Motion Planning: Overview and Foundations

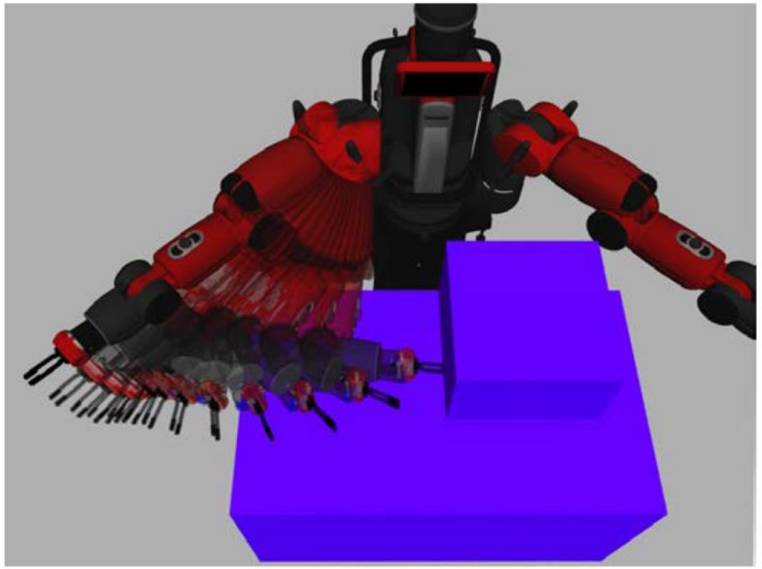
CS 6301 Special Topics: Introduction to Robot Manipulation and Navigation

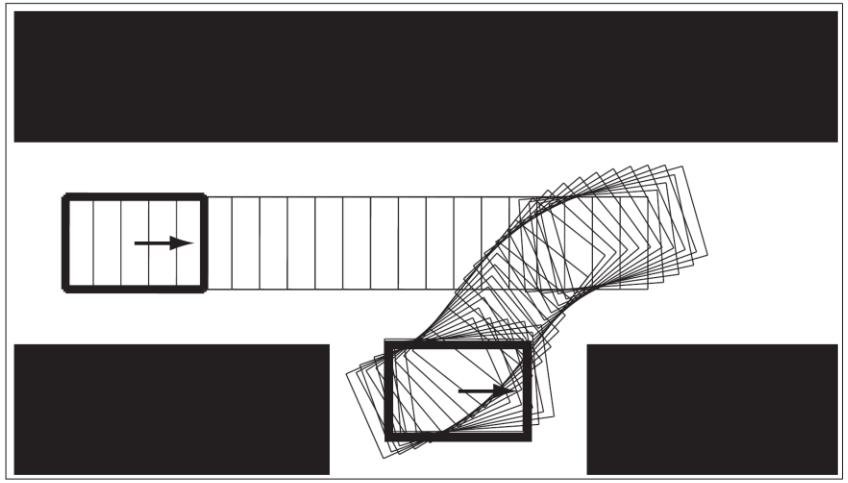
Professor Yu Xiang

The University of Texas at Dallas

Motion Planning

- Motion planning: finding a robot motion from a start state to a goal state (A to B)
  - Avoids obstacles
  - Satisfies other constraints such as joint limits or torque limits





Configuration Space

- The configuration of a robot arm with n joints
  - n joint positions $q = (\theta_1, \ldots, \theta_n)$
- Free C-space $\mathcal{C}_{\text{free}}$
  - Configurations where the robot neither penetrates an obstacle nor violated a joint limit

Robot State

- For second order dynamics, state is configuration and velocity

State $x = (q, v) \in \mathcal{X}$ $v = \dot{q}$

Control input $u \in \mathcal{U} \subset \mathbb{R}^m$ Force (acceleration)

- For first order dynamics, state is the configuration

State $q(x)$

Control input: velocity $\mathcal{X}_{\text{free}} = \{x \mid q(x) \in \mathcal{C}_{\text{free}}\}$

