Topic #2

16.30/31 Feedback Control Systems

Basic Root Locus

- Basic aircraft control concepts
- Basic control approaches

Aircraft Longitudinal Control

- Consider the short period approximate model of an 747 aircraft.

$$\dot{x}_{sp} = A_{sp} x_{sp} + B_{sp} \delta_e$$

where δ_e is the elevator input, and

$$x_{sp} = \begin{bmatrix} w \\ q \end{bmatrix} \quad , \quad A_{sp} = \begin{bmatrix} Z_w/m & U_0 \\ I_{yy}^{-1}\left(M_w + M_{\dot{w}} Z_w/m\right) & I_{yy}^{-1}\left(M_q + M_{\dot{w}} U_0\right) \end{bmatrix}$$

$$B_{sp} = \begin{bmatrix} Z_{\delta_e}/m \\ I_{yy}^{-1}\left(M_{\delta_e} + M_{\dot{w}} Z_{\delta_e}/m\right) \end{bmatrix}$$

- Add that $\dot{\theta} = q$, so $s\theta = q$

- Take the output as θ, input is δ_e, then form the transfer function[1]

$$\frac{\theta(s)}{\delta_e(s)} = \frac{1}{s}\frac{q(s)}{\delta_e(s)} = \frac{1}{s}\begin{bmatrix} 0 & 1 \end{bmatrix}(sI - A_{sp})^{-1} B_{sp}$$

- For the 747 (40Kft, $\mathbf{M} = 0.8$) this reduces to:

$$\frac{\theta(s)}{\delta_e(s)} = -\frac{1.1569s + 0.3435}{s(s^2 + 0.7410s + 0.9272)} \equiv G_{\theta\delta_e}(s)$$

so that the dominant roots have a frequency of approximately 1 rad/sec and damping of about 0.4

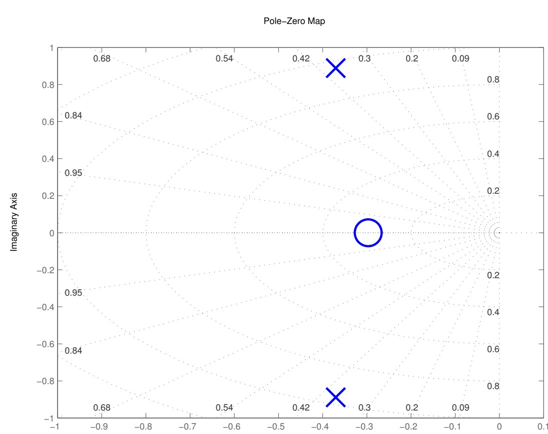


Fig. 1: Note - this is the Pole-zero map for $G_{q\delta_e}$

[1] Much more on how to do this part later

- Basic problem is that there are vast quantities of empirical data to show that pilots do not like the flying qualities of an aircraft with this combination of frequency and damping
  - What is preferred?

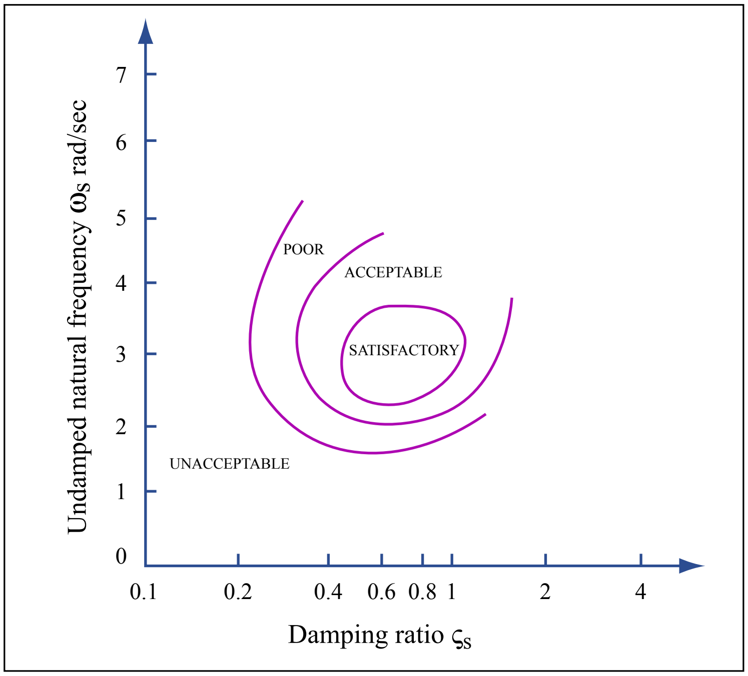


Image by MIT OpenCourseWare.

