Motion Planning Algorithm

Basic Terminologies/Concept

- **Trajectory** is a path with timed component such as velocity and acceleration.
- **Optimization** is making Joint velocity, and accelerations, smooth and within the safety limit
- **C space** also known as Configuration Space is where each dimension of space is joint of the robot. i.e., N DOF robot will have a N dimensional configuration space
- We can represent **all the possible configuration** of robot simple as **point in C-space**.

A) Search-Based Planning Library

It uses **graph search** methods to compute paths or trajectories over a discrete representation of the problem. Because a graph is inherently discrete, all graph-search algorithms require this **discrete representation**.

- Formation of Graph
- Find best path in the graph
    - Grass-fire
    - Breadth first search
    - Depth fist search
    - Dijkstra's
    - A*
    - Weighted A*
    - D*

Pros

- Efficient solution

Cons

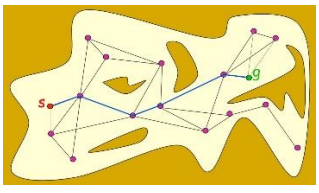
- Computationally expensive than all other methods
- Do not scale well to high dimensions due to heuristic approach

B) Open Motion Planning Library

Sampling-based algorithms

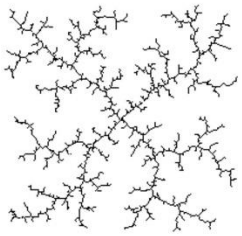
Instead of discretization of C- space, it uses the continuous C-space and pick points randomly, trying to find path. The two main approaches are PRM (Multi-query) and RRT(Single-query)

Probabilistic Road-Map-PRM



- Configurations are sampled by picking coordinates at random
- Sampled configurations are tested for collision
- The collision-free configurations are retained as milestones
- Each milestone is linked by straight paths to its nearest neighbors
- The collision-free links are retained as local paths to form the PRM
- The start and goal configurations are included as milestones
- The PRM is searched for a path from start (s) to goal (g)

Rapidly exploring Random Tree-RRT



- 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

	Multi-query	Single-query
Phases	1. Roadmap construction 2. Searching	Roadmap construction and searching online
Typical algorithm	Probabilistic Roadmap (PRM)	Rapidly Exploring Random Tree (RRT)
Pros	Fast searching	No preprocessing
Cons	Inability to deal with environment changes	No memory

Pros

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

Cons

- Typically produce jerky / redundant motion.
- Paths require post-processing, smoothing, optimization.
- Completeness and optimality

C) Covariant Hamiltonian Optimization and Motion Planning

Potential fields

- Form an artificial potential such that the robot is:
    - attracted toward goal and repelled by obstacle.
- Gradient of the potential field represent the overall vector summation of slope of each axis. I.e., they represent rate of change for a multi-variable function.
- Problem is the formation of local minima other than desired goal point.

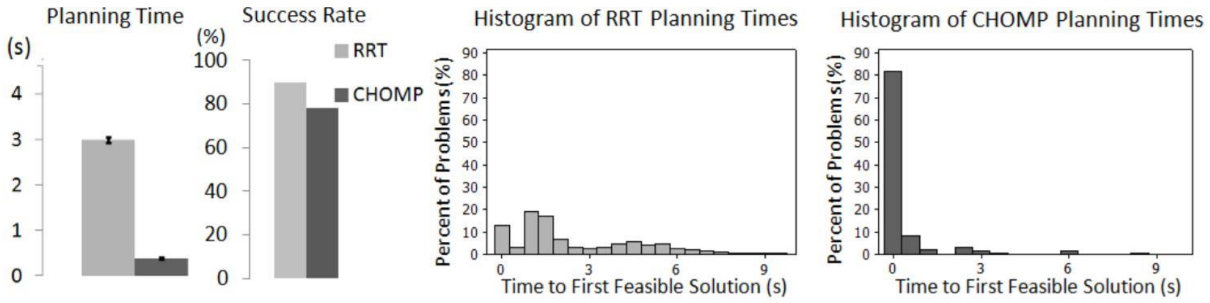
Principle

- Simultaneous Trajectory generation and Optimization.
- Everything is done in space of infinite dimension known as **Hamiltonian space.**
- Create a **naive initial trajectory** from start to goal.
- Define **cost** = trajectory smoothness cost + obstacle collision cost.

    - Obstacle collisions cost is cartesian distance between the point and obstacle transformed into C space and inflated.
        - Each cell has the above value and it forms a field know as **Distance Field**
    - The **obstacle** functional governs the **shape** of the path.
    - The **smoothness** functional governs the **timing** along the path.
- Run **Covariant gradient** on this cost function.
- Every update to the trajectory is ensured to be smooth.

Pros

Fluid motion and avoids unusual or awkward joint angles.



Cons

- Can get stuck in local minima due to gradient.
  Though theoretically sound, it is difficult to work with in practice.
- It introduces additional parameters which need to be tuned, and requires multiple random restarts to obtain a successful solution.

D) Stochastic Trajectory Optimization for Motion Planning

Stochastic - "Has random value which can be statistically analyzed but not predicted exactly"

It treats motion planning as an optimization problem, to search for a smooth trajectory that minimizes costs corresponding to collisions and constraints.

- It generates noisy trajectories to explore the space around an initial (possibly infeasible) trajectory
- They are then combined to produce an updated trajectory with lower cost.
- A cost function based on a combination of obstacle, smoothness cost, and **motor toque** is optimized in each iteration.
- No gradient information is required for the optimization algorithm that we use and so general costs for which derivatives may not be available (e.g. costs corresponding to constraints and motor torques) can be included in the cost function.

Pros

- Requires minimal parameter tuning
- Does not need cost function gradients.
- Motor torque is included in cost function

Cons

- Unable to find CON by googling ☹, should use this practically and go deeper into details as, "**devils are in the details**"