Robot Motion Planning

Sampling Based Algorithms

A. Narayanan[1]

[1]Department of Informatics

August 15, 2022

Outline

Overview of Sampling Based Approaches

Probablilistic Roadmaps

Expansiveness

Difficulty with Classical Approaches

- Curse of dimensionality
- Running time increases exponentially with dimensions of configuration space
- Several variants of the path planning problem have proven to be PSPACE-hard

Drawbacks of Combinatorial Approaches

Multiple Query Roadmaps

A multiple query approach tries to capture the connectivity of the free space as good as possible, such that multiple, different queries for paths can be answered very fast. In other words: create a roadmap that is suitable for as many use cases as p

PRM - Probablistic Roadmaps

Basic steps to construct a PRM:

1. sample vertices and keep vertices that do not lie on an obstacle
2. find neighbour vertices - k-nearest neighbour or neighbours within a specified radius
3. connect neighbouring vertices with edges (lines) (and check for collisions on connecting line using e.g. discretized line search)
4. add vertices and edges until roadmap is dense enough

PRM Visualized

Drawbacks

PRMs don't perform well when there are narrow passages.

How to Improve?

- Increase number of milestones
- Random walk
- Path Correction
- Sample at obstacle boundaries

OBPRM - Obstacle Based PRM

Obstacle-based PRMs are constructed by sampling only close to obstacles. During sampling, the first goal is to find a point that lies inside an obstacle. Then, another point is sampled at an arbitrary distance to the first point. Using step-wise approximation, a point sufficiently close to the obstacle border is searched.

Single Query Roadmaps

Single query planners try to solve a single query as fast as possible, without trying to cover the whole free space

Weighted Randomized Tree Expansion

1. expand trees from start and goal
2. pick a node with probability $= 1/w(x)$, with $w(x)$ being the amount of neighbors within radius (measurement for exploration around x)
3. sample k points $(y_1, \ldots, y_k)$ around x
4. add y_i to the tree if
   - $1/w(y_i) > 1/w(x)$
   - y_i is collision free
   - y_i can see x
5. if a pair of nodes from start tree and goal tree are close and can see each other, then connect them and terminate

Rapidly Exploring Random Trees

1. pick q_{start} as the first node
2. pick a random target location (every n^{th} iteration, choose q_{goal})
3. find closest vertex in roadmap
4. extend this vertex towards target location
5. repeat steps until q_{goal} is reached

For faster execution, the tree can be grown from both q_{start} and q_{goal}

How to sample

Path Smoothening

Outline

Overview of Sampling Based Approaches

Probablilistic Roadmaps

Expansiveness

ϵ, α, β - Expansiveness