ECE 481: Computer Project #2 Description

Due Date: December 10, 2015 by 8 pm

Project Overview:
The objective of the second computer project is for you to gain experience with the challenges involved in implementing a kinematic simulation of an articulated object. Your program must generate the joint angles required to move a planar, n-link robot so the robot's hand follows a specified trajectory. Generating the joint angles will require you to calculate Jacobians and perform both forward and inverse kinematics.

Description of Input Files:
Two files are provided in the project folder and will need to be inputs to your MATLAB script. The first file, which is called *arm*, contains a subset of the Denavit-Hartenberg parameters for the robot arm in its initial configuration. Specifically, the format for *arm* is as follows:

$$\begin{array}{cc} n & \lambda \\ l_1 & \theta_1(0) \\ l_2 & \theta_2(0) \\ \vdots & \vdots \\ l_n & \theta_n(0) \end{array}$$

where l_i is the length of link i and $\theta_i(0)$ is the initial angle of joint i. Note the value of λ (for Damped Least Squares) is included in the *arm* file.

The second input file is called *trajectory* and contains the desired end-effector trajectory for the robot. The file format is as follows:

$$\begin{array}{cc} m & \lambda \\ x(0) & y(0) \\ x(1) & y(1) \\ \vdots & \vdots \\ x(m) & y(m) \end{array}$$

where m is the number of desired positions specified for the robot. Note, m depends on the input file and should be a variable within your program. The pair $[x(i),\ y(i)]$ is the desired position of the end-effector at the ith frame.

From these two input files, you need to generate an output file called *angles* which specifies the joint angles needed to reach each point in *trajectory*. The format of *angles* should be:

$$\begin{array}{cccc} \theta_1(0) & \theta_2(0) & \dots & \theta_n(0) \\ \theta_1(1) & \theta_2(1) & \dots & \theta_n(1) \\ \vdots & \vdots & \vdots & \vdots \\ \theta_1(m) & \theta_2(m) & \dots & \theta_n(m) \end{array}$$

where m corresponds to the number of points provided in *trajectory* and n again refers to the total number of joints.

In short - your code should read two files, *arm* and *trajectory*, and output a single file called *angles* as shown in the block diagram of Figure 1.

Outline of General Procedure:
You will first need to import the provided input files, *arm* and *trajectory*. Consider using the *fscanf* command to read the files or use *dlmread* as for Project 1. Make sure the imported data matches the corresponding format as described. Using forward kinematics and the initial robot configuration described by *arm*, determine the location of the end effector at $t = 0$ using forward kinematics. Compare the actual end-effector location to the desired location as specified in *trajectory* to generate a position error.

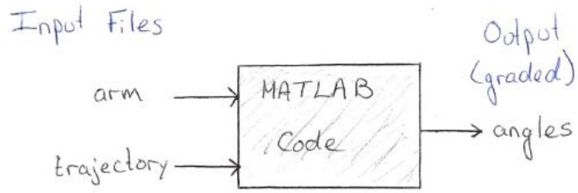

Figure 1: Box diagram of Input/Output for project

The next task is to reduce the position error by determining the joint angles required to reach the desired position, aka. perform inverse kinematics. You are to use the Damped Least Squares (DLS) technique which is iterative. Within the DLS loop, you first need to form the Jacobian of the end-effector for the current robot configuration. Using the Jacobian and given value of λ, perform DLS to calculate the change in joint angles, $\Delta\theta$ which are added to the initial angle values. Determine the robot's end-effector position using the updated angle values and repeat the angle update until the position error is satisfactorily small or you are unable to get any closer to the specified point (i.e. the desired trajectory may not be within the robot's reachable workspace). Store the total change in joint angles as this is the first row in your output file. Repeat the entire process for the next point in *trajectory*.

Additional Comments:
You can evaluate the performance of your robot in many ways. One suggestion is to plot the desired position and the actual position - a task accomplished by the single MATLAB script provided with the project. The script *check_angles.m* loads the *angles* file you generated, the *arm* parameters as needed for forward kinematics, along with the *trajectory* file. From these files, a plot of the actual positions and desired positions is generated along with a curve indicating the workspace boundary for your robot.

Another available check is provided through RobotStudio. In order to simulate your *angles* in RobotStudio, follow the *RobotStudio Instructions* provided in the project folder to unpack the *RobotStudio Workspace.rspag* package provided in the project folder. After the workspace is unpacked, you will need to move the *PositionGenerator.m* MATLAB script into the same folder as your other MATLAB scripts. Run *PositionGenerator.m* to reformat your *angles* and *trajectory* files so they are compatible with RobotStudio. Make sure the newly generated files are placed as described in the *RobotStudio Instructions* which may require manually moving the files.

What to Submit: When you are satisfied with the performance of you code, type a short README document as described in the *Project Guidelines* document from the course website. Email the README file and a .zip folder containing all MATLAB code to Megan at *mremmons@rams.colostate.edu* by the assignment deadline.