Challenges faced during PA#3:

One of our main challenges we faced during our PA#3 was curve fitting, specific to the order of the point selection. As long as the x values of the co-ordinate points chosen by the user were either in ascending or descending order, the curve fitting by the spline method was smooth and followed the order of the points chosen, as seen in our simulation video and the below figure (a). Otherwise, the curve doesn't follow the points order. After our discussion with the professor, we figured out that the as the "interp1" function (vq = interp1(x, v, xq, method), where method = 'spline', 'linear', 'nearest', etc.) provides us data based on our input point values vector, since the x sample values in the vector aren't in an linear order, the subsequent values of vq generated from xq query points are not as well. Thus, based on the values received from the interp1 function with spline method, the curve is fitted accordingly. Another issue we faced was the robot movement when the curve was out of workspace bounds. We overcame this issue by moving the robot arm along the inner or outer circle perimeters, depending on whether the part of the curve lies inside or outside the workspace. We also use a theta buffer so that lefty or the righty solution will be used depending on the previous point to avoid the jitter issue.

For robot path designs, we experimented with other methods where the robot movement along the user defined points was maintained. We tried the Bezier curve for the design as shown in Fig. (b) and circumcircle approach for a circular curve fit between subsequent points as shown in Fig. (c).

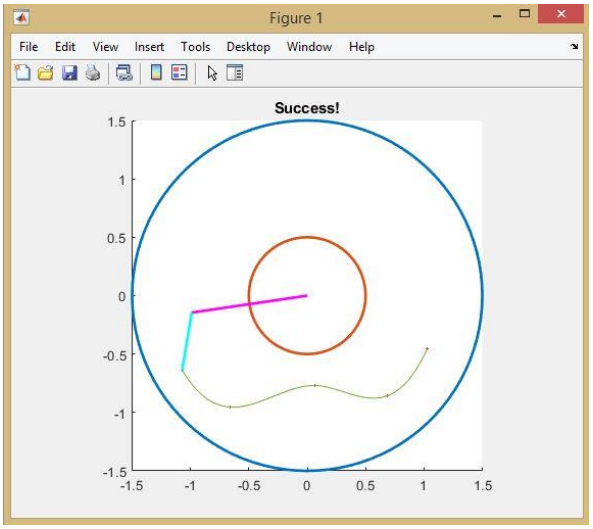

Fig. (a) : Spline curve fit

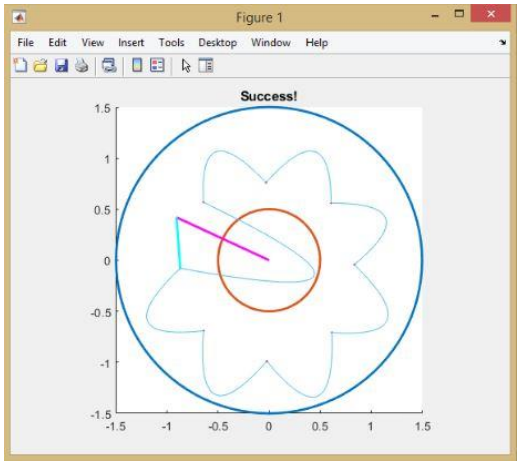

Fig. (b) Bezier curve fit

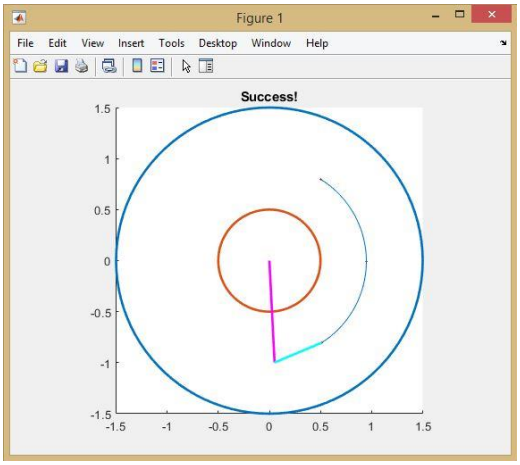

Fig. (c) Circumcircle curve fit