Kasi Karuppiah Alagappan
alagappan.k@husky.neu.edu

CS 4610/ 5335: Robotics Science and Systems (Fall 2017)

Directed Project : Project #3

Introduction:

Motion planning is a process of generating a discrete motion path satisfying the movement constraints of the robot and optimizing the motion between two locations. The key aspects taken into consideration are that the robot must not collide with any of the obstacles while moving under the assumptions that the obstacles are stationary and do not change shape. Solving of motion planning is very important as it increases the ability of a robot to do its job and reducing its dependability. The fixed base robots are widely used in the industries and their movements are quite limited. Exact motion planning solutions are slow and costly, hence most of these industrial robots use a fast but reliable solutions. One of the basic and overused approach is the sampling based motion planning method. The planner first constructs a roadmap consisting of free paths connected together in the environment. The robot can then travel on these free paths. Then the free paths are improved by using smoothing techniques.

Probabilistic Roadmap Path Planning:

Probabilistic roadmap (PRM) planners is the most basic and easy to implement algorithm that is widely used to solve motion planning problems. It is not the most exact motion planning solution but is fast and it can be used with many robots. This algorithm involves two phases, planning phase and querying phase. In the planning phase a network of connected free configurations are created and in the querying phase the network created in the planning phase is used to solve path problems.

Algorithm:

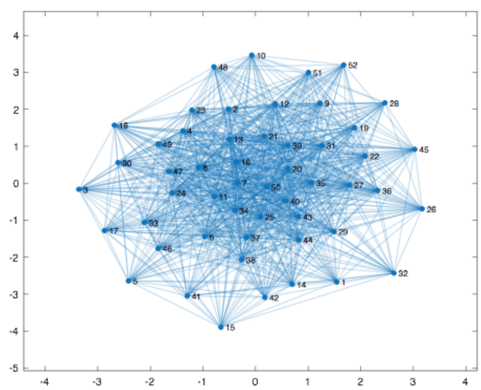
The PRM algorithm consists of the three parts

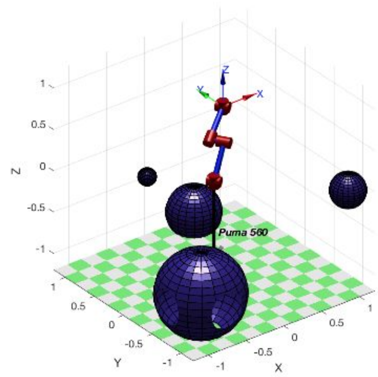
1. Constructing Roadmap
2. Querying for a Path
3. Path Improvement

Constructing Roadmap:

The algorithm starts with an empty roadmap. A free configuration is randomly generated using sampling method. If the configuration is in collision free space then it is added to the roadmap. This process is repeated until N vertices are chosen. Then a set of closest neighbours to the configuration space generated are chosen using range comparison and checked for collision. If a collision free path exists between the two configuration spaces then an edge between the two configurations is added to the roadmap.

The figure below on the left shows a roadmap generated with 50 free configurations connected to neighbouring configuration with range 1.0. The figure below on the right shows the puma 560 with collision models in 3D coordinate.





Querying For Path:

The initial and goal configurations are added to the roadmap and connected to the neighbouring configurations similar to free configurations added in the planning phase. Dijkstra's algorithm is used to find the shortest path from the initial configuration to the goal configuration.

Path Improvement:

Once a path is found using it is optimized using greedy approach by connecting configuration points in the path and reducing the number of configuration points between the initial and goal configuration.

Experiments:

Since the PRM is a sampling based model there are cases when the random configurations are generated there are cases when the initial and goal configurations don't end up in the same connected component of the generated roadmap. In these cases no actual path from initial to goal configuration is achieved.

Roadmap with 50 free configuration space

Range between Neighbours	0.5	1.0
Failures (100 runs)	23	0

Increasing the range to find k-neighbours solves a part of the problem and makes the roadmap dense.

Roadmap with range 0.5

Free Configurations	50	100
Failures (100 runs)	23	2

The problem could also be solved by increasing the number of free configurations but it doesn’t guarantee finding a path. As seen above increasing the free configurations to 100 does reduce the failures but doesn’t guarantee a path.

The above problem of failure to find path between initial and final configuration can sometimes be solved by recomputing the roadmap. The failures in the previous roadmaps with retry logic.

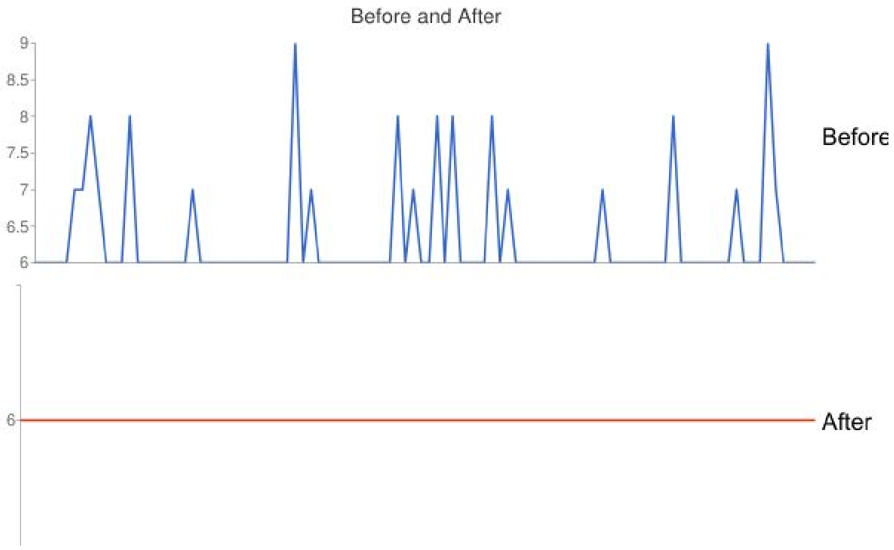
Roadmap with range 0.5 and retry

Free Configurations	50	100
Total Roadmaps (100	129	102

runs)		

The retry logic vastly improves the path planner algorithm and ensures a valid path is obtained. The max retries in roadmap with 50 free configurations was 4 and the max retries in roadmap with 100 free configurations was 2.

Path found using Dijkstra's can further be improved by smoothing the configurations. Below graph is a comparison of the smoothed results.



Results:

The results from the above runs show that PRM models are fast and works well with small roadmaps. In configuration spaces with narrow collision models it is difficult to obtain paths between initial and final configuration. In such cases more configuration points needs to be added to the roadmap. The goal would be to reduce the number of connected components to achieve full connectivity. This would increase the time required to build the roadmaps. The PRM model works fine with static environments but when used in real world applications then the roadmap needs to be rebuilt. This increase in the cost to recompute the entire roadmap is a drawback in PRM models. The path returned by querying is not the best path and needs smoothing to improve the wavering effect. The path are to be generally improved before feeding it to the robot.