CONFIGURATION SPACE

Configuration : $q \in C$

Everything we need to describe where the robot is now.

$A(q) : C \rightarrow W \rightarrow \text{world} = R^2 \text{ or } R^3$

Example: $q = (x, y) \in R^2$ $A(q) =$ 

$q = (x, y, \theta) \in SE(2)$ $A(q) =$ 

$q = (\theta_1, \theta_2) \in S^2$ $A(q) =$ 

$q = (x, y, z, R) \in SE(3)$ $A(q) =$ 

C: Configuration Space

Any q in C is a point in C

Define Obstacle Region $O \subset W$

$$C_{obs} = \{q \in C \mid A(q) \cap O \neq \emptyset\}$$

$$C_{free} = C \setminus C_{obs}$$

Technical note: C_{obs} is a closed set

C_{obs} includes O and $A(q)$ touching

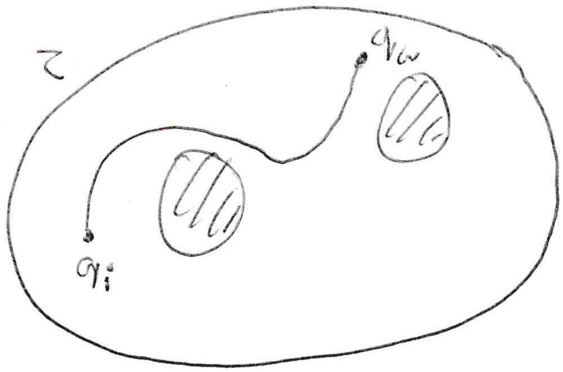
$\Rightarrow$ C_{free} is an open set

$\Rightarrow$ No touching

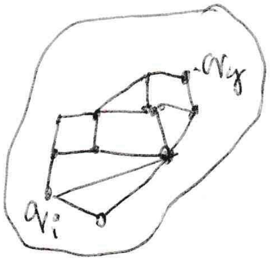
If we want to allow touching, then ?

PIANO MOVERS' PROBLEM

1. A world $W = \mathbb{R}^2$ or $\mathbb{R}^3$
2. An obstacle region $O \subset W$
3. A robot A and its C-space C
4. $C_{free} = C \setminus C_{obs}$
5. $q_i \in C_{free}$ INITIAL CONFIG
6. $q_g \in C_{free}$ GOAL CONFIG
7. Compute a path $\tau : [0,1] \rightarrow C_{free}$ such that $q_i = \tau(0)$, $q_g = \tau(1)$



GRAPH-BASED ALGORITHMS



1. Create a graph $G(V, E)$ — V → vertices, E → edges
2. Plan a path in G.

IMPLICIT vs. EXPLICIT GRAPH

- IMPLICIT → Graph only accessible via succ function (succ → successor)
- EXPLICIT → "roadmaps"; V, E available a-priori

$$succ(u) = \{(u,v) \mid (u,v) \in E\}$$

KEY CHALLENGE: Collision checking > 90% of planning effort

KEY IDEA: Perform edge evaluations LAZILY

$$c(e) = dist(u,v) \text{ if not in collision}$$
$$= \infty \text{ if in collision}$$

↑ Cost of an edge