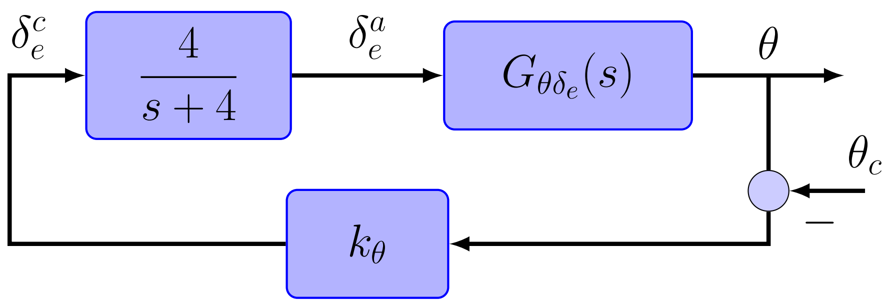
Fig. 2: "Thumb Print" criterion

- This criterion was developed in 1950's, and more recent data is provided in MILSPEC8785C
  - Based on this plot, a good target: frequency ≈ 3 rad/sec and damping of about ≈ 0.6

- Problem is that the short period dynamics are no where near these numbers, so we must modify them.
  - Could do it by redesigning the aircraft, but it is a bit late for that...

First Short Period Autopilot

- First attempt to control the vehicle response: measure θ and feed it back to the elevator command δ_e.
  - Unfortunately the actuator is slow, so there is an apparent lag in the response that we must model



- Dynamics: δ_e^a is the actual elevator deflection, δ_e^c is the actuator command created by our controller

$$\theta = G_{\theta\delta_e}(s)\delta_e^a; \quad \delta_e^a = H(s)\delta_e^c; \quad H(s) = \frac{4}{s+4}$$

The control is just basic proportional feedback

$$\delta_e^c = -k_\theta(\theta - \theta_c)$$

which gives that

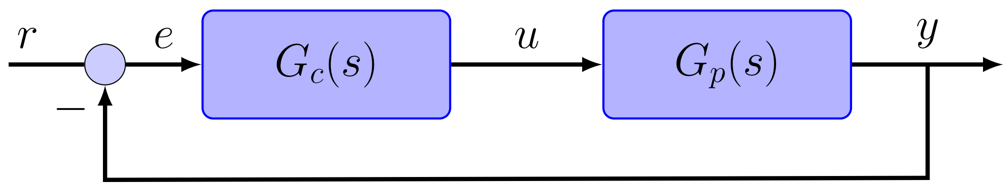
$$\theta = -G_{\theta\delta_e}(s)H(s)k_\theta(\theta - \theta_c)$$

or that

$$\frac{\theta(s)}{\theta_c(s)} = \frac{G_{\theta\delta_e}(s)H(s)k_\theta}{1 + G_{\theta\delta_e}(s)H(s)k_\theta}$$

- Looks good, but how do we analyze what is going on?
  - Need to be able to predict where the poles are going as a function of $k_\theta \Rightarrow$ **Root Locus**

Root Locus Basics



- Assume that the plant transfer function is of the form

$$G_p = K_p \frac{N_p}{D_p} = K_p \frac{\prod_i^{n_{pz}} (s - z_{pi})}{\prod_i^{n_{pp}} (s - p_{pi})}$$

and the controller transfer function is

$$G_c(s) = K_c \frac{N_c}{D_c} = K_c \frac{\prod_i^{n_{cz}} (s - z_{ci})}{\prod_i^{n_{cp}} (s - p_{ci})}$$

  - Assume that $n_{pp} > n_{pz}$ and $n_{cp} > n_{cz}$

[2]

- Signals are:

u	control commands
y	output/measurements
r	reference input
e	response error

- Unity feedback form. We could add the controller G_c in the feedback path without changing the pole locations.
- Will discuss performance and add disturbances later, but for now just focus on the pole locations

[2]Errata: Added n values for the number of poles and zeros

- Basic questions:
  - **Analysis**: Given N_c and D_c, where do the closed loop poles go as a function of K_c?

  - **Synthesis**: Given K_p, N_p and D_p, how should we chose K_c, N_c, D_c to put the closed loop poles in the desired locations?

- **Block diagram analysis**: Since $y = G_pG_ce$ and $e = r - y$, then easy to show that

$$\frac{y}{r} = \frac{G_cG_p}{1 + G_cG_p} \equiv G_{cl}(s)$$

where

$$G_{cl}(s) = \frac{K_cK_pN_cN_p}{D_cD_p + K_cK_pN_cN_p}$$

is the **closed loop transfer function**

  - Denominator called **characteristic equation** $\phi_c(s)$ and the roots of $\phi_c(s) = 0$ are called the **closed-loop poles** (CLP).

