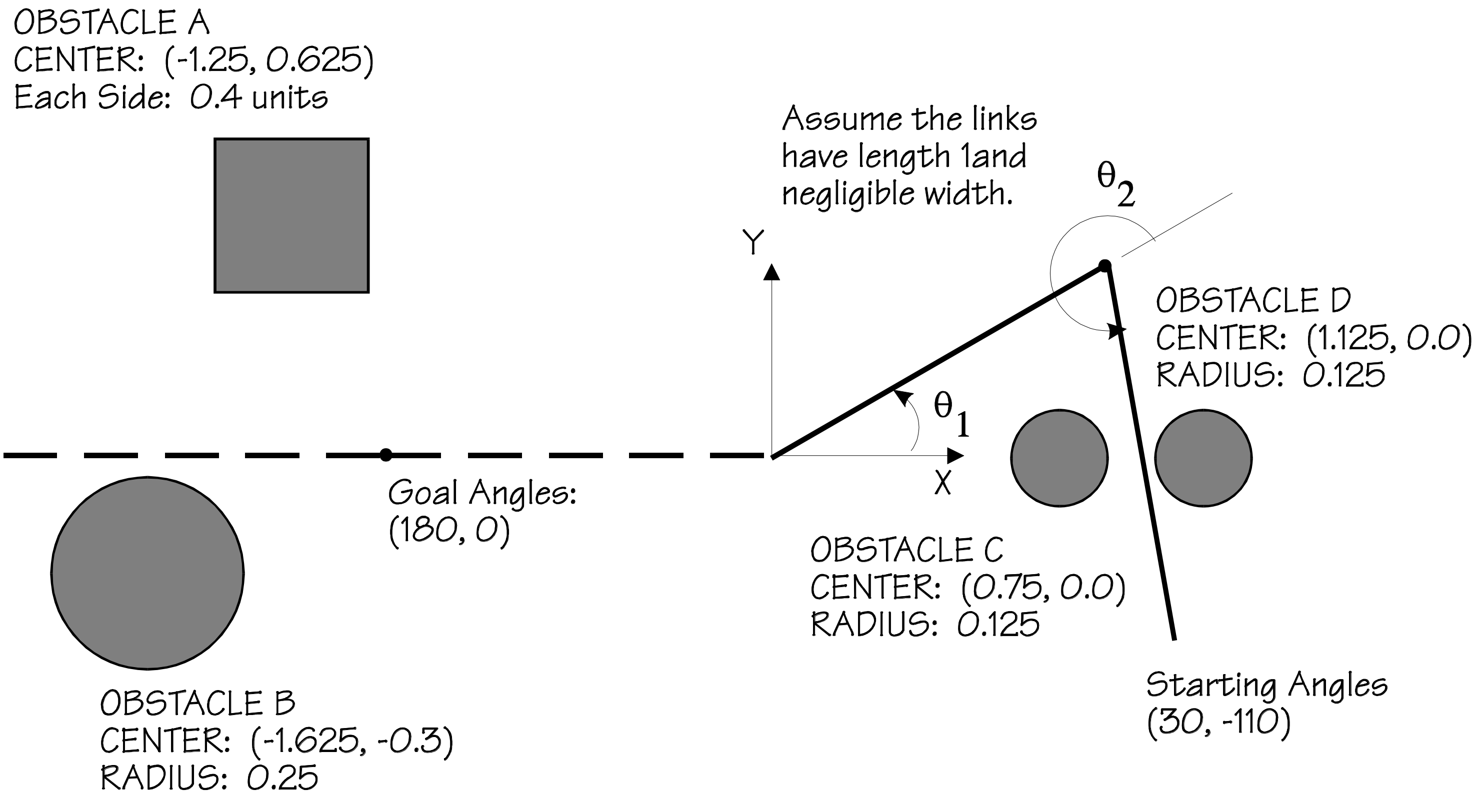
Spring 2016

Advance Robotics

Homework #4
Programming/Design Problem

March 11, 2016

Consider the two-link robot below. Note that its workspace contains some obstacles.

