16.333 Lecture # 9

Basic Longitudinal Control

- Basic aircraft control concepts
- Basic control approaches

Basic Longitudinal Control

- Goal: analyze aircraft longitudinal dynamics to determine if the behavior is acceptable, and if not, then modify it using feedback control.

- Note that we could (and will) work with the full dynamics model, but for now, let's focus on the short period approximate model from lecture 7–5.

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

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

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

- As shown in the code, 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

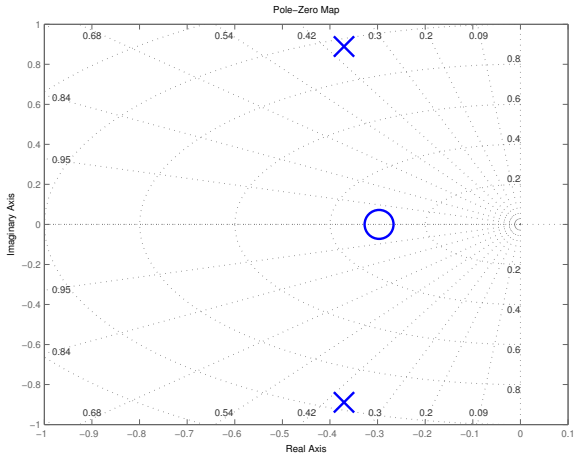


Figure 1: Pole-zero map for $G_{q\delta_e}$

- 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 do they prefer?

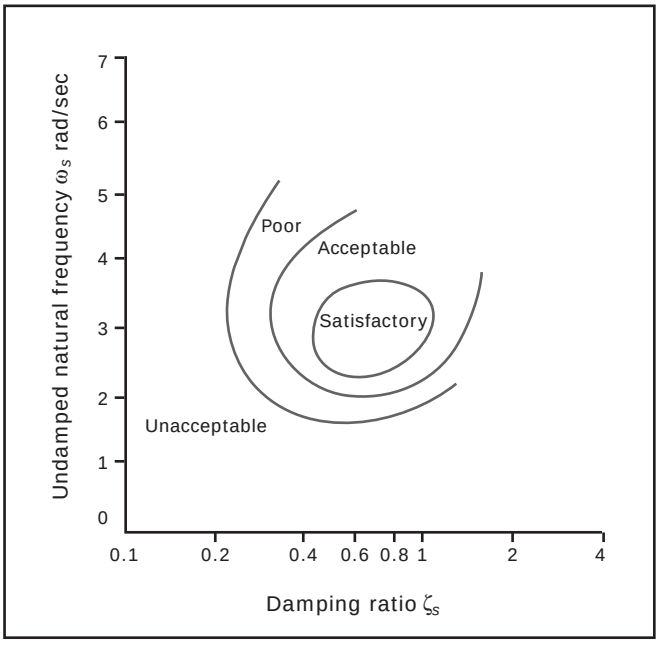


Figure 2: "Thumb Print" criterion

- This criterion has been around since the 1950's, but it is still valid.
- 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...

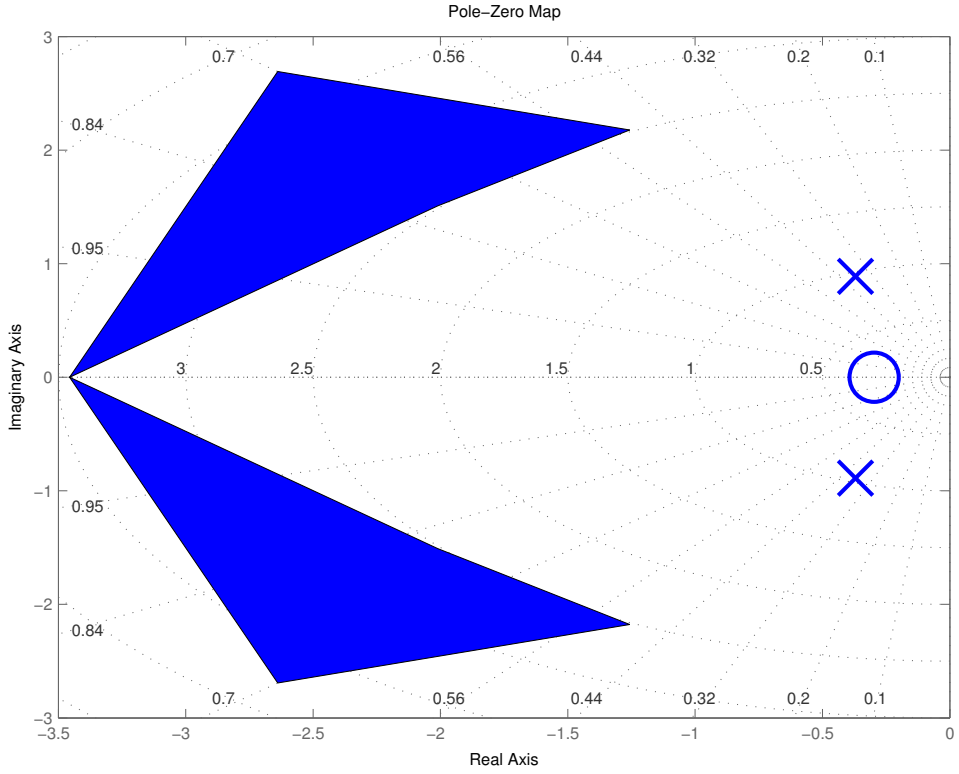
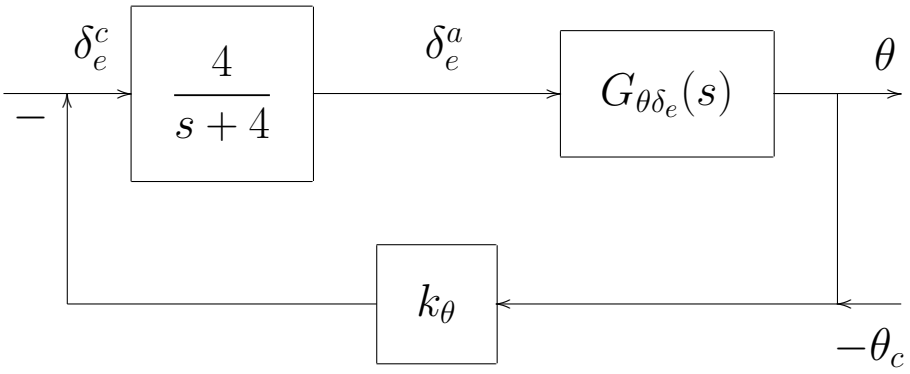


Figure 3: Pole-zero map and target pole locations

- Of course there are plenty of other things that we will consider when we design the controllers
  - Small steady state error to commands
  - δ_e within limits
  - No oscillations
  - Speed control

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

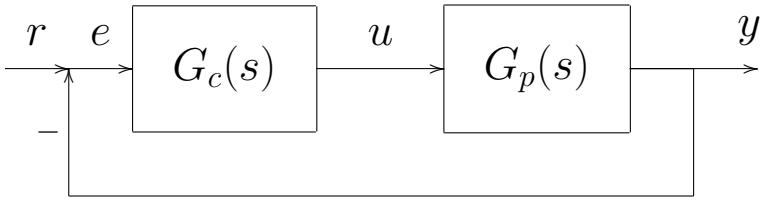
$$\theta = -G_{\theta\delta_e}(s)H(s)k_\theta(\theta - \theta_c)$$

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

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

and the controller transfer function is

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

- Signals are:

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

- This is the unity feedback form. We could add the controller G_c in the feedback path without changing the pole locations.

- Will add disturbances later.