  - The CLP are clearly functions of K_c for a given K_p, N_p, D_p, N_c, D_c $\Rightarrow$ a "locus of roots" [Evans, 1948]

Root Locus Analysis

- General root locus is hard to determine by hand and requires Matlab tools such as `rlocus(num,den)` to obtain full result, but we can get some important insights by developing a short set of plotting rules.
  - Full rules in FPE, page 279 (4th edition).
- Basic questions:
  1. What points are on the root locus?
  2. Where does the root locus start?
  3. Where does the root locus end?
  4. When/where is the locus on the real line?
  5. Given that s_0 is found to be on the locus, what gain is need for that to become the closed-loop pole location?
  6. What are the departure and arrival angles?
  7. Where are the multiple points on the locus?

- **Question #1:** is point s_0 on the root locus? Assume that N_c and D_c are known, let

$$L_d = \frac{N_c}{D_c}\frac{N_p}{D_p} \quad \text{and } K = K_c K_p$$

$$\Rightarrow \phi_c(s) = 1 + K L_d(s) = 0$$

So values of s for which $L_d(s) = -1/K$, with K real are on the RL.

  - For K positive, s_0 is on the root locus if

$$\measuredangle L_d(s_0) = 180^\circ \pm l \cdot 360^\circ, \quad l = 0, 1, \ldots$$

  - If K negative, s_0 is on the root locus if [0° locus]

$$\measuredangle L_d(s_0) = 0^\circ \pm l \cdot 360^\circ, \quad l = 0, 1, \ldots$$

These are known as the **phase conditions.**

- **Question #2:** Where does the root locus start?

$$\phi_c = 1 + K\frac{N_c N_p}{D_c D_p} = 0$$

$$\Rightarrow D_c D_p + K N_c N_p = 0$$

So if $K \to 0$, then locus starts at solutions of $D_c D_p = 0$ which are the **poles** of the plant and compensator.

- **Question #3:** Where does the root locus end?

Already shown that for s_0 to be on the locus, must have

$$L_d(s_0) = -\frac{1}{K}$$

So if $K \to \infty$, the poles must satisfy:

$$L_d = \frac{N_c N_p}{D_c D_p} = 0$$

- There are several possibilities:

1. Poles are located at values of s for which $N_c N_p = 0$, which are the **zeros** of the plant and the compensator

2. If Loop $L_d(s)$ has more poles than zeros
   - As $|s| \to \infty$, $|L_d(s)| \to 0$, but we must ensure that the phase condition is still satisfied.

- More details as $K \to \infty$:
  - Assume there are n zeros and p poles of $L_d(s)$
  - Then for large $|s|$,

$$L_d(s) \approx \frac{1}{(s-\alpha)^{p-n}}$$

  - So the root locus degenerates to:

$$1 + \frac{1}{(s-\alpha)^{p-n}} = 0$$

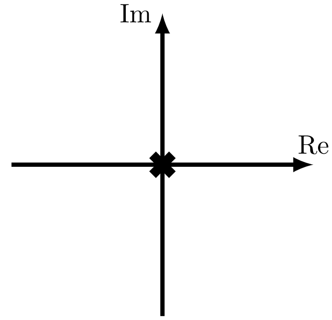
  - So n poles head to the zeros of $L_d(s)$
  - Remaining $p-n$ poles head to $|s| = \infty$ along **asymptotes** defined by the radial lines

$$\phi_l = \frac{180^\circ + 360^\circ \cdot (l-1)}{p-n} \quad l = 1, 2, \ldots$$

    so that the number of asymptotes is governed by the number of poles compared to the number of zeros (relative degree).
  - If z_i are the zeros if L_d and p_j are the poles, then the **centroid of the asymptotes** is given by:

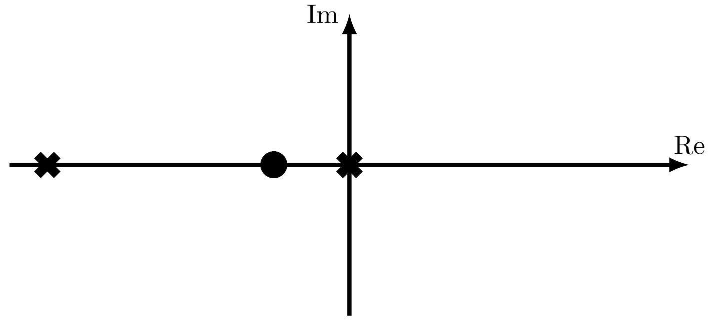
$$\alpha = \frac{\sum^p p_j - \sum^n z_i}{p-n}$$

- Example: $L(s) = s^{-4}$



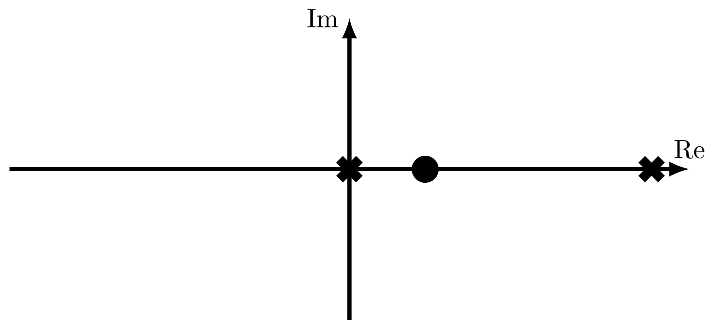
  - Number of asymptotes and α?

