ME552 Fall 19 Lab 3: DC Motor

Lab Assignment

Prof. Shorya Awtar

Mechanical Engineering Department, U of M

Lab 3 Guidelines:

I. Hardware demonstration deadline: October 18th, Friday, Mechatronics Lab Section

II. For the hardware demonstration, follow the demo requirement instructions.

   a. You will also be evaluated for the quality of your bread-board circuit layout and wiring. Please do a clean job so that your circuit and wiring looks uncluttered and well organized.

   b. You will also be evaluated for the quality of your LabVIEW code. Again, both the front panel and the block diagram should be uncluttered and well-organized.

III. Lab assignment submission deadline: October 18th, Friday, 9:00am on Canvas. The submission should be on Canvas only. You may want to keep a copy for your own records.

IV. There is ONLY ONE submission required per team. However, there are strict requirements placed on a fair contribution from each team member. Each person should contribute equally to the experiments and the written assignment. In particular, for the assignment each person of the team must be intellectually involved in each problem. You should not divide up the various questions among various members of team. Each team should discuss and work together on each component of the submission. At the time of assignment submission, each student will be required to turn in a confidential PEER EVALUATION under Quizzes section on Canvas. PEER EVALUATION for this assignment is due by 9:00am October 18th. Failure to submit peer evaluation will cause penalties to your individual score of lab 3.

V. Any questions/clarifications related this assignment should be posted on Canvas → Discussions → Lab 3: DC Motor.

   It is recommended to export the measured data into MATLAB and carry out the plotting using MATLAB (LabVIEW is powerful in data acquisition; MATLAB is powerful in data post-processing).

Questions (4 questions, Total 143 pts):

1. **Overall System Modeling (38 pts)**

   a. **DC Motor Model:** Develop a physical system model and associated mathematical model that includes the mechanical and electrical aspects of the DC motor. Consider the fact that the motor has a rotor, bearings, commutator, etc. that involve inertia and/or damping.

   For the mechanical aspect of the DC motor, draw a Free Body Diagram (FBD) of the motor rotor, consider ALL RELEVANT torques on this rotor, and derive the Equation of Motion (EOM) for the motor rotor that relates the motor driving torque τ_m and rotor position θ. Make sure you define positive and negative directions and stick to these definitions. Make sure you list all the assumptions you have made when modeling mechanical aspects of DC motor. **(3)**

   For the electrical aspect of the DC motor, drawing an electric circuit diagram that includes a voltage input V_{in} to the motor. Derive the electrical equation that relates the input voltage V_{in} to motor driving torque τ_m and rotor speed $\dot{\theta} = \Omega$. Make sure you define positive and negative polarities for voltage and current and stick to these definitions. Make sure you list all the assumptions you have made when modeling mechanical aspects of DC motor. **(3)**

   Use the above two models (mechanical and electrical) to derive the mathematical relations between all the relevant inputs (including the controlled input V_{in} as well as any disturbance and noise sources) and outputs (rotor position θ and rotor speed Ω) of the DC motor system. **(1)**

   b. **Parameter Identification.** Provide a list of all physical parameters that are needed in your DC motor model. Identify the values of these parameters from the provided motor data-sheet. **(2)** Now suppose that motor datasheet was not available, but the actual DC motor was given to you, suggest simple measurements or experiments that you would conduct to identify the parameter values that you need for your DC motor model. **(3)** Run these measurements / experiments and compare the parameter values with those provided in the datasheet **(12)**. Is there any significant difference between your measurements and the datasheet? If so, please explain possible reasons. **(3)**

   c. **Actuator Driver Modeling.** Review the actuator driver data-sheet provided to you. If needed, create a model, show input (V_{ref} from the controller) and outputs (voltage input V_{in} to the motor and current i_m in the motor), and list any necessary assumptions. **(2)**

   d. **Sensor Modeling.** Next, do we need a model for sensor (encoder)? If needed, describe/create a model, show inputs and outputs of this model, and list any necessary assumptions. **(2)**

   e. **Position Control Block Diagram.** Suppose we have a controller $C_1(s)$ for rotor position control. Create a block diagram that captures your overall physical and control system. Include all necessary sub-components, including controller, driver, actuator, sensor, etc. Label all connection lines in your block diagram. Label all sub-system models. Show transfer functions in each block if possible. Label all power supplies. Use a dashed line box to highlight the DC motor in your block diagram. This overall physical and feedback control system comprising a DC motor is referred to as servo-system. **(3)**

   f. **Velocity Control Block Diagram.** Suppose we have a controller $C_2(s)$ for rotor velocity control. Create a block diagram that captures your overall physical and control system. Include all necessary sub-components, including controller, driver, actuator, sensor, etc. Label all connection lines in your block diagram. Label all sub-system models. Show transfer functions in each block, if possible. Label all power supplies. Use a dashed line box to highlight the DC motor in your block diagram. This overall physical and feedback control system comprising a DC motor is referred to as servo-system. **(3)**

g. Simulink Model. Create a Simulink model for the open loop system (from V_{ref} to θ and to Ω). Create a MATLAB file that lists all the motor parameters as variables, and use these variables when creating Simulink model. Make sure to use a consistent set of units (SI) in all your modeling and list these units in your MATLAB file. In addition to the above aspects that you have modeled (Question 1(a)-(c)), make sure you capture any other parasitic effects that might be relevant such as friction, saturation, noise and disturbance sources, etc. in your Simulink model. Provide an image of your Simulink model in your assignment submission **(5)**.

