# CDS 101/110a: Lecture 8-2
# Limits on Performance


Richard M. Murray


19 November 2008

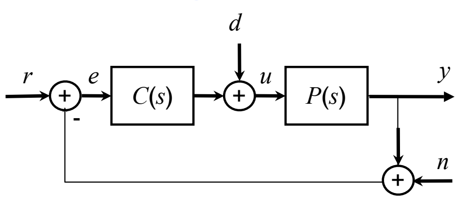
**Goals:**

- Describe limits of performance on feedback systems
- Introduce Bode's integral formula and the "waterbed" effect
- Show some of the limitations of feedback due to RHP poles and zeros

**Reading:**

- Åström and Murray, *Feedback Systems,* Ch 11
- *Advanced*: Lewis, Chapter 9
- CDS 210: DFT, Ch 6

# Algebraic Constraints on Performance




$$H_{er} = \frac{1}{1+PC} =: S$$

Sensitivity function

$$H_{yn} = \frac{PC}{1+PC} =: T$$

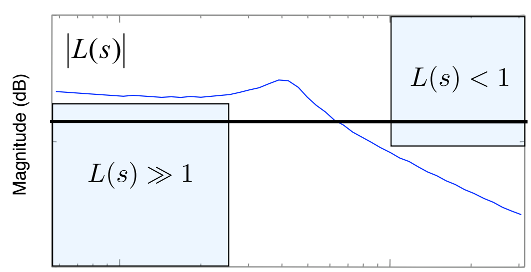
Complementary sensitivity function

**Goal: keep S & T small**

- S small ⇒ low tracking error
- T small ⇒ good noise rejection (and robustness [CDS 110b])

**Problem: S + T = 1**

- Can't make *both* S & T small at the same frequency
- Solution: keep S small at low frequency and T small at high frequency
- Loop gain interpretation: keep L large at low frequency, and small at high frequency



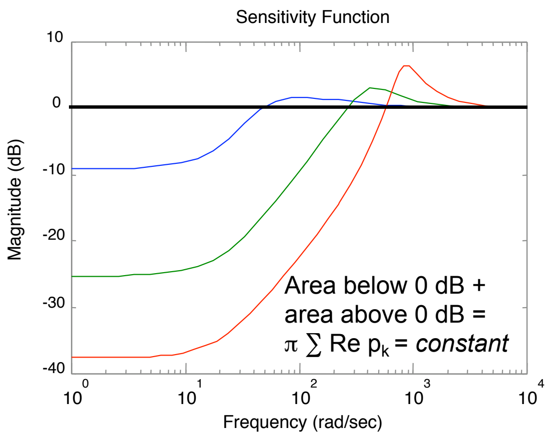

- Transition between large gain and small gain complicated by stability (phase margin)

# Bode's Integral Formula and the Waterbed Effect

**Bode's integral formula for S = 1/(1+PC) = 1/(1+L):**

- Let $p_k$ be the *unstable* poles of $L(s)$ and assume relative degree of $L(s) \geq 2$
- **Theorem**: the area under the sensitivity function is a conserved quantity:

$$\int_0^\infty \log_e |S(j\omega)| d\omega = \int_0^\infty \log_e \frac{1}{|1+L(j\omega)|} d\omega = \pi \sum \text{Re}\, p_k$$

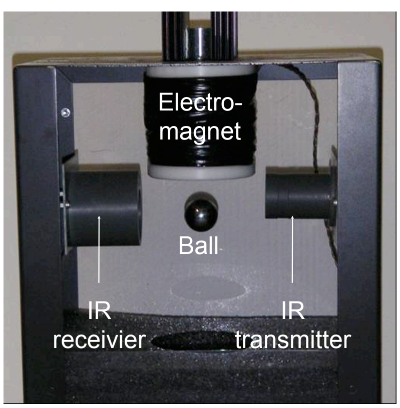



**Waterbed effect:**

- Making sensitivity smaller over some frequency range requires *increase* in sensitivity someplace else
- Presence of RHP poles makes this effect worse
- Actuator bandwidth further limits what you can do
- Note: area formula is linear in $\omega$; Bode plots are logarithmic