- Example $G(s) = \dfrac{s+1}{s^2(s+4)}$



  - Number of asymptotes and α?

- Example $G(s) = \dfrac{s-1}{s^2(s-4)}$



  - Number of asymptotes and α?

- **Question #4:** When/where is the locus on the real line?
  - Locus points on the real line are to the **left** of an **odd number** of real axis poles and zeros [K positive].
  - Follows from the phase condition and the fact that the phase contribution of the complex poles/zeros cancels out

- **Question #5:** Given that s_0 is found to be on the locus, what gain is needed for that to become the closed-loop pole location?
  - Need

$$K \equiv \frac{1}{|L_d(s_0)|} = \left| \frac{D_p(s_0)D_c(s_0)}{N_p(s_0)N_c(s_0)} \right|$$

  - Since $K = K_p K_c$, sign of K_c depends on sign of K_p
    - e.g., assume that $\measuredangle L_d(s_0) = 180°$, then need K_c and K_p to be same sign so that $K > 0$

Root Locus Examples

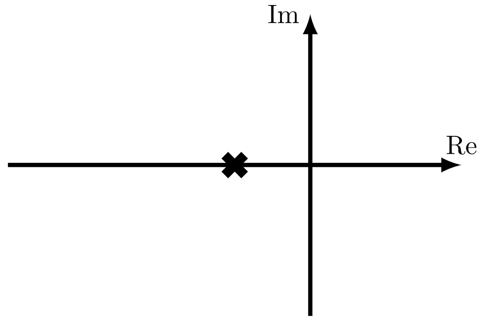


Fig. 3: Basic

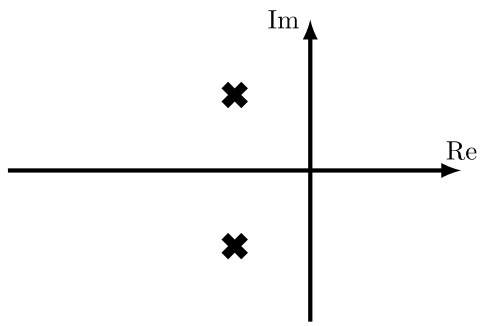


Fig. 4: Two poles

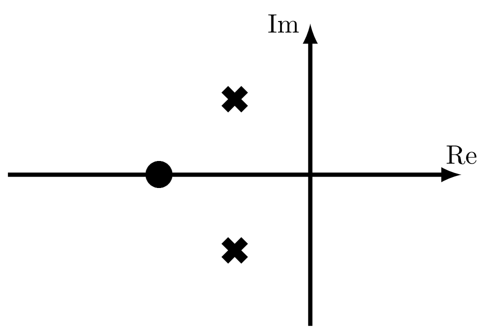


Fig. 5: Add zero

- Examples similar to control design process: add compensator dynamics to modify root locus and then chose gain to place CLP at desired location on the locus.

