Lab 3. Robot ping-pong ball toss competition.

Deliverables: **Each group must submit a *short* (~3-page) Lab Report, due at the start of your Lab 4 session.** For fun, we will run a contest, to be defined more precisely by the TA's in lab, to compare accuracy, repeatability, and distance of your robot's toss.

Overview

The goal of this lab is to control the two-link arm to launch a ping-pong ball into a receptacle.

As always, you are encouraged to explore various solutions, depending on your personal knowledge, interests, curiosity, and persistence.

As in Lab 2, you report must explain your approach and must verify you have completed the tasks listed below. **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. We don't need/want a lab introduction, for example. Also, well-labeled plots are always appreciated – and often can be much more effective in communicating details than excessive text.*

Your tasks in this lab include:

1. **Design a trajectory for the end effector.** On the lab website, we have included an m-file function called `trajectory_draw.m` that may be helpful as a starting point. The function shows the approximate workspace for the end effector and allows you to click points, which are then connected via a spline. (x,y) points are returned, as well as joint angles (a1, a2). The website also includes an "appendix" that gives data for a human (TA Brian Satzinger) tossing a ball; we suggest a similar trajectory may be a good starting point in your design process.

2. Calculate inverse kinematics, to obtain trajectories for the joint angles. There are two m-files you may download from the lab website to aid in calculating kinematics and IK. You may need to modify these files to increase the accuracy of calculations, as joint lengths and angle offsets are currently only approximate guesses. These files are: `ang2xy.m, xy2ang.m`

3. Set controller as desired. Your controller design from Lab 2 may be good enough. Otherwise, improve your controller and state estimation (filtering for velocity?) as desired.

4. Modify a simulink model so that the arm follows the desired trajectory and tosses the ball as desired. This step will require iteration and experimentation! Good luck.