

Robot Motion Planning 2

Jiří Vokřínek

A4M36PAH - 7.4.2014

Literature

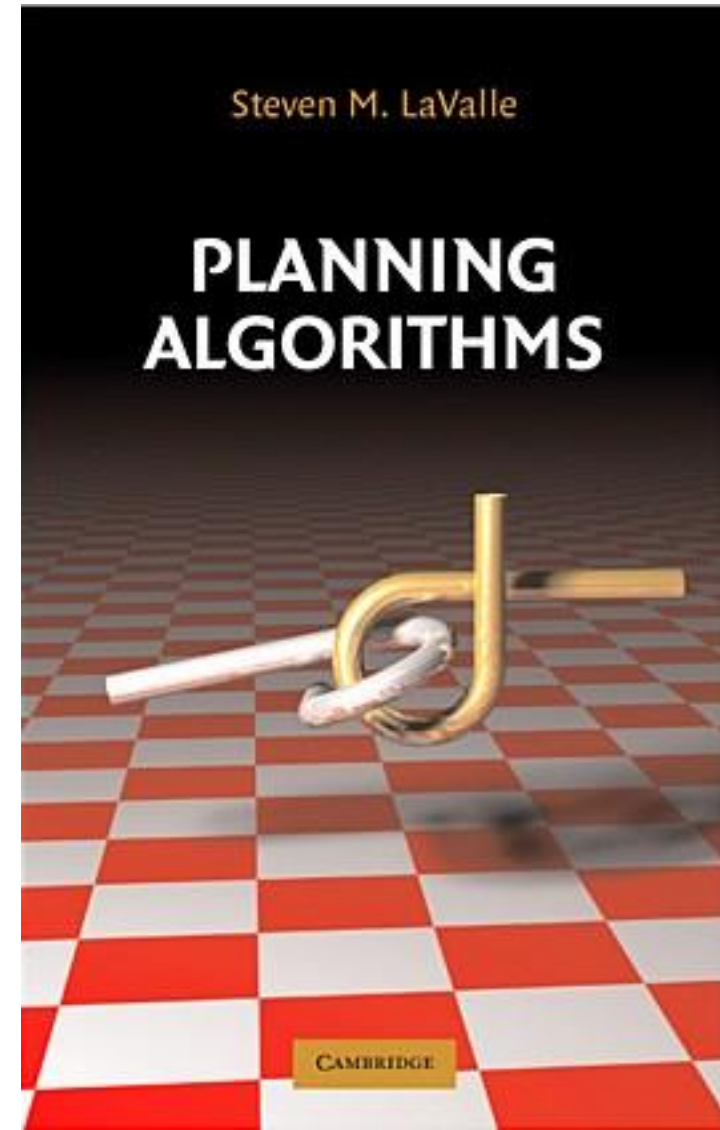
Steven M. LaValle.

Planning Algorithms.

Cambridge University Press, 2006.

Available online*:

<http://planning.cs.uiuc.edu/>

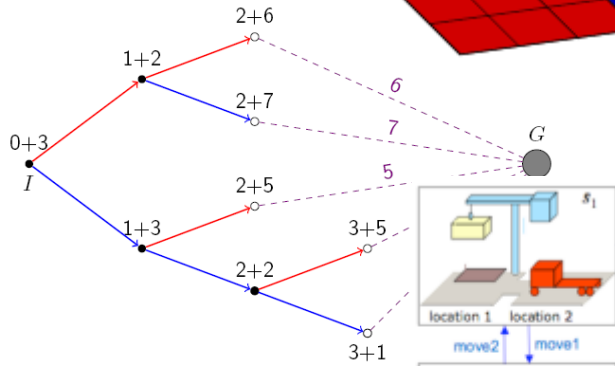
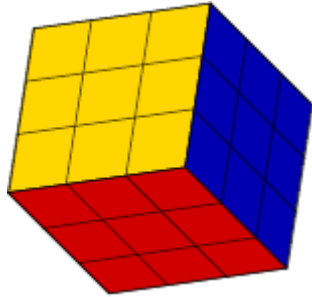
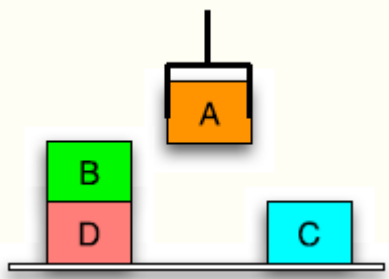


*many pics are taken from here

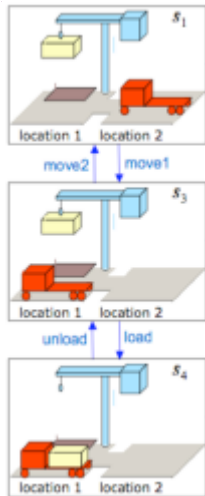
Robotic Planning Problems

- Localization
- Mapping and Navigation
- Collision detection/avoidance
 - Obstacles
 - Other robots
- Motion planning
 - Roadmap, visibility graphs
 - Cell decomposition
 - Potential fields
- Coverage planning

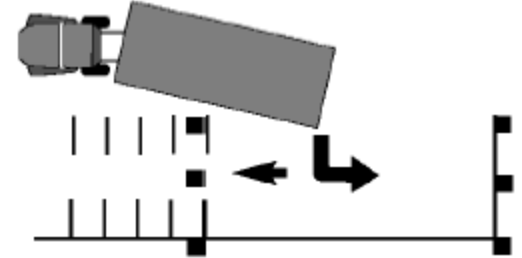
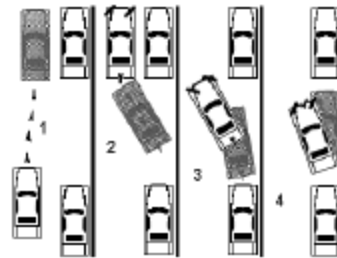
Robotic Planning Problems



1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	



VS



Robotic Planning Problems

- Goal:
 - Compute motion strategies (geometric time-parameterized paths/trajectories)
 - Move to the specific position
 - Build a map of the region
 - Find a target, explore an area
 - Assemble/disassemble parts

Robotic Planning Problems

- Problem: compute a collision-free path for a moving object among static obstacles
- Input:
 - Geometry of a moving object and obstacles
 - Kinematics of the robot (degrees of freedom)
 - Initial and goal robot configurations (positions & orientations)
- Output: continuous sequence of collision-free robot configurations connecting the initial and goal configurations

Robotic Planning Problems

- Configuration space
- Sampling-based motion planning
- Combinatorial motion planning

- Several variants of the path planning problem have been proven to be PSPACE-hard.
- A complete algorithm may take exponential time (complete algorithm finds a path if one exists and reports no path exists otherwise).

