Homework II:
Planning for a high-DOF planar arm

Andrew ID- rmohanan

1) Running the Code:

The planner should always be complied before running the **runtest** command on any map by following the steps as below:

1) Go to Code Folder
2) In MATLAB, run the command **mex planner.cpp** (For compiling)
3) In MATLAB To run the planner:

```
>>startQ = [0.826266 2.25291 1.72636 0.596928 2.16931];
>>goalQ = [0.335185 2.12194 0.588108 2.50622 0.405175];
>>planner_id = 0 % planning IDs…0-RRT,1-RRT-connect, 2-RRT*, 3-PRM
>>runtest('map2.txt',startQ, goalQ, planner_id);
```

Additionally, to run the 20 sample points all at once - Run the file **random_samples_run_file** by changing the planner_id for the required algorithm in MATLAB

2) Code Implementation:

i) The implementation is similar to the pseudo code provided in the lecture slide for RRT, RRT-connect, RRT* and PRM
ii) For PRM kd-trees is used for nearest neighbor search.
iii) Collision checking is done by interpolation between current config and sample config, checking each intermediate config for collision.
iv) The number of steps for interpolation used is 50 for RRT and RRT* and 80 for RRT-Connect.
v) Epsilon values for RRT -0.15, RRT-connect - 0.4, RRT*- 0.2, PRM-0.16
vi) Goal Based Sampling are done for RRT-Connect and RRT*
vii) Path Quality is the L2 Norm between vertices.

3) Results:

The sampling-based planner results for same 20 random sample points on MAP-2 are as below:

A) For RRT

Table 1

Iteration Number	Planning times (sec)	Solution generated in under 5 seconds	Number of vertices generated (in a constructed graph/tree)	Path qualities
1	0.2819	yes	454	7.201
2	0.0012	yes	10	1.895
3	0.0120	yes	240	5.706
4	0.0430	yes	674	13.890
5	0.0011	yes	10	1.039
6	0.0005	yes	6	0.262
7	0.0097	yes	159	1.586
8	0.0075	yes	233	2.281
9	0.0092	yes	172	7.239
10	0.0898	yes	1359	9.590
11	0.0760	yes	3084	12.501
12	0.0005	yes	4	1.213
13	0.0571	yes	943	6.185
14	2.2224	yes	57124	25.097
15	0.0007	yes	8	0.491
16	0.0008	yes	8	2.231
17	0.0125	yes	204	8.471
18	0.0046	yes	81	5.967
19	0.0098	yes	179	7.555
20	0.0779	yes	1305	4.944
Average Values	**0.1459**	**yes**	**3312.85**	**6.267**

B) RRT-Connect

Table 2

Iteration Number	Planning times (sec)	Solution generated in under 5 seconds	Number of vertices generated (in a constructed graph/tree)	Path qualities
1	0.1069	yes	98	14.324
2	0.0002	yes	1	4.803
3	0.0003	yes	2	3.069
4	0.0011	yes	163	10.683

5	0.0002	yes	1	2.453
6	0.0177	yes	4522	7.701
7	0.0002	yes	2	5.906
8	0.0059	yes	1263	14.075
9	0.0035	yes	364	12.044
10	0.0038	yes	473	7.072
11	0.0003	yes	10	3.224
12	0.0002	yes	1	5.625
13	0.0030	yes	393	14.673
14	0.0008	yes	148	5.245
15	0.0005	yes	42	8.448
16	0.0002	yes	1	2.684
17	0.0002	yes	1	3.921
18	0.0015	yes	287	10.644
19	0.0005	yes	53	6.429
20	0.0002	yes	2	5.982
Average Values	**0.0074**	**yes**	**391.35**	**7.450**

