EE211: Robotic Perception and Intelligence

Lecture 3 Motion Planning

Jiankun WANG

Department of Electronic and Electrical Engineering
Southern University of Science and Technology

Undergraduate Course, Sep 2025

Outline

Outline

Formulating The Motion Planning Problem

The Piano Mover's Problem

Given

- A world $\mathcal{W}$, where $\mathcal{W} = \mathbb{R}^2$ or $\mathcal{W} = \mathbb{R}^3$.
- A semi-algebraic obstacle region $\mathcal{O} \subset \mathcal{W}$ in the world.
- A semi-algebraic robot $\mathcal{R}$ in $\mathcal{W}$. It may be a rigid robot or a collection of m rigid elements (links).
- The configuration space $\mathcal{C}$ determined by specifying the set of all possible transformations that can be applied to the robot. Then $\mathcal{C}_{obs}$ and $\mathcal{C}_{free}$ are derived.
- A configuration $q_I \in \mathcal{X}_{free}$ designated as initial configuration.
- A configuration $q_G \in \mathcal{X}_{free}$ designated as goal configuration.

A complete algorithm must compute a continuous path $\tau : [0, 1] \to \mathcal{C}_{free}$, such that $\tau(0) = q_I$ and $\tau(1) = q_G$, or correctly report that such path does not exist.

It has been shown that this problem is PSPACE-hard, which implies NP-hard. Hence, unless P=NP, it is a very hard problem to solve.

Path Planning versus Motion Planning

- Subproblem of the general motion planning problem.
- Path planning is the purely geometric problem, without concern for the dynamics, the duration of motion, or constraints on the motion or on the control inputs.
- The path can be time scaled to create a feasible trajectory.

Other Concepts

- Control inputs: $m = n$ versus $m < n$. Incapable of following any path, even if they are collision-free. A car has $n = 3$ (the position and orientation of the chassis in the plane) but $m = 2$ (forward-backward motion and steering).
- Online versus offline: A motion planning problem requiring an immediate result, because obstacles appear, disappear, or move unpredictably, calls for a fast, online, planner.
- Optimal versus feasible: Objective function $J = \int_0^T L(x(t), u(t))dt$.
- Exact versus approximate: Final state $||x(T) - x_{goal}|| < \epsilon$.
- With or without obstacles: Can be challenging even in the absence of obstacles, particularly if $m < n$ or optimality is desired.

Properties of Motion Planners

- Multiple-query versus single-query planning: Type of problems.
- Anytime planning: Continue to look for better solutions after a first solution is found, until satisfying certain requirements.
- 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: Characterizations of the amount of time the planner takes to run or the amount of memory it requires.

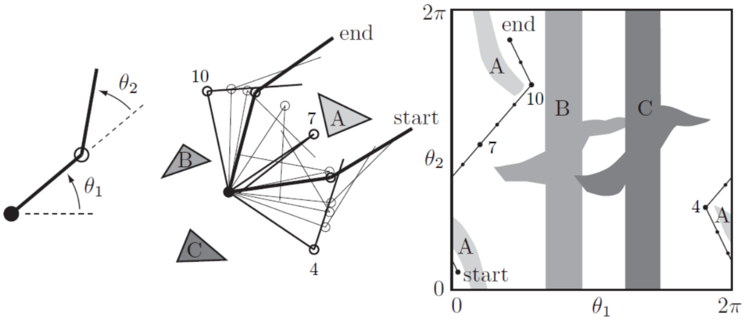
Motion Planning Methods

- Complete methods: Exact representations of the geometry of $\mathcal{C}_{free}$.
- Grid methods: Discretize $\mathcal{C}_{free}$ into a grid.
- Sampling methods: A random or deterministic function to choose a sample from the C-space or state space.
- Virtual potential fields: Online implementation to avoid obstacles.
- Nonlinear optimization: Require an initial guess at the solution, then represent the trajectory or controls by a finite number of design parameters, such as the coefficients of a polynomial.
- Smoothing: Postprocessing to improve the smoothness.

