A Brief Introduction:

The Motion Planning Problem
Configuration Space
Basic Path Planning Methods

Acknowledgement: Parts of these course notes are based on notes from courses given by Jean-Claude Latombe at Stanford University (and Chapter 1 in his text *Robot Motion Planning*, Kluwer, 1991), O. Burçhan Bayazıt at Washington University in St. Louis. Seth Hutchinson at the University of Illinois at Urbana-Champaign, and Leo Joskowicz at Hebrew University.

What is Motion Planning?

General Goal: compute motion 'commands' to achieve a goal arrangement of physical objects from an initial arrangement

---

Basic problem: Collision-free path planning for one rigid or articulated object (the "robot") among static obstacles.

INPUTS:

- geometric descriptions of the obstacles and the robot
- kinematic and dynamic properties of the robot
- initial and goal positions (configurations) of the robot

OUTPUT:

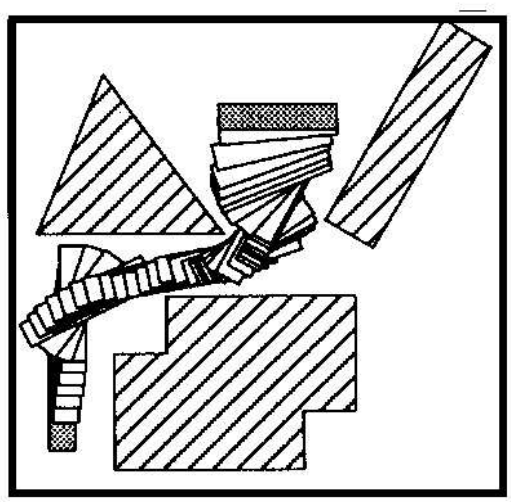
- Continuous sequence of collision-free configurations connecting the initial and goal configurations

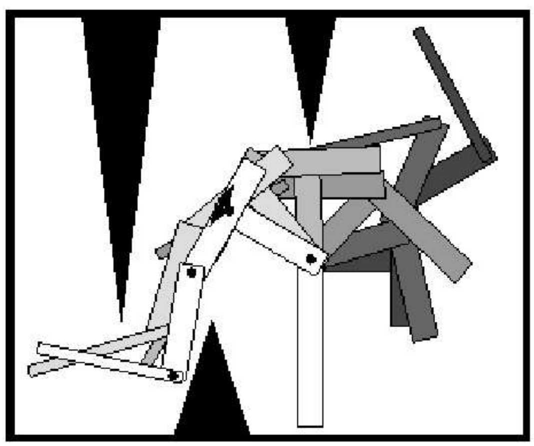
---

Examples:

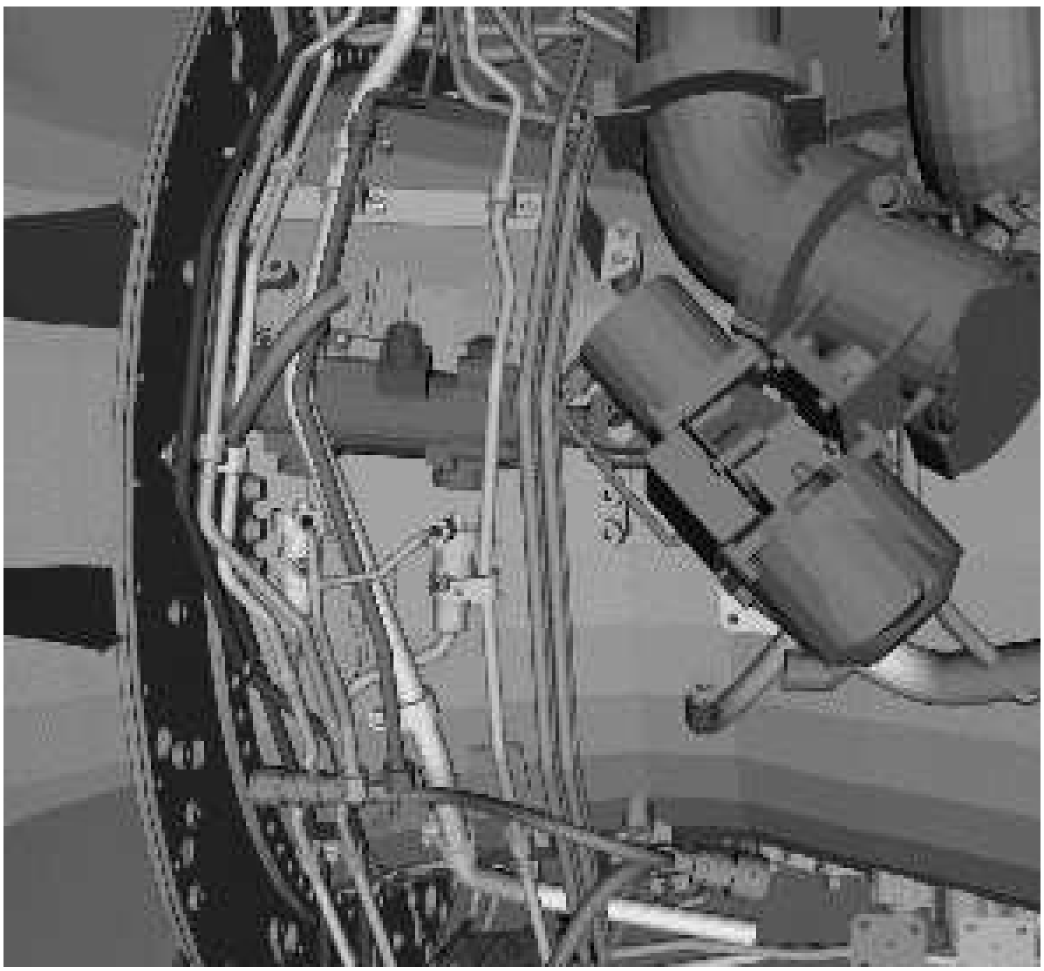
- autonomous robot delivering coffee (from kitchen to your office)
- analyzing/animating CAD objects (part removal)
- drug design (molecule docking)
- hip implant design (insertability studies)