- 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 $y = G_p G_c e$ and $e = r - y$, then easy to show that

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

where

$$G_{cl}(s) = \frac{K_c K_p N_c N_p}{D_c D_p + K_c K_p N_c N_p}$$

is the **closed loop transfer function**

- The denominator is called the **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 260.
- 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?

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

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

and $K = K_c K_p$ then

$$\phi_c(s) = 1 + KL_d(s) = 0$$

or equivalently for values of s for which $L_d(s) = -1/K$, with K real.

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

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

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

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

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:

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

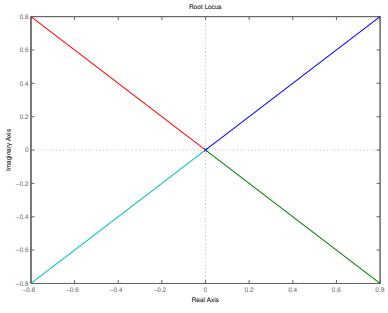
  - So n poles head to the zeros of $L_d(s)$
  - Remaining $p-n$ poles head to $|s| = \infty$ along **asymptotes** defined by the radial lines

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

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



- 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].
  - Explanation a bit too detailed and not that relevant

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

Root Locus Examples

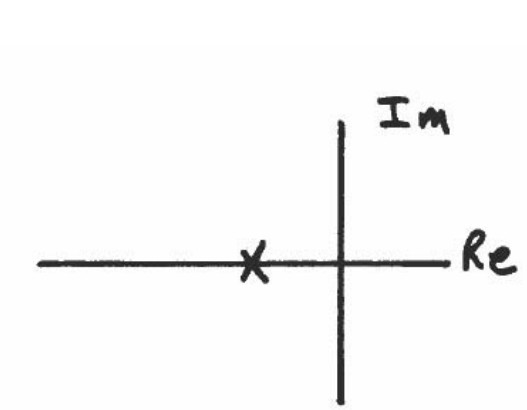


Figure 4: Basic

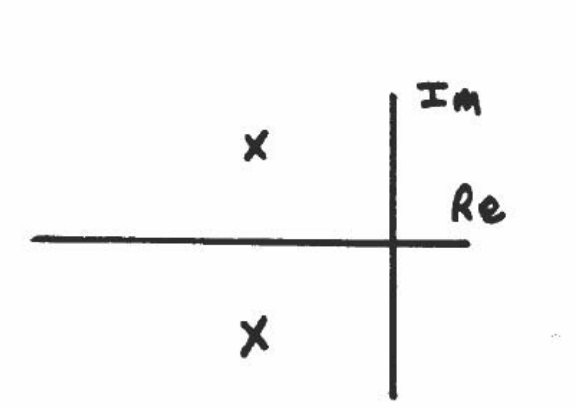


Figure 5: Two poles

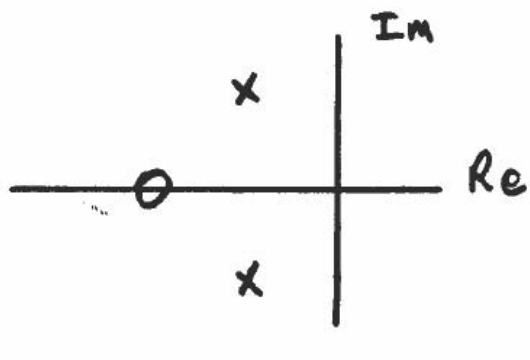


Figure 6: Add zero

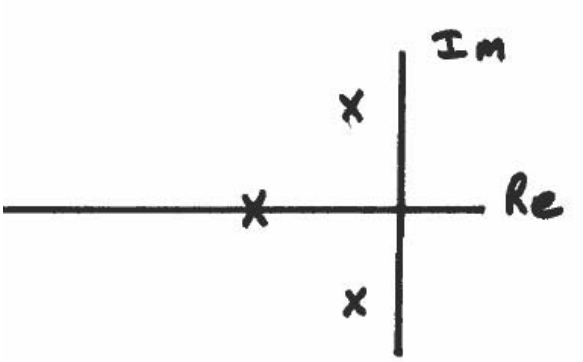


Figure 7: Three poles

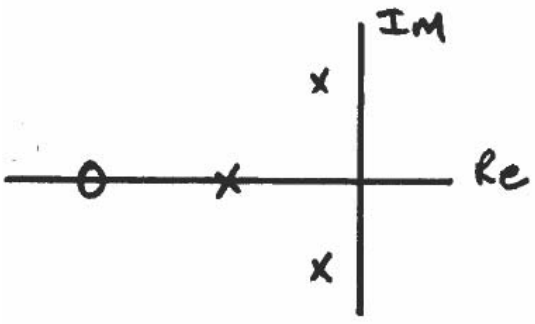


Figure 8: Add zero

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.

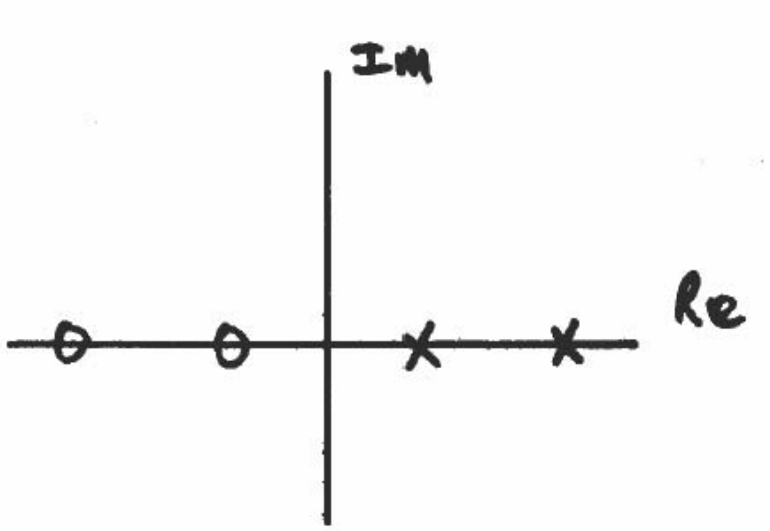


Figure 9: Complex case

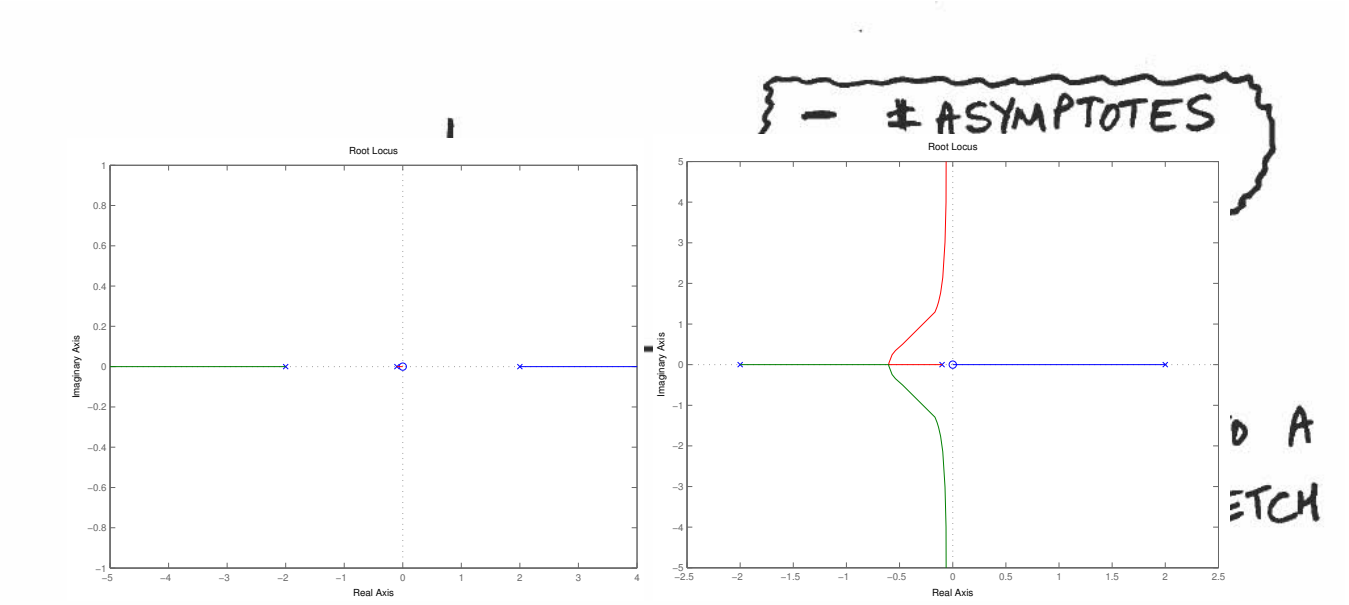


Figure 10: Very complex case

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

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

– 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.