C) RRT*

Table 3

Iteration Number	Planning times (sec)	Solution generated in under 5 seconds	Number of vertices generated (in a constructed graph/tree)	Path qualities
1	0.1879	yes	3823	12.155
2	0.0004	yes	11	4.330
3	0.0083	yes	1394	6.118
4	0.0439	yes	9497	9.468
5	0.0002	yes	1	0.466
6	0.0142	yes	2437	1.781
7	0.0093	yes	131	8.051
8	0.0007	yes	85	2.354
9	4.2304	yes	7193	14.974
10	0.0383	yes	4354	7.814
11	0.1739	yes	35921	7.885
12	0.0130	yes	61	1.945
13	0.0452	yes	3851	6.352
14	0.6243	yes	122090	9.759
15	0.0010	yes	135	0.573

16	0.0003	yes	6	1.926
17	0.0148	yes	1715	4.171
18	0.0046	yes	475	3.753
19	0.0882	yes	7080	6.619
20	0.0023	yes	189	6.253
Average Values	**0.3094**	**yes**	**10022.45**	**5.837**

D) PRM

Table 4

Iteration Number	**Planning times (sec)**	**Solution generated in under 5 seconds**	**Number of vertices for sampling (in a constructed graph/tree)**	**Number of Edges generated (in a constructed graph/tree)**	**Path qualities**
1	1.2139	yes	15000	208844	4.278
2	1.1321	yes	15000	208394	6.033
3	1.1169	yes	15000	208040	4.314
4	1.3056	yes	15000	208046	11.193
5	1.1709	yes	15000	207560	3.201
6	1.0874	yes	15000	208516	13.694
7	1.1408	yes	15000	208030	7.965
8	1.1581	yes	15000	208096	13.140
9	1.0818	yes	15000	209088	10.768
10	1.1033	yes	15000	207672	8.013
11	1.1042	yes	15000	208276	7.002
12	1.0699	yes	15000	208488	7.530
13	1.1105	yes	15000	208346	10.086
14	1.1473	yes	15000	208520	5.661
15	1.1511	yes	15000	207912	9.720
16	1.0033	yes	15000	208680	3.342
17	1.0861	yes	15000	208066	5.415
18	1.1158	yes	15000	208658	4.591
19	1.0876	yes	15000	207974	7.292
20	1.0904	yes	15000	207600	8.756
Average Values	**1.1239**	**yes**	**15000**	**208240.3**	**7.600**

