Robotic Manipulation Course

Exercise 5

Tran Nguyen Le

March 11, 2020

1 Prerequisite

Before starting this exercise you need to update the packages called *lumi_testbed*. If you have forked the repositories into your own Gitlab group and cloned them from there we advise you to first remove those repositories and instead clone them directly from `https://version.aalto.fi/gitlab/robotic_manipulation_2020`.

More specifically,

- cd into your *your_ros_ws*/*src*. Then type

  ```
  rm -rf lumi_testbed
  ```

- next, in the same folder, type

  ```
  git clone git@version.aalto.fi:robotic_manipulation_2020/lumi_testbed.git
  ```

Now the packages should be updated and you can start with this exercise. In this assignment you will use two robots instead of one. We have configured the two robot setup on a separate branch called *two_robots* in *lumi_testbed*. To switch to this branch, open the terminal and cd into your *lumi_testbed* folder. Once there, do the following:

```
git fetch

git checkout two_robots
```

Remember to recompile your code after this. To change back to the branch you used for the previous exercises do the same as above but instead of typing

```
git checkout two_robots
```

type

```
git checkout master
```

Now, you can clone the exercise5 folder to your ros workspace. **Before compile the ros workspace using catkin_make command, you need to remove or move the exercise4 folder away from the ros workspace.**

2 Assignment

The goal is to plan a stable 2D grasp for a polygonal object with two separate robots. **Assume a friction coefficient of 1 for all objects (This is extremely important!)**. The polygonal objects are specified as an Nx2 matrix where each row corresponding to an (x,y)-coordinate of a point so that a closed 2-D shape is obtained by connecting subsequent points by line segments (e.g. (x1,y1) to (x2, y2)) and by connecting the last point to the first ((xN, yN) to (x1,y1)). All vertices are given in centimeters. Your solution should work for any general polygonal object (triangle, square, pentagon, etc.). We have provided a test object in the file *exercise5/data/vertices.txt* which is visualized in Figure 1. To test with other objects you could simply delete or add lines in that file.

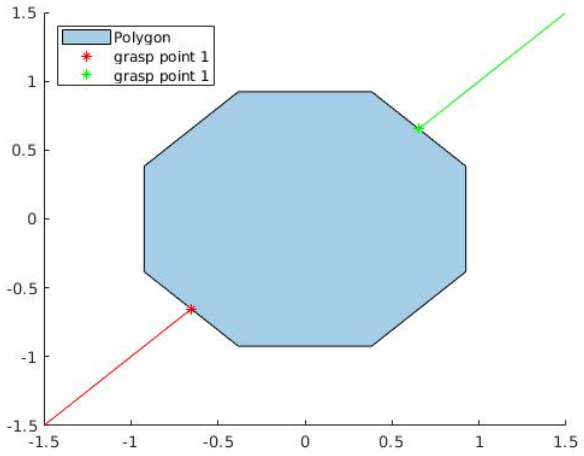

Figure 1: The polygon is centered at (0,0) in the x-, y-plane. **Note, this grasp is only an example of a stable grasp and you do not need to get the exact same one!**

Given this object, you are supposed to plan a stable 2D grasp using either python, matlab, or C++ (We are not providing any boilerplate code for the grasp planning due to the freedom of choosing programming language). The grasp itself is defined as two separate contact points and the angle between the x-axis and the grasp approach direction on the object. The grasp you calculate (one for each robot) are written into the file *data/grasp_data.txt* as a 2×3 matrix in the following form

```
x_1 y_1 alpha_1
x_2 y_2 alpha_2
```

where x_1, y_1, x_2, y_2 are the x and y coordinates of the grasp location for the first and second robot respectively. Because the units are different between the simulator (MuJoCo) and the units given for the vertices you need to scale all calculated grasping point x_i by 0.1 before writing it to the file. alpha_1 and alpha_2 are the angles between the approach vector and the x-axis for the first and second robot respectively. If, for example, alpha_1 represented the green line and alpha_2 the red line in Figure 1, then the actual angle for alpha_1 is 225° and for alpha_2 45°.

To test the grasps in the simulation environment, you need to use the original object given in *exercise5/data/vertices.txt* as this corresponds to the object in the simulator. Then, compile the workspace and use the following to launch the simulation with two robots

```
roslaunch exercise5 sim.launch
```

Then run the following script:

```
rosrun exercise5 run
```

which will read the data from the file *data/grasp_data.txt*, plan to the grasp position, and then apply force in the approach directions specified as alpha_1 and alpha_2 to the object. The expected behavior is for the contacts to stabilize for a short while, as in exercise4, but after some time they will diverge. The robots will not attempt to lift the actual object as this is one of the goals for the next assignment.

Bonus task: For the bonus task, instead of generating any stable grasp you are supposed to choose an optimal grasp according to a metric of your choice (*e.g.* robust to placement uncertainty). There are ample literature on this online and you are allowed and encouraged to implement one of those approaches. If you choose to do the bonus task you also have to write about the new results (more info about this in Section 3). If your implementation is correct and the answers are properly reported and discussed, it will grant you 30% extra points over the maximum number of possible points for this exercise.

3 Report

In addition to code, you are supposed to write a technical report (pdf) in which you will document the steps performed to fulfill the assignment. Your report should contains:

- your name, student number, date, exercise number and course name;
- The approach chosen for planning the grasp;
- How to run your grasp planning code;
- Visualization of the grasping points, approach direction, and friction cones on the object polygon;
- Discussion of the results and answers to the following questions:
  - Is the grasp both force-closure and form-closure, either of, or neither? Explain why.
  - If the object was not force-closure, where would we need to place additional contacts to achieve it?
- For the **bonus task** specifically report the approach and the metric used for planning the stable grasp;
- Estimate of time spent on this exercise.

4 Submission

To submit your code and report, fork a repository named *robotic_manipulation_2020/exercise5* to your gitlab. Modify the code in forked repository. Be sure to push your code before the assignment deadline. The commits pushed after the deadline will be ignored.

5 Deadline

Deadline for this assignment is 22nd of March at 23:59.