

Motion Planning

Ajay G

Introduction

Motion Planning address the question on how to reach the desired configuration without colliding the environment and itself, form given start configuration. It gives a trajectory, which on executing, will reach the desired configuration. Major approaches to motion planning are:

1. Search Based (Discretization)
2. Sampling Based
3. Potential Function

Basic Terminology

1. Configuration

A particular position, and oreintation, of the robotic arm is known as configuration.

2. Configuration Space

It is the space where each axis, corresponds to a particular joint angle of the robot. i.e., a N DOF robotic manipulator arm will have N dimensional Configuration Space. A particular configuration, will correspond to a point in C-space. Mostly it will be topological space.

3. DOF

“N” Degrees of Freedom means the arm has N independent motions which can execute simultaneously.

4. Path

A set of way points which on executing will reach the desired configuration.

5. Trajectory

A path with timed component such as velocity and accleration.

6. Map

A data containing the information about the environment and robot. It can be represented by three ways: topological, geometric, and grids.

Search Based

Discretize the C-space and solve as if finding a path in a graph. They are very sensitive to dimensionality of state space. Formulation of the problem:

1. State Space

They are usually grids. The choice of grid is crucial to performance and accuracy. In real world it is continuous, here we have approximated.

2. Action

Moving to a neighboring cell. Actions are also discretized.

3. Successor Function

A cell's neighbor. These are largely determined by the action set. Successors may not be known a priori. i.e., you have to try each action in your action set to see which cell you end in

4. Action Cost

We try to optimize the cost. Minimum path, smoothness, etc. Sometimes we use a linear combination of various cost.

There are several class of algorithms such as Breadth-First, Depth, Dijkstra's, Best-First, A* and its variant. Their major advantage is:

1. generality
2. cost minimization.
3. rigorous guarantees on completeness and quality of solutions.

However their main disadvantage is:

1. Not suitable for high-dimensional motion planning problem.
2. Not suitable when control set can't be easily discretized.
3. No uncertainty is taken into account.
4. Discrete search run-time and memory requirements are very sensitive to branching factor.
5. Number of successors depend on dimension

Note: Recent research work are working to cover and solve many of these issues and they might have been improved.

Sampling Based

Sampling based method overcomes many of the problems faced by search based method by weakening the requirements on the optimality and completeness.

Ideal	Practical in High Dimension
Complete	Probabilistically Complete
Optimal	Feasible (recent methods show asymptotic optimality)

Probabilistic completeness: A path planner is probabilistically complete if, given a solvable problem, the probability that the planner solves the problem goes to 1 as time goes to infinity. Feasibility: Path obeys all constraints (usually obstacles). A feasible path can be optimized locally after it is found.

The main idea is to pick a point in the C-space randomly and check whether a path exist, if it does, it is solved. Major sampling based methods are:

1. Rapidly Random Exploring Tree
2. Probabilistic Road Map
3. Randomized Path Planner

Rapidly Random Exploring Tree

Searches for a path from the initial configuration to the goal configuration by expanding a search tree. For each step, it determines a target configuration and expands the tree towards it. The target can either be a random configuration or the goal configuration itself, depending upon the probability value defined by the user. During expanding, the algorithm only needs to verify whether each step is collision free but does not need to avoid obstacles.

Probabilistic Road Map

Roadmap $R = (N, E)$ is class of topological map. It is an undirected graph of one dimensional curves in free C-space. The nodes N are a set of configuration of the robot. The edges E corresponds to (simple) path and edge (a, b) corresponds to a feasible path connecting the configurations of a and b. Some of the major types of roadmap are: visibility maps, deformation retracts, retract-like structure, piecewise retracts and silhouettes. It is called Probabilistic Roadmap (PRM) because it is constructed by repeatedly generating random(probabilistically) C-space free configuration of the robot and connecting these configuration using simple, fast, local planner. Then R postprocessed(expansion phase) to improve the connectivity. This entire process is known as the **learning phase**.

The final phase is the **query**. In which we ask for a path between two free configuration of the robot. First, It attempts to find a path from the start and goal configurations to the nodes of the roadmap. Next, a graph search is done to find the sequence of edges connecting these nodes on the roadmap. The learning and the query phase does not have to executed sequentially. Instead they can be interwoven. It is known as *multi-query* because if the robot configuration and environment didn't change much, we can query multiple number of times to find a new path from a single roadmap.

Pros of sampling based method:

1. Find feasible collision-free paths very fast.
2. Computationally efficient than Search based Algorithm
3. Scalable to high dimension

Cons of sampling based method:

1. Typically produce jerky / redundant motion.
2. Paths require post-processing, smoothing, optimization.
3. Completeness and optimality are compromised.
4. Probability of sampling an area depends on the area's size. i.e., hard to sample narrow passages.

Randomized Path Planner

Developed by Barraquand and Latombe in 1991 at Stanford. Main idea: Follow a potential function, occasionally introduce random motion. Potential field biases search toward goal. Random motion avoids getting stuck in local minima.

Advantage: Doesn't get stuck in local minima.

Disadvantage: define potential field, decide when to apply random motion, how much random motion to apply.

Potential Function

References