Outline

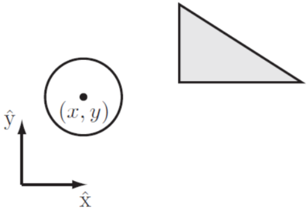
Configuration Space Obstacles

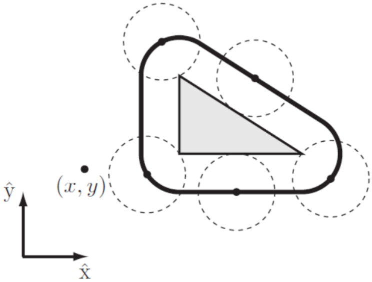
- Degree of C-space equals the degree of freedom of robot.
- Determining whether a robot at a configuration q is in collision with a known environment generally requires a complex operation involving a CAD model of the environment and robot.
- $\mathcal{C} = \mathcal{C}_{free} \cup \mathcal{C}_{obs}$.
- The explicit mathematical representation of a C-obstacle can be exceedingly complex, and for that reason C-obstacles are rarely represented exactly.



- The joint angles of a 2R robot arm.
- The arm navigating among obstacles A, B, and C.
- The same motion in C-space. Three intermediate points, 4, 7, and 10, along the path are labeled.

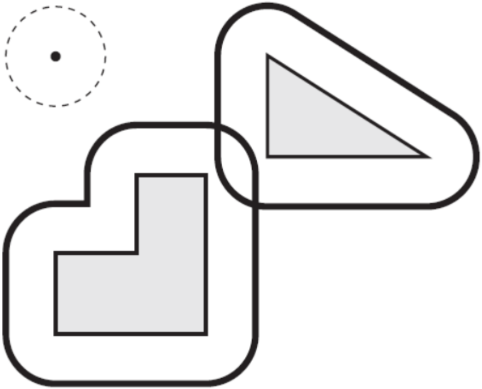
A Circular Planar Mobile Robot





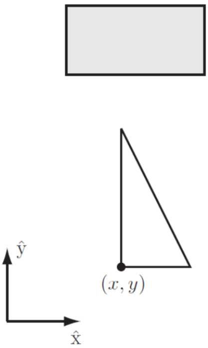
- A circular mobile robot (open circle) and a workspace obstacle (gray triangle). The configuration of the robot is represented by (x, y), the center of the robot.
- In the C-space, the obstacle is "grown" by the radius of the robot and the robot is treated as a point. Any (x, y) configuration outside the bold line is collision-free.

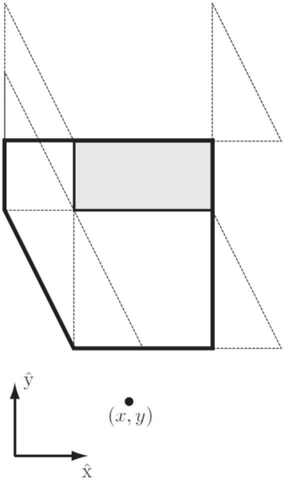
A Circular Planar Mobile Robot



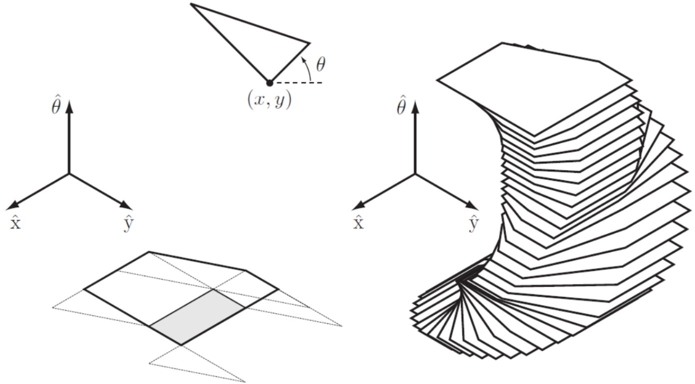
- The “grown” C-space obstacles corresponding to two workspace obstacles and a circular mobile robot. The overlapping boundaries mean that the robot cannot move between the two obstacles.

A Planar Mobile Robot That Translates and Rotates





A Planar Mobile Robot That Translates and Rotates



- A triangular mobile robot that can both rotate and translate, represented by the configuration (x, y, θ).
- The C-space obstacle when the robot is restricted to $\theta = 0$.
- Full 3-dimensional C-space obstacle shown in slices at 10° increments.

