ME552 Lab 3: DC Motor Servo System (Fall 2016)
Lab Assignment 3 (100 points total)
Hardware Demonstration Due: October 14th, Friday, 9:30am during lab
Written Report Due: October 21st, Friday, 9:30am during lab

For the hardware demonstration, please demonstrate to the instructor/GSI the ability to spin the motor shaft in response to a command from LabView and read shaft angular position output from the encoder using LabView, as shown in the lab instructions.

Feel free to use bullets, paragraphs, figures, references, data-sheets, etc., to make your answers clear while responding to the following questions. More details are better rather than short/ incomplete responses. For any derivations, show your work in step by step detail. Reports may be typed or hand-written (should be clear and legible in the latter case).

Note that the following questions are intentionally meant to be open-ended to expose you to a more realistic work/research environment. Use your best judgment in answering them. It is important to provide a "complete" plan / structure of your solution approach, and then fill in details at each step, as needed. However, if there are certain questions that turn out be a substantial time sink without much learning-value in your opinion, please let Prof. Awtar know ASAP.

1. Study the AMC servo-amplifier Datasheet and Part 3 of the Lab 3 Instructions (both on Canvas). The servo-amps should already be configured and ready to use in the 'Current Mode', with a gain of 1A/V. On the datasheet, study the circuit block diagram on the first page.

    a. By now you must have had enough experience with op-amp circuits to identify and recognize the functionality of the various op-amp modules that you see on the circuit. Identify and describe qualitatively as well as quantitatively the function of the op-amps U1, U3 and U4 (6).

    b. What is the role of the resistor (10K) that you see next to the "Current Sense" block (1)? How would you justify its large resistance value (1)?

    c. Refer to the motor data-sheet (on Canvas) and determine the maximum continuous current that it can handle in your opinion. Do not simply copy and paste a number/value from the data-sheet. Instead, provide a physical and mathematical justification for the maximum continuous current value that you are proposing. How does this compare with the maximum rated current listed on the Motor Datasheet (4)? Attempt a similar "educated" estimate of maximum intermittent (or non-continuous, or peak) current that the motor can handle. Explain the data-sheet's AND/OR your interpretation of maximum intermittent current (3).

    d. Based on the Servo-Amp Datasheet and Part 3 of Lab3 Instructions, what are the maximum continuous and intermittent (or peak) current values that the servo-amp can output to the motor? Instead of simply citing numbers, please provide a clear and detailed explanation. Explain the Servo-Amp data-sheet's interpretation of intermittent (or peak) current (4).

    e. Based on your responses to parts c. and d. above, what command saturation limit would you include in your LabView controller? The purpose of such a software based saturation limit would be to ensure that the LabView controller never outputs a command signal that is high enough to damage either the servo-amp or the motor. Please discuss and rationalize the saturation limit that you used in your LabView controller (2). In general, how would you ensure a safe operation (from a current and power stand-point) of all the hardware involved in this experiment? In other words, what is your "Safety Strategy" (2)?

In addition to the Datasheet and Lab Instructions, the Canvas folder 'Servo Amplifier Details' contains several engineering references that you should browse through. This will help hone your hardware operation and debugging skills in this and subsequent labs/projects.

2. Study the motor model covered in the class. Make sure you understand the underlying assumptions and simplifications. You will use this model in designing a servo-system in this lab. Study the provided motor datasheet and make sure you can find values of all motor parameters that you need for the model.

   a. Present your motor model along with a list of parameter values that are needed in your model. Identify the values of these parameters from the provided motor data-sheet (3). Clearly state the assumptions that you used creating your model (1).

   b. In a hypothetical scenario where you did not have the motor data-sheet readily available but the actual physical motor was given to you, suggest measurements or experiments that you would conduct to identify the parameter values that you need for your motor model (9).

3. Create a model for your entire system (DC motor, optical encoder, and servo-amplifier) (4). Generate transfer functions between V_{ref} (Input) and Motor Angle (Output), and V_{ref} (Input) and Motor Velocity (Output). Present your model in a clearly labeled block diagram. Also, create your model in Simulink. Make sure to use a consistent set of units (SI) in all your modeling. Create a MATLAB file that lists all the parameters used in your model. Include all of this in your assignment submission (4).

   Based on your overall system model, design two controllers in MATLAB/Simulink – one for position control and one for velocity control. These controllers are to be implemented separately in LabView at a later stage (Problems 4, 5, and 6). For each controller design, you may use a traditional PID or lead/lag controller type. Alternatively, you may choose to design and implement other controllers to achieve better closed-loop (or servo-system) performance.

   The objective for the position control system is to

   a. Keep steady state error below 2% for a step command of pi radians (3).
   b. Provide command tracking: Amplitude should remain within 5% of a commanded amplitude of pi radians, up until 5 Hz (3).
   c. Attenuate a 60Hz noise signal coming in either at the servo-amplifier or the sensor by at least 10 times (3).

   The objective for the velocity control system is to

   d. Keep steady state error below 2% for a step command of 2*pi radians/sec (3).
   e. Provide command tracking: Amplitude should remain within 5% of command up until 5 Hz for a "command amplitude" of pi/2 radians/sec (3).
   f. Attenuate a 60Hz noise signal coming in either at the servo-amplifier or the sensor by at least 5 times (3).

