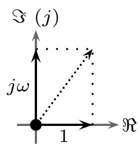
ECE 557: *Control, Signals, and Systems Laboratory*

Notes for Lab 6 (Lead Compensation for Position Control of a DC Servo)

1. Return lag compensation pre-lab and give some notes (biggest problem: incorrect gains).
   - Small detail: Don't compare vectors directly; make your 2-norms explicit.



Complex plane is just $\mathbb{R}^2$ (Euclidean 2-space) with a rotation operator.

$$\frac{j\omega + 1}{j\omega\alpha + 1} \xrightarrow{\omega\to\infty} \text{meaning?}$$

$$\lim_{\omega\to\infty}\left|\frac{j\omega+1}{j\omega\alpha+1}\right| = \underbrace{\lim_{\omega\to\infty}\frac{\sqrt{\omega^2+1}}{\sqrt{(\omega\alpha)^2+1}}}_{\text{Don't skip this step.}} = \frac{1}{|\alpha|} \stackrel{\alpha\geq 0}{=} \frac{1}{\alpha}$$

   - For LTI feedback problems, use `margin` in MATLAB instead of `bode` (helps check your work).
   - When lag compensator is for **improving phase margin**, want *unity* **DC** compensator gain.

$$G_{c_{\text{PM}}}(s) = \underbrace{\frac{1}{\alpha}\frac{s+a}{s+b}}_{\substack{\text{Pole-zero(-gain)}\\ \text{form}}} = \frac{b}{a}\frac{s+a}{s+b} = \underbrace{\frac{1+\frac{s}{a}}{1+\frac{s}{b}}}_{\boxed{\substack{\text{Natural-frequency}\\ \text{form}}}} = \begin{cases} 1 & \text{@ DC,} \\ \frac{1}{\alpha} & \text{@ AC} \end{cases} \qquad (\alpha > 1, \quad a > b)$$

We *attenuate* high frequencies for stability margins and less ringing. Make sure implementation has correct gain. It makes sense to use **natural-frequency form** here.

- If you forget the $1/\alpha$ gain, the additional amplification reduces your desired stability margins.
- In the pre-lab, leaving out the $1/\alpha$ gain results in no change in phase margin and a decrease in steady-state error.
- Notice the *pole-zero(-gain)* and *natural-frequency* forms of the same transfer function carry *different gains.*

The α is a gain. It should *not* be in deciBel units. Solve raw or convert plot deciBels.

- To increase phase margin, design compensator to provide $\alpha \triangleq 1/G_{\text{plant}}$ gain at some ω_c.
- $G_{c_{\text{PM}}} G_{\text{plant}}$ system will have unity gain at ω_c (i.e., $1/G_{\text{plant}}(\omega_c) \times G_{\text{plant}}(\omega_c) = 1$).
- Pick $a \ll \omega_c$ so that compensator looks like constant AC gain.
  - $\angle(G_{c_{\text{PM}}}(\omega_c)\, G_{\text{plant}}(\omega_c)) = \angle G_{c_{\text{PM}}}(\omega_c) + \angle G_{\text{plant}}(\omega_c) \approx \angle G_{\text{plant}}(\omega_c)$.
  - So long as lag compensator concentrates its effect near DC, we only care about its *gain.*

   - When lag compensator is for **reducing steady-state error**, want *unity* **AC** compensator gain.

$$G_{c_{\text{SS}}}(s) = \underbrace{\frac{s+a}{s+b}}_{\boxed{\substack{\text{Pole-zero(-gain)}\\ \text{form}}}} = \frac{s+\alpha b}{s+b} = \underbrace{\alpha\frac{1+\frac{s}{\alpha b}}{1+\frac{s}{b}}}_{\substack{\text{Natural-frequency}\\ \text{form}}} = \begin{cases} \alpha & \text{@ DC,} \\ 1 & \text{@ AC} \end{cases} \qquad (\alpha > 1, \quad a > b)$$

We *amplify* DC (i.e., position error) response to reduce steady-state error. Make sure implementation has correct gain. It makes sense to use the **pole-zero(-gain) form** here.

- The α is only needed in the *natural frequency form.* Not including it removes any improvement in steady-state error.
- Putting the α in the *pole-zero(-gain)* form causes control signal to escape constraints.