Problem Formulation for Point (Holonomic) Robot

- Input:
 - Robot represented as a point in the plane
 - Obstacles represented as polygons
 - Initial and goal positions
- Output:
 - A collision-free path between the initial and goal positions

Problem Formulation for Point Robot

continuous representation

(configuration space formulation)



discretization

(random sampling, processing critical geometric events)

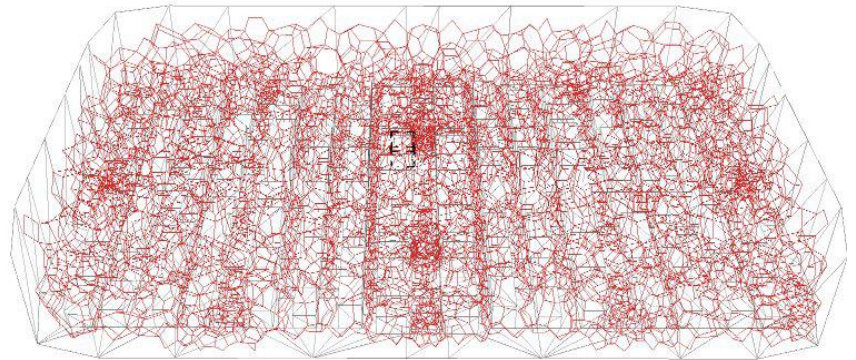


graph searching

(breadth-first, best-first, A*)

Random Sampling Methods

- Probabilistic roadmaps
- Rapidly exploring random tree



Probabilistic Roadmaps

- Randomly generated discrete representation of the continuous space
- Graph consisting of
 - **Nodes** – admissible configuration of the robot
 - **Edges** – feasible path between the particular configurations

Probabilistic Roadmaps

- Randomly generated discrete representation of the continuous space
- Graph consisting of
 - **Nodes** – admissible configuration of the robot
 - **Edges** – feasible path between the particular configurations
- Probabilistic complete algorithms: with increasing number of samples an admissible solution would be found

Probabilistic Roadmaps

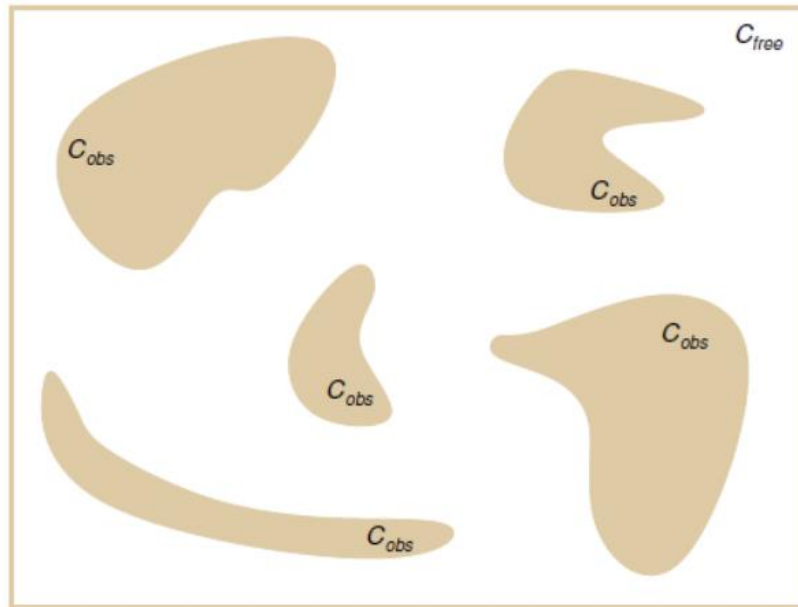
- Multi-query
 - Single roadmap used for planning queries several time (PRM)
- Single-query
 - New roadmap for each planning problem
 - Configuration subspace relevant to the problem
 - i.e. RRT (rapidly exploring random tree)

Multi-query Strategy

- Learning phase
 - Find random (non colliding) samples of Configuration space (**node**)
 - Connect random configurations using local planner (**edge**)
 - A connection represent admissible path between configurations
 - Consider only nodes within $\varepsilon > 0$ distance
 - Collision detection can be performed for configurations “on the edge”

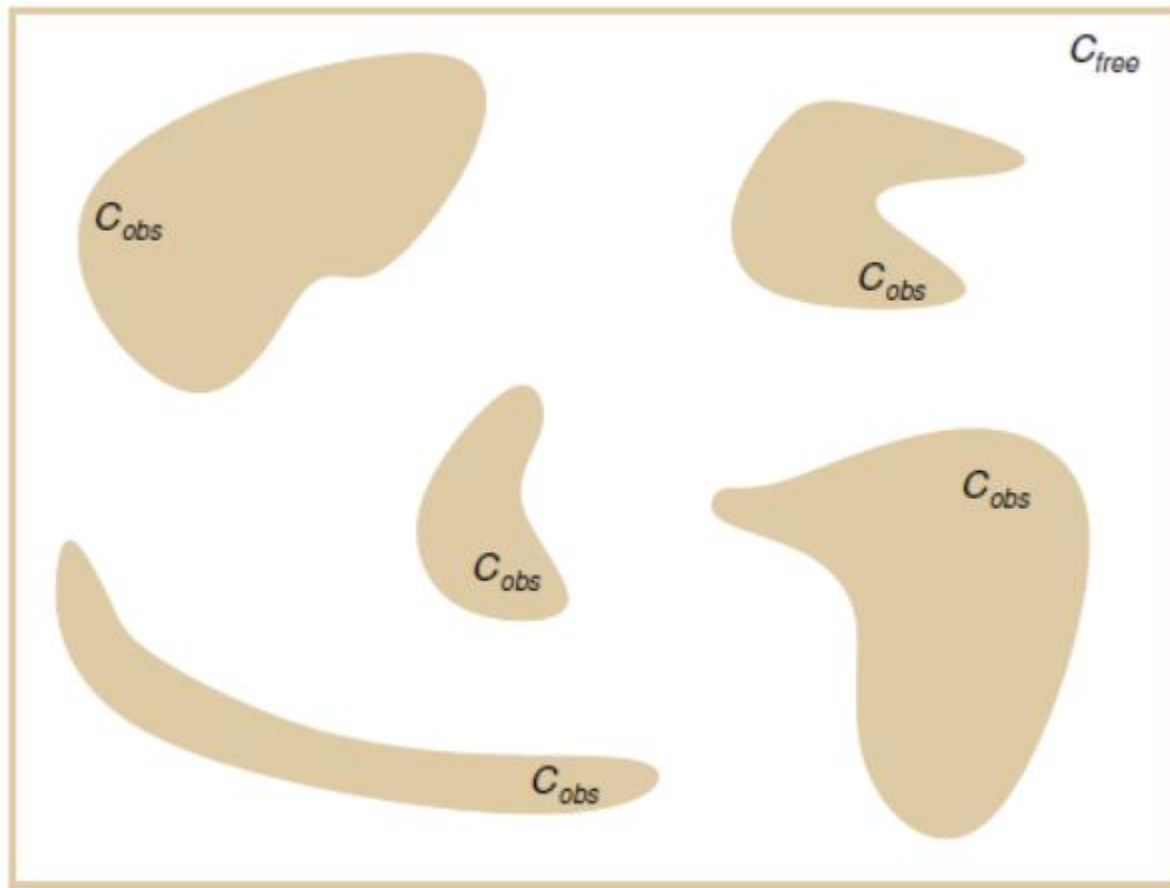
PRM Construction

- Query phase
 - Connect start and goal configuration with PRM using local planner
 - Use the graph to search the path