2. Position Control using Driver Current Mode of the Driver (36 pts)

In this question, you are asked to design and implement a position controller $C_1(s)$ for the servo system.

a. Controller Design in MATLAB/Simulink. Based on your overall system model developed in Question 1, design a controller for motor position control. You may design either a PID controller or a lead-lag controller. Alternatively, you may choose other controllers as long as you are able to achieve good closed loop performance.

Since you are designing a controller in the continuous time domain using classical controls tools, make any linearization assumptions that are necessary based on the models developed in Question 1, and derive the open loop transfer function $P_1(s)$ of the physical system between actuator driver input V_{ref} (Input) and rotor angle θ (Output). Use this linearize physical system model to design your controller. The designed controller must satisfy following position control specifications for the servo-system:

i. Keep steady state error below 2% for a step command of 1 radian **(3)**.
ii. Keep overshoot less than 30% of the step command value of 1 radian **(3)**.
iii. Provide command tracking: Amplitude should remain within 5% of commanded amplitude of 1 radian, up until 5 Hz **(3)**.
iv. Attenuate 1 kHz noise by at least 10 times **(3)**.

Show your step by step procedure for designing the above controller (e.g. root-locus, frequency domain, SISO tool, bode plots, etc.). Show the performance and effectiveness of your controller in MATLAB/Simulink. Make sure to include a saturation block at the output of your controller in your Simulink block diagram. This is a practical safety mechanism that you will use in your actual controller to avoid damage to the physical system. Please provide a rationale for the saturation limits that you should choose for your controller output **(1)**.

If you find it difficult to achieve all four objectives above, it is OK to present whatever results you are able to achieve. Provide a detailed discussion of your findings, observations, and challenges if you are not able to achieve the four goals simultaneously, for the above controller **(8)**.

Note that, so far, you should be designing and testing the controller only in simulation and not in hardware. Before you proceed to implement your controller 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.
- Have each member of the team individually check all connections and the above consistency before turning on power to the overall system.

b. Controller Implementation and Evaluation in LabVIEW. Implement your designed position controller in LabVIEW. Modify control gains if necessary to achieve the above listed control specifications and overcome limitations in the real hardware that you may not have modelled adequately. Demonstrate that your controller meets the specification stated in Question 2(a) (i.) through (iii.) above. YOU DO NOT have to experimentally demonstrate the noise attenuation listed in 2(a) (iv.). Report your closed-loop experimental results, graphically and numerically, and compare them with your simulation results (from Question 2(a)). If you encounter any challenges / difficulties in meeting these requirements, provide a detailed discussion on possible reasons. What is the maximum closed-loop bandwidth that you can achieve based on your simulation and based on your experiment **(15)**?

3. <u>Velocity Control using Driver Current Mode of the Driver (35pts)</u>

In this question, you are asked to design and implement a position controller $C_2(s)$ for the servo system.

a. Controller Design in MATLAB/Simulink. Based on your overall system model developed in Question 1, design a controller in MATLAB/Simulink for motor velocity control. You may design either a PID controller or a lead-lag controller. Alternatively, you may choose other controllers as long as you are able to achieve good closed loop performance.

Since you are designing a controller in the continuous time domain using classical controls tools, make any linearization assumptions that are necessary based on the models developed in Question 1, and derive the open loop transfer function $P_2(s)$ of the physical system between actuator driver input V_{ref} (Input) and rotor speed Ω (Output). Use this linearize physical system model to design your controller. The designed controller must satisfy following velocity control specifications for the servo-system:

i. Keep steady state error below 2% for a step command of 2*pi radians/sec **(3)**.
ii. Keep overshoot less than 30% of the step command value of 2*pi radians/sec **(3)**.
iii. Provide command tracking: Amplitude should remain within 5% of command up until 5 Hz for "command amplitude" of pi/2 radians/sec **(3)**.

iv. Attenuate 1 kHz noise by at least 10 times **(3)**.

Show your step by step procedure for designing the above controller (e.g. root-locus, frequency domain, SISO tool, bode plots, etc.). Show the performance and effectiveness of your controller in MATLAB/Simulink. Make sure to include a saturation block at the output of your controller in your Simulink block diagram. This is a practical safety mechanism that you will use in your actual controller to avoid damage to the physical system.

