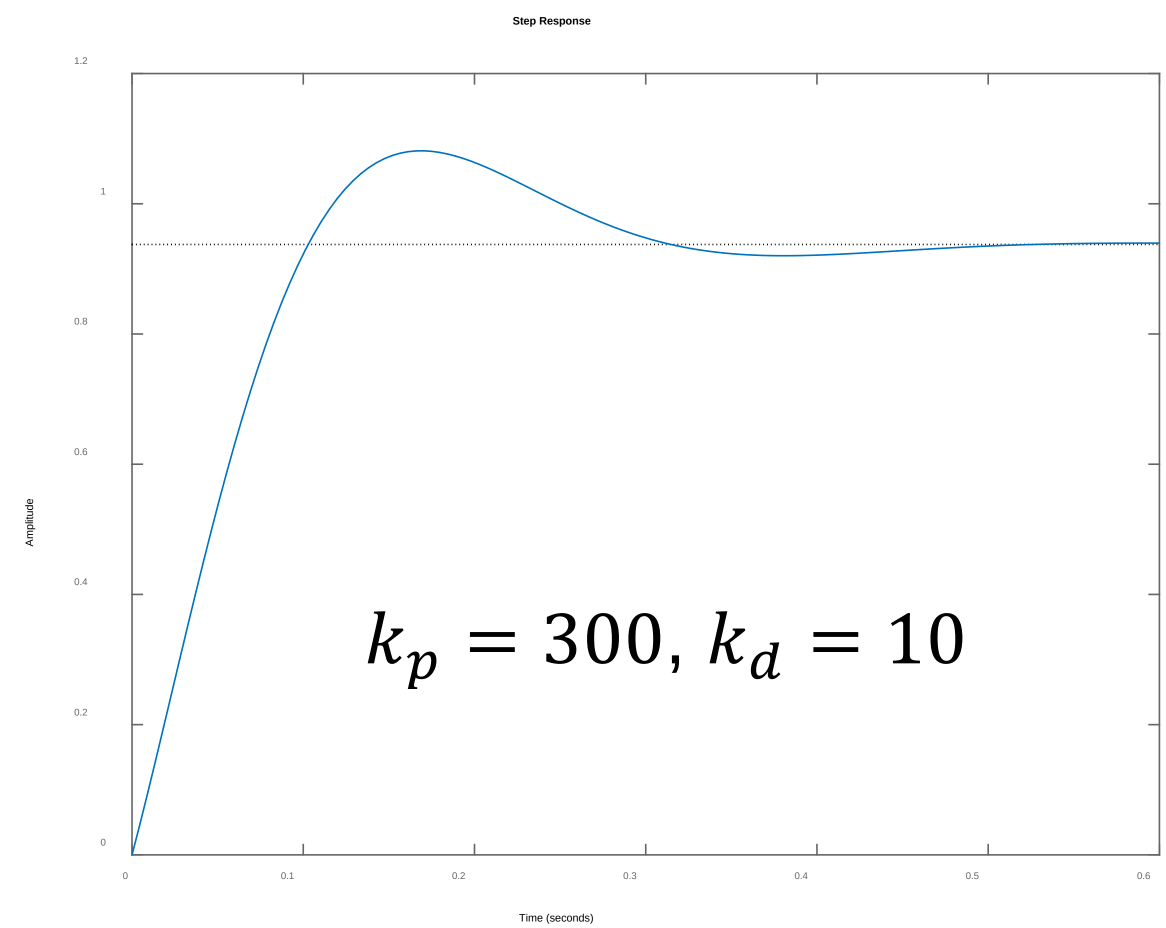
$$G_{yr}(s) = \frac{k_p}{ms^2 + bs + (k + k_p)}$$



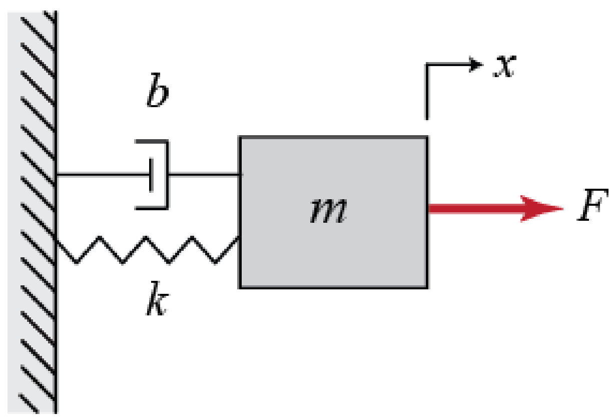
Proportional + Derivative Control:

$$F = k_p e + k_d \dot{e}$$

$$G_{yr}(s) = \frac{k_d s + k_p}{ms^2 + (b + k_d)s + (k + k_p)}$$



Example: Mass-Spring-Damper (3)



Proportional + Derivative + Integral Control:

$$F = k_p e + k_d \dot{e} + k_i \int e(\tau) d\tau$$

$$\mathrm{C(s)} = \frac{k_d s^2 + k_p s + k_i}{s} = \frac{k_p T_d}{T_i} \frac{(s + 1/T_i)(s + 1/T_d)}{s}$$