Robot Path Planning

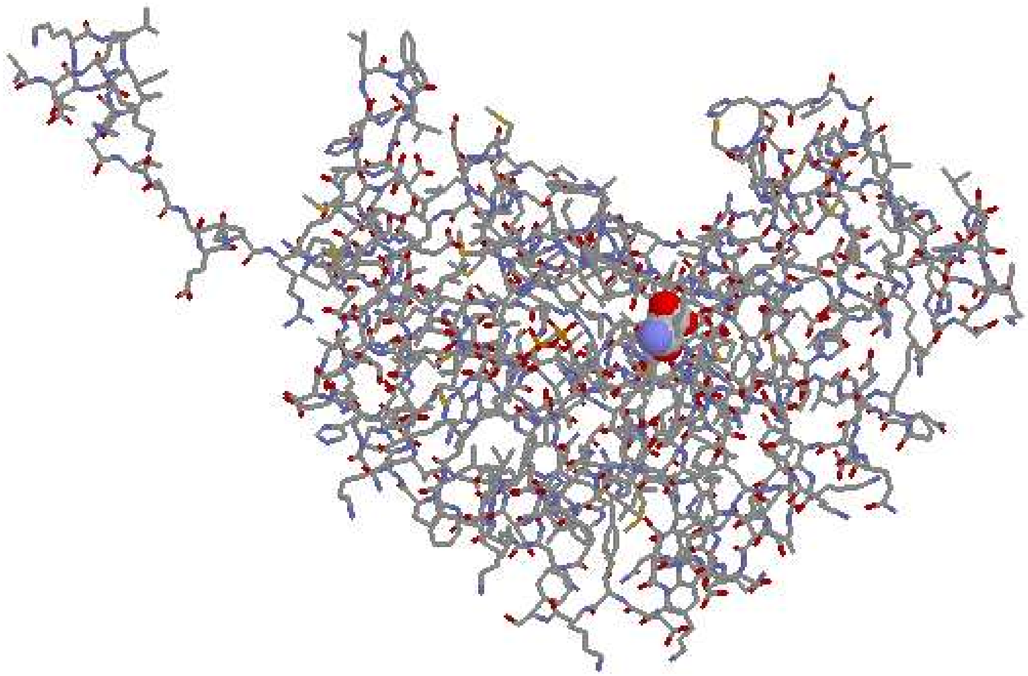




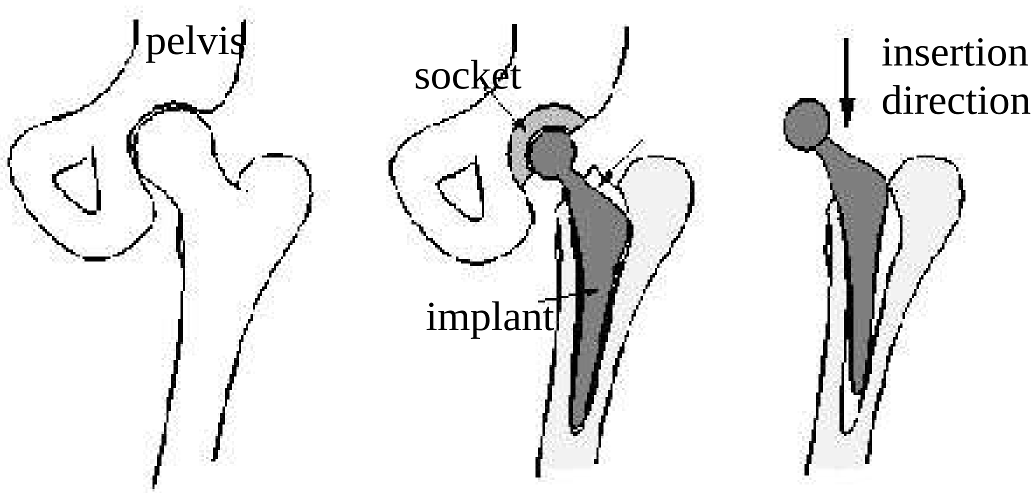
CAD Studies – Part Removal



Drug Design – Docking Motions



Hip Implant Insertability Analysis



(a) Damaged Joint (b) Replaced Implant (c) Implant Insertion

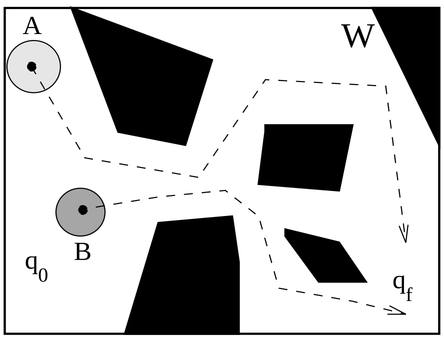
Extensions to the Basic Problem:

- movable obstacles
- moving obstacles
- multiple robots
- incomplete knowledge/uncertainty in geometry, sensing, etc.
- nonholonomic constraints

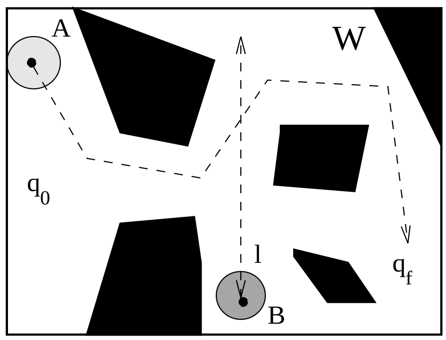
Examples:

- robot assembling parts (manipulation planning)
- assembly sequencing (what order to assemble parts)
- autonomous vehicles (nonholonomic constraints)

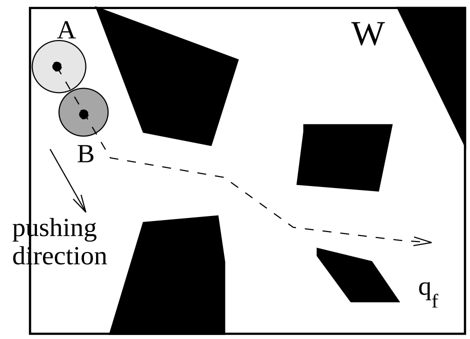
Multiple Robots, Moving/Movable Obstacles