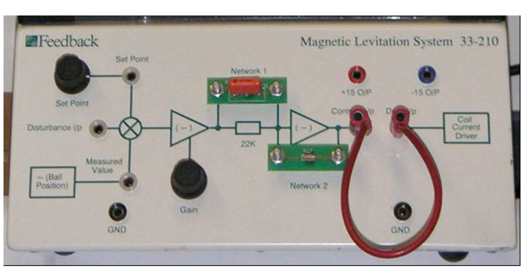


# Example: Magnetic Levitation




**System description**

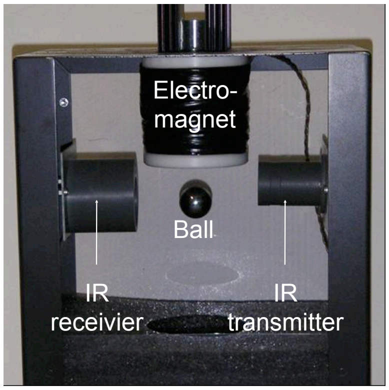
- Ball levitated by electromagnet
- Inputs: current thru electromagnet
- Outputs: position of ball (from IR sensor)
- States: $z, \; \dot{z}$
- Dynamics: *F* = *ma, F* = magnetic force generated by wire coil
- See MATLAB handout for details




**Controller circuit**

- Active R/C filter network
- Inputs: set point, disturbance, ball position
- States: currents and voltages
- Outputs: electromagnet current

# Equations of Motion




**Process: actuation, sensing, dynamics**

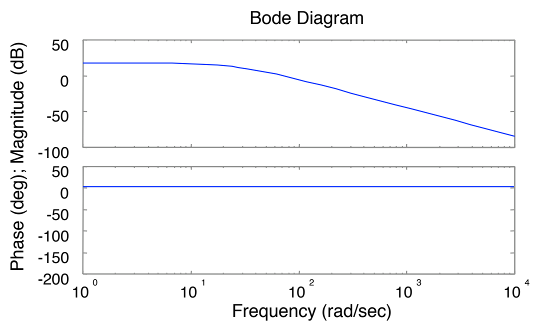
$$m\ddot{z} = mg - k_m(k_A u)^2/z^2$$
$$v_{ir} = k_T z + v_0$$

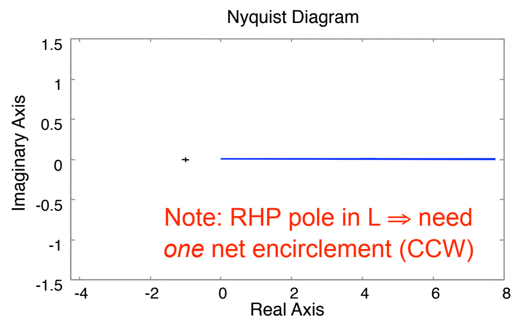
- $u$ = current to electromagnet
- $v_{ir}$ = voltage from IR sensor

**Linearization:**

$$P(s) = \frac{-k}{s^2 - r^2} \qquad k, r > 0$$

- Poles at $s = \pm r \Rightarrow$ open loop unstable






# Control Design

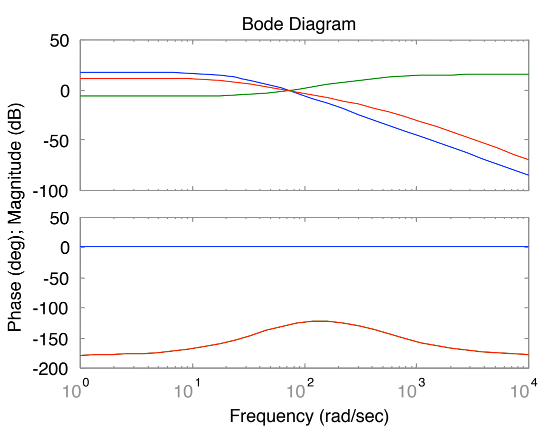
**Need to *create* encirclement**

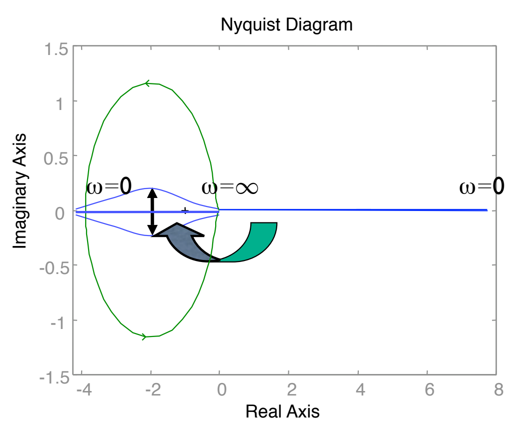
- Loop shaping is not useful here
- Flip gain to bring Nyquist plot over -1 point
- Insert phase to create CCW encirclement

