ROBOTIC ARM CONTROL

Focus:

The objective of this lab is to investigate the problem of positional control of robots by providing a series of motion instructions to the motors of a robotic arm.

Overview:

Robots have become important over a wide range of applications – from manufacturing, to surgery, to the handling of hazardous materials. Consequently, it is important to understand how they work and what problems exist in designing effective robots.

One significant area of robotics is positional control. For instance, one of a robot's functions is to move to a specified location or along a predetermined path so it can perform a task. Motion may consist of the robot itself moving, or of an articulated arm being actuated from a fixed pivot position. In many applications of robotics, tasks are easy to describe but difficult to implement. And, although simple sets of equations may be written to describe their operation, these equations may not have closed-form solutions.

Procedure:

The assignment is a virtual lab and is found at **http://www.jhu.edu/~virtlab.**

In this lab, we want to consider the problem of controlling the motion of a very simple articulated arm – a two-segment arm that can move only in the x-y plane and pivots about the position x=0, y=0. The tip of a two-segment robotic arm must traverse a narrow channel. Each segment of the arm is positioned by means of a "stepper motor" whose angular position and rate of change of position can be controlled. A stepper motor M_1 at (0,0) is attached to the first arm segment L_1 and controls the angle of L_1 (Angle 1) with respect to the x-axis. A second stepper motor M_2, affixed to the end of L_1, is attached to a second arm segment L_2 and controls its angle (Angle 2) with respect to the x-axis. Each arm segment is 100 units long, for a total maximum extension of 200 units. Neither of the arm segments may move below the base (y=0.)

As stated before, the objective is to devise a sequence of motor commands to move the tip of the second arm segment as far as possible along a prescribed channel. These commands will change the angles 1 and 2 as a function of time t as follows:

$$\text{Angle 1} = a_1 + b_1(t-t_0) + c_1(t-t_0)^2$$
$$\text{Angle 2} = a_2 + b_2(t-t_0) + c_2(t-t_0)^2$$

The coefficients **a**, **b**, and **c** determine the behavior of each motor: **a** is the motor's starting angle, **b** is the linear rate of change, and **c** is the acceleration of angular change.

Each motor command begins at time t_0 and will run for a duration of one time unit, i.e., the range of $(t-t_0)$ is 0 – 1.

When you begin the experiment, you will be presented with the experiment's display screen. Click **Begin/Clear**. This will produce a horizontal channel (randomly selected) through which you must move the tip of arm 2. Before you can begin to program the motor action to carry this out, you must first place the tip of the arm inside the channel, preferably as far to the right (or left) as possible. Do this by selecting values for the two leading coefficients (the **a**'s in the equations above.) Then click **Initialize**. This will reset the angles of the motors accordingly. If the tip still does not fall within the channel, enter new values for the coefficients and click **Initialize** until it does. Now you are ready to program the motors.

Deduce what combination of coefficients will move the tip on a trajectory that remains in the channel. Your objective is to move the tip from one end of the channel to the other. When the coefficients have been entered, click on **Run**. The arm will move according to your "program" either for a duration of $(t-t_0) = 1$ or until it runs into the channel wall. The arm will also stop if either of the arms hits the base (in black.) To move the tip further along the channel, you will need another program segment. Click on **Another segment**. The **a**'s will be automatically reset to the tip's current position. Enter a new combination of **b**'s and **c**'s. Start the motion for this new segment by clicking again on **Run**. Repeat the **Another segment / Run** sequence until you have successfully moved the tip to the opposite end of the channel.

Every time you run a new program segment, a segment counter is incremented. The object is to traverse the channel in the fewest number of segments. Clever programming can dramatically reduce the number of program segments. At any time, you can click on **Begin/Clear** to restart the problem. However, each click of **Begin/Clear** generates a different channel.

There are three modes in which to operate the experiment:

1) **Practice mode**: There is no constraining channel. Use this mode to familiarize yourself with the behavior of the arm segments with respect to the programming parameters.

2) **Easy mode**: The constraining channel is relatively wide. Large arm-motion deviations can be accommodated.

3) **Hard mode**: The constraining channel is relatively narrow. Only small arm-motion deviations can be accommodated.

To see how the arms behave, program a few arm movements in the **Practice mode**.

