Team Name: ByteMe
Connor Thompson
Chandler Garthwaite
Ian Brobin
Austin Albert

Lab 6 Write Up

1. If you are able to detect an obstacle, you can use a progressively adapting algorithm to gracefully avoid the obstacle and continue on course. First, the robot can attempt to use an IK feedback controller to go around the object without recalculating the path. This allows the robot to make a "free-form" path around the obstacle and resume the original path after it is avoided. This works well if a small box is placed in the middle of a room, for example. If the robot cannot solve a path around an object within a certain search threshold (like if a box blocks a doorway), then it can recalculate a new optimal path starting from its current position. Then use the code to convert the new Dijkstra's into new way points and continue using the new way points in your inverse kinematics code.

2.
    a. (1.2, 0.2) -> (0.225, 0.2) -> (0.225, 0.975)
    b. (0.9, 0.3) -> (0.72375, 0.3) -> (0.72375, 0.63825) -> (0.9, 0.63825) -> (0.9, 0.75)
    c. (1.2, 0.2) -> (0.225, 0.2) -> (0.225, 0.975)
    d. (0.225, 0.6) -> (0.225, 0.42675) -> (0.61275, 0.42675) -> (0.61275, 0.3) -> (1.35, 0.3)
3.
    a. Video Links: https://drive.google.com/open?id=1vxLaWVphhiGFHT81jtS_Kn31-Wd1ir6H
4. Connor Thompson, Chandler Garthwaite, Ian Brobin, Austin Albert
5. We spent ~6 hours programming this lab