**Can accomplish using a lead compensator**

- Produce phase lead at crossover
- Generates loop in Nyquist plot

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



# Performance Limits

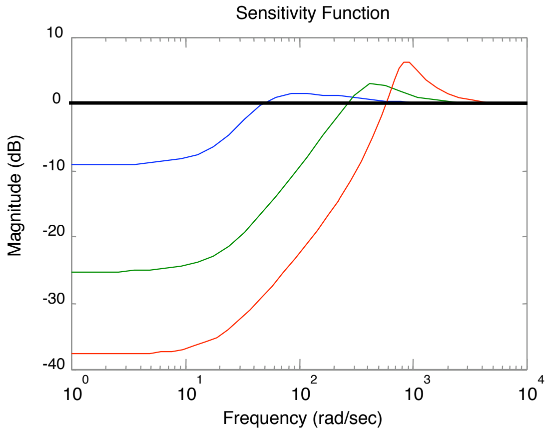
**Nominal design gives low perf**

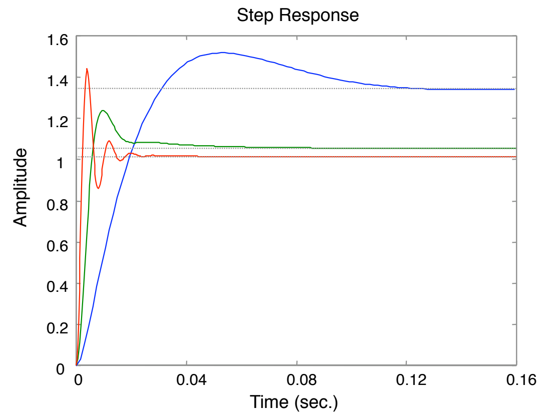
- Not enough gain at low frequency
- Try to adjust overall gain to improve low frequency response
- Works well at moderate gain, but notice waterbed effect

**Bode integral limits improvement**

$$\int_0^\infty \log |S(j\omega)| d\omega = \pi r$$

- Must increase sensitivity at some point



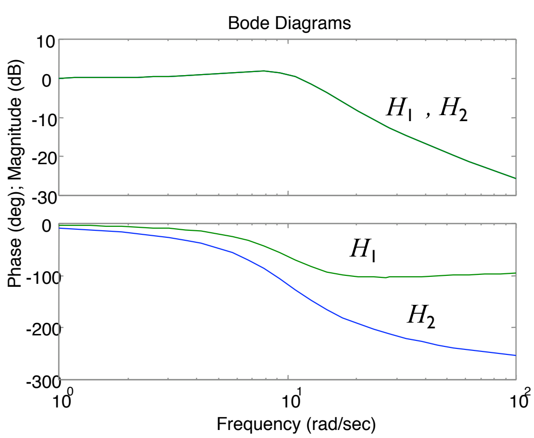


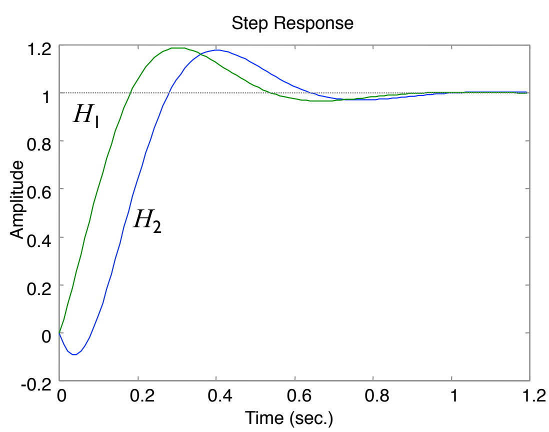

# Right Half Plane Zeros

**Right half plane zeros produce "non-minimum phase" behavior**

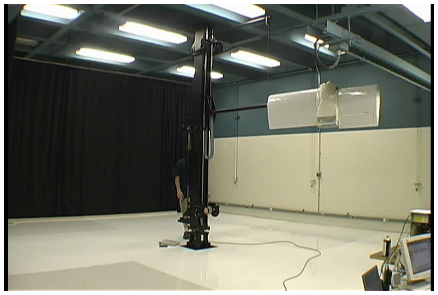
- Phase of frequency response has additional phase lag for given magnitude
- Can cause output to move *opposite* from input for a short period of time

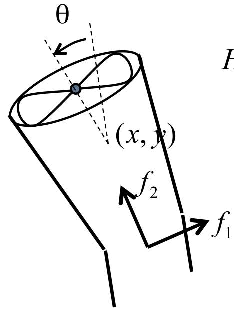
**Example:** $H_1(s) = \dfrac{s+a}{s^2 + 2\zeta\omega_n s + \omega_n^2}$ **vs** $H_2(s) = \dfrac{s-a}{s^2 + 2\zeta\omega_n s + \omega_n^2}$