4) Results Comparison

A) Planning time

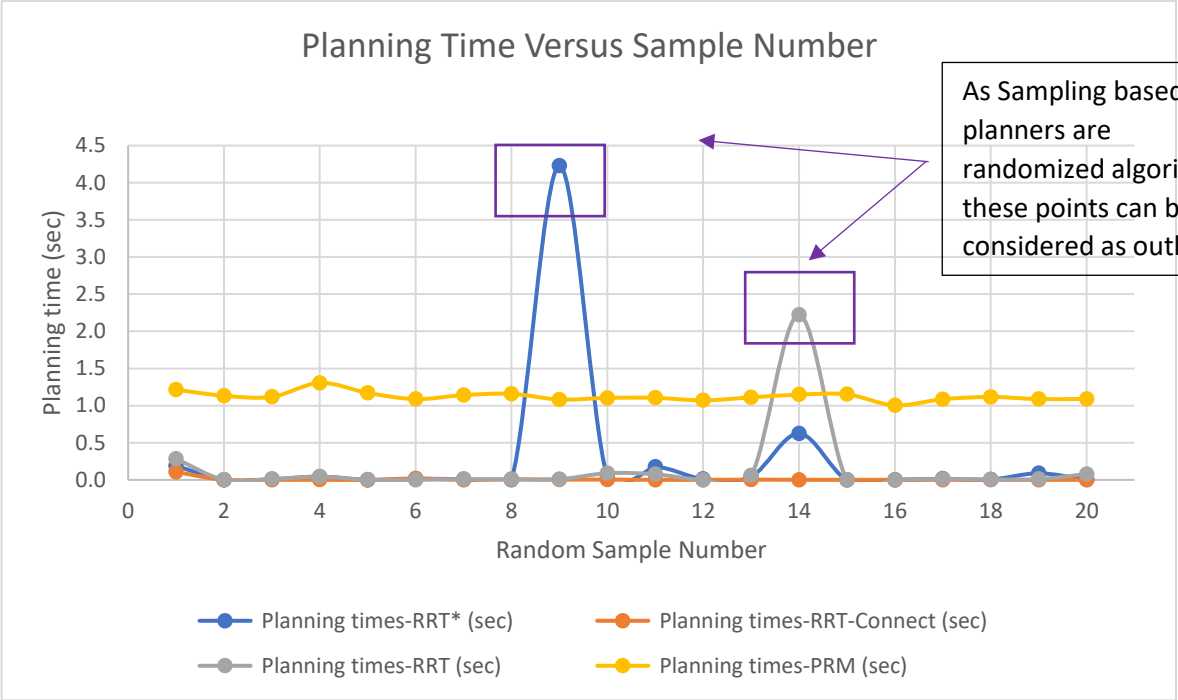

Figure 1

From the above sampling time comparison plot, it can be seen that the RRT-connect takes the least time for planning and PRM takes the largest time (ignoring the outliers)

B) Path Qualities

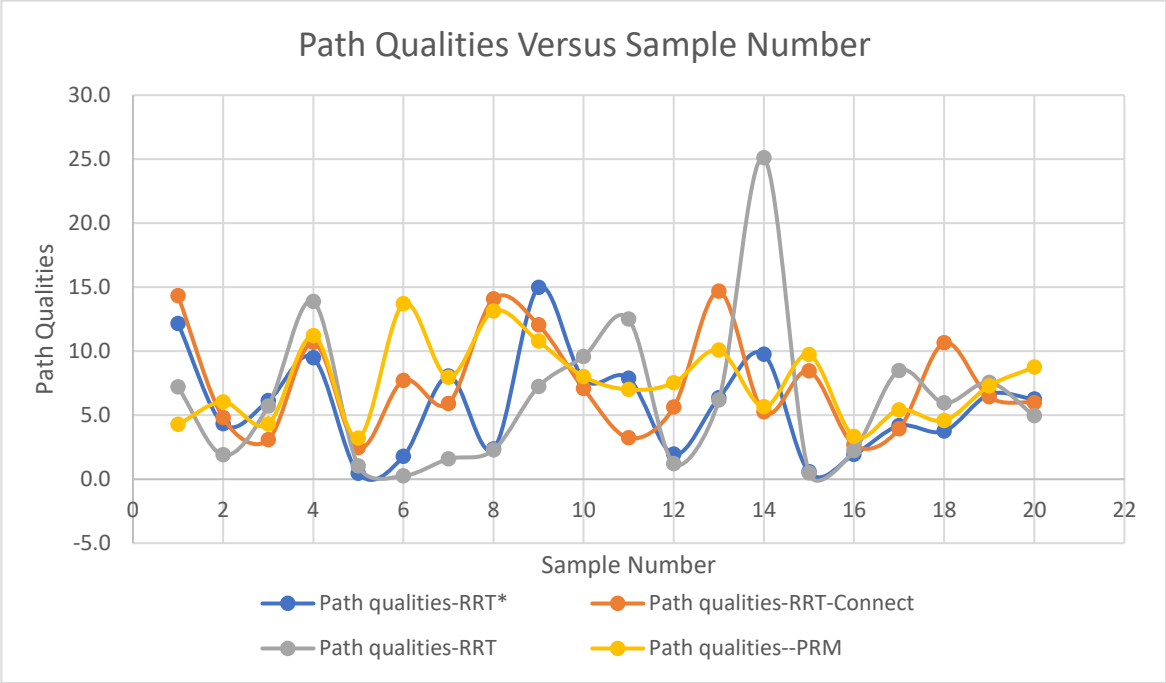

Figure 2

Above plot shows path qualities of RRT* mostly has values in the lower range as compared to other algorithms. This is because RRT* algorithm provides asymptotically optimal solutions