(a) B is a moving robot

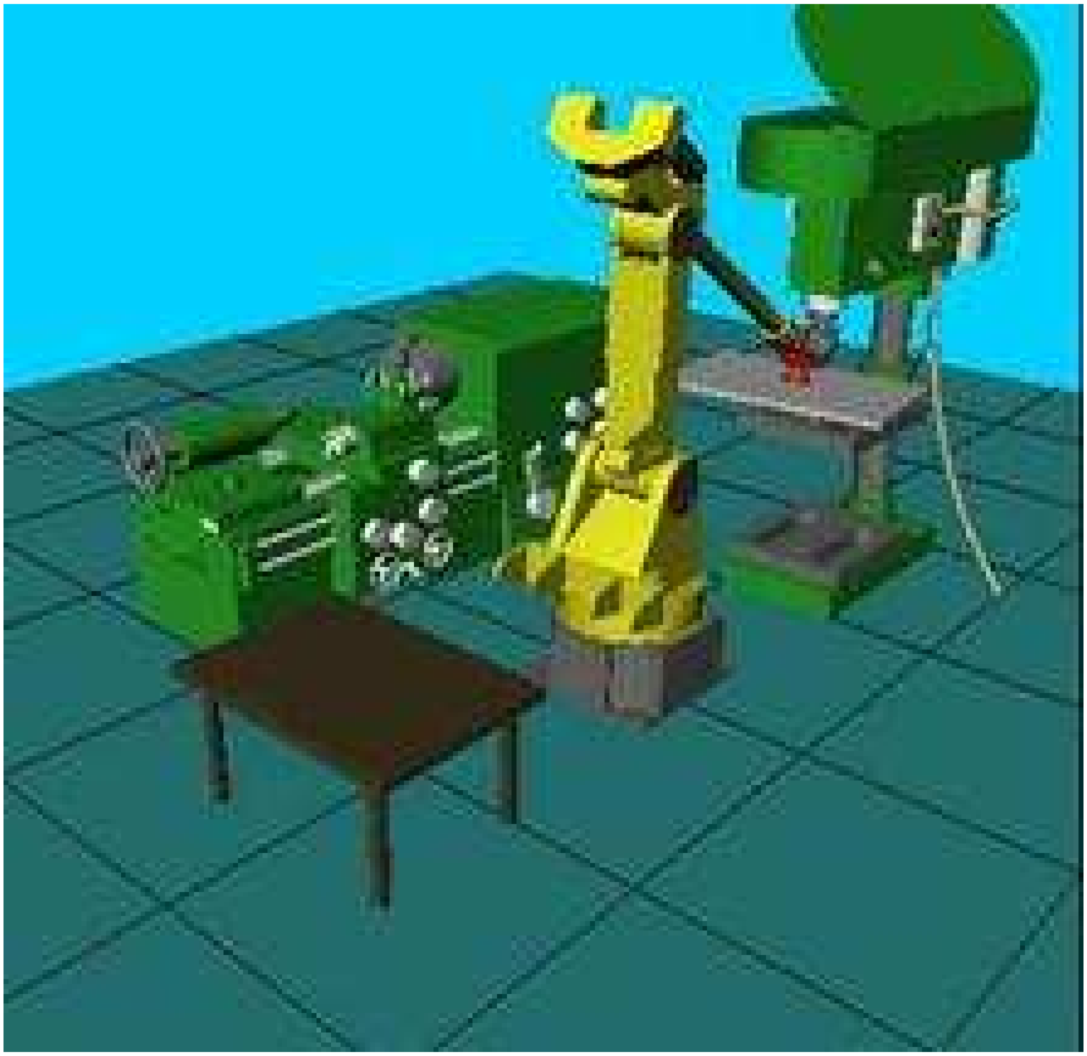


(b) B is an obstacle moving on a vertical line

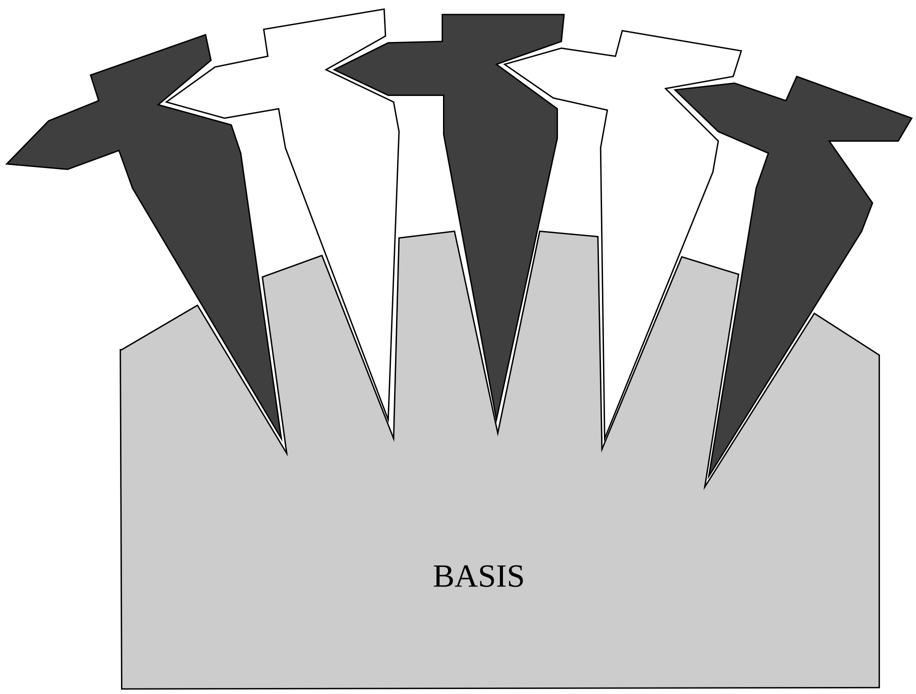


(c) B is an object pushed by the robot

Manipulation Planning



Assembly Sequencing



Assembly requiring K hands to dissassemble

1. Each peg requires a different direction of motion
2. No peg can move by itself
3. No two, three, four pegs can be moved together in the same direction

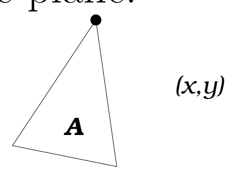
Configuration Space: A useful Abstraction

Main Idea: Represent the robot as a point, called a *configuration*, in a parameter space, the *configuration space* (or *C-space*).

