CDS 101: Lecture 8.2
Tools for PID & Loop Shaping

Richard M. Murray

17 November 2004

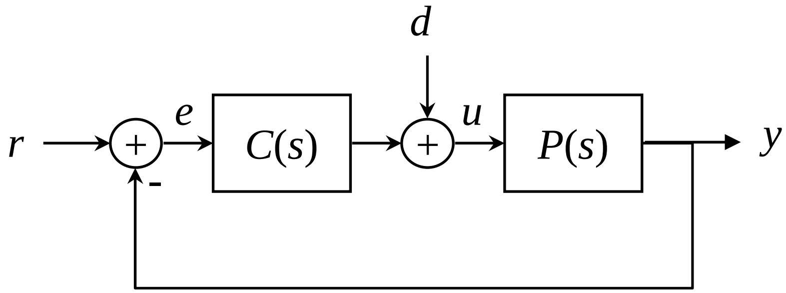
Goals:

- Show how to use “loop shaping” to achieve a performance specification
- Introduce new tools for loop shaping design: Ziegler-Nichols, root locus, lead compensation
- Work through some example control design problems

Reading:

- Åström and Murray, *Analysis and Design of Feedback Systems,* Ch 8

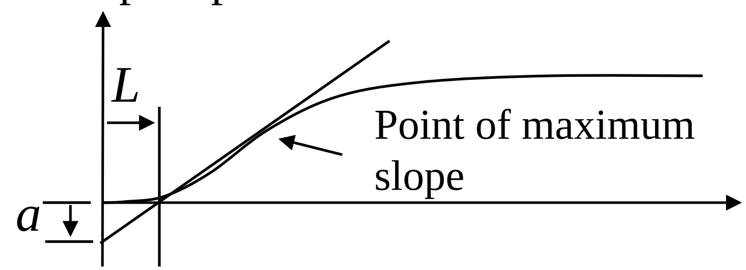
Tools for Designing PID controllers



$$C(s) = K\left(1 + \frac{1}{T_I s} + T_D s\right)$$

Zeigler-Nichols tuning

- Design PID gains based on step response
- Works OK for many plants (but underdamped)
- Good way to get a first cut controller
- Frequency domain version also exists



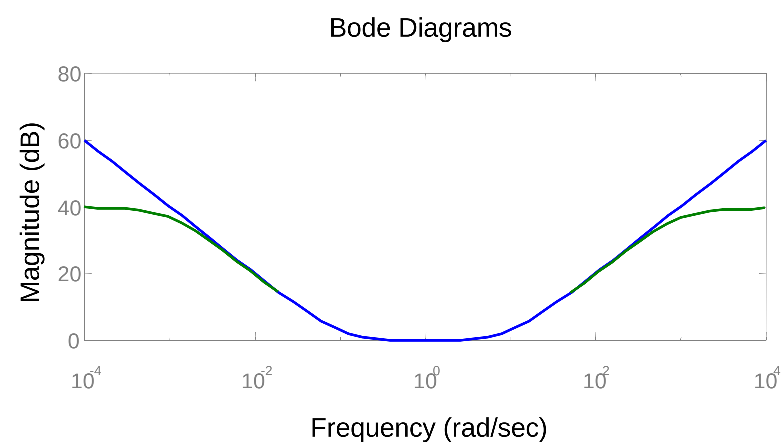
$$K = 1.2/a \qquad T_I = 2 * L \qquad T_D = L/2$$

Caution: PID amplifies high frequency noise

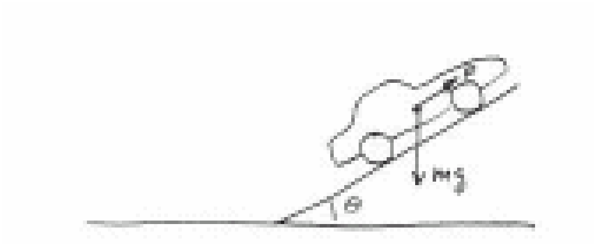
- Sol'n: pole at high frequency

Caution: Integrator windup

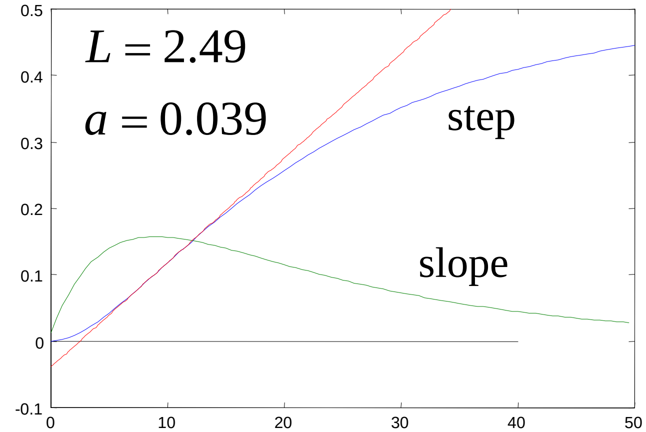
- Prolonged error causes large integrated error
- Effect: large undershoot (to reset integrator)
- Sol'n: move pole at zero to very small value
- Fancier sol'n: anti-windup compensation