Equations of Motion

- The equations of motion of a robot

$$\dot{x} = f(x, u)$$

Forward dynamics

Robot state

Control inputs $u \in \mathcal{U} \subset \mathbb{R}^m$

- Integral form

$$x(T) = x(0) + \int_0^T f(x(t), u(t))dt$$

Motion Planning

- Given an initial state $x(0) = x_{\text{start}}$ and a desired final state x_{goal} find a time T and a set of control $u : [0, T] \to \mathcal{U}$ such that the motion

$$x(T) = x(0) + \int_0^T f(x(t), u(t))dt$$

satisfies

$$x(T) = x_{\text{goal}}$$

$$q(x(t)) \in \mathcal{C}_{\text{free}} \ \textit{for all} \ t \in [0, T]$$

Types of Motion Planning Algorithms

- Path planning vs. motion planning
  - Path planning is a purely geometric problem of finding a collision-free path

    $$q(s), s \in [0, 1] \quad q(0) = q_{\text{start}} \quad q(1) = q_{\text{goal}}$$
  - No concern about dynamics/control inputs
- Control inputs: $m = n$ **versus** $m < n$
  - When m < n, the robot cannot follow many paths
  - E.g., a car, n = 3 (the position and orientation of the chassis in the plane) m = 2 (forward-backward motion and steering)
- Online vs. Offline
  - Online is needed when the environment is dynamic

Types of Motion Planning Algorithms

- Optimal vs. satisficing
  - In addition to reaching the goal state, we might want the motion planner to minimize a cost

$$J = \int_0^T L(x(t), u(t))dt$$

Time-optimal L=1

Minimum-effort $L = u^{\mathrm{T}}(t)u(t)$

- Exact vs. approximate
  - Approximate $\|x(T) - x_{\text{goal}}\| < \epsilon$

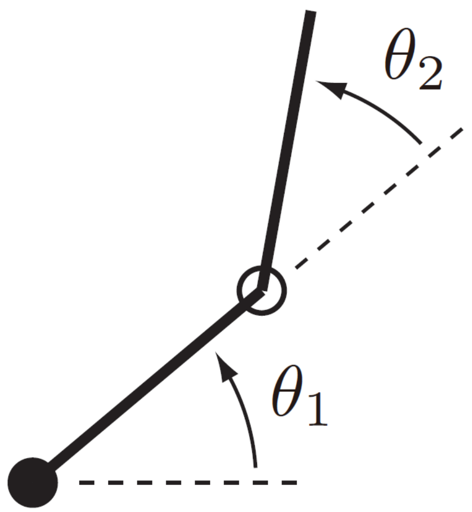
- With or without obstacles
  - Some motion planning problems are challenging even without obstacles
  - When m< n or optimality is desired