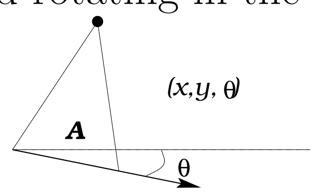
- A robot configuration is a specification of the positions of all robot points relative to a fixed coordinate system
- configurations often expressed as vector of position/orientation parameters

<u>*Examples:*</u>

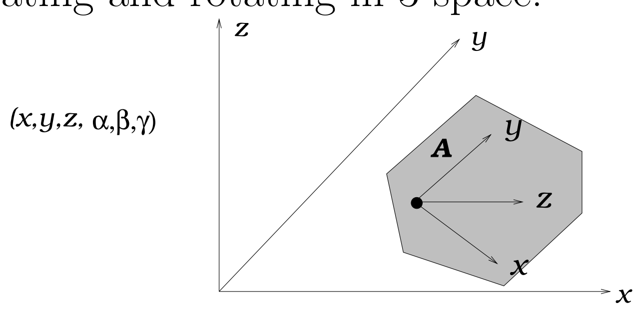
- Rigid robot translating in the plane:



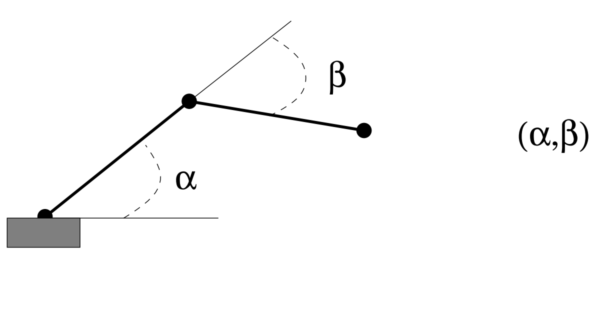
- Rigid robot translating and rotating in the plane:



- Rigid robot translating and rotating in 3-space:



- Articulated robot:



Notation

- Workspace: $\mathcal{W}$
- Robot: $\mathcal{A}$
- Configuration: $\mathbf{q}$
  - a vector of parameters specifying position of A in W
  - *degrees of freedom (dof)* of A is number of parameters needed to specify a position
- Configuration Space: $\mathcal{C}$
  - set of all possible configurations
- Region of $\mathcal{W}$ occupied by $\mathcal{A}$ when at configuration $\mathbf{q}$: $\mathcal{A}(\mathbf{q})$
- Position of point $a \in \mathcal{A}$ in $\mathcal{W}$ when $\mathcal{A}$ is at $\mathbf{q}$: $a(\mathbf{q})$
- Obstacle region in $\mathcal{W}$: $\mathcal{B}$
  - the blocked or forbidden region of $\mathcal{C}$

Robot Configurations can be:

1. *free configurations*: robot and obstacles do not overlap
2. *contact configurations*: robot and obstacles touch
3. *blocked configurations*: robot and obstacles overlap

Configuration Space partitioned into free ($\mathcal{C}_{free}$), contact, and blocked sets.

Obstacle Region in C-Space

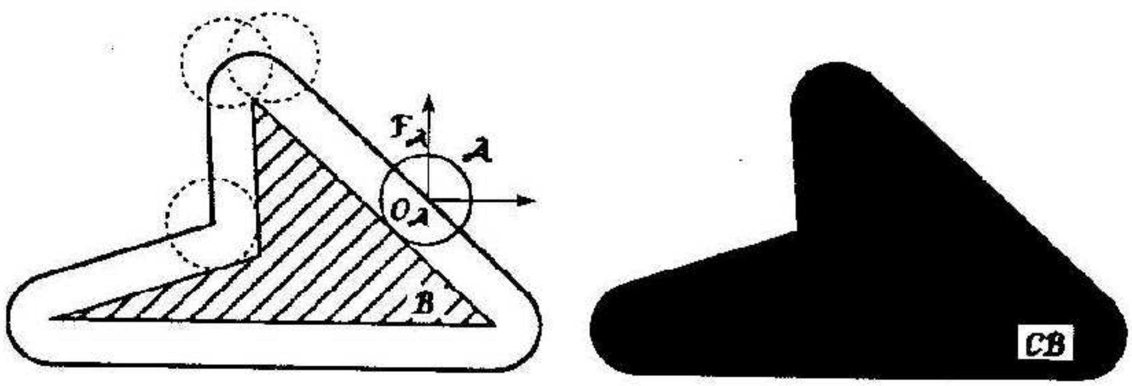
Workspace Obstacles map to forbidden regions (*C-obstacles*) in C-space:

$$\mathcal{CB} = \{\mathbf{q} \,|\, \mathcal{A}(\mathbf{q}) \cap \mathcal{B} \neq \emptyset\}$$

i.e., for each obstacle $\mathcal{B}_i$ in $\mathcal{W}$, we have a corresponding C-obstacle $\mathcal{CB}_i$ which represents all configurations of the robot $\mathcal{A}$ that collide (intersect) with $\mathcal{B}_i$.

Example: $\mathcal{A}$ is a disc and $\mathcal{W} = \mathbb{R}^2$

- $\mathcal{CB}$ of polygon $\mathcal{B}$ obtained by growing $\mathcal{B}$ by radius of $\mathcal{A}$



Many Possible Notions of Free (Blocked) Configurations:

- Traditional meaning for path planning is collision-free (colliding).
- The robot is far enough (too close) from the obstacles. (So, some collision-free cfgs are invalid.)
- The configuration of the molecule has low enough (too high) potential energy.
- etc.

