Introduction to Robotics
Lecture: Motion Planning

ECE 470/AE 482/ME 445

Motion Planning

- **Motion planning** is the problem of finding a robot motion from a start state to a goal state that avoids obstacles in the environment and satisfies other constraints, such as joint limits or torque limits.
- Recall the configuration space (C-space): every point in the C-space $\mathcal{C} \subset \mathbb{R}^n$ corresponds to a unique configuration q of the robot
  - Example: configuration of a robot arm is $q = (\theta_1, \dots, \theta_n)$
- The free C-space $\mathcal{C}_{\text{free}}$ consists of the configurations where the robot neither collides with obstacles nor violates constraints

Equations of Motion for Motion Planning

- Control inputs (m-vector)
    - $u \in U \subset \mathbb{R}^m$
- States
    - The **state** of the robot is defined by its configuration and velocity
    - $x = (q, v) \in X$ (for $q \in \mathbb{R}^n$, typically $v = \dot{q}$)

- $X_{free} = \{x | q(x) \ \in \mathcal{C}_{\text{free}}\}$, where $q(x)$ is the configuration q corresponding to the state x
- The equation of motion of the robot

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

or

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

Motion Planning

- Given an initial state $x(0) = x_{start}$ and a desired final state x_{goal}, find a time T and a set of controls u: $[0, T] \rightarrow U$ such that the motion satisfies $x(T) = x_{goal}$ and $q(x(t)) \in \mathcal{C}_{\text{free}}$ for all $t \in [0, T]$

- Assumptions:
    1. A feedback controller can ensure that the planned motion is followed closely
    2. An accurate model of the robot and environment will evaluate $\mathcal{C}_{\text{free}}$ during motion planning

Types of Motion Planning Problems

- Path planning versus motion planning
  - Trajectory generation versus these lectures
- Control inputs: $m = n$ versus $m < n$
- Online versus offline
  - How reactive does your planner need to be?
- Optimal versus satisficing
  - Minimum cost or just reach goal?
- Exact versus approximate
  - What is sufficiently close to goal?
- With or without obstacles
  - How challenging is the problem?

Properties of Motion Planners

- Multiple-query versus single-query planning
- "Anytime" planning
  - Continues to look for better solutions after first solution is found
- Completeness
  - A planner is complete if it is guaranteed to find a solution in finite time if one exists, and report failure if no feasible plan exists
  - A planner is resolution complete if it is guaranteed to find a solution, if one exists, at the resolution of a discretized representation
  - A planner is probabilistically complete if the probability of finding a solution, if one exists, tends to 1 as planning time goes to infinity
- Computational complexity
  - Characterization of the amount of time a planner takes to run or the amount of memory it requires

Motion Planning Methods

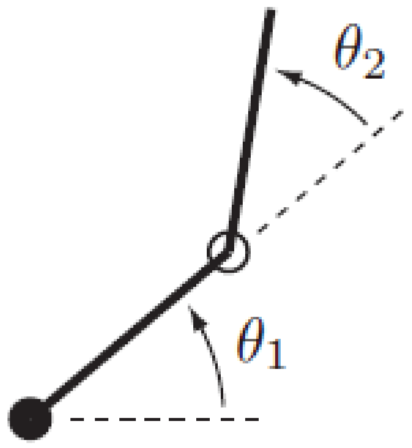
- Complete methods
  - exact representations of the geometry of the problem and space
- Grid methods
  - discretize $\mathcal{C}_{\mathrm{free}}$ and search the grid from q_{start} to goal
- Sampling methods
  - randomly sample from the C-space, evaluate if the sample is in X_{free}, and add new sample to previous samples
- Virtual potential fields
  - create forces on the robot that pull it toward goal and away from obstacles
- Nonlinear optimization
  - minimize some cost subject to constraints on the controls, obstacles, and goal
- Smoothing
  - given some guess or motion planning output, improve the smoothness while avoiding collisions

