

# Robotics

## Exercise 7

Lecturer: Jim Mainprice

TAs: Philipp Kratzer, Janik Hager, Yoojin Oh

Machine Learning & Robotics lab, U Stuttgart

Universitätsstraße 38, 70569 Stuttgart, Germany

June 29, 2021



### 1 RRTs for path finding (7 points)

1. For python you can run: `'jupyter-notebook course1-Lectures/06-rrt/06-rrt.ipynb'`
2. For C++ run: `'cd ource1-Lectures/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 ( $< \text{stepSize}$ ) 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  $q^*$ ? (2 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 (5 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 4 on slide 07:29) we need to find the nearest configuration  $q_{\text{near}}$  to a  $q_{\text{target}}$ . But what does “near” mean in phase space? This exercise explores this question.

- a) Explain why the Euclidean distance in phase space is not a good measurement of “near”. (1P)
- b) Consider a current state  $x_0 = (q_0, \dot{q}_0)$ . A simple motion model is to use constant acceleration  $a$  that takes time  $T$  to reach a target state  $x_1 = (q_1, \dot{q}_1)$ . Come up with equations to calculate  $a$  and  $T$ . (1P)
- c) Given the solution from task b), how would you quantify/measure the distance? Give an analytic expression. (1P)
- d) Given a set (tree) of states  $x_{1:n}$  and you pick the closest to  $x_{\text{target}}$ , how would you “grow” the tree from this closest point towards  $x_{\text{target}}$ ? (1P)
- e) We can not go to every point in phase space with constant acceleration motion. Come up with example configurations  $x_0$  and  $x_1$  where this is not possible. What would happen to your distance measure when plugging in those configurations? Can we still use the distance measure from c) in the context of RRTs? (1P)

(a) Euclidean distance:  $d = \sqrt{(q_1 - q_0)^2 + (\dot{q}_1 - \dot{q}_0)^2}$

(b)  $\dot{q}_1 = \dot{q}_0 + aT$ ,  $q_1 = q_0 + \dot{q}_0 T + \frac{1}{2} a T^2$

(c)  $q_1 = q_0 + \dot{q}_0 T + \frac{1}{2} (a - \dot{q}_1) T = q_0 + \frac{1}{2} (\dot{q}_1 + \dot{q}_0) T$   
 $T = 2 \frac{q_1 - q_0}{\dot{q}_1 + \dot{q}_0}$ ,  $a = \frac{1}{2} \frac{\dot{q}_1^2 - \dot{q}_0^2}{q_1 - q_0}$