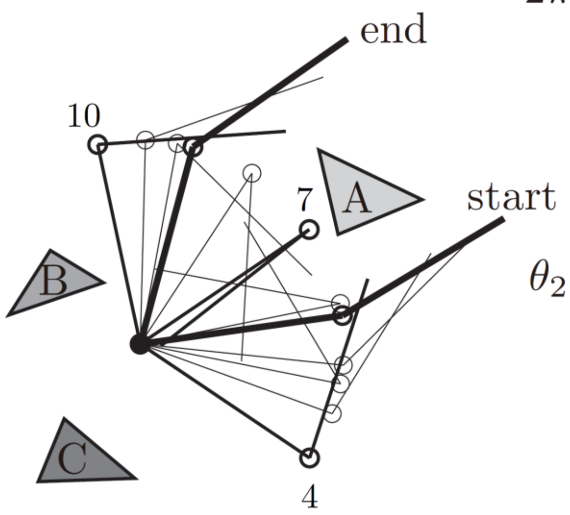
Properties of Motion Planners

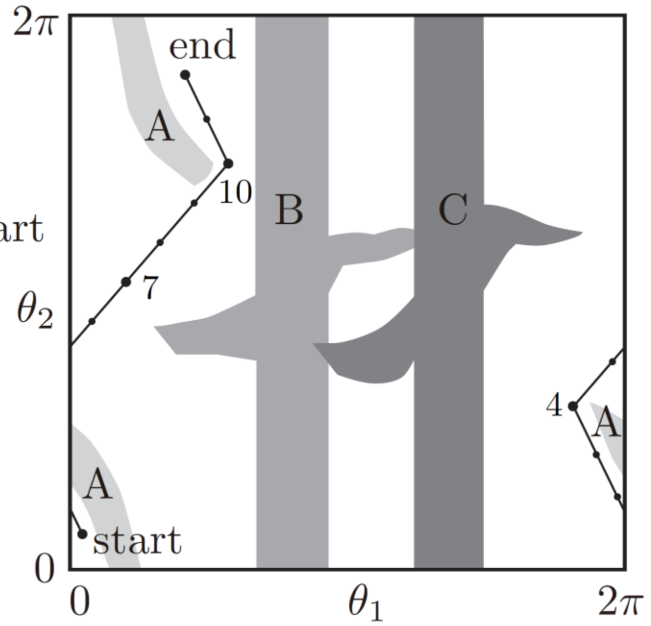
- Multiple-query vs. single-query planning
  - Multiple-query can build a data structure for $\mathcal{C}_{\text{free}}$
- "Anytime" planning
  - Continues to look for a better solution after a first solution is found
  - The planner can be stopped at anytime
- Completeness
  - A motion planner is said to be complete if it is guaranteed to find a solution in finite time if one exists, and to report failure if there is no feasible motion plan
- Computational complexity
  - The amount of time the planner takes to run or the amount of memory it requires

Configuration Space Obstacles

- Workspace obstacles partition the configuration space into two sets
  - Free space and obstacle space $\mathcal{C} = \mathcal{C}_{\text{free}} \cup \mathcal{C}_{\text{obs}}$
  - Joint limits are treated as obstacle in the configuration space
- A 2R planar arm



