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

Notes for Lab 5 (Lag Compensation for Speed Control of a DC Servo)

1. Return gain compensation pre-lab and give some notes.

   - It's nice to calculate error constants while keeping $G_c = K$. Consider

     $$G_f(s) = G_c(s)\,G(s) = \frac{K}{s(0.0026s+0.1081)} \triangleq \frac{K}{s(a\tau s+a)} = K\,\frac{1}{s}\,\frac{1}{(a\tau s+a)} = \overbrace{\frac{K}{a}\,\frac{1}{s}}^{\text{“Gain”}}\,\frac{1}{(\tau s+1)}.$$

     - The system of type 1 because it includes a *one* integrator.
       - In feedback configurations, these systems will eventually track a step input with no error.
       - Put another way, its own internal *undamped* dynamics are identical to the input's. Whenever the error signal is nonzero, it adds energy to the system which reconfigures the "initial" conditions on the undamped dynamics. Eventually those dynamics are synchronized with the input, and the error vanishes. With no error, the system rings continuously.
       - So the system will track *any* input that is eventually *constant* (i.e., any system with a pole at zero and no other undamped (i.e., purely imaginary) poles) – not just steps.
     - Consider *error constants* with K term included. Constants are SS error-to-output *GAINS*.
       - $y_{\text{ss}} = K_p e_{\text{ss}}$, and so $K_p = \infty$ means $e_{\text{ss}} = 0$ if y_{ss} is constant (i.e., error decays to zero).
       - $y'_{\text{ss}} = K_v e_{\text{ss}}$, and so $K_v = K/a$ means $e_{\text{ss}} = 0$ if $y'_{\text{ss}} = 0$ (i.e., again, error decays to zero).
       - $y''_{\text{ss}} = K_a e_{\text{ss}}$, and so $K_a = 0$ means error has no influence on y''_{ss} (no acceleration feedback).

       A type-2 system has $K_a > 0$, and a type-3 system has $K_a = \infty$.
       - Increasing system type increases how many derivatives get infinite gain (i.e., zero error).
         - A system of type n has zero tracking error (i.e., ∞ gain) for first $n-1$ derivatives.
         - Type indicates where finite K/a gain is placed (i.e., finite gain for n^{th} derivative $y^{(n)}$).
         - Can get zero tracking error when *Taylor series* is finite sum (i.e., polynomial in t).
       - Inputs made of sinusoids (i.e., *any* eventually periodic input) cannot be tracked because sinusoids have an infinite number of derivatives – need "type-∞" system.
         - Tracking periodic signals requires a system with its own periodic components.

2. Return time-domain system ID lab reports and give some notes.

   - Step responses had noisy "steady-state" values (due to electromechanical noise).
     - *Constant* input through *stable* closed-loop system will have *constant* output.
     - Use `mean` in MATLAB to filter out (zero-mean) noise. Remember that "DC" means "average."
     - Expected steady-state value is 0.530 and *not* 0.5.
   - Some groups assumed *unit* step size to find K and added 4 V effect later.
   - Method (from book) for finding t_{stop} is bad. Using shaft encoders to find revolution time is better.
     - Small errors in K_{tach} estimation caused large errors in estimated value of K_{motor}.
   - Steady-state values near expectation, but corners all near $\sim$27 rad/s $\approx$ expected 41.6 rad/s $\div$ 1.5.
     - Motor voltage V_m and current I_m described by the first-order equation

       $$V_m \quad = \quad I_m R_m + K_m \omega_m \quad = \quad I_m R_m + \frac{1}{C_m}\int I_m \,\mathrm{d}t \quad = \quad I_m\left(R_m + \frac{1}{sC_m}\right).$$

       Motor acceleration $\dot{\omega}_m$ is proportional to I_m. Capacitance C_m comes from inertia (i.e., *mass*).
     - Book says $R_m = 2.6\,\Omega$, but *static friction nonlinearity* makes $R_{m,\text{fit}} = R_m \times 1.5$ a better fit.