Remember to calculate expected steady-state error. Include α, K_c, and K_p.

$$\widetilde{K}_p \triangleq \lim_{s\to 0}\left(G_{c_{\text{SS}}}(s)\, G_{\text{plant}}(s)\right) = G_{c_{\text{SS}}}(0)\, K_p = \alpha\, K_c\, K_p \qquad \text{and} \qquad e_{\text{SS}} = \frac{1}{1+\widetilde{K}_p} = \frac{1}{1+\alpha\, K_c\, K_p}$$

Increasing K_c will decrease e_{SS}, but it reduces gain margin and increases control effort. The extra boost from pole-zero compensator α gives steady-state leverage on a small K_c.

2. Return gain compensation lab reports and give some notes.
   - When showing step-response data, change time and data axes to focus on interesting region.
     - No need to show entire capture time.
     - Put some space above data maximum to show you're not cutting anything off.
   - Few people listed pre-lab results for comparison.
     - Given your pre-lab controller designs, it's easy to re-generate your pre-lab data.
     - Compare measured data with expected data. Explain differences.
   - Voltage drives motor *speed*, and *position* is fed back. So steady-state output must have *zero* error.
     - Otherwise motor speed would be nonzero and position would be change (system has type 1).
     - All reports showed *nonzero* steady-state error, but none found that remarkable.
       * Quantization? Friction?
       * Other reasons why motor would be steady while its input is nonzero?
3. Fun math from first part of pre-lab assignment.
   (a) Unconstrained maximization of phase angle (i.e., use calculus).
   - Let
   $$f'(\omega) \triangleq \frac{\partial}{\partial\omega} f(\omega) \quad \text{and} \quad f''(\omega) \triangleq \frac{\partial^2}{\partial\omega^2} f(\omega).$$
   - If *both* $f'(\omega_m) = 0$ and $f''(\omega_m) < 0$ then ω_m must be a maximum of $f(\omega)$.
   - The sign of $f''(\omega_m)$ determines whether ω_m is a maximum or a minimum.
     - Its sign is negative when $a < b$ (lead case).
     - Its sign is positive when $a > b$ (lag case).

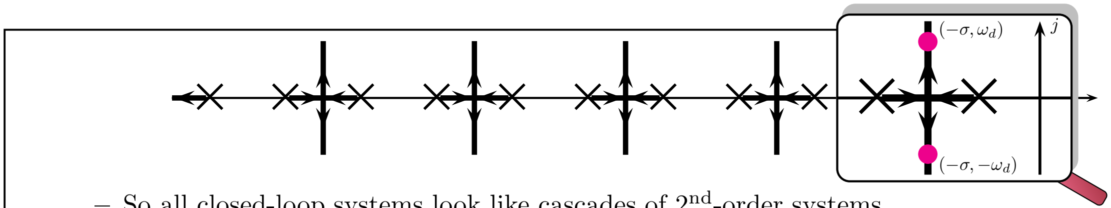
     So $\omega_m = \sqrt{ab}$ also represents frequency of ***minimum*** *phase* in *lag compensation*.
   - The expression $\sqrt{ab}$ is the *geometric mean* of a and b.
     - Generally, the **geometric mean** of x_1, x_2, x_3, …, and x_n is $\sqrt[n]{x_1\, x_2\, x_3 \cdots x_n}$.
     - Compare to the **arithmetic mean** of x_1, x_2, …, and x_n, which is $(x_1 + x_2 + \cdots + x_n)/n$.
     - For a list of non-negative real numbers, $(x_1 + x_2 + \cdots + x_n)/n \geq \sqrt[n]{x_1\, x_2 \cdots x_n}$.
       * Equality only occurs when all numbers in the list are the same.
       * The *AM–GM inequality* is a useful tool. Remember it.

   (b) Don't worry; it's bonus.
   - Geometric solution (trigonometric identities — sine, tangent, cosine, and triangles):
     (i) Draw *right triangles* corresponding to $\operatorname{atan}(x/1)$ (i.e., $\tan^{-1}(x/1)$) and $\operatorname{atan}(1/x)$.
       - The relationship between these two **angles** should be obvious.