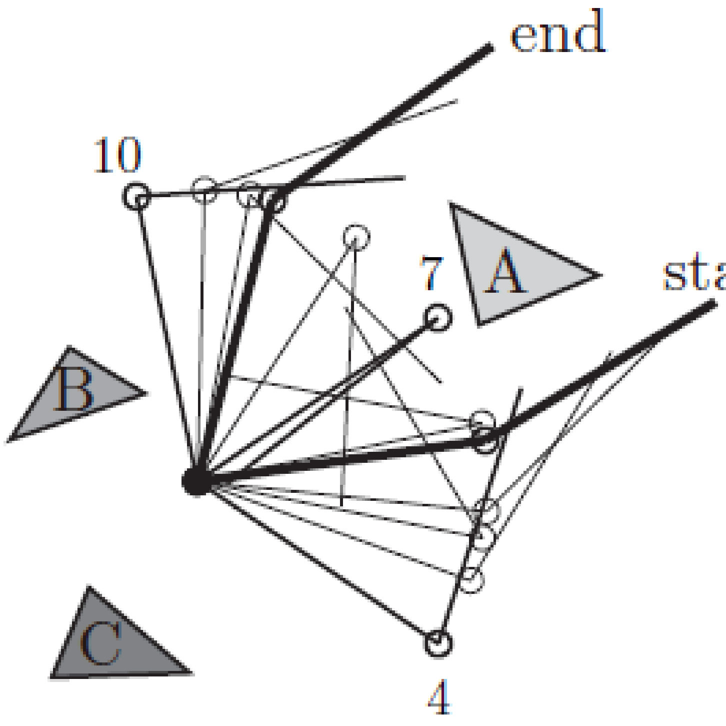
Configuration Space Obstacles

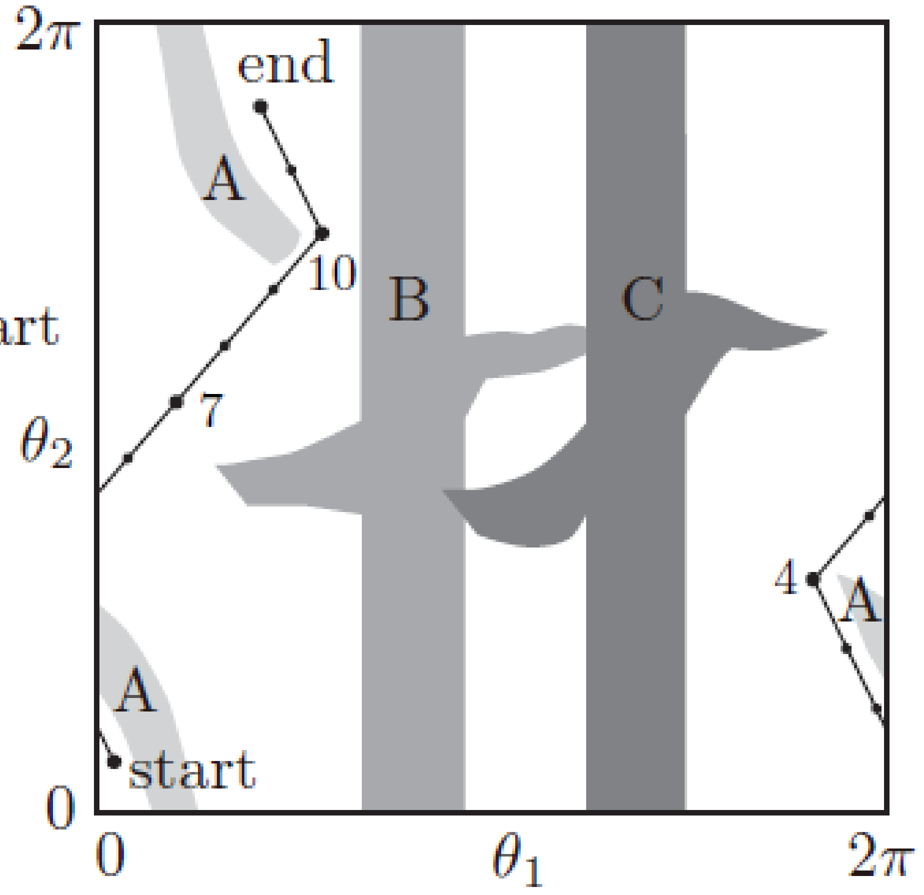
- We want to partition our C-space into two sets
  - The free space $\mathcal{C}_{\mathrm{free}}$
  - The obstacle space $\mathcal{C}_{\mathrm{obs}}$
  - Where $\mathcal{C} = \mathcal{C}_{\mathrm{free}} \cup \mathcal{C}_{\mathrm{obs}}$

- If obstacles break $\mathcal{C}_{\mathrm{free}}$ into separate components and q_{start} and q_{goal} are not in the same connected components, then there is no collision-free path