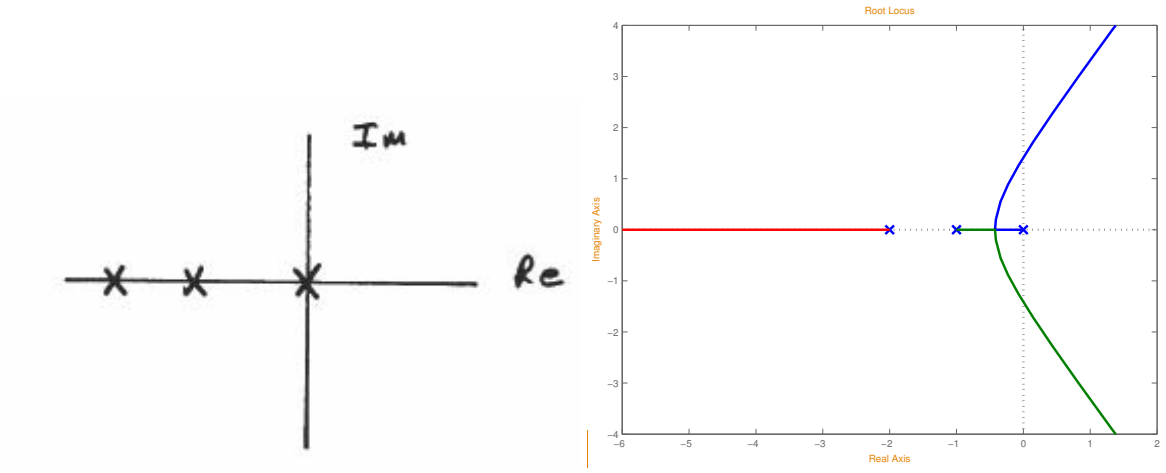


Figure 11: RL after adding integral FB

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

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 (I).

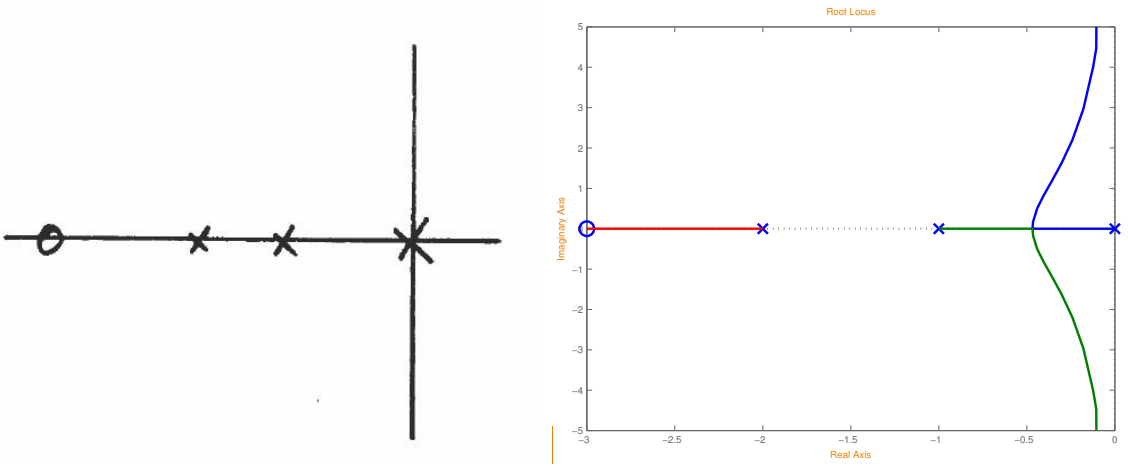


Figure 12: RL with proportional and integral FB

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 $G_c(s) = K_d s$

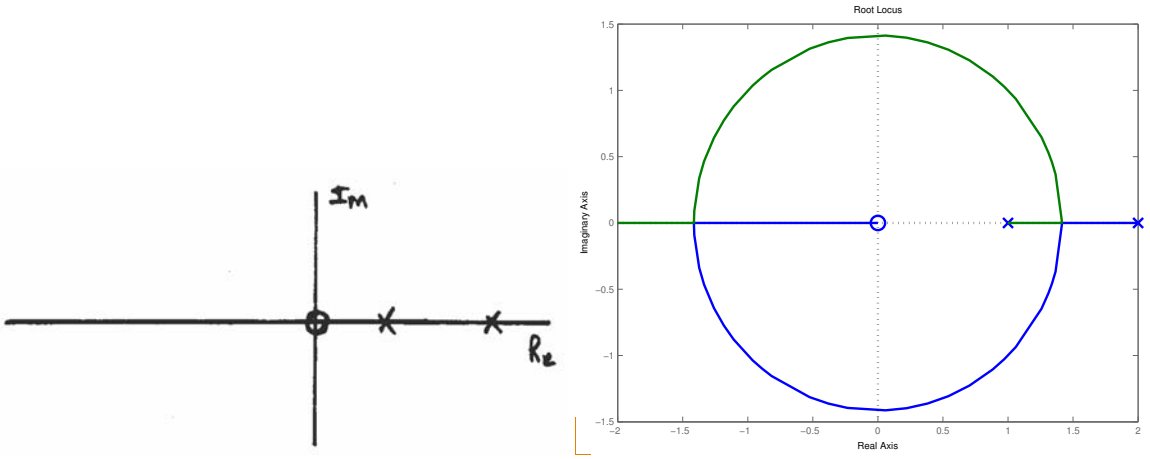


Figure 13: RL with derivative FB

   - 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.

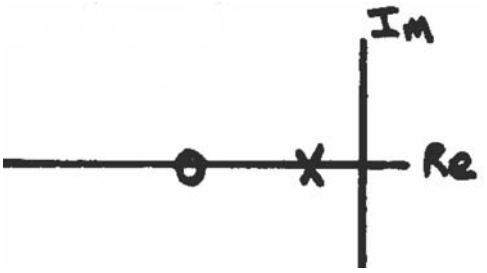
Controller Synthesis

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

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

- Can be made to look like various controllers, depending on how we pick K_c, p, and z
  - If we pick $z > p$, with p small, then

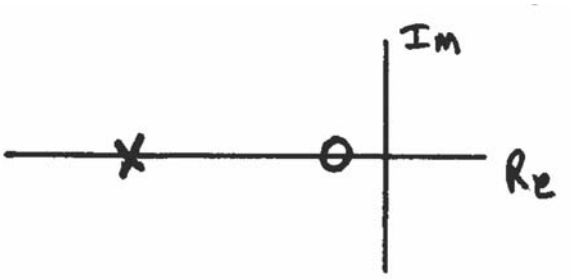
$$G_B(s) \approx K_c \frac{(s+z)}{s}$$



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

  - If we 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 (see 16.31 course notes online)

Pole Placement

- One option for simple systems is called pole placement.
- Consider a simple system $G_p = s^{-2}$ for which we want the closed loop poles to be at $-1 \pm 2\mathbf{i}$
- Proportional control clearly not sufficient, so use a compensator with 1 pole.