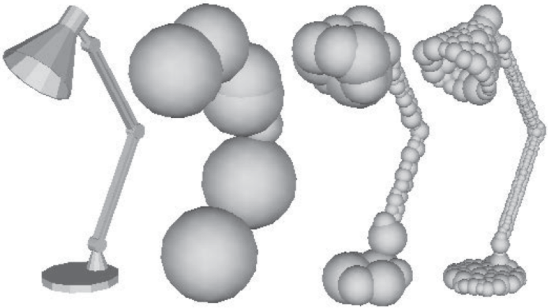
Distance to Obstacles and Collision Detection

- Given a C-obstacle $\mathcal{B}$ and a configuration q, let $d(q, \mathcal{B})$ be the distance between robot and obstacle, then
  - $d(q, \mathcal{B}) > 0$, no contact with the obstacle;
  - $d(q, \mathcal{B}) = 0$, contact;
  - $d(q, \mathcal{B}) < 0$, penetration.
- A distance-measurement algorithm determines $d(q, \mathcal{B})$.
- A collision-detection routine determines whether $d(q, \mathcal{B}_i) \leq 0$ for any C-obstacle $\mathcal{B}_i$.
- A collision-detection routine returns a binary result and may or may not utilize a distance-measurement algorithm at its core.

Distance to Obstacles and Collision Detection

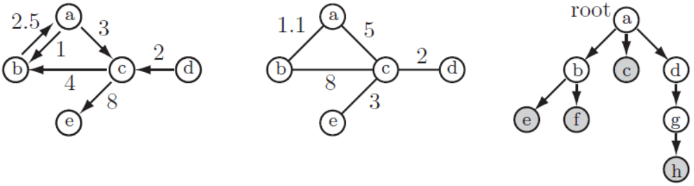
- Approximate the robot and obstacles as unions of overlapping spheres.
- Given a robot at q represented by k spheres of radius R_i centered at $r_i(q)$, $i = 1, ..., k$, and an obstacle $\mathcal{B}$ represented by l spheres of radius $\mathcal{B}_j$ centered at b_j, $j = 1, ..., l$, the distance

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



Graphs and Trees

- Many motion planners explicitly or implicitly represent the C-space or state space as a graph.
- Node $\rightarrow$ configuration or state; edge $\rightarrow$ the ability to move from n_1 to n_2 while satisfying all constraints.
- Directed or undirected.
- Weighted or unweighted.
- A tree is a digraph: (1) no cycles; (2) each node has at most one parent node.



Graphs Search

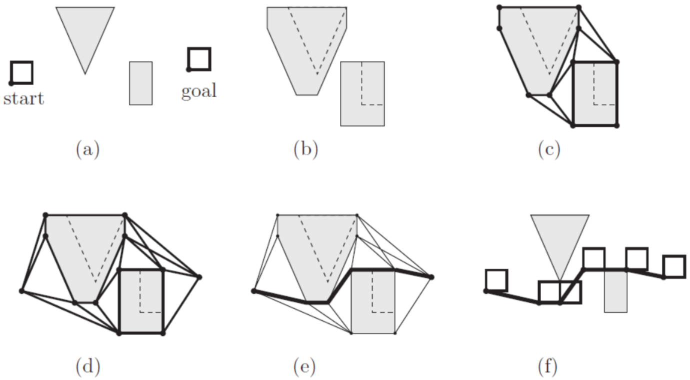
- Once the free space is represented as a graph, a motion plan can be found by searching the graph for a path from the start to the goal.
- Breadth-first search (BFS).
- Depth-first search (DFS).
- Uniform-cost search (UCS).
- Greedy search.
- A* search.

Outline

Complete Path Planners

- Complete path planners rely on an exact representation of the free C-space $\mathcal{C}_{free}$. Mathematically and algorithmically sophisticated, impractical for many real systems.
- Reachability: From every point $q \in \mathcal{C}_{free}$, a free path to a point $q' \in R$ can be found trivially (e.g., a straight-line path).
- Connectivity: For each connected component of $\mathcal{C}_{free}$, there is one connected component of Roadmap R.
- While constructing a roadmap of $\mathcal{C}_{free}$ is complex in general, some problems admit simple roadmaps.
- A suitable roadmap is the weighted undirected visibility graph, with nodes at the vertices of the C-obstacles and edges between the nodes that can "see" each other.

