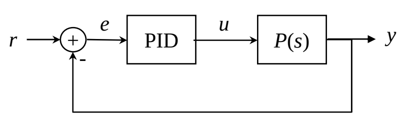
CDS 101: Lecture 9.2
PID and Root Locus

Richard M. Murray
26 November 2003

Goals:
- Define PID controllers and describe how to use them
- Describe root locus diagram and show how to use it to choose loop gain

Reading:
- Astrom, Sec 6.1-6.4, 6.6
- *Optional*: PPH, Sec 13
- *Advanced*: Lewis, Chapter 12 + Sec 13.1

Overview: PID control



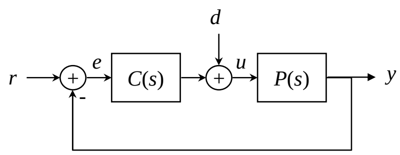
$$u = K_p e + K_I \int e + K_D \dot{e}$$

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

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”
- Also realized that “small deviations” (linearization) could be used to understand the (nonlinear) system dynamics under 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 (see reading)

Frequency domain compensation with PID

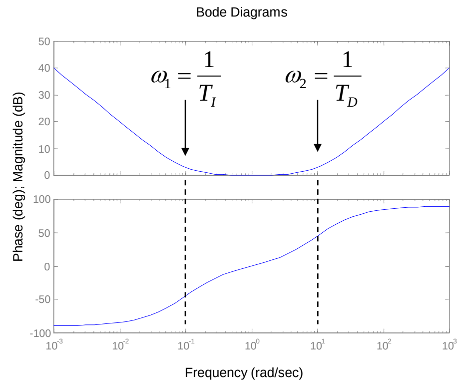


$$C(s) = K_p + K_I \cdot \frac{1}{s} + K_D s$$
$$= k\left(1 + \frac{1}{T_I s} + T_D s\right)$$
$$= \frac{kT_D}{T_I} \frac{(s + 1/T_I)(s + 1/T_D)}{s}$$

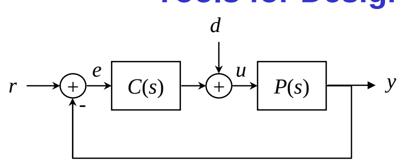
Transfer function for PID controller

$$u = K_p e + K_I \int e + K_D \dot{e}$$
$$\downarrow$$
$$H_{ue}(s) = K_p + K_I \cdot \frac{1}{s} + K_D s$$

- Roughly equivalent to a PI controller with lead compensation
- Idea: gives high gain at low frequency plus phase lead at high frequency
- Place below desired crossover freq



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

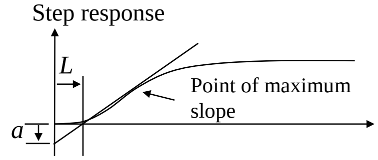
Caution: PID amplifies high frequency noise

- Sol'n: pole at high frequency

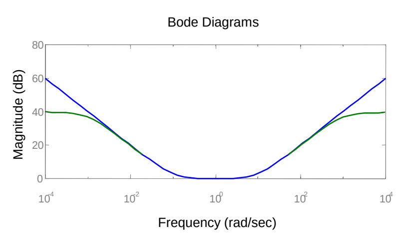
Caution: Integrator windup

- 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

Step response



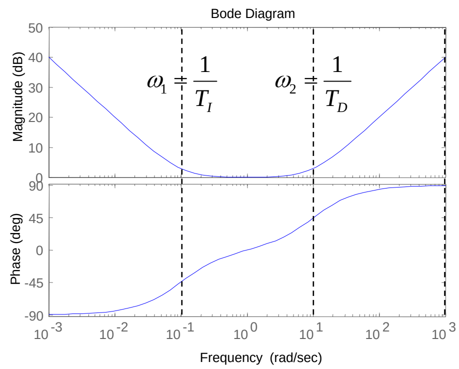
$$K = 1.2/a \quad T_I = 2 * L \quad T_D = L/2$$



