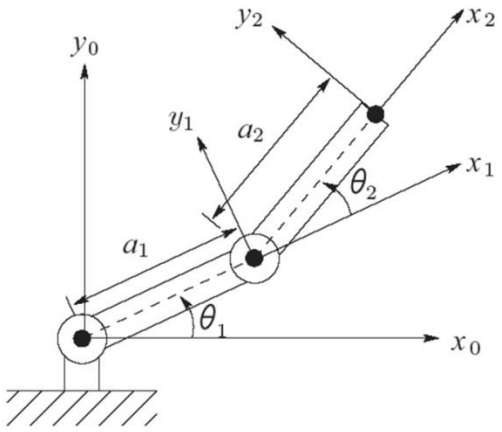
MAE 6245 (Spring 2020)

Robotic Systems

Assignment # 5

Total Points: 40

1) For the manipulator shown below, do the following (assume a_1=a_2=10) (ignore the frames shown in the figure – use your own frame definition):



a) Find the closed form inverse kinematic solution (using whatever method you like) to find θ_1, θ_2 given x, y (no orientation given). [This should be a simpler problem than the one worked out in class, and you may be able to make use of similar identities to work out the solution.]. Also write a MATLAB function to compute the joint angles, given x, y (using the closed form inverse solution directly).

b) Use one of the numerical techniques shown in class (root finding or optimization) to find the solution of the equation – given x, y. Basically create another MATLAB function to compute the angles, when end effector position is given.

c) Write a function in MATLAB to use the brute force technique to find θ_1, θ_2 given particular values for x, y. Use the allowable range as: $\theta_1 \in [0°, 90°]$, $\theta_2 \in [-90°, 90°]$.

d) Choose a circle in the workspace of the manipulator. [One way to choose the circle is to plot the workspace and find an equation of a circle that fits within the space. You are free to choose the radius. Try to choose something big enough that it is not miniscule, and yet fits within the workspace.] Figure out a way to define several points (10 to 20 points) around the circle (preferably equi-distant).

e) Write another function (or include it in the main script) to plot the robot configuration for given values of θ_1, θ_2.

f) Now, for each point on the circle, use the function in part (a) to calculate the inverse solution and plot it. Your function would like this:

```
tic
for each point on the circle
        Find inverse kinematics solution
```

```
        Plot robot configuration (overwriting previous plot)
    end
    toc
```

Note that you should be able to calculate the next point on the circle automatically. This process will be very laborious if you do it by hand.

Also note that "tic" and "toc" are MATLAB commands that give the time to execute any function that is defined in between them. So this will give you the time that the function took to execute.

g) Now repeat the process with functions developed in (b) and (c).

What are the times to execute each of these? Can all of them be used for real-time solutions?

You have just developed 3 control techniques to make the manipulator move in a circle. Congratulations !

[25 points]

2) For the problem solved in class using "fminunc" function, solve the same problem using "fmincon". Note that this is the function for constrained optimization, so you will first need to pose the problem as a constrained optimization problem [Hint: We discussed both formulations in class]. Compare results with the "correct" solution found in class.

[15 points]