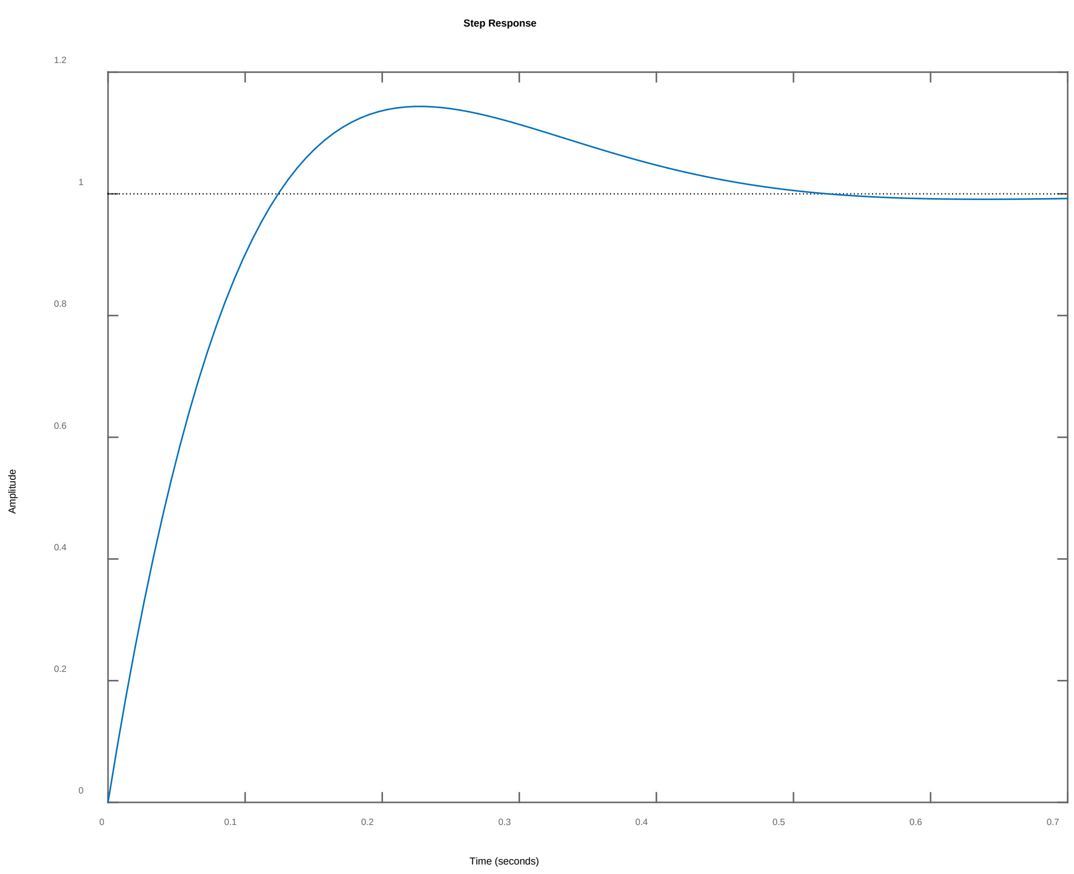
$$\mathrm{G_{yr}(s)} = \frac{P(s)C(s)}{1 + P(s)C(s)}$$

$$= \frac{n_P(s) n_C(s)}{d_P(s) d_C(s) + n_P(s) n_C(s)}$$

$$= \frac{k_d s^2 + k_p s + k_i}{m s^3 + (b + k_d) s^2 + \left(k + k_p\right) s + k_i}$$

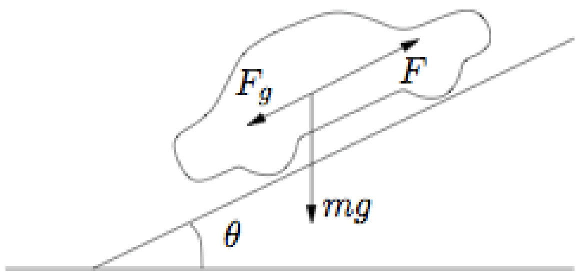
Choose char poly:

$$(s + 10)\left(s^2 + 14s + 100\right)$$
$$= \mathrm{s^3 + 24s^2 + 240s + 1000}$$