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

  So there are 3 CLP.
- Know that the desired characteristic equation is

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

- The 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 $\rightarrow$ Good rule of thumb is to take $p = 10z$ as a starting point.
- Compare the characteristic equations:

$$\phi_c(s) = s^2 + 10zs^2 + Ks + Kz = 0$$

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

  gives

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

solve for α, z, K

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

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

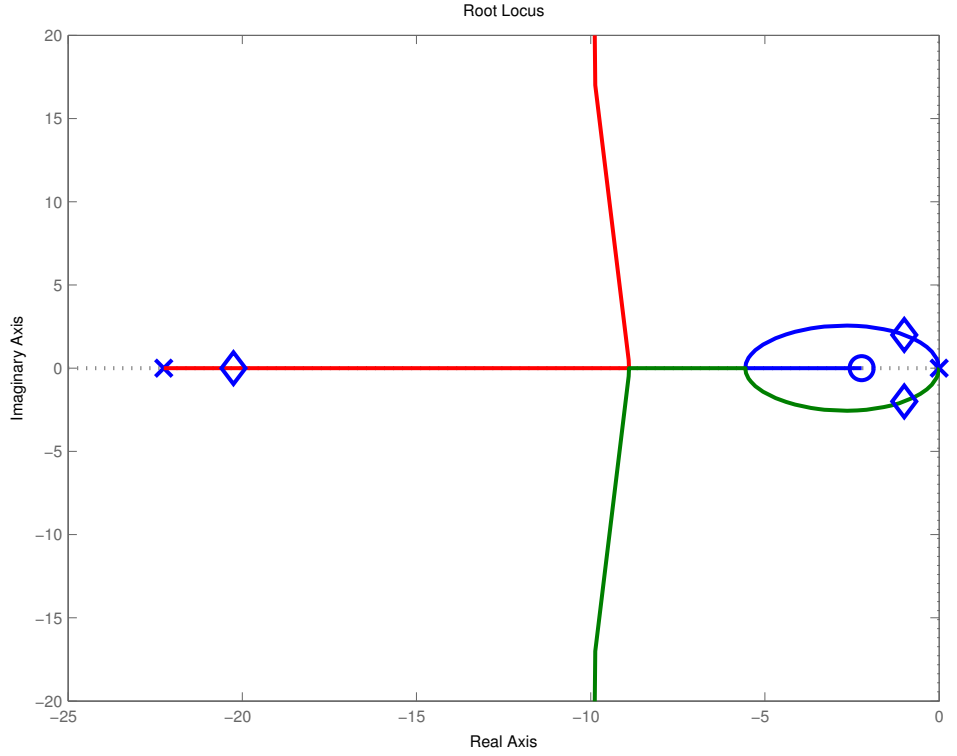


Figure 14: CLP with pole placement

Autopilot

- Back to the problem on 9–4: θ feedback to δ_e to control short period model

$$\phi_c(s) = 1 + G_{\theta\delta_e}(s)H(s)k_\theta = 0$$

with

$$G_{\theta\delta_e}(s) = -\frac{1.1569s + 0.3435}{s^3 + 0.7410s^2 + 0.9272s}, \qquad H(s) = \frac{4}{s+4}$$

- Pole/zero map:

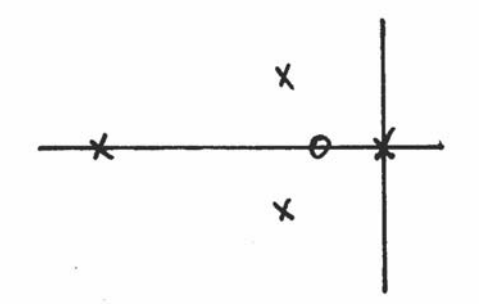


Figure 15: Pole/zero map of the short period autopilot with θ feedback

- Note: $K_p < 0$, so if $k_\theta > 0$, then we would have to draw a 0° locus
  - Pole at origin would move to right along real axis $\Rightarrow$ unstable.

