

Configuration Spaces

EECS 367
Intro. to Autonomous Robotics

ROB 320
Robot Operating Systems

Winter 2022



autorob.org

more than meets the eye

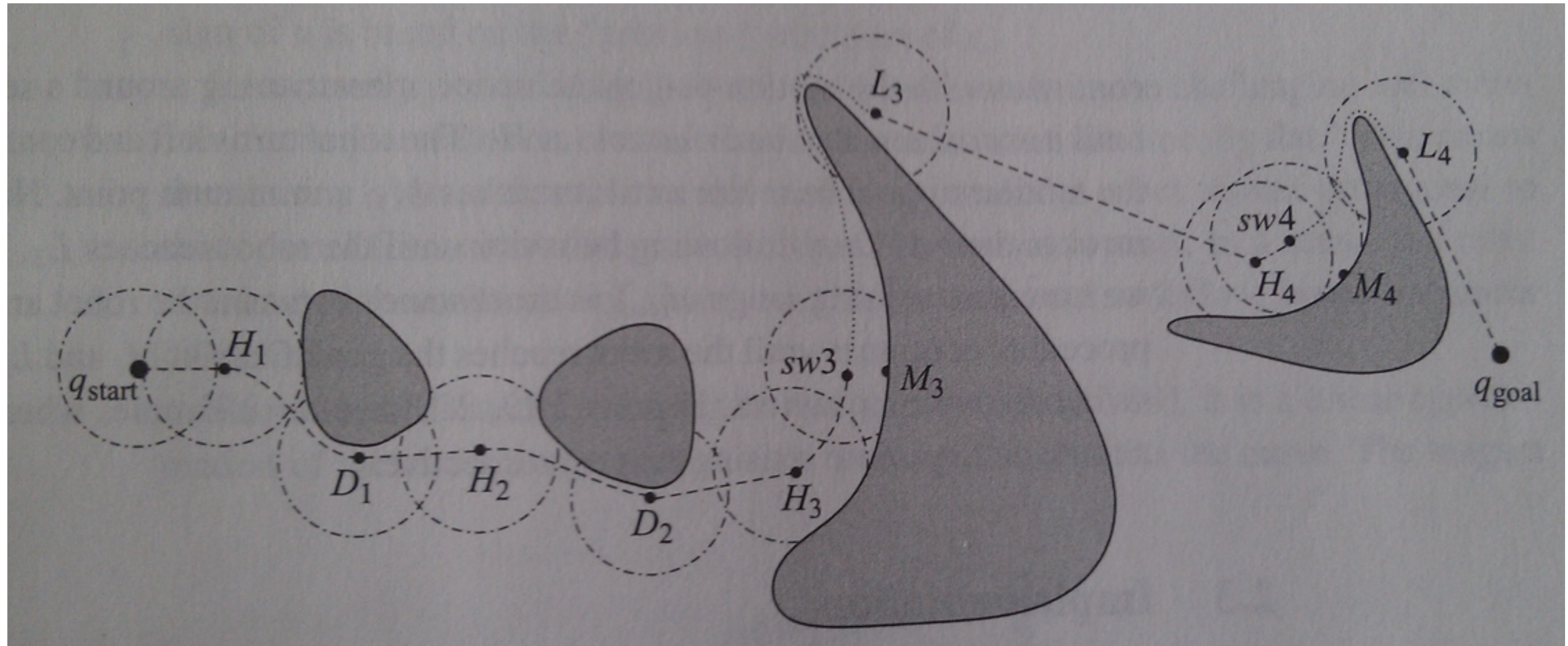
Configuration Spaces



autorob.org

more than meets the eye

Last time: Tangent Bug



What does BugX assume that Random
Walk does not?

What does BugX assume that Random Walk does not?

Localization: knowing the robot's location, at least wrt. distance to goal

What does BugX assume that Random Walk does not?

Localization: knowing the robot's location, at least wrt. distance to goal

What do search algorithms assume that BugX does not?

What does BugX assume that Random Walk does not?

Localization: knowing the robot's location, at least wrt. distance to goal

What do search algorithms assume that BugX does not?

A graph of valid locations that can be traversed

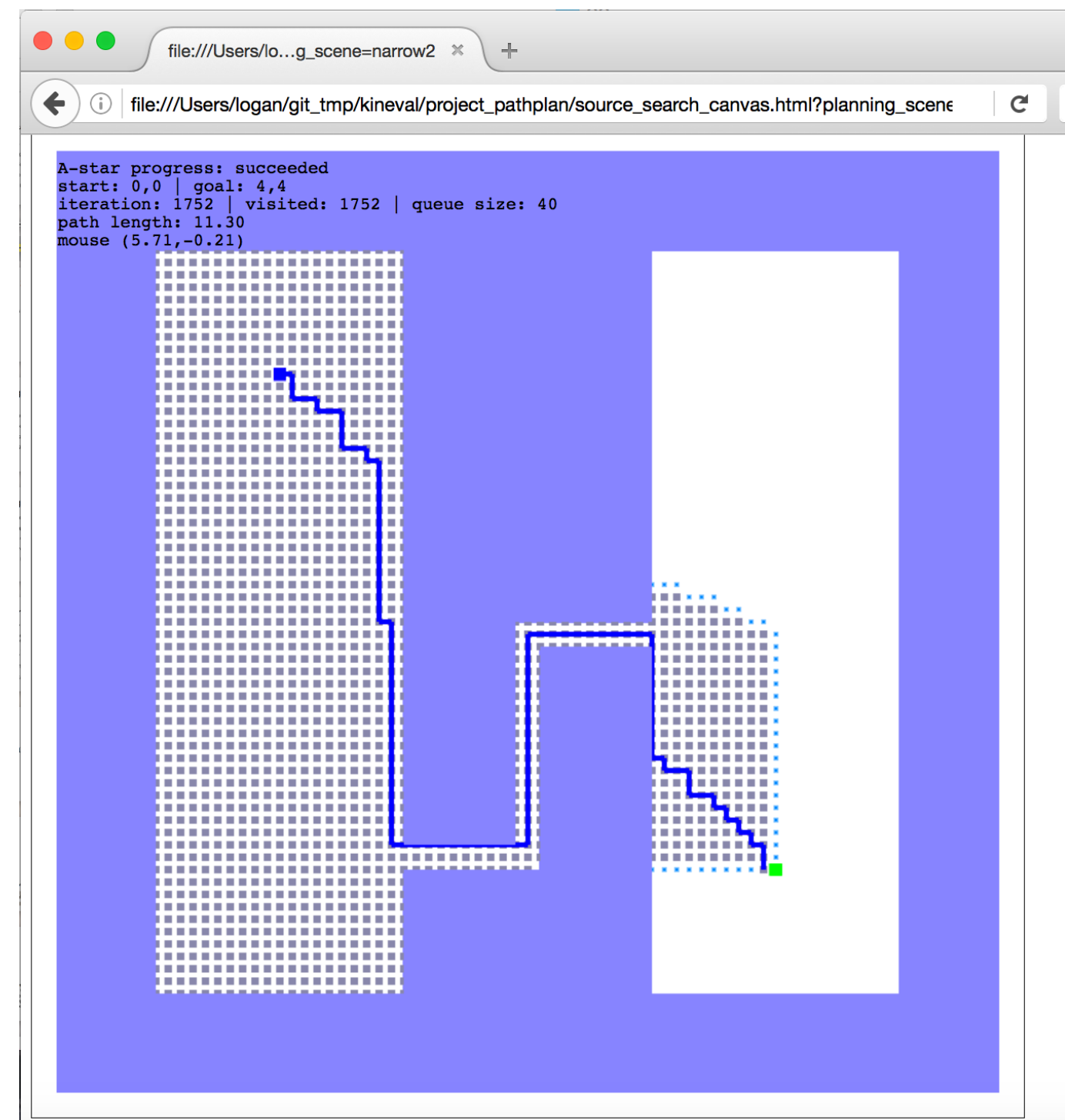
Suppose we have or can build such a graph...

Approaches to motion planning

- Bug algorithms: Bug[0-2], Tangent Bug
- Graph Search (fixed graph)
 - Depth-first, Breadth-first, Dijkstra, A-star, Greedy best-first
- **Roadmap Search (build graph):**
 - **Probabilistic Road Maps, Rapidly-exploring Random Trees**
- Optimization (local search):
 - Gradient descent, potential fields, Wavefront

Will our current search methods apply to this robot?

Project 1:
2D Path Planning



Project 6:
N-dimensional Motion Planning



Will our current search methods apply to this robot?