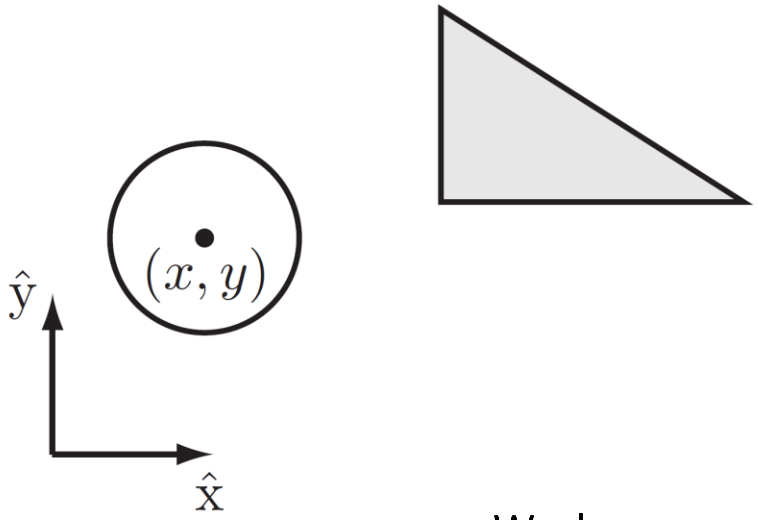




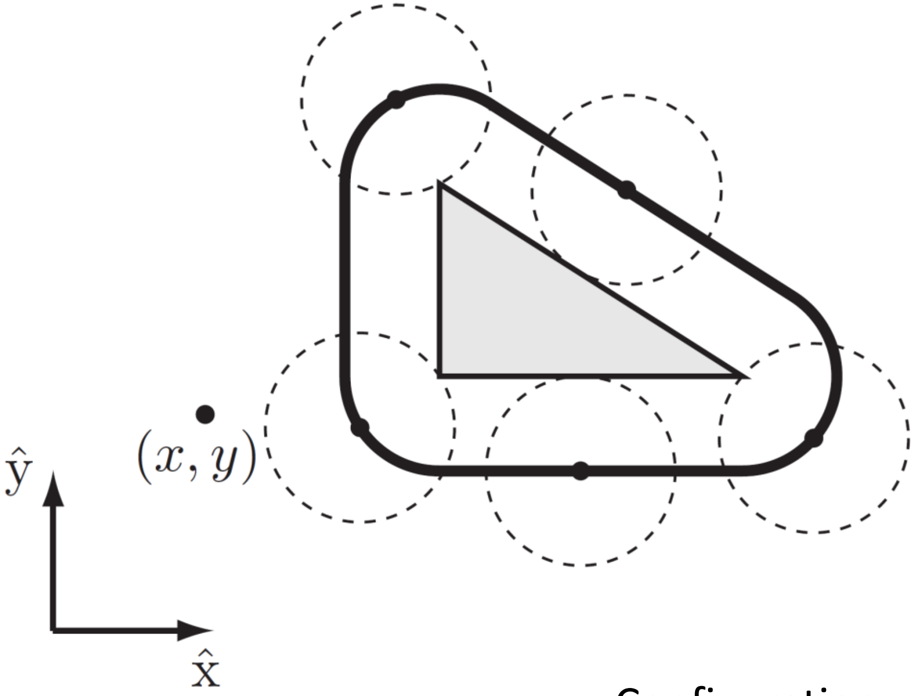
Configuration space

Configuration Space Obstacles

- A circular planar mobile robot



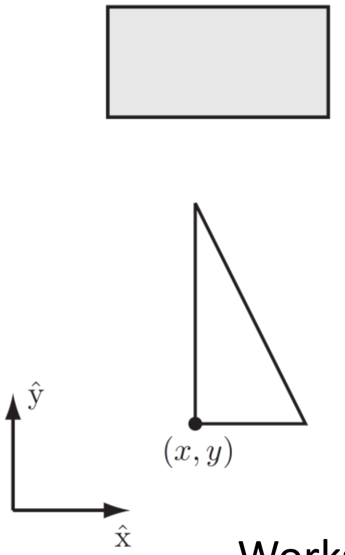
(a) Workspace space



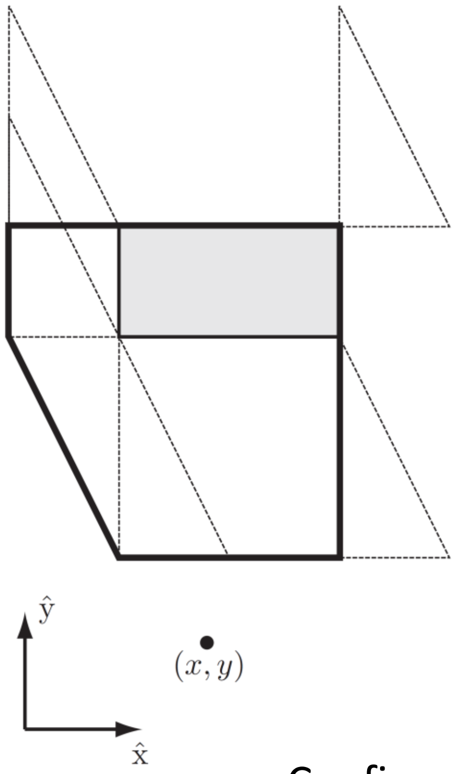
(b) Configuration space

Configuration Space Obstacles

- A Polygonal Planar Mobile Robot That Translates



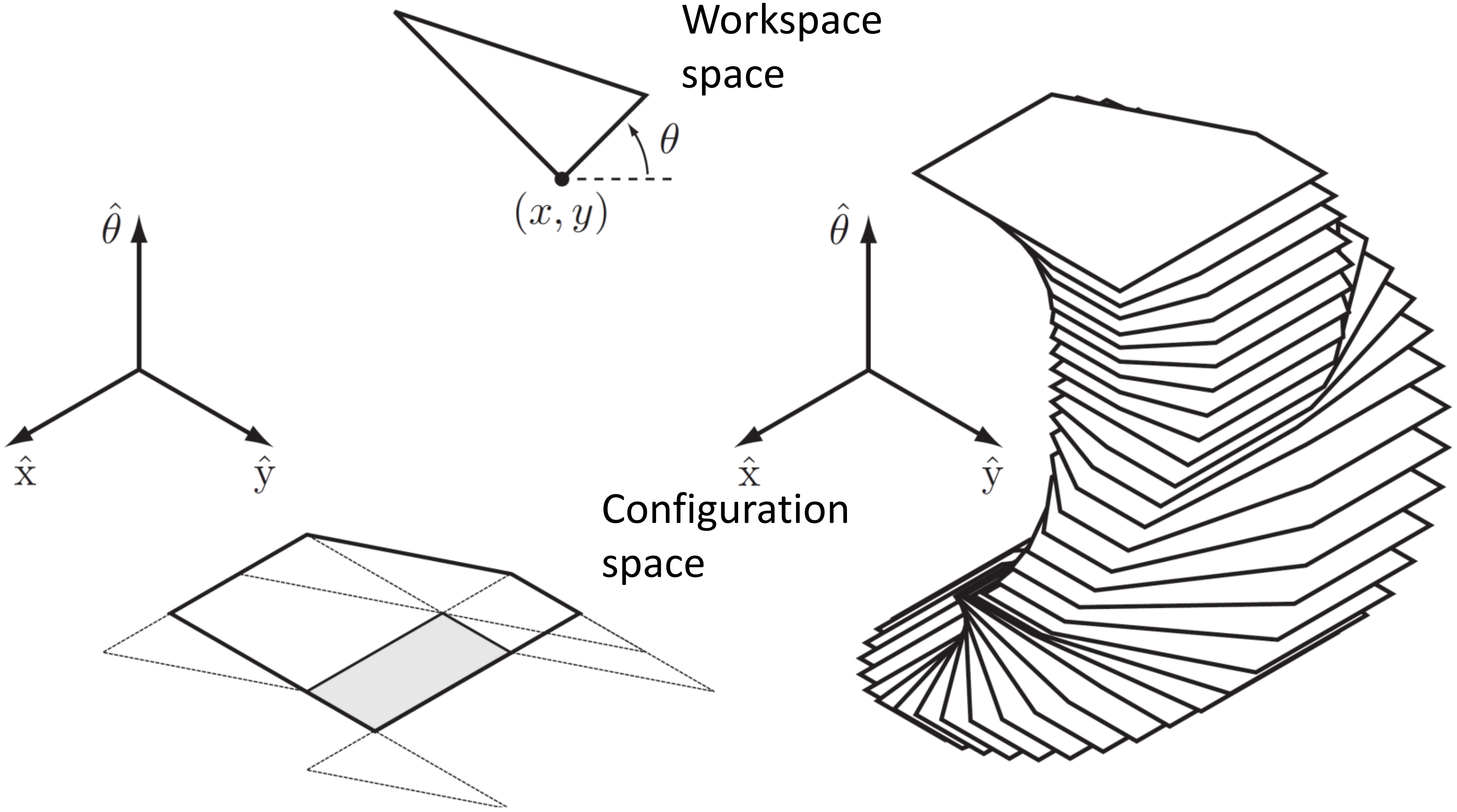
(a) Workspace space



(b) Configuration space