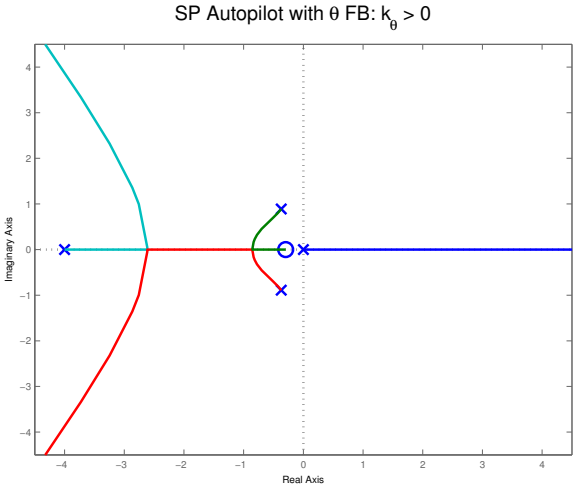


Figure 16: With $k_\theta > 0$

- So must use $k_\theta < 0$

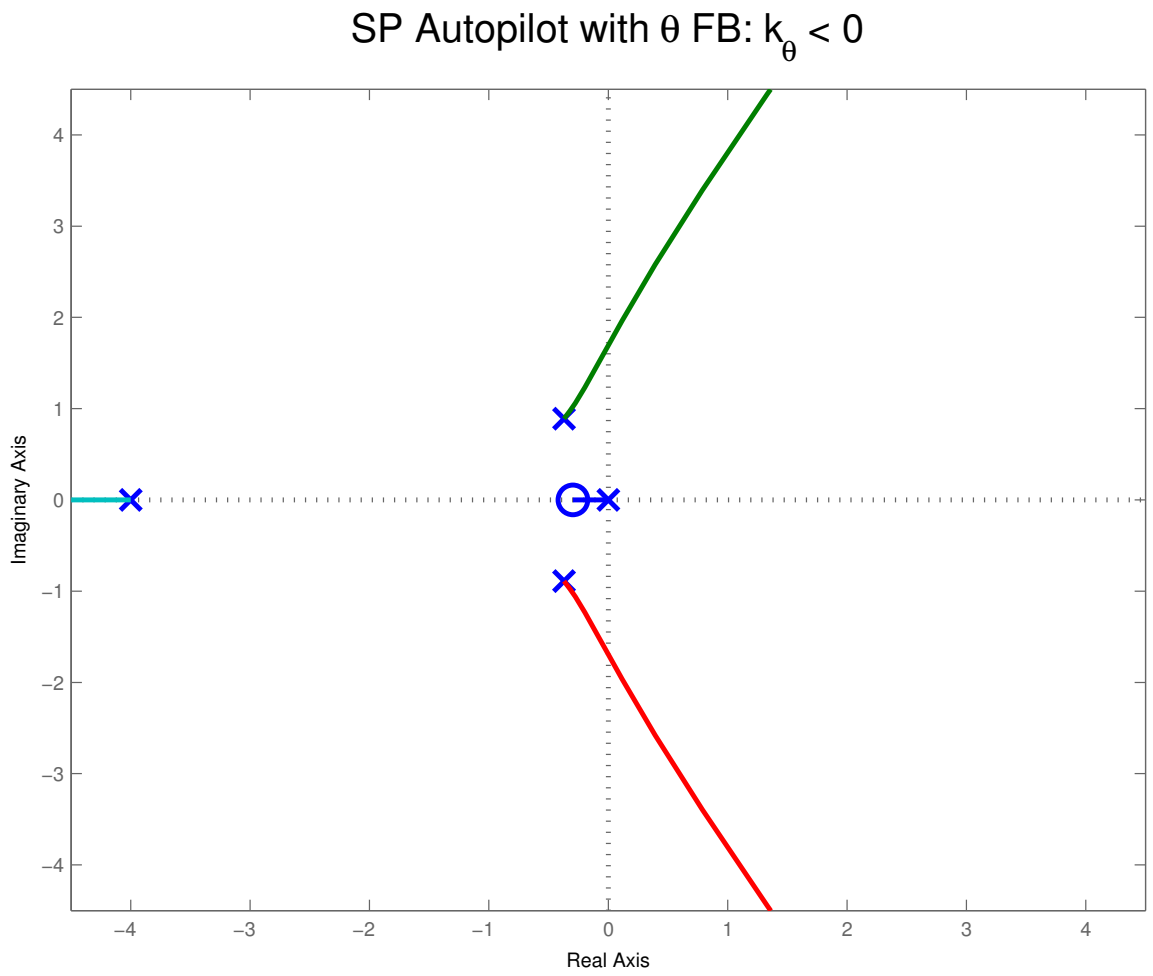


Figure 17: With $k_\theta < 0$

- Clear from the plot that θ feedback alone is not going to work particularly well.
  - Need to increase the gain to move the pole from the origin, but in doing so the SP poles start to move very close to the imaginary axis $\Rightarrow$ making the response worse

- Could use a rate gyro as the sensor instead and feedback q

$$G_{q\delta_e}(s) = -\frac{1.1569s + 0.3435}{s^2 + 0.7410s + 0.9272}, \quad H(s) = \frac{4}{s+4}$$

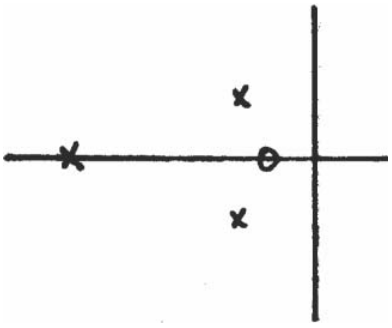


Figure 18: PZmap with q FB

- Again, pick the gain $k_q < 0$. Note locus for the real pole.

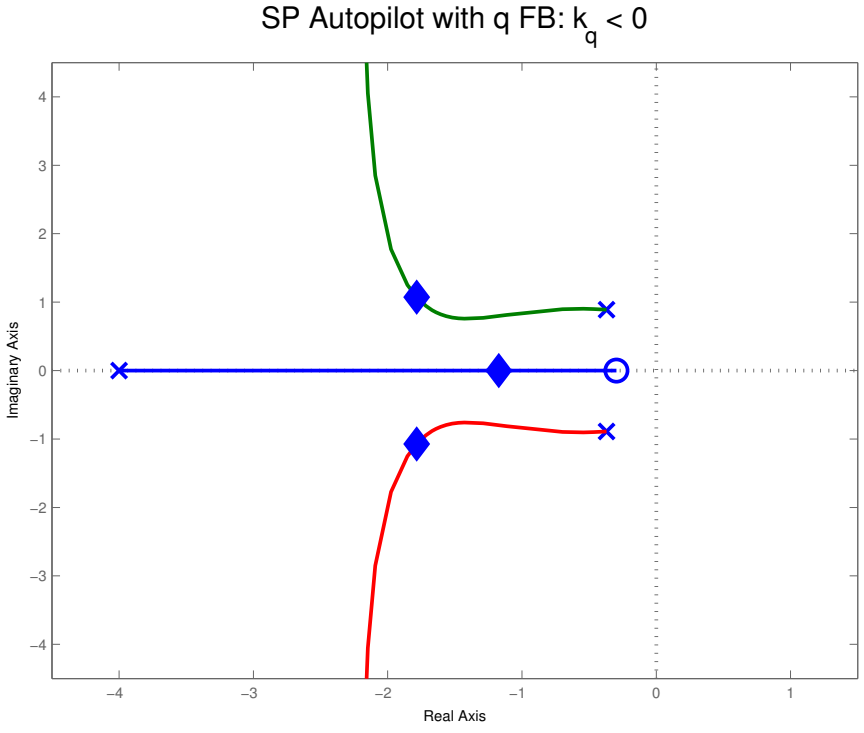
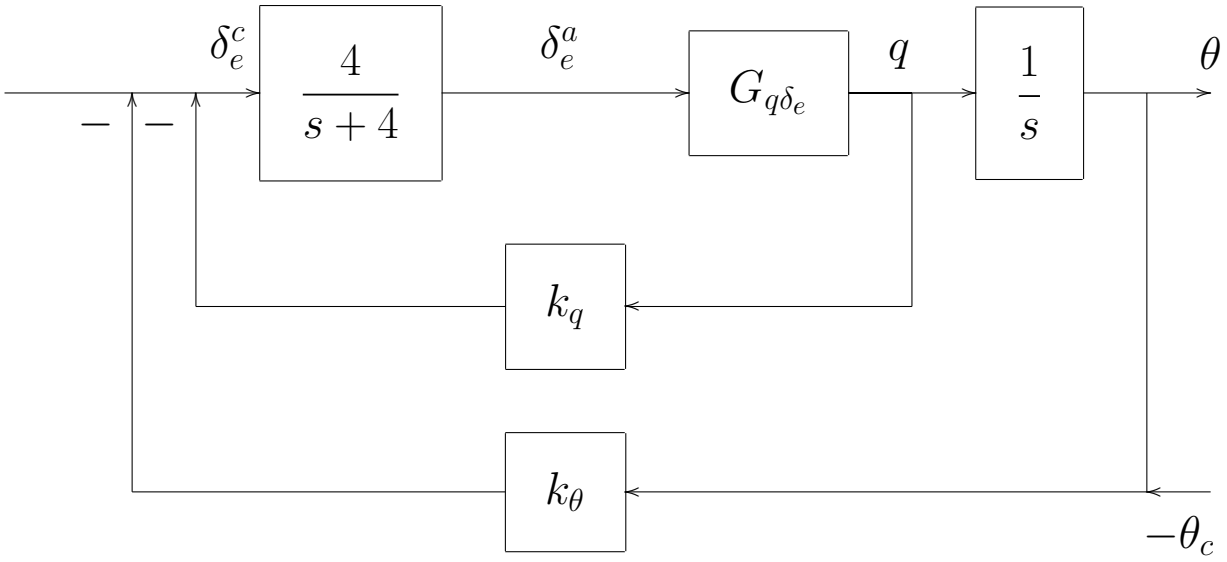


Figure 19: With $k_q < 0$. ♦ give the CLP with $k_q = -1$

- Can get an even better result if we combine the q and θ feedback – like PD on the measurement θ.



$$\begin{aligned} \delta_e^c &= -k_q q - k_\theta(\theta - \theta_c) \qquad \delta_e^a = H(s)\delta_e^c \\ q &= s\theta \end{aligned}$$

$$\begin{aligned} \theta = \frac{1}{s} G_{q\delta_e}\delta_e^a &= G_{\theta\delta_e}\delta_e^a \\ &= -G_{\theta\delta_e} H\left[(k_\theta + k_q s)\theta - k_\theta\theta_c\right] \end{aligned}$$

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

- Now pick $k_q = \zeta k_\theta$, and do RL versus k_θ

$$\phi_c(s) = 1 + G_{\theta\delta_e} H(1 + \zeta s)k_\theta = 0$$

  – Places a zero at $s = -1/\zeta$

- Step response is reasonable, but now need to pick the k_θ and k_q to get the desired ω_{sp} and ζ_{sp}