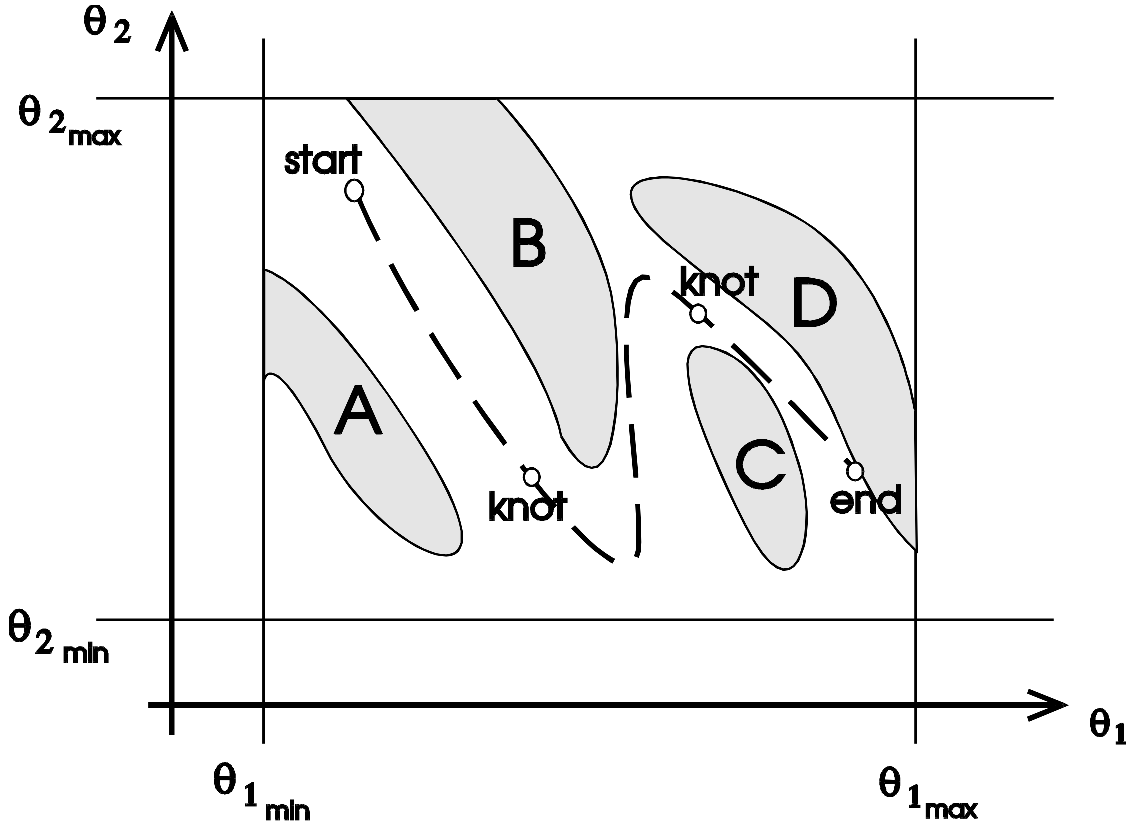


For this robot we will assume that θ_1 is limited to the range (-270, 270) and θ_2 is limited to the ranges (-135, 135).

We have seen that the inverse kinematics problem is a way of "mapping" a point in "XYZ-space" into "$\theta_1\theta_2$-space." In this problem, you will extend this idea so that entire objects (the obstacles) are mapped into "theta-world."

a) You are to write a computer program that creates a map of the manipulator's workspace, using coordinates θ_1 and θ_2 as defined above. Shade in the regions which are forbidden by the presence of the obstacles, in the same way the regions are shaded in the XY-coordinates above. Clearly mark which obstacle is which.

   That is, you are to create a map that will look something like this:



Note that in this world, the robot is just a ***point***, because a pair of angles completely specifies the configuration of the robot.

b) Afterward, you must plan a trajectory by choosing appropriate interpolation points. Use a starting point of $\theta_1(0) = 30°$, $\theta_2(0) = -110°$ and an ending point of $\theta_1(1) = 180°$, $\theta_2(1) = 0°$ (the final time need not be 1). You may have to choose several knot points, knot times, and spline functions as you see fit. You will want to choose knot points from the map you create, because it is more difficult to do from the XY-plot. Draw the resulting trajectory on the θ_2 vs. θ_1 plot you create. Also show plots of $\theta_1(t)$ and $\theta_2(t)$. The best trajectories will be the ones that use the fewest knot points (use at least two), and have the lowest-order polynomials functions.

Turn in the following:

i. A flow chart and explanation of the algorithm you use.
ii. A listing of your program (documented source code). (you may use any language you choose).
iii. The "theta-world" map you generate, showing the locations, shapes, and identities of all obstacles.
iv. The trajectory and locations of knot-points as drawn on your theta-world map *and* as time functions.
v. Equations for your spline functions in each interval of your trajectory.
vi. (optional) A figure showing the path your robot takes in XY-coordinates.

TIP: Start generating the map soon; the trajectory problem is not trivial and will take some trial and error.