# CSCI 545 – Introduction to Robotics

## Lab 3 - Motion Planning with a 6-DOF Manipulator

Team 17 - QuirkyBot

Zishen Wei - Jiashang Cao - Srija Madarapu - Mahya Gheini - Zhen Qin

## 5 Observation

In Question 5, compared to Question 4, the robot is more likely to move toward the goal, whereas in Question 4, the robot is more likely to move in a random direction. Also, in Question 5, the robot moves more smoothly when close to the goal.

In Question 4, it has a quicker convergence when it gets close to the goal (the last step before it converges). Inversely, the Question 5 version converges quicker when it is not close to the goal because it has a 20% chance to sample around the goal points.

## 6 Problem of getting samples near the goal with probability = 1.0

If the robot only samples near the goal, it restricts itself to a smaller free space. A solution might be hard to find or not exist within this search space, so the RRT might take longer or never converge.

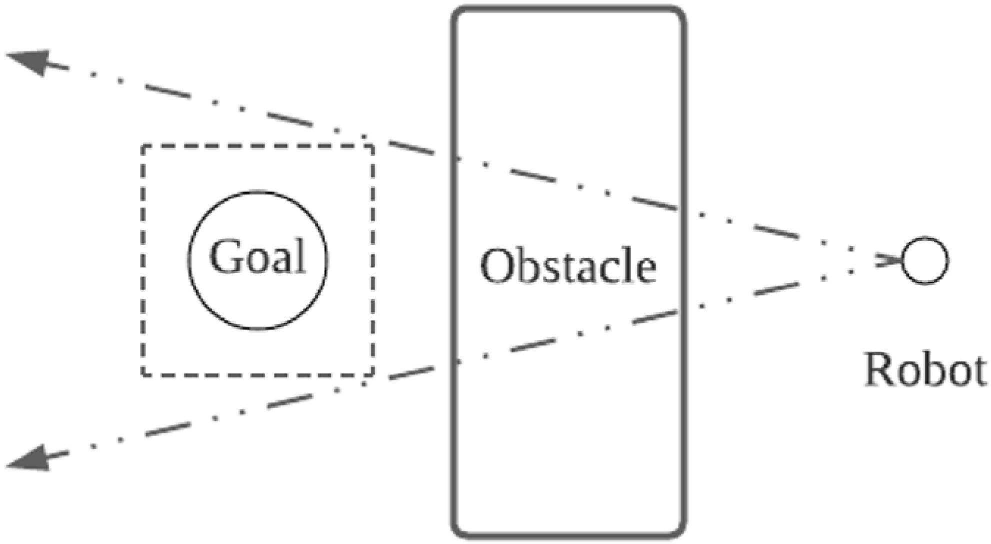


Figure 1: Example of a problem with sampling near goal with probability = 1.0

In the example presented in Figure 1, the sampling space is within the dotted box. The tree can only be developed within the area bounded by the two arrows. There’s no solution within this area; thus, the RRT will fail to converge in this case.

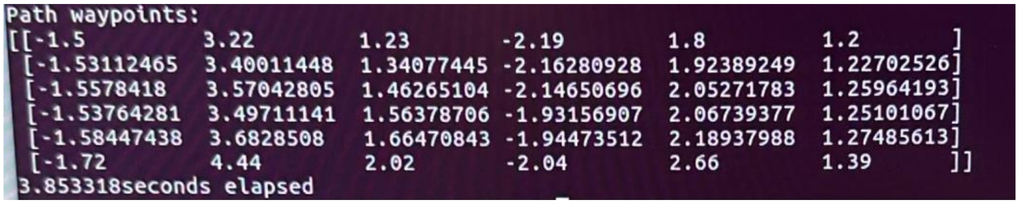
# In Lab Exercises

## 2 Cartesian vs. Angular velocity

Controlling the robot in cartesian mode was easier. Because the human mind is more familiar with cartesian space, for reaching the bottle, we would think the robot should go forward and straight down and things such as these. However, as it is mentioned in the instructions in cartesian velocity mode, the end-effector would try to maintain its pose as much as possible, so in cases that we needed to rotate the end-effector to reach a better pose for fingers to grab the bottle, angular velocity was more straightforward to execute.