C) Number of generated vertices

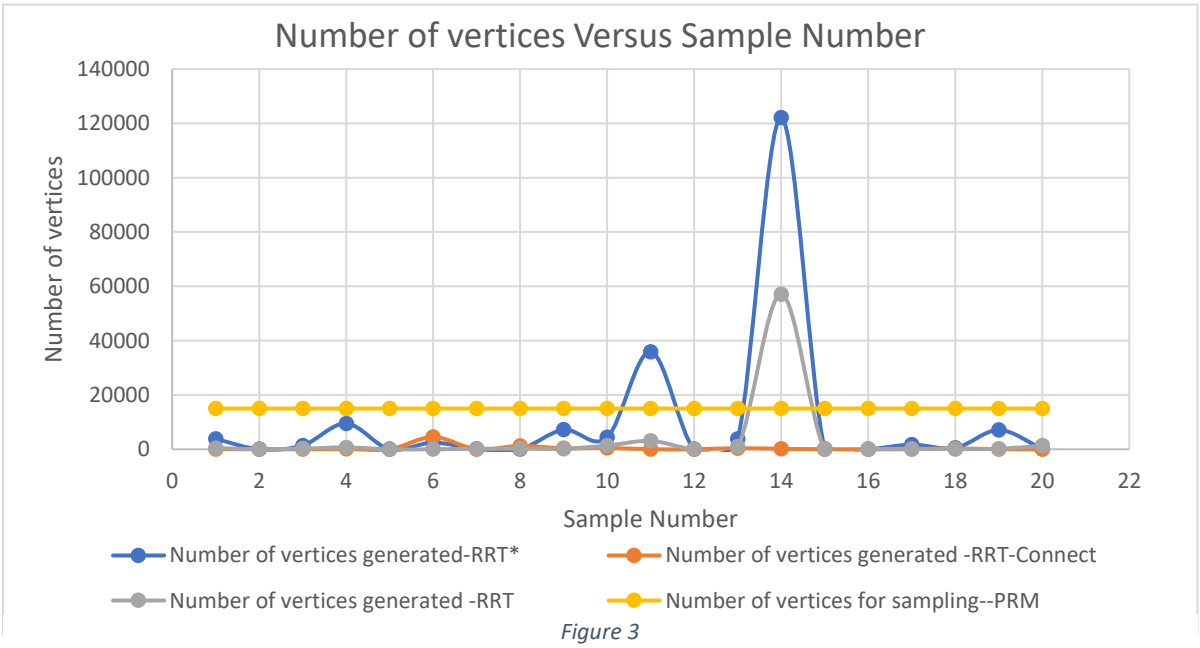

Figure 3

Figure 3 shows the least number of vertices are generated for RRT connect algorithm, whereas the PRM has constant values as they generate graph prior to search.

5) Summary Table

Table 5

Algorithms	Average planning times	Success rates for generating solutions in under 5 seconds	Average number of vertices generated (in a constructed graph/tree)	Average path qualities
RRT	**0.1459**	**100%**	**3312.85**	**6.267**
RRT- Connect	**0.0074**	**100%**	**391.35**	**7.450**
RRT*	**0.3094**	**100%**	**10022.45**	**5.837**
PRM	**1.1239**	**100%**	**15000**	**7.600**

Conclusion:

From the above plots and summary table we can see that the fastest sampling-based algorithm is RRT-Connect. Using RRT-connect will save a lot of computation effort, as very few nodes are generated. But in the case of dynamic environment or the environment where the goal position is not defined (not position but region is defined), it will be difficult to implement RRT- connect as RRT-connect requires both goal and start node to begin the planning. In such scenarios it will be useful to look for other sampling-based algorithms such as RRT* which will provide asymptotically optimal solutions. Among RRTs, RRT* is time consuming due to rewiring but it will take lesser time as compared to PRM. PRM is well-suited for repeated planning in between different pairs of qI, gG.