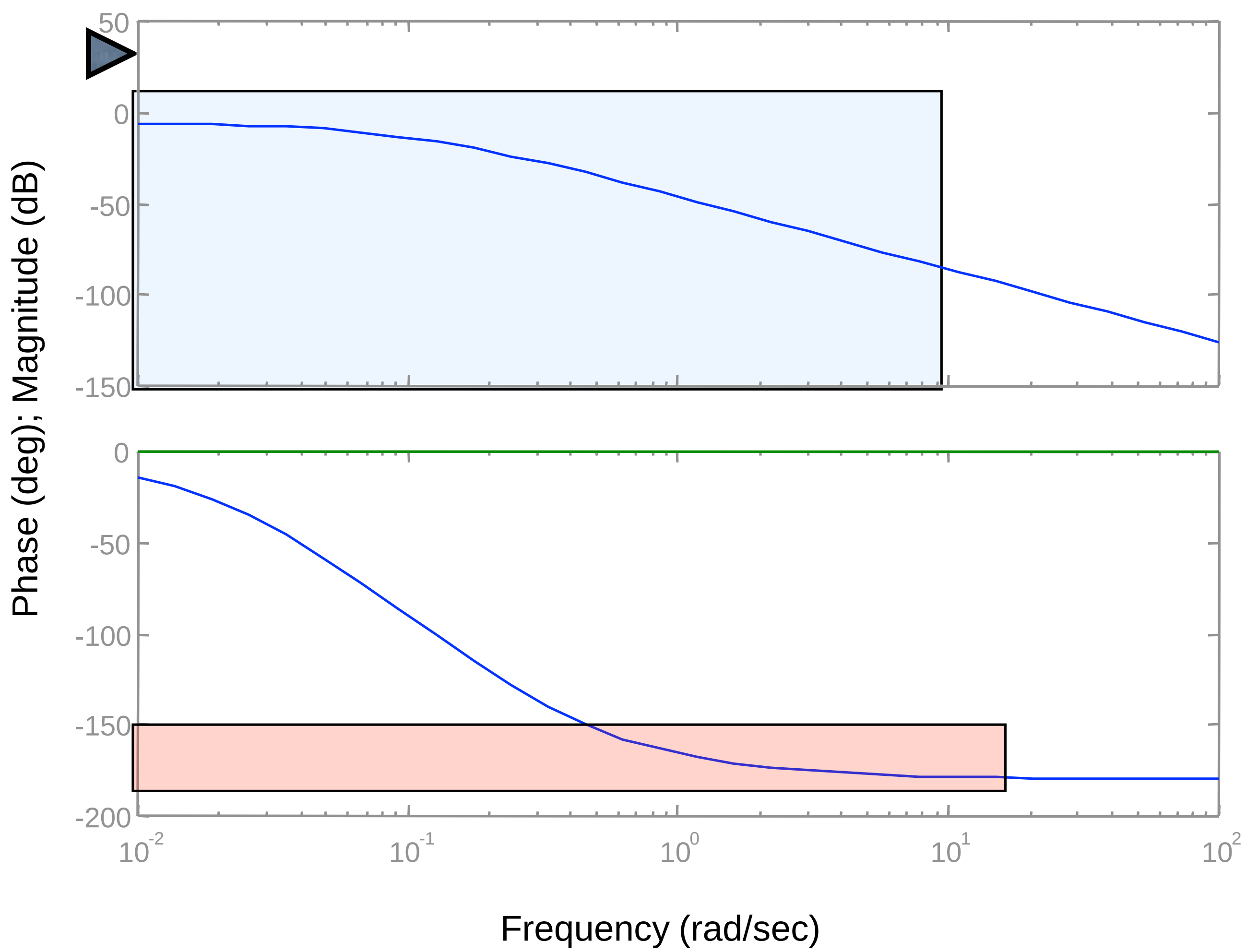


$$k_p = 220,\ k_d = 14,\ k_i = 1000$$

Example: Cruise Control using PID - Specification



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



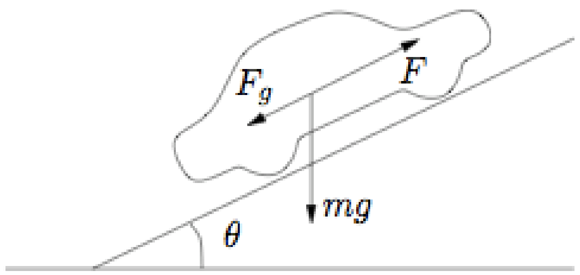
Performance Specification

- ≤ 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

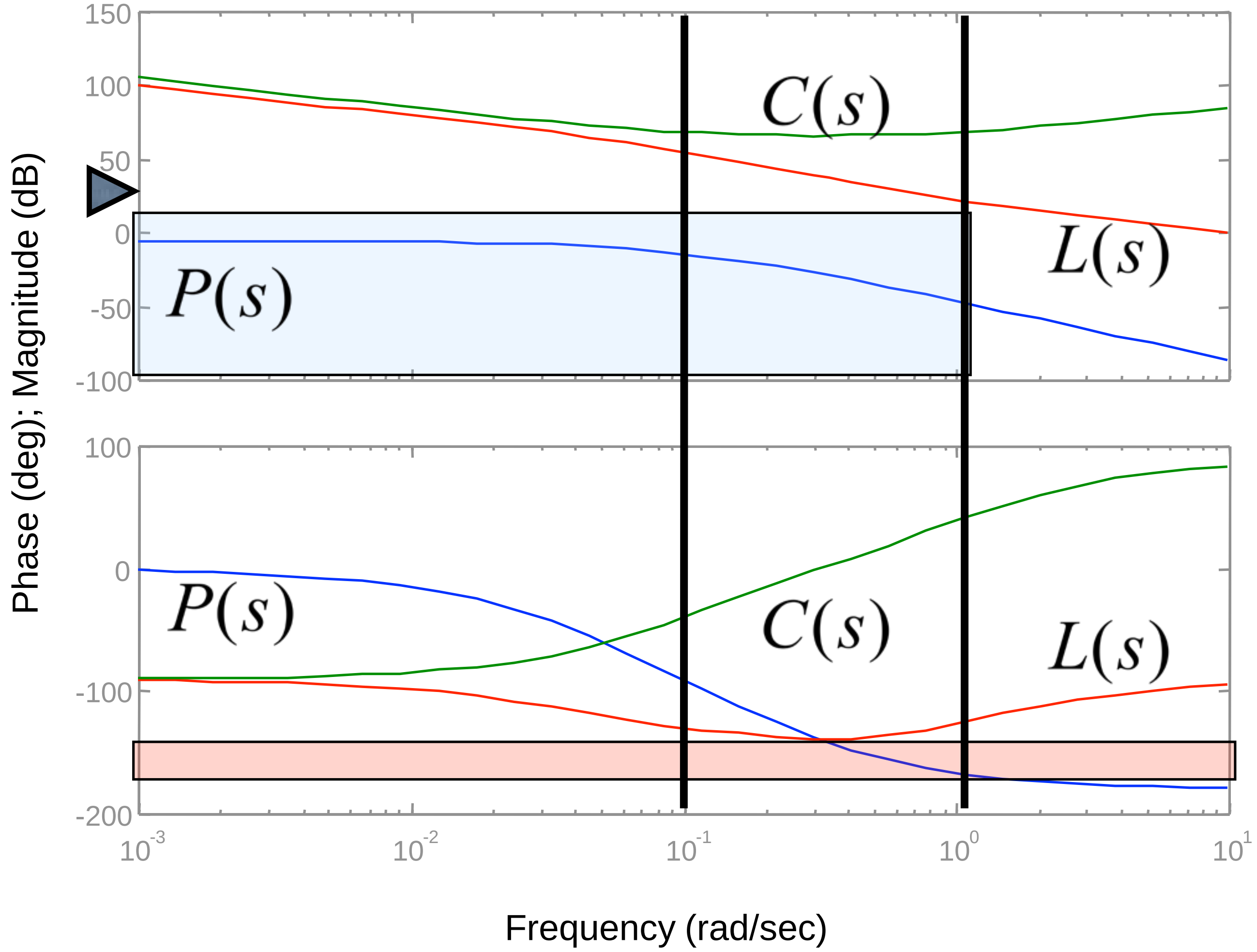
Observations

- 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

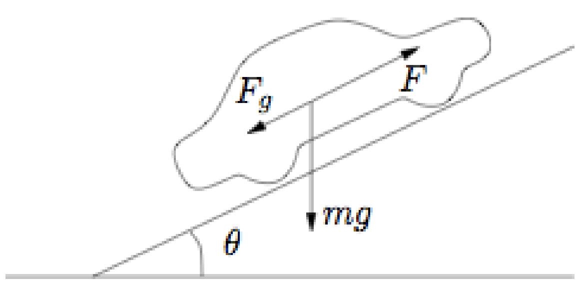
$$C(s) = 2000 \frac{s^2 + 1.1s + 0.1}{s}$$

