Topic #4

16.30/31 Feedback Control Systems

Control Design using Bode Plots

• Performance Issues
• Synthesis
• Lead/Lag examples
Bode’s Gain Phase Relationship

• Control synthesis by classical means would be very hard if we had to consider both the magnitude and phase plots of the loop, but that is not the case.

• **Theorem:** For any stable, minimum phase system with transfer function $G(s)$, $\angle G(j\omega)$ is **uniquely related** to the slope of $|G(j\omega)|$.
  
  • Relationship is that, on a log-log plot, if slope of the magnitude plot is constant over a decade in frequency, with slope $n$, then
  
  $$\angle G(j\omega) \approx 90^\circ n$$

• So in the crossover region, where $L(j\omega) \approx 1$ if the magnitude plot is (locally):

  - $s^0$ slope of 0, so no crossover possible
  - $s^{-1}$ slope of -1, so about $90^\circ$ PM
  - $s^{-2}$ slope of -2, so PM very small

• **Basic rule** of classical control design:

  Select $G_c(s)$ so that **LTF crosses over with a slope of -1.**
Performance Issues

- Step error response

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

and we can determine \( G_c(0)G_p(0) \) from the low frequency Bode plot for a type 0 system.

- For a type 1 system, the DC gain is infinite, but define

\[ K_v = \lim_{s \to 0} sG_c(s)G_p(s) \Rightarrow e_{ss} = 1/K_v \]

- So can easily determine this from the low frequency slope of the Bode plot.
Performance Issues II

- Classic question: how much phase margin do we need? Time response of a second order system gives:

1. Closed-loop pole damping ratio $\zeta \approx PM/100$, $PM < 70^\circ$

2. Closed-loop resonant peak $M_r = \frac{1}{2\zeta \sqrt{1-\zeta^2}} \approx \frac{1}{2 \sin(PM/2)}$, near
   $$\omega_r = \omega_n \sqrt{1 - 2\zeta^2}$$

3. Closed-loop bandwidth
   $$\omega_{BW} = \omega_n \sqrt{1 - 2\zeta^2 + \sqrt{2 - 4\zeta^2 + 4\zeta^4}}$$
   and $$\omega_c = \omega_n \sqrt{1 + 4\zeta^4 - 2\zeta^2}$$

Fig. 1: Frequency domain performance specifications.

- So typically specify $\omega_c$, $PM$, and error constant as design goals
Fig. 2: Crossover frequency for second order system

- Other rules of thumb come from approximating the system as having a 2nd order dominant response:

  10-90% rise time \[ t_r = \frac{1 + 1.1\zeta + 1.4\zeta^2}{\omega_n} \]

  Settling time (5%) \[ t_s = \frac{3}{\zeta\omega_n} \]

  Time to peak amplitude \[ t_p = \frac{\pi}{\omega_n \sqrt{1 - \zeta^2}} \]

  Peak overshoot \[ M_p = e^{-\zeta\omega_n t_p} \]
Frequency Domain Design

- Looked at the building block

\[ G_c(s) = K_c \frac{s + z}{s + p} \]

- Question: how choose \( G_c(s) \) to modify the LTF \( L(s) = G_c(s)G_p(s) \) to get the desired bandwidth, phase margin, and error constants?

- Lead Controller (\(|z| < |p|\))
  - Zero at a lower frequency than the pole
  - Gain increases with frequency (slope +1)
  - Phase positive (i.e. this adds phase lead)

Fig. 3: Lead: frequency domain.
Lead Mechanics

- Maximum phase added
  \[ \sin \phi_{\text{max}} = \frac{1 - \alpha}{1 + \alpha} \]
  where \( \alpha = |z|/|p| \), which implies that
  \[ \alpha = \frac{1 - \sin \phi_{\text{max}}}{1 + \sin \phi_{\text{max}}} \]

- Frequency of the maximum phase addition is \( \omega_{\text{max}} = \sqrt{|z| \cdot |p|} \)
  - Usually try to place this near \( \omega_c \)

- High frequency gain increase is by \( 1/\alpha \)

- So there is a compromise between wanting to add a large amount of phase (\( \alpha \) small) and the tendency to generate large gains at high frequency
  - So try to keep \( 1/\alpha \leq 10 \), which means \( |p| \leq 10|z| \) and
  \[ \phi_{\text{max}} \leq 60^\circ \]

- If more phase lead is needed, use multiple lead controllers
  \[ G_c(s) = k_c \frac{(s + z_1)(s + z_2)}{(s + p_1)(s + p_2)} \]

- Select of the overall gain is problem specific.
  - One approach is to force the desired crossover frequency so that
  \[ |L(j\omega_c)| = 1 \]

- Use Lead to add phase \( \Rightarrow \) increases PM \( \Rightarrow \) improves transient response.
Lead Mechanics II

- Adding a lead to the LTF changes both the magnitude and phase, so it is difficult to predict the new crossover point (which is where we should be adding the extra phase).

![Diagram showing dB and phase margins with crossover point.

Image by MIT OpenCourseWare.

Fig. 4: Lead example

- The process is slightly simpler if we target the lead compensator design at a particular desired $\omega_c$

  1. Find the $\phi_{\text{max}}$ required
     - Note that $\phi_{\text{required}} = PM - (180^\circ + \angle G(j\omega_c))$
  2. Put $\phi_{\text{max}}$ at the crossover frequency, so that
     $\omega_c^2 = |p| \cdot |z|$
  3. Select $K_c$ so that crossover is at $\omega_c$
Design Example

- Design a compensator for the system $G(s) = \frac{1}{s(s+1)}$
  - Want $\omega_c = 10\text{rad/sec}$ and $PM \approx 40^\circ$
- Note that at 10 rad/sec, the slope of $|G|$ is -2, corresponding to a plant phase of approximately $180^\circ$
- So need to add a lead compensator $\Rightarrow$ adds a slope of +1 (locally), changing the LTF slope to -1, and thus increasing the phase margin

![Diagram showing magnitude and phase of G(s) and GG_c](Image by MIT OpenCourseWare.)