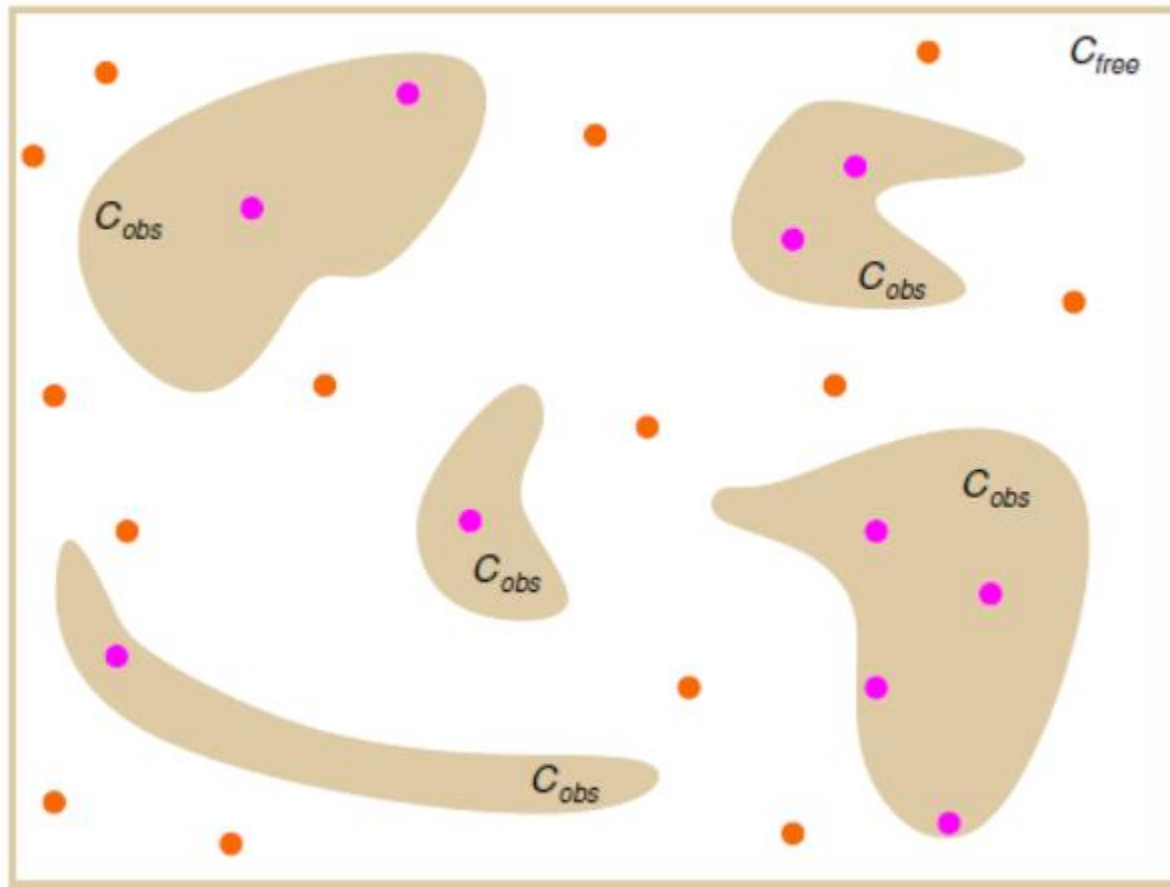
PRM Construction

- Given problem domain



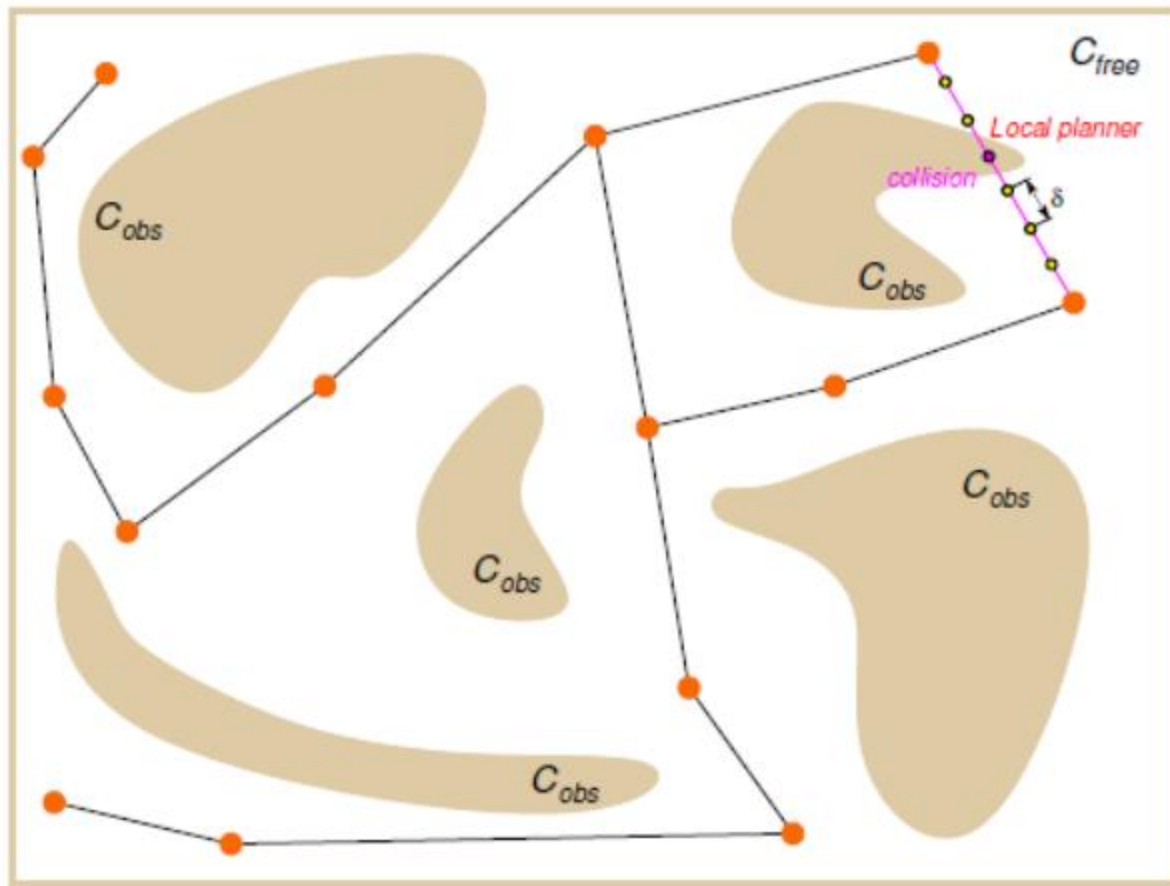
PRM Construction

- Generate random configurations