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

Closed loop system

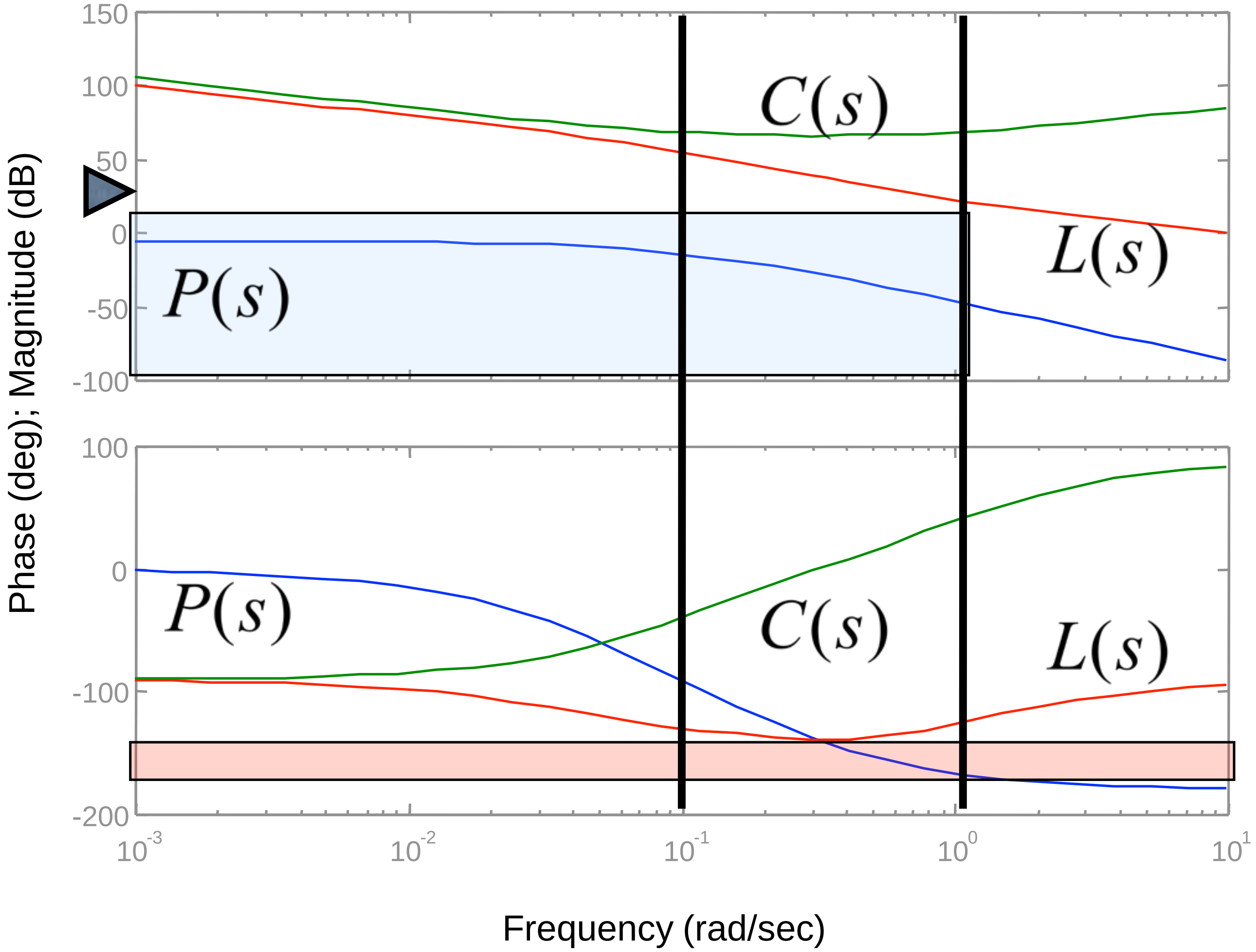
- Very high steady state gain
- Adequate tracking @ 1 rad/sec
- ~80° phase margin
- Verify with Nyquist

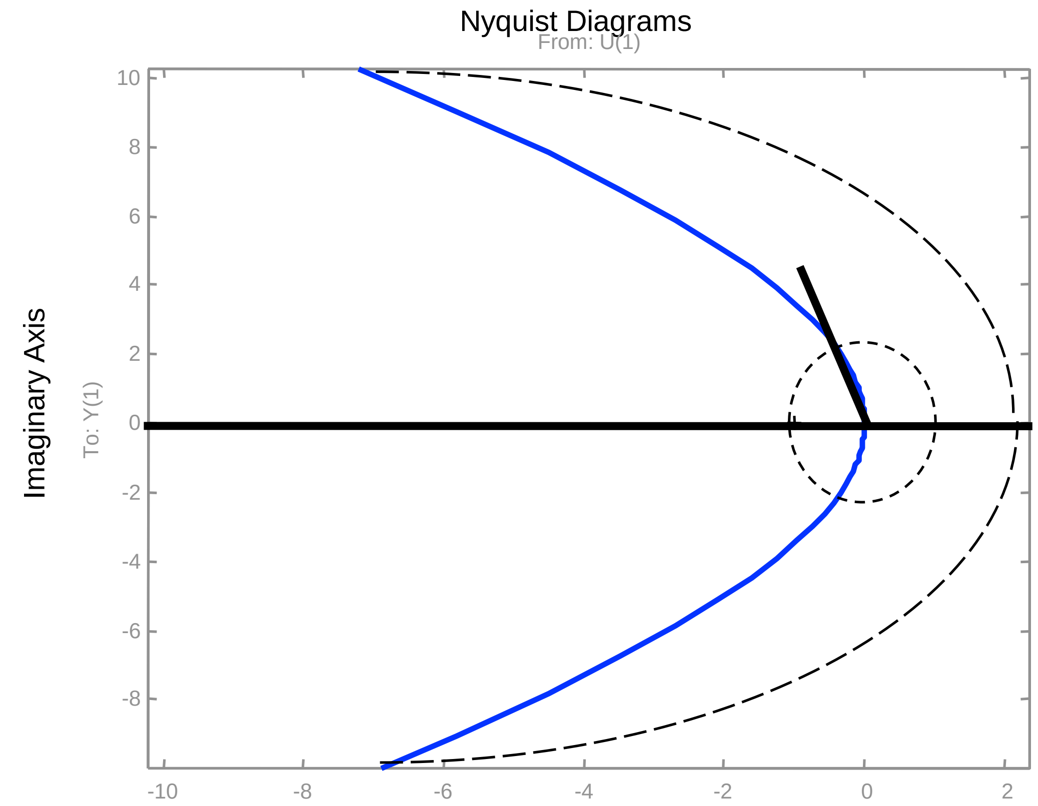
Example: Cruise Control using PID - Verification