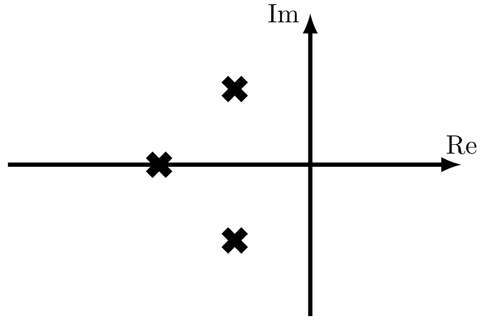


Fig. 6: Three poles

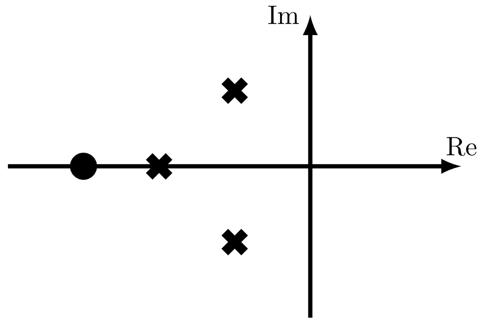


Fig. 7: Add a zero again

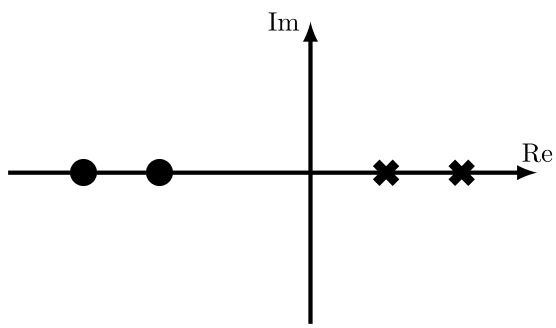


Fig. 8: Complex Case

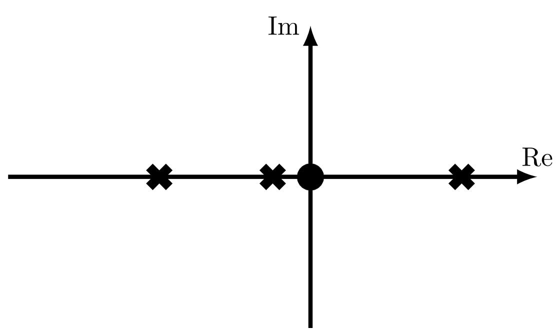


Fig. 9: Very Complex Case

Performance Issues

- Interested in knowing how well our closed loop system can track various inputs
  - Steps, ramps, parabolas
  - Both transient and steady state

- For perfect steady state tracking want error to approach zero

$$\lim_{t\to\infty} e(t) = 0$$

  - Can determine this using the closed-loop transfer function and the final value theorem

$$\lim_{t\to\infty} e(t) = \lim_{s\to 0} se(s)$$

- So for a step input $r(t) = \mathbf{1}(t) \to r(s) = 1/s$

$$\frac{y(s)}{r(s)} = \frac{G_c(s)G_p(s)}{1+G_c(s)G_p(s)} \quad \frac{y(s)}{e(s)} = G_c(s)G_p(s)$$

$$\frac{e(s)}{r(s)} = \frac{1}{1+G_c(s)G_p(s)}$$

so in the case of a step input, we have

$$e(s) = \frac{r(s)}{1+G_c(s)G_p(s)} = \frac{1/s}{1+G_c(s)G_p(s)}$$

$$\Rightarrow \lim_{s\to 0} se(s) = \lim_{s\to 0} s\frac{1/s}{1+G_c(s)G_p(s)} = \frac{1}{1+G_c(0)G_p(0)} \equiv e(\infty)$$

- So the steady state error to a step is given by

$$e_{ss} = \frac{1}{1+G_c(0)G_p(0)}$$

  - To make the error small, we need to make one (or both) of $G_c(0)$, $G_p(0)$ very large

- Clearly if $G_p(s)$ has a free integrator (or two) so that it resembles $\frac{1}{s^n(s+\alpha)^m}$ with $n \geq 1$, then

$$\lim_{s\to 0} G_p(s) \to \infty \quad \Rightarrow \quad e_{ss} \to 0$$

- Can continue this discussion by looking at various input types (step, ramp, parabola) with systems that have a different number of free integrators (type), but the summary is this:

	step	ramp	parabola
type 0	$\dfrac{1}{1+K_p}$	∞	∞
type 1	0	$\dfrac{1}{K_v}$	∞
type 2	0	0	$\dfrac{1}{K_a}$

where

$$K_p = \lim_{s\to 0} G_c(s)G_p(s) \qquad \text{Position Error Constant}$$

$$K_v = \lim_{s\to 0} sG_c(s)G_p(s) \qquad \text{Velocity Error Constant}$$

$$K_a = \lim_{s\to 0} s^2G_c(s)G_p(s) \qquad \text{Acceleration Error Constant}$$

which are a good simple way to keep track of how well your system is doing in terms of steady state tracking performance.

Dynamic Compensation

- For a given plant, can draw a root locus versus K. But if desired pole locations are not on that locus, then need to modify it using **dynamic compensation**.
  - Basic root locus plots give us an indication of the effect of adding compensator dynamics. But need to know what to add to place the poles where we want them.
- New questions:
  - What type of compensation is required?
  - How do we determine where to put the additional dynamics?
- There are three classic types of controllers $u = G_c(s)e$

1. **Proportional feedback:** $G_c \equiv K_g$ a gain, so that $N_c = D_c = 1$
   - Same case we have been looking at.

2. **Integral feedback:**

$$u(t) = K_i \int_0^t e(\tau)d\tau \Rightarrow G_c(s) = \frac{K_i}{s}$$

   - Used to reduce/eliminate steady-state error
   - If $e(\tau)$ is approximately constant, then $u(t)$ will grow to be very large and thus hopefully correct the error.

