Lab 2. Motor system ID, control, and trajectory following.

Due date: **Each group must submit a *short but well-written* (3-4 pages) Lab Report by 5pm Mon., April 23 (in dropbox).**

Overview

The goal of this lab is to develop a controller to move the two-link arm in some desired trajectory. Along the way, you will need to indentify a model for the Lego motor, and you will develop several basic “tools” for use (and reuse and modification in future labs) with the Lego NXT system, implemented via a combination of Simulink blocks and MATLAB m-files.

However, as always, given a general goal, you are encouraged to explore various solutions, depending on the knowledge you bring with you to the class and your interests and curiosity.

Credit for the lab includes prelab, in-class check-off by TA, and a group lab report to explain your approach and to verify you have completed the tasks listed below. As there is no prelab for Lab 3, we strongly encourage you to complete your Lab 2 report before Lab 3, if possible! **We are creative/flexible about how you approach particular problems.** For full credit, you must include appropriate documentation of your process in obtaining each result (e.g., a diagram, sketch, or explanation of how you modified your simulink model, any needed paper calculations, a print-out of resulting data, and/or data needed to perform calculations, etc.). *Please be concise and efficient in your presentation; well-labeled plots are always appreciated – and often can be much more effective in communicating details than excessive text.*

Your specific mission task (should you choose to accept it) in this lab are:

---

1. **Characterize (via system id) the plant.** The goal is to get a model that is "good enough" that you can design a controller. You may or may not need to identify the same set of parameters suggested in the prelab.

2. Modify your simulink model to log an estimate of angular velocity, in addition to angular position. There are several possible solutions here, including a DT filter or even designing an observer (if you already know how). You will get partial credit if you can "post-process" data to estimate velocity, and this is a very good way to start, regardless.

3. Design and implement a simple controller. ***Be gentle with the lego system: first test things on a motor that is NOT connected to an arm!*** It may be useful to begin with a proportional controller, but later see if you can implement PD control. You might use your derivative estimate (from part 2) directly, or you might first design the PD controller in continuous time and then include an LTI system block with a discrete-time version of this controller. You may need to include a "high-frequency" pole, to smooth the quantization effects of the encoder.

<u>Finally, use your SISO controller to move the 2 links of the robot arm:</u>

4. Plan a simple trajectory for the end effector and modify your simulink model so that the 2-link arm tracks this trajectory. Provide data showing tracking performance: How much error is there? How fast is the dynamic response of the controlled system? Be sure to highlight any particular issues with your controller that you may notice (as potential problems).

Bonus ideas, as time permits: (extra credit)

5. Modify your controller is some "interesting" way. For example you might:

- Try to compensate for friction.
- Try to achieve a stable but very "soft" 2nd order response. Can you "feel a difference" at the output, or is the system always entirely non-backdriveable during active control?
- Try to compensate for gravity, to minimize any time varying errors in trajectory following.
- Experiment with different trajectory commands. Imagine your goal is to get to some end state with high repeatability and accuracy – for example, to tap a ball (think "pool cue") to roll it in a particular direction, or to throw a rubber ball. Can you minimize the effects of backlash, etc?