3. Extend time-domain system ID results to other labs.
   - For **gain compensation lab**, nominal K shifts from 2.25 to 1.5.
     - Slower plant is sluggish and tends to overshoot. Reduce K to prevent large swings.
   - For **lag compensation lab**, Bode design's zero and pole are both 1.5× smaller.
   - When calculating parameters to use in laboratory experiments, use the book's ideal motor model
     $$\frac{\Omega(s)}{V(s)} = \frac{1}{0.0026s + 0.1081} = 9.25\,\frac{41.58}{s + 41.58} = 9.25\,\frac{1}{0.024s + 1}$$
     and the slower power-limited model
     $$\frac{\Omega(s)}{V(s)} = \frac{1}{0.0039s + 0.1081} = 9.25\,\frac{27.72}{s + 27.72} = 9.25\,\frac{1}{0.036s + 1}.$$
     There is no need to submit pre-lab plots for the slow model. However, you should get better performing controllers in class if you have slower designs ready. Additionally, you'll be able to explain variations in your lab report data using the slower model.

4. Some comments on lab reports for this class.
   - Your post-lab submissions are *reports*. They are not simply answers to the post-lab exercises. The lab exercises are guidance for the content of your *report*.
     - The lab reports are explicitly mentioned in the ECE 557's ABET accreditation.
       - ABET is the Accreditation Board for Engineering and Technology.
       - Sample copies of student work are made of every assignment.
       - During ABET audits, those samples are reviewed.
     - Your reports should *not* include a list of the post-lab exercises and responses.
       - As a *whole*, your document should include responses to the post-lab exercises.
         - Do *not* number your responses. Keep everything in paragraph form.
         - Use transitions. More importantly, make educated connections.
       - Your document should be a fluid discussion of your measured and expected data.
       - Your document should have a format suitable for professional use (e.g., a *technical report*).
       - Your document should report on an *experiment* that you conducted for some *purpose*.
         - The purpose of each experiment is *not* to follow the steps in the book.
         - *Purpose* describes *what you expect to learn or gain*.
         - When writing, imagine that the class (e.g., book, assignments, etc.) does not exist.
   - *ADMIT* when your results don't match your expectations. Try to explain why.
     - This isn't ECE 209, and the instructor is not an idiot.
       - Unexpected outcomes are *educational features* of laboratories. Do not explain them away with "*inaccuracies happen.*"
       - Avoid blanket explanations and loaded words like *noise*.
         - "Noise" often refers to *random* uncorrelated (i.e., "white") zero-mean signal variations.
         - A *systematic* variation in one direction signals something significant (i.e., interesting).
       - **Try** to identify the offending parameters in the model.
         - Identify direction of variation, give **educated guess** about cause, and suggest a test.
       - If you cannot explain a significant trend in your data, at least *note* it.
     - Math is just a convenient way to apply **logic** to physical observations.
       - Many make the mistake of underestimating the power of a good model.
       - Models *follow* from *simple* statements about the plant.
       - If measurements don't match, find the logic to explain why.
         - Is something significant not modeled? If so, why? What's wrong with the logic?
         - Can parameter variation explain things? Do observed parameter variations make sense?
         - Can the model or experiment be modified to fix things?

5. Lag Compensation: Stability and Steady-State Error
    - Recall: Stability margins and feedback
        - For stability, gain must be < 1 when phase shift (i.e., delay) $= -180°$ (i.e., when "negative").
        - The *gain margin* is how much extra gain can be added before instability.
        - The *phase margin* is how much extra "delay" (phase) that can be added before instability.
            - Small phase margin corresponds to "ringing" response (i.e., slow and gentle damping).
    - Recall: Steady-state error to step input is $1/(1 + G_0)$ where G_0 is open-loop "DC gain."
        - Systems of type ≥ 1 have an integrator for infinite DC gain (i.e., no steady-state error).
            - Strictly zero steady-state error can be overkill for some applications.
            - Integrators slow response of system – add 90° of delay *and* attenuate high frequencies.
            - Root locus example: Adding integrator to first-order system yields slow and settling time that cannot be decreased with gain (i.e., gain increases sensitivity but not damping).
        - Alternatively, can use gain compensation to reduce steady-state error.
            - Gain can destabilize system by speeding up its response (amplifies high frequencies).
                - Root locus with > 2 unmatched poles eventually crosses into right-hand plane.
                - Gain margin is reduced.
            - Gain "compensation" does not shape root locus, and so it is often impossible to use.
        - Ideally, we want to place high gain *only* at DC.
    - Lag compensation gives high DC gain while allowing control over AC gain.
        - Place a pole near DC followed by a zero.
            - The (tent) pole lifts the natural DC gain of the system.
            - The zero hides the effect of the pole at high frequencies.
            - The result is something like a "low-frequency integrator."
        - As a result, for same steady-state error, compensator gain is less.
            - The compensated system has naturally high DC gain.
            - By reducing compensator gain, problematic high frequency components stay attenuated.
            - Phase margin is typically increased and overall response "rings" less.
    - Example: Operational Amplifier (feedback amplifier with high gain)
        - Gain is very high to provide low tracking error (i.e., $V_+ \approx V_-$).
        - Easy to pick up three poles (e.g., Miller effect, high frequency roll-off) or low phase margin.
        - Unity-gain feedback configuration may be unstable.
        - Add lag compensation to keep DC gain high while burying gain at problem frequencies.
        - Lag compensation implemented by hanging small series RC network off of output (as a load).

