Introduction to Robot Motion Planning

Robotics meet Computer Science
Example

A robot arm is to build an assembly from a set of parts.

Tasks for the robot:

• Grasping: position gripper on object
design a path to this position

• Transferring: determine geometry path for arm
avoid obstacles + clearance

• Positioning
Information required

- Knowledge of spatial arrangement of workspace. E.g., location of obstacles

- Full knowledge → full motion planning

- Partial knowledge → combine planning and execution

motion planning = collection of problems
Basic Problem

A simplified version of the problem assumes:

- Robot is the only moving object in the workspace
- No dynamics, no temporal issues
- Only non-contact motions

$MP = \text{pure "geometrical" problem}$
Components of BMPP

- **A**: single rigid object - the robot - moving in Euclidean space $W$ (the wkspace).

  $$W = \mathbb{R}^N, \; N=2,3$$

- **$B_i$, i=1,...,q.** Rigid objects in $W$. The obstacles
Assume

- Geometry of $A$ and $B_i$ is perfectly known
- Location of $B_i$ is known
- No kinematic constraints on $A$: a “free flying” object
Components of BMPP (cont.)

• The Problem:
  – Given an initial position and orientation $PO_{init}$
  – Given a goal position and orientation $PO_{goal}$
  – Generate: continuous path $t$ from $PO_{init}$ to $PO_{goal}$

• $t$ is a continuous sequence of Pos’
**Configuration Space Idea**

1. represent robot as point in space
2. map obstacles into this space
3. transform problem from planning object motion to planning point motion
Configuration Space (cont.)

$W$: Euclidean space in which motion occurs

$A$ at a given position is a compact in $W$. Attach $F_A$

$B_i$ closed subset of $W$.

$F_W$ is a frame fixed in $W$
Def: configuration of an object

Position of every point of the object w.r.t. $F_W$

Def: Configuration $q$ of $A$

Position $T$ and orientation $O$ of $F_A$ w.r.t. $F_W$

Def: configuration space of $A$

Is the space $T$ of all configurations of $A$

- $A(q)$: subset of $W$ occupied by $A$ at $q$
- $a(q)$: is a point in $A(q)$
Information Required

• Example: $T$: N-dimensional vector

$O$: NxN rotation matrix

• In this case, $q = (T,O)$, a subset of $\mathbb{R}^{N(N+1)}$

• Note that $C$ is locally like $\mathbb{R}^3$ or $\mathbb{R}^6$.

Notice: no global correspondence
Mathematic Notion of Path

- Need a notion of continuity
- Define a distance function $d : C \times C \rightarrow R^+$
  - Example: $d(q, q') = \max_{a \in A} \|a(q) - a(q')\|_1$
Notion of Path (cont.)

- **Def:** A path of $A$ from $q_{\text{init}}$ to $q_{\text{goal}}$ is a continuous map $t: [0,1] \rightarrow C$

  \[ t(0) = q_{\text{init}} \text{ and } t(1) = q_{\text{goal}} \]

- **Property:** $t$ is continuous if for each $s_o$ in $(0,1)$, 
  \[ \lim_{s \rightarrow s_o} d(s, s_o) = 0 \] when $s \rightarrow s_o$
Obstacles in Configuration Space

- Obstacle $B_i$ maps in $C$ to a region

$$CB_i = \{ q \in C, \text{ s.t. } A(q) \text{ and } B_i \text{ are not disjoint} \}$$

- Example: “round” robot with no preferred orientation
Obstacles in C- Space (cont.)

- Obstacles in $C$ are called $C$-obstacles.
- $C$-obstacle region: Union of all $Cb_i$
- Free space: $C_{\text{free}} = C - U Cb_i$
- $q$ is a free configuration if $q$ belongs to $C_{\text{free}}$
- Def: Free Path.

Is a path between $q_{\text{init}}$ and $q_{\text{goal}}$, $t: [0,1] \rightarrow C_{\text{free}}$
Obstacles in $C$ (cont.)

- **Def:** Connected Component

  $q_1, q_2$ belong to the same connected component of $C_{\text{free}}$ iff they are connected by a free path.

**Objective of Motion Planning:**

generate a free path between 2 configurations if one exists or report that no free path exists.
Examples of $C$-Obstacles

- Translational Case:
  1. $A$ is a single point -> no orientation
    \[ W = \mathbb{R}^N = C \]
  2. $A$ is a disk or dimensioned object allow to translate freely but without rotation.

$C$-Obstacles: obstacles “grown” by the shape of $A$
Planning Approaches

• 3 approaches: road maps, cell decomposition and potential field

1- Roadmap

Captures connectivity of $C_{\text{free}}$ in a network of 1-D curves called “the roadmaps.”

Once a roadmap is constructed: use a standard path.

Roadmap Construction Methods: 1) Visibility Graph, 2) Voronoi Diagram, 3) Freeway Net and 4) Silhouette.
Visibility graph in 2D CS. Nodes: initial and final config + vertices of C-obstacles.
Cell Decomposition

- Decompose the free space into simple regions called cells
- Construct a non-directed graph representing adjacencies: the *connectivity graph*
- Search for a path forming a “channel”
- Two variations:
  - Exact: union of cells is exactly the free space
  - Approximate: union included in the free space
Cell Decomposition: Example

$q_{\text{init}}$

$q_{\text{goal}}$

$\text{init}$

$\text{goal}$
Extensions of the Basic Problem

- Multiple moving objects
  - Multiple obstacles
  - Multiple Robots
  - Articulated Robots

- Kinematic Constraints

- Uncertainty

- Movable objects
Computational Complexity

- Instances may differ in “size”: dimension of C-space and # of obstacles

- **Result 1**: planning a free path for a robot made of an arbitrary # of polyhedral bodies connected by joints, among a finite set of polyhedral obstacles is a PSPACE-hard problem

- **Result 2**: A free path in a C-space of fixed dimension m, when the free space is defined by n polynomials of max degree d, can be computed exponentially in m and polynomial in n and d