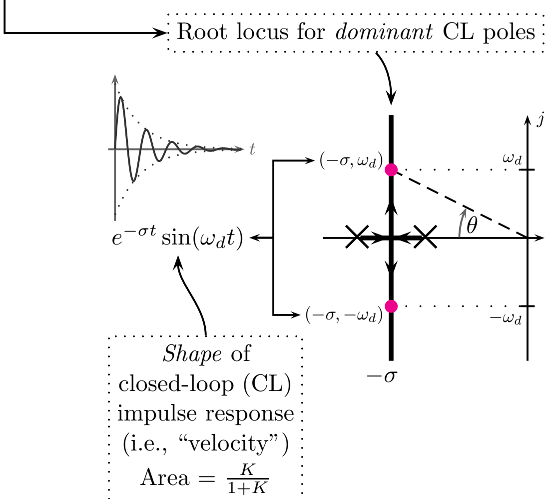
     (ii) Recall that $\cos(2x) = 2\cos^2(x) - 1$ (think about Fourier transformation of $\cos^2(x)$).
     (iii) Go back to the triangles to figure out $\cos(\operatorname{atan}(x))$.
   - Algebraic solution:
     (i) Recall that $G_c(j\omega_m) = |G_c(j\omega_m)|\, e^{j\phi_m}$ (and $e^{j\phi_m} = \cos(\phi_m) + j\sin(\phi_m)$).
     (ii) Equate imaginary parts (recall how to *rationalize a denominator*).
     (iii) Solve for $\sin(\phi_m)$.

   (c) Going from $20 \log |G_c|$ to $-10 \log \alpha$ (not a tpyo…er…typo).
   - Recall that $\log_{10} \sqrt{x} = \log_{10} x^{0.5} = 0.5 \log_{10} x$.
     - That's why you start with $20 \log_{10}$ and end up with $10 \log_{10}$.
   - The quantity $\sqrt{\alpha}$ can be more useful than α when designing using a computer.

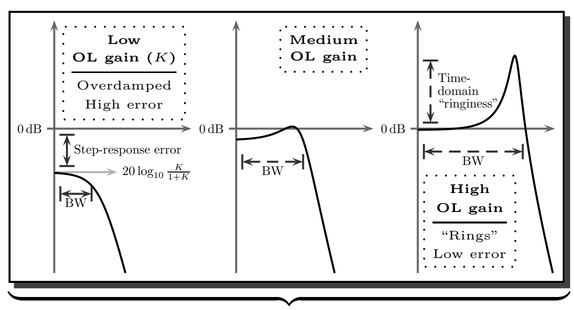
4. Gain adjustments on generic all(-real)-pole feedback system: consider root locus.
   - From DC and moving *left*, each *pair* of open-loop poles generates closed-loop poles between them.



   - So all closed-loop systems look like cascades of 2nd-order systems.
     - ∗ Closed-loop poles move toward each other and hit a *breakaway point* between them.
       - · Quadratic formula discriminants determine breakaway point.
       - · All breakaways are initially vertical and then bend away from other asymptotes.
   - Dominant characteristics come from slowest pair.
     - ∗ 2-pole systems (i.e., 2nd order models) capture much of richness in behavior.
       - · Similar to approximating motion with position, velocity, and acceleration only.
     - ∗ Other poles have some influence on position of breakaway and steepness of initial slope.
       - · They have negligible influence if they are sufficiently far away.
   - To determine control, focus on slowest (i.e., dominant) pair of open-loop poles.



Damping ratio (ζ) = $\cos(\theta)$						
OL Gain (K)		θ		DR (ζ)		Phase Margin
High	$\Longrightarrow$	High	$\Longrightarrow$	Low	$\Longrightarrow$	Low
Medium	$\Longrightarrow$	Medium	$\Longrightarrow$	Medium	$\Longrightarrow$	Medium
Low	$\Longrightarrow$	Low	$\Longrightarrow$	High	$\Longrightarrow$	High



Closed-loop (CL) frequency-magnitude responses

   - The impulse response of the closed-loop system characterizes behavior of transients.
     - ∗ Increasing open-loop gain increases *bandwidth* by making damped frequency ω_d larger. The system becomes more sensitive to faster signals. Step-response **rise time** decreases.
     - ∗ Even though rise time decreases, *settling time* can increase because fast $\sin(\omega_d t)$ can peak many times before $e^{-\sigma t}$ decreases significantly. The system "rings" and settles **slowly.**
     - ∗ The extra "ringiness" corresponds to a low **phase margin**. A small delay will cause instability. So increasing gain moves the system "closer to instability."
     - ∗ So a **fast** and **robustly stable** system needs high gain *and* high damping ratio (ζ).
   - High "ringiness" also brings **dangerous** **control effort** and **step-response overshoot**.
   - Designing by adjusting gain **only** means accepting the root locus as it is.
     - ∗ If we are unhappy with the system characteristics as they are, we must *compensate* for them by adding our own characteristics.
     - ∗ *Compensation* corresponds to *re-shaping* the root locus by *moving* the breakaway point.
     - ∗ *Lag compensation* shifts breakway *right* to reduce steady-state error without high gain.
     - ∗ *Lead compensation* shifts breakaway *left* to make system *settle faster* (i.e., damps ringing).
     - ∗ So compensation increases speed or reduces error without increasing ring.
   - PID combines the three effects: (P,I,D) = (bandwidth [gain], reduced error [lag], speed [lead]).

5. Complete the *Lead Compensation for Position Control of a DC Servo* lab
   - Implement lag compensation for velocity regulation of DC servo.
     - In *Simulink*, the Summing Junction (or *sum*) component is in the Math section of the library.
       - Change the |++ to |+- to make one of the inputs negative.
     - Use Zero-Pole or Transfer Function from Continuous section of *Simulink* library.
       - All poles and zeros are *negative*! Remember to use the correct form and gains!!
     - *If you wish,* wire up a simulated system for comparison. Capture its output as well.
       - You might relate this to using an *observer* (a subject of ECE 650 and ECE 750).
   - Start with your Bode-type controller design from the pre-lab (provides 90° phase margin).
     - ⋆ Make sure your design does *NOT* use α in deciBels (dB)! ($0.4 < \alpha < 0.6$)
     - Gather step response data in *dSPACE ControlDesk*.
   - Change to your root-locus-type controller design from the pre-lab (improves speed).
     - Gather step response data in *dSPACE ControlDesk*.
   - You do not need separate controllers for the slow version of system, but keep slow system in mind when analyzing data in report!
   - ⋆ AT ANY TIME, IF MOTOR STARTS CLICKING VERY QUICKLY, STOP THE EXPERIMENT — DISCONNECT THE MOTOR IF NECESSARY!! High-frequency switching can cause **permanent damage**! It can be caused by unstable systems (e.g., high gains or positive poles).
   - ⋆ There is no manual tuning in this experiment.
   - ⋆ Ideally, this type-1 system (i.e., system with one integrator) should have no steady-state error.
     - We are driving output voltage with *position* error.
       - *Any* nonzero voltage should cause the motor to move (i.e., voltage and speed are related).
       - If the motor is not moving, its speed is zero, and so the output voltage *must* be zero.
       - Because of ***position*** *feedback*, the output voltage is only zero if ***position*** *error* is zero.
       - So the closed-loop system should have **no steady-state error for a step input.**
     - Unfortunately, nonlinearities and time variance in the system couple with quantization noise in the DAC and make the system far less than ideal. Unless gain is very high, steady-state error will be nonzero (e.g., motor **stalls** if it doesn't overcome starting friction).
       - You can use a digital voltmeter (DVM) to verify motor stalls with small nonzero input.
       - So the controller is doing what it *should*, but the motor is not behaving linearly. It has a *dead zone*.

- Tips:
  - Do **work** out of directory on **local** hard drive — use as MATLAB working directory.
  - In *Simulink*, the hotkey for building a model is Ctrl - B .
  - Start *dSPACE ControlDesk* before doing *Simulink* builds.
  - In MATLAB, change *Termination* settings for DAC block — check box to set 0 V stop value.
  - In *dSPACE* add a simState control.
    - (i) Wire simState to 2-option radio button — Setup options "Run" (2) and "Stop" (0).
    - (ii) Set Capture Settings to *automatically* restart and set *capture time* to simulation time.

    Restart simulation as needed by using simState control (i.e., no need to change modes).
    - (i) To stop early, change simState to Stop.
    - (ii) Before restarting, re-initialize Capture Settings by clicking Stop and then Start.
    - (iii) When you're ready to start (e.g., after changing gains), set simState to Run.