Problems and Programming Methods:

You will assign initial θ's (angles) so that the arm tip is at one end of the channel. From there you will want to program the θ's so that the arm tip moves to the opposite end. Most likely you will have to do this in a number of programming steps. For each step you must consider what effect changing the θ's will have on the *x,y* positions—especially the *y*-position. The position of the tip is given by:

$$
\begin{aligned}
x &= L\cos(\theta_1) + L\cos(\theta_2) \\
y &= L\sin(\theta_1) + L\sin(\theta_2)
\end{aligned}
$$

(Note: θ_1 = Angle 1, θ_2 = Angle 2) To describe how changes in these θ's relate to changes in the *y* - position , use the total differential to find that

$$dy = L\cos(\theta_1)d\theta_1 + L\cos(\theta_2)d\theta_2$$

For y to be constant, $dy = 0$. So changes in θ 's must conform to

$$d\theta_2 = -\frac{\cos(\theta_1)}{\cos(\theta_2)}d\theta_1 = k\ d\theta_1$$

But this ratio will keep y constant only at the initial values of θ_1, θ_2. As the values of the θ's change, *k* will also have to change. And that is the problem.

Problem 1:

Use the "easy" mode. To begin, pick θ's, say, θ_1^{init}, θ_2^{init}, so that the arm tip is in the channel. Calculate *k* for these angles. Call this value k_{init}. You can now program the stepper motors to change their angles over a period of one time unit. This is accomplished in the robot arm simulator by the coefficients of the term $(t - t_0)$. These coefficients represent angular change (degrees) per unit time. For example, if you want θ_1 to move through 50 degrees with respect to θ_1^{init} in one time unit and your constraint for $dy = 0$ is $k = k_{init}$ = -1.5, then you would enter coefficients for these terms as 50, -75. This would add 50 degrees over time to θ_1^{init} while simultaneously subtracting 75 degrees from θ_2^{init}. If you run into the channel boundary, you'll have to program another

segment—perhaps not with $dy = 0$. How many segments do you need to traverse the channel? Is this number dependent on the channel position, i.e., its vertical location?

Problem 2:

Use the "easy" mode. Rather than using $dy = 0$ as a criterion for programming, set up instructions which will change the θ's so that the arm tip will be in the channel both at the beginning and end of the program segment. This differs from the first problem, in that, here, you are specifying motion based on beginning and end conditions rather than on beginning conditions only. Again, how many segments do you need to traverse the channel? Is this number dependent on the vertical position of the channel?

Problem 3:

Begin with the "easy" mode, then try the "hard" mode. With the previous simpler programming schemes, the arm tip might run into the channel boundary because *k* would not be changing to accommodate the changing θ's. A way to improve on the two earlier schemes is to add a third parameter to refine the motion. This will allow you to, in effect, change *k* as a function of time.

The way to effect a changing *k* within a program step is to use the $(t-t_0)^2$ terms. Since non-zero coefficients here will produce "accelerations" in angular changes, an initial value of *k* can be linearly changed throughout a program step. One way to proceed using these acceleration terms is as follows: Let $\theta_1^{init}, \theta_2^{init}$ be the angles at the beginning of the program step. Choose a value for θ_1^{end}, and calculate an appropriate θ_2^{end} which will keep the arm tip in the channel at the end of the program step. Calculate k_{init} for the beginning angles. From this you can develop program coefficients to move the arms from $\theta_1^{init}, \theta_2^{init}$ to $\theta_1^{end}, \theta_2^{end}$. Let θ_1 change linearly in time from θ_1^{init} to θ_1^{end} with a $(t-t_0)$ coefficient for θ_1 equal to (θ_1^{end} - θ_1^{init}). Let θ_2 change according to k_{init} so that the $(t-t_0)$ coefficient for θ_2 equals k_{init}(θ_1^{end} - θ_1^{init}). Then, accelerate θ_2's change with a coefficient for $(t-t_0)^2$, so that, by the end of one time unit, θ_2 will have reached θ_2^{end}.

For example, suppose y = constant = 100; L = 100; $\theta_1^{init}, \theta_2^{init}$ = 30°. Further suppose that θ_1^{end} is to be 60°. Then θ_2^{end} will have to be 7.7° if the tip is to be at y = 100 at the end of the program segment. Since k_{init} = -1, the $(t-t_0)$ coefficients will have to be 30 and -30 for θ_1 and θ_2, respectively. But, now the change in θ_2 will have to change throughout the unit time increment so that at the end of the period θ_2 will equal θ_2^{end}, in this case 7.7 °. Thus, for this example, the program coefficients are

$$\text{angle } 1 = 30 + 30^* \ (t-t_0) + \ 0.0^* \ (t-t_0)^2$$
$$\text{angle } 2 = 30 - \ 30^* \ (t-t_0) + \ 7.7 ^*(t-t_0)^2 .$$

Note that at the end of a unit of time, i.e., $(t-t_0)$ = 1, θ_2 = 30 - 30 + 7.7 = 7.7
How good is this scheme compared to the others? Remember all these schemes are approximate.

Problem 4:

This is a theoretical problem. Calculate and plot the minimum and maximum horizontal limits of motion for a two-segment arm as a function of a channel's vertical position. Assume arm lengths of 100, an infinitesimal channel width, and no arm positions below y = 0.

Write-up:

Answer the questions. Support your answers with screen captures.