ROVI - Exercise 2

Petr Batek, Bjarki Páll Sigurðsson

October 8, 2016

1 Introduction

Purpose of this exercise was to get familiar with working with Robwork libraries and use simple planning algorithm RRT in our C++ project. We used provided template `pathplanning.cpp` as a starting point for our implementation. We extended template file with several functions. One is for creating a Lua file, which is used for animations in Robwork studio. Another function was written for creation of file with results from numerous simulations. These results were used further in order to find reasonable value for ϵ, which is the smallest step in radians for joints of robot manipulator.

2 C++ and Lua file

It was important to grab the bottle frame before we started path planner algorithm. This was done by commands shown on listing 1.

Listing 1: Grabbing Bottle frame into Gripper

```
// Finds the frame of gripper
const string gripperName = "Tool";
rw::kinematics::Frame *gripper = wc->findFrame(gripperName);

// Finds the frame of Bottle
const string itemName = "Bottle";
rw::kinematics::MovableFrame *item;
item = (MovableFrame *) wc->findFrame(itemName);

// Loads inital state of the scene
State state = wc->getDefaultState();
// Before Grapping the Bottle, manipulator must be se to correct possition
device->setQ(from, state);
// When Manip. is in correct possition, it can grip the frame
Kinematics::gripFrame(item, gripper, state);
```

We used provided `Lua` file as a template for creating animation scripts. In our function `createLuaFile` we first copy the first part of the script. After that there are inserted commands for setting of joints coordinates to values computed by C++ program. Finally the ending part of template is copied to new `Lua` file.

3 Chossing of ϵ value

To find reasonable ϵ value we had run several simulations, which statistics were finally compared. We selected a grid of different ϵ values and for each of them run 1000 of simulations. We set random seed for random number generator in order to receive different results for each simulation.

Histograms for 5 different values of ϵ are plotted on Figure 1. It can be seen that Computation Time and No. of Nodes are seriously affected by the value of ϵ. On the other hand total path length isn't affected in such a large scale.