## 3 Path planning with different goals

1. Goal = [-1.72, 4.44, 2.02, -2.04, 2.66, 1,39]

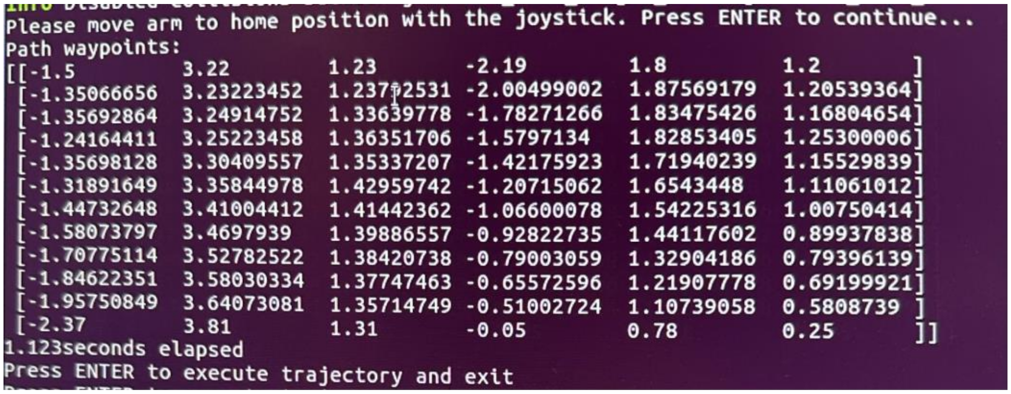


```
Path waypoints:
[[-1.5         3.22        1.23        -2.19        1.8         1.2        ]
 [-1.53112465  3.40011448  1.34077445 -2.16280928  1.92389249  1.22702526]
 [-1.5578418   3.57042805  1.46265104 -2.14650696  2.05271783  1.25964193]
 [-1.53764281  3.49711141  1.56378706 -1.93156907  2.06739377  1.25101067]
 [-1.58447438  3.6828508   1.66470843 -1.94473512  2.18937988  1.27485613]
 [-1.72        4.44        2.02        -2.04        2.66        1.39       ]]
3.853318seconds elapsed
```

Figure 2: Path to Goal#1

2. Goal = [-2.37, 3.81, 1.31, -0.05, 0.78, 0.25]

As we can see in Figure 3, it took the robot 1.123 seconds to find a path from the home position to the desired goal and execute the path.

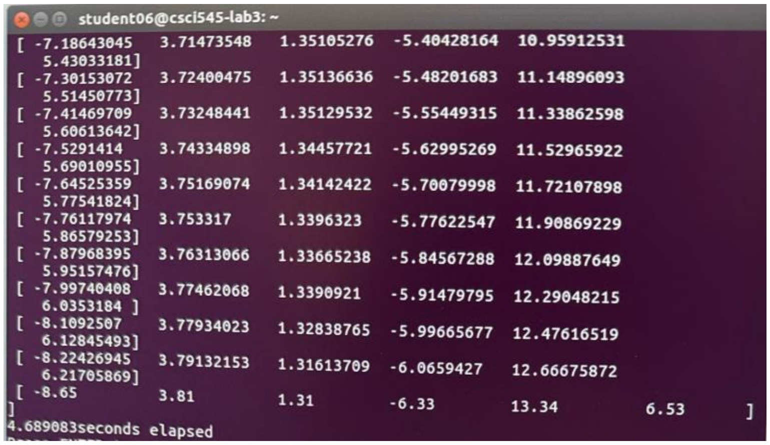


```
Please move arm to home position with the joystick. Press ENTER to continue...
Path waypoints:
[[-1.5         3.22        1.23        -2.19        1.8         1.2        ]
 [-1.35066656  3.23223452  1.23772531 -2.00499002  1.87569179  1.20539364]
 [-1.35692864  3.24914752  1.33639778 -1.78271266  1.83475426  1.16804654]
 [-1.24164411  3.25223458  1.36351706 -1.5797134   1.82853405  1.25300006]
 [-1.35698128  3.30409557  1.35337207 -1.42175923  1.71940239  1.15529839]
 [-1.31891649  3.35844978  1.42959742 -1.20715062  1.6543448   1.11061012]
 [-1.44732648  3.41004412  1.41442362 -1.06600078  1.54225316  1.00750414]
 [-1.58073797  3.4697939   1.39886557 -0.92822735  1.44117602  0.89937838]
 [-1.70775114  3.52782522  1.38420738 -0.79003059  1.32904186  0.79396139]
 [-1.84622351  3.58030334  1.37747463 -0.65572596  1.21907778  0.69199921]
 [-1.95750849  3.64073081  1.35714749 -0.51002724  1.10739058  0.5808739 ]
 [-2.37        3.81        1.31        -0.05        0.78        0.25       ]]
1.123seconds elapsed
Press ENTER to execute trajectory and exit
```