Complete Path Planners

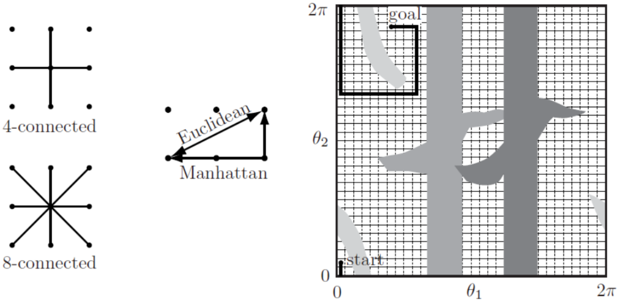


Outline

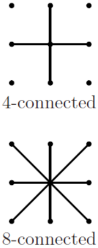
Introduction

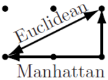
- A search algorithm like A* requires a discretization of the search space. The simplest discretization of C-space is a grid.
- For n-dimensional configuration space, we desire k grid points along each dimension, the C-space is represented by k^n grid points.
- Move in axis-aligned directions or multiple dimensions simultaneously?
- If only axis-aligned motions $\rightarrow$ Manhattan distance.
- A grid-based path planner is resolution complete: it will find a solution if one exists at the level of discretization of the C-space.

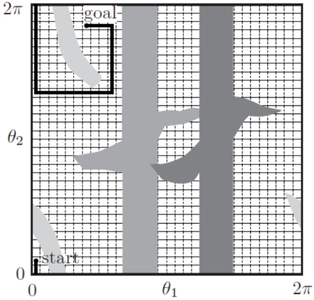
Introduction

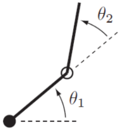


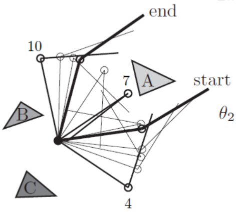
- A 4-connected grid point and an 8-connected grid point for a space $n = 2$.
- Grid points spaced at unit intervals. The Euclidean distance between the two points indicated is $\sqrt{5}$ while the Manhattan distance is 3.
- A grid representation of the C-space and a minimum-length Manhattan-distance path.

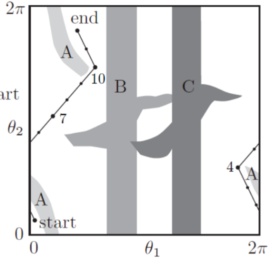

4-connected
8-connected


Euclidean
Manhattan


2π
goal
θ_2
start
0
0
θ_1
2π

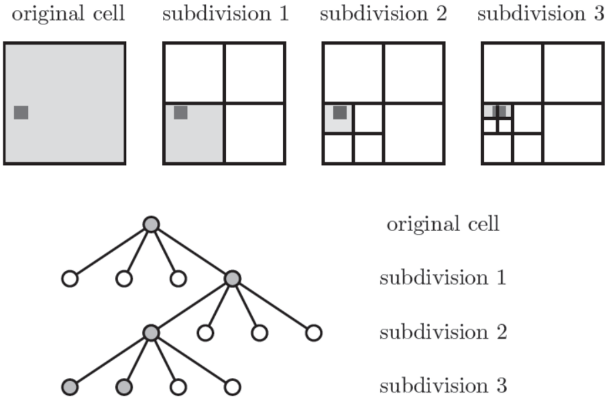

θ_2
θ_1


end
10
7
A
B
C
start
4


2π
end
A
10
B
C
7
θ_2
4
A
A
start
0
0
θ_1
2π

Multi-Resolution Grid Representation

- Any cells that are in contact with a C-obstacle are subdivided further, up to a specified maximum resolution.
- Use only 10 cells to represent an obstacle at the same resolution as a fixed grid that uses 64 cells.



Outline

Introduction

- Inspired by potential energy fields in nature, such as gravitational and magnetic fields.
- E.g., the potential energy of a mass m in height h is mgh and the force acting on it is $-mg$. The force will cause the mass to fall to the Earth's surface.
- In robotics, the goal is assigned a low virtual potential and obstacles are assigned a high virtual potential.
- Typically the gradient of the field can be calculated quickly, so the motion can be calculated in real time (reactive control) instead of planned in advance.
- The robot can get stuck in local minima of the potential field, away from the goal, even when a feasible motion to the goal exists.

A Point in C-space

- Assume a point robot in its C-space. q_{goal} is typically encoded by a quadratic potential energy "bowl" with zero energy at the goal

$$\mathcal{P}_{goal}(q) = \frac{1}{2}(q - q_{goal})^T K (q - q_{goal}),$$

where K is a symmetric positive-definite weighting matrix.

