

CSE 478 Robot Autonomy

Motion Planning

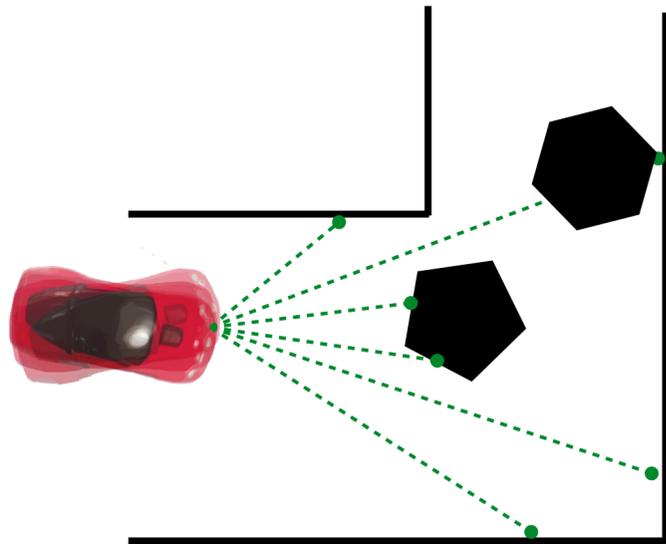
Abhishek Gupta (abhgupta@cs)
Siddhartha Srinivasa (siddh@cs)

TAs:

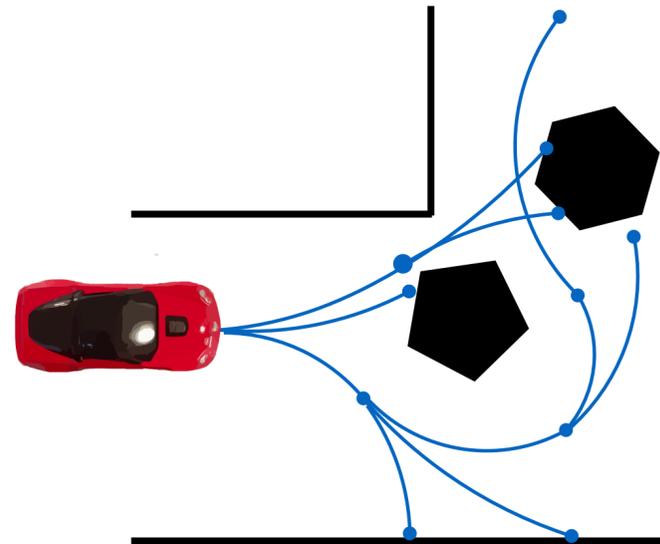
Carolina Higuera (chiguera@cs)
Rishabh Jain (jrishabh@cs)
Entong Su (ensu@cs)



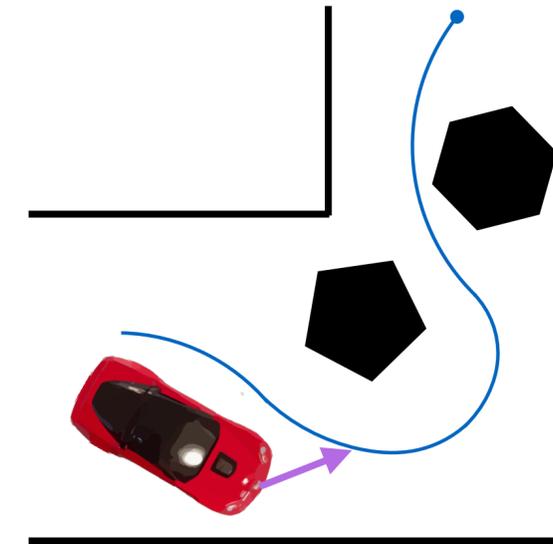
**Estimate
state**

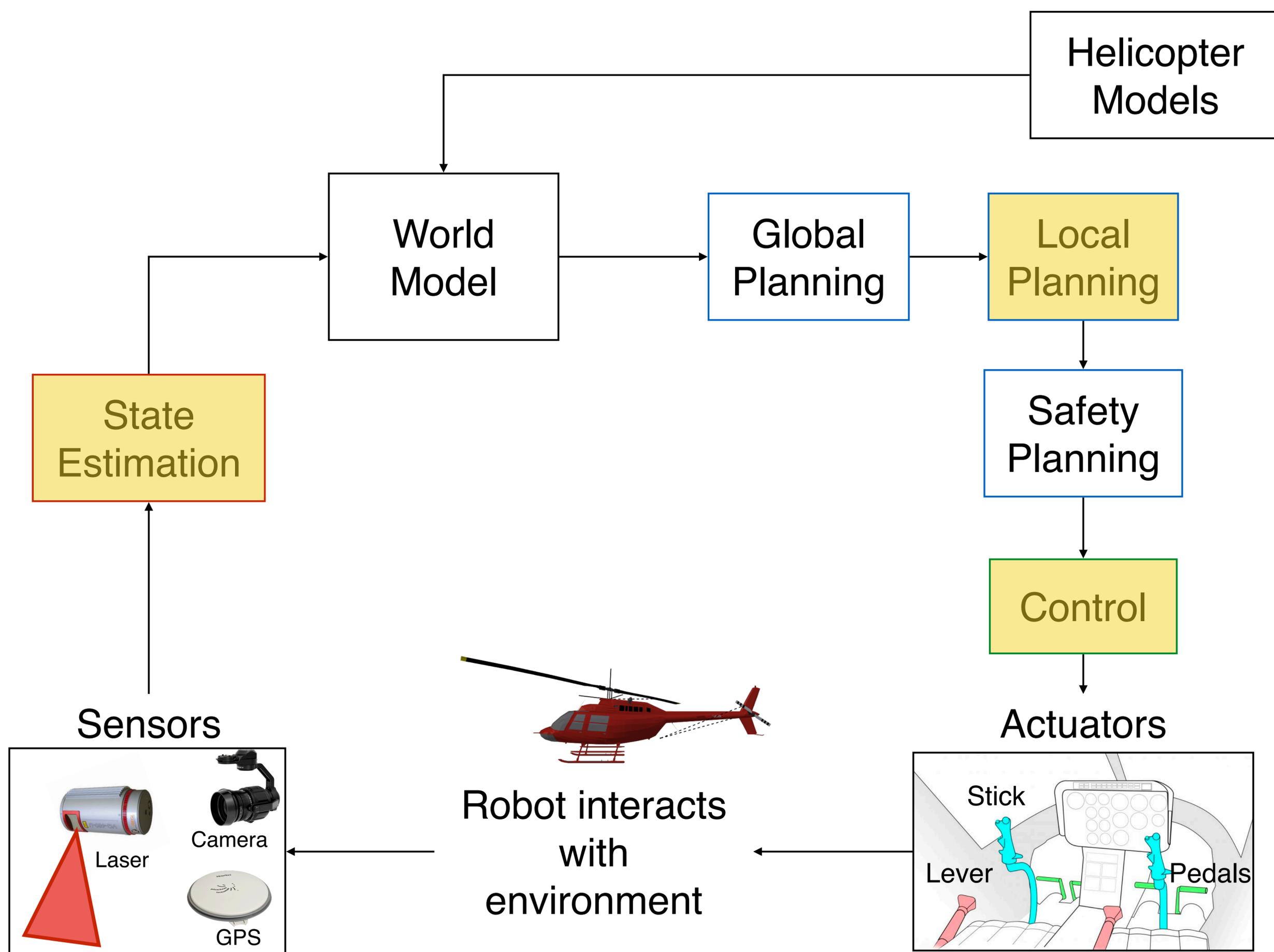


**Plan a
sequence of
motions**



**Control
robot to
follow plan**





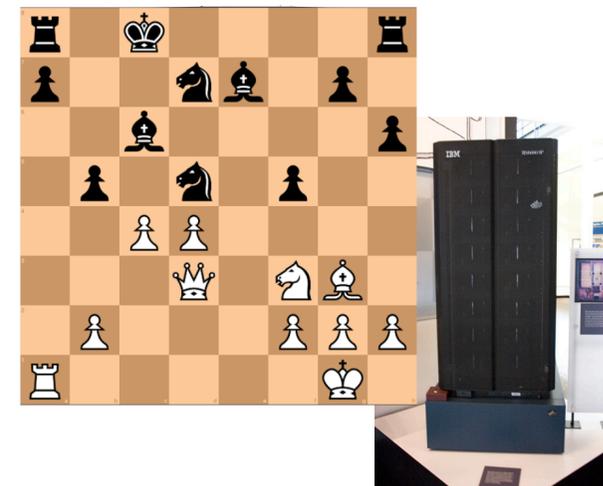
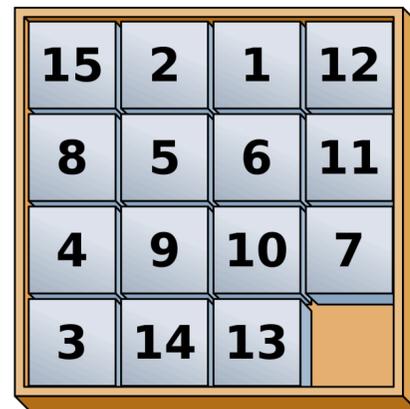
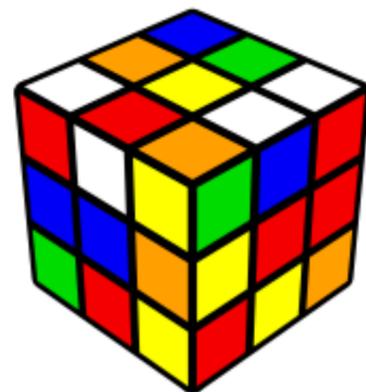
Flying from Seattle to LA?

- High-level sequence of actions: get to the terminal, board an airplane, etc.
- If we have a detailed plan to get to the terminal (and some idea of how to check in and board), should we also plan our route through LAX?
- Rental car? Lodging? What future problems should we solve now?
- **How do I get out of my house?**

(EXAMPLE FROM LESLIE KAEHLING)

What Makes (Motion) Planning Difficult?

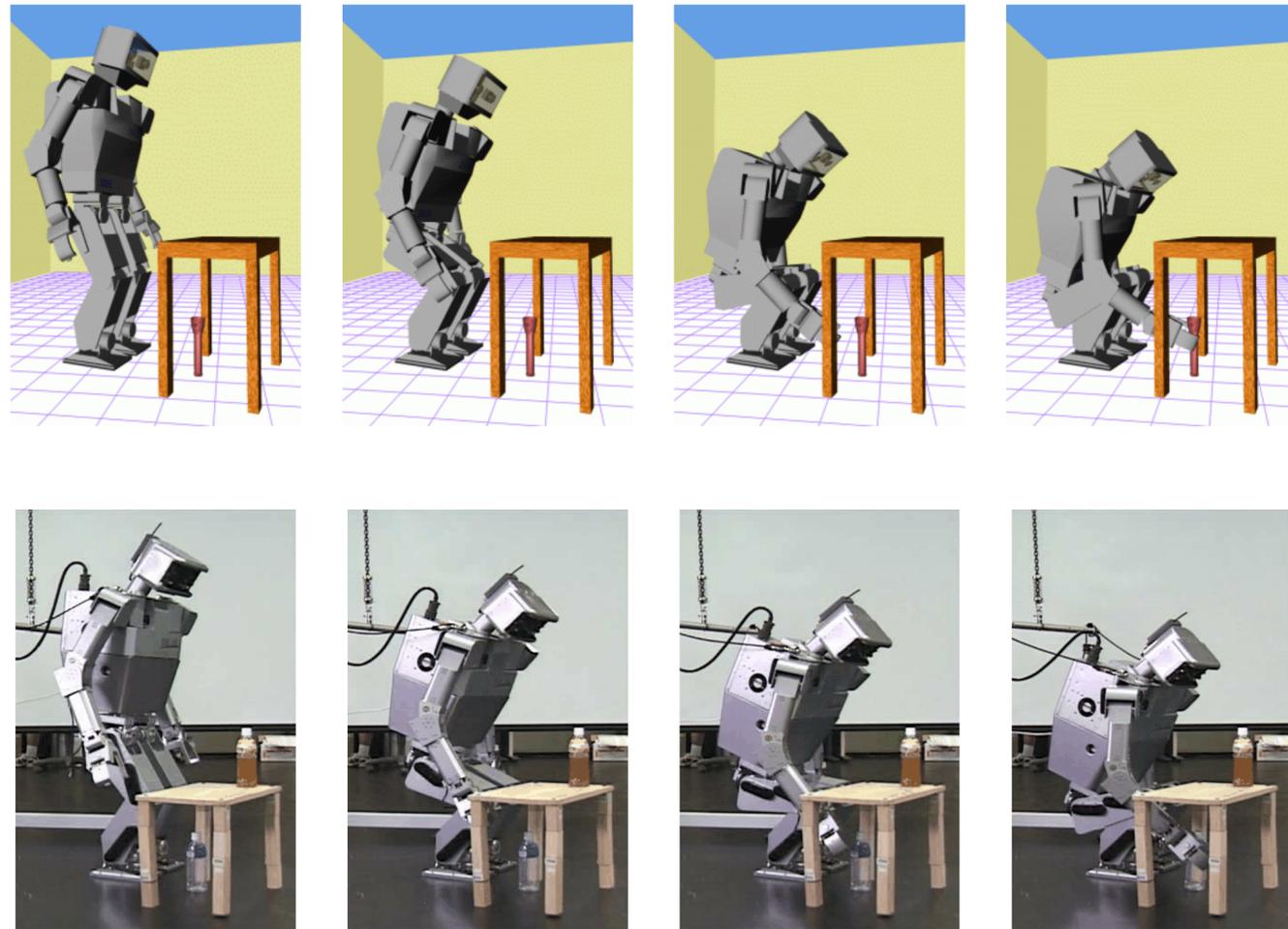
- Classic AI planning problems: Rubik's cube, sliding-tile puzzle, chess
 - Discrete state space, strictly-defined rules, humans have great intuition
 - Developed many of the tools that are still used today!
- (Some) challenges in motion planning: continuous state space, expensive action simulation, robot model uncertainty, nonholonomic constraints



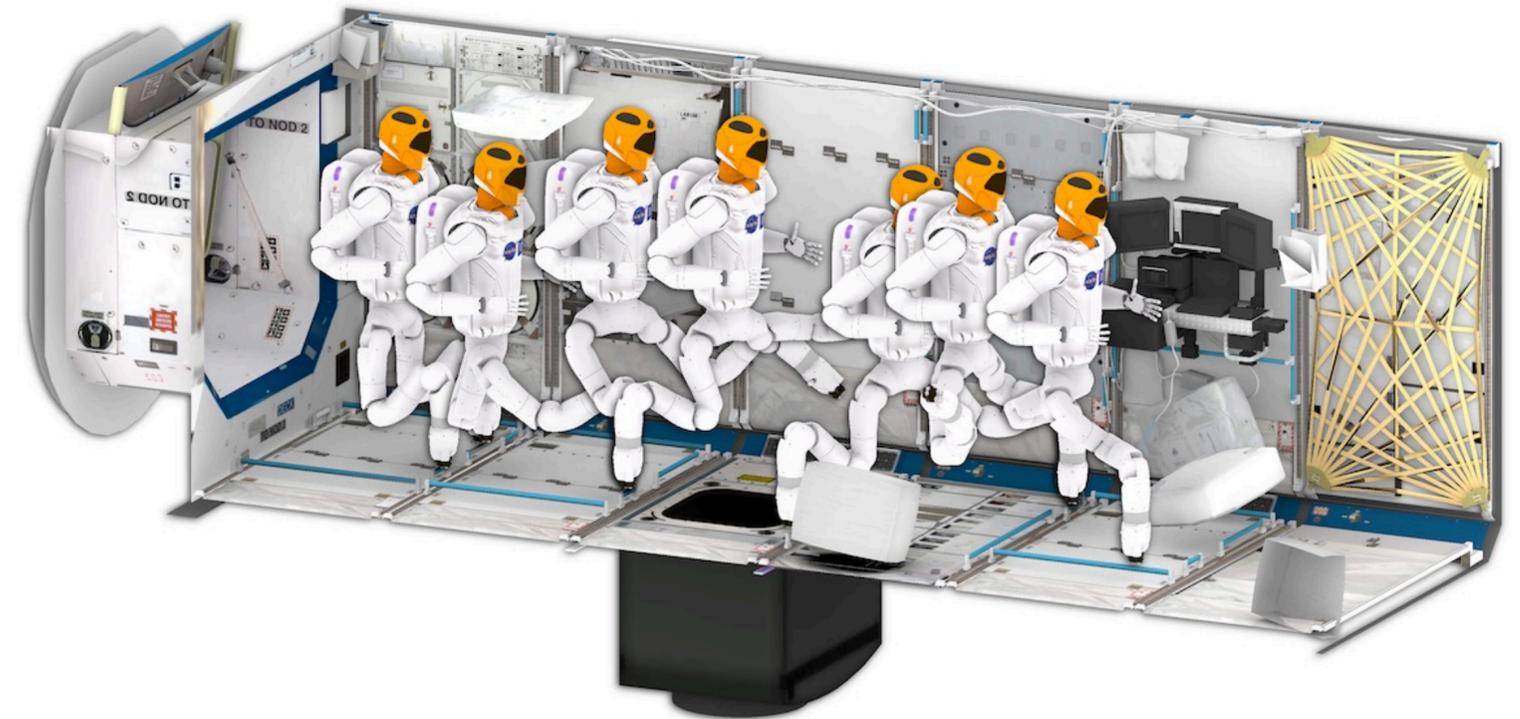
The Piano Mover's Problem



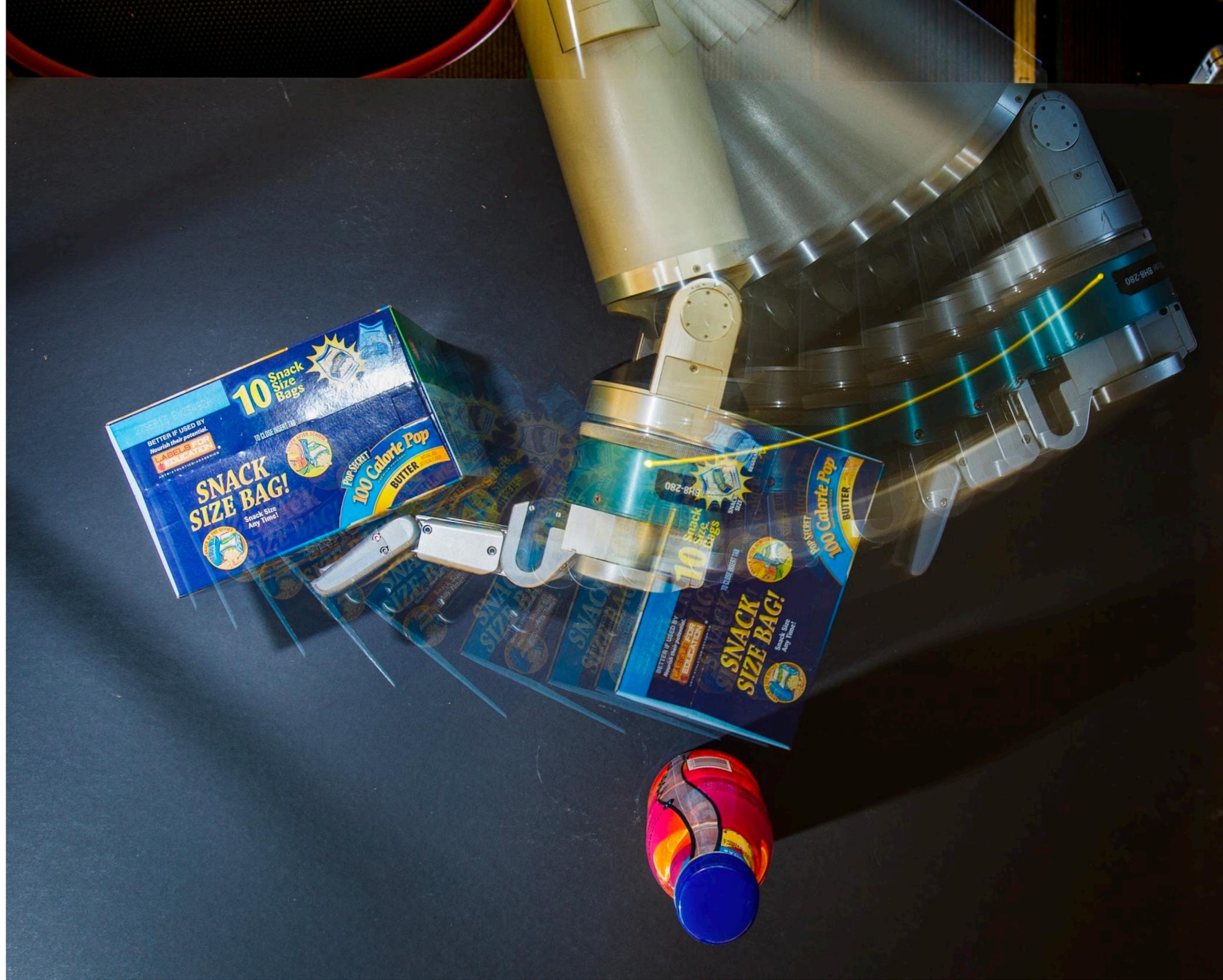
High-Dimensional Planning



HONDA H7 HUMANOID ROBOT
KUFFNER ET AL., 2003



NASA R2 HUMANOID ROBOT
KINGSTON ET AL., 2019



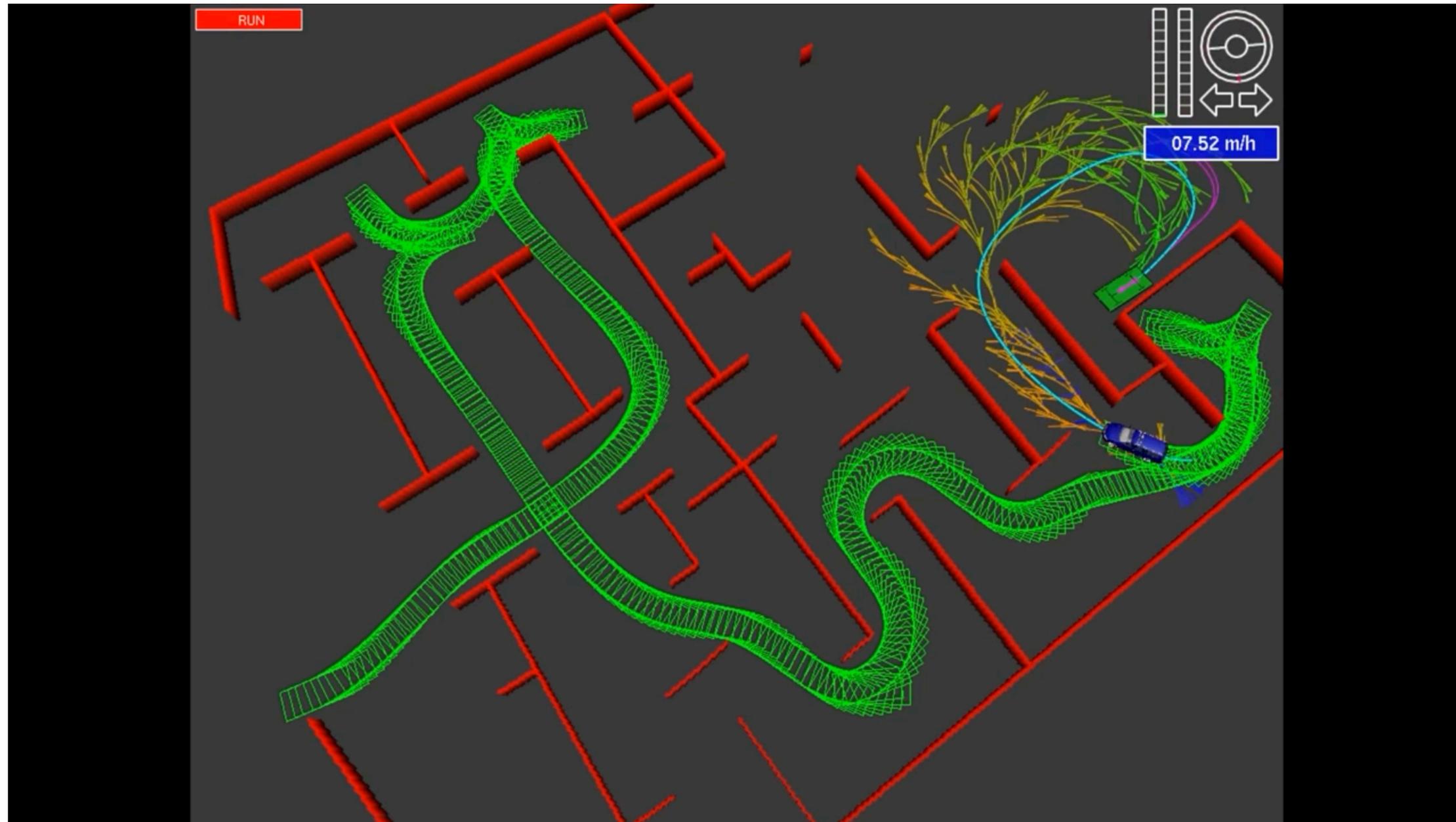
MANIPULATION PLANNING, HERB

Real-time planning



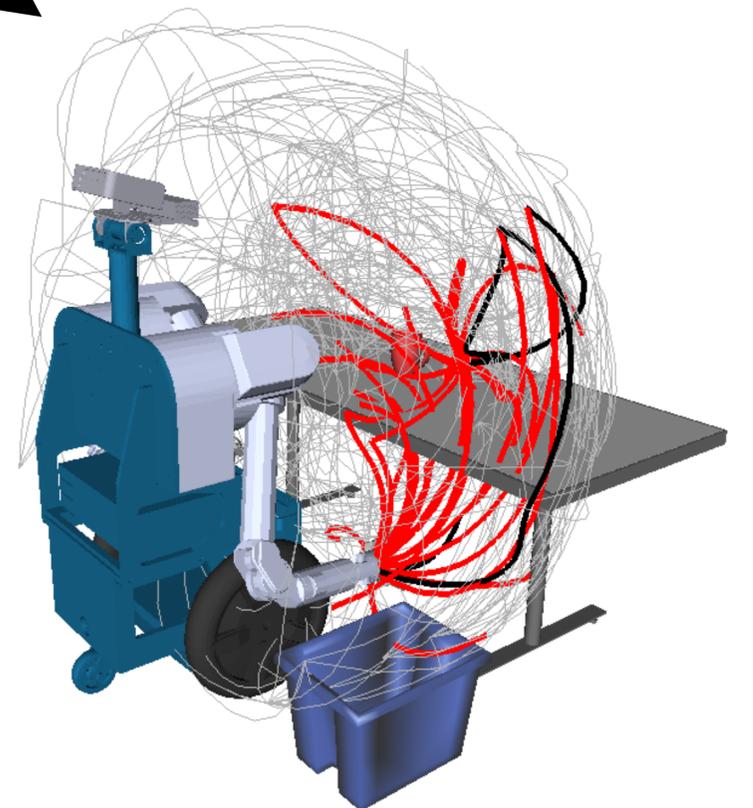
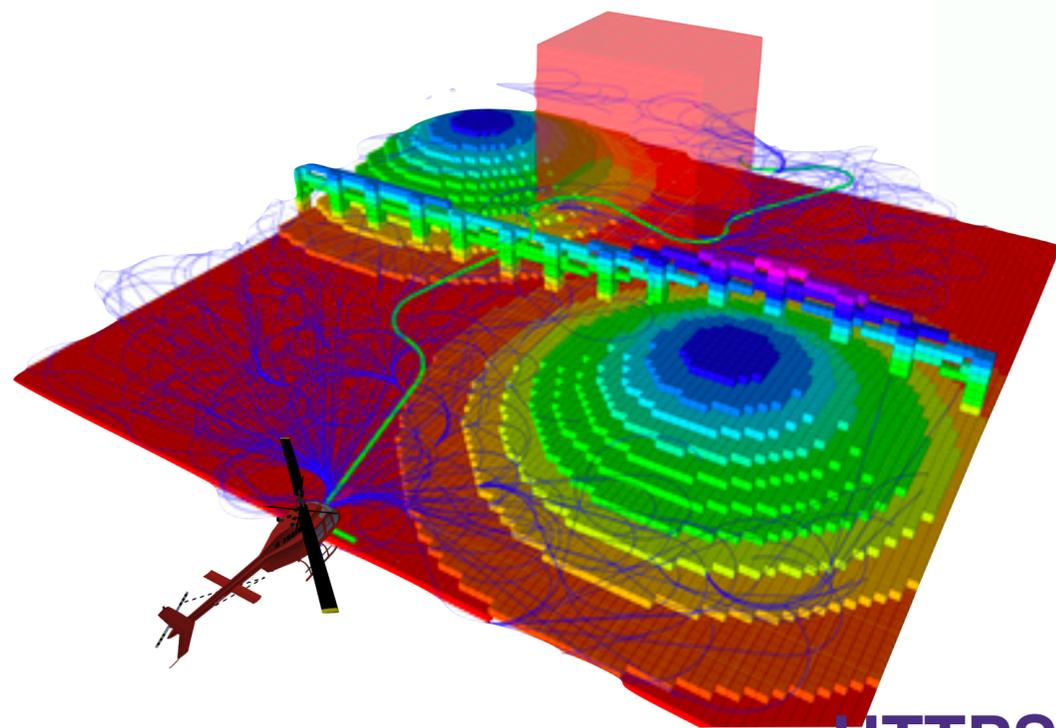
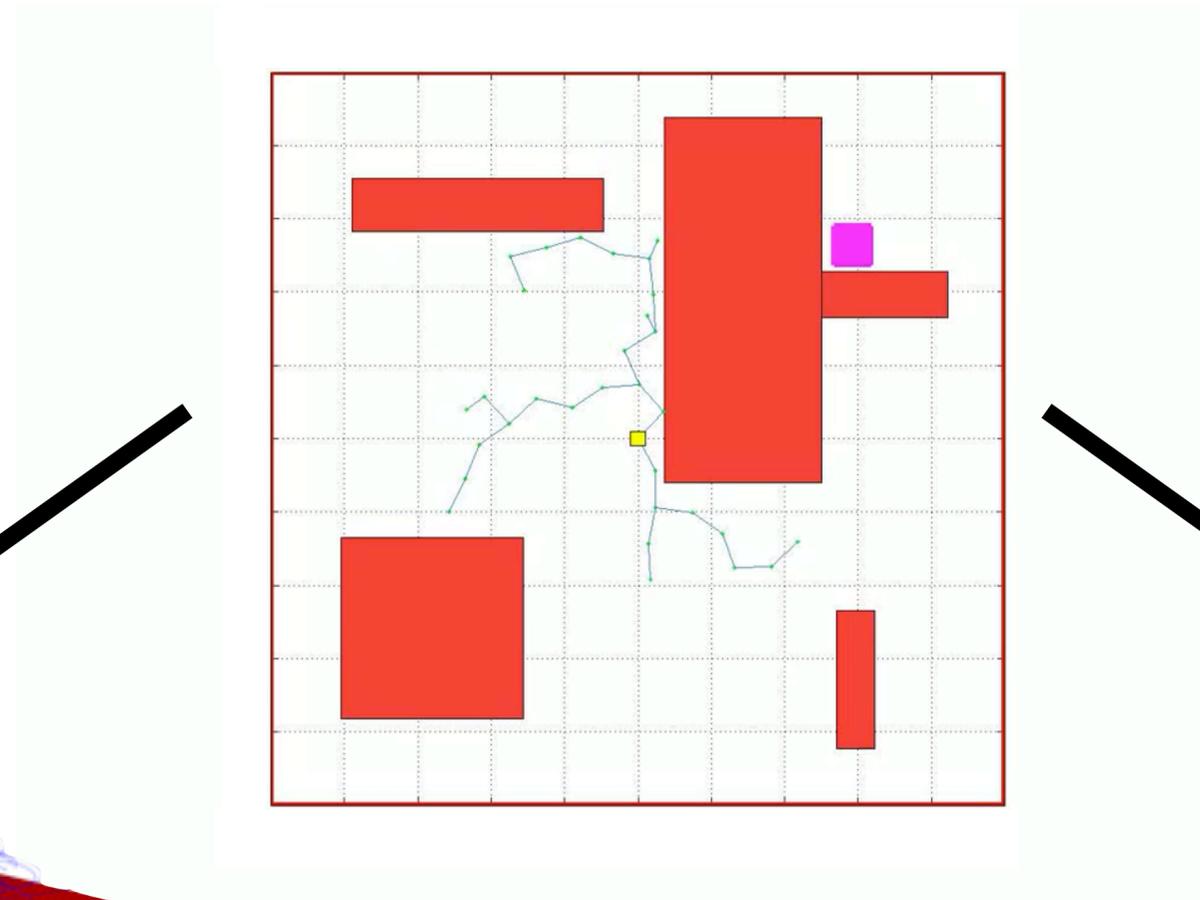
Willow garage, 2009

Real-Time Planning



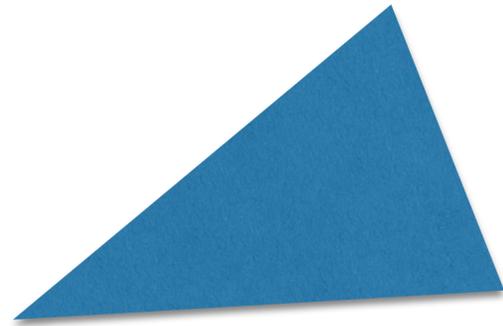
[HTTPS://YOUTU.BE/QXZT-B7IUYW](https://youtu.be/QXZT-B7IUYW)

Algorithms that Generalize

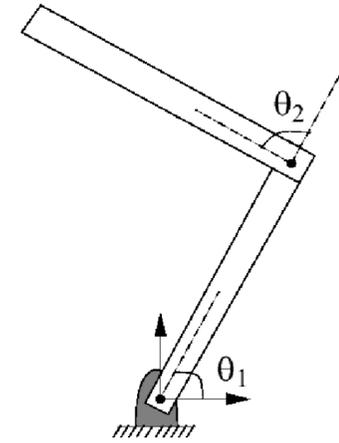


[HTTPS://YOUTU.BE/FAFW8DOKVIK](https://youtu.be/FAFW8DOKVIK)

Representations that Generalize



(a) Translating Triangle



(b) 2-joint planar arm



(c) Racecar



(d) Manipulator

The Configuration Space

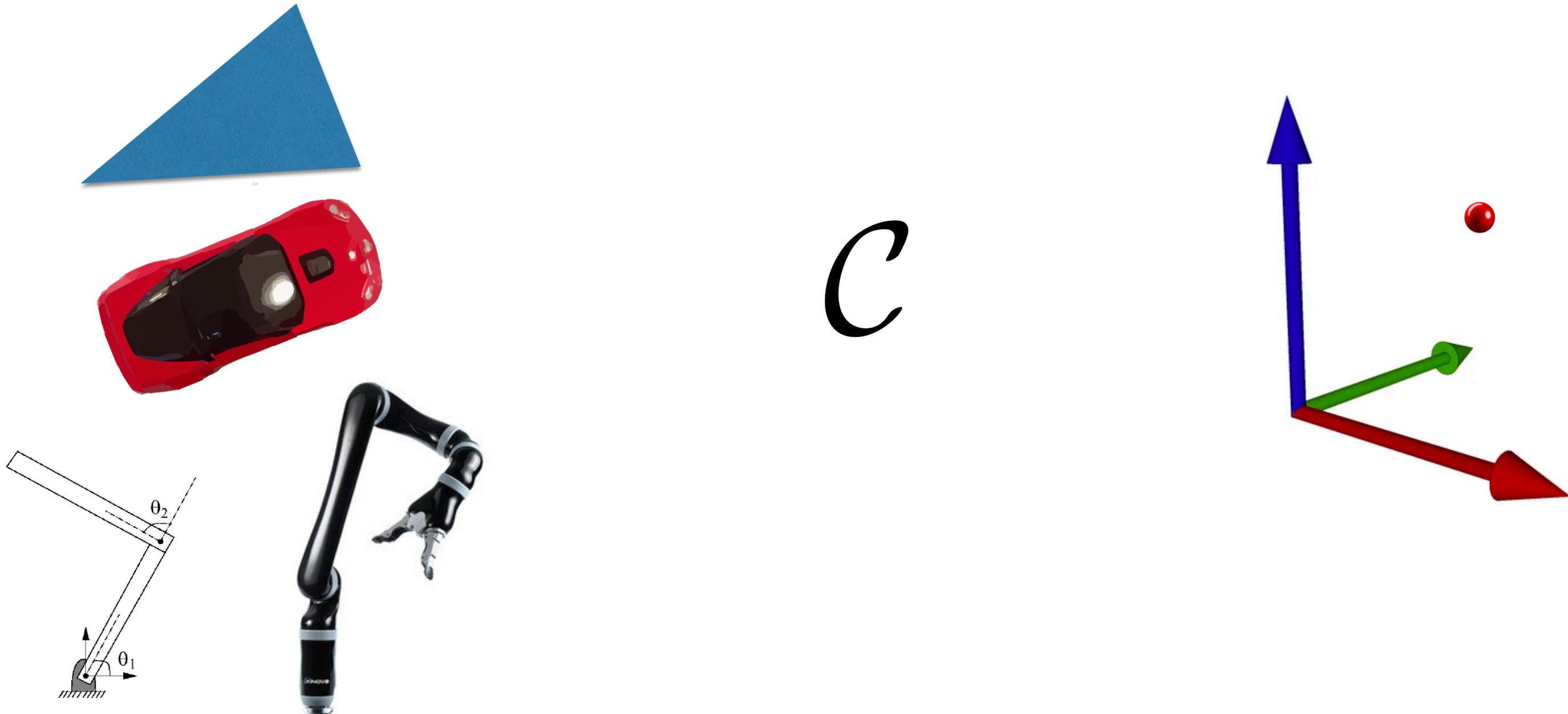
The configuration space or C-space is the **manifold** that contains the set of transformations achievable by the robot.

\mathcal{C}

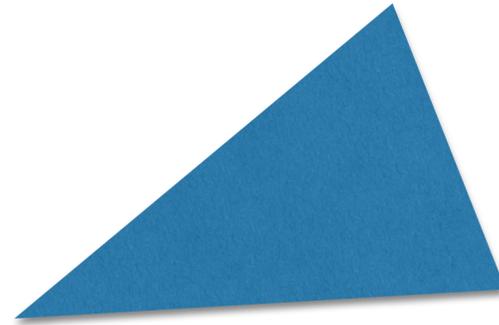
Complete specification of the
location of every point on robot geometry

Key Insight

Represent the robot as a point in some high-dimensional space



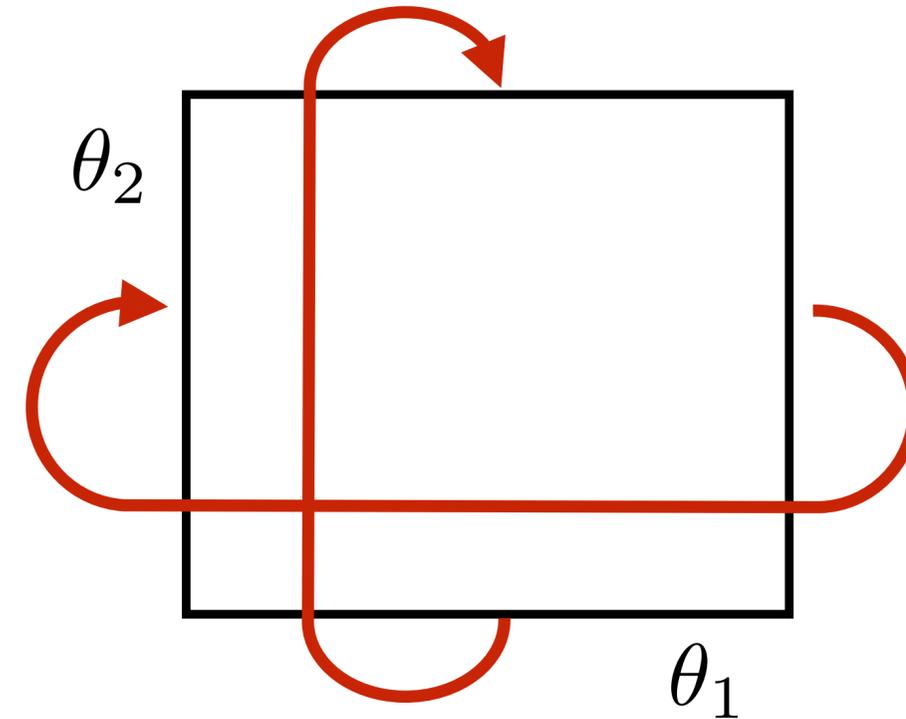
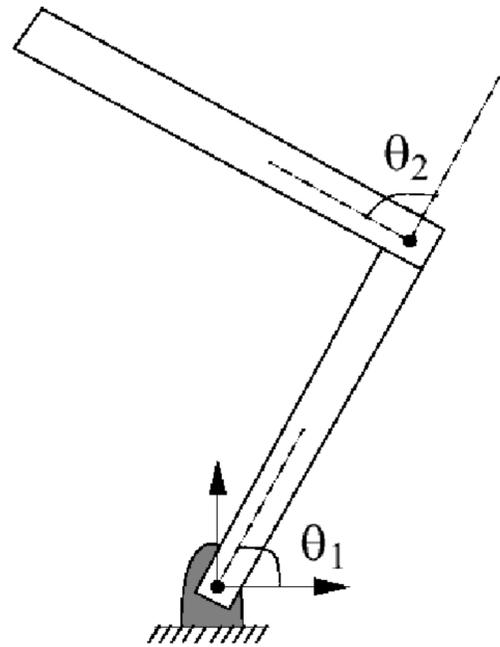
Example 1: Translating triangle



$$\mathbb{R} \times \mathbb{R} = \mathbb{R}^2$$

(cartesian product)

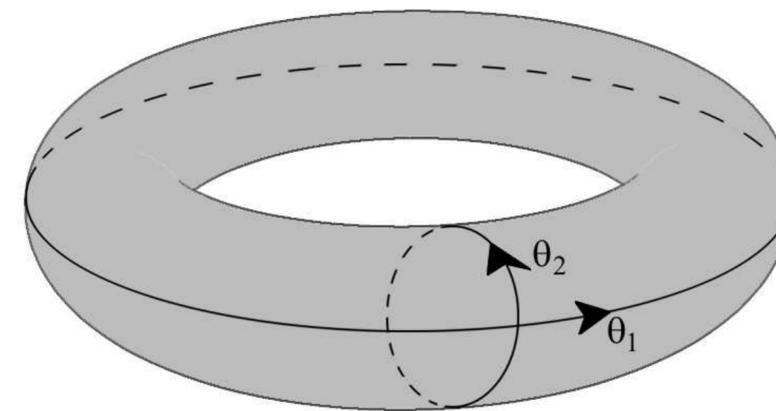
Example 2: 2-joint planar arm



$$S^1 \times S^1 = T^2$$

Circle

$$S^1 = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 = 1\}.$$



Example 3: Racecar



$$\mathbb{R}^2 \times S^1$$

special euclidean group $SE(2)$

Common C-spaces

Type of Robot	C-space Representation
Mobile robot translating in the plane	\mathbb{R}^2
Mobile robot translating and rotating in the plane	$SE(2)$ or $\mathbb{R}^2 \times S^1$
Rigid body translating in the three-space	\mathbb{R}^3
A spacecraft	$SE(3)$ or $\mathbb{R}^3 \times SO(3)$
An n -joint revolute arm	T^n
A planar mobile robot with an attached n -joint arm	$SE(2) \times T^n$

(Kavraki and LaValle)

Obstacles

Obstacle specification

Robot operates in a 2D / 3D workspace $\mathcal{W} = \mathbb{R}^2$ or \mathbb{R}^3

Subset of this space is obstacles $\mathcal{O} \subset \mathcal{W}$

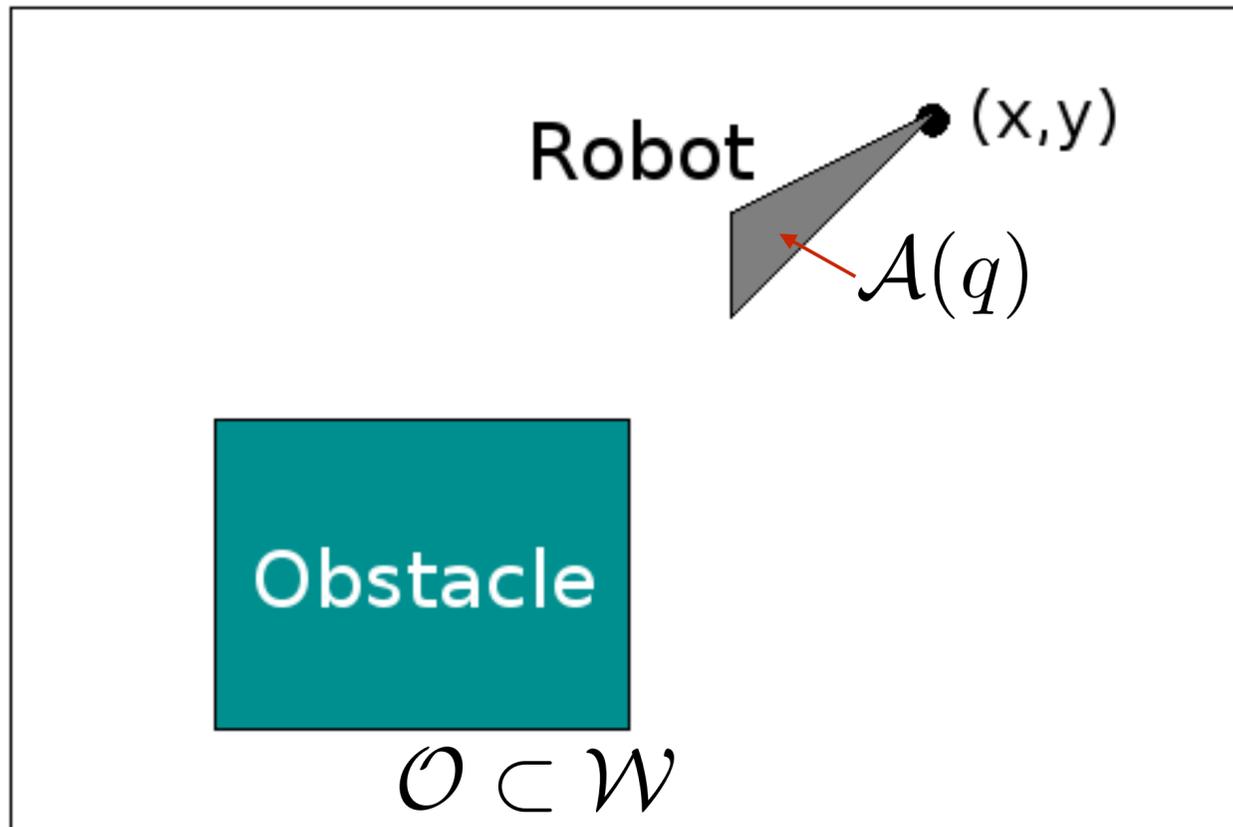
semi-algebraic models (polygons, polyhedra)

Geometric shape of the robot
(set of points occupied by robot at a config) $\mathcal{A}(q) \subset \mathcal{W}$

C-space obstacle region $\mathcal{C}_{obs} = \{q \in \mathcal{C} \mid \mathcal{A}(q) \cap \mathcal{O} \neq \emptyset\}$

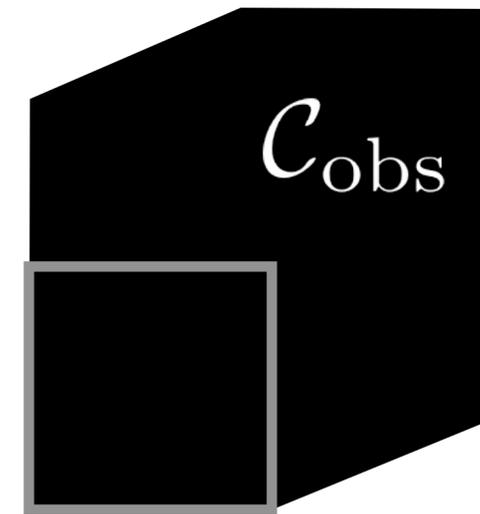
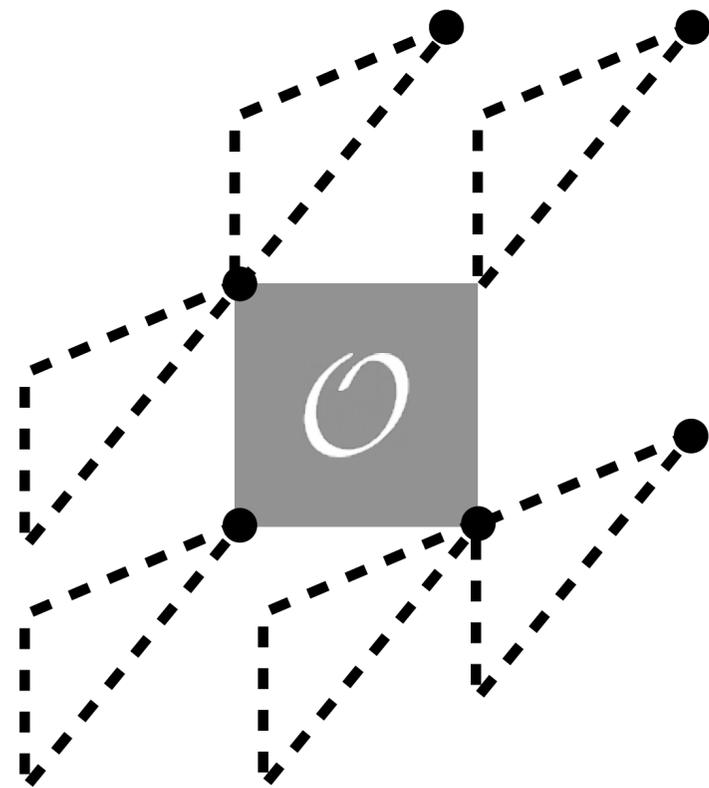
$$\mathcal{C}_{free} = \mathcal{C} \setminus \mathcal{C}_{obs}$$

Example 1: Translating triangle



Can be efficiently computed using Minkowski sum

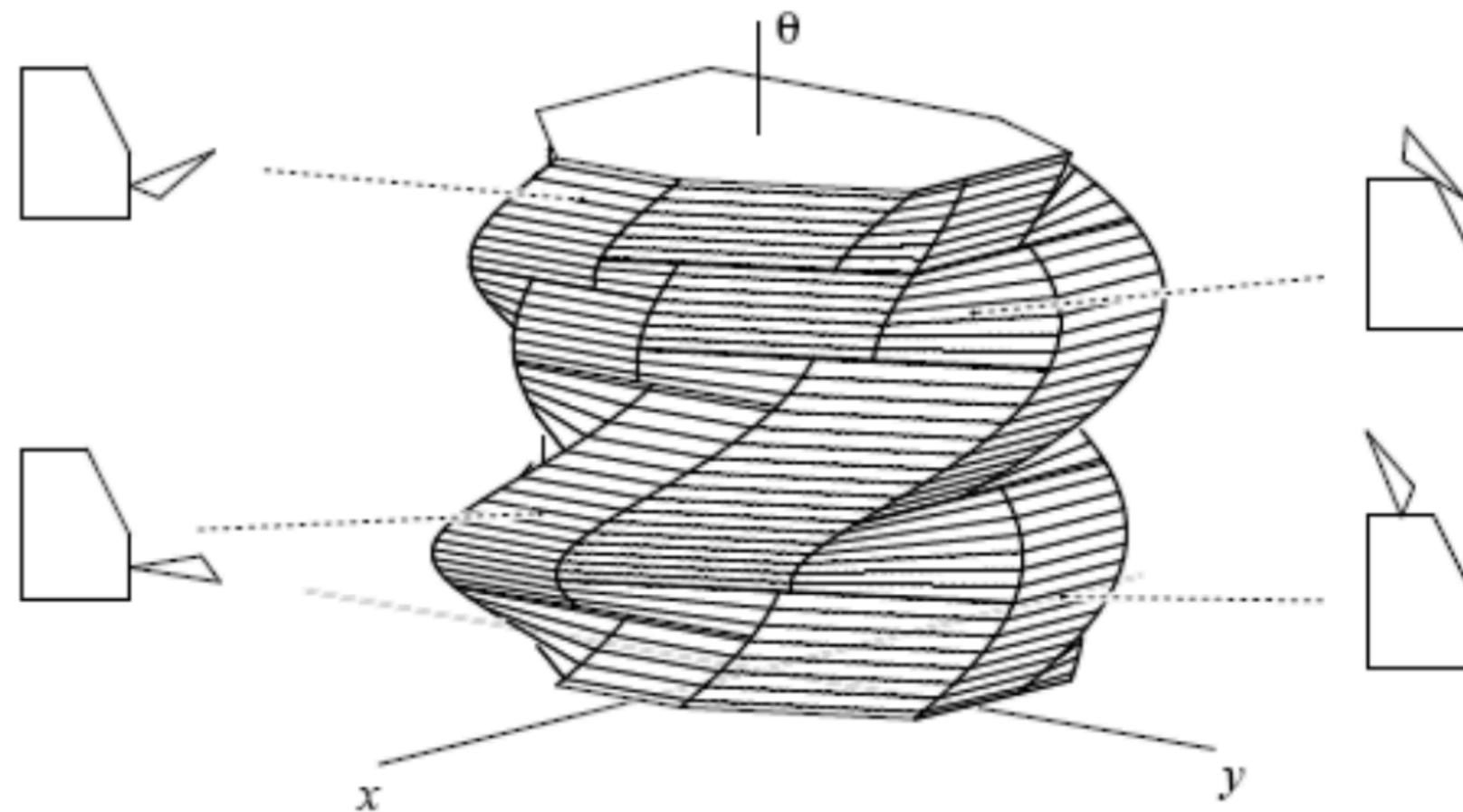
Example: Translating Triangle in Plane



Can be computed for
convex polygons
(Minkowski sum)

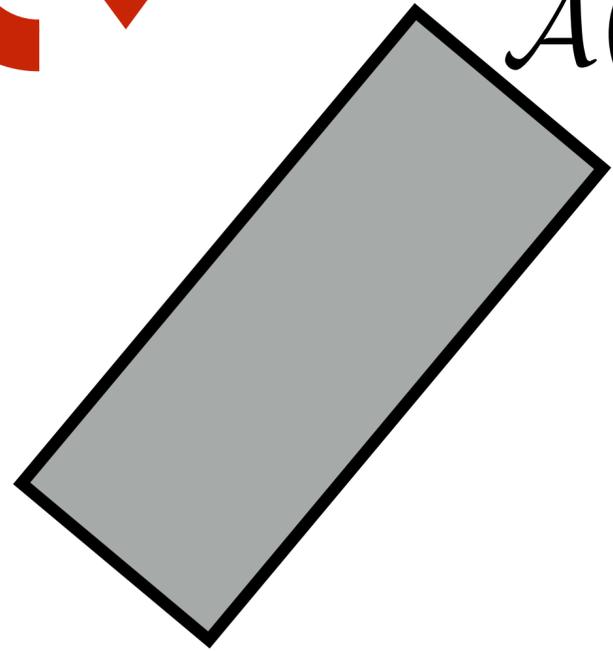
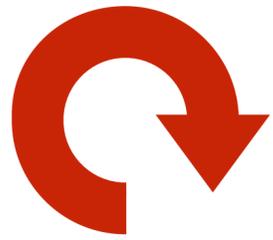
(EXAMPLE FROM LYDIA KAVRAKI AND STEVEN LAVALLE)

Example: Translating and Rotating Triangle

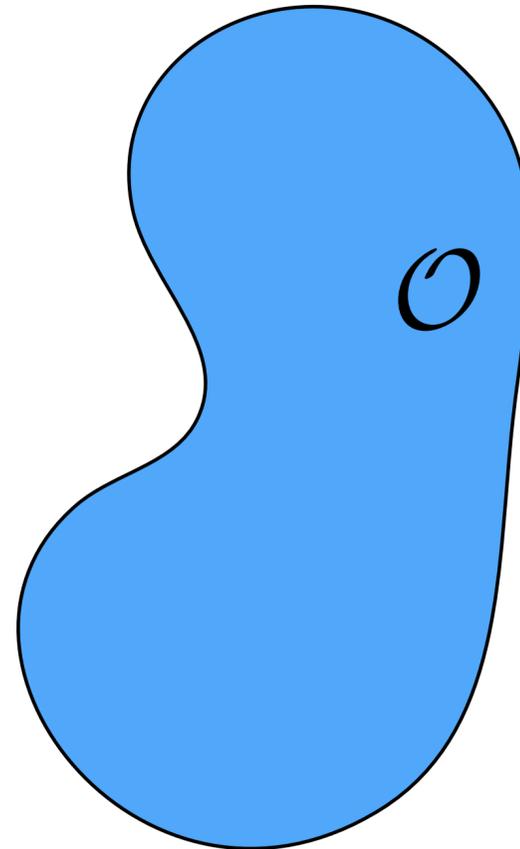


(EXAMPLE FROM HOWIE CHOSET)

Example 2: SE(2) robot

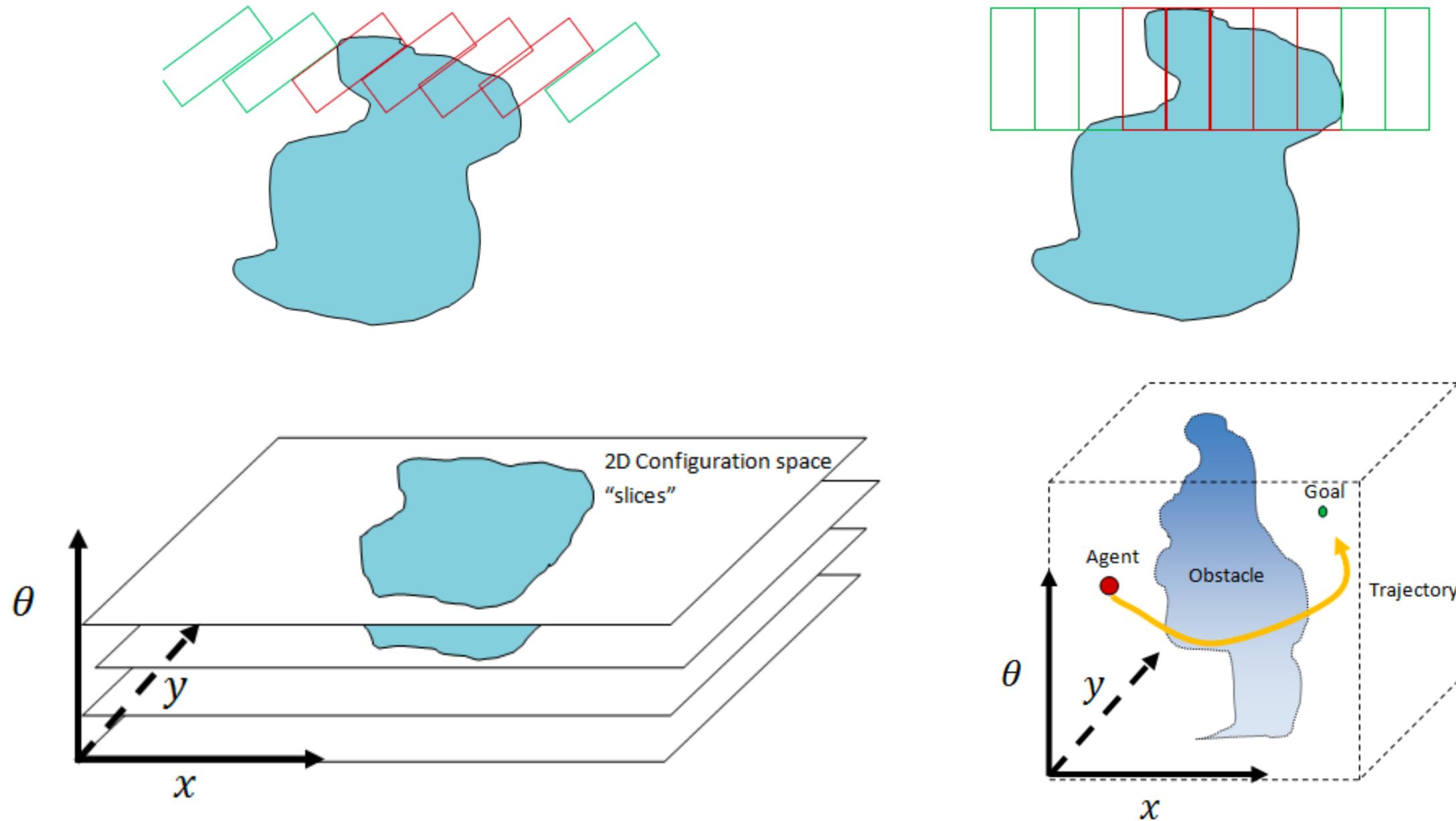


$$A(q) \subset \mathcal{W}$$



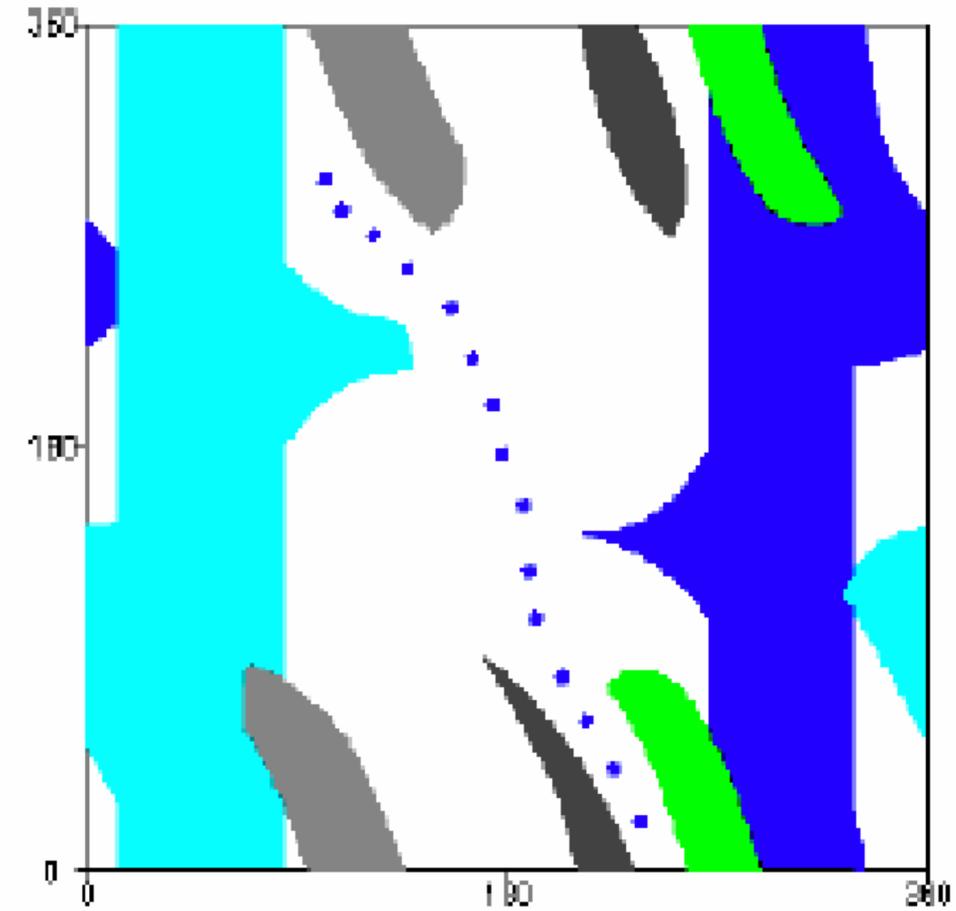
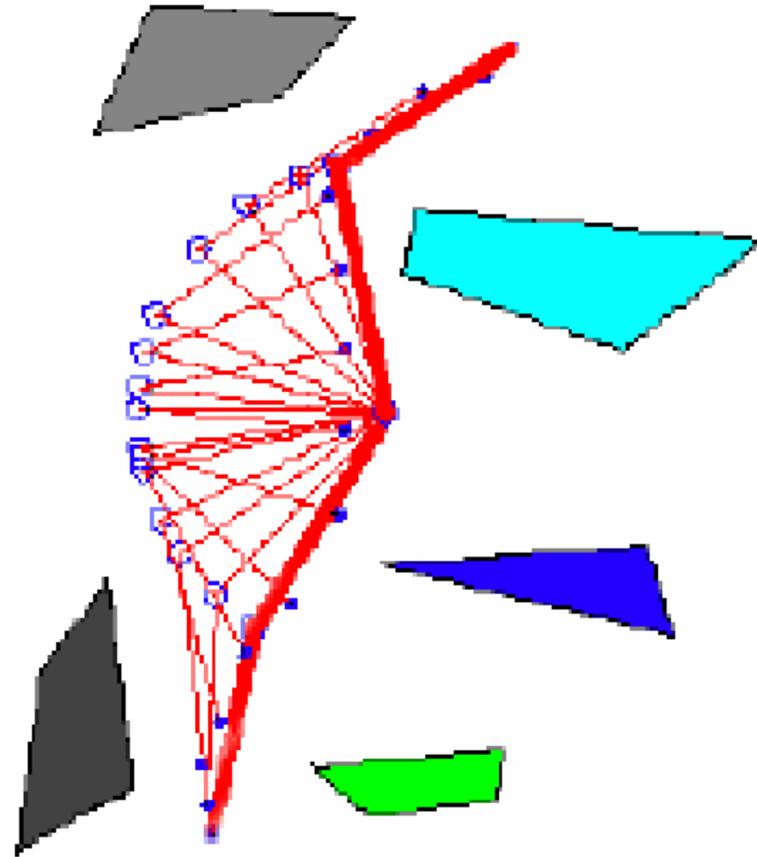
$$\mathcal{O} \subset \mathcal{W}$$

Example 2: SE(2) robot

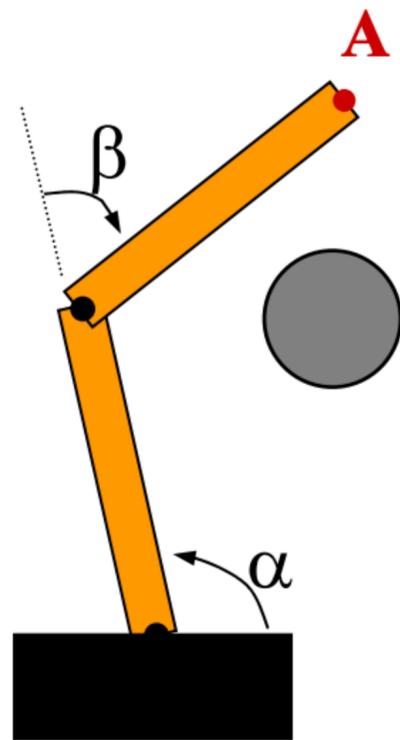


(Courtesy Matt Klingensmith)

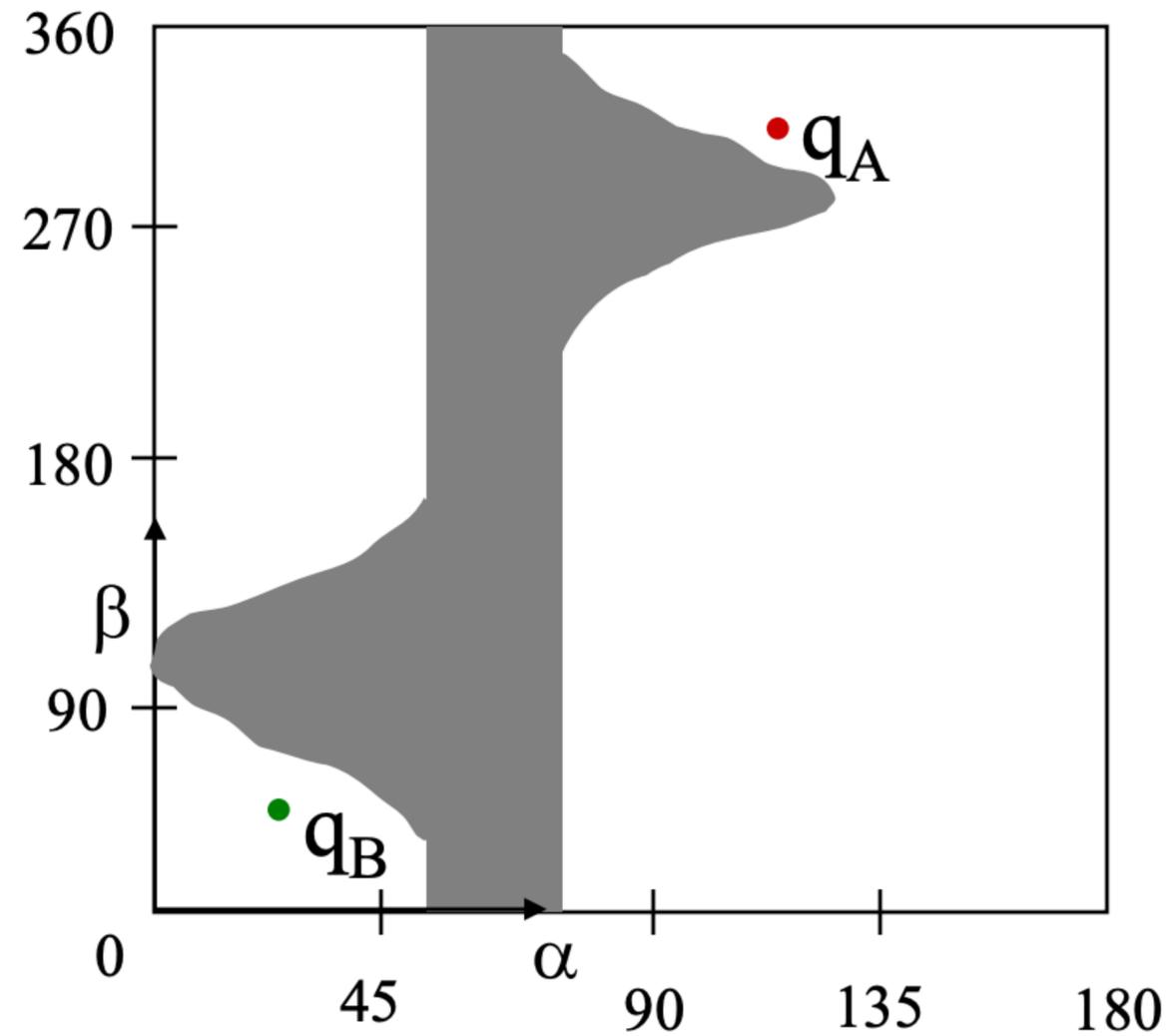
Example 3: 2-link planar arm



Example 3: 2-link planar arm

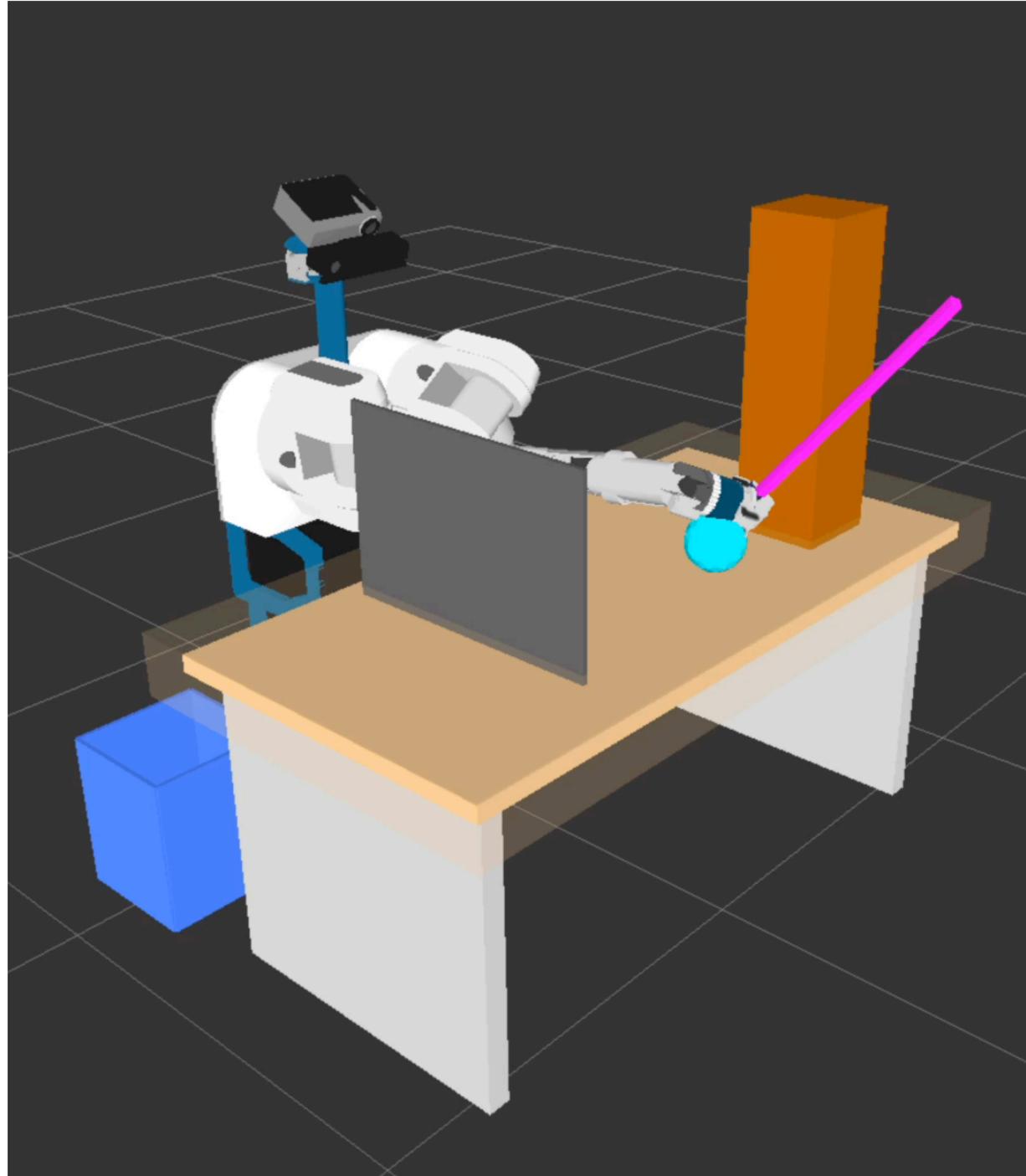


B



Geometric Path Planning Problem

Geometric Path Planning Problem



Also known as
Piano Mover's Problem (Reif 79)

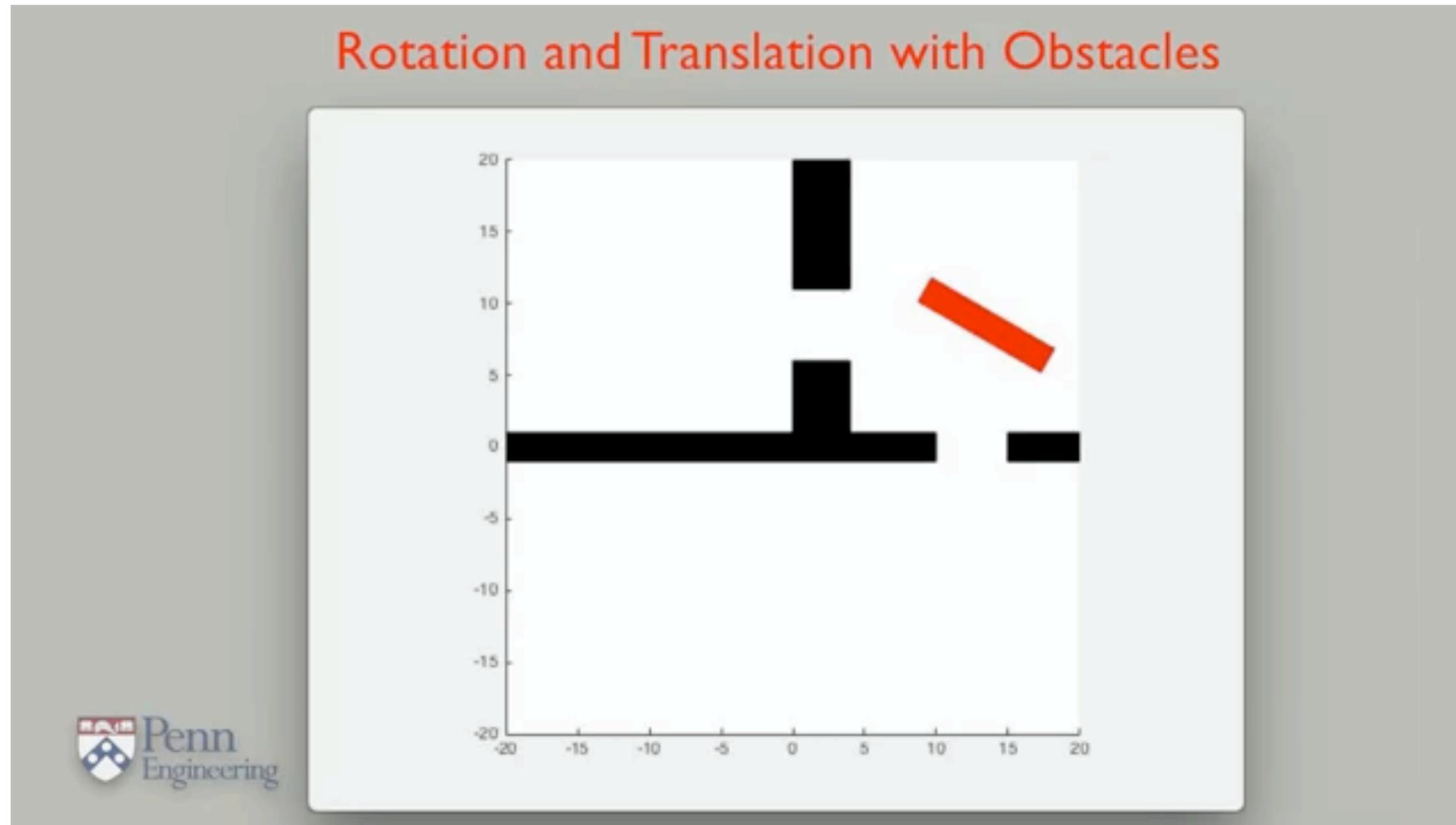
Given:

1. A *workspace* \mathcal{W} , where either $\mathcal{W} = \mathbb{R}^2$ or $\mathcal{W} = \mathbb{R}^3$.
2. An *obstacle region* $\mathcal{O} \subset \mathcal{W}$.
3. A *robot* defined in \mathcal{W} . Either a rigid body \mathcal{A} or a collection of m links: $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_m$.
4. The *configuration space* \mathcal{C} (\mathcal{C}_{obs} and \mathcal{C}_{free} are then defined).
5. An *initial configuration* $\mathbf{q}_I \in \mathcal{C}_{free}$.
6. A *goal configuration* $\mathbf{q}_G \in \mathcal{C}_{free}$. The initial and goal configuration are often called a *query* $(\mathbf{q}_I, \mathbf{q}_G)$.

Compute a (continuous) path, $\tau : [0, 1] \rightarrow \mathcal{C}_{free}$, such that $\tau(0) = \mathbf{q}_I$ and $\tau(1) = \mathbf{q}_G$.

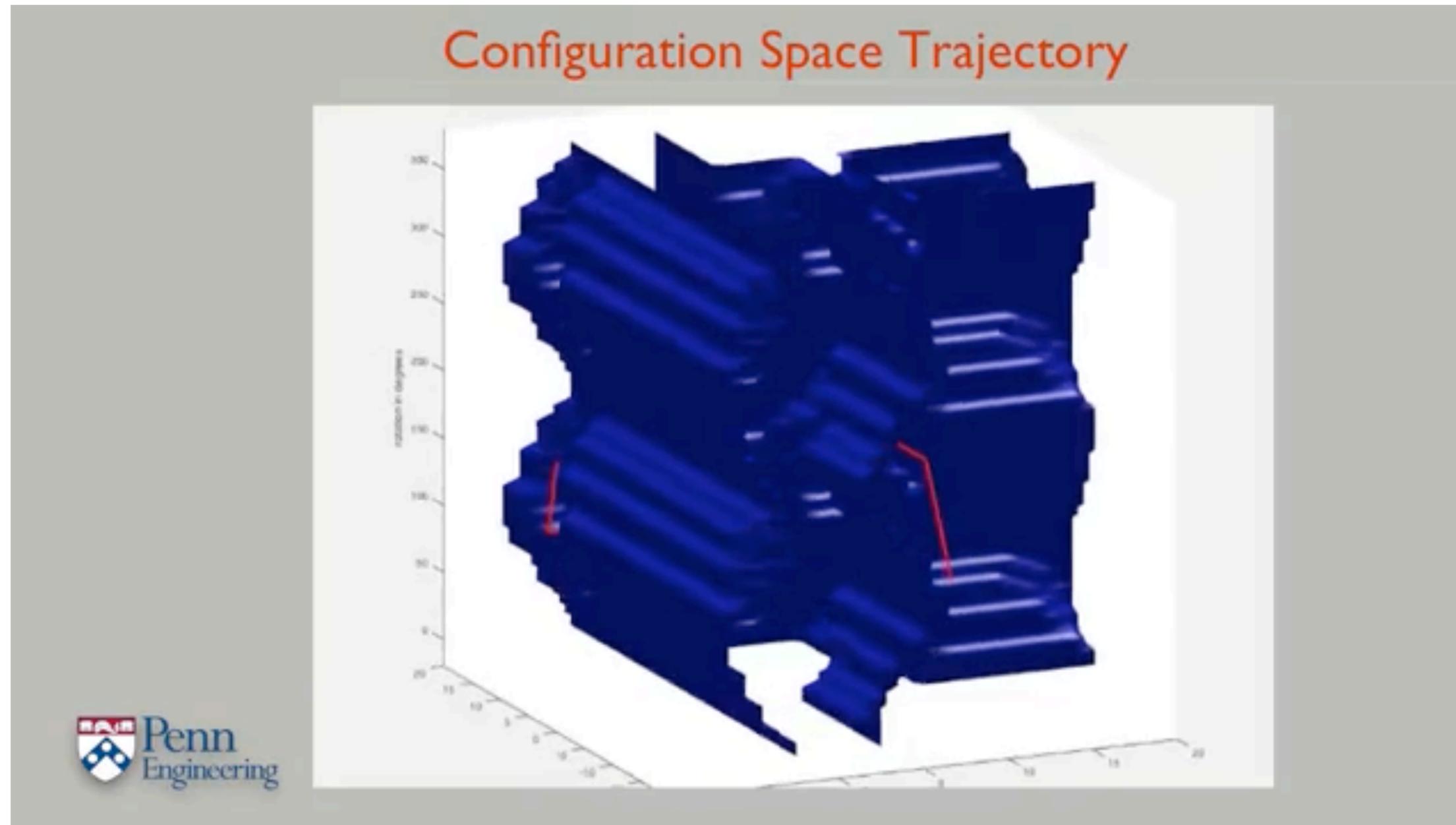
Also may want to minimize cost
 $c(\tau)$

Planning in Configuration Space



(EXAMPLE FROM CJ TAYLOR)

Planning in Configuration Space



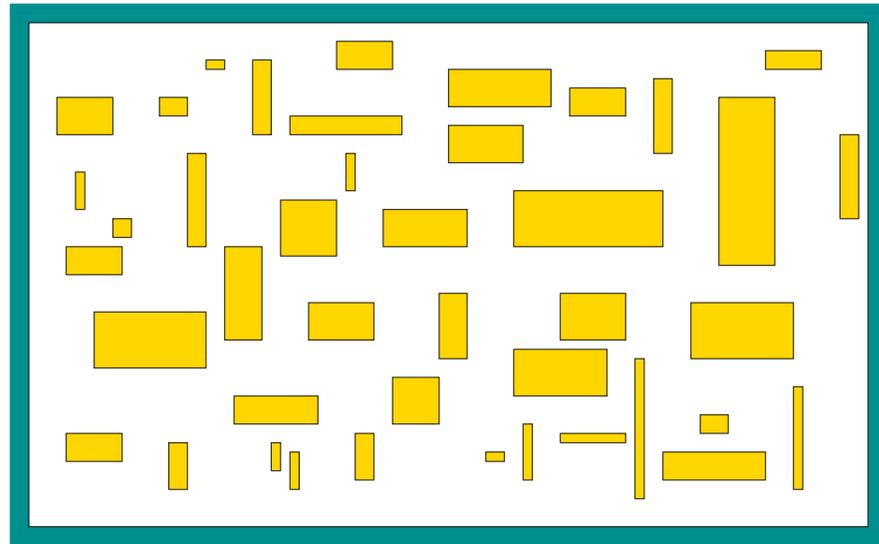
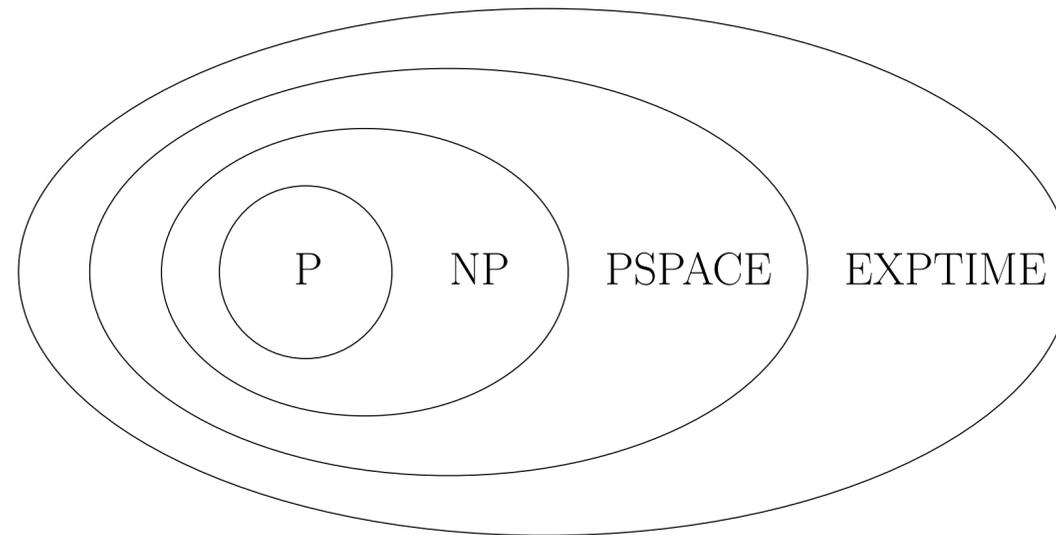
(EXAMPLE FROM CJ TAYLOR)

So, are we done?

No! Planning is **hard**

Hardness of motion planning

Piano Mover's problem is PSPACE-hard (Reif et al.)



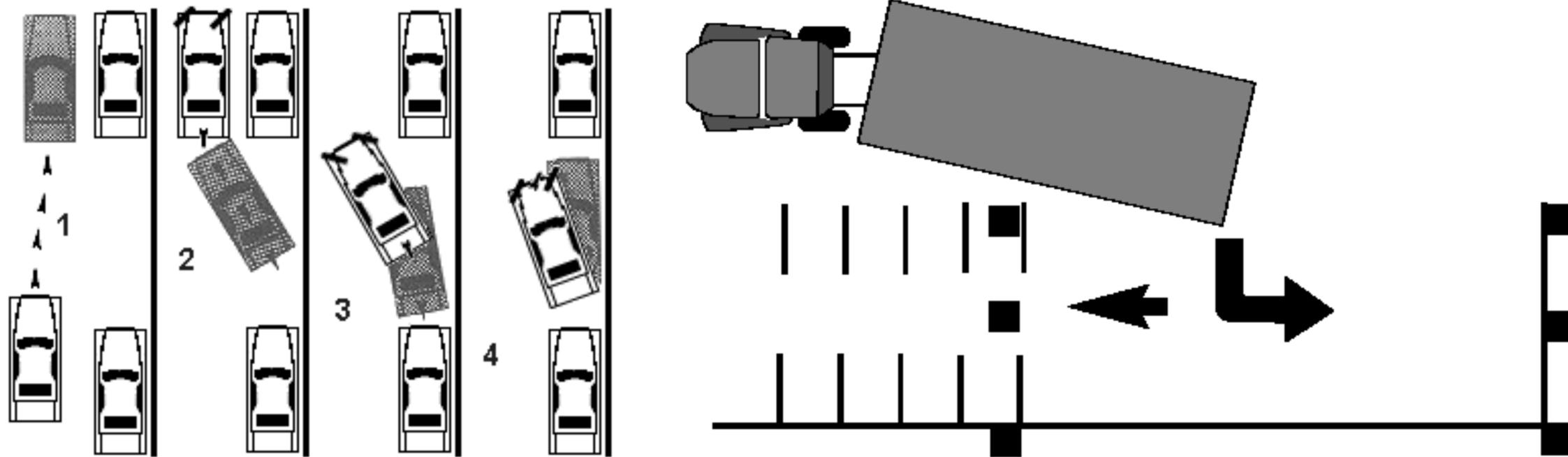
Even planning for translating rectangles is PSPACE-hard!

(Hopcroft et al. 84)

Certain 3D robot planning under uncertain is NEXPTIME-hard!

(Canny et al. 87)

Differential constraints make things **even harder**



These are examples of **non-holonomic system**

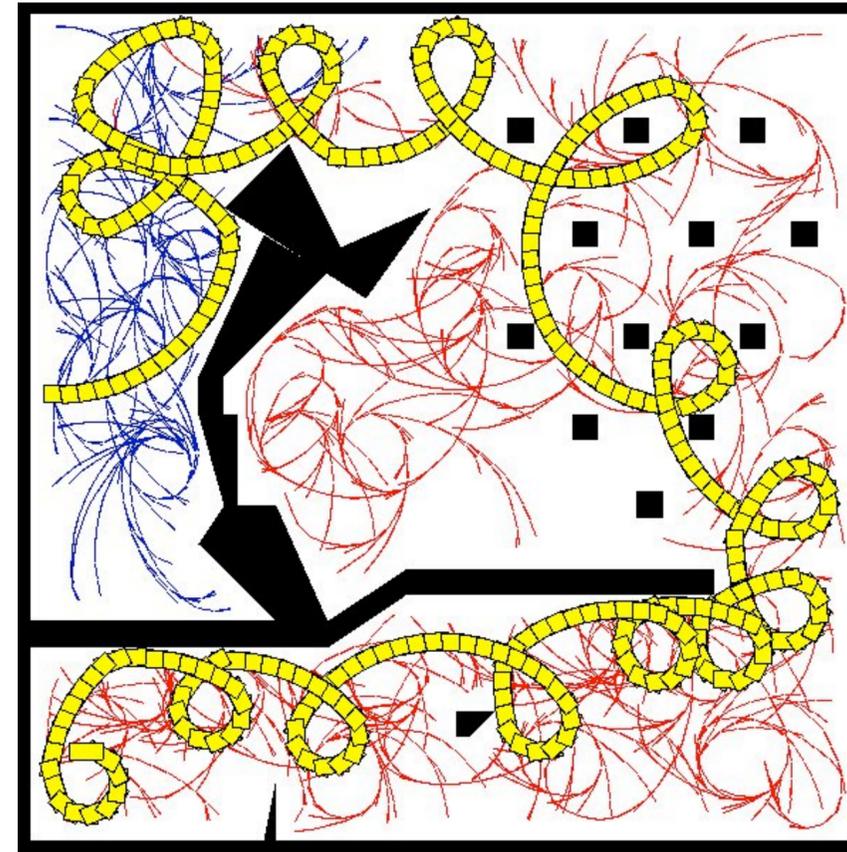
non-holonomic differential constraints are not completely integrable

i.e. the system is trapped in some sub-manifold of the config space

Differential constraints make things **even harder**



Emergency landing where UAV can only turn left



“Left-turning-car”

These are examples of **non-holonomic system**

non-holonomic differential constraints are not completely integrable

i.e. the system is trapped in some sub-manifold of the config space

Geometric Path Planning

**WORKSPACE,
OBSTACLE**

$$\mathcal{W}, \mathcal{O} \subset \mathcal{W}$$

**CONFIGURATION
SPACE**

$$\mathcal{C}, \mathcal{C}_{\text{obs}}, \mathcal{C}_{\text{free}}$$

**START + GOAL
CONFIGURATION**

$$q_s, q_g$$

PLANNING

$$\xi : [0, 1] \rightarrow \mathcal{C}_{\text{free}}$$
$$\xi(0) = q_s, \xi(1) = q_g$$

**COLLISION-
FREE PATH**

**(WHICH MAY
MINIMIZE COST)**

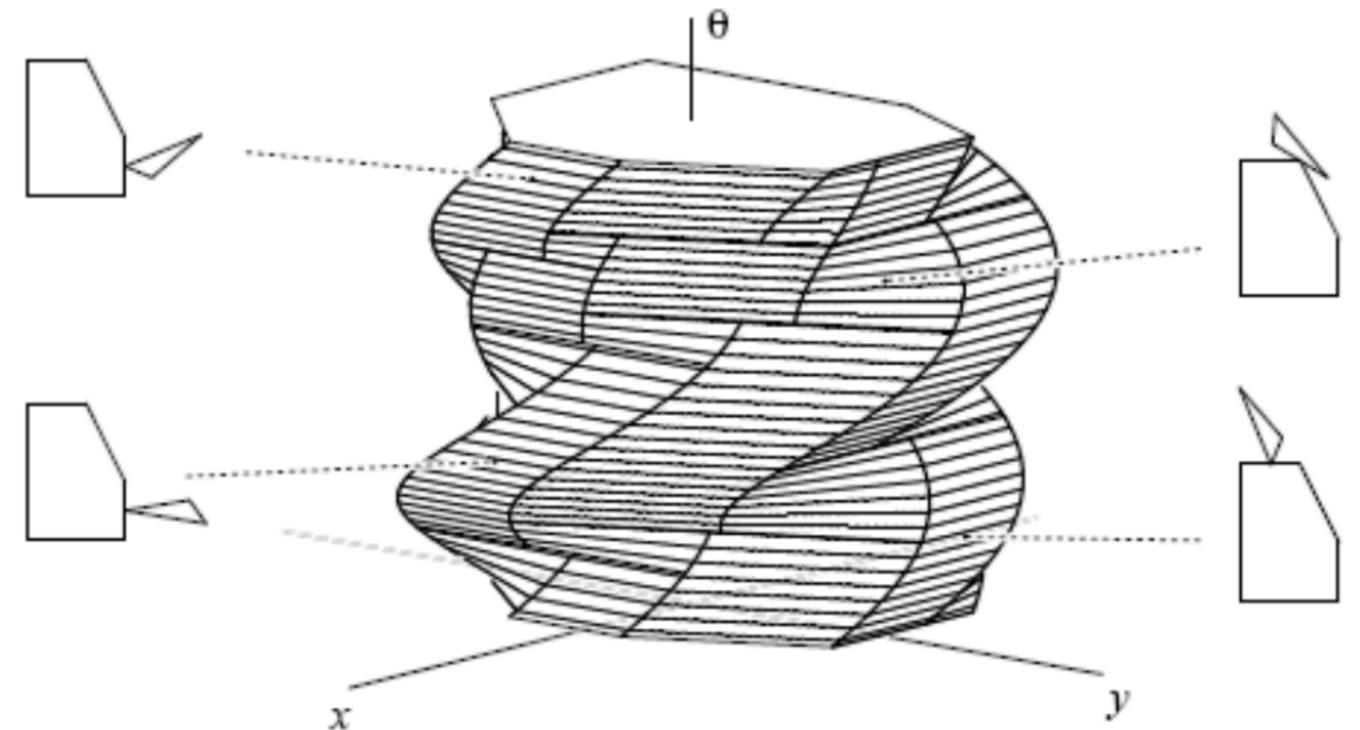
Challenges in Motion Planning

- Computing configuration-space obstacles
- Planning in continuous high-dimensional space

HARD!

HARD!

Goal: tractable approximations
with provable guarantees!



(EXAMPLE FROM HOWIE CHOSET)

CSE 478 Robot Autonomy

Motion Planning

Abhishek Gupta (abhgupta@cs)
Siddhartha Srinivasa (siddh@cs)

TAs:

Carolina Higuera (chiguera@cs)
Rishabh Jain (jrishabh@cs)
Entong Su (ensu@cs)