Assumptions:

- Known graph of traversability
 - How big is this graph? How was this graph built?
- Known localization and map/obstacles
 - How do we detect collisions?
 - Is our robot just a point in workspace?
- Known link geometry
 - Does robot geometry change wrt. configuration?



Configuration Space

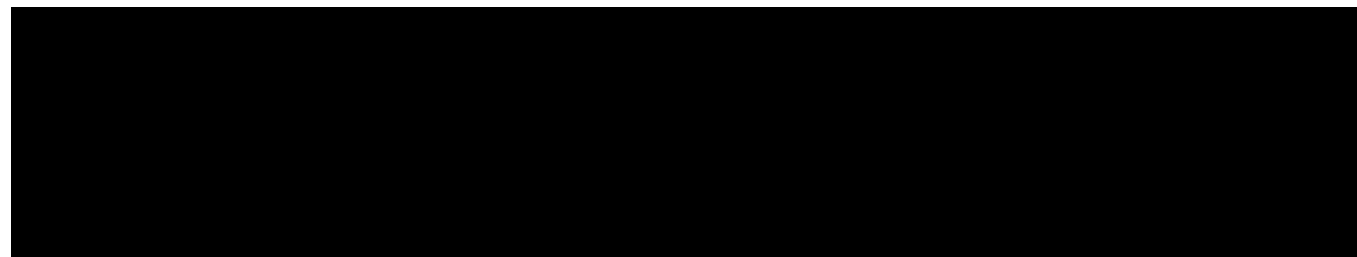
(or C-space)

- C-space (Q) is the space of all possible configurations (q) of a system
 - kinematics: geometry of possible configurations, without respect to physics
 - dynamics: evolution of configurations over time wrt. physics
- Each degree of freedom (q_i) is a dimension of C-space
- The span of C-space is constrained by obstacles (QO_i), joint limits, etc.

Consider some examples of
configuration spaces

Configuration Space

- Consider a robot $d=21$ DOFs, where each DOF can take 1 of $n=10$ angular values
- How many configurations?



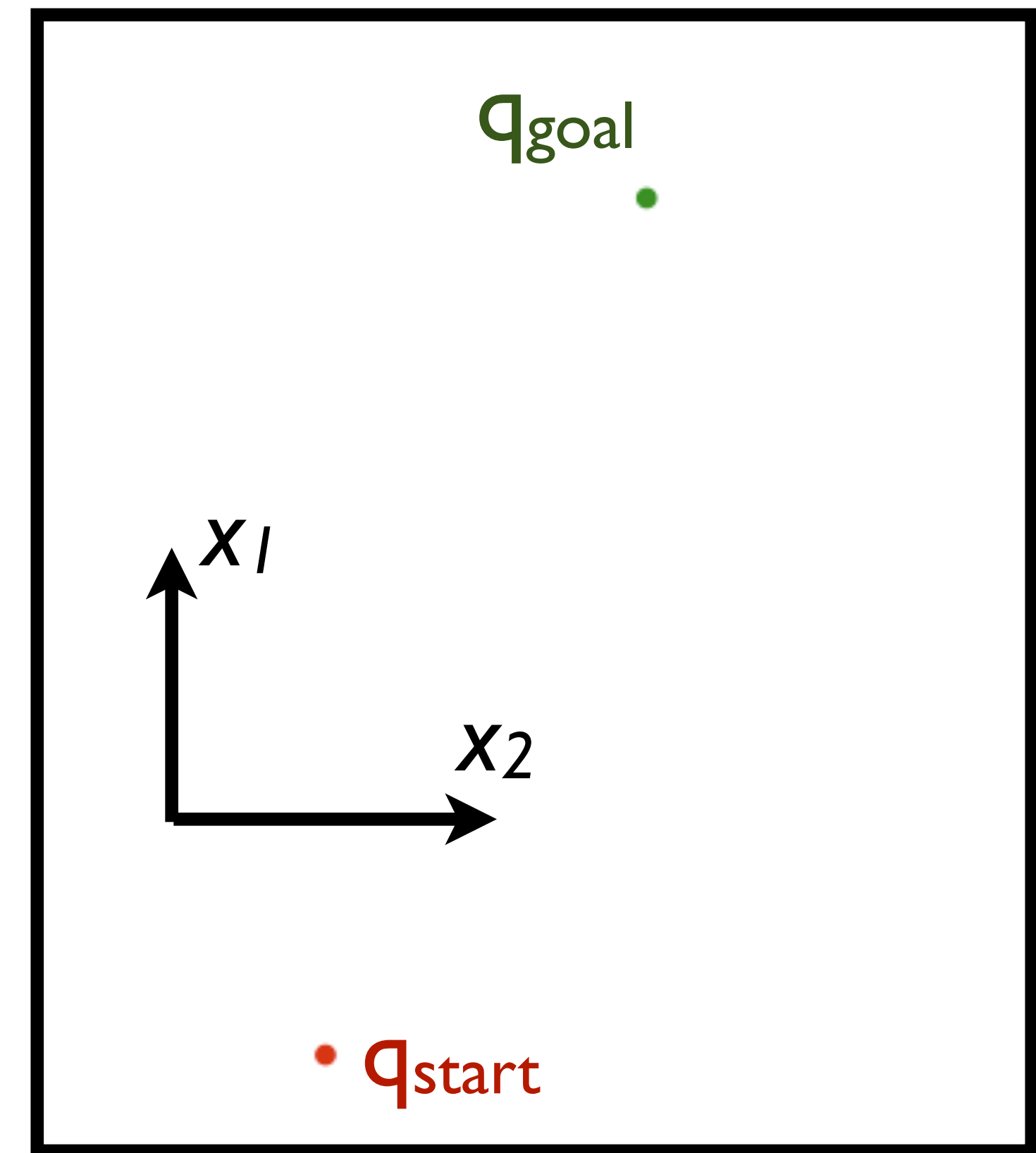
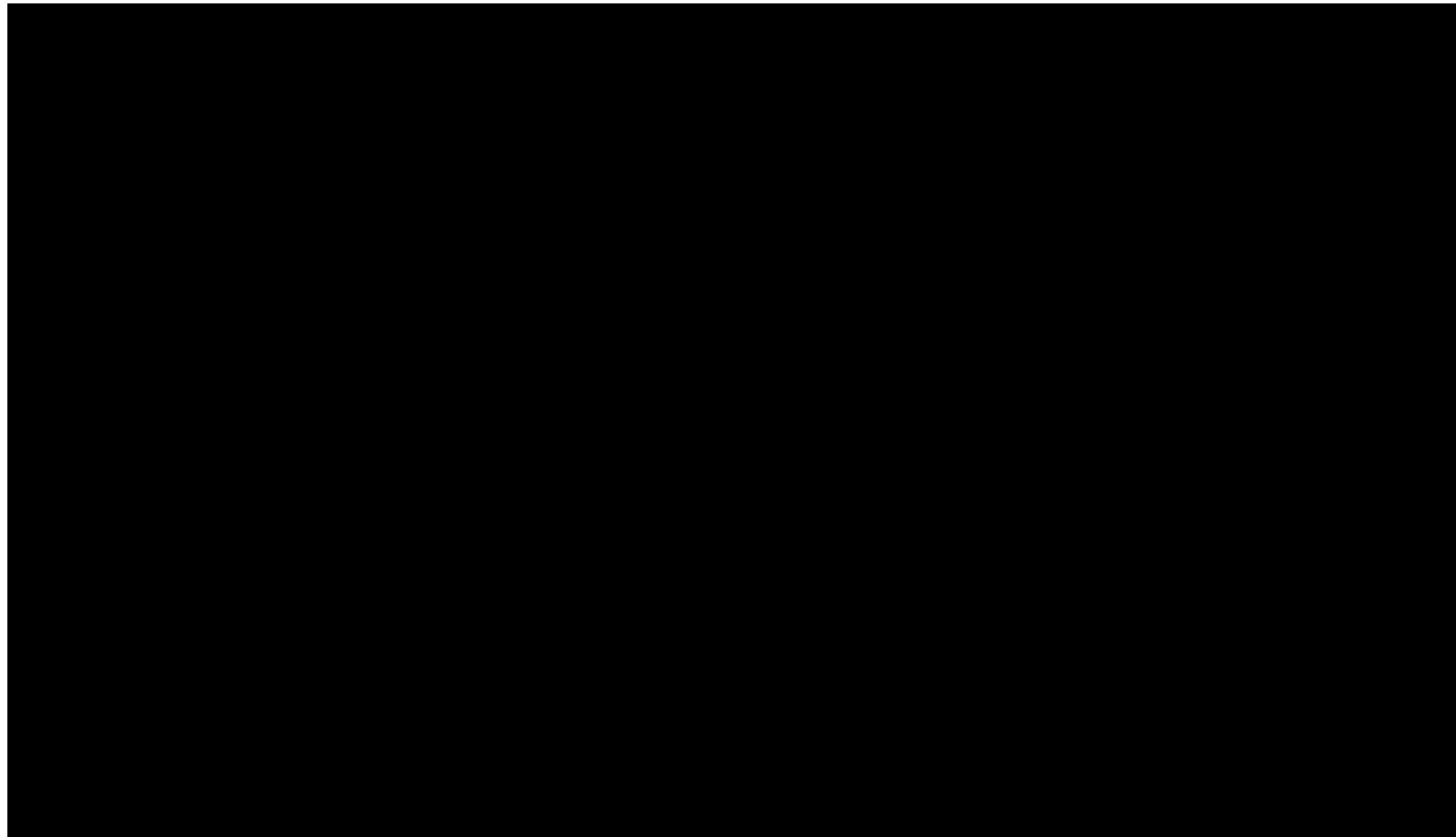
Configuration Space