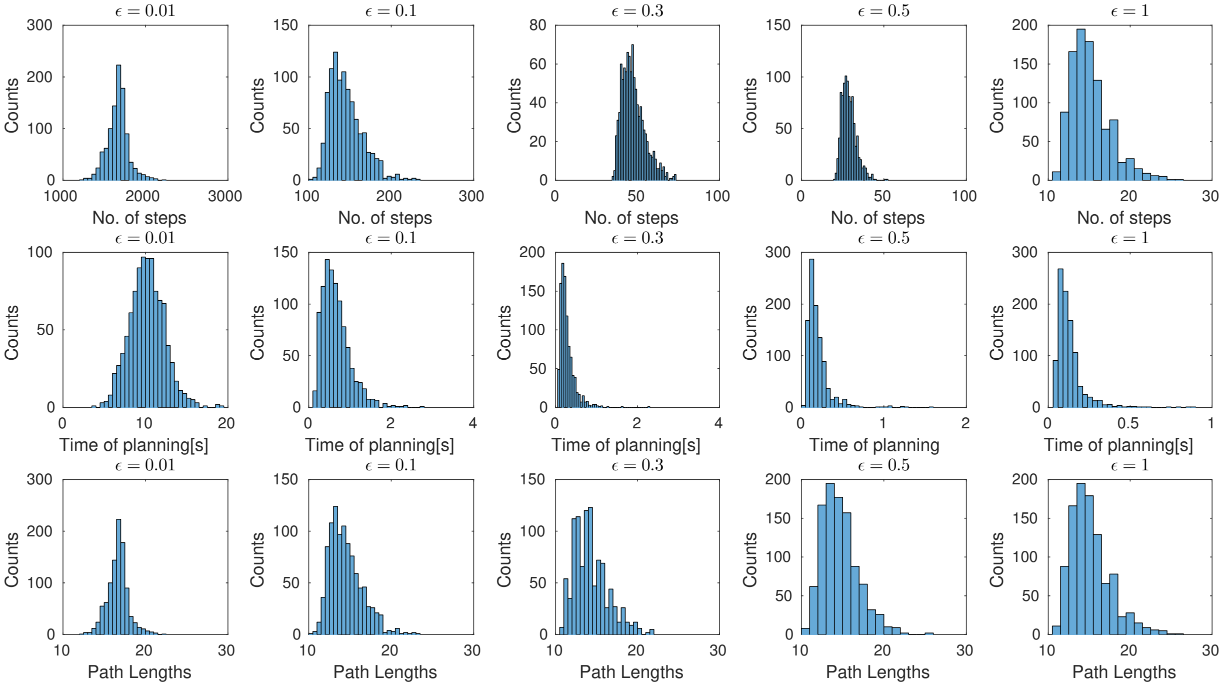


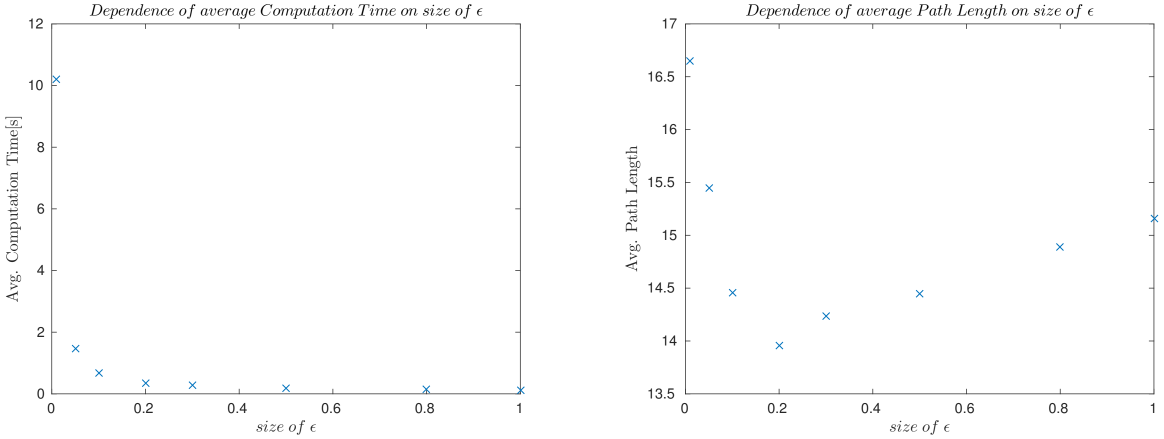
Figure 1: Histograms of simulations results

Mean values of simulation results were computed after that. We compared these parameters:

- average Number of Nodes = number of joints configurations before reaching the goal state
- average Computation time
- average Path Length = average total distance in joints coordinate frame, computed according equation:

$$PathLength = \#of\ Nodes \cdot \epsilon$$

There are results on the Figure 2. It is apparent that average computation time is decreasing for larger ϵ values. In contrast, Average Path Lenght isn't a monotone function. There is a minimum for value $\epsilon = 0.2$. It seems that this is our reasonable value.



(a) Average Computation Time

(b) Average Path length

Figure 2: Comparison of different Computation Time and Path Lengths for ϵ parameter

For another point of view we plotted a comparison of Average Path Length versus Average Computation Time into a single graph. It is plotted on Figure 3. This plot offers new insights into our problem. The closer is the point to the origin, the better performance regarding computation time and path length. This insight confirms, that value of $\epsilon = 0.2$ is suitable for our problem. From this graph it is also possible to predict, that any value in interval $\epsilon \in [0.1, 0.5]$ will result in short path with low computation time. ϵ with values 0.8 and 1 also results in low computation time, however the path length is considerable larger. If we consider parameter values lower than 0.1, it can be seen that they are worse in both aspects, longer computation time and total path length.

4 Conclusion

In this exercise we used Robwork RRT path planning method to find path for transfer object from start point to end point. We got familiar with Lua script used for animations in Robwork studio. We have found reasonable value for path planner parametr ϵ after performation of statistical tests. In order to find optimal ϵ value with respect to Computation time and Path length, it would be possible to create a finer grid of ϵ values and run statistical tests for each of them. Nevertheless we consider value of $\epsilon = 0.2$ as good enough.

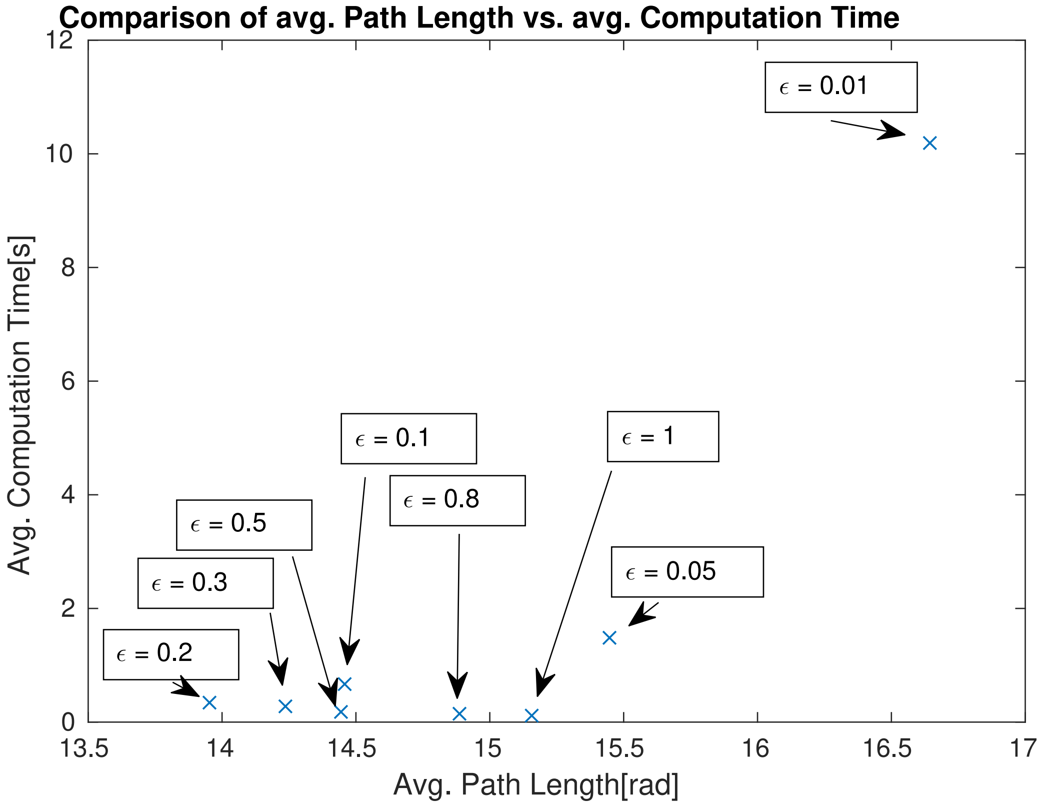


Figure 3: Average Path Length versus Average Computation Time