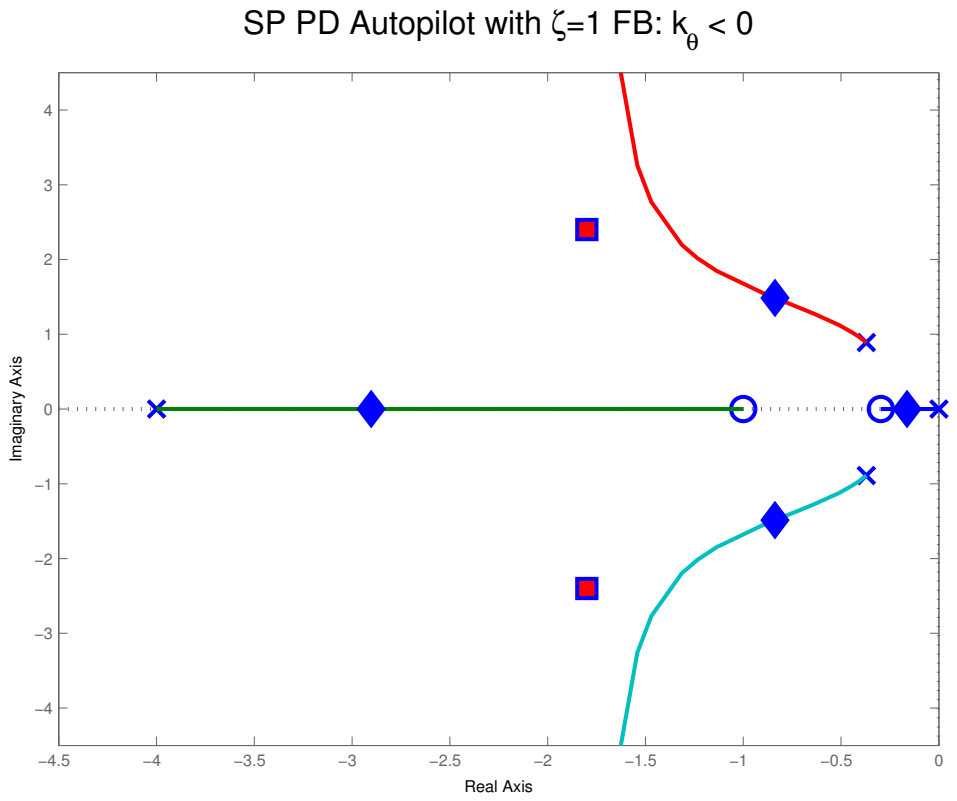


Figure 20: PD feedback with $\zeta = 1$. ♦ are the closed loop poles.

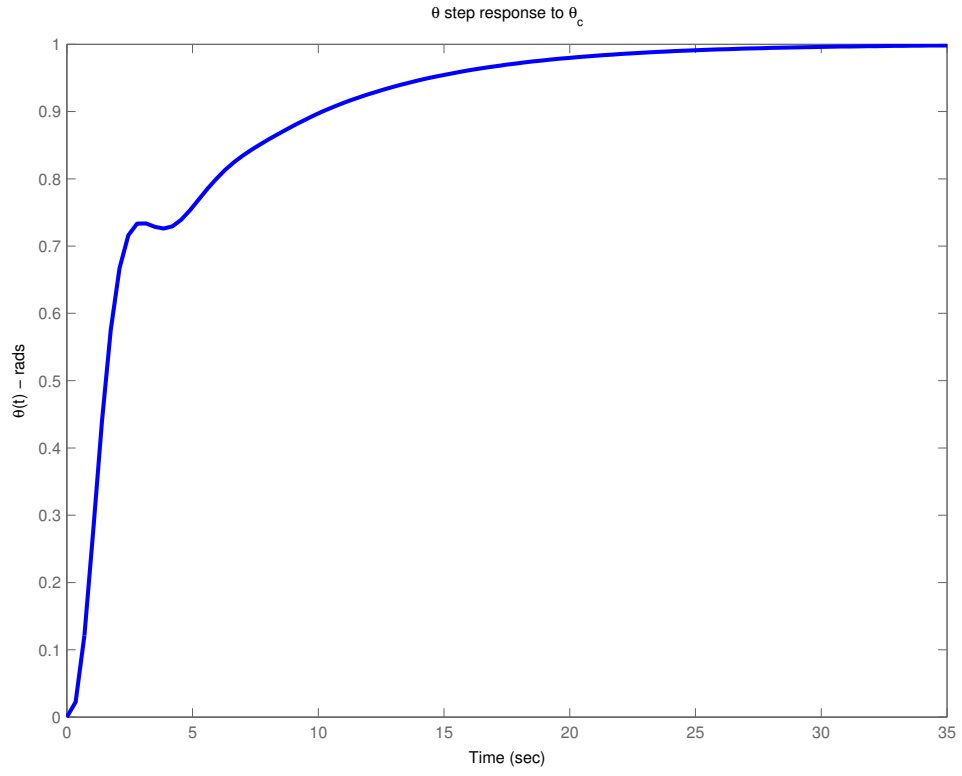


Figure 21: PD feedback with $\zeta = 1$ - step

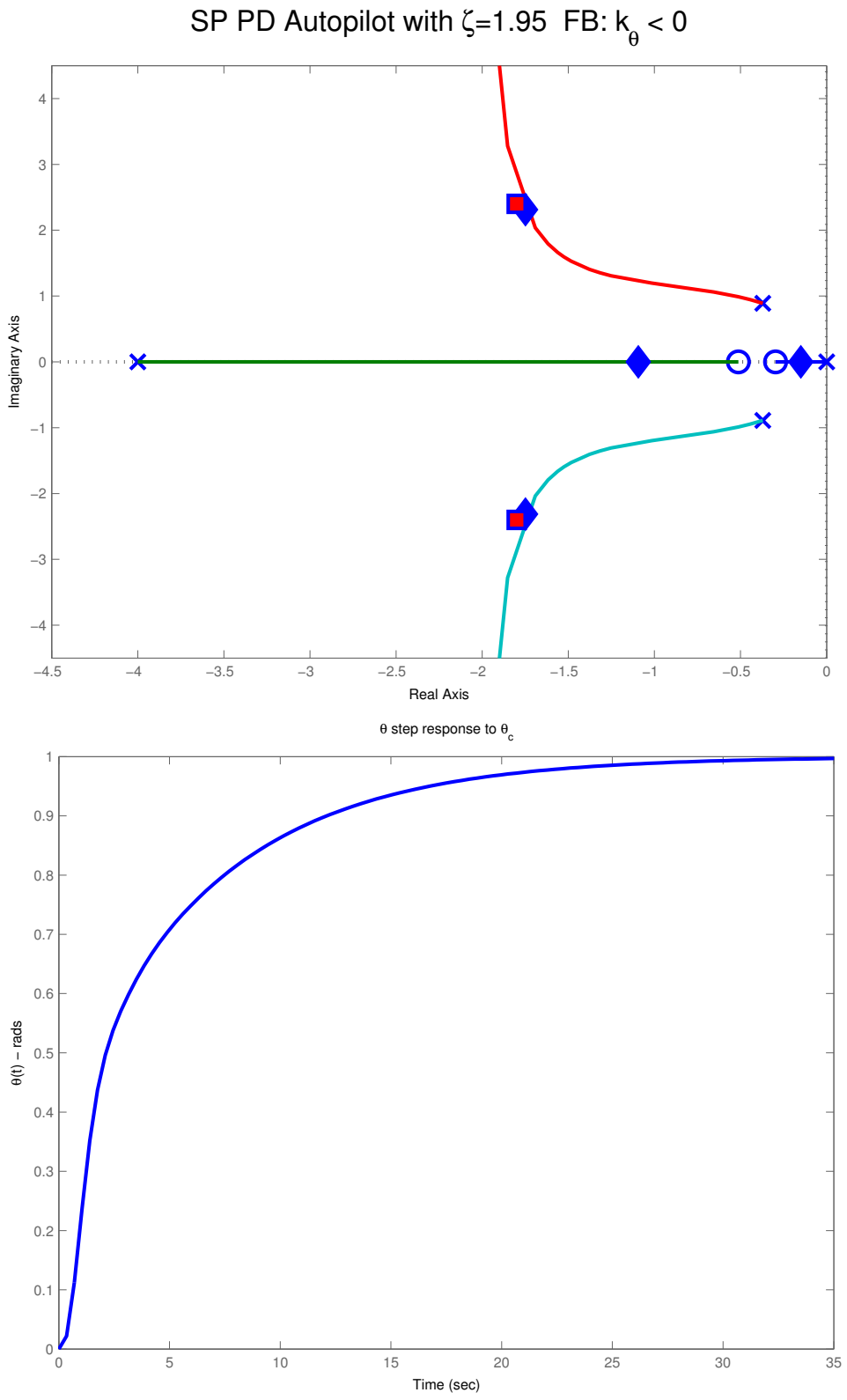


Figure 22: PD FB - step with $\zeta = 1.95$ $k_\theta = -1$. ♦ are the closed loop poles.