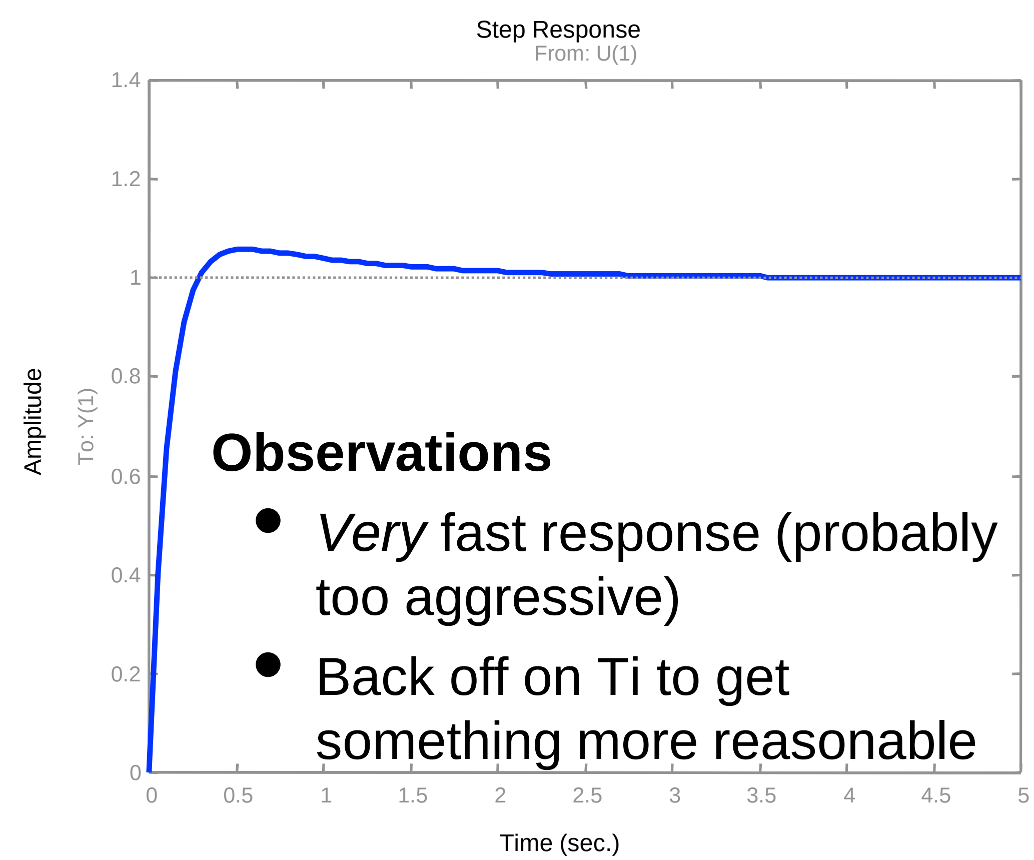


$$P(s) = \frac{1/m}{s + b/m} \times \frac{r}{s+a}$$

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



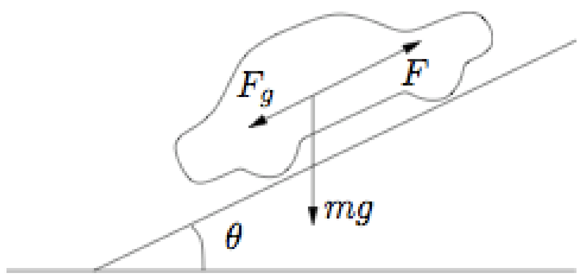




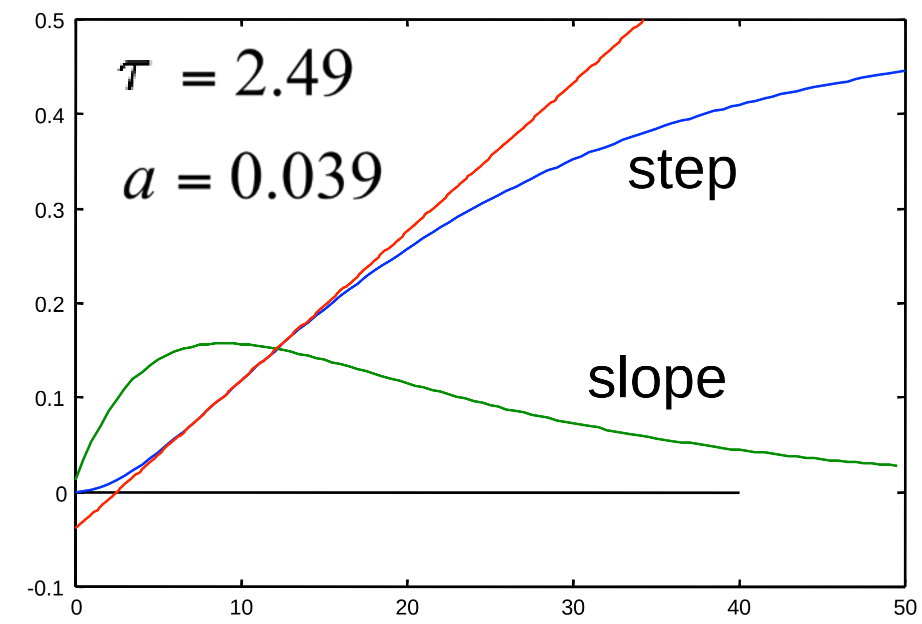
Observations

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

Example: PID cruise control

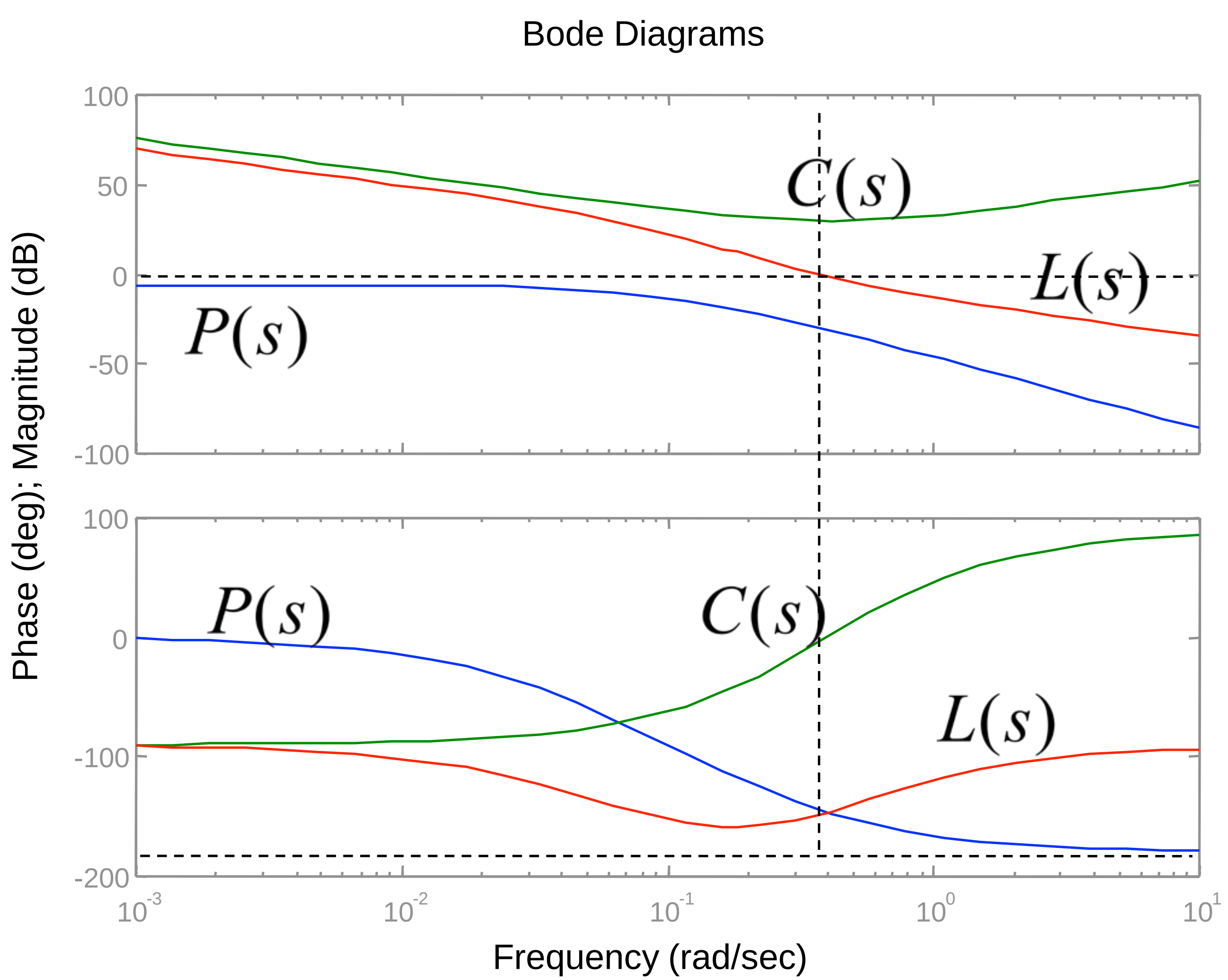