- Consider error response of $G_p(s) = 1/(s+a)(s+b)$ $(a > 0,\ b > 0)$ to a step,

$$r(t) = \mathbf{1}(t) \rightarrow r(s) = 1/s$$

where

$$\frac{e}{r} = \frac{1}{1 + G_c G_p} = S(s) \quad \rightarrow \quad e(s) = \frac{r(s)}{(1 + G_c G_p)}$$

- where $S(s)$ is the **Sensitivity Transfer Function** for the closed-loop system

- To analyze error, use FVT $\lim_{t\to\infty} e(t) = \lim_{s\to 0} se(s)$ so that with proportional control,

$$\lim_{t\to\infty} e_{ss} = \lim_{s\to 0} \left(\frac{s}{s}\right) \frac{1}{1 + K_g G_p(s)} = \frac{1}{1 + \frac{K_g}{ab}}$$

so can make e_{ss} small, but only with a very large K_g

- With integral control, $\lim_{s\to 0} G_c(s) = \infty$, so $e_{ss} \rightarrow 0$

- Integral control **improves the steady state,** but this is at the **expense of the transient response**
  * Typically gets worse because the system is less well damped

Example #1: $G(s) = \dfrac{1}{(s+a)(s+b)}$, add integral feedback to improve the steady state response.

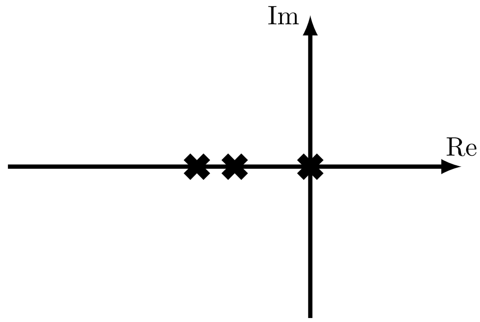


Fig. 10: RL after adding integral FB

- Increasing K_i to increase speed of the response pushes the poles towards the imaginary axis $\rightarrow$ more oscillatory response.

Now combine **proportional and integral** (PI) feedback:

$$G_c = K_1 + \frac{K_2}{s} = \frac{K_1 s + K_2}{s}$$

which introduces a pole at the origin and zero at $s = -K_2/K_1$

- PI solves many of the problems with just integral control

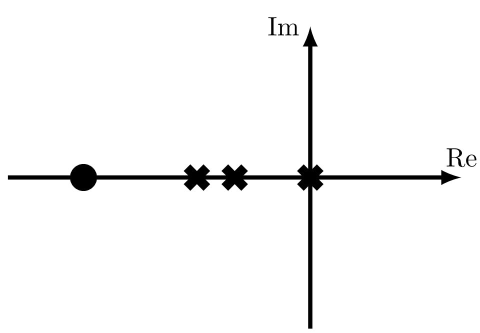


Fig. 11: RL with proportional and integral FB

3. **Derivative Feedback:** $u = K_d \dot{e}$ so that $G_c(s) = K_d s$
    - Does not help with the steady state
    - Provides feedback on the rate of change of $e(t)$ so that the control can anticipate future errors.

Example # 2: $G(s) = \dfrac{1}{(s-a)(s-b)}$, $(a > 0, b > 0)$
with $G_c(s) = K_d s$

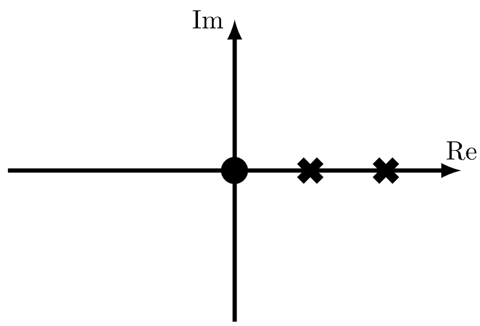


Fig. 12: RL with derivative FB

- Derivative feedback is very useful for pulling the root locus into the LHP - increases damping and more stable response.

Typically used in combination with proportional feedback to form proportional-derivative feedback **PD**

$$G_c(s) = K_1 + K_2 s$$

which moves the zero from the origin.

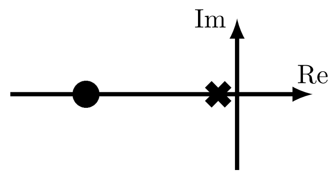
- Unfortunately pure PD is not realizable in the lab as pure differentiation of a measured signal is typically a bad idea
    - Typically use band-limited differentiation instead, by rolling-off the PD control with a high-frequency pole (or two).

Controller Synthesis

- First determine where the poles should be located
- Will proportional feedback do the job?
- What types of dynamics need to be added? Use main building block

$$G_B(s) = K_c \frac{(s+z)}{(s+p)}$$

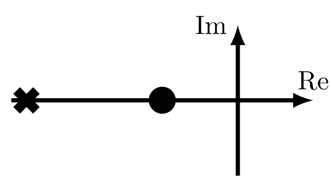
- Looks like various controllers, depending how K_c, p, and z picked
  - If pick $z > p$, with p small, then



$$G_B(s) \approx K_c \frac{(s+z)}{s}$$

which is essentially a PI compensator, called a **lag.**

  - If pick $p \gg z$, then at low frequency, the impact of $p/(s+p)$ is small, so



$$G_B(s) \approx K_c(s+z)$$

which is essentially PD compensator, called a **lead.**

- Various algorithms exist to design the components of the lead and lag compensators

Classic Root Locus Approach

- Consider a simple system $G_p = s^{-2}$ for which we want the closed loop poles to be at $-1 \pm 2\mathbf{j}$
- Will proportional control be sufficient? no
- So use compensator with 1 pole.

$$G_c = K\frac{(s+z)}{(s+p)}$$

So there are 3 CLP.