PID vs lead/lag compensation

PID Control

$$C(s) = K_p + K_I \cdot \frac{1}{s} + K_D s$$

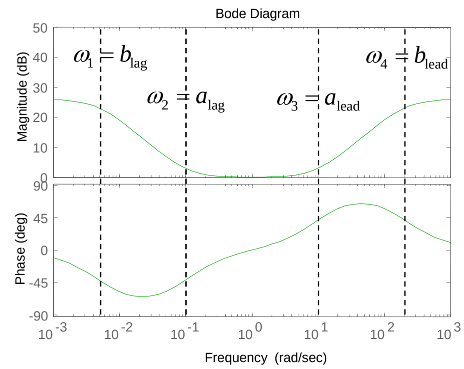


Pros: easy to design, implement

Cons: low freq lag, high freq gain

Lead/Lag Compensation

$$C(s) = K \frac{s + a_{\text{lag}}}{s + b_{\text{lag}}} \cdot \frac{s + a_{\text{lead}}}{s + b_{\text{lead}}}$$

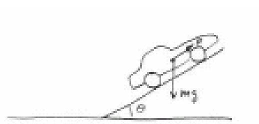


Pros: low freq phase, high freq rolloff

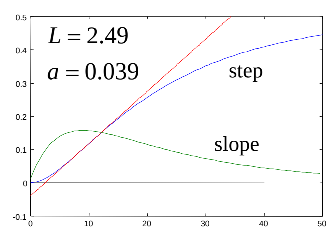
Cons: more complicated (slightly)

26 Nov 03 R. M. Murray, Caltech CDS 5

Example: PID cruise control

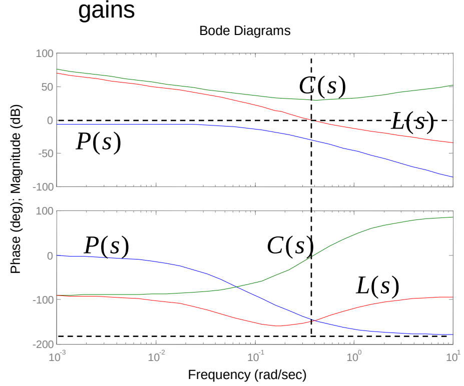


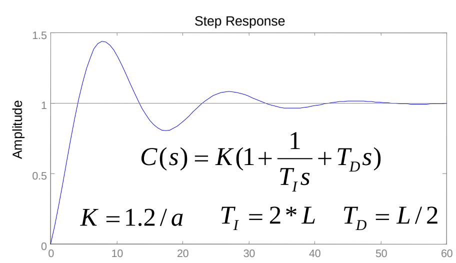
$$P(s) = \frac{1/m}{s + b/m} \cdot \frac{r}{s + a}$$



Ziegler-Nichols design for cruise controller

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

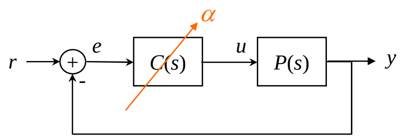


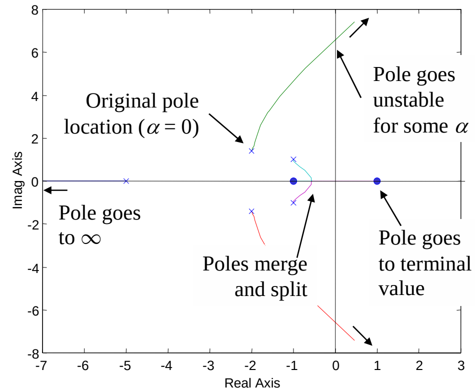


- Result: *sluggish* ⇒ increase loop gain

26 Nov 03 R. M. Murray, Caltech CDS 6

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:

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

α

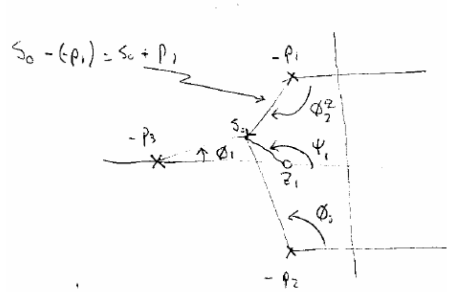
Root locus = locus of roots as parameter value is changed

- 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



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

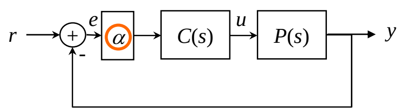
$$\angle G(s_0) = \angle(s_0 + z_1) + \cdots + \angle(s_0 + z_m) - \angle(s_0 + p_1) - \cdots - \angle(s_0 + p_n)$$

ϕ_i = phase contribution from s_0 to $-p_i$
ψ_i = phase contribution from s_0 to $-z_i$

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*

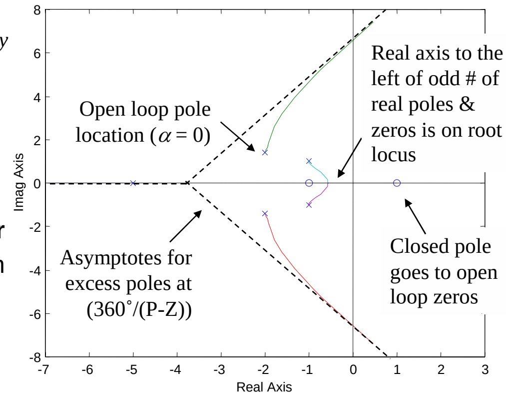
Root Locus for *Loop Gain*



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

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)

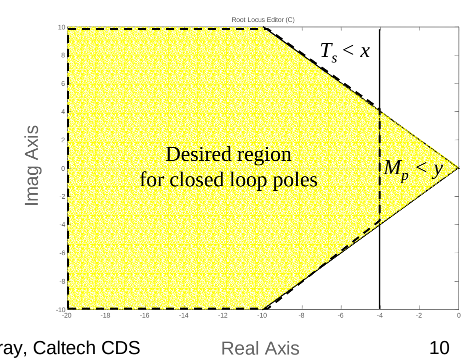
$$H(s)=\frac{\omega_n^2}{s^2+2\varsigma\omega_n s+\omega_n^2}=\frac{\omega_n^2}{(s+\varsigma\omega_n+j\omega_d)(s+\varsigma\omega_n-j\omega_d)} \qquad \omega_d=\omega_n\sqrt{1-\varsigma^2}$$

- Performance specifications

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

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

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



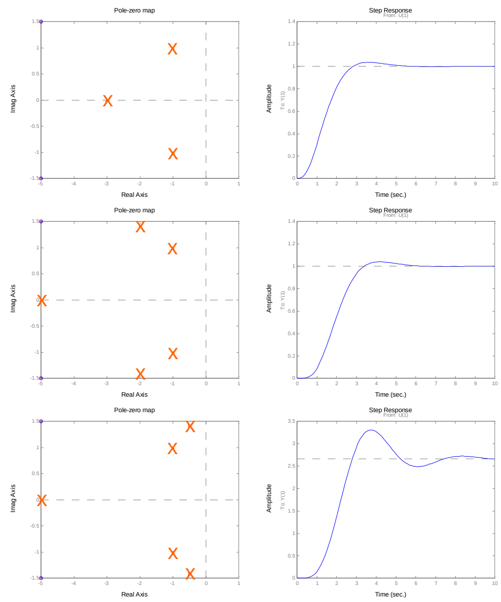
Effect of pole location on performance

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])



