Mobile Robotics

Configuration Space - Basic Path-Planning Methods
What is a Path?
Tool: Configuration Space (C-Space C)
Configuration Space

\[ q = (q_1, \ldots, q_n) \]
Definition

A robot **configuration** is a specification of the positions of all robot points relative to a fixed coordinate system.

Usually a configuration is expressed as a “**vector**” of position/orientation parameters.
Rigid Robot Example

- 3-parameter representation: \( q = (x, y, a) \)
- In a 3-D workspace \( q \) would be of the form \((x, y, z, a, b, g)\)
Articulated Robot Example

\[ q = (q_1, q_2, ..., q_{10}) \]
Configuration Space of a Robot

- Space of all its possible configurations
- But the topology of this space is usually not that of a Cartesian space
Configuration Space of a Robot

- Space of all its possible configurations
- But the topology of this space is usually not that of a Cartesian space

\[ C = S^1 \times S^1 \]
Configuration Space of a Robot

- Space of all its possible configurations
- But the topology of this space is usually not that of a Cartesian space
It is a manifold
For each point $q$, there is a 1-to-1 map between a neighborhood of $q$ and a Cartesian space $\mathbb{R}^n$, where $n$ is the dimension of $C$

This map is a local coordinate system called a chart.
$C$ can always be covered by a finite number of charts. Such a set is called an atlas
Example
Case of a Planar Rigid Robot

- 3-parameter representation: \( q = (x, y, q) \) with \( q \in [0, 2\pi) \). Two charts are needed.
- Other representation: \( q = (x, y, \cos q, \sin q) \)
- \( c \)-space is a 3-D cylinder \( \mathbb{R}^2 \times S^1 \) embedded in a 4-D space.
Rigid Robot in 3-D Workspace

- \( q = (x, y, z, a, b, g) \)

The c-space is a 6-D space (manifold) embedded in a 12-D Cartesian space. It is denoted by \( R^3 \times SO(3) \)

- Other representation: \( q = (x, y, z, r_{11}, r_{12}, ..., r_{33}) \) where \( r_{11}, r_{12}, ..., r_{33} \) are the elements of rotation matrix \( R \):

\[
\begin{pmatrix}
  r_{11} & r_{12} & r_{13} \\
  r_{21} & r_{22} & r_{23} \\
  r_{31} & r_{32} & r_{33}
\end{pmatrix}
\]

with:
- \( r_{i1}^2 + r_{i2}^2 + r_{i3}^2 = 1 \)
- \( r_{i1}r_{j1} + r_{i2}r_{2j} + r_{i3}r_{j3} = 0 \)
- \( \det(R) = +1 \)
Parameterization of $\text{SO}(3)$

- **Euler angles:** $(f,q,y)$

- **Unit quaternion:**
  
  $$(\cos \frac{q}{2}, n_1 \sin \frac{q}{2}, n_2 \sin \frac{q}{2}, n_3 \sin \frac{q}{2})$$
Metric in Configuration Space

A metric or distance function $d$ in $C$ is a map

$$d: (q_1, q_2) \in C^2 \rightarrow d(q_1, q_2) \geq 0$$

such that:

- $d(q_1, q_2) = 0$ if and only if $q_1 = q_2$
- $d(q_1, q_2) = d(q_2, q_1)$
- $d(q_1, q_2) \leq d(q_1, q_3) + d(q_3, q_2)$
Metric in Configuration Space

Example:

• Robot $A$ and point $x$ of $A$

• $x(q)$: location of $x$ in the workspace when $A$ is at configuration $q$

• A distance $d$ in $C$ is defined by:

$$d(q,q') = \max_{x \in \mathcal{W}_A} ||x(q)-x(q')||$$

where $||a - b||$ denotes the Euclidean distance between points $a$ and $b$ in the workspace
Specific Examples in $\mathbb{R}^2 \times S^1$

- $q = (x,y,q), \ q' = (x',y',q')$ with $q, q' \in [0,2\pi)$
- $a = \min\{|q-q'|, 2\pi - |q-q'|\}$

\[
d(q,q') = \sqrt{(x-x')^2 + (y-y')^2 + a^2}
\]

$\text{where } r \text{ is the maximal distance between the reference point and a robot point}$
A path in $C$ is a piece of continuous curve connecting two configurations $q$ and $q'$:
\[ t : s \rightarrow [0,1] \rightarrow t(s) \in C \]
\[ s' \rightarrow s \rightarrow d(t(s), t(s')) \leq 0 \]
Other Possible Constraints on Path

- Finite length, smoothness, curvature, etc...
- A trajectory is a path parameterized by time:

\[ t : \mathbb{R} \rightarrow [0, T] \rightarrow t(t) \in \mathbb{C} \]
A configuration \( q \) is \textit{collision-free}, or \textit{free}, if the robot placed at \( q \) has null intersection with the obstacles in the workspace.

The \textit{free space} \( F \) is the set of free configurations.