Example: PID cruise control

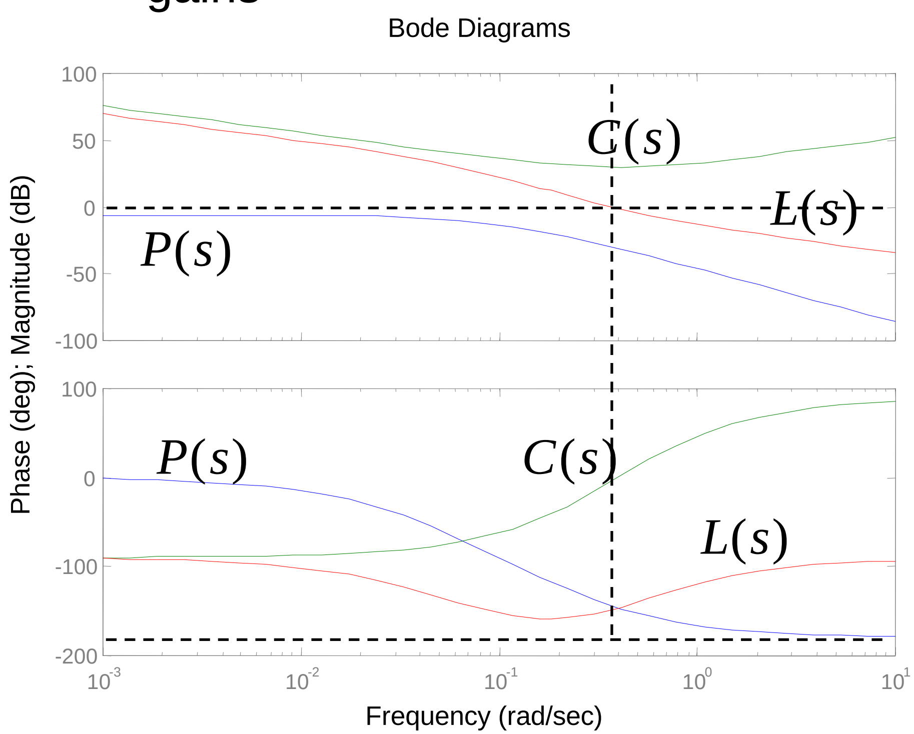


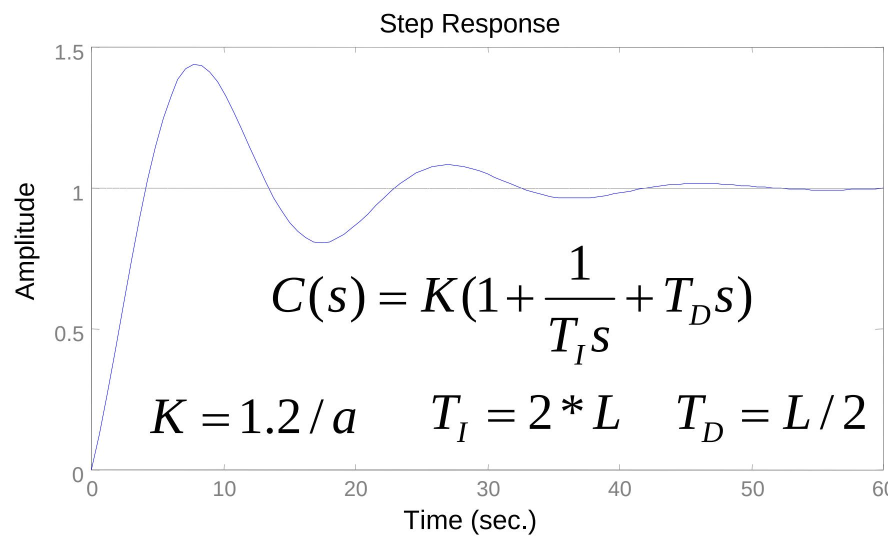
$$P(s) = \frac{1/m}{s + b/m} \cdot \frac{r}{s + a}$$



Ziegler-Nichols design for cruise controller

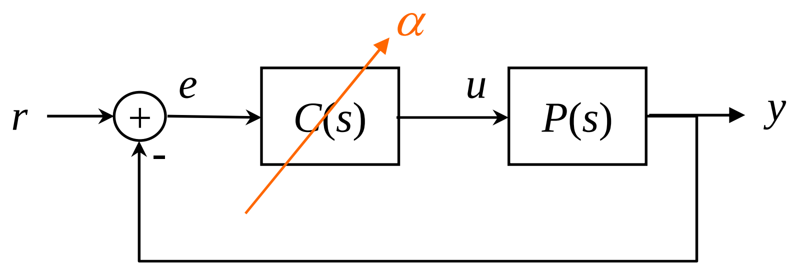
- Plot step response, extract *L* and *a*, compute gains