Configuration Space Obstacles

- A Polygonal Planar Mobile Robot That Translates and Rotates



Distance to Obstacles

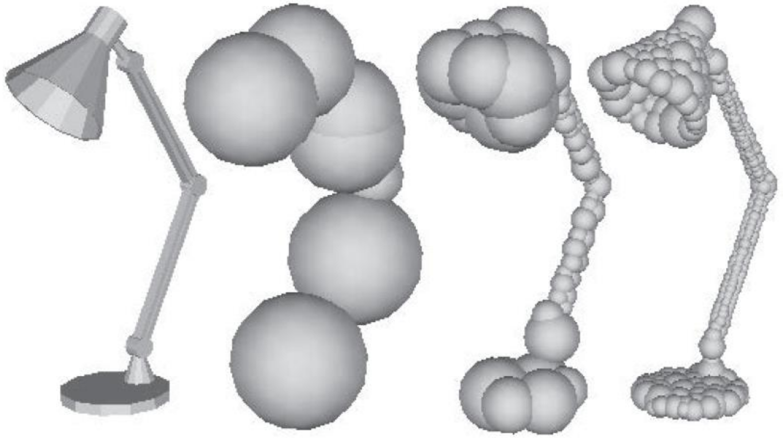
- Given a C-obstacle $\mathcal{B}$ and a configuration q, the distance between a robot and the obstacle

$$\begin{aligned} d(q, \mathcal{B}) &> 0 && \text{(no contact with the obstacle)}, \\ d(q, \mathcal{B}) &= 0 && \text{(contact)}, \\ d(q, \mathcal{B}) &< 0 && \text{(penetration)}. \end{aligned}$$