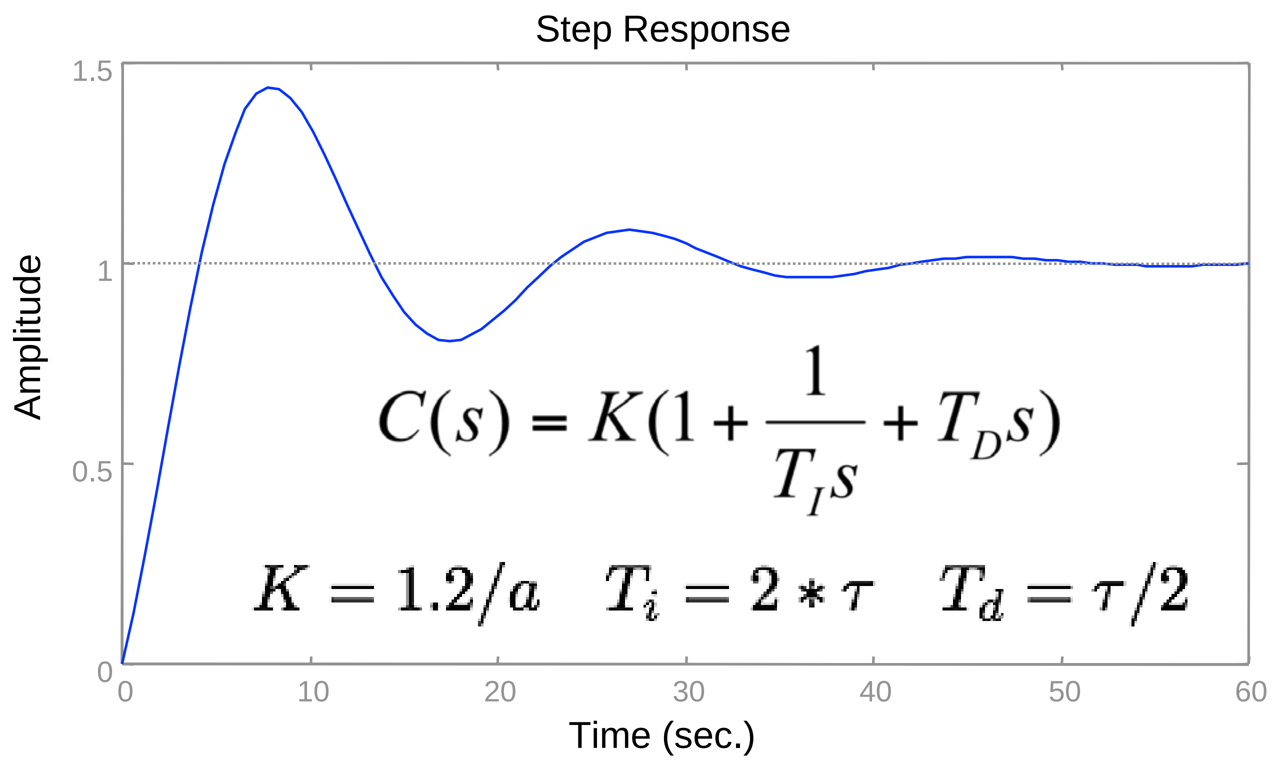
$$P(s) = \frac{1/m}{s + b/m} \times \frac{r}{s+a}$$



Ziegler-Nichols design for cruise controller

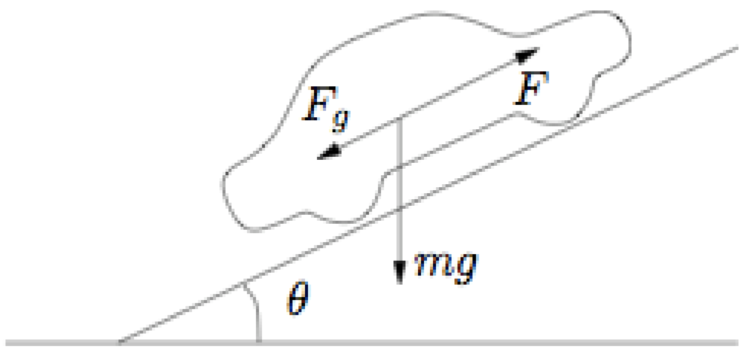
- Plot step response, extract τ and a, compute gains

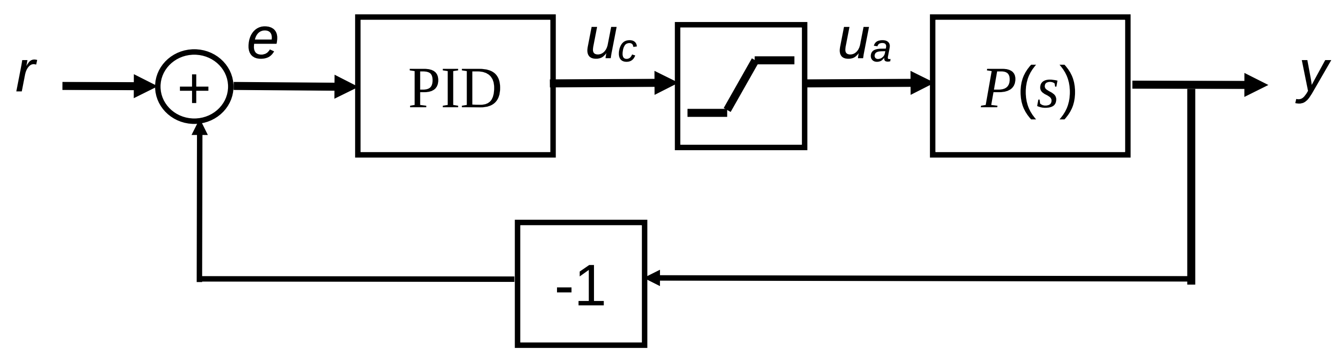




- Result: *sluggish* ⇒ increase loop gain + more phase margin (shift zero)