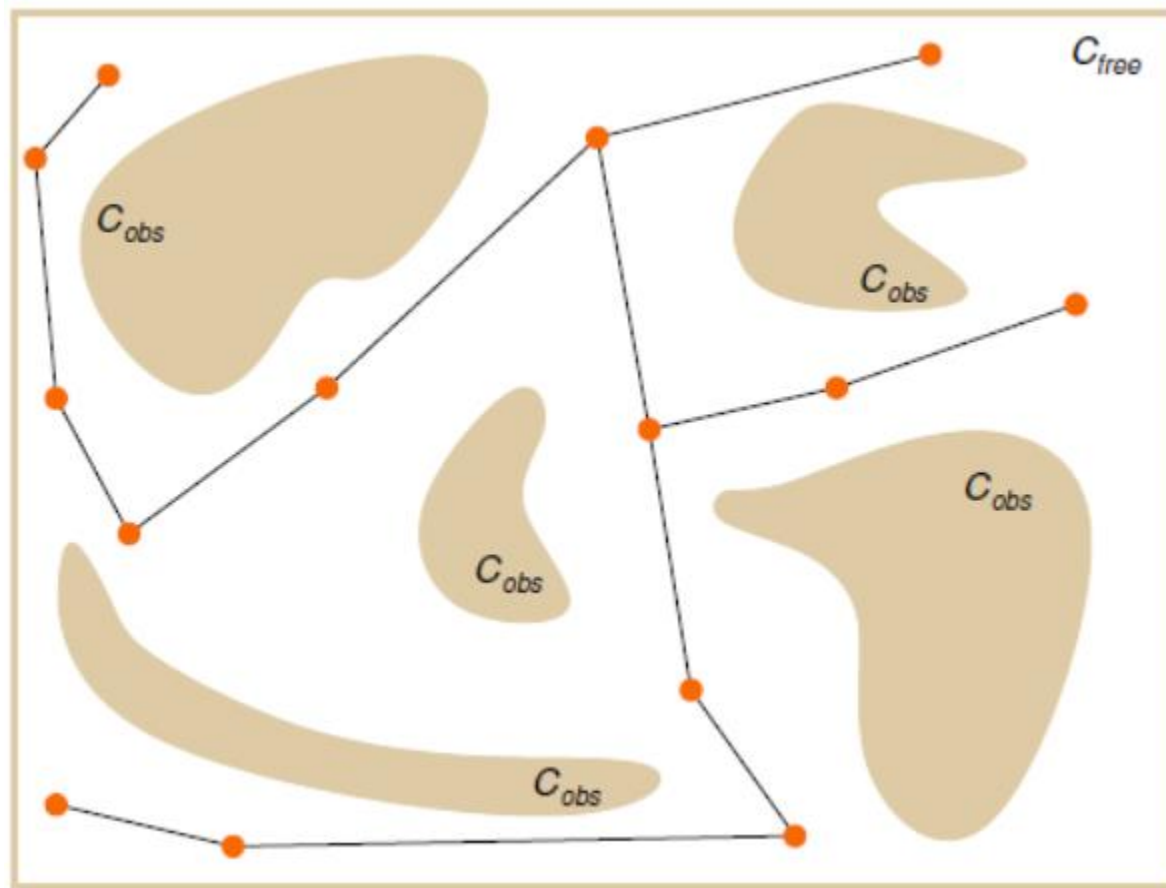
PRM Construction

- Connect random configuration samples



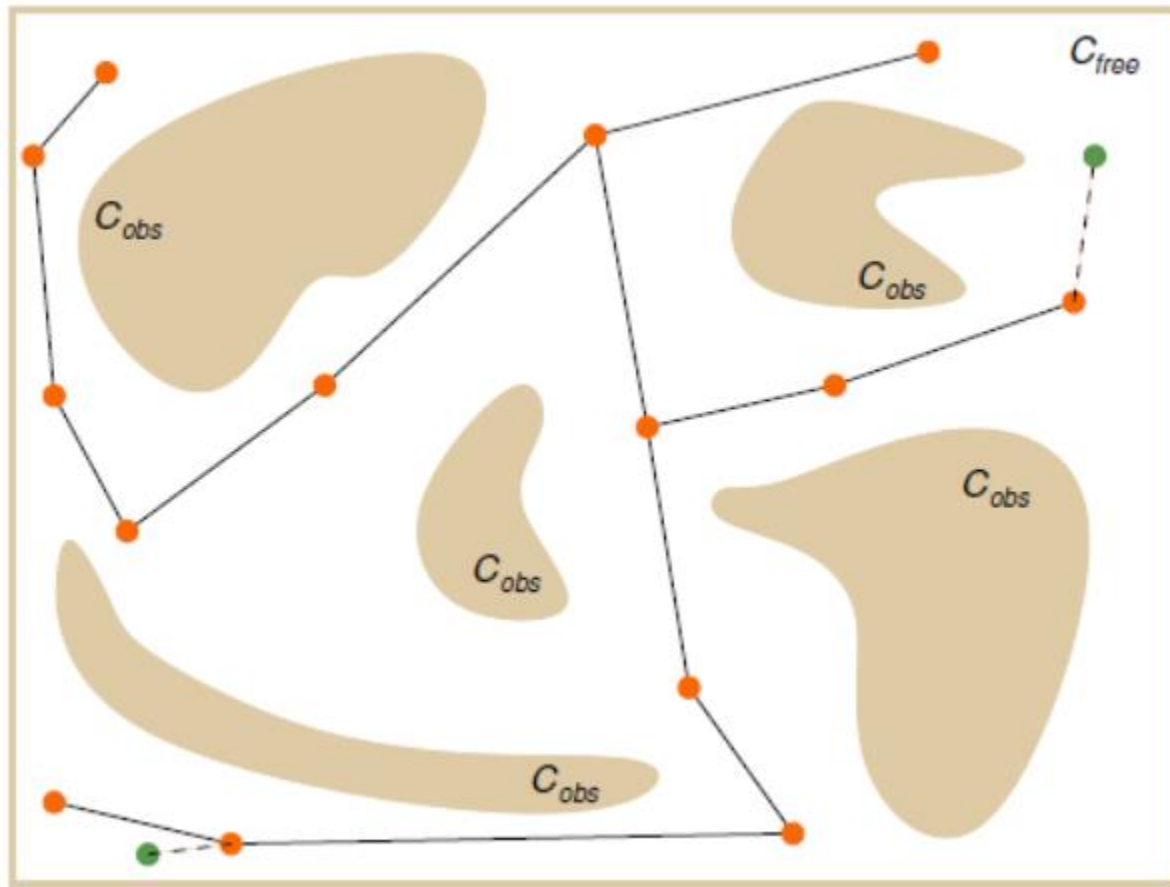
PRM Construction

- Get the connected roadmap (graph)