- Consider a robot $d=21$ DOFs, where each DOF can take 1 of $n=10$ angular values
- How many configurations?
 - 10^{21} , n^d in general
- **“Curse of dimensionality”**
 - exponential growth of C-space wrt. number of DOFs
- Obstacles also create discontinuities and nonlinearities in C-space



C-space examples

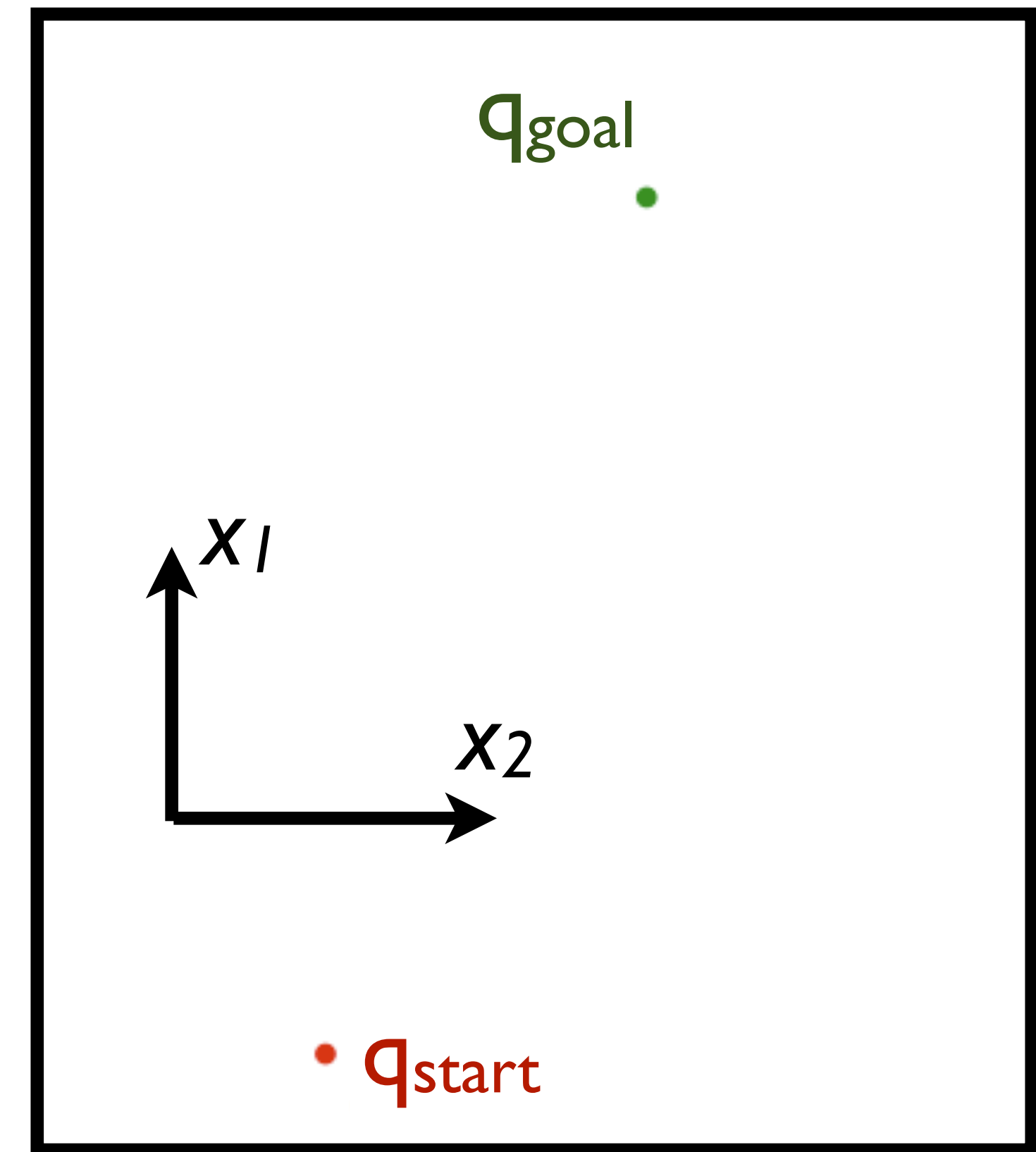
- How many configurations are in the C-space of a planar point robot in a bounded rectangular world?