Summary

- Presented only basics of control design and analysis, but this is enough for us to get started
- Working with smaller models is good for design, but need to confirm that the full model still stable
- Could work with the full model, but that is much easier with state space tools

SP control codes (lect9a.m)

```
%
% Fall 2004
% 16.333
% Codes to investigate control for the SP
%
close all
figure(1);clf
set(gcf,'DefaultLineLineWidth',2)
set(gcf,'DefaultlineMarkerSize',10)
rlocus([1],conv([1 0],[1 3 2]))
print -depsc rl_pi1

load b747 Asp Bsp
figure(10);pzmap(ss(Asp,Bsp,[0 1],0));grid on;
hh=get(10,'children');hhh=get(hh(1),'children')
set(hhh(1),'MarkerSize',24);set(hhh(1),'LineWidth',2)
set(hhh(2),'MarkerSize',24);set(hhh(2),'LineWidth',2)
axis([-1 0.1 -1 1])
print -depsc pz_sp
jpdf('pz_sp')

figure(10);clf
test=[% thumbnail data from Nelson 167
.5 .4;
.8 .4;
1 .55;
.7 .6]
test(:,2)=test(:,2)*2*pi;
stest=[-test(:,1).*test(:,2) test(:,2).*sqrt(1-test(:,1).^2)]
fill(stest(:,1),stest(:,2),'blue',stest(:,1),-stest(:,2),'blue')
hold on
pzmap(ss(Asp,Bsp,[0 1],0));grid on;
hh=get(10,'children');hhh=get(hh(1),'children')
set(hhh(1),'MarkerSize',24);set(hhh(1),'LineWidth',2)
set(hhh(2),'MarkerSize',24);set(hhh(2),'LineWidth',2)
hold off
print -depsc pz_sp2
jpdf('pz_sp2')

figure(1);clf
set(gcf,'DefaultLineLineWidth',2)
set(gcf,'DefaultlineMarkerSize',10)
rlocus([1],conv([1 0],[1 3 2]))
print -depsc rl_pi1

figure(2);
set(gcf,'DefaultLineLineWidth',2)
set(gcf,'DefaultlineMarkerSize',12)
rlocus([1 3],conv([1 0],[1 3 2]))
print -depsc rl_pi2

figure(2);
set(gcf,'DefaultLineLineWidth',2)
set(gcf,'DefaultlineMarkerSize',12)
rlocus([1 0],conv([1 -2],[1 -1]))
print -depsc rl_d1

%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)'
print -depsc rl_pp
```

```
67
68  jpdf('rl_pi1')
69  jpdf('rl_pi2')
70  jpdf('rl_d1')
71  jpdf('rl_pp')
72
73  load b747 Asp Bsp
74  G=tf(ss(Asp,Bsp,[0 1],0))*tf(1,[1 0]);
75  H=tf(4,[1 4]);
76  rlocus(G*H)
77  axis([-3 3 -3 3]*1.5)
78  title('SP Autopilot with \theta FB: k_\theta > 0','FontSize',16)
79  print -depsc sp_ap1
80  jpdf('sp_ap1')
81
82  rlocus(-G*H)
83  axis([-3 3 -3 3]*1.5)
84  title('SP Autopilot with \theta FB: k_\theta < 0','FontSize',16)
85  print -depsc sp_ap2
86  jpdf('sp_ap2')
87
88  load b747 Asp Bsp
89  G=tf(ss(Asp,Bsp,[0 1],0));
90  H=tf(4,[1 4]);
91  rlocus(-G*H)
92  rr=rlocus(-G*H,1)
93  hold on
94  plot(rr+sqrt(-1)*eps,'d','MarkerFaceColor','b')
95  hold off
96  axis([-3 1 -3 3]*1.5)
97  title('SP Autopilot with q FB: k_q < 0','FontSize',16)
98  print -depsc sp_ap3
99  jpdf('sp_ap3')
100
101 load b747 Asp Bsp
102 zeta=1;
103 k_th=1;
104 PD=tf([zeta 1],1);
105 G=tf(ss(Asp,Bsp,[0 1],0))*tf(1,[1 0]);
106 H=tf(4,[1 4]);
107 rlocus(-G*H*PD)
108 rr=rlocus(-G*H*PD,k_th)
109 hold on
110 plot(rr+sqrt(-1)*eps,'d','MarkerFaceColor','b')
111 plot(-1.8+2.4*sqrt(-1),'s','MarkerFaceColor','r')
112 plot(-1.8-2.4*sqrt(-1),'s','MarkerFaceColor','r')
113 hold off
114 axis([-3 0 -3 3]*1.5)
115 title('SP PD Autopilot with \zeta=1 FB: k_\theta < 0','FontSize',16)
116 print -depsc sp_pd1
117 jpdf('sp_pd1')
118
119 figure(2);
120 set(gcf,'DefaultLineLineWidth',2)
121 set(gcf,'DefaultlineMarkerSize',12)
122 G_cl=G*H*(-k_th)/(1+G*H*PD*(-k_th));
123 step(G_cl,35)
124 h=get(gcf,'children');hh=get(h(1),'children');set(hh(1),'LineWidth',2)
125 title('\theta step response to \theta_c')
126 ylabel('\theta(t) - rads')
127 print -depsc tstep
128 jpdf('tstep')
```

SP control codes: Part 2 (lect9b.m)

```
% 16.333 Fall 2004
% SP design meeting the performance goals a bit better
%
load b747 Asp Bsp
G=tf(ss(Asp,Bsp,[0 1],0))*tf(1,[1 0]);
H=tf(4,[1 4]);

zeta=1.95;
k_th=1;
PD=tf([zeta 1],1);

rlocus(-G*H*PD)
rr=rlocus(-G*H*PD,k_th)
hold on
plot(rr+sqrt(-1)*eps,'d','MarkerFaceColor','b')
plot(-1.8+2.4*sqrt(-1),'s','MarkerFaceColor','r')
plot(-1.8-2.4*sqrt(-1),'s','MarkerFaceColor','r')
hold off
axis([-3 0 -3 3]*1.5)
title(['SP PD Autopilot with \zeta=',num2str(zeta),'  FB: k_\theta < 0'],'FontSize',16)
print -depsc sp_pd2
jpdf('sp_pd2')

figure(2);
set(gcf,'DefaultLineLineWidth',2)
set(gcf,'DefaultlineMarkerSize',12)
G_cl=G*H*(-k_th)/(1+G*H*PD*(-k_th));
step(G_cl,35)
h=get(gcf,'children');hh=get(h(1),'children');set(hh(1),'LineWidth',2)
title('\theta step response to \theta_c')
ylabel('\theta(t) - rads')
print -depsc tstep2
jpdf('tstep2')

load b747 A B
Gf=tf(ss(A,B(:,1),[0 0 0 1],0));
H=tf(4,[1 4]);

zeta=1.95;
k_th=1;
PD=tf([zeta 1],1);

rlocus(-Gf*H*PD)
rrf=rlocus(-Gf*H*PD,k_th)
hold on
plot(rrf+sqrt(-1)*eps,'s','MarkerFaceColor','b')
plot(rr+sqrt(-1)*eps,'rd','MarkerFaceColor','r')
hold off
axis([-3 0 -3 3]*1.5)
title(['SP PD Autopilot with \zeta=',num2str(zeta),'  FB: k_\theta < 0'],'FontSize',16)
print -depsc sp_f1
jpdf('sp_f1')
```

SP control: Basic Dynamics (lect9.m)

```
clear all
prt=1;
%
% B747 longitudinal dynamics
% 16.333 Fall 2004
%
% B747 at Mach 0.8, 40,000ft, level-flight
% From Etkin and Reid
%
        % metric
    Xu=-1.982e3;Xw=4.025e3;
        Zu=-2.595e4;Zw=-9.030e4;Zq=-4.524e5;Zwd=1.909e3;
        Mu=1.593e4;Mw=-1.563e5;Mq=-1.521e7;Mwd=-1.702e4;

        g=9.81;theta0=0;S=511;cbar=8.324;
        U0=235.9;Iyy=.449e8;m=2.83176e6/g;cbar=8.324;rho=0.3045;
        Xdp=.3*m*g;Zdp=0;Mdp=0;
        Xde=-3.818e-6*(1/2*rho*U0^2*S);Zde=-0.3648*(1/2*rho*U0^2*S);
        Mde=-1.444*(1/2*rho*U0^2*S*cbar);;

A=[Xu/m Xw/m 0 -g*cos(theta0);[Zu Zw Zq+m*U0 -m*g*sin(theta0)]/(m-Zwd);
   [Mu+Zu*Mwd/(m-Zwd) Mw+Zw*Mwd/(m-Zwd) Mq+(Zq+m*U0)*Mwd/(m-Zwd) ...
         -m*g*sin(theta0)*Mwd/(m-Zwd)]/Iyy;
   [ 0 0 1 0]];
B=[Xde/m Xdp/m;Zde/(m-Zwd) Zdp/(m-Zwd);(Mde+Zde*Mwd/(m-Zwd))/Iyy ...
      (Mdp+Zdp*Mwd/(m-Zwd))/Iyy;0 0];

% Short-period Approx
Asp=[Zw/m U0;
   [Mw+Zw*Mwd/m Mq+U0*Mwd]/Iyy];
Bsp=[Zde/m;(Mde+Zde/m*Mwd)/Iyy];
[nsp,dsp]=ss2tf(Asp,Bsp,eye(2),zeros(2,1));
[Vsp,evsp]=eig(Asp);evsp=diag(evsp);
%rifd(evsp)

%
% CONTROL
%
% actuator dynamics are a lag at 4
actn=4;actd=[1 4]; % H(s) in notes
%
% use the short period model since that is all we are trying to control
%
[nsp,dsp]=ss2tf(Asp,Bsp,eye(2),zeros(2,1));
[nfull,dfull]=ss2tf(A,B(:,1),eye(4),zeros(4,1));
%
% form num and den for the "plant" = act_dyn*G == N/D
%
% short period model
N=conv(actn,nsp(2,:));% q is second state
D=conv(actd,dsp);
% full model
Nqfull=conv(actn,nfull(3,:));% q is third state
Ntfull=conv(actn,nfull(4,:));% theta is fourth state
Dfull=conv(actd,dfull);
%
% add an extra pole to get the integrator for the \dot theta = q
%
Ntheta=conv(N,1);Dtheta=conv(D,[1 0]);

figure(1);clf;
K_th0=-1;
rlocus(sign(K_th0)*Ntheta,Dtheta);
r_th0=rlocus(Ntheta,Dtheta,K_th0)';hold on;plot(r_th0+eps*sqrt(-1),'v');
hold off
sgrid(.6,3); % gives target damping and freqs
```

```
axis([-5 .5 -4 4]);grid
title('Closed-loop poles using only theta FB')
if prt
   print -depsc ac_th0
end

figure(2);clf;
K_q=-1;
rlocus(sign(K_q)*N,D);
r_q=rlocus(N,D,K_q)';hold on;plot(r_q+eps*sqrt(-1),'v');
hold off
sgrid(.6,3);
axis([-5 .5 -4 4]);grid
title('Closed-loop poles using q FB')
if prt
   print -depsc ac_q
end
%
%closed-loop dynamics with q FB in place (inner loop)
% GH/(1+k_q GH) = (N/D) / (1+k_q N/D) = N/(D+k_q N)
%
Nq=N;Dq=K_q*[0 N]+D;
%
% add integrator for q ==> theta
%
K_th=-1;
%
figure(3);clf;
rlocus(sign(K_q)*Nq,conv(Dq,[1 0]));grid
r_th=rlocus(Nq,conv(Dq,[1 0]),K_th)';hold on;
plot(r_q+eps*sqrt(-1),'v');plot(r_th+eps*sqrt(-1),'^');
hold off
sgrid(.6,3);
axis([-5 .5 -4 4]);grid
title('Closed-loop poles also using theta FB')
if prt
   print -depsc ac_th
end
%
% as a final check, form the closed-loop dynamics
% using the original short period model
Ncl=Ntheta*K_th;
Dcl=Dtheta+conv([0 N],[K_q K_th]);
roots(Dcl)
%
SPcl=tf(Ncl,Dcl);
[y,t]=step(SPcl);
figure(4)
plot(t,y);xlabel('time');ylabel('\theta rad','FontSize',12)
if prt
   print -depsc ac_th_step
end

%
% as a final check, form the closed-loop dynamics
% using the full dynamics from del_e to theta
Nfcl=Ntfull*K_th;
Dfcl=Dfull+conv(Ntfull,[K_q K_th]);
roots(Dfcl)
figure(3)
hold on;plot(roots(Dfcl)+eps*sqrt(-1),'r*');hold off
axis([-3 .5 -2 2]);grid
title('Closed-loop poles with q and th FB on FULL model')
if prt
   print -depsc ac_full
end

%return

%
% add theta state to the short period approx model
%
```

```
Ksp=place(Asp,Bsp,[roots([1 2*.6*3 3^2])'])
Asp2=[Asp [0 0]';0 1 0];Bsp2=[Bsp;0];
Ksp2=place(Asp2,Bsp2,[roots([1 2*.6*3 3^2])',-.25])
%
% add actuator model to SP with theta
%
Plist=[roots([1 2*.6*3 3^2])',-.25,-3];
At2=[Asp2 Bsp2(:,1);zeros(1,3) -4];Bt2=[zeros(3,1);4];
Kt2=place(At2,Bt2,[Plist])
step(ss(At2-Bt2*Kt2,Bt2,[0 0 1 0],0),35)
hh=get(gcf,'children');hhh=get(hh(1),'children')
set(hhh(1),'MarkerSize',24);set(hhh(1),'LineWidth',2)
ylabel('\theta rads','FontSize',12)
print -depsc fsfb_step.eps
jpdf('fsfb_step')
%
ev=eig(A);
% dampe short period, but leave the phugoid where it is
Plist=[roots([1 2*.6*3 3^2])' ev([3 4],1)'];
K1=place(A,B(:,1),Plist)
%
% add actuator model
%
At=[A B(:,1);zeros(1,4) -4];Bt=[zeros(4,1);4];
Kt=place(At,Bt,[Plist -3])

save b747 A B Asp Bsp

%!eps2pdf /f="ac7_fig1.eps"
%!mv c:\ac7_fig1.pdf ./ac7_fig1.pdf


```