Figure 3: Path to Goal#2

3. Goal = [-8.65, 3.81, 1.31, -6.33, 13.34, 6.53]

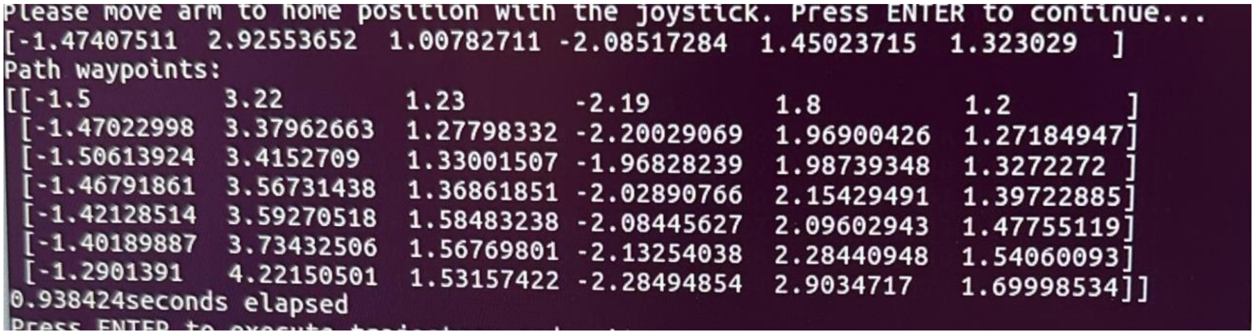
Our RRT could find a path from the home position to the desired goal without relaxing joint limit constraints. However, as you can see in Figure 3, it took the robot longer to find the path and execute it. (4.69 seconds)



```
student06@csci545-lab3: ~
[ -7.18643045   3.71473548   1.35105276  -5.40428164  10.95912531
   5.43033181]
[ -7.30153072   3.72400475   1.35136636  -5.48201683  11.14896093
   5.51450773]
[ -7.41469709   3.73248441   1.35129532  -5.55449315  11.33862598
   5.60613642]
[ -7.5291414    3.74334898   1.34457721  -5.62995269  11.52965922
   5.69010955]
[ -7.64525359   3.75169074   1.34142422  -5.70079998  11.72107898
   5.77541824]
[ -7.76117974   3.753317     1.3396323   -5.77622547  11.90869229
   5.86579253]
[ -7.87968395   3.76313066   1.33665238  -5.84567288  12.09887649
   5.95157476]
[ -7.99740408   3.77462068   1.3390921   -5.91479795  12.29048215
   6.0353184 ]
[ -8.1092507    3.77934023   1.32838765  -5.99665677  12.47616519
   6.12845493]
[ -8.22426945   3.79132153   1.31613709  -6.0659427   12.66675872
   6.21705869]
[ -8.65         3.81         1.31        -6.33         13.34         6.53      ]
]
4.689083seconds elapsed
```

Figure 4: Path to Goal#3

## 4 New goal configuration



```
Please move arm to home position with the joystick. Press ENTER to continue...
[-1.47407511  2.92553652  1.00782711 -2.08517284  1.45023715  1.323029  ]
Path waypoints:
[[-1.5         3.22        1.23       -2.19        1.8         1.2       ]
 [-1.47022998  3.37962663  1.27798332 -2.20029069  1.96900426  1.27184947]
 [-1.50613924  3.4152709   1.33001507 -1.96828239  1.98739348  1.3272272 ]
 [-1.46791861  3.56731438  1.36861851 -2.02890766  2.15429491  1.39722885]
 [-1.42128514  3.59270518  1.58483238 -2.08445627  2.09602943  1.47755119]
 [-1.40189887  3.73432506  1.56769801 -2.13254038  2.28440948  1.54060093]
 [-1.2901391   4.22150501  1.53157422 -2.28494854  2.9034717   1.69998534]]
0.938424seconds elapsed
```

Figure 5: Path to New Goal