6) Extra Credit:

A) Variance Observed for Planning Times

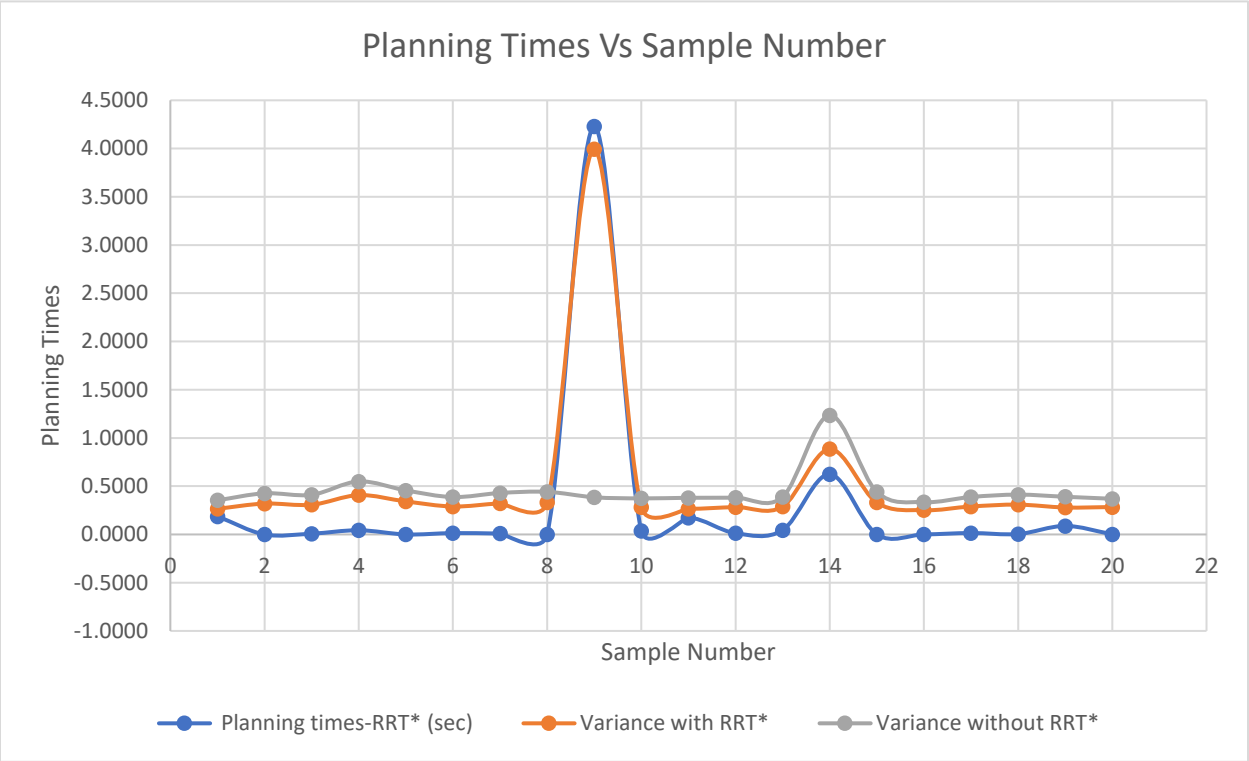

Figure 4

The above plot has planning times variance values of the algorithms(with and without RRT*) and RRT* values with respect to every sample number . It can be seen that major contributor of the higher variance values are RRT* values, as for sample number 9 due to the RRT* the variance value shoots up. Variance follows the trend of RRT*