Windup and Anti-Windup Compensation

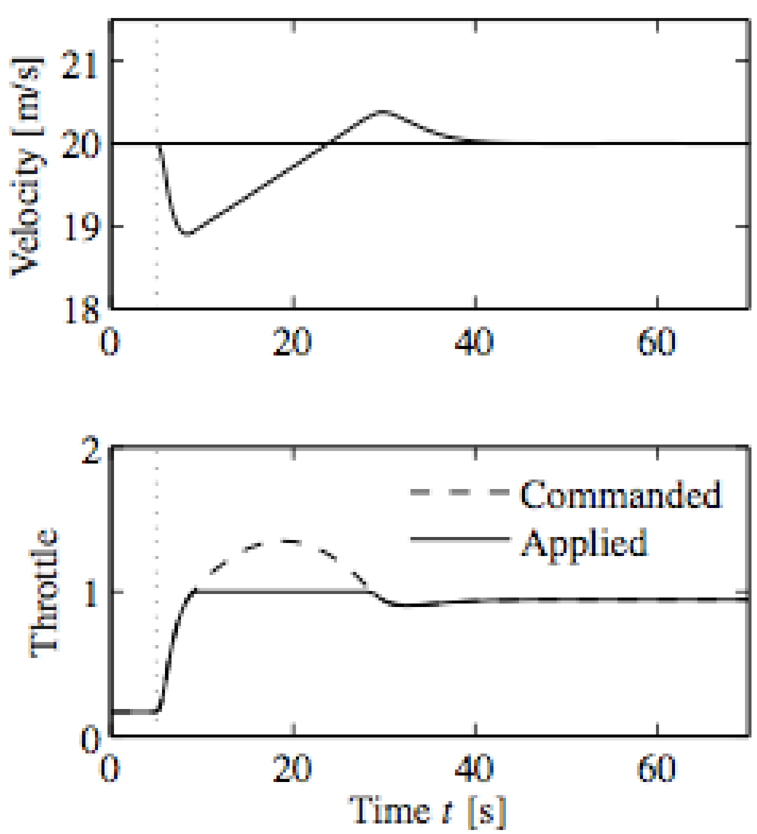




Problem

- Actuators have limits
- High integral error "saturates" inputs

- Control essentially open loop
- Error continues to build
- Integrator "winds up" => overshoot

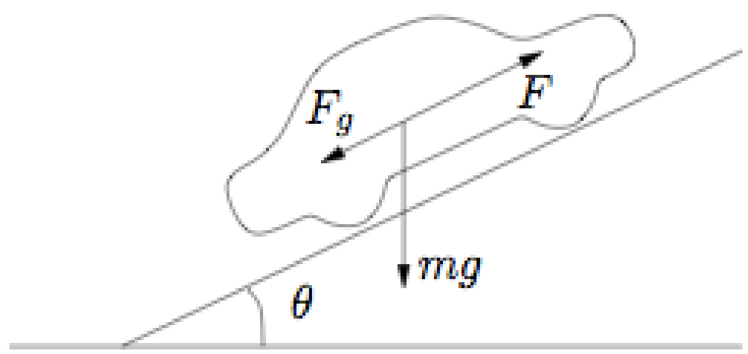


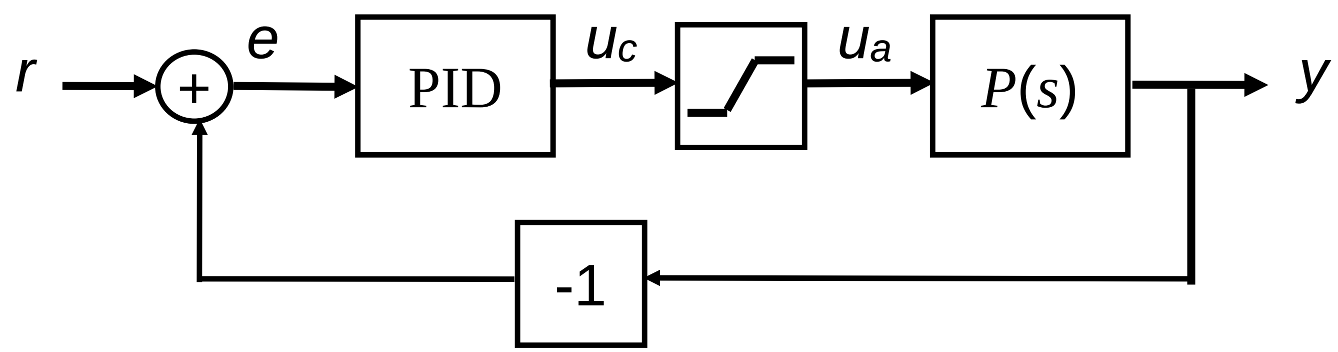
(a) Windup

Car starts up steep hill

- Speed slows—error increases
- Integral term demands more torque
- Engine "saturates" at torque limit
- Torque is enough to eventually overcome gravity
- Car will eventually reach desired speed
- Large overshoot while integral term slowly decreases