Paths in C-Space

A *path in C-space* is a 'piece' of a continuous curve connecting two configurations q_{init} and q_{goal}. Or, more formally, a path is continuous map:

$$\tau : s \in [0,1] \mapsto \tau(s) \in \mathcal{C}$$

where $\tau(0) = \mathbf{q}_{init}$ is the initial configuration and $\tau(1) = \mathbf{q}_{goal}$ is the goal configuration of the path.

"Continuous map" means that:

$$\forall s_1, s_2 \in [0,1] : \lim_{s_2 \to s_1} d(\tau(s_1), \tau(s_2)) = 0$$

where $d : \mathcal{C} \times \mathcal{C} \to \mathbf{R}^+ \cup \{0\}$ is the chosen metric over $\mathcal{C}$.

Example of a metric in $\mathcal{C}$:

$$d(\mathbf{q}_1, \mathbf{q}_2) = max_{a \in \mathcal{A}} \|a(\mathbf{q}_1) - a(\mathbf{q}_2)\|$$

where $\|x - y\|$ is the Euclidean metric in $\mathcal{W}$.

Other Potential Constraints on the path

- minimal length, smoothness, number of turns, limits on curvature, etc.
- trajectory can be parameterized by time (effectively adding another dimention to configuration space)

Paths

Types of Paths:

- *Free path*: $\tau : [0,1] \rightarrow \mathcal{C}_{free} = \mathcal{C} \backslash \mathcal{CB}$
  (entirely in $\mathcal{C}_{\mathrm{free}}$, no contact)
- *Semi-free path*: $\tau : [0,1] \rightarrow cl(\mathcal{C}_{free})$ (may have contact, but not invalid)

Homotopic Paths

- Two paths with the same start and goal configurations are *homotopic* if one can be continuously deformed into the other

Connectedness of C-Space

- C is connected if all pairs of configurations in C can be connected by a path
- C is simply connected if any two paths connecting the same endpoints are homotopic.
  Examples: $\mathbb{R}^2$ or $\mathbb{R}^3$
- Otherwise, C is multiply connected.
  Examples: S^1 (infinite number of homotopy classes) and $SO(3)$ (two homotopy classes)

Why Configuration Space?

- *reduce problem* of finding a dimensioned robot (many links) in Euclidean space to finding a path for a point robot in (higher-dimensional) C-space
- *uniform framework* in which to study problem and compare/evaluate algorithms
- *(potentially) easier* to plan point trajectories
- nearly everyone uses it...

Difficulty: motion planning is continuous (configuration space formulation)

Approach: discretize and then use graph search

- discretize the planning space (C-space)
- discretize the path
- discretize both

Computational Complexity of Path Planning

There is strong evidence that path planning takes:

- *exponential time* in the number of dimensions of the C-space;
- polynomial time in the complexity of the C-obstacle region (number of surface patches and degree of their algebraic equations).

Two general path planning methods are known (mainly of theoretical interest):

- An exact cell decomposition method based on the notion of a cylindrical decomposition (Collins decomposition) by Schwartz and Sharir (1983) – doubly exponential in dof (e.g., 2^{2^d})
- A roadmap method based on computing the "silhouette" of the free space by Canny (1987) – singly exponential in dof (e.g., 2^d)

The above complexity still holds in specific cases, e.g., the robot is a set of independent translating rectangles, the robot is a planar linkage with revolute joints.

It holds for *complete* planning, i.e.: the planner correctly returns a solution path whenever one exists and declares that no path exists otherwise.

Common Approaches: identify and exploit the structure of important special cases and tasks e.g., simplify robot/obstacle geometry, limit dof, simplify paths

Basic Solution Methods

Two Step Methods:

- PREPROCESSING:
  – Capture free space connectivity into a graph or a function
- QUERY PROCESSING:
  – Search the graph for a path

Path Planning Approaches:

- **Cell decomposition:**
  decompose C-free into simple cells and represent the connectivity of the free space by the adjacency graph of the cells
- **Roadmaps:**
  represent the connectivity of C-free by a graph, a network of 1D curves
- **Potential field:**
  define a function over the free space that has a global minimum at the goal configuration and follow its steepest descent

Cell Decomposition Methods

Cell: Connected region of free space, such that computing a path between any two points in the same cell is straightforward.

