Q1. Three link planar robot manipulator [16 points]

Consider the three link manipulator shown in Figure 1. The lengths of the links are: OA = ℓ₁, AB = ℓ₂, and BC = ℓ₃. When numbers are needed, use ℓ₁ = 1, ℓ₂ = 0.8, and ℓ₃ = 0.3.

a) Derive an expression for the end point position \((x_C, y_C)\) as a function of the relative joint angles \(\theta_1, \theta_2, \text{ and } \theta_3\), and the link lengths. [3pts]

b) Write a MATLAB program to compute position \((x_C, y_C)\) of the end point C, when given the values of the joint angles \(\theta_1, \theta_2, \text{ and } \theta_3\). What is the end point position when \(\theta_1 = \pi/23, \theta_2 = \pi/21, \text{ and } \theta_3 = \pi/7\)? Angles always in radians, unless otherwise specified. [3 pts]

c) Write a MATLAB program to draw, approximately, the reachable workspace of the end point C of the three-link manipulator given the following joint ranges of motion: \(\theta_1\) can range between 0 to \(\pi/6\), \(\theta_2\) can range between 0 and \(\pi/2\), and \(\theta_3\) can range between 0 and \(\pi/7\). Show a well-labeled printout of the plot, in addition to uploading code. [6pts]

d) Write a MATLAB program to draw the three-link manipulator given values for \(\theta_1, \theta_2, \text{ and } \theta_3\). Show the output of your program for \(\theta_1 = \pi/3, \theta_2 = \pi/4, \text{ and } \theta_3 = \pi/6\). Drawing the manipulator = drawing the three line segments OA, AB, and BC. Be sure to label your drawing and use “axis equal.” [4pts]

e) Write a MATLAB program to show an animation of the three link manipulator, when the joint angles are given by the following functions of time \(t\):

\[
\begin{align*}
\theta_1(t) & = \sin(t) \\
\theta_2(t) & = 0.8 \cdot \sin(0.7t) \\
\theta_3(t) & = 2 \cdot \sin(1.2t)
\end{align*}
\]

for time \(t = 0 \text{ to } t = 10\). Make sure the animation is slow enough for the human eye to be able to see it, on your computer. [10pts]
Be sure to use “axis equal” for the plot, otherwise the scales on the plot will not be equal. For Q1, please upload the MATLAB code for (c,d,e) through Carmen. Make sure that when we run the code, we see the animation or other expected output, with nothing else to be done by the user.

Figure 1: A planar three link manipulator. C is the end point of the third segment. The point O is the origin (0,0)

Q2. Using fsolve. [12 pts]

a) Using fsolve, solve the equations $x^2 - y = 0$ and $x = 2y$. Upload program.

b) Consider the 3 link manipulator in Q1. For a given point $(x_C, y_C)$ inside the reachable workspace (not on the boundary), how many solutions for $\theta_1$, $\theta_2$, and $\theta_3$ do you think there might be? Explain.

c) Write a MATLAB program for inverse kinematics, to compute angles $\theta_1$, $\theta_2$, and $\theta_3$ using fsolve. Pick some point $(x_C, y_C)$ inside the reachable workspace (not on the boundary). Try different initial seeds to see what answers you get. Do you get the same answer for $(\theta_1, \theta_2, \theta_3)$?

Q3. Using inverse kinematics . . . [6 pts]

Consider the two-link manipulator that we discussed in class, with link lengths $\ell_1 = \ell_2 = 1$. Now, let’s say you want to make an animation of the two-link manipulator as the end-point $P_2$ goes from $(1,0)$ to $(0,0)$ at some “constant speed” (say). How would you do it?

First explain in words and then submit MATLAB code that achieves such an animation. Hint: inverse kinematics, but there are lots of ways to achieve this goal.

Q4. Splines and polynomial approximations [6 points]

Consider the function $f(t) = \sin(t)$ from $0 \leq t \leq 2\pi$. Evaluate this function at $N$ equidistant points. Using interp1, use a cubic spline to interpolate at a number of points between these $N$ points. How big is the error between the spline interpolant and the original sine curve? Try different $N$. Does the error go down with $N$ as expected?