$$\frac{R + \frac{1}{sC}}{r_{\text{out}} + R + \frac{1}{sC}} = \frac{sRC + 1}{s(r_{\text{out}} + R)C + 1} = \frac{R}{r_{\text{out}} + R} \frac{s + \frac{1}{RC}}{s + \frac{1}{(r_{\text{out}}+R)C}} = \begin{cases} 1, & \text{@ DC} \\ \frac{R}{r_{\text{out}}+R}, & \text{@ AC} \end{cases}$$

Amplifier output resistance r_{out} is very small (by definition). R is picked to also be small.

- No effect at DC (unity gain), and so steady-state error is still low.
- Acts like r_{out}–R voltage divider for sufficiently high frequencies.
- Stabilizes feedback system by attenuating components that change too quickly.
- Because high frequencies are attenuated, overall response is slowed (tradeoff).
- Phase margin is increased.

- Operational amplifiers can be compensated internally as well.
    - Internal compensation introduces a low-frequency pole that dominates others.
        - Compensated operational amplifier is approximated as an *integrator*.
        - Stability comes at the cost of *speed* (a gain–bandwidth tradeoff).
    - Uncompensated operational amplifiers give designer gain–bandwidth flexibility.
        - Sub-unity-gain feedback (i.e., closed-loop gain > 1) often stable without compensation.

6. Complete the *Lag Compensation for Speed 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.
            - After placing it on your model, double-click on it.
            - Change the |++ to |+- to make one of the inputs negative.
            - These characters determine the number, type, and position of the inputs.
                - The | is a placeholder for no input. It changes the *spacing* of the inputs.
                - The + means the input is positive, and the - means the input is negative.
        - Use Zero-Pole or Transfer Function from Continuous section of *Simulink* library.
            - All poles and zeros are *negative*!
        - *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 125° phase margin).
        - ⋆ Make sure your design does *NOT* use α in deciBels (dB)! (correct $\alpha \approx 5.95$)
        - Equivalent Bode design for the slow model is formed by *dividing both* pole and zero by 1.5.
        - Gather step response data in *dSPACE ControlDesk* for **both** (i.e., fast and slow) designs.
    - Change to your root-locus-type controller design from the pre-lab (improves steady-state error).
        - ⋆ **IMPORTANT FOR SAFETY:** Divide K_c gain by 4! We're not using a *unit* step input!
        - Gather step response data in *dSPACE ControlDesk*.
    - ⋆ AT ANY TIME, IF MOTOR STARTS CLICKING BACK AND FORTH VERY QUICKLY, STOP THE EXPERIMENT – DISCONNECT THE MOTOR IF NECESSARY!! The high-frequency switching can cause **permanent damage**! Switching has several possible causes.
        - *Saturation* of large control signals adds nonlinearities that can destabilize the system.
            - Recall "describing functions" from ECE 752. Nonlinearities can lead to fast "limit cycles."
        - Unstable or underdamped closed-loop poles with large imaginary parts (i.e., high frequency and no significant damping) can lead to fast (and possibly growing) oscillations.
            - Large controller gains are often the cause of this problem.
        - Your closed-loop system can *chatter* when your error or control signals get buried in digital quantization noise. In other words, digital quantization can introduce a switching nonlinearity that prevents smooth movement.
            - This nonlinear effect is similar to a motor stalling when it does not overcome its starting friction. Quantization levels *can* act like friction thresholds.
    - ⋆ There is no manual tuning in this experiment.
    - ⋆ For your report, remember that the input is a step of amplitude 4 (not 1 as in the pre-lab).

- 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.