example: cells are convex polygons

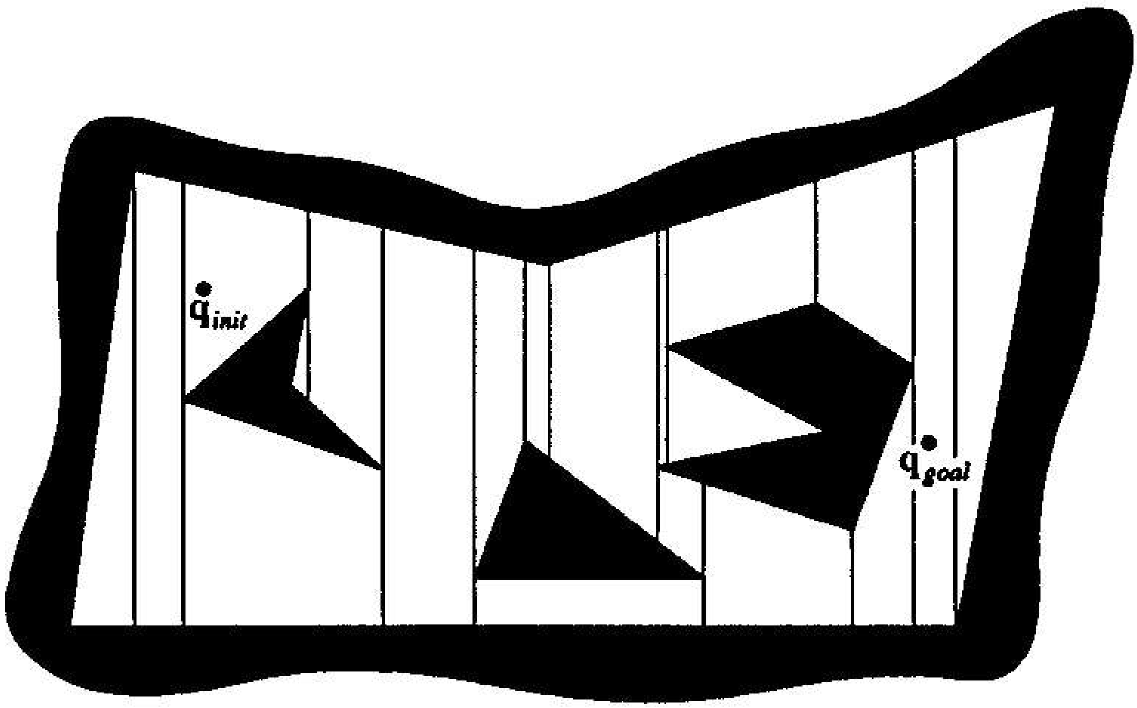
- PREPROCESSING:
  - Represent the free space as a collection of cells.
  - Generate the *connectivity graph* representing the adjacency relation between cells.
- QUERY PROCESSING:
  - Search the connectivity graph for a sequence of adjacent cells connecting the initial to the goal cell.
  - Transform the sequence of cells (if one has been produced) into a path.

Two major variants of methods: *exact* and *approximate* cell decomposition

Example of Cell Decomposition Method

Trapezoidal (or Vertical) Decomposition

(Exact Decomposition)



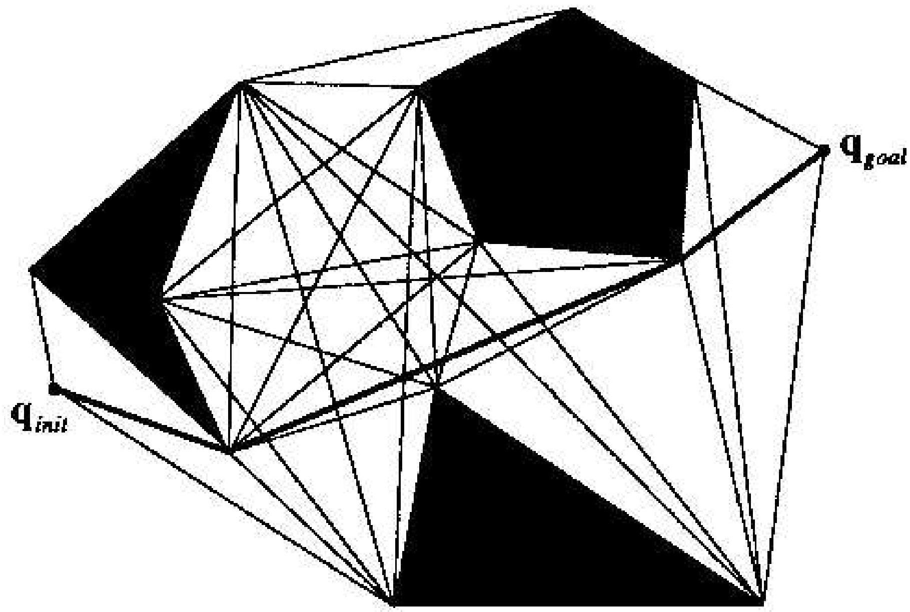
Roadmap Methods

- PREPROCESSING:
  - construct a network of curves (the *roadmap*) that "adequately" represents the connectivity of the free space
- QUERY PROCESSING:
  - connect the initial and goal positions of the robot to the roadmap
  - search the resulting graph for a path between the initial and goal positions of the robot.

Many possible definitions of the roadmap, e.g., visibility graph, Voronoi diagram.