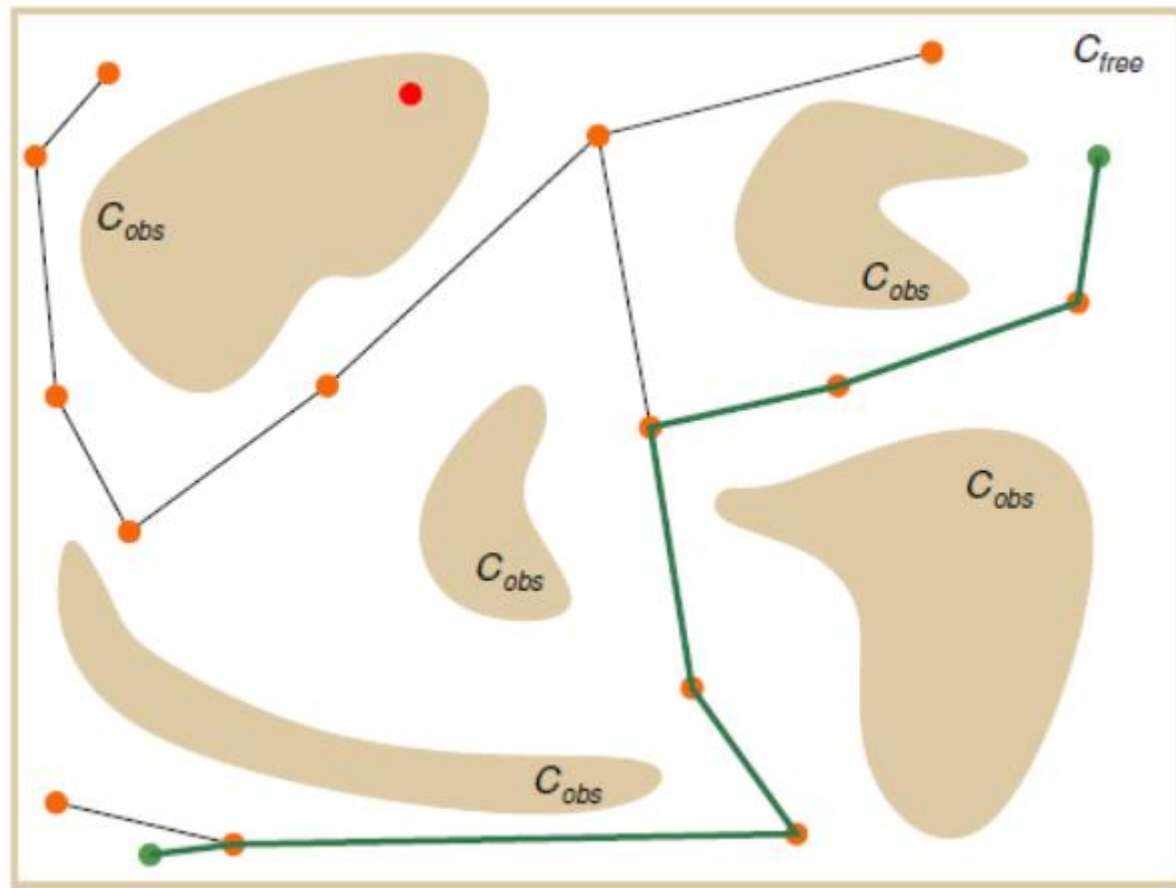
Query Configurations

- Add start and goal



Query Configurations

- Find the path



Random Sampling

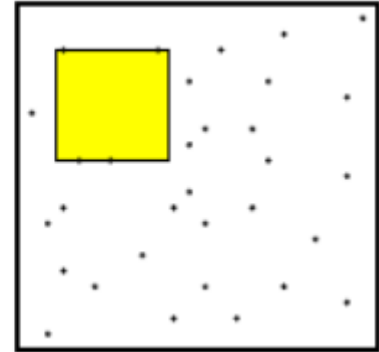
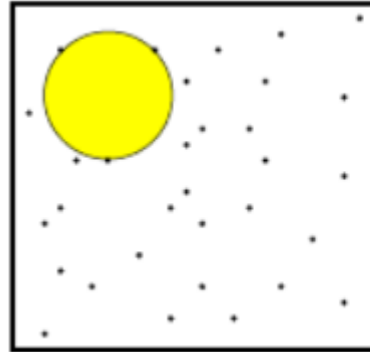
- A solution can be found using only a few number of samples (Oraculum)
- Sampling strategies matters

Random Sampling

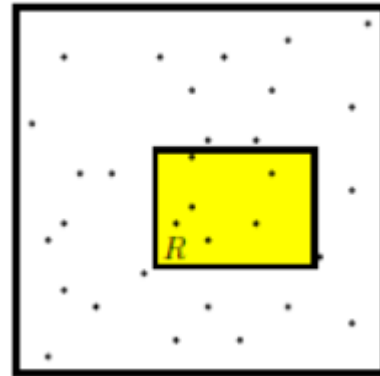
- A solution can be found using only a few number of samples (Oraculum)
- Sampling strategies matters
 - Near obstacles
 - Narrow passages
 - Grid-based
 - Uniform sampling (be carefull)

Random Sampling

- Dispersion



- Discrepancy



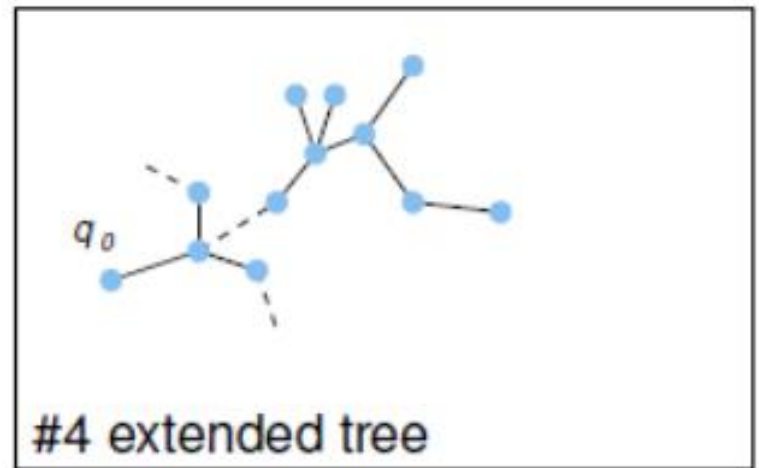
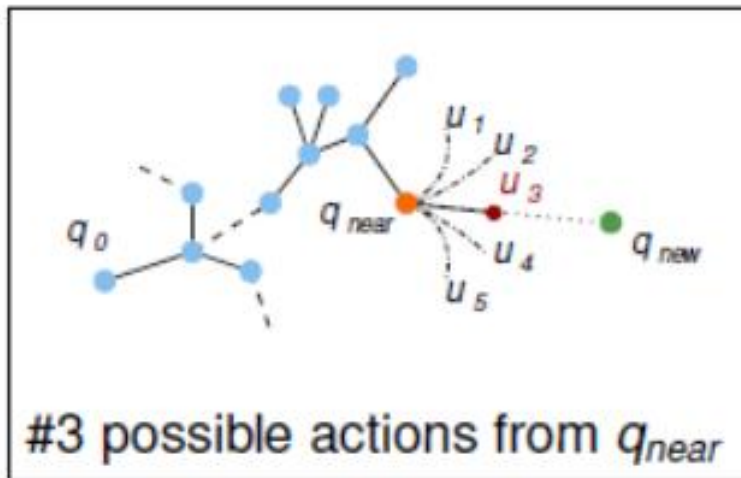
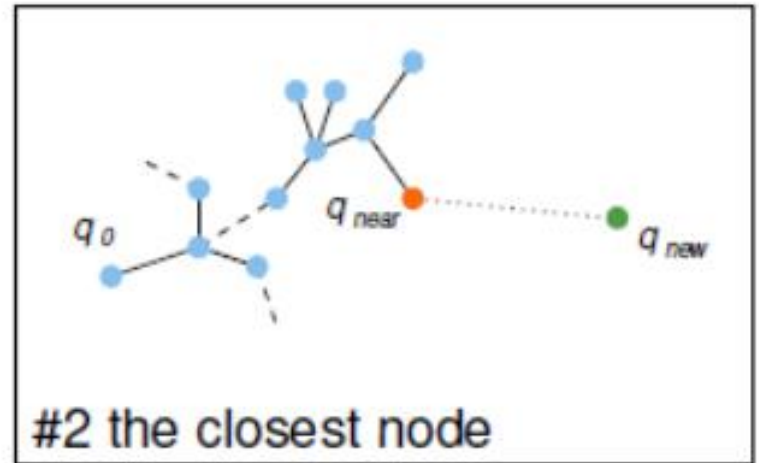
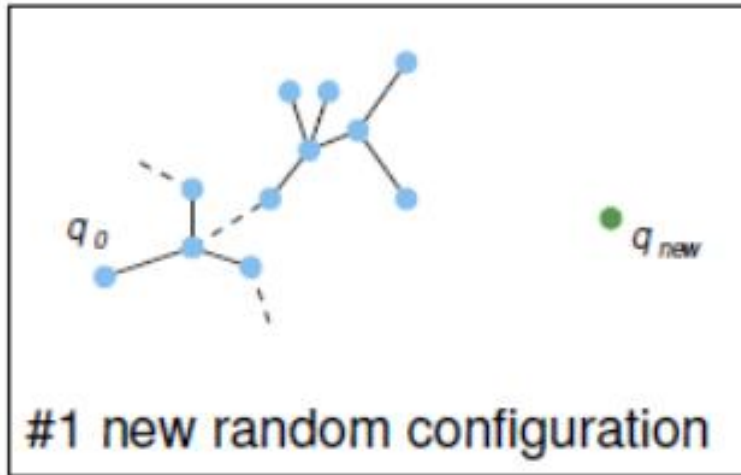
- Bounding box