  - To determine how to pick the p, z, and k, we must use the phase and magnitude conditions of the RL

- To proceed, evaluate the phase of the loop

$$L_d(s) = \frac{s+z}{(s+p)s^2}$$

at $s_0 = -1 + 2\mathbf{j}$. Since we want s_0 to be on the new locus, we know that $\measuredangle L_d(s_0) = 180° \pm 360° l$

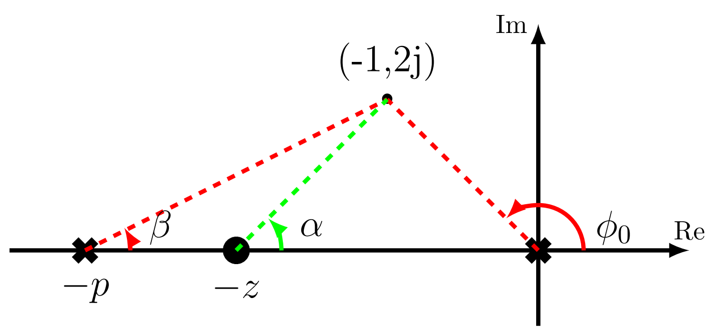


Fig. 13: Phase Condition

  - As shown in the figure, there are four terms in $\measuredangle L_d(s_0)$ – the two poles at the origin contribute 117° each
  - Given the assumed location of the compensator pole/zero, can work out their contribution as well

- Geometry for the real zero: $\tan\alpha = \frac{2}{z-1}$ and for the real pole: $\tan\beta = \frac{2}{p-1}$
- Since we expect the zero to be closer to the origin, put it first on the negative real line, and then assume that $p = \gamma z$, where typically $5 \leq \gamma \leq 10$ is a good ratio.

- So the phase condition gives:

$$\begin{aligned} -2(117^\circ) + \alpha - \beta &= 180^\circ \\ \arctan\left(\frac{2}{z-1}\right) - \arctan\left(\frac{2}{10z-1}\right) &= 53^\circ \end{aligned}$$

but recall that

$$\tan(A-B) = \frac{\tan(A) - \tan(B)}{1 + \tan(A)\tan(B)}$$

so

$$\frac{(\frac{2}{z-1}) - (\frac{2}{10z-1})}{1 + (\frac{2}{z-1})(\frac{2}{10z-1})} = 1.33$$

which give $z = 2.2253$, $p = 22.2531$, $k_c = 45.5062$

```
% RL design using angles
clear all
target = -1+2*j;
phi_origin= 180-atan(imag(target)/-real(target))*180/pi;
syms z M; ratio=10;
phi_z=(imag(target)/(z+real(target)));
phi_p=(imag(target)/(ratio*z+real(target)));
M=(phi_z-phi_p)/(1+phi_z*phi_p);
test=solve(M-tan(pi/180*(2*phi_origin-180)));
Z=eval(test(1));
P=ratio*Z;
K=1/abs((target+Z)/(target^2*(target+P)));
[Z P K]
```

Pole Placement

- Another option for simple systems is called pole placement.

- Know that the desired characteristic equation is

$$\phi_d(s) = (s^2 + 2s + 5)(s + \alpha) = 0$$

- Actual closed loop poles solve:

$$\begin{aligned} \phi_c(s) = 1 + G_p G_c &= 0 \\ \rightarrow s^2(s + p) + K(s + z) &= 0 \\ \rightarrow s^3 + s^2 p + Ks + Kz &= 0 \end{aligned}$$

- Clearly need to pull the poles at the origin into the LHP, so need a lead compensator → Rule of thumb:[3] take $p = (5 - 10)z$.

- Compare the characteristic equations:

$$\begin{aligned} \phi_c(s) &= s^2 + 10zs^2 + Ks + Kz = 0 \\ \phi_d(s) &= (s^2 + 2s + 5)(s + \alpha) \\ &= s^3 + s^2(\alpha + 2) + s(2\alpha + 5) + 5\alpha = 0 \end{aligned}$$

gives

$$\begin{array}{c||r} s^2 & \alpha + 2 = 10z \\ s & 2\alpha + 5 = K \\ s^0 & 5\alpha = zK \end{array}$$

solve for α, z, K

$$K = \frac{25}{5 - 2z}; \quad \alpha = \frac{5z}{5 - 2z}$$

$\rightarrow z = 2.23,\ \alpha = 20.25,\ K = 45.5$

[3] Errata: changed the rule of the pole zero ratio.

