Robotics
Exercise 8

Marc Toussaint
Lecturer: Duy Nguyen-Tuong
TAs: Philipp Kratzer, Janik Hager, Yoojin Oh
Machine Learning & Robotics lab, U Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany

December 17, 2018

1 RRTs for path finding (8 points)

1. **Please update the version of the repository** by running the following.

```
git pull
git submodule update
make -C rai dependAll
make -j4
```

2. For python you can run: *'jupyter-notebook py/06-rrt/06-rrt.ipynb'*
3. For C++ run: *'cd cpp/06-rrt'*, *'make'*, *'./x.exe'*

The code demonstrates an RRT exploration that randomly samples from Q and displays the explored endeffector positions.

a) First grow an RRT *backward* from the target configuration $q^* = (0.945499, 0.431195, -1.97155, 0.623969, 2.22355, -0.665206, -1.48356)$. Grow the RRT directly towards $q = 0$ with a probability of $\beta = 0.5$. 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^*? (3 points)

b) 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. (3 points)

c) Propose a simple method to make the found path smoother (while keeping it collision free). Implement the method and display the smooth trajectory. (2 points)

2 A distance measure in phase space (4 points)

Consider the 1D point mass with mass $m = 1$ state $x = (q, \dot{q})$. The 2D space of $(q, \dot{q})$ combining position and velocity is also called phase space. In RRT's (in line 5 on slide 05:39) we need to find the nearest configuration q_{near} to a q_{target}. But what does "near" mean in phase space? This exercise explores this question.

Consider a current state $x_0 = (0, 1)$ (at position 0 with velocity 1). Pick *any* random phase state $x_{\text{target}} \in \mathbb{R}^2$. How would you connect x_0 with x_{target} in a way that fulfils the differential constraints of the point mass dynamics? Given this trajectory connecting x_0 with x_{target}, how would you quantify/meassure the distance? (If you defined the connecting trajectory appropriately, you should be able to give an analytic expression for this distance.) Given a set (tree) of states $x_{1:n}$ and you pick the closest to x_{target}, how would you "grow" the tree from this closest point towards x_{target}?