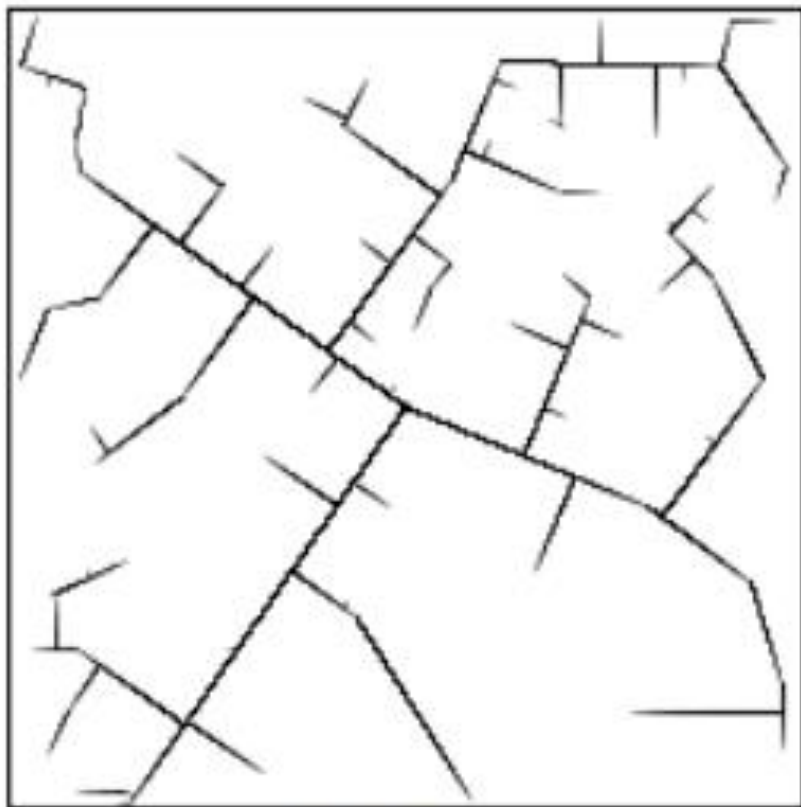
Rapidly-exploring Random Tree

- Incrementally builds a graph towards the goal area
 - Start with the initial configuration (root)
 - Generate a new random configuration q_{new}
 - Find the closest node q_{near} in the tree (KD-tree)
 - Extend q_{near} towards q_{new} (move close to position)
 - Repeat random generation until distance from goal is sufficient
 - Terminate after decided running time

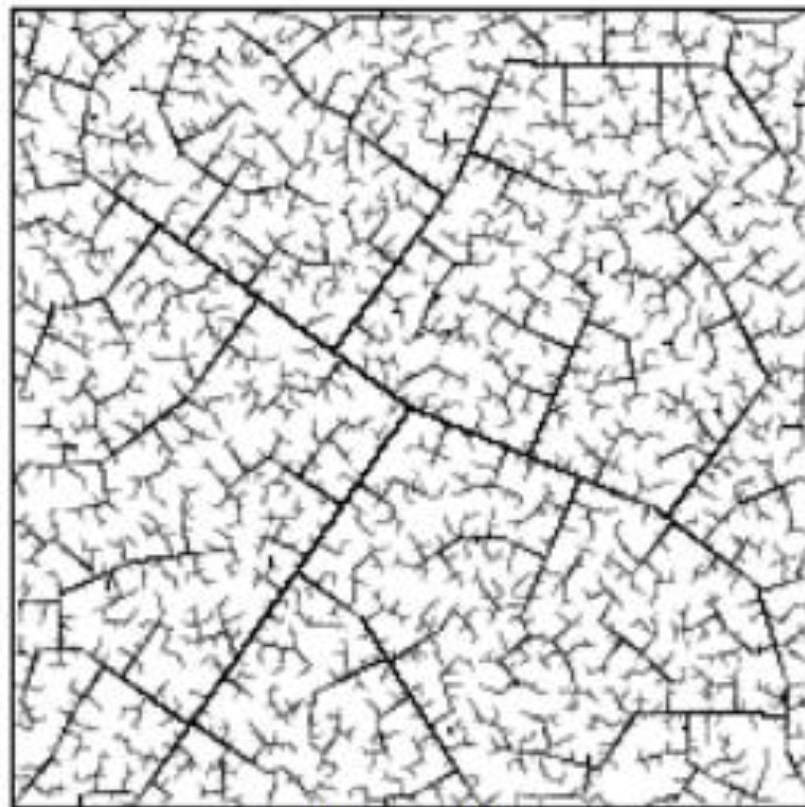
Rapidly-exploring Random Tree



Rapidly-exploring Random Tree



45 iterations

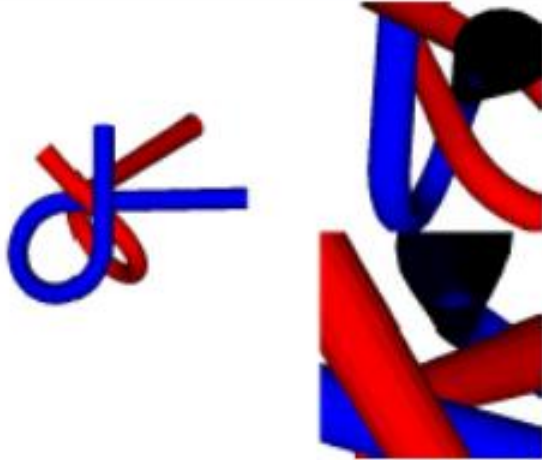


2345 iterations

Rapidly-exploring Random Tree

- Rapidly explores the space
- Allows kinematic and dynamic constraints
- Can provide trajectory as a sequence of control commands
- Collision detection test is usually used as “black-box”
- Poor performance in narrow passages
- Provides feasible (not optimal) paths
- Many variants

