

Prelab 1

Prelab 1 is due on Thursday, Oct. 4, in the dropbox outside lab (3120A Harold Frank Hall).

**** Student enrollment includes a wide range of backgrounds. If you feel uncomfortable with MATLAB, please come to Prof. Byl's office hours Monday, Oct. 1, which I will hold in our lab in 3120A Harold Frank Hall, for a special, brief MATLAB tutorial. ****

Deliverables for the Prelab 1 assignment are as follows:

- Print-outs of your MATLAB m-files.
- Print-out of the final frame of your animation, showing the final position of the 2-link arm and the full end effector path. This will look similar to Fig. 1, below, except that you should see a full word written out in script, and the arm will be in a different location.
- Answers (either by hand calculations or via MATLAB code) to the questions.
- Also, you **must** bring an electronic copy of your m-file to lab. (Alternatively, email it to yourself so you can retrieve it at the start of lab, via internet.)

Lab 1 involves recording data for and implementing simple control of the joint angles of a simple 2-link robot arm. After first moving the arms by hand and (later) running the controlled arm, you will be asked to animate the arm and to plot the path traversed by the end effector. This lab is largely intended to get you familiar with our LEGO NXT setup and MATLAB interface. Lab groups may (unfortunately) be large, so it is important that you spend some time preparing before your lab meets! That said, this should (hopefully) be a relatively short assignment.

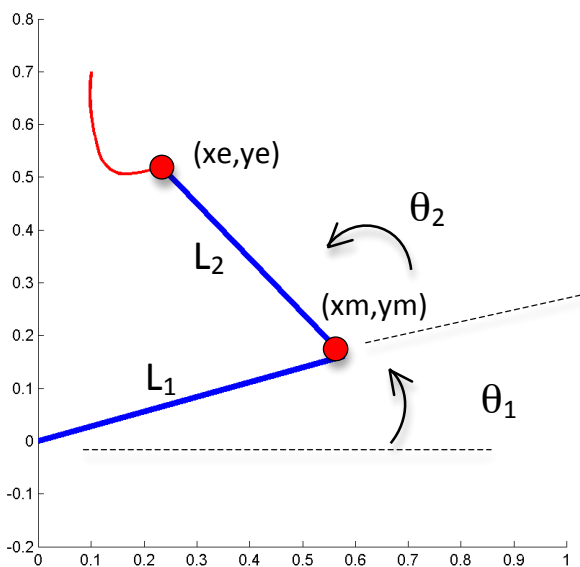


Figure 1: 2-link arm.

At left is an example frame from an animation showing the 2-link arm (in thick, blue) and the path so far for the end effector (shown as a thinner, red line).

For the purposes of the prelab, use:

$$L1 = 0.6$$

$$L2 = 0.5$$

Use the same axes depicted, too:

```
axis equal
axis([0 1 -0.2 0.8])
```

For all parts below, define the lengths of the arm segments, $L1$ and $L2$, near the top of your functions. In lab, you will need to measure the actual robot to get the correct values for the true LEGO robot. For this prelab, simply use $L1=0.6$, $L2=0.5$.

1. Write two, short MATLAB functions, each described below.
 - a) A function called “ang2xy.m”, that takes the two theta angles, “th1” and “th2”, as inputs and outputs coordinates on the arm, “x” and “y”. $x=[0, x_m, x_e]$, and $y=[0, y_m, y_e]$. These are just coordinates for the origin, elbow (x_m, y_m) , and end effector (x_e, y_e) .
 - b) A function called “xy2ang.m”, which takes the end effector coordinates, “xe” and “ye”, as inputs and outputs the “theta1” and “theta2”. Here, assume “thetas” must always be between 0 and pi radians, to ensure your solution is unique. Hint: You can use the function “LawOfCosines.m” to help find the angles of a triangle for which all three side lengths are known. You may download this function from the lab website.
2. On the course website, go to the Laboratories tab, download the following file for Lab 1, and load it in MATLAB:

```
load prelab1_thetas.mat
```

This mat-file contains two variables, th1 and th2. These are the absolute angle of the first link (th1), measured counterclockwise from the positive x axis, and the relative angle of the second link (th2), measured counterclockwise with respect to the first link’s angle (see Fig. 1). Write a MATLAB m-file function that takes th1 and th2 as inputs and animates the motion of the 2-link arm over time. Draw both the actual arm and the path of its end effector, as shown in Fig. 1.

Hints:

- Rather than re-drawing the 2-link arm and the end effector path for each animation, create an object and redefine the Xdata and Ydata fields for each frame. e.g.,


```
p1 = plot([0 1], [0 0], 'b-', 'LineWidth', 4);
set(p1, 'Xdata', [0 .9], 'Ydata', [0 0.5])
```

 For your code, you will have two objects: one for the arm and one for its path.
 - Use thick lines for both the arm and the end effector. The ‘LineWidth’ setting in the previous example will do this.
 - Be sure the axes have a 1:1 ratio and that they are fixed over time. For the prelab, use `axis([0 1 -0.2 .8])`, as shown in Fig. 1. The command “axis image” forces the x and y axes to use the same scaling, so the aspect ratio in your drawing is not distorted.
3. Include a print-out of the final frame of your animation in your prelab. It should look similar to Figure 1, except that the full end effector path should spell a word in cursive handwriting.
 4. Consider the region $0 \leq x \leq 1$ and $-0.2 \leq y \leq 0.8$ (i.e., the 1x1 square shown in Figure 1). Sketch (by hand or with MATLAB) the reachable workspace within this unit square for the arm’s end effector. What fraction of the 1 x 1 square can the end effector reach? (MATLAB will be useful for calculating or estimating areas. Try to estimate to within 2 sig figs.)
 5. Solve for the lengths, L1 and L2, needed to allow the arm to reach any point in the 1x1 square, while minimizing the length of L2.
- **Copy your m-files to a USB key or email it to yourself, so you will be able to retrieve it at the start of lab.** (Do not submit your code electronically to us: just make sure you will have access to your m-file in lab, for editing...)