C-space examples

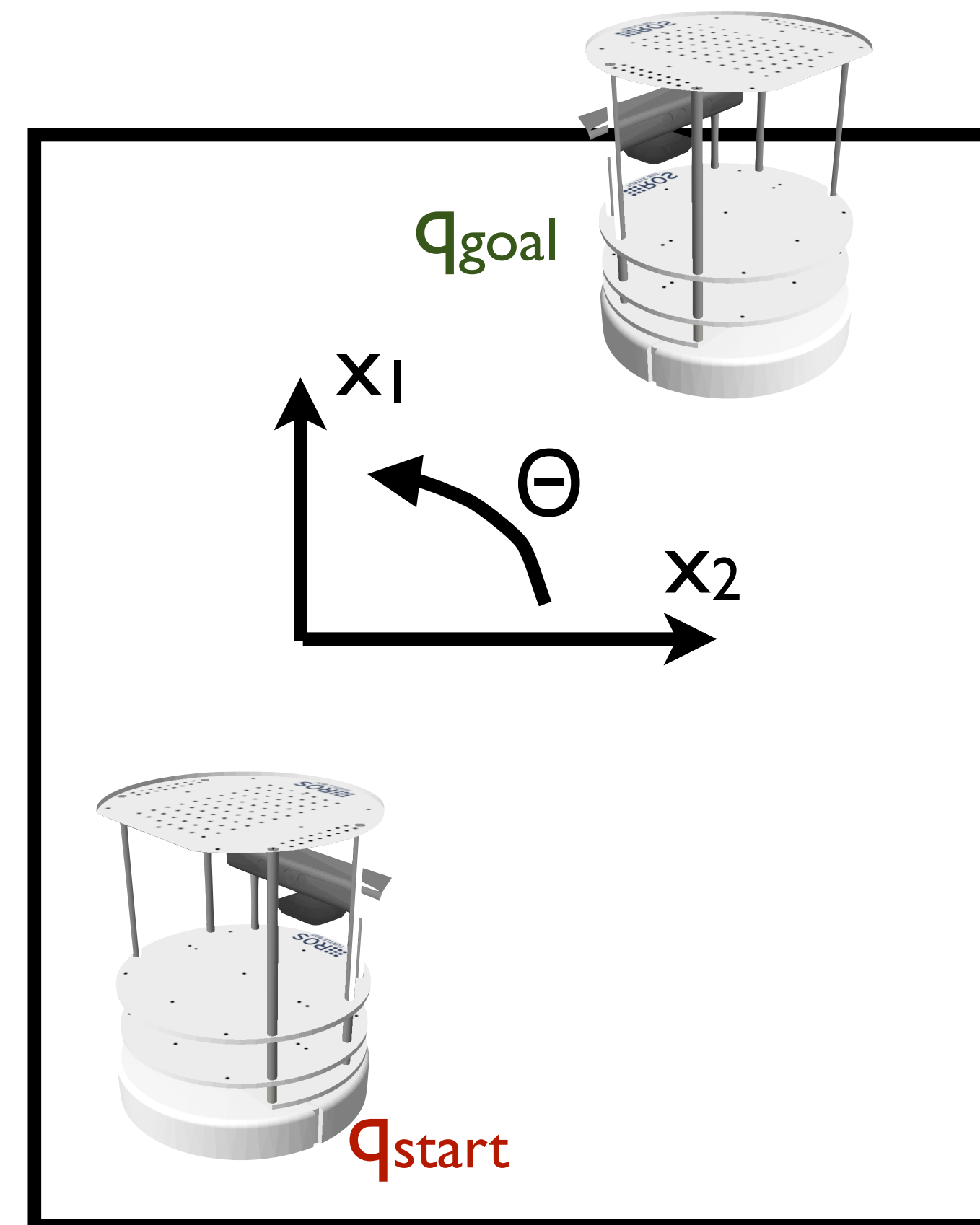
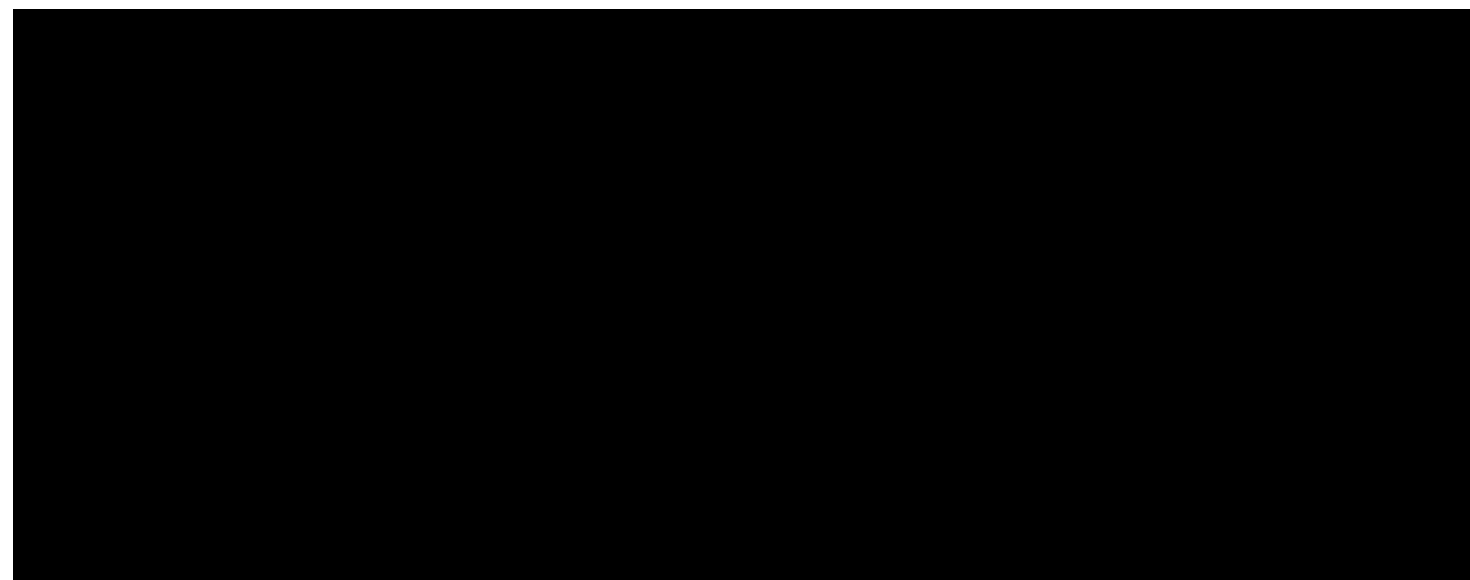
- How many configurations are in the C-space of a planar point robot in a bounded rectangular world?
 - DOFs: 2, $\{x_1, x_2\}$
 - Number of poses is infinite
 - C-space: $\mathbb{R}^2$

Topologically, this C-space is a homeomorphism of $\mathbb{R}^2$



C-space examples

- What is the C-space of a Turtlebot?



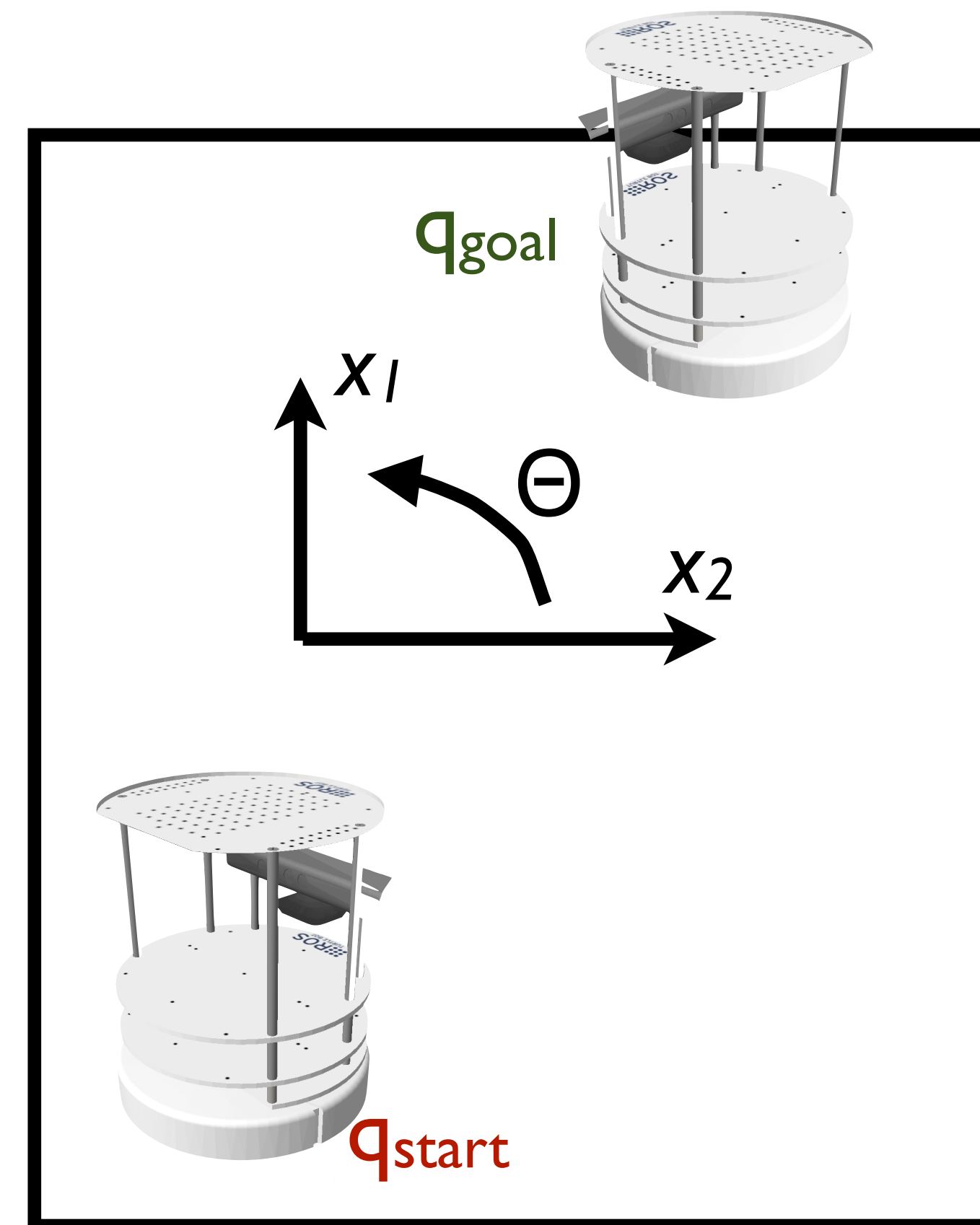
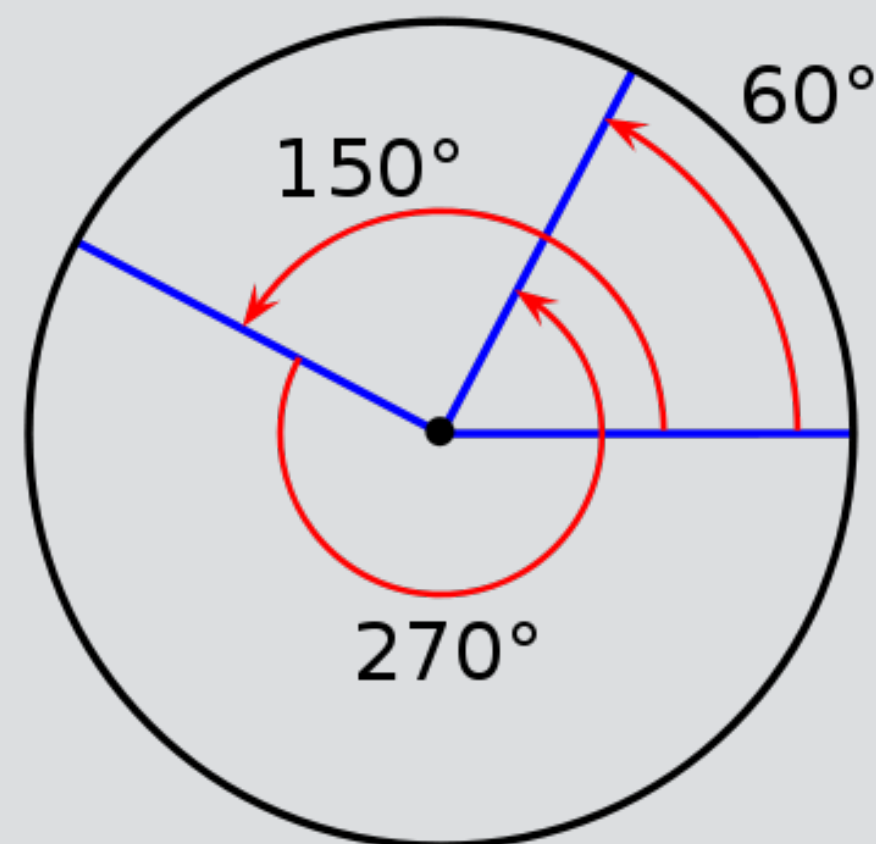
C-space examples