- A distance measurement algorithm determines $d(q, \mathcal{B})$
- A collision detection algorithm determines whether $d(q, \mathcal{B}_i) \leq 0$

Distance to Obstacles

- Approximation of 3D shapes using 3D spheres

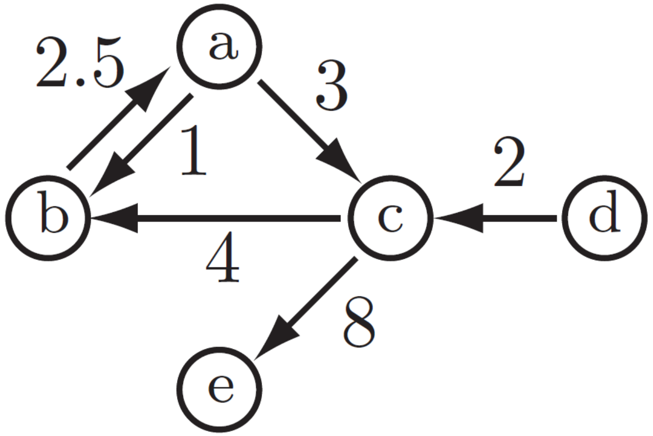


- Robot: k spheres of radius R_i centered at $r_i(q)$
- Obstacle: l spheres of radius B_j centered at b_j
- The distance between the robot and the obstacle

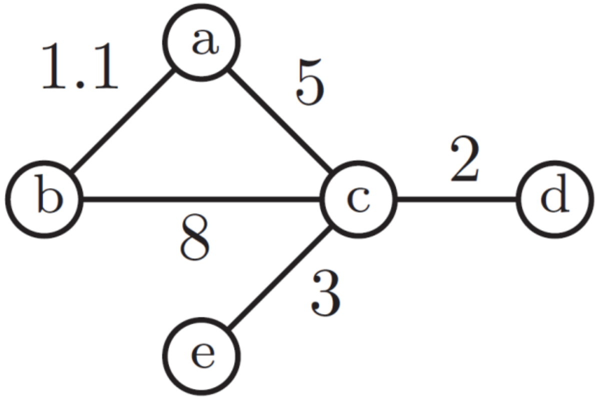
$$d(q, \mathcal{B}) = \min_{i,j} \|r_i(q) - b_j\| - R_i - B_j$$

Graphs for Motion Planning

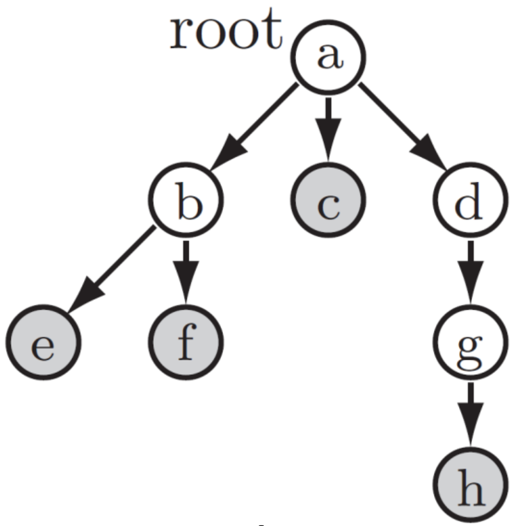
- Node: a configuration or a state
- Edge: the ability to move between nodes without penetrating an obstacle or violating other constraints



A weighted digraph



A weighted undirected graph



A tree

Summary

- Overview of motion planning
- Configuration space obstacle
- Distance to obstacles
- Graphs for motion planning

Further Reading

- Chapter 10 in Kevin M. Lynch and Frank C. Park. Modern Robotics: Mechanics, Planning, and Control. 1st Edition, 2017.