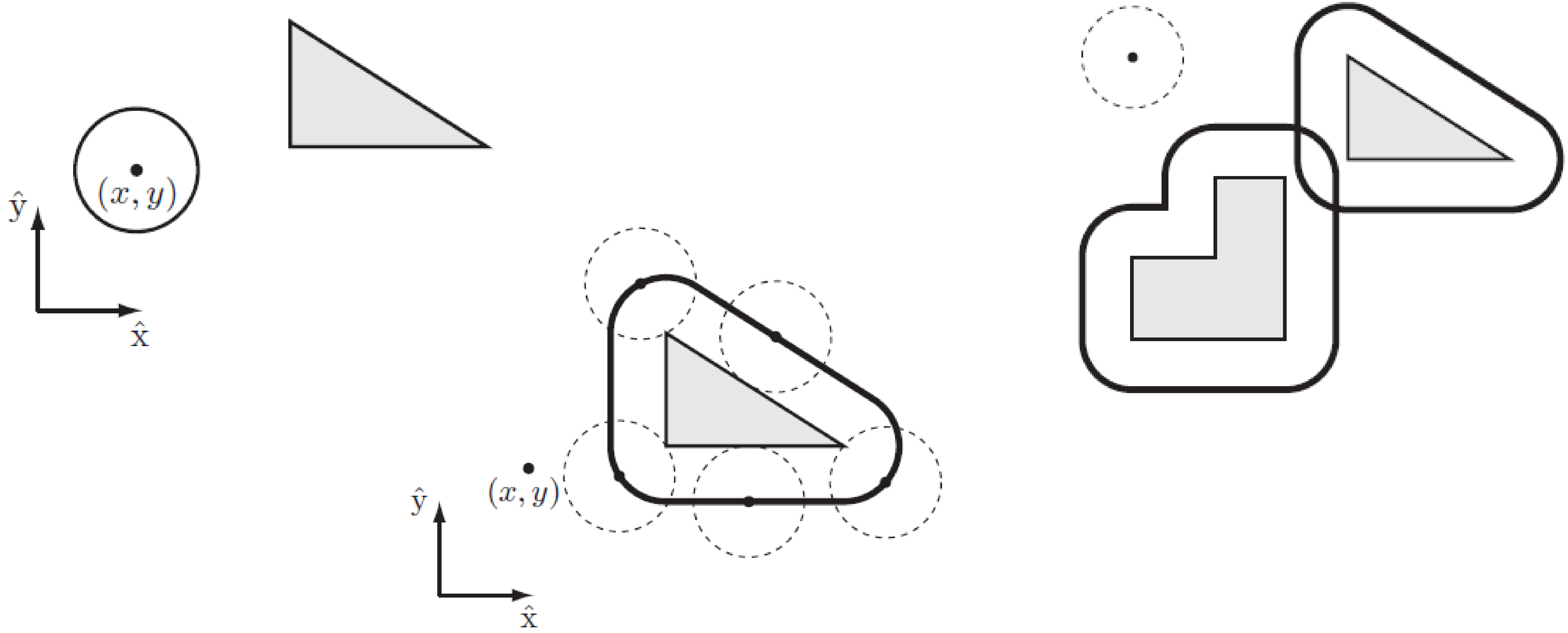
Configuration Space: 2R Planar Arm



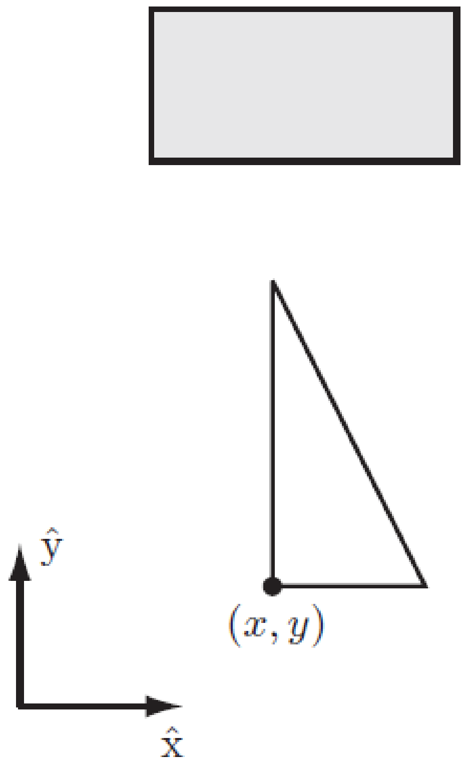


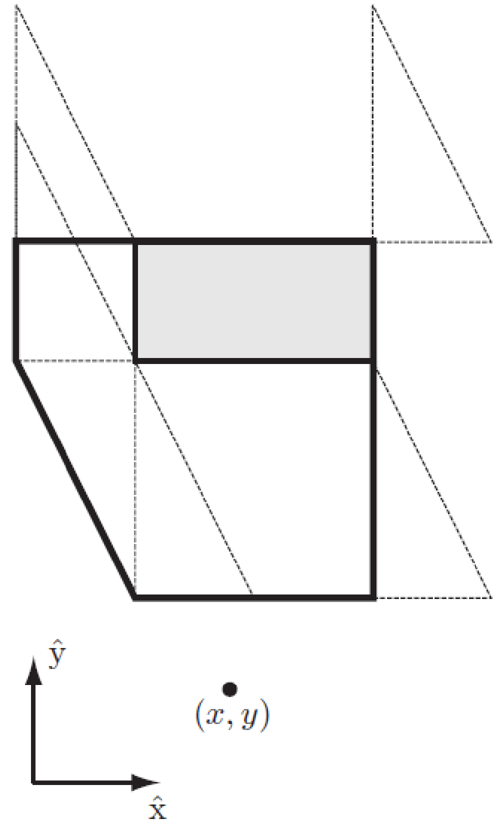


Configuration Space: Circular Mobile Bot

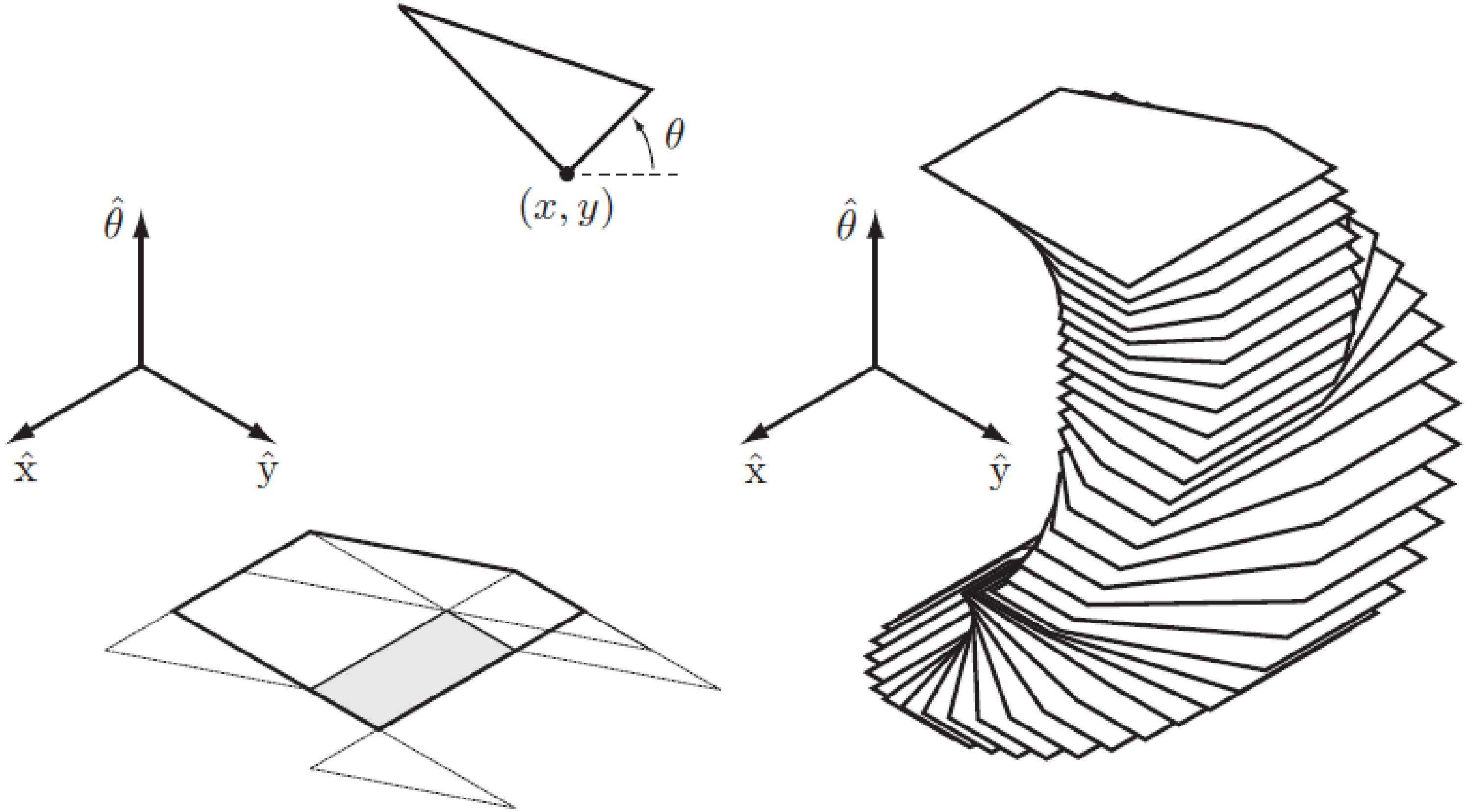


Polygonal Planar robot that translates and rotates





Polygonal Planar robot that translates and rotates



Collision Detection

- Given a C-obstacle B and a configuration q,

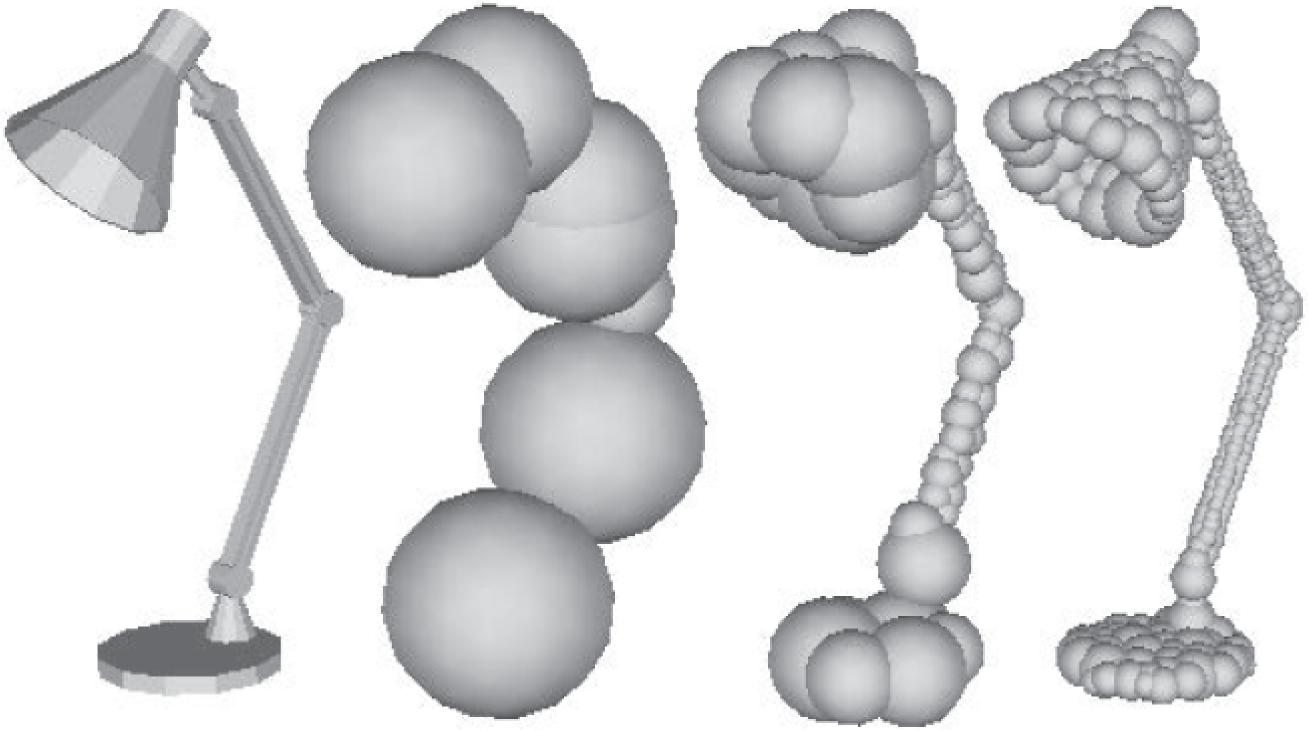
$$d(q, B) = \text{distance between the robot and the obstacle}$$

  - $d(q, B) > 0$ => no contact with the obstacle
  - $d(q, B) = 0$ => contact
  - $d(q, B) < 0$ => penetration
- Distance-measurement algorithm
  - Determines $d(q, B)$
- Collision-detection routine
  - Determines whether $d(q, B_i) \leq 0$ for any C-obstacle B_i

Spherical Approximations

- One simple method is to approximate the robot and obstacles as unions of overlapping spheres

- Approximations must be conservative



Distance Measures

- Given a robot at q represented by k spheres of radius R_i centered at $r_i(q)$, $i = 1, \ldots, k$, and an obstacle B represented by l spheres of radius B_j centered at b_j , $j = 1, \ldots, l$, the distance between the robot and the obstacle can be calculated as $d(q, B)$

$$d(q, B) = \min \left\| r_i(q) - b_j \right\| - R_i - B_j$$

Summary

- Defined and discussed concepts relating to **motion planning**
- Overviewed motion planning types, properties, and methods
- Discussed configuration space components $\mathcal{C}_{\mathrm{obs}}$and $\mathcal{C}_{\mathrm{free}}$
- Gave example distance measurement approaches to check collision detection