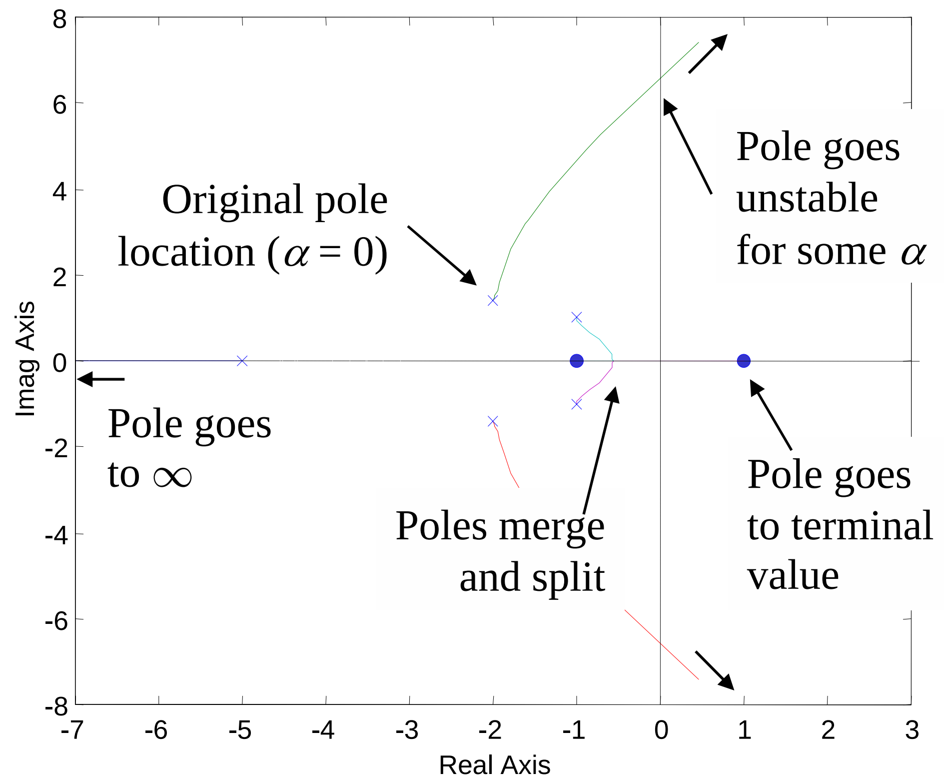




- Result: *sluggish* $\Rightarrow$ increase loop gain

Pole Zero Diagrams and Root Locus Plots





Pole zero diagram verifies stability

- Roots of 1 + *PC* give closed loop poles
- Can *trace* the poles as a parameter is changed:

$$C(s) = K\left(1 + \frac{1}{T_I s} + T_D s\right)$$

α

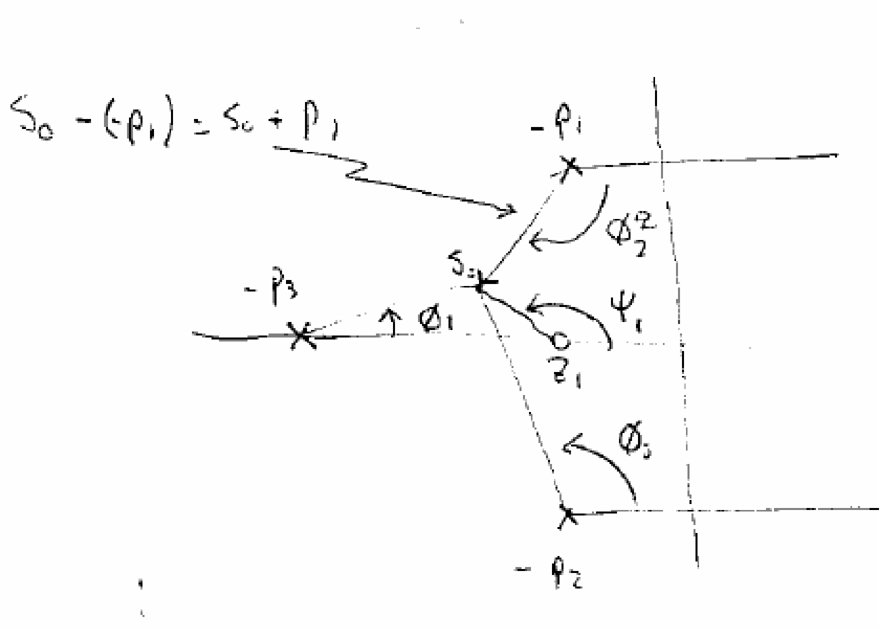
Root locus = locus of roots as parameter value is changed

- Can plot pole location versus *any* parameter; just repeatedly solve for roots
- Common choice in control is to vary the loop gain (*K*)

One Parameter Root Locus

Basic idea: convert to "standard problem": $a(s) + \alpha b(s) = 0$

- Look at location of roots as α is varied over *positive real* numbers
- If "phase" of $a(s)/b(s)$ = 180°, we can always choose a real α to solve eqn
- Can compute the phase from the pole/zero diagram



ϕ_i = phase contribution from s_0 to $-p_i$

ψ_l = phase contribution from s_0 to $-z_i$

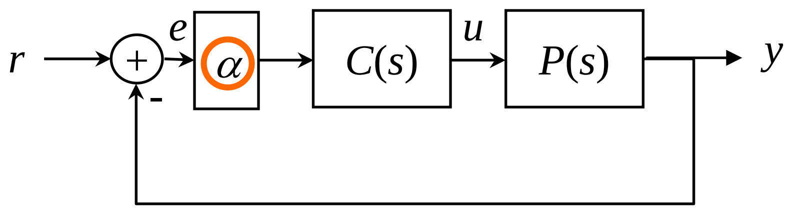
$$G(s) = \frac{a(s)}{b(s)} = k\frac{(s+z_1)(s+z_2)\cdots(s+z_m)}{(s+p_1)(s+p_2)\cdots(s+p_n)}$$

$$\angle G(s_0) = \angle(s_0+z_1) + \cdots + \angle(s_0+z_m) - \angle(s_0+p_1) - \cdots - \angle(s_0+p_n)$$

Trace out positions in plane where phase = 180°

- At each of these points, there exists gain α to satisfy $a(s) + \alpha b(s) = 0$
- All such points are on *root locus*

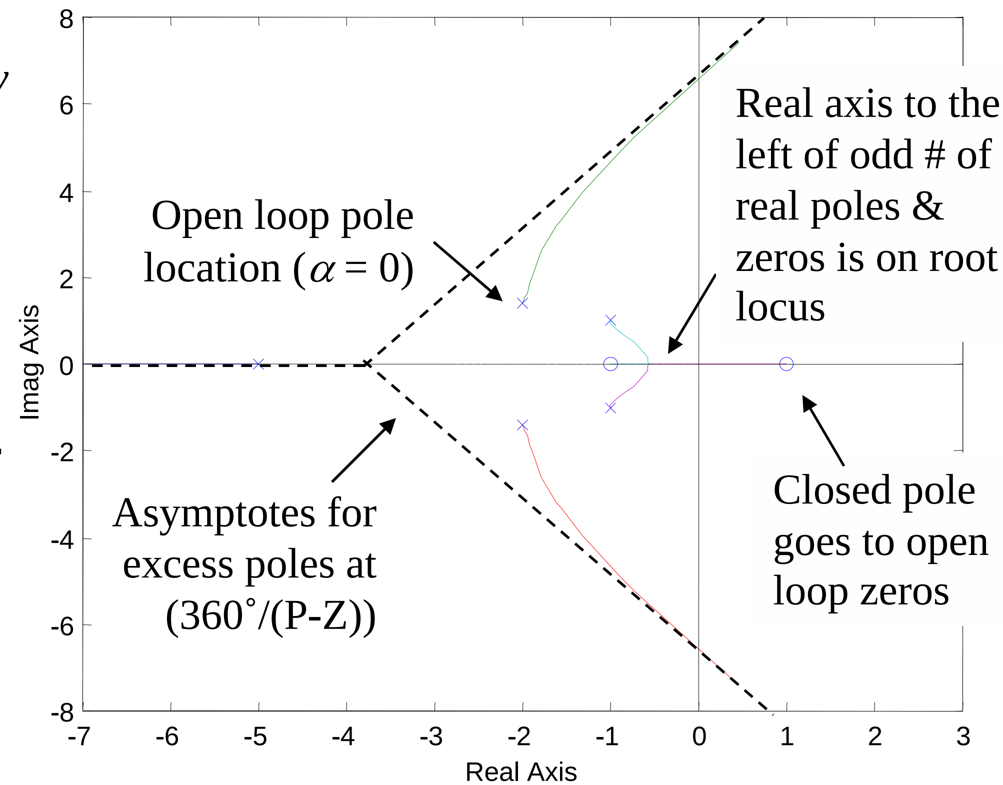
Root Locus for *Loop Gain*



$$1+\alpha\frac{n(s)}{d(s)} \rightarrow d(s)+\alpha n(s)=0$$

Loop gain as root locus parameter

- Common choice for control design
- Special properties for loop gain
  - Roots go from poles of PC to zeros of PC
  - Excess poles go to infinity
  - Can compute asymptotes, break points, etc
- *Very* useful tool for control design
- MATLAB: rlocus



Additional comments

- Although loop gain is the most common parameter, *don't forget* that you can plot roots versus *any* parameter
- Need to link root location to performance…

Second Order System Response

Second order system response

- Spring mass dynamics, written in canonical form

Guidelines for pole placement

- Damping ratio gives Re/Im ratio
- Setting time determined by –Re(λ)

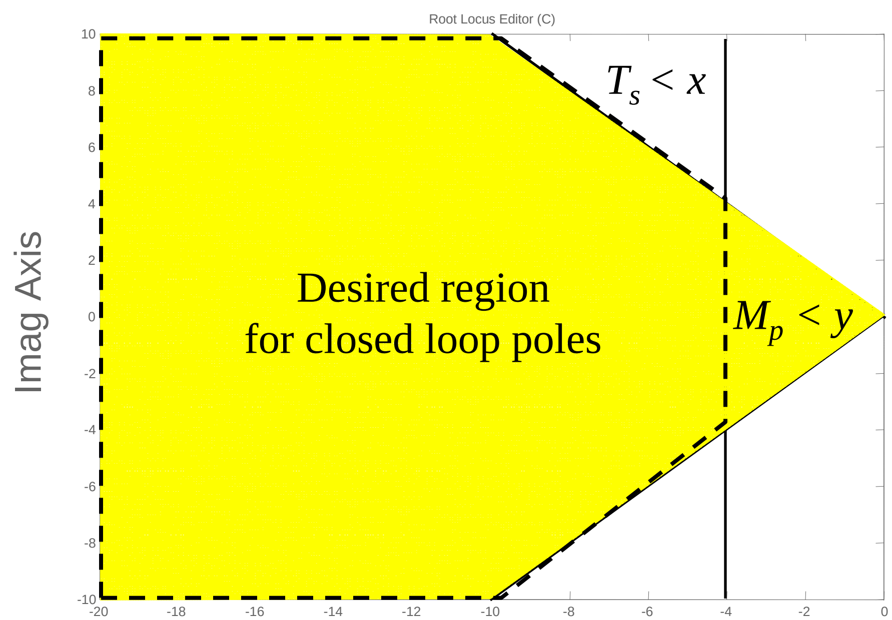
$$H(s) = \frac{\omega_n^2}{s^2 + 2\varsigma\omega_n s + \omega_n^2} = \frac{\omega_n^2}{(s + \varsigma\omega_n + j\omega_d)(s + \varsigma\omega_n - j\omega_d)} \qquad \omega_d = \omega_n\sqrt{1-\varsigma^2}$$

- Performance specifications

$$T_r \approx 1.8/\omega_n \qquad M_p \approx e^{-\pi\varsigma/\sqrt{1-\varsigma^2}}$$