- What is the C-space of a Turtlebot?
 - DOFs: 3, $\{x_1, x_2, \Theta\}$
 - C-space: $\mathbb{R}^2 \times S^1$

S^1 is the 1-sphere
group of 1D rotations

S^n is the n-sphere

$$S^1 \times S^1 \neq S^2$$



C-space examples

- What is the C-space of a Turtlebot?

- DOFs: 3, $\{x_1, x_2, \Theta\}$

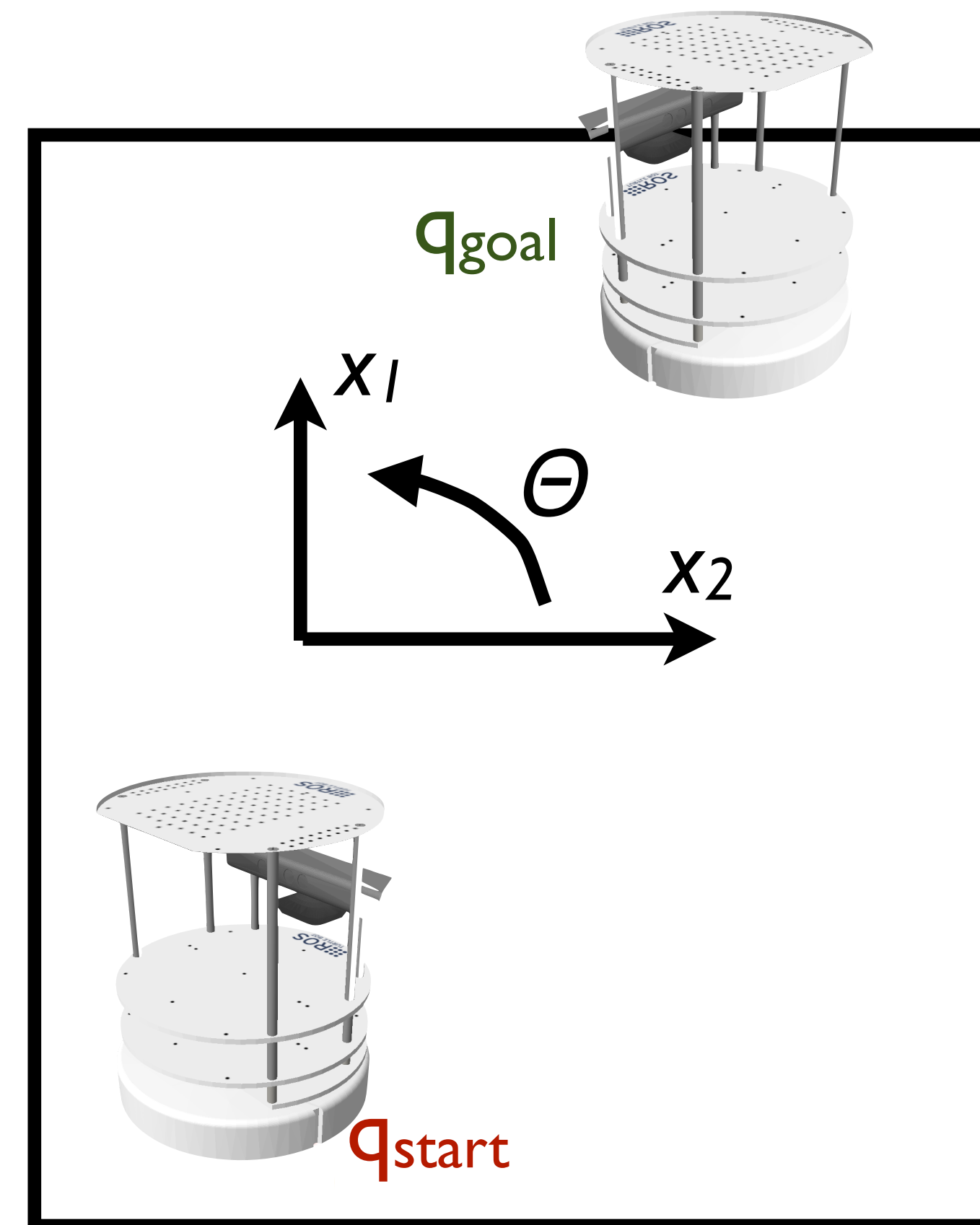
- C-space: $\mathbb{R}^2 \times S^1$

2D translation

rotation in 2D

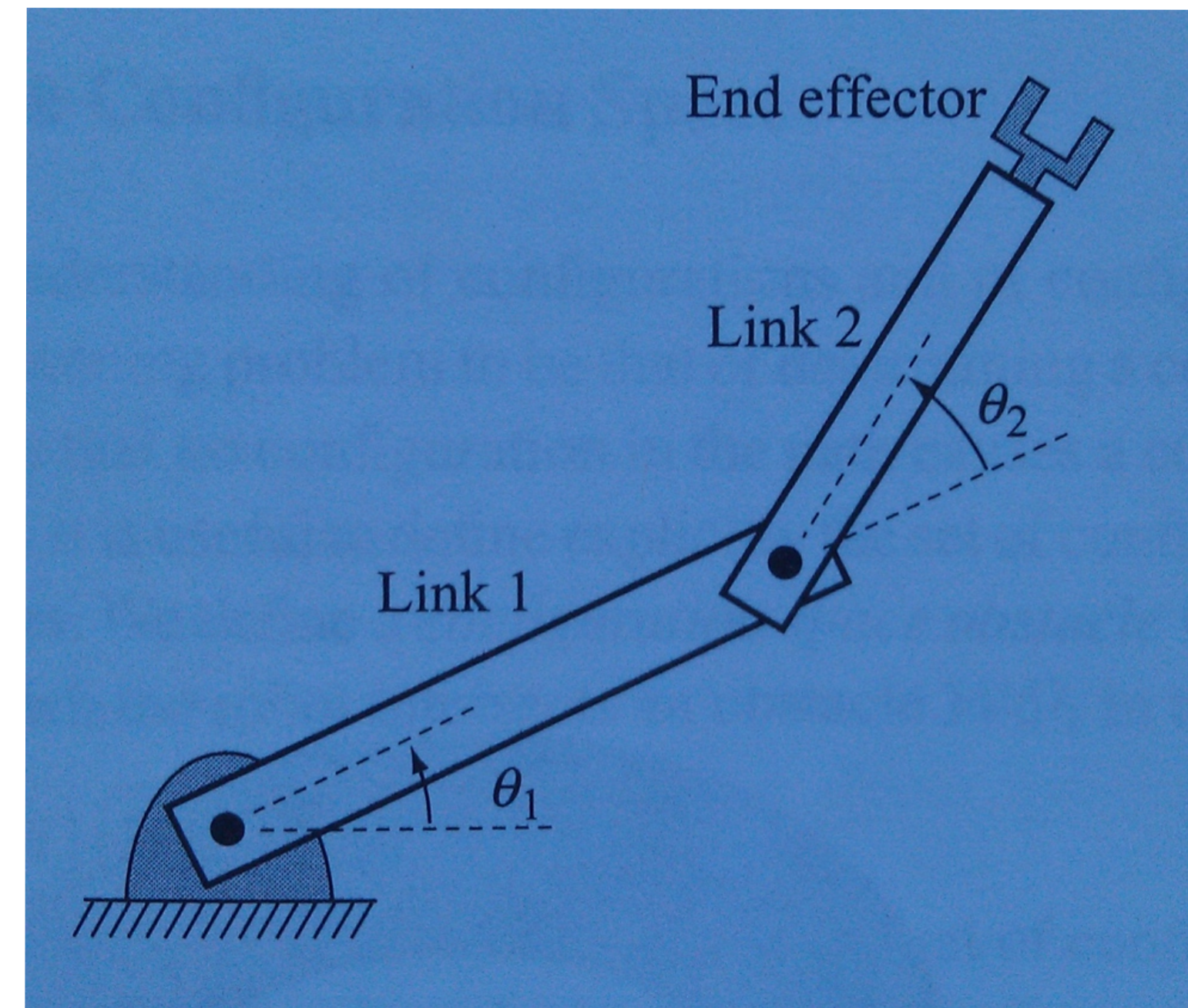
$\mathbb{R}^2 \times S^1$ is also known as the $SE(2)$ group.