A \textit{C-obstacle} is the set of configurations where the robot collides with a given workspace obstacle.

A configuration is \textit{semi-free} if the robot at this configuration touches obstacles without overlap.
Disc Robot in 2-D Workspace
Rigid Robot Translating in 2-D

CB = B \iff \mathcal{A} = \{ b-a \mid a \in \mathbb{A}, b \in \mathbb{B} \}
Linear-Time Computation of C-Obstacle in 2-D

(convex polygons)
Rigid Robot Translating and Rotating in 2-D
C-Obstacle for Articulated Robot
Free and Semi-Free Paths

- A **free path** lies entirely in the free space F
- A **semi-free path** lies entirely in the semi-free space
Remark on Free-Space Topology

• The robot and the obstacles are modeled as closed subsets, meaning that they contain their boundaries.
• One can show that the C-obstacles are closed subsets of the configuration space C as well.
• Consequently, the free space F is an open subset of C. Hence, each free configuration is the center of a ball of non-zero radius entirely contained in F.
• The semi-free space is a closed subset of C. Its boundary is a superset of the boundary of F.
Two paths with the same endpoints are **homotopic** if one can be continuously deformed into the other.

**R x S^1** example:

- \( t_1 \) and \( t_2 \) are homotopic.
- \( t_1 \) and \( t_3 \) are not homotopic.
- In this example, infinity of homotopy classes.
Connectedness of $C$-Space

$C$ is **connected** if every two configurations 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: $S^1$ and $SO(3)$ are multiply-connected:

- In $S^1$, infinity of homotopy classes
- In $SO(3)$, only two homotopy classes
Classes of Homotopic Free Paths
Example for Articulated Robot
Motion-Planning Framework

Continuous representation
(configuration space formulation)

Discretization

Graph searching
(blind, best-first, A*)
Path-Planning Approaches

1. **Roadmap**
   Represent the connectivity of the free space by a network of 1-D curves

2. **Cell decomposition**
   Decompose the free space into simple cells and represent the connectivity of the free space by the adjacency graph of these cells

3. **Potential field**
   Define a function over the free space that has a global minimum at the goal configuration and follow its steepest descent
Roadmaps // Retraction

Roadmap methods are also known as retraction methods.

This is based on the core mathematical relation (usually unstated) that roadmaps are based on a retraction mapping, or projection:

\[ f : \mathbb{R}^n \rightarrow \mathbb{R} \]

- Each point in \( \mathbb{R}^n \) maps into some point in the roadmap.
- Each point on the roadmap maps onto itself.
- The mapping is smooth almost everywhere:
  \[ d^n(f(a), f(b)) < k \, d(a, b) \]
Roadmap Methods

- **Visibility graph from**
  Introduced in the Shakey project at SRI in the late 60s. Can produce shortest paths in 2-D configuration spaces.
Roadmap Methods

- Visibility graph
- Voronoi diagram

Introduced by Computational Geometry researchers. Generate paths that maximizes clearance. Applicable mostly to 2-D configuration spaces.
Roadmap Methods

- Visibility graph
- Voronoi diagram
- Silhouette
  First complete general method that applies to spaces of any dimension and is singly exponential in # of dimensions [Canny, 87]
- Probabilistic roadmaps
Path-Planning Approaches

1. **Roadmap**
   Represent the connectivity of the free space by a network of 1-D curves

2. **Cell decomposition**
   Decompose the free space into simple cells and represent the connectivity of the free space by the adjacency graph of these cells

3. **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

Two families of methods:

- **Exact cell decomposition**
  The free space $F$ is represented by a collection of non-overlapping cells whose union is exactly $F$

*Examples: trapezoidal and cylindrical decompositions*
Trapezoidal decomposition
Two families of methods:

- **Exact cell decomposition**
- **Approximate cell decomposition**

F is represented by a collection of non-overlapping cells whose union is contained in F.

**Examples:** quadtree, octree, $2^n$-tree
Octree Decomposition
Path-Planning Approaches

1. **Roadmap**
   Represent the connectivity of the free space by a network of 1-D curves

2. **Cell decomposition**
   Decompose the free space into simple cells and represent the connectivity of the free space by the adjacency graph of these cells

3. **Potential field**
   Define a function over the free space that has a global minimum at the goal configuration and follow its steepest descent
Potential Field Methods

- Approach initially proposed for real-time collision avoidance [Khatib, 86]. Hundreds of papers published on it.

Path planning:
- Regular grid $G$ is placed over $C$-space
- $G$ is searched using a best-first algorithm with potential field as the heuristic function
Potential Field Methods

- Approach initially proposed for real-time collision avoidance [Khatib, 86]. Hundreds of papers published on this topic.
- **Potential field**: Scalar function over the free space
- **Ideal field** (**navigation function**): Smooth, global minimum at the goal, no local minima, grows to infinity near obstacles
- Force applied to robot: Negated gradient of the potential field. Always move along that force