$$T_s \approx 3.9/\varsigma\omega_n \qquad e_{SS} = 0$$

ζ	M_p	Slope
0.707	4%	-1
0.5	16%	-1.7
0.25	44%	-3.9



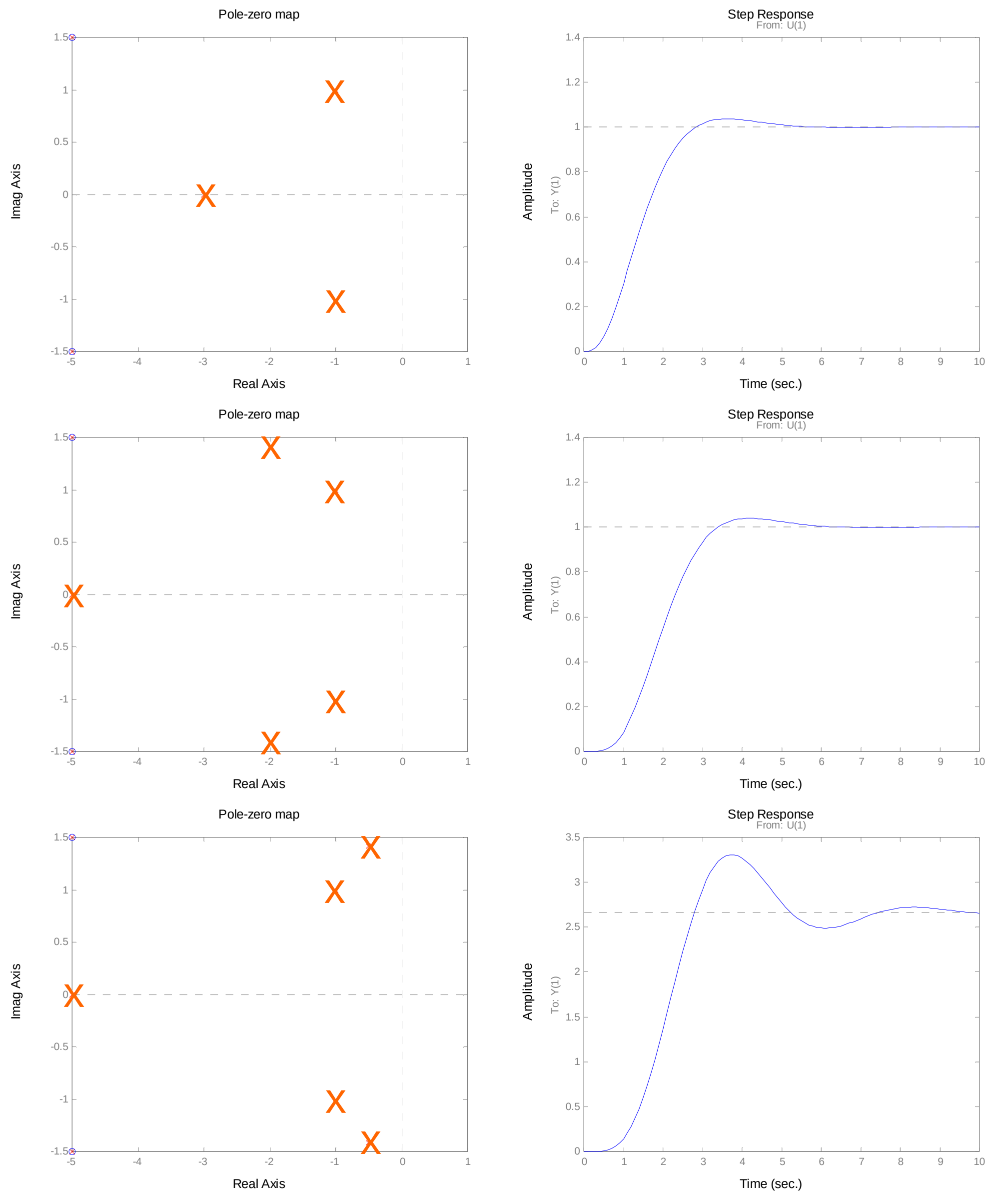
Effect of pole location on performance

Idea: look at "dominant poles"

- Poles nearest the imaginary axis (nearest to instability)
- Analyze using analogy to second order system

PZmap complements information on Bode/Nyquist plots

- Similar to gain and phase calculations
- Shows performance in terms of the *closed* loop poles
- Particularly useful for choosing system gain
- Also useful for deciding where to put controller poles and zeros (with practice [and SISOtool])



Example: PID cruise control

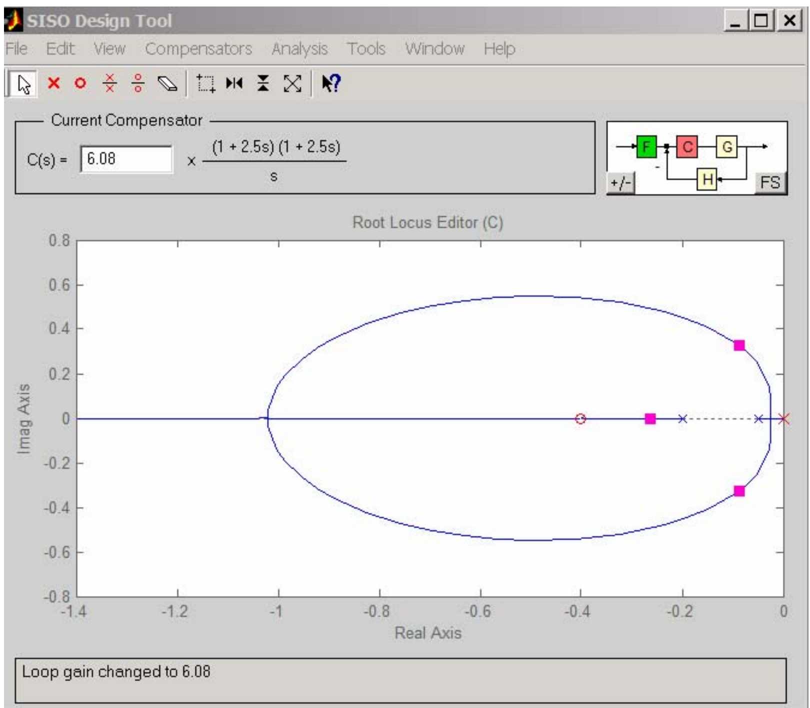
Start with PID control design:

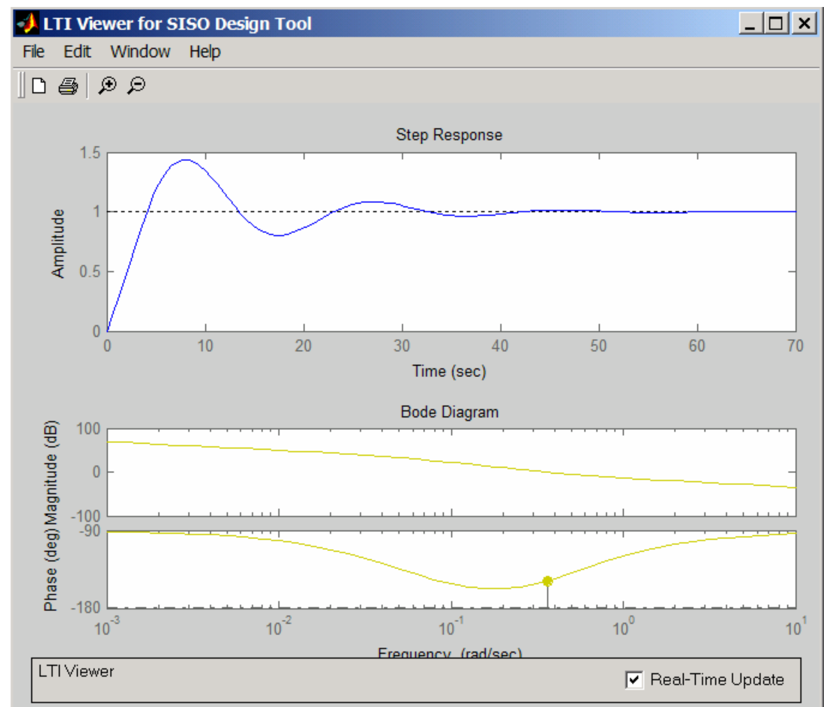
$$P(s) = \frac{1/m}{s + b/m} \cdot \frac{r}{s+a}$$

$$C(s) = K\left(1 + \frac{1}{T_I s} + T_D s\right)$$

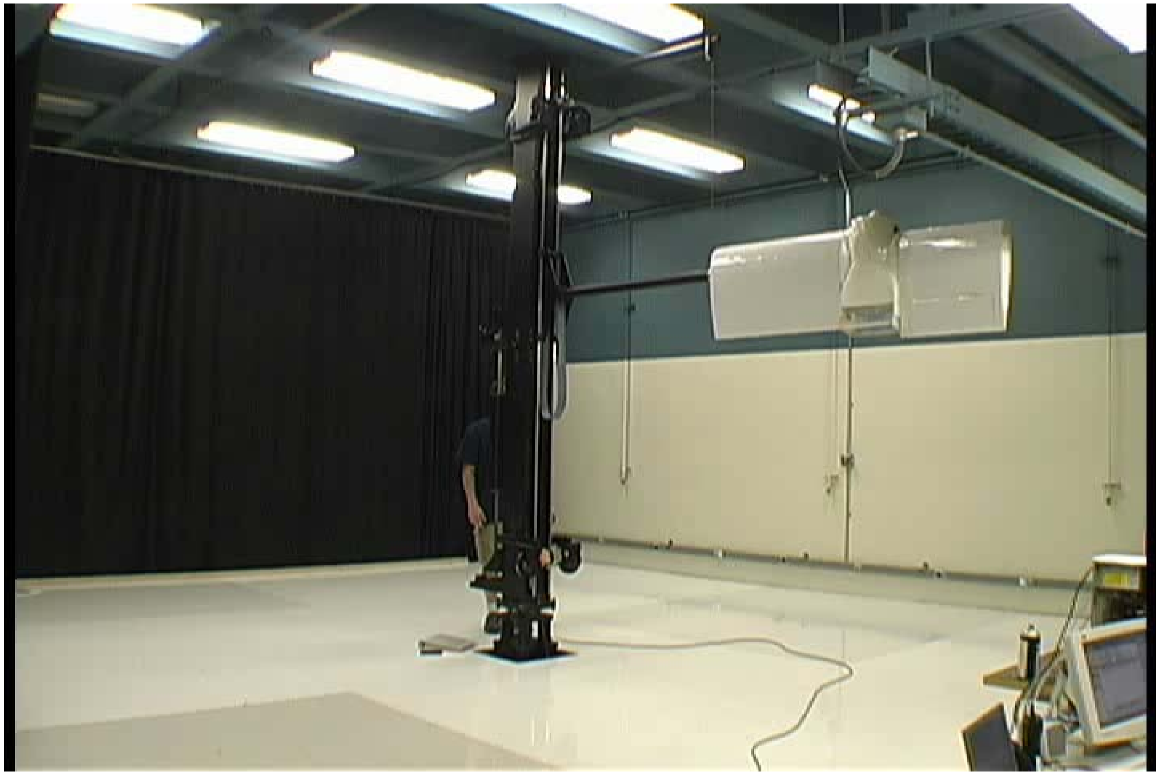
Modify gain to improve performance

- Use MATLAB sisotool
- Adjust loop gain (*K*) to reduce overshoot and decrease settling time
  - $\zeta \approx 1 \Rightarrow$ less than 5% overshoot
  - Re(p) < -0.5 $\Rightarrow T_s$ less than 2 sec





Example: Pitch Control for Caltech Ducted Fan



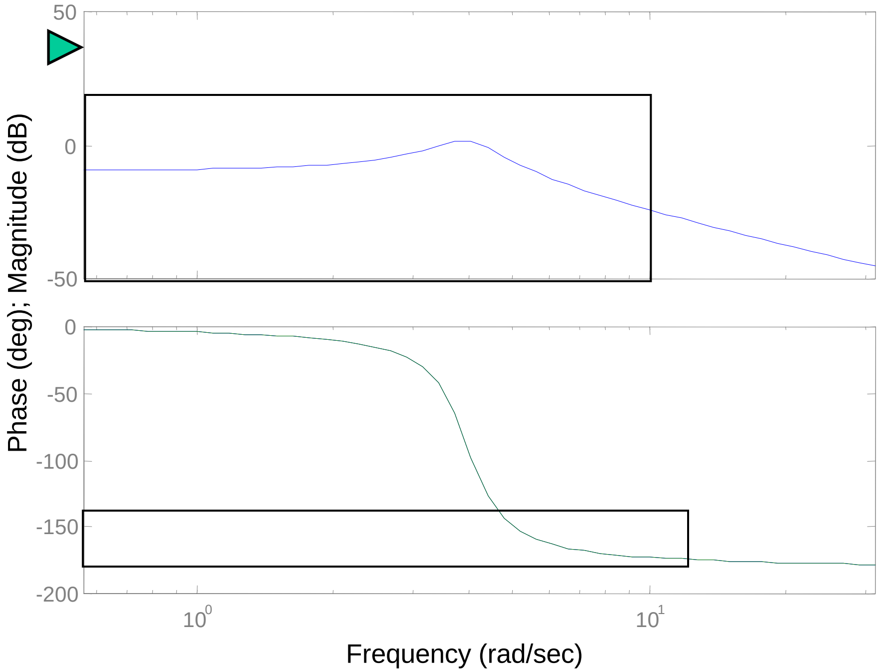
System description

- Vector thrust engine attached to wing
- Inputs: fan thrust, thrust angle (vectored)
- Outputs: position and orientation
- States: x, y, θ + derivatives
- Dynamics: flight aerodynamics

Control approach

- Design "inner loop" control law to regulate pitch (θ) using thrust vectoring
- Second "outer loop" controller regulates the position and altitude by commanding the pitch and thrust
- Basically the same approach as aircraft control laws

Performance Specification and Design Approach



Performance Specification

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

$$P(s) = \frac{r}{Js^2 + ds + mgl}$$

Design approach

- Open loop plant has poor phase margin
- Add phase lead in 5-50 rad/sec range
- Increase the gain to achieve steady state and tracking performance specs
- Avoid integrator to minimize phase

$$C(s) = K\frac{s+a}{s+b}$$

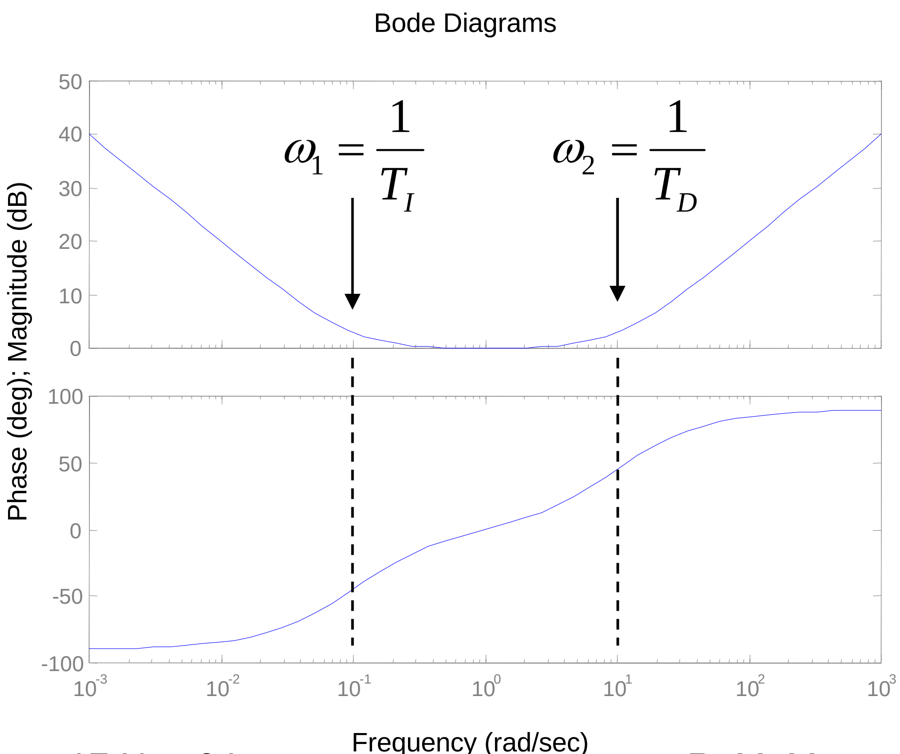
$$a = 25$$
$$b = 300$$
$$K = 15 \cdot 300$$

Summary: PID and Root Locus

PID control design

- Very common (and classical) control technique
- Good tools for choosing gains

$$u = K_p e + K_I \int e + K_D \dot{e}$$



Root locus

- Show closed loop poles as function of a free parameter

Performance limits

- RHP poles and zeros place limits on achievable performance
- Waterbed effect