26 Nov 03 R. M. Murray, Caltech CDS 11

Example: PID cruise control

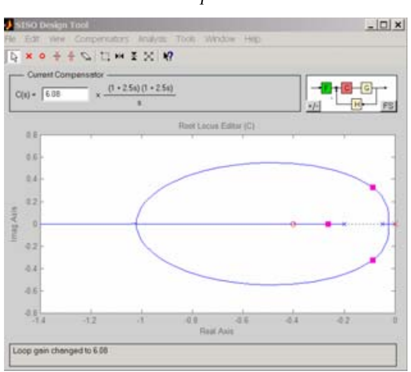
Start with PID control design:

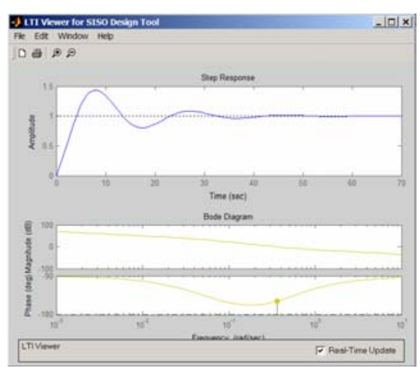
$$P(s) = \frac{1/m}{s + b/m} \cdot \frac{r}{s+a}$$

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

Modify gain to improve performance

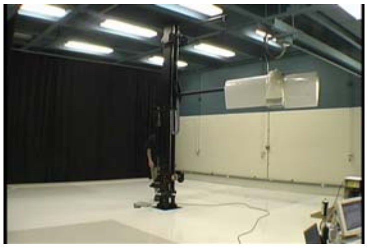
- Use MATLAB sisotool
- Adjust loop gain (K) to reduce overshoot and decrease settling time
  - $\zeta \approx 1 \Rightarrow$ less than 5% overshoot
  - Re(p) < -0.5 $\Rightarrow T_s$ less than 2 sec

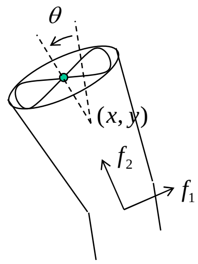




26 Nov 03 R. M. Murray, Caltech CDS 12

Example: Ducted fan lateral position control

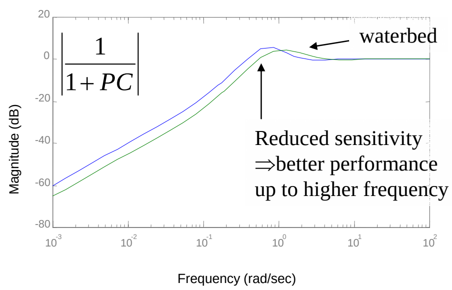


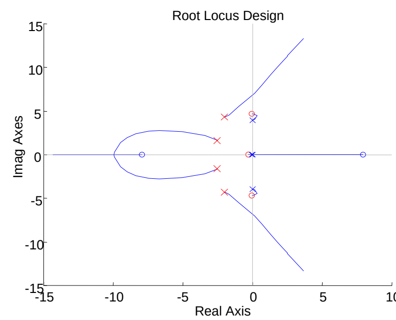


$$P(s) = \frac{(s^2 - mgl)}{s^2(Js^2 + ds + mgl)}$$

Lateral control (*x*)

- Right half plane zero makes design *very* tricky using y as output



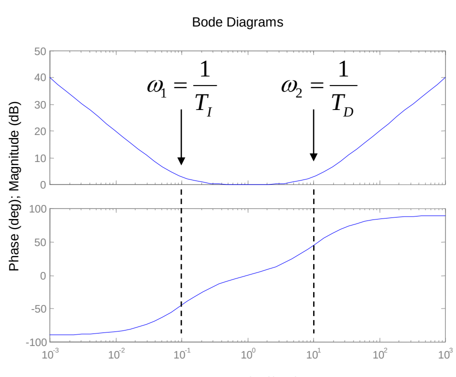


Summary: PID and Root Locus

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