Lab 1. Data Logging and Visualization.

The primary purpose of this lab is to help you learn about how to build and download C code for the Lego NXT (Mindstorms) controller, via Simulink models in MATLAB.

However, it is always much more fun to have a goal in mind, to learn the "details" only as a means of accomplishing some particular task(s).

And so for this lab, your mission (should you choose to accept it) is to accomplish the following two tasks:

---

1. **Modify code that you are given so that you display (within a figure in MATLAB) an animation of the two-link Lego arm as you move it through space.**

Side notes: As part of the lab check-off, you may be asked how your code works, what steps you used to make your visualization of the data more accurate (if any), and also what you notice about performance: e.g., Does the animation "look right" to you? How did you set the "initial conditions" for each trial, and how accurate was your initialization method, would you guess? How bad is the time delay between the true motion and your animation, roughly? (Creativity is encouraged in making such estimates… play with the system!)

---

2. Assume the "end effector" is the end of the distal (more distant from the origin) arm link. **Modify code that you are given so that you display (within a figure in MATLAB) a 2D plot of the locus of points the end effector has traversed over time.**

Side notes: In other word, imagine you have a "virtual pen" at the tip of the robot arm, and you would like to use MATLAB to plot the trajectory of the pen over time. How accurate is your virtual plotting tool? For example, if you trace the flat surface so that the end effector must travel in a straight line, is the output you plot in MATLAB also a straight line (as desired)? Do you see the same data plotted when you move left-to-right versus right-to-left (between the same two points, A and B)? Can you improve your calculation of the kinematics based on errors you observe in your output plots? What do think the dominant sources of error (between actual and plotted kinematics) may be?

Bonus round: Now, let's say you wish to "pick up" the pen every so often, so you can plot a series of disconnected lines for a more interesting drawing. Implement this using the elements you have. You might use the button switch to toggle the pen state, for example, or holding the button might keep the pen "raised" while releasing it lowers the pen. Or perhaps you can use the 3rd motor encoder (not part of the arm) in a clever way. Note that it may be easier to draw pen locations as points, '.', rather than as individual lines, '-' in MATLAB, unless you have a clever method to keep track of individual link segments. Clever solutions are encouraged, though!

Take away message: Visualizing data can be quite useful for debugging! We encourage you to animate or otherwise display data you log in a meaningful way in future labs.