Homework 3

Robot Autonomy
CMU 16-662, Spring 2013

TA: Shervin Javdani

1 Introduction

This homework will cover sampling based motion planning and using Jacobians for motion planning. There will be two problems. In the first, you will implement an RRT to perform an object grasp. In the second, you will use Jacobians to move the end effector in a straight line while keeping the orientation fixed.

2 RRT

Implement an RRT algorithm in `hw3_rrt.py`. You can do a single directional RRT, which takes in a set of goals and plans to any of them. We'll make one change to the vanilla RRT as described in class - some of the time, you should force your algorithm to sample one of the goals and attempt to extend towards that (as opposed to a random sample in configuration space).

3 Jacobian Straight Line Planner

Implement a Jacobian based planner in hw3_jacobian.py to move the hand in a straight line while keeping the orientation fixed. You should move as far as you can forward up to the distance specified (but if you hit a collision or joint limit before that, return the trajectory up to that point). I left some functions there which I thought you might find useful (which show how to grab Jacobians and quaternions). Note that in OpenRAVE, you have two Jacobians - a spatial one (mapping joint differences to position differences) and a rotation one (mappings joint differences to quaternion differences). You'll have to combine them to simultaneously move forward and keep the orientation fixed.

Note: For quaternion differences, I believe OpenRAVE expects finite differences where you subtract two quaternions, and NOT the usual rotation in between. That is, if q_d is your desired quaternion and q_s your start, you have $q_{err} = q_d - q_s$.

(This isn't for your homework, but just general knowledge: What's the usual thing to do? You use the rotation between two, or $q_d q_s^{-1}$. And you don't look at the scalar element. But use the q_{err} I defined above for your homework!)

4 Deliverables and Grading

I've marked the sections that need to be filled in with a `TODO`. Please turn in your code, a pdf writeup, and each of the videos I requested (see below). I expect 5 files - a modified `hw3_rrt.py` and `hw3_jacobian.py`, a writeup in pdf form, and 2 videos. Only one person per group needs to submit, but please make sure everyone's name and andrewid is on the pdf.

Writeup items cannot exceed 5 sentences - get your point across quickly!

- In your writeup, describe how you would sample an item from configuration space for an RRT. (1 pt)
- Implement an RRT, and submit a video displaying it working. (7 pts)
- In your writeup, comment on the quality of this path compared to performing A* search. (1 pt)

- Now change the random seed and run it again (don't submit another video). Does it find a similar path? Comment on how this consistency (or lack thereof) might affect how useful the planner is. (1 pt)
- Describe your Jacobian planner. What is the equation being solved at every step? (3 pts)
- Implement the straight line Jacobian path planner, and submit a video displaying it working. (6 pts)
- Your Jacobian planner should run very quickly. However, comment on some potential downsides of using this type of method. (1 pts)