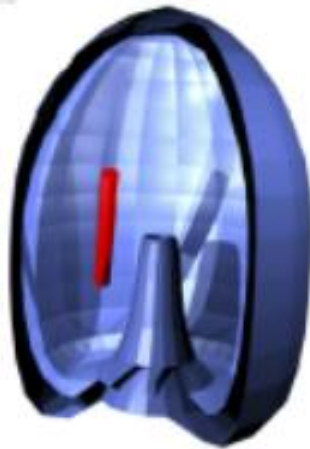
Rapidly-exploring Random Tree



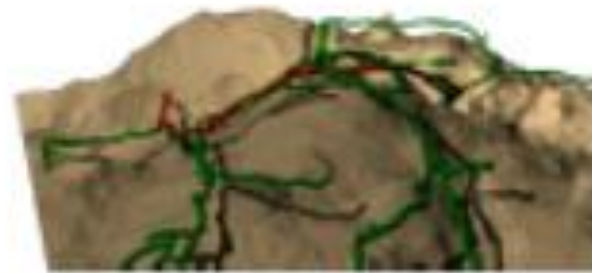
Alpha puzzle benchmark



Apply rotations to reach the goal



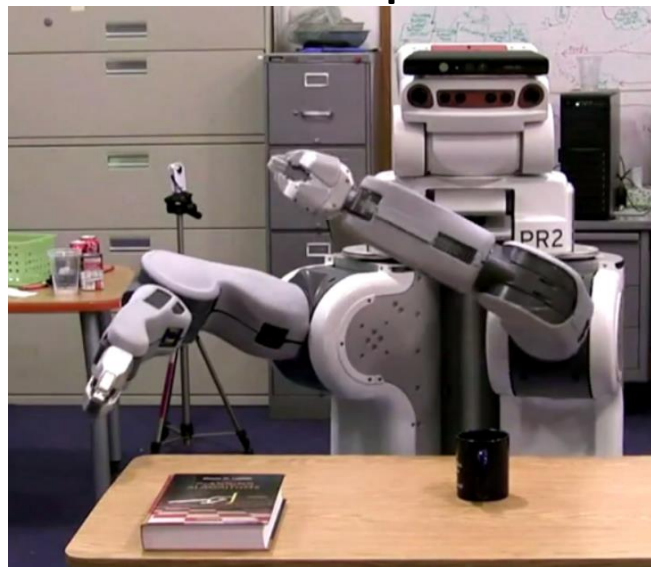
Bugtrap benchmark



Planning on a 3D surface

RRT-star

- PRM and RRT are theoretically probabilistic complete (feasible solution without quality guarantee)
- RRT-star: asymptotically optimal RRT
 - Based on guaranteed dispersion



Extension of Robotic Problem

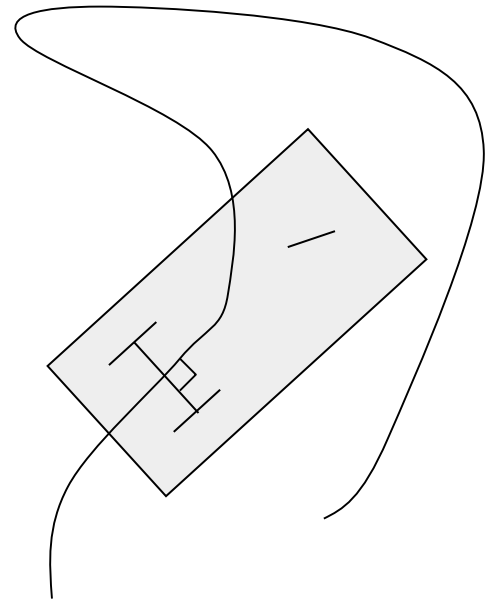
- More complex robots
 - Multiple robots
 - Movable objects, moving obstacles
 - Non-holonomic & dynamic constraints
 - Physical models and deformable objects
 - Sensorless motions (exploiting task mechanics)
 - Uncertainty in control and/or sensing
 - Optimal motion planning
 - Integration of planning and control

Extension of Robotic Problem

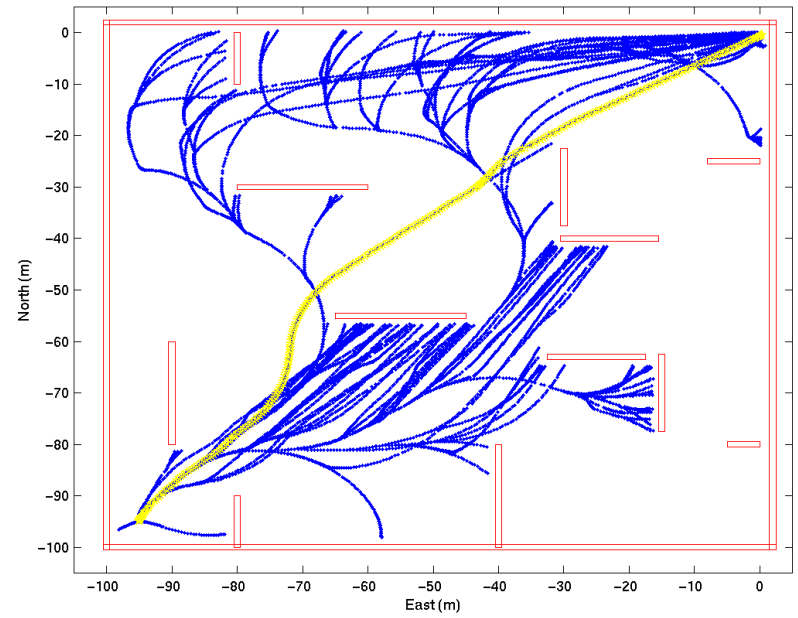
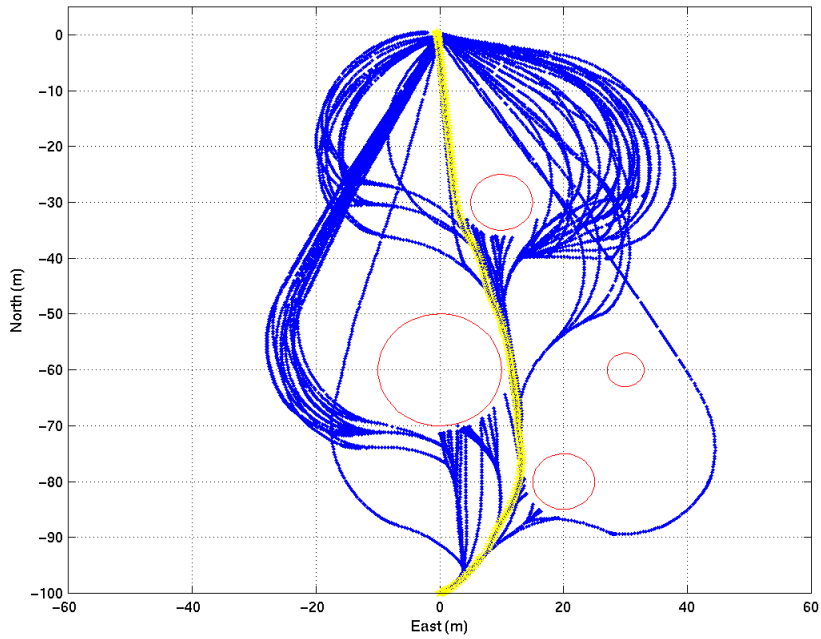
- More complex robots
 - Multiple robots
 - Movable objects, moving obstacles
 - **Non-holonomic** & dynamic constraints
 - Physical models and deformable objects
 - Sensorless motions (exploiting task mechanics)
 - Uncertainty in control and/or sensing
 - Optimal motion planning
 - Integration of planning and control

Integrating Dynamics

- Point robot trajectory vs. system dynamics
- Controlling problem
- Feasible trajectories reduce configuration space
- Trajectory primitives
 - Maneuvers
- Motion planning incl. system dynamics