If you find it difficult to achieve all four objectives above, it is OK to present whatever results you are able to achieve. Provide a detailed discussion of your findings, observations, and challenges if you are not able to achieve the four goals simultaneously, for the above controller **(8)**.

b. Controller Implementation and Evaluation in LabVIEW. Implement your designed velocity controller in LabVIEW. Modify control gains if necessary to achieve the above listed control specifications and overcome limitations in the real hardware that you may not have modelled adequately. Demonstrate that your controller meets the specification stated in Question 3(a) (i.) through (iii.) above. YOU DO NOT have to experimentally demonstrate the noise attenuation listed in 3(a)(iv.). Report your closed-loop experimental results, graphically and numerically, and compare them with your simulation results (from Question 3(a)). If you encounter any challenges / difficulties in meeting these requirements, provide a detailed discussion on possible reasons. What is the maximum closed-loop bandwidth that you can achieve based on your simulation and based on your experiment **(15)**?

Note that one intrinsic difference when implementing velocity controller is that velocity (which what we want to control) is not measured directly in this lab. In other words, you never know the exact motor velocity at any time (unlike the motor position, which is known from the optical encoder measurement). You have to reasonably estimate the rotor velocity from the rotor position measurement.

4. Controller Implementation Practical Considerations. <u>**(34 pts)**</u>

Consider the following practical factors that are relevant to controller design versus controller implementation for both position control and velocity control. The following questions are pertinent to both Position Control and Velocity Control. You can provide your answers with support from your modeling and experimental results from either of these two controllers.

i. <u>*Continuous vs. Discrete Control*</u>. When designing the controllers above via SISO Tool, frequency domain methods, time domain methods, etc., did you assume continuous time domain / discrete time domain **(1)**? How about when you simulated the system response in Simulink **(1)**? Is LabVIEW Control & Simulation Loop environment continuous time domain / discrete time domain **(1)**? If the control implementation is different from the controls model, then what is the impact of this difference on servo-system performance **(1)**?

In a discrete control system, all inputs and outputs are updated at a certain frequency (loop rate), which results in an inherent delay in the overall system. To better understand the impact of time discretization and the fact that a control system works a certain finite loop rate (e.g. 1kHz in your case) rather than instantaneously, capture the impact of time discretization in your Simulink model (from Question 2(a) or Question 3(a)). Within Simulink, gradually increase the controller cycle time from the value you used in your lab (e.g. 0.001s) until system behavior changes significantly. Include a plot that shows how system performance changes through this process. Provide a discussion **(6)**.

Now go back to the experiment and change the loop-rate in your Simulation loop properties from 1kHz to lower values and observe and report how the servo-system performance changes. **(4)**

ii. *Numerical Derivative*. If your controller involves a 'D' term (when estimating velocity or when implementing a controller), your LabVIEW implementation involves a numerical derivative. Numerical derivative can introduce spikes when signal is quantized (e.g. steps in encoder) and the control implementation is digital (e.g. finite loop rate). What is the dominant frequency content of these spikes? (**2**) One way to smooth out the spikes is to add a first-order filter $\frac{1}{\tau s+1}$ (or other more sophisticated filters) to the signal passing through the 'D' operation. How do you select the time constant τ in this filter? What is the effect when τ is too large (e.g. 1s) **(2)**? How about when τ is too small (e.g. 0.0001s) **(2)**? What is impact of these spikes resulting from optical encoder quantization on the overall controller performance? (**2**)

You can model and study this quantization effect and its resulting velocity estimation challenges in your Simulink model. Start with the quantization based on the encoder resolution (from datasheet) and keep making the quantization steps larger and larger. At what point does the encoder resolution start affecting your simulated closed-loop performance? Submit plots/screenshots along with discussion **(4)**.

iii. *Saturation*. You have implemented a controller output saturation in your LabView program to protect your hardware. There might be other sources of saturation that you might have identified in your physical system. 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. What is the effect of such saturation limits on your controller performance **(1)**? If your controller performance is not satisfactory, would you consider modifying this saturation limit? Why or why not **(3)**?

In your Simulink simulation, preserve the saturation level that you set at your controller output. Now, increasing the amplitude of your step command (1rad, 1pi, 2pi, 4pi, and so on). Report your observations with a plot that shows how the time response behavior changes with increasing magnitude of the step command **(2)**. Per your simulation, at what value of the commanded step input does the output start to saturate? **(1)**.

Next, try this out in your experiment. Keep increasing the step command and look for change in the time domain response. Report your findings. Provide a discussion. **(2)**

iv. *Coulomb Friction*. What are the sources of Coulomb Friction in your system? Name at least 2 sources **(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 in your Simulink model. Incorporate a friction model in your Simulink model to make your simulation based predictions better. Show/explain how you modeled friction **(4)**. What is the impact of friction in your control system design and how do you mitigate its negative impact? **(1).** In your Simulink model, gradually increase Coulomb friction (x2, x4, x8, x16 etc. from its nominal value) until the system behavior changes significantly. Report and discuss your observations. **(2)** What will you do different from a control system design stand-point if friction is high and what is resulting trade-off if any? **(2)**