**Fig. 5: Lead example**

- Design steps:
  1. \[ \begin{cases} \frac{z}{p} = 1 - \sin \phi_m \\ \frac{z}{p} = 1 + \sin \phi_m \end{cases} \quad \begin{cases} \angle G(j\omega_c) \approx -180^\circ \\ PM = 40^\circ \end{cases} \Rightarrow \phi_m = 40^\circ \]
  \[ \frac{z}{p} = 0.22 \]
  2. $\omega_c^2 = z \cdot p = 10^2 \Rightarrow z = 4.7, p = 21.4$
  3. Pick $k_c$ so that $|G_cG(j\omega_c)| = 1$
Fig. 6: Lead example

Code: Lead Example

```matlab
% Lead Example 2009
% close all
set(0, 'DefaultLineWidth','2')
set(0, 'DefaultMarkerSize',10)
set(0, 'DefaultMarkerFace','b')
set(0, 'DefaultFontSize', 14, 'DefaultFontWeight','demi')
set(0, 'DefaultTextFontSize', 14, 'DefaultTextFontWeight','demi')

% g=1/s/(s+1)
figure(1);clf;
wc=10;
PM=40*pi/180;

G=tf(1,conv([1 0],[1 1]));
phi_G=phase(evalfr(G,j*wc))*180/pi;

phi_m=PM-(180+phi_G);

zdp=(1-sin(phi_m))/(1+sin(phi_m));
z=sqrt(wc^2*zdp);
p=z/zdp;

Gc=tf([1 z],[1 p]);
k_c=1/abs(evalfr(Gc+w));
Gc=G*c*k_c;
w=logspace(-2,3,300);
f_G=freqresp(G,w);f_Gc=freqresp(Gc,w);f_L=freqresp(G+Gc,w);
f_G=squeeze(f_G);f_Gc=squeeze(f_Gc);f_L=squeeze(f_L);
loglog(w,abs(f_G),w,abs(f_Gc),w,abs(f_L));grid on
legend('Plant G','Comp Gc','LTF L','Location','SouthWest');grid on
print -dpng -r300 lead_examp2.png
```
Lag Mechanics

- If pole at a lower frequency than zero:
  - Gain decreases at high frequency – typically scale lag up so that HF gain is 1, and thus the low frequency gain is higher.
  - Add negative phase (i.e., adds lag)

Fig. 7: Lag: frequency domain $G_{lag} = k_c \frac{s/z + 1}{s/p + 1}$

- Typically use a lag to add $20 \log \alpha$ to the low frequency gain with (hopefully) a small impact to the PM (at crossover)
  - Pick the desired gain reduction at high frequency $20 \log (1/\alpha)$, where $\alpha = |z|/|p|$
  - Pick $|G_{lag}|_{s=0} = k_c$ to give the desired low frequency gain increase for the LTF (shift the magnitude plot up)
  - Heuristic: want to limit frequency of the zero (and pole) so that there is a minimal impact of the phase lag at $\omega_c \Rightarrow z = \omega_c/10$

September 19, 2010
Lag Compensation

- Assumption is that we need to modify (increase) the DC gain of the LTF to reduce the tracking error.
- Two ways to get desired low frequency gain:
  1. Using just a gain increase, which increases $\omega_c$ and decrease $PM$
  2. Add Lag dynamics that increase increase the gain at low frequency and leave the gain near and above $\omega_c$ unchanged

![Diagram of phase margin and gain crossover](Image by MIT OpenCourseWare)
Lag Example

Lag example with $G(s) = \frac{3}{((s + 1)^2(s/2 + 1))}$ and on right $G_c(s) = 1$, left $G_c(s) = \frac{(s + 0.1)}{(s + 0.01)}$, which should reduce steady state step error from 0.25 to 0.032.
Code: Lag Example

```matlab
%% Lag Example 2009

% close all
set(0, 'DefaultLineLineWidth', 2);
set(0, 'DefaultLineMarkerSize', 10);
set(0, 'DefaultAxesFontSize', 14, 'DefaultAxesFontWeight', 'demi');
set(0, 'DefaultTextFontSize', 14, 'DefaultTextFontWeight', 'demi');

G = tf([3], conv(conv([1/1 1], [1/2 1]), [1 1]));
wc = j; zl = wc/10; pl = zl/10;

Gc = zl/pl*tf([1/(zl) 1], [1/(pl) 1]);
figure(1); clf;
set(gcf,'DefaultLineLineWidth', 2)
Gc = 1;
Gc2 = 2;
figure(4); set(gcf,'DefaultLineLineWidth', 2)
figure(3); rlocus(L); rr = rlocus(L, 1); hold on;

Gc = zl/pl*tf([1/(zl) 1], [1/(pl) 1]);
figure(2); clf;
set(gcf,'DefaultLineLineWidth', 2)

w = logspace(-2, 2, 300); %
figure(5); clf;
figure(6); rlocus(L); rr = rlocus(L, Gc); rr2 = rlocus(L, Gc2); hold on;

w = logspace(-2, 2, 300);
figure(2); clf;
figure(3); rlocus(L); rr = rlocus(L, 1); hold on;
```

September 19, 2010
Summary

- The design summary online is old, but gives an excellent summary of the steps involved.
- Typically find that this process is highly iterative because the final performance doesn’t match the specifications (second order assumptions)
- Practice is the only way to absorb these approaches.