- The force induced by this potential is

$$F_{goal}(q) = -\frac{\partial \mathcal{P}_{goal}}{\partial q} = K(q_{goal} - q),$$

an attractive force proportional to the distance from the goal.

A Point in C-space

- The repulsive potential induced by a C-obstacle $\mathcal{B}$ can be calculated from the distance $d(q, \mathcal{B})$ to the obstacle

$$\mathcal{P}_{\mathcal{B}}(q) = \frac{k}{2d^2(q, \mathcal{B})},$$

where $k > 0$ is a scaling factor, $d(q, \mathcal{B}) > 0$.

- The force induced by the obstacle potential is

$$F_{\mathcal{B}}(q) = -\frac{\partial \mathcal{P}_{\mathcal{B}}}{\partial q} = \frac{k}{d^3(q, \mathcal{B})} \frac{\partial d}{\partial q}.$$

A Point in C-space

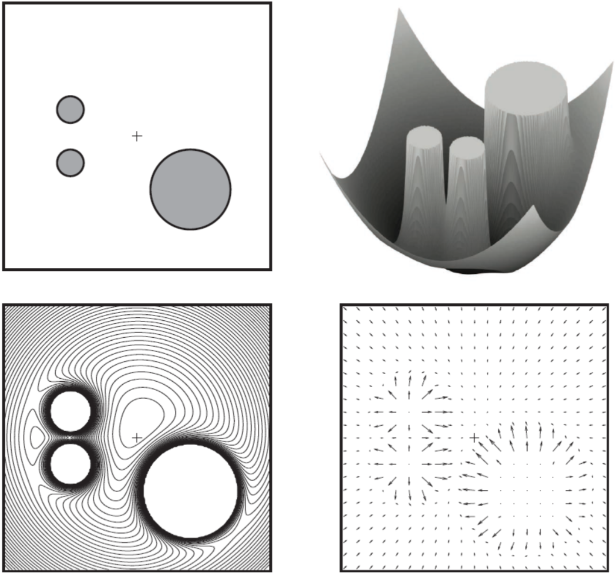
- The total potential is obtained by summing the attractive goal potential and the repulsive obstacle potentials,

$$\mathcal{P}(q) = \mathcal{P}_{goal}(q) + \sum_i \mathcal{P}_{\mathcal{B}_i}(q),$$

yielding a total force

$$F(q) = F_{goal}(q) + \sum_i F_{\mathcal{B}_i}(q).$$

A Point in C-space



A Point in C-space

- To actually control the robot using the calculated $F(q)$, we treat the calculated force as a commanded velocity:

$$u = F(q).$$

- Using the simple obstacle potential, even distant obstacles have a nonzero effect on the motion of the robot.
- To speed up evaluation of the repulsive terms, distant obstacles could be ignored.

$$U_{\mathcal{B}}(q) = \begin{cases} \frac{k}{2}\left(\frac{d_r - d(q,\mathcal{B})}{d_r d(q,\mathcal{B})}\right)^2 & \text{if } d(q,\mathcal{B}) < d_r, \\ 0 & \text{otherwise.} \end{cases}$$

Navigation Functions (Option)

- A significant problem with the potential field method is local minima.
- An approach to local-minimum-free gradient following is based on replacing the virtual potential function with a navigation function.
- A navigation function $\phi(q)$ is a type of virtual potential function that
  - is smooth (or at least twice differentiable) on q;
  - has a bounded maximum value on the boundaries of all obstacles;
  - has a single minimum at q_{goal};
  - has a full-rank Hessian $\partial^2\phi/\partial q^2$ at all critical points q where $\partial\phi/\partial q = 0$.

Navigation Functions (Option)

- A significant problem with the potential field method is local minima.
- An approach to local-minimum-free gradient following is based on replacing the virtual potential function with a navigation function.
- A navigation function $\phi(q)$ is a type of virtual potential function that
  - is smooth (or at least twice differentiable) on q;
  - has a bounded maximum value on the boundaries of all obstacles;
  - has a single minimum at q_{goal};
  - has a full-rank Hessian $\partial^2\phi/\partial q^2$ at all critical points q where $\partial\phi/\partial q = 0$.

Q & A

1. Overview of Motion Planning
2. Foundations
3. Complete Path Planners
4. Grid Methods
5. Virtual Potential Fields