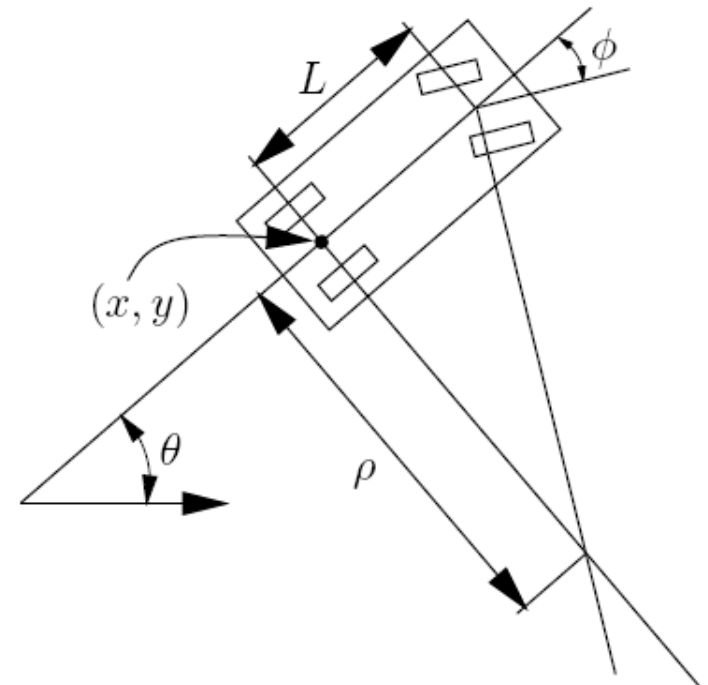


Maneuver Tree



Maneuvers

- **Car-like robot**
- Dubins curves (two templates, 6 seq.)
 - Optimal path for wheeled vehicles
 - Consist of three primitives
 - No reverse direction allowed
 - Constant speed
- Reeds-Shepp Curves (46)
 - Reverse direction allowed

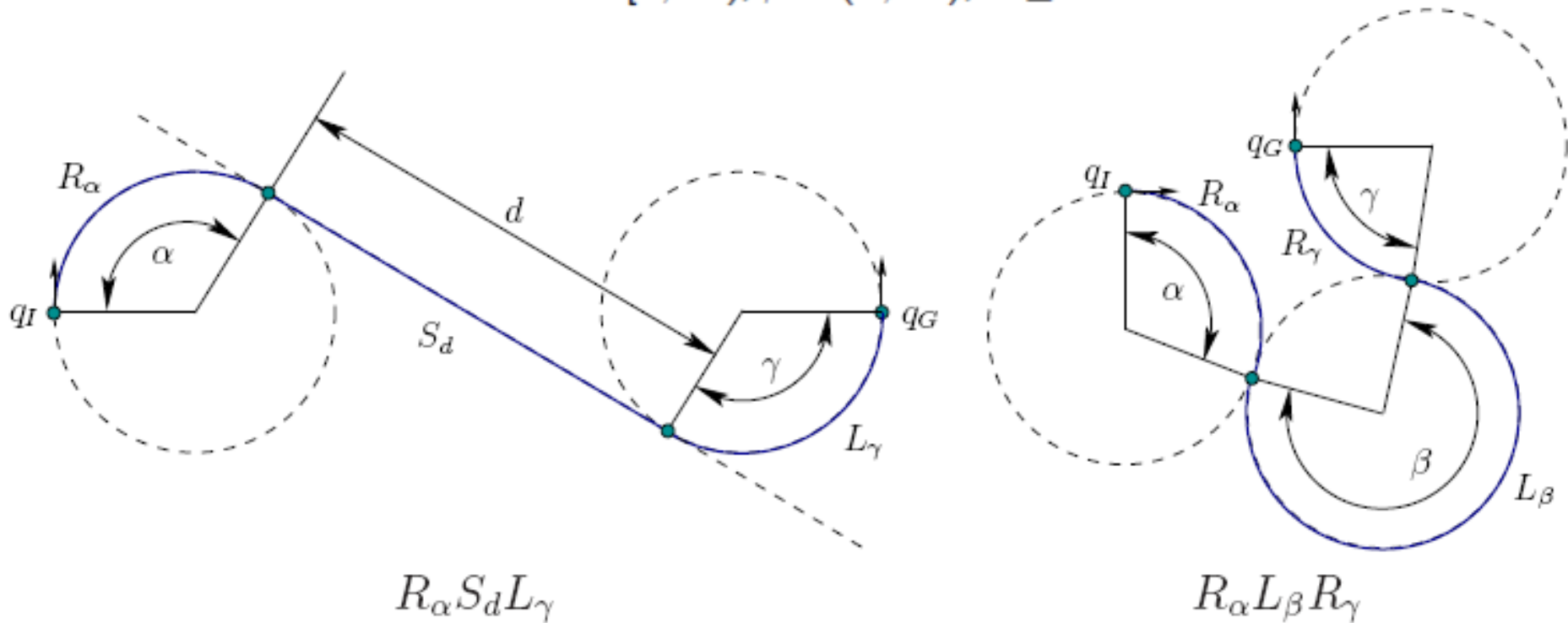


Dubins Curves

- Trajectory parameterization (tree expansion)

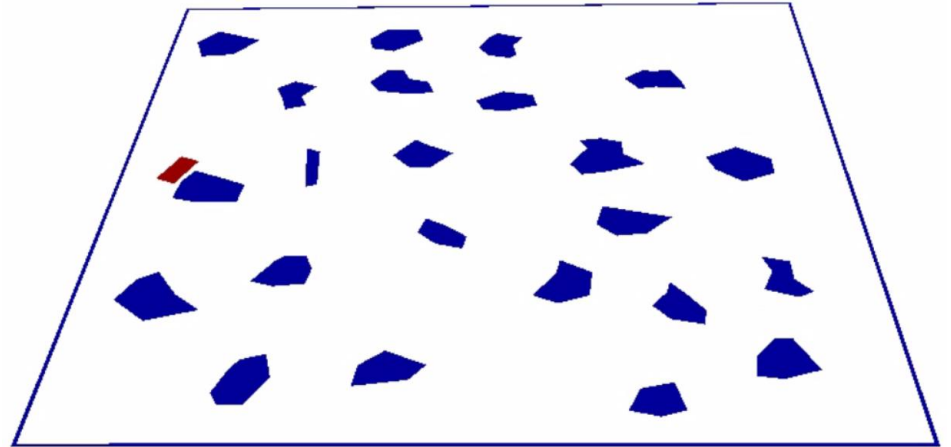
$$\{L_\alpha R_\beta L_\gamma, R_\alpha L_\beta R_\gamma, L_\alpha S_d L_\gamma, L_\alpha S_d R_\gamma, R_\alpha S_d L_\gamma, R_\alpha S_d R_\gamma\}$$

$$\text{for } \alpha \in [0, 2\pi), \beta \in (\pi, 2\pi), d \geq 0$$

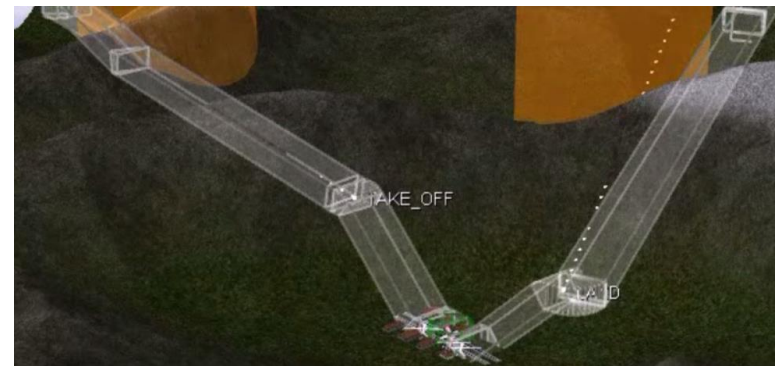


Maneuvers Examples

- Car-like example



- Flight example
 - 3D space
 - Climb/descend maneuvers
 - Adjustable speed



Adaptive Path Planner

- Cooperative surveillance mission context
 - Roadmap for a high-level planning
 - Low-level motion planning using Reeds-Shepp curves
 - Replanning in the case of collision