   Show your step by step procedure for designing the above controllers. Show the performance and effectiveness of your controllers in simulation using graphs, calculation, etc. (whatever you find appropriate). For each control task, it is generally difficult to achieve all three goals simultaneously. It is OK to present whatever results you are able to achieve. In particular, please provide a detailed discussion of your findings, observations, and challenges if you are not able to achieve the three goals simultaneously, for either or both of the above controllers (4). (Note that, so far, you should be testing the controllers only in simulation and not in hardware.)

Before you proceed to implement your controllers on the provided hardware, make sure you set up and debug all components individually. This is particularly important when you work on a complex project. If you have made sure individual components work as desired, it is much easier to debug the overall system once you make all the connections. Most importantly,

- Use a consistent set of units.
- Be consistent in signs and directions. This is the single most important and common mistake made in servo-system design. If directions and signs are not accounted for, you may end up implementing positive feedback instead of negative feedback. Therefore, make sure that a positive command to the servo-amp rotates the motor in a 'CW' direction, and a 'CW' rotation of the encoder disk produces a 'positive' output that is measured within your controller. Of course, you could have chosen the 'CCW' direction, instead, for all of this. If signs are not consistent the way you desire, feel free to swap the leads of your motor where it connects to the servo-amp, or simply add a '–' sign to the command signal in your LabView control VI. HOWEVER, DO NOT SWAP THE LEADS ON YOUR POWER SUPPLY.
- Have each member of the team individually check all connections and the above consistency before turning on power to the overall system.

Now you are ready to implement your controllers. PLEASE NOTE that you are not required to experimentally demonstrate the noise attenuation effect of your controller in your physical system because that involves a complicated procedure to measure the noise from the power source and the servo amplifier.

4. First implement the **Position Control** using the encoder as the feedback sensor.

   Generate any signals that you need from this single measurement, e.g. generate integral (if you are using an Integral in your PID) using the '1/s' operator. Similarly, generate velocity from the position simply by using the derivative operator 's', followed by a filter $1/(\tau s+1)$. You will have to choose an appropriate value of τ over the course of your experiment. Note any difficulties that you see in implementing your controller. For better velocity estimation techniques using encoder position output, you may consider implementing one of the schemes suggested in the references posted under "Optical Encoders Velocity Estimation" in Canvas.

   a. Report your closed-loop experimental results, graphically and numerically, and compare them with your simulation results (from Problem 3) (2). What is the maximum closed-loop bandwidth that you can achieve (2)? Try this out experimentally simply by increasing the command frequency. Do you face any challenges in experimentally meeting the performance specifications stated in Problem 3a and 3b? Provide a detailed discussion (4).

   Note that you are not required to experimentally demonstrate the noise attenuation effect of your position controller in your physical system (that is, you are not required to check if your controller is meeting the performance specification 3c), because that will involve a complicated procedure to measure the noise from the power source and the servo amplifier.

5. Continuing with **Position Control.** Your system has various non-linear parasitic effects that were not included in the motor model covered in the lecture. However, these effects exist in most real-life applications.

    a. Coulomb Friction. What are the sources of Coulomb Friction in your system (1)? Suggest an experiment to measure friction in your system (2). Either run this experiment to obtain the friction value or use a value based on the motor data-sheet. Incorporate friction in your Simulink model to make your simulation based predictions better (2).

    b. Saturation. All real power supplies saturate; motors and servo-amps have current limits, and controllers have saturation blocks built in to prevent damage to the system. Include the effect of saturation in your Simulink model. Do your new simulation predictions better agree with what you see experimentally? Submit plots/screenshots (1). In both your simulation as well as experiment, keep increasing the amplitude of your step command (pi, 1.5pi, 2pi and so on). Report your experimental observations (2). Do you expect the outputs to look the same? Per your simulation and experiment, at what value of the commanded step input does the output start to appear different/distorted? Justify (4).

    c. Quantization. Position output from the encoder has been quantized, which apart from making velocity estimation difficult, also places a resolution limit on the position measurement. Model this effect in your Simulink model (2). Keep making the quantization steps larger and larger starting from your current value of N steps / rev (based on the encoder that you are using). At what point does the resolution start affecting your simulated closed-loop performance? Submit plots/screenshots (4).

Based on these steps, comment on how the above parasitic effects influence the closed-loop response? Does including them in your simulation bring your simulated and experimental results closer to each other? Why / Why not?

6. Now implement the **Velocity Controller you designed in problem 3** using feedback from the encoder.

Generate any signals that you need for your PID (or lead/lag) velocity controller from this single measurement from the encoder. Do you face any challenges in meeting the performance specifications? If you are not able to experimentally reach the specifications in problem 3d and 3e, simultaneously, then design and implement controllers to separately reach the goals, one at a time. Report your closed-loop experimental results, graphically and numerically, and compare them with your simulation results (from Problem 3). If you think it is relevant, you should include the parasitic effects listed in Problem 5 in your Simulation. Comment on how well your experimental results compare with the simulated results. Did you face any challenges in experimentally meeting the performance specifications stated in Problem 3d and 3e? Provide a detailed discussion (6).

What is the maximum closed-loop bandwidth that you can achieve (2)? Try this out experimentally simply by increasing the input command frequency.

Note that you are not required to experimentally demonstrate the noise attenuation effect of your velocity controller in your physical system, because that will involve a complicated procedure to measure the noise from the power source and the servo amplifier.