Example of Roadmap Method

Visibility Graph



Note: obtain semi-free paths (have contact)

- brute force computation of visibility graph: $O(n^3)$.
- computation using rotating sweep-rays: $O(n^2 \log n)$.
- other techniques: $O(n^2)$.

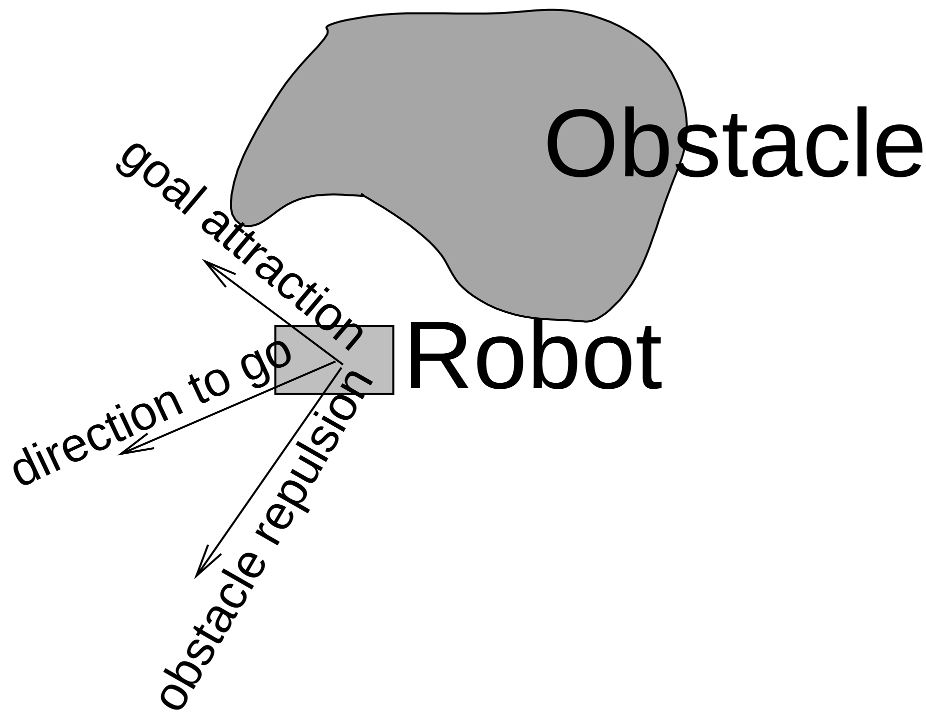
Shortest Path obtained by graph search using $A^\star$ (with Euclidean distance to the goal as admissible heuristic): $O(n^2 \log n)$.

Potential Field Methods

- PREPROCESSING:
  - place a grid over the robot's space
  - define a function (the *potential field*) over the robot's space, with a global minimum at the goal.
- PREPROCESSING:
  - search the grid for a path using the potential field as the heuristic (e.g., follow the steepest descent).

Example of Potential Field Method

- Goal



Task/Solution Type

TASK TYPE

- path planning
- manipulation planning
- task-level robot programming
- assembly planning and verification
- design, analysis, modification (CAD)
- simulation, physical modeling, virtual reality
- molecules

SOLUTION TYPE

- exact or approximate solution
- optimal, close enough, any solution
- soundness/completeness of solution: heuristic, guaranteed, probabilistic
- ’local’ or global techniques

Problem Characteristics

PHYSICAL CHARACTERISTICS

- **object geometry:** 2 or 3 dimensions, shape primitives
- **kinematics:** degrees of freedom, axes of motion, mobility constraints
- **object interactions:** rigid or flexible, interpenetration
- **dynamics:** friction, inertia
- **robot capabilities:** control, sensing, and reacting

KNOWLEDGE CHARACTERISTICS

- **completeness:** complete, incomplete, varying
- **uncertainty:** knowledge acquisition, approximations, motion execution

In this course...

$$\boxed{\textit{motion planning} = \textit{geometry} + \textit{search}}$$

<u>OF INTEREST</u>

- rigid, non-penetrating objects
- purely kinematic (no dynamics)
- path planning, manipulation planning
- multiple robots
- applications (computer animation, structural biology, etc)

<u>OUTSIDE SCOPE</u>

- "classical" robotics
- dynamics
- robot control and sensing

Summary

Motion Planning

- is an important problem with many applications
- = geometry + search
- is hard $\Longrightarrow$ identify/study special cases; use heuristics
- is continuous $\Longrightarrow$ it requires discretization

Course Organization

MOTION PLANNING TECHNIQUES (2-3 lectures)

- configuration space construction
- exact and approximate cell decomposition
- potential field methods
- roadmap methods
- probabilistic roadmap methods

APPLICATIONS (3-4 lectures)

- animation (coordinated group behavior)
- structural biology (protein folding, protein/ligand binding)