Windup and Anti-Windup Compensation

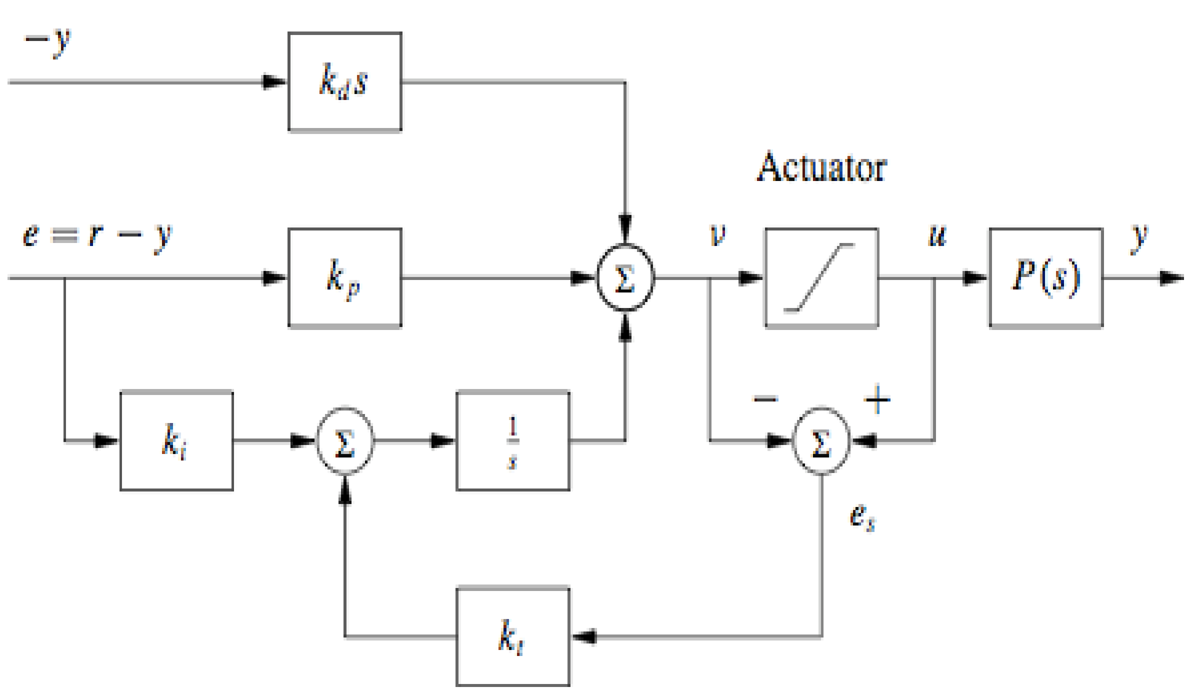


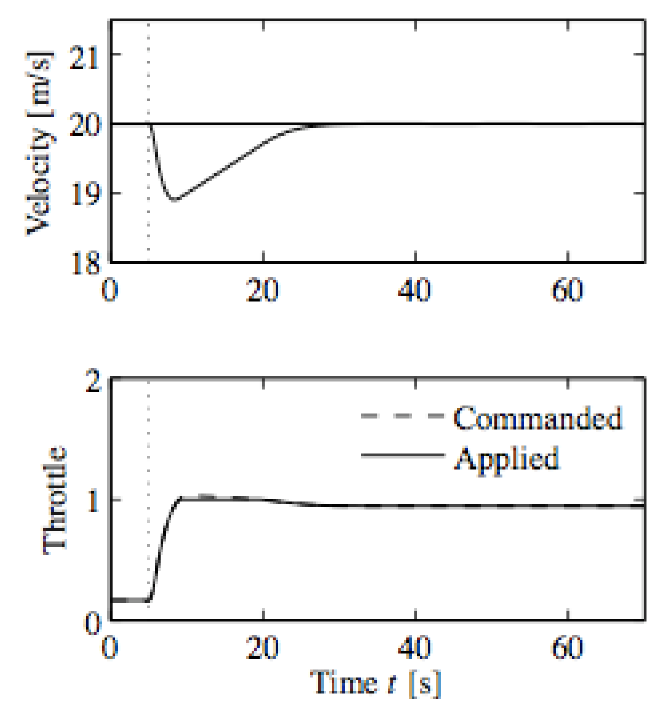


Solution(s)

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

- Essentially a different feedback path
- Keeps controller output near saturation





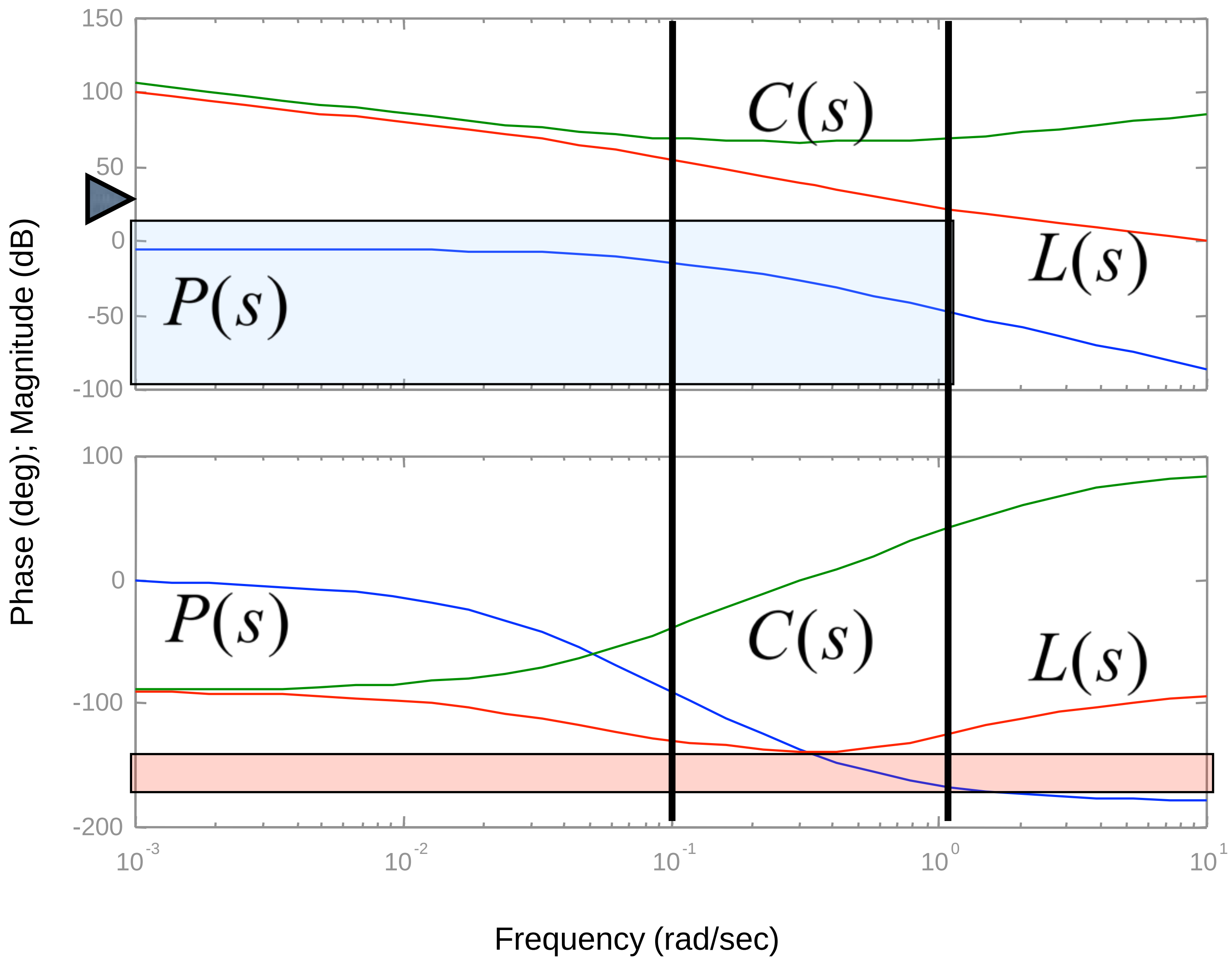
(b) Anti-windup

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