# Example: Lateral Control of the Ducted Fan



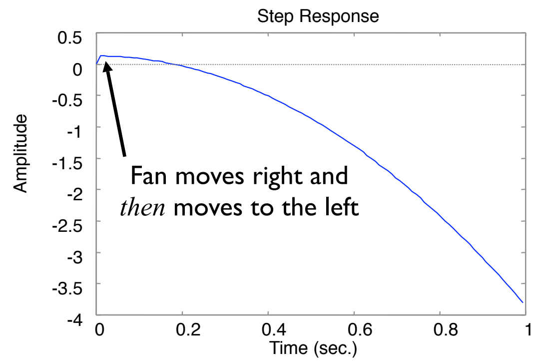



$$H_{xf_1}(s) = \frac{(s^2 - mgl)}{s^2(Js^2 + ds + mgl)}$$

- Poles: 0, 0, $-\sigma \pm j\,\omega_d$
- Zeros: $\pm\sqrt{mgl}$

**Source of non-minimum phase behavior**

- To move left, need to make $\theta > 0$
- To generate positive $\theta$, need $f_1 > 0$
- Positive $f_1$ causes fan to move *right* initially
- Fan starts to move left after short time (as fan rotates)






# Stability in the Presence of Zeros

**Loop gain limitations**

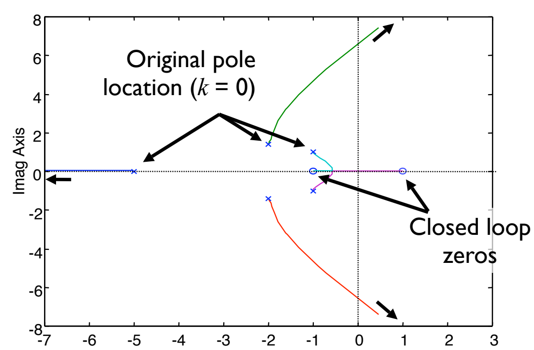
- Poles of closed loop = poles of 1 + L. Suppose $C = k\, n_c/d_c$, where $k$ is the gain of the controller

$$1 + L = 1 + k\frac{n_c n_p}{d_c d_p} = \frac{d_c d_p + k n_c n_p}{d_c d_p}$$

- For large k, closed loop poles approach open loop zeros
- RHP zeros limit maximum gain ⇒ serious design constraint!

**Root locus interpretation**

- Plot location of eigenvalues as a function of the loop gain k
- Can show that closed loop poles go from open loop poles (k = 0) to open loop zeros (k = \infty)



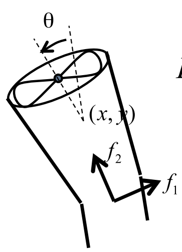
## Additional performance limits due to RHP zeros

**Another waterbed-like effect:** look at maximum of $H_{er}$ over frequency range:

$$M_1 = \max_{\omega_1 \le \omega \le \omega_2} |H_{er}(j\omega)| \qquad M_2 = \max_{0 \le \omega \le \infty} |H_{er}(j\omega)|$$

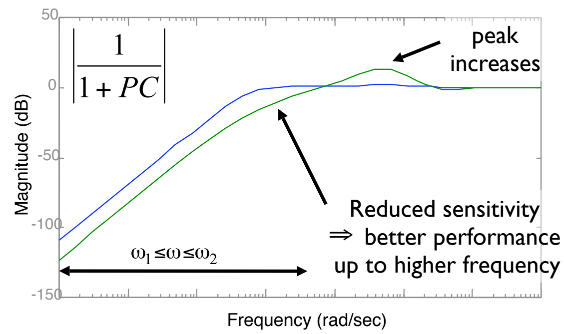
**Thm:** Suppose that $P$ has a RHP zero at $z$. Then there exist constants $c_1$ and $c_2$ (depending on $\omega_1$, $\omega_2$, $z$) such that $\quad c_1 \log M_1 + c_2 \log M_2 \ge 0$

- $M_1$ typically << 1 $\Rightarrow$ $M_2$ must be larger than 1 (since sum is positive)
- If we increase performance in active range (make $M_1$ and $H_{er}$ smaller), we must lose performance ($H_{er}$ increases) some place else
- Note that this affects *peaks* not integrals (different from RHP poles)




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

- Poles: 0, 0, $-\sigma \pm j\,\omega_d$
- Zeros: $\pm\sqrt{mgl}$





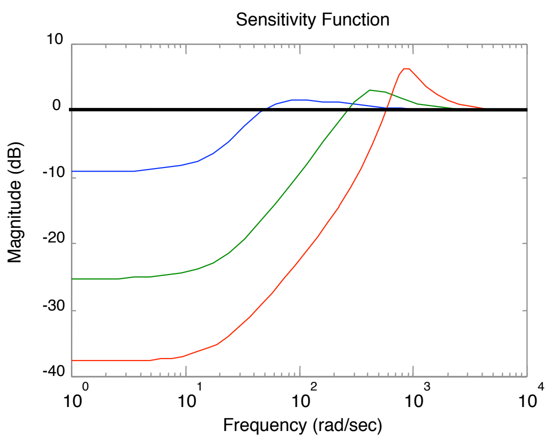

## Summary: Limits of Performance

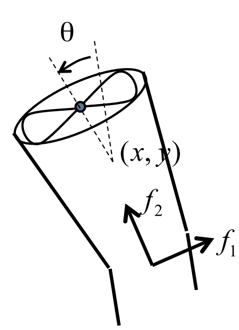
**Many limits to performance**

- Algebraic: S + T = 1
- RHP poles: Bode integral formula
- RHP zeros: Waterbed effect on peak of S

**Main message: try to avoid RHP poles and zeros whenever possible (eg, re-design)**

$$\int_0^\infty \log_e |S(j\omega)| d\omega = \int_0^\infty \log_e \frac{1}{|1 + L(j\omega)|} d\omega = \pi \sum \mathrm{Re}\, p_k$$




```
% L9_1_maglev.m - MATLAB code for lecture 9.1
%
% RMM, 24 Nov 03
%
% Files needed: none
%

%%
%% Magnetic Levitation System
%%
%%   z1 = position of ball from centerline (down is positive)
%%   z2 = velocity of ball
%%   Vy = output (position sensor) voltage
%%   Vu = input (electromagnet) voltage
%%
%% The dynamics are of the form xdot = f(x, u), y = h(x):
%%
%%        [           z2           ]
%%    f = [ g-km/m*(kA*Vu)^2/z1^2 ]
%%
%%    h = kT*z1+v0
%%

% Parameter values for system
kT = 613.65;                    % gain between position and voltage
v0 = -16.18;                    % voltage offset at zero position
m = 0.2;                        % mass of ball, kg
g = 9.81;                       % gravitational constant
kA = 1;                         % electromagnet conductance
km = 3.13e-3 * m/2 / kA^2;      % gain on magnetic attractive force

% Equilibrium point calculation
z10=-v0/kT;
z20=0;
Vu0=1/kA*sqrt(z10^2*m*g/km);

% Compute linearization around the equilibrium
A = [0, 1; 2*km/m*(kA*Vu0)^2/(z10^3), 0];
B = [0; -2*km/m*(kA^2)*Vu0/(z10^2)];
C = [kT 0];
D = 0;

% create a MATLAB LTI system from this representation
magP = ss(A, B, C, D);

%%
%% Maglev controller circuit
%%
%% Note: these values are only approximately equivalent to what exists
%% in the circuit.
%%

k1 = 0.5;                               % gain set by gain pot
R1 = 22000;                             % Internal resistor
R2 = 22000;                             % Resistor plug-in
R = 2000; C = 1e-6;                     % RC plug-in

% Controller based on analog circuit
magC1 = -tf([(R1+R)*C 1], [R*C 1])*k1*R2/R1;

magL1 = magP*magC1;                     % loop transfer function
magS1 = feedback(1, magP*magC1);        % sensitivity function
magT1 = feedback(magP*magC1, 1);        % closed loop response
figure(1); nyquist(magP, magL1);        % Nyquist plots
figure(2); bode(magP, magC1, magL1);    % Open loop Bode plot

%%
%% Now try to improve the performance by increasing DC gain
%%
```



```
magC2 = magC1*5;                        % increased gain
magL2 = magP*magC2;                     % loop transfer function
magS2 = feedback(1, magP*magC2);        % sensitivity function
magT2 = feedback(magP*magC2, 1);        % closed loop response

magC3 = magC1*20;                       % increased gain
magL3 = magP*magC3;                     % loop transfer function
magS3 = feedback(1, magP*magC3);        % sensitivity function
magT3 = feedback(magP*magC3, 1);        % closed loop response

figure(3); step(magT1, magT2, magT3);   % Step response
figure(4); bode(magS1, magS2, magS3);   % Closed loop Bode plot
```