Robotics
Exercise 4

Marc Toussaint
Machine Learning & Robotics lab, U Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany

November 8, 2013

1 The Dijkstra algorithm

Write a proper pseudo code for the Dijkstra algorithm on a general undirected graph $G = (V, E)$. A graph is defined by the set V of nodes and the set E of edges; each edge $e \in E$ is a tuple $e = (v_1, v_2)$ of nodes.[1] Determine the computational complexity of the algorithm.

2 RRTs for path finding

In our `libRoboticsCourse.12.tgz` in `teaching/RoboticsCourse/04-rrt` you find an example problem (rename `main.problem.cpp` to `main.cpp`).

a) The code demonstrates an RRT exploration and displays the explored endeffector positions. What is the end-effector's exploration distribution in the limit $n \to \infty$? Specify such a distribution analytically for a planar 2 link arm.

b) First grow an RRT *backward* target configuration $q^* = (0.945499, 0.431195, -1.97155, 0.623969, 2.22355, -0.665206, -1.48356)$ that we computed in the last exercises. Stop when there exists a node close (<`stepSize`) to the $q = 0$ configuration. Read out the collision free path from the tree and display it. Why would it be more difficult to grow the tree *forward* from $q = 0$ to q^*?

c) Find a collision free path using bi-directional RRTs (that is, 2 RRTs growing together). Use q^* to root the backward tree and $q = 0$ to root the forward tree. Stop when a newly added node is close to the other tree. Read out the collision free path from the tree and display it.

d) (Bonus) Think of a method to make the found path smoother (while keeping it collision free). You're free to try anything. Implement the method and display the smooth trajectory.

e) (Bonus) Follow the smooth trajectory using a sinus motion profile using kinematic control.

[1] Ideally, use the LaTeX package `algpseudocode` to write the pseudo code (see http://en.wikibooks.org/wiki/LaTeX/Algorithms#Typesetting_using_the_algorithmicx_package)