B) Variance Observed for Path Qualities

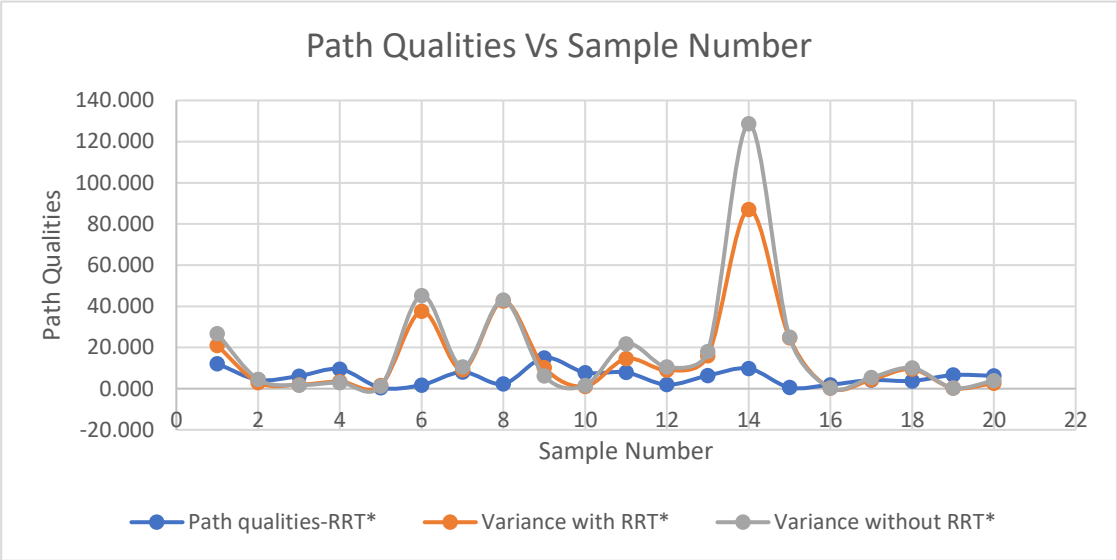

Figure 5

From the above plot it can be seen that RRT* does not contribute much to the spread of the data. The variance plot and RRT* plot does not follow the same trend for path qualities.

C) Variance Observed for Vertices Generation.

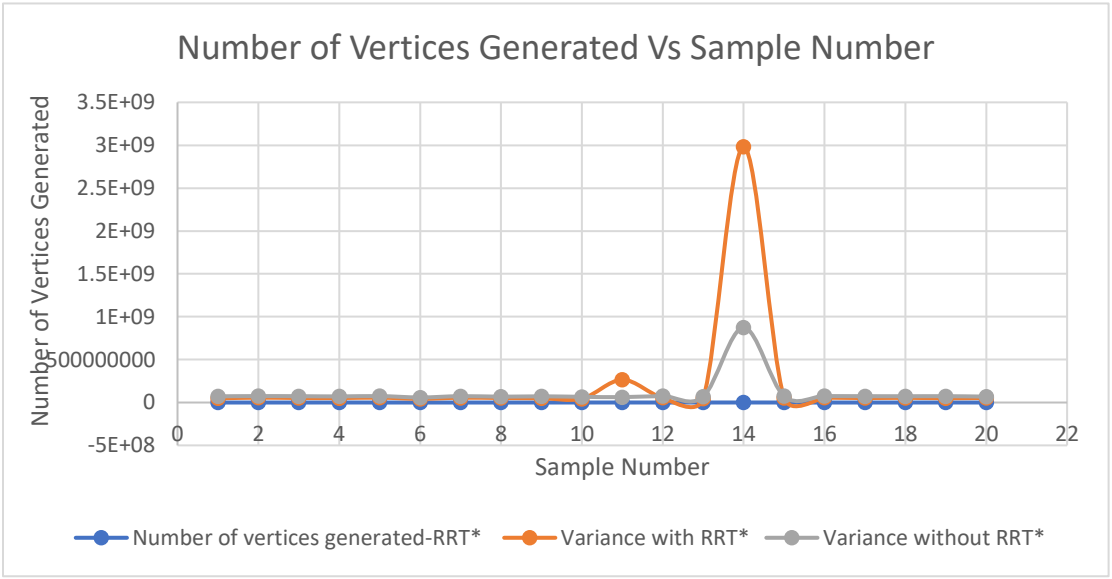

Figure 6

The RRT* values at sample 14 leads to higher variance value thus higher spread among the algorithm vertices number.