LAG-LEAD COMPENSATOR DESIGN IN FREQUENCY DOMAIN

Table of Contents

AME 451 : Linear Control Systems I

Lead Compensator

Used to improve stability margins and transient response parameters Will accentuate high-frequency noise effects Has no influence on the steady-state accuracy Increases Bandwidth Requires an additional increase in gain to offset the attenuation inherent in the lead network ===> Will require larger gain, which implies larger space, greater weight, and higher cost

Lag Compensator

Used to improve steady-state accuracy and to suppress the high-frequency noise signals Increases the transient response time (system becomes slower) Decreases bandwidth Reduces the system gain at higher frequencies without reducing the system gain at lower frequencies Tends to integrate the input signal ==> makes the system less stable(need to ensure that its time constant is sufficiently large than the largest time constant of the system) may cause conditional stability when a system has saturation or limitation

Lag-Lead Compensator

Used when both fast response, larger stability margins, larger bandwidths, and good steady-state accuracy are desired Increases system order by 2, making it more difficult to control the transient response behavior

Example

System is defined with its transfer function Go(s) = 1 / s(s+1)(s+2) Design requirements: 1. (e(∞))ramp = 10% 2. Gain margin Kg ≥ 10 dB 3. Phase Margin γ ≥ 50°

Definition

Lag-Lead Controller: Gc(s) = Kc*((1+(T_1)*s)/(γ+(T_1)*s))((1+(T_2)*s)/(1+β*T_2*s)) where $\beta > 1$, and $\gamma > 1$

Assumption: β = γ

Then the controller becomes: Gc(s) = Kc * ((1+(T_1)*s)/(1+(T_1/β)*s))*((1+(T_2)*s)/(1+β*T_2*s)) where the first term ((1+(T_1)*s)/(1+(T_1/β)*s)) produces the effect of the lead network, and the second term ((1+(T_2)*s)/(1+β*T_2*s)) produces the effect of the lag network.

Step 1 - Satisfy Steady-State Error Requirement

```
s = tf('s');
Go = 1/(s*(s+1)*(s+2));  % Original system

% Compute velocity error constant
Kv_current = dcgain(s*Go);
Kv_required = 1/0.1;     % For 10% steady-state error
k = Kv_required/Kv_current;

fprintf('Step 1 Results:\nKv_current = %.2f\n', Kv_current);
fprintf('Kv_required = %.1f\n', Kv_required);
fprintf('Required gain k = %.2f\n\n', k);
```

```
Step 1 Results:
Kv_current = 0.50
Kv_required = 10.0
Required gain k = 20.00
```

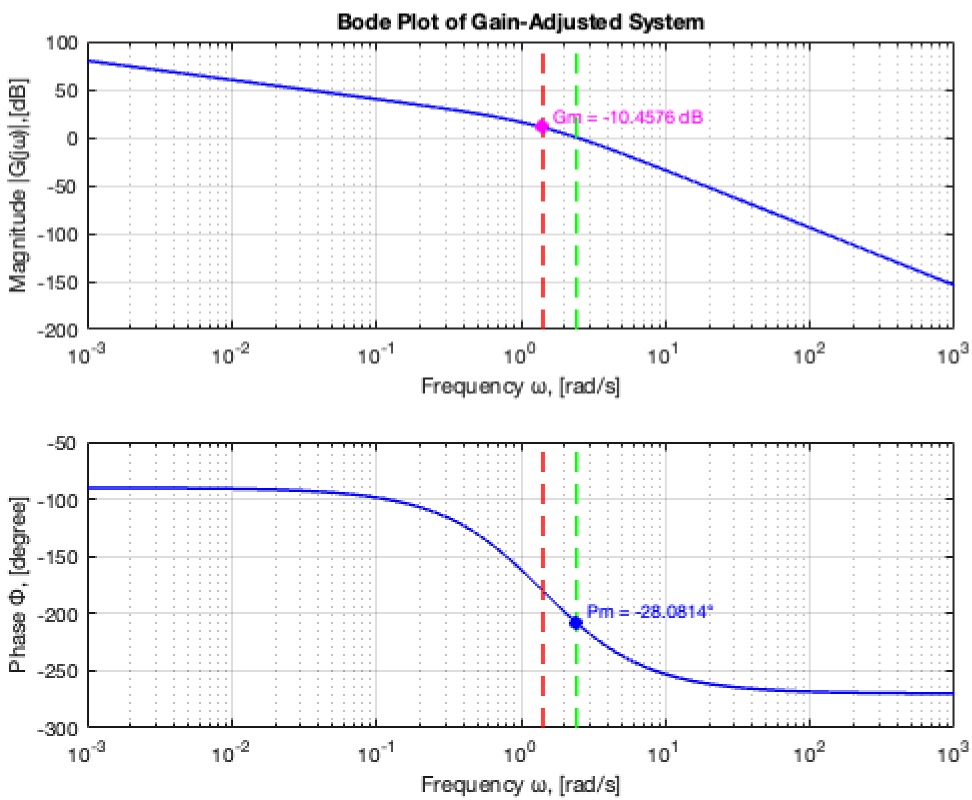
Step 2 - Analyze Gain-Adjusted System

```
Gk = k*Go;
[Gm_initial,Pm_initial,Wcp_initial,Wcg_initial] = margin(Gk);

w = logspace(-3,3,1000); % Adjust the frequency scale to designer's choice.


% Extract magnitude, phase, and frequency information from MATLAB's
% inbuilt bode generator function.

[mag,phase,wout] = bode(Gk,w);

mag = squeeze(mag);
phase = squeeze(phase);

% Calculate Stability Margins
GmdB = 20*log10(Gm_initial);
```

```
figure;
% Plot Bode magnitude
subplot(2,1,1);
semilogx(wout,20*log10(mag),'b','LineWidth',1.5); grid on;
xlabel('Frequency ω, [rad/s]');
ylabel('Magnitude |G(jω)|,[dB]')
title('Bode Plot of Gain-Adjusted System')

% Plot Bode phase
subplot(2,1,2);
semilogx(wout, phase, 'b', 'LineWidth', 1.5); grid on;
xlabel('Frequency ω, [rad/s]');
ylabel('Phase Φ, [degree]');


% Draw vertical lines at Wcp and Wcg
subplot(2,1,1); % Magnitude plot
hold on;
yl = ylim;
plot([Wcp_initial Wcp_initial], yl, 'r--', 'LineWidth', 1.5); % Phase
crossover (red)
plot([Wcg_initial Wcg_initial], yl, 'g--', 'LineWidth', 1.5); % Gain
crossover (green)
hold off;

subplot(2,1,2); % Phase plot
hold on;
yl = ylim;
plot([Wcp_initial Wcp_initial], yl, 'r--', 'LineWidth', 1.5); % Phase
crossover (red)
plot([Wcg_initial Wcg_initial], yl, 'g--', 'LineWidth', 1.5); % Gain
crossover (green)
hold off;

% Annotate Gain Margin on magnitude plot
subplot(2,1,1);
hold on;
plot(Wcp_initial, -GmdB, 'mo', 'MarkerFaceColor', 'm');
text(Wcp_initial, -GmdB, sprintf('  Gm = %.4f dB', GmdB), 'Color', 'm', ...
    'VerticalAlignment', 'bottom', 'FontSize', 9);
hold off;

% Annotate Phase Margin on phase plot
subplot(2,1,2);
hold on;
plot(Wcg_initial, -180+Pm_initial, 'bo', 'MarkerFaceColor', 'b');
text(Wcg_initial, -180+Pm_initial, sprintf('  Pm = %.4f°', Pm_initial),
'Color', 'b', ...
    'VerticalAlignment', 'bottom', 'FontSize', 9);
hold off;

% Display calculated values
disp(['Gain Margin (dB): ', num2str(GmdB)]);
disp(['Phase Margin (deg): ', num2str(Pm_initial)]);
```

```
disp(['Phase crossover frequency (rad/s): ', num2str(Wcp_initial)]);
disp(['Gain crossover frequency (rad/s): ', num2str(Wcg_initial)]);

fprintf('Step 2 Results:\nPhase Crossover Freq = %.4f rad/s\n', Wcg_initial);
fprintf('Gain Margin = %.4f (%.2f dB)\n', Gm_initial, 20*log10(Gm_initial));
fprintf('Gain Crossover Freq = %.4f rad/s\n', Wcp_initial);
fprintf('Phase Margin = %.4f degrees\n\n', Pm_initial);
```

```
Warning: The closed-loop system is unstable.
Gain Margin (dB): -10.4576
Phase Margin (deg): -28.0814
Phase crossover frequency (rad/s): 1.4142
Gain crossover frequency (rad/s): 2.4253
Step 2 Results:
Phase Crossover Freq = 2.4253 rad/s
Gain Margin = 0.3000 (-10.46 dB)
Gain Crossover Freq = 1.4142 rad/s
Phase Margin = -28.0814 degrees
```



Step 3 - Design Lag Compensator

```
phi_m = 55;  % Phase margin + 5° safety factor
beta = (1 + sind(phi_m))/(1 - sind(phi_m));
wg_new = 1.5;  % New gain crossover frequency aprroximately 1.5 from 1.4142
z_lag = wg_new/10;
```

```
T2 = 1/z_lag;
p_lag = 1/(beta*T2);

Glag = (s + z_lag)/(s + p_lag);

fprintf('Step 3 Results:\nβ = %.4f\n', beta);
fprintf('T2 = %.4f\n', T2);
fprintf('Z_lag = %.4f\n', z_lag);
fprintf('P_lag = %.4f\n\n', p_lag);
```

```
Step 3 Results:
β = 10.0590
T2 = 6.6667
Z_lag = 0.1500
P_lag = 0.0149
```

Step 4 - Design Lead Compensator

From Bode plot analysis at wg = 1.5 rad/s

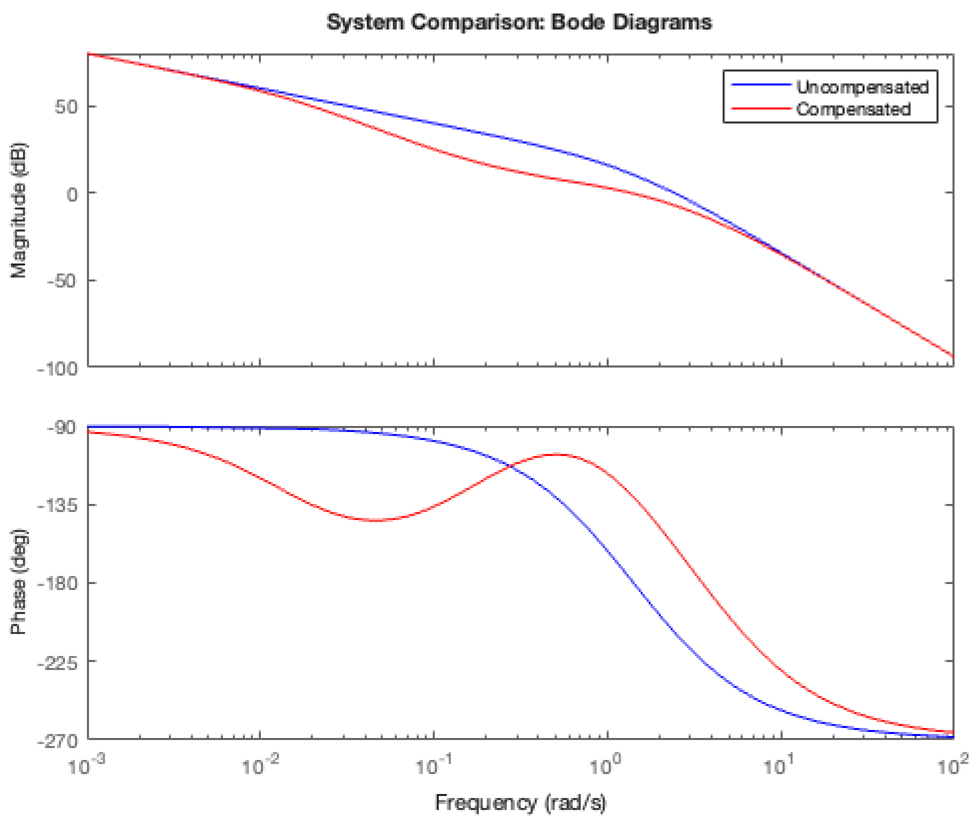
```
z_lead = 0.5;
p_lead = 5;
Glead = (s + z_lead)/(s + p_lead);
```

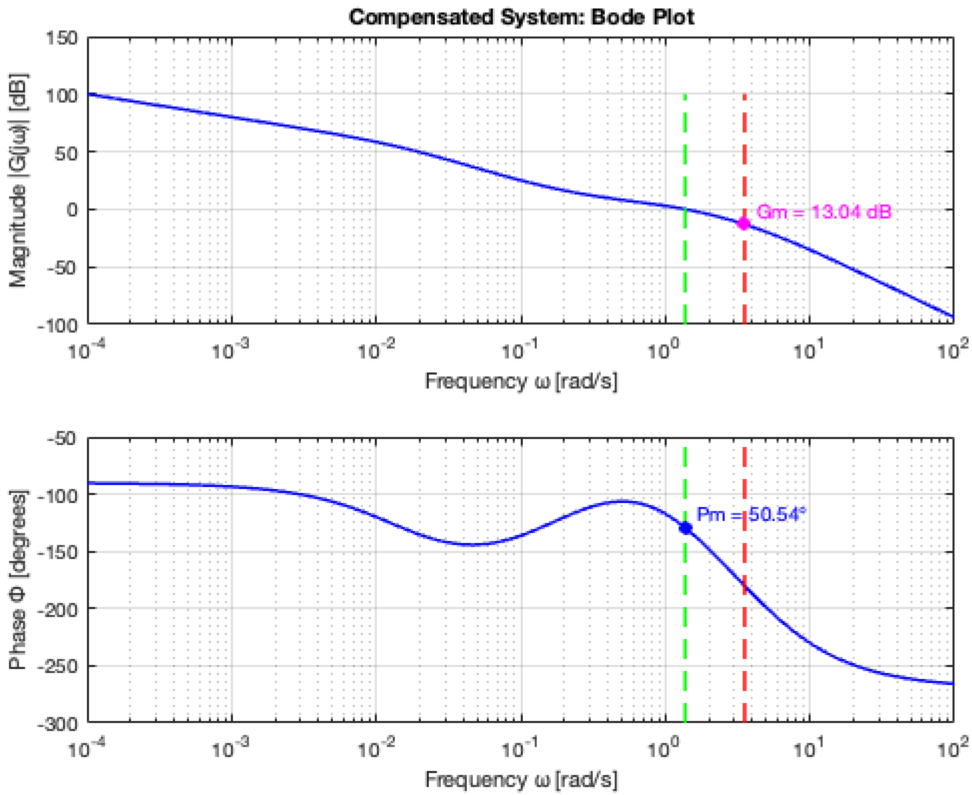
Combine Compensators and Verify

```
Gc = Glag * Glead;
G_total = Gc * Gk;

% Get stability margins for FINAL system
[Gm_final,Pm_final,Wcp_final,Wcg_final] = margin(G_total);

% Get Bode data for plotting
[mag_total, phase_total, wout] = bode(G_total);
mag_total = squeeze(mag_total);
phase_total = squeeze(phase_total);

% Comparison plot (original vs compensated)
figure;
bode(Gk, 'b', G_total, 'r');
legend('Uncompensated', 'Compensated');
title('System Comparison: Bode Diagrams');

% Detailed annotated plot (compensated system)
figure;
% Plot Bode magnitude
subplot(2,1,1);
semilogx(wout, 20*log10(mag_total), 'b', 'LineWidth', 1.5);
grid on;
xlabel('Frequency ω [rad/s]');
ylabel('Magnitude |G(jω)| [dB]');
title('Compensated System: Bode Plot');
```

```
% Plot Bode phase
subplot(2,1,2);
semilogx(wout, phase_total, 'b', 'LineWidth', 1.5);
grid on;
xlabel('Frequency ω [rad/s]');
ylabel('Phase Φ [degrees]');

% Add crossover lines and annotations
for i = 1:2
    subplot(2,1,i);
    hold on;
    yl = ylim;
    plot([Wcp_final Wcp_final], yl, 'r--', 'LineWidth', 1.5);  % Phase
crossover
    plot([Wcg_final Wcg_final], yl, 'g--', 'LineWidth', 1.5);  % Gain
crossover
    hold off;
end

% Annotate Gain Margin
subplot(2,1,1);
hold on;
gm_dB = 20*log10(Gm_final);
plot(Wcp_final, -gm_dB, 'mo', 'MarkerFaceColor', 'm');
text(Wcp_final, -gm_dB, sprintf('  Gm = %.2f dB', gm_dB),'Color', 'm',
'VerticalAlignment', 'bottom');
hold off;

% Annotate Phase Margin
subplot(2,1,2);
hold on;
plot(Wcg_final, -180 + Pm_final, 'bo', 'MarkerFaceColor', 'b');
text(Wcg_final, -180 + Pm_final, sprintf('  Pm = %.2f°', Pm_final),'Color',
'b', 'VerticalAlignment', 'bottom');
hold off;


fprintf('Step 4 Results:\nFinal Gain Margin = %.2f dB\n', 20*log10(Gm_final));
fprintf('Final Phase Margin = %.2f degrees\n', Pm_final);
fprintf('Phase Crossover Freq = %.2f rad/s\n', Wcg_final);
fprintf('Gain Crossover Freq = %.2f rad/s\n\n', Wcp_final);

G_closed = feedback(G_total,1);
[mag,phase,w] = bode(G_closed);
mag_db = 20*log10(squeeze(mag));
[peak_gain, peak_idx] = max(mag_db);
peak_freq = w(peak_idx);
bw = bandwidth(G_closed);

fprintf('Closed-Loop Characteristics:\n');
fprintf('Resonant Peak: %.2f dB at %.2f rad/s\n', peak_gain, peak_freq);
fprintf('Bandwidth: %.2f rad/s\n', bw);
```

```
Step 4 Results:
Final Gain Margin = 13.04 dB
Final Phase Margin = 50.54 degrees
Phase Crossover Freq = 1.38 rad/s
Gain Crossover Freq = 3.52 rad/s

Closed-Loop Characteristics:
Resonant Peak: 1.53 dB at 1.62 rad/s
Bandwidth: 2.47 rad/s
```





Published with MATLAB® R2024b