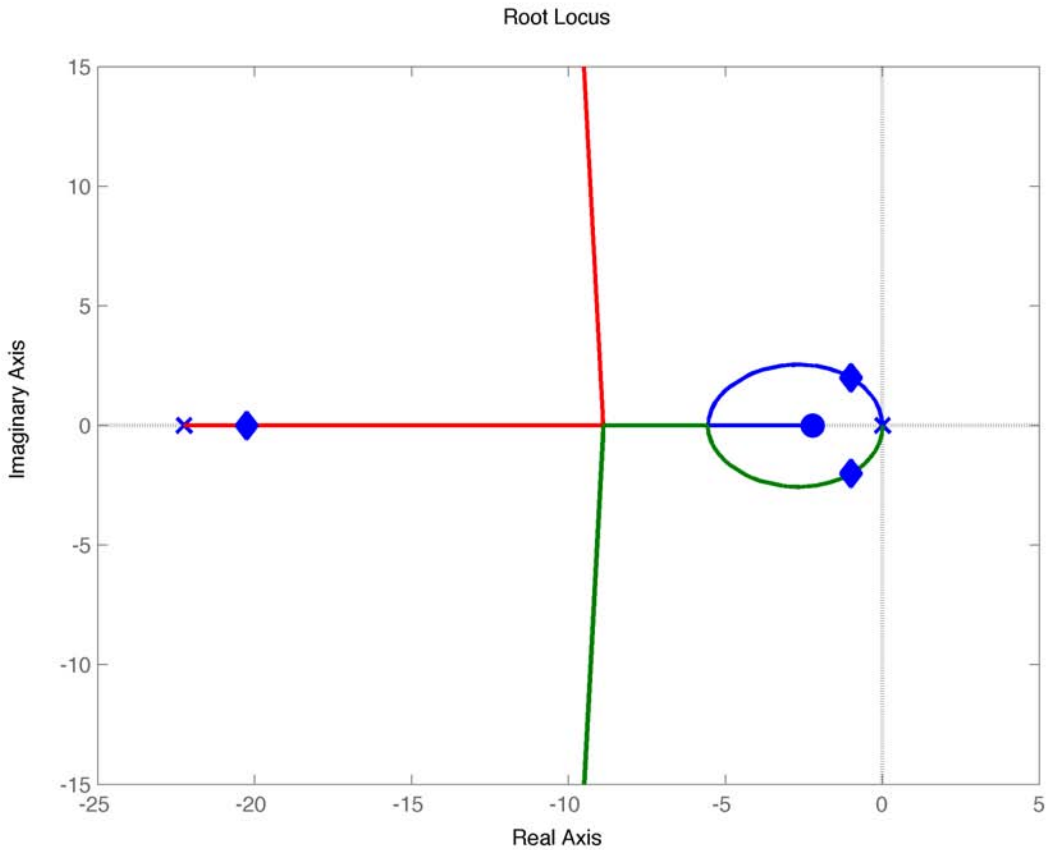


Fig. 14: CLP with pole placement

Code: Pole Placement

```
%
% Fall 2009
%
close all
figure(1);clf
set(gcf,'DefaultLineLineWidth',2)
set(gcf,'DefaultlineMarkerSize',10)
set(gcf,'DefaultlineMarkerFace','b')
clear all;%close all;
set(0, 'DefaultAxesFontSize', 14, 'DefaultAxesFontWeight','demi')
set(0, 'DefaultTextFontSize', 14, 'DefaultTextFontWeight','demi')

%Example: G(s)=1/2^2
%Design Gc(s) to put the clp poles at -1 + 2j
z=roots([-20 49 -10]);z=max(z),k=25/(5-2*z),alpha=5*z/(5-2*z),
num=1;den=[1 0 0];
knum=k*[1 z];kden=[1 10*z];
rlocus(conv(num,knum),conv(den,kden));
hold;plot(-alpha+eps*j,'d');plot([-1+2*j,-1-2*j],'d');hold off
r=rlocus(conv(num,knum),conv(den,kden),1)'
axis([-25 5 -15 15])
print -dpng -r300 rl_pp.png
```

Observations

- In a root locus design it is easy to see the pole locations, and thus we can relatively easily identify the dominant time response
  - Caveat is that near pole/zero cancelation complicates the process of determining which set of poles will dominate
- Some of the performance specifications are given in the frequency response, and it is difficult to determine those (and the corresponding system error gains) in the RL plot
- Easy for low-order systems, very difficult / time consuming for higher order ones
- As we will see, extremely difficult to identify the robustness margins using a RL plot
- A good approach for a fast/rough initial design

- Matlab tool called `sisotool` provides a great interface for designing and analyzing controllers