Group of homogeneous transformations in 2D



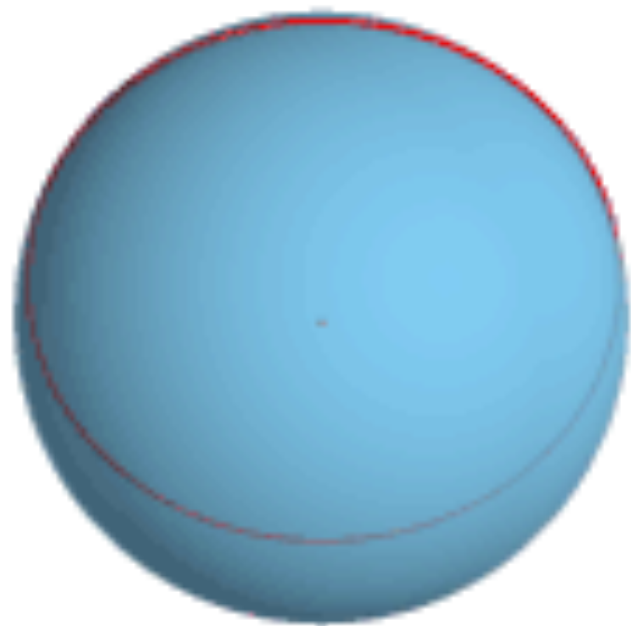
C-space examples

- What is the C-space of a planar arm with 2 rotational joints?
 - DOFs: [REDACTED]
 - C-space: [REDACTED]

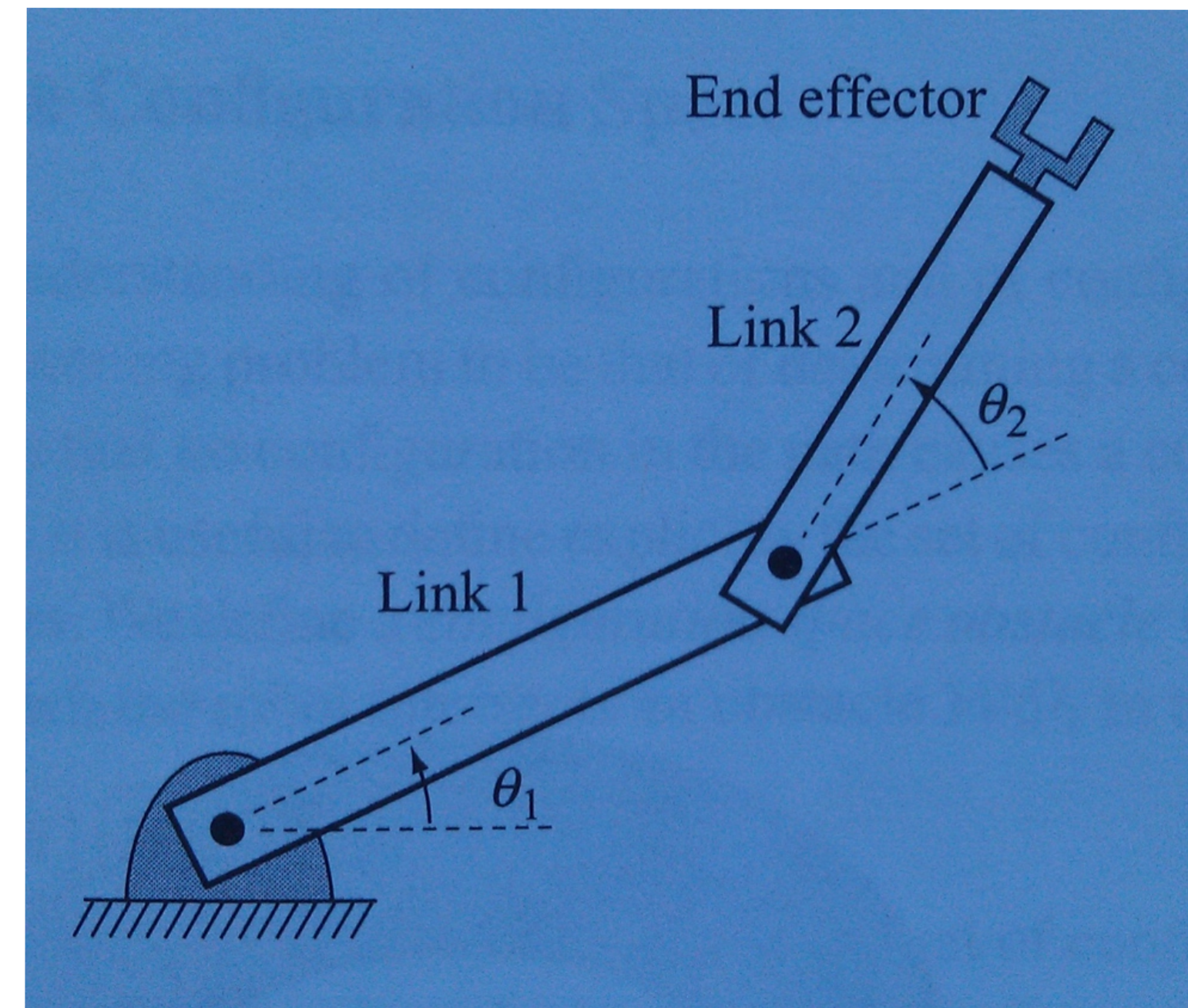


C-space examples

- What is the C-space of a planar arm with 2 rotational joints?
 - DOFs: 2, $\{\theta_1, \theta_2\}$
 - C-space: $\mathbb{R}^2$ or S^2 or $S^1 \times S^1$?

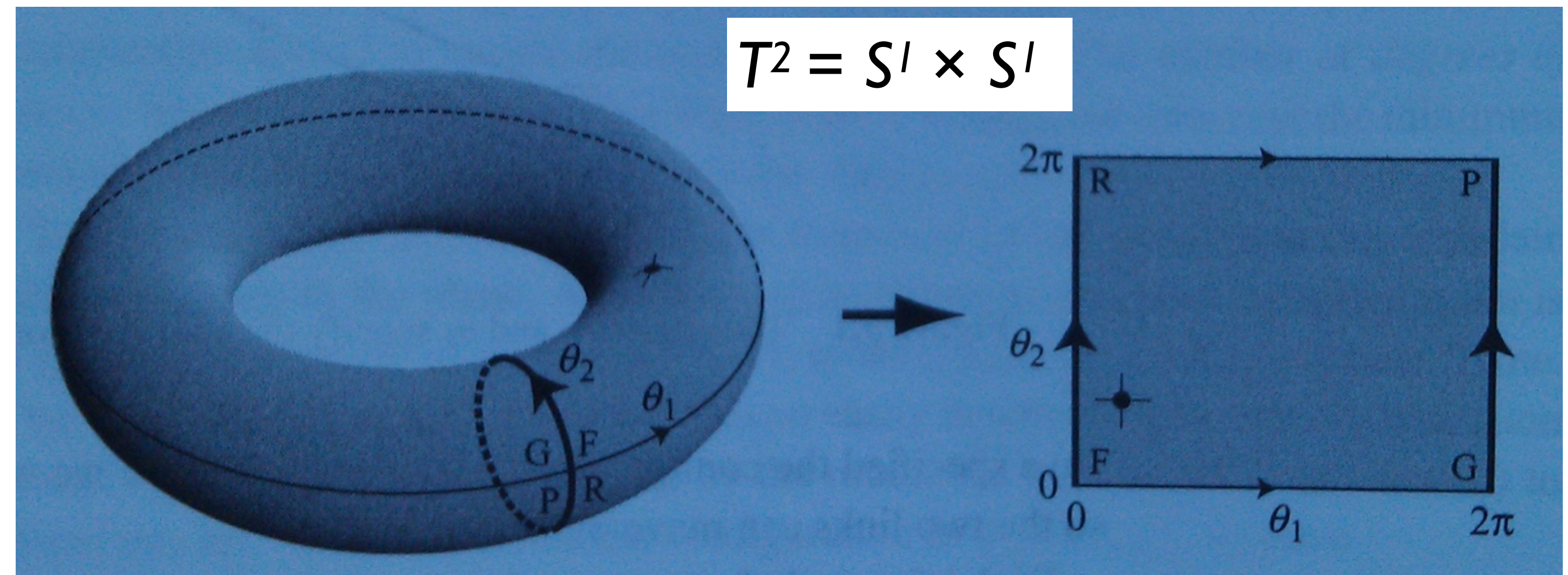
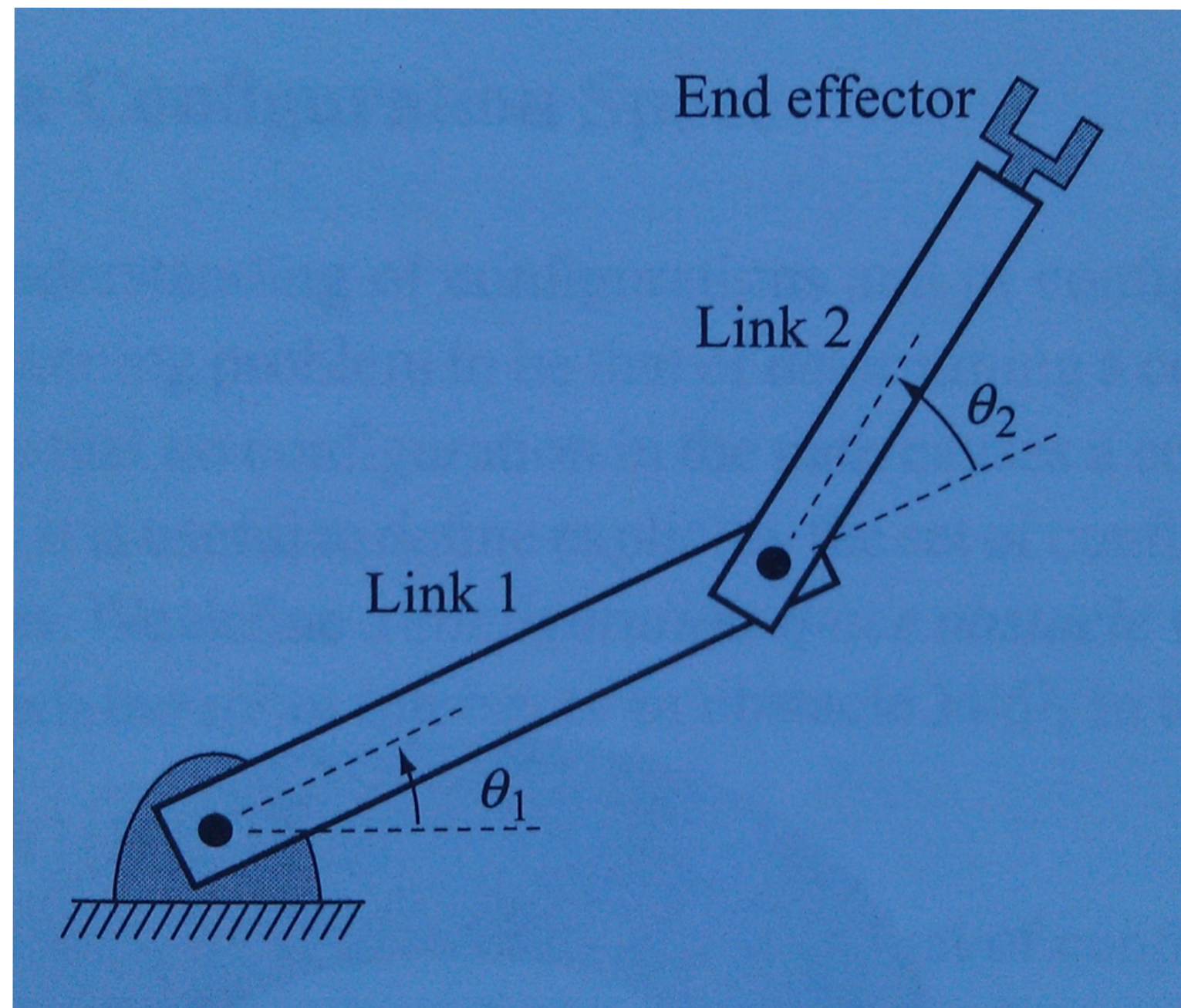


$S^1 \times S^1 = S^2$ when torus axis on surface



T² Torus Group

Space must fuse on each DOF where $2\pi = 0$



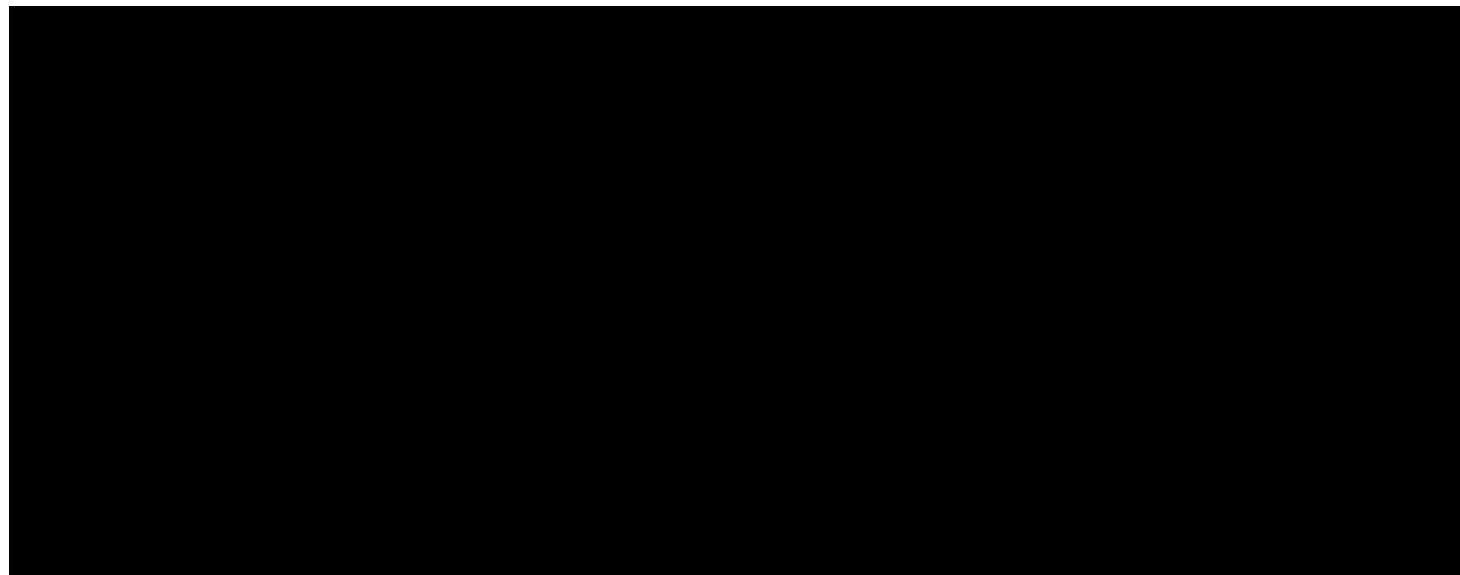
$$T^2 = S^1 \times S^1$$

T^n is the torus group for an N-D rotational system

$$T^n = \underbrace{S^1 \times S^1 \times \dots \times S^1}_n$$

C-space examples

- What is the C-space of a Barrett WAM arm with 4 rotational joints, not including fingers of gripper?



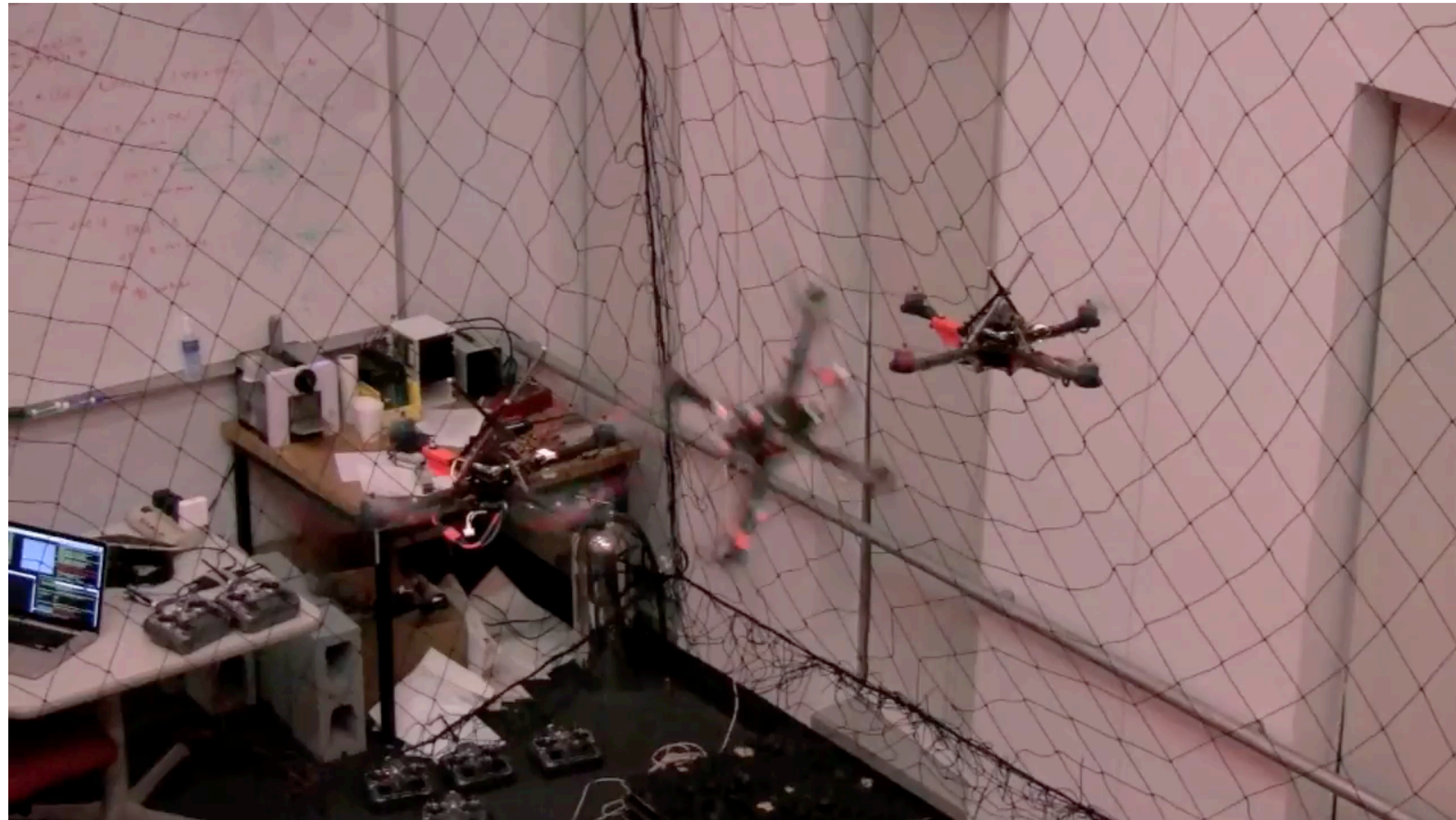
C-space examples

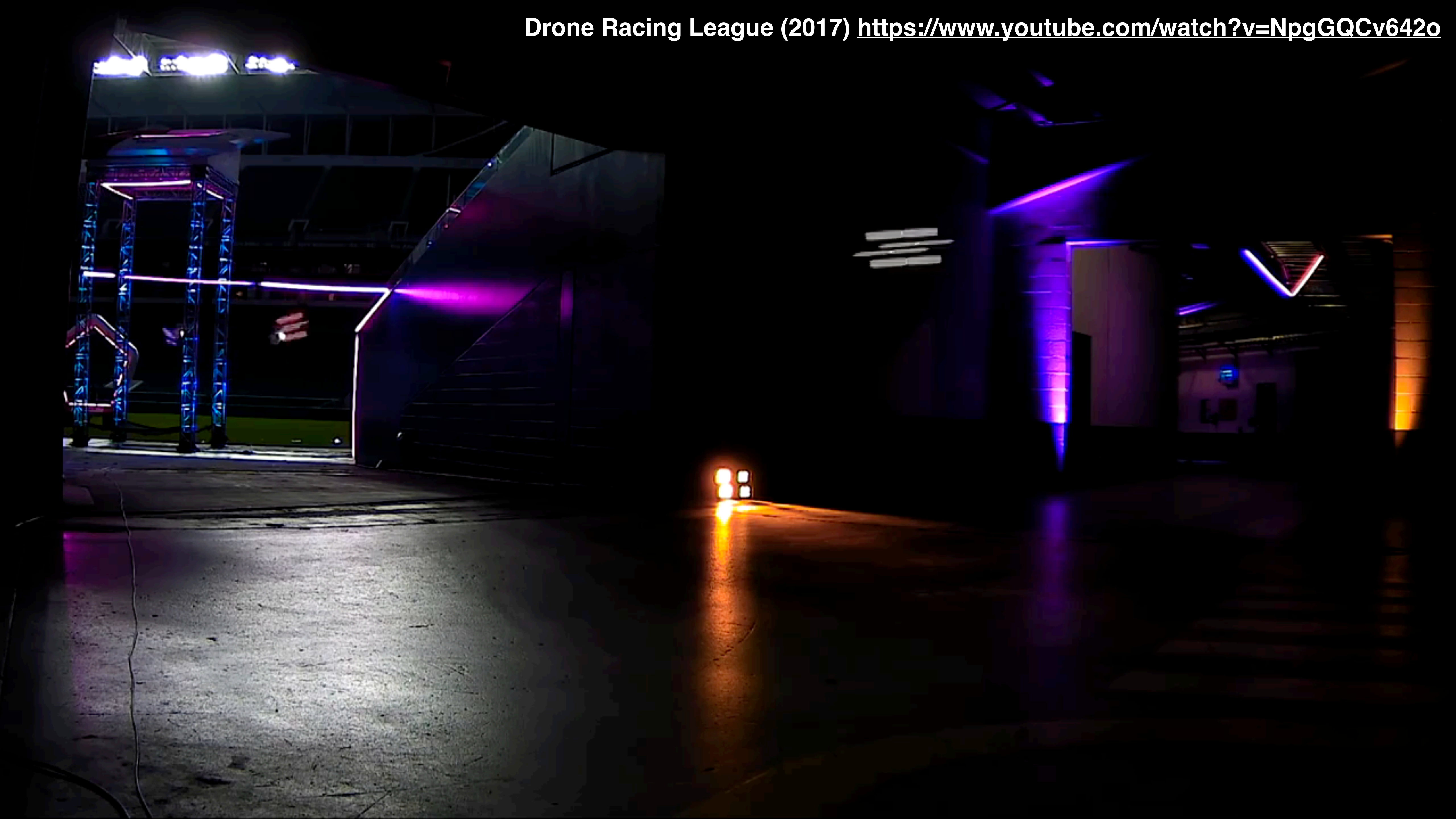
- What is the C-space of a Barrett WAM arm with 4 rotational joints, not including fingers of gripper?
 - DOFs: 4
 - C-space: T^4



C-space examples

- What is the C-space of a quad rotor helicopter?







C-space examples

- What is the C-space of a quad rotor helicopter?

- DOFs: 6

- C-space: $SE(3)$,

- or $\mathbb{R}^3 \times SO(3)$

3D translation

3D rotation

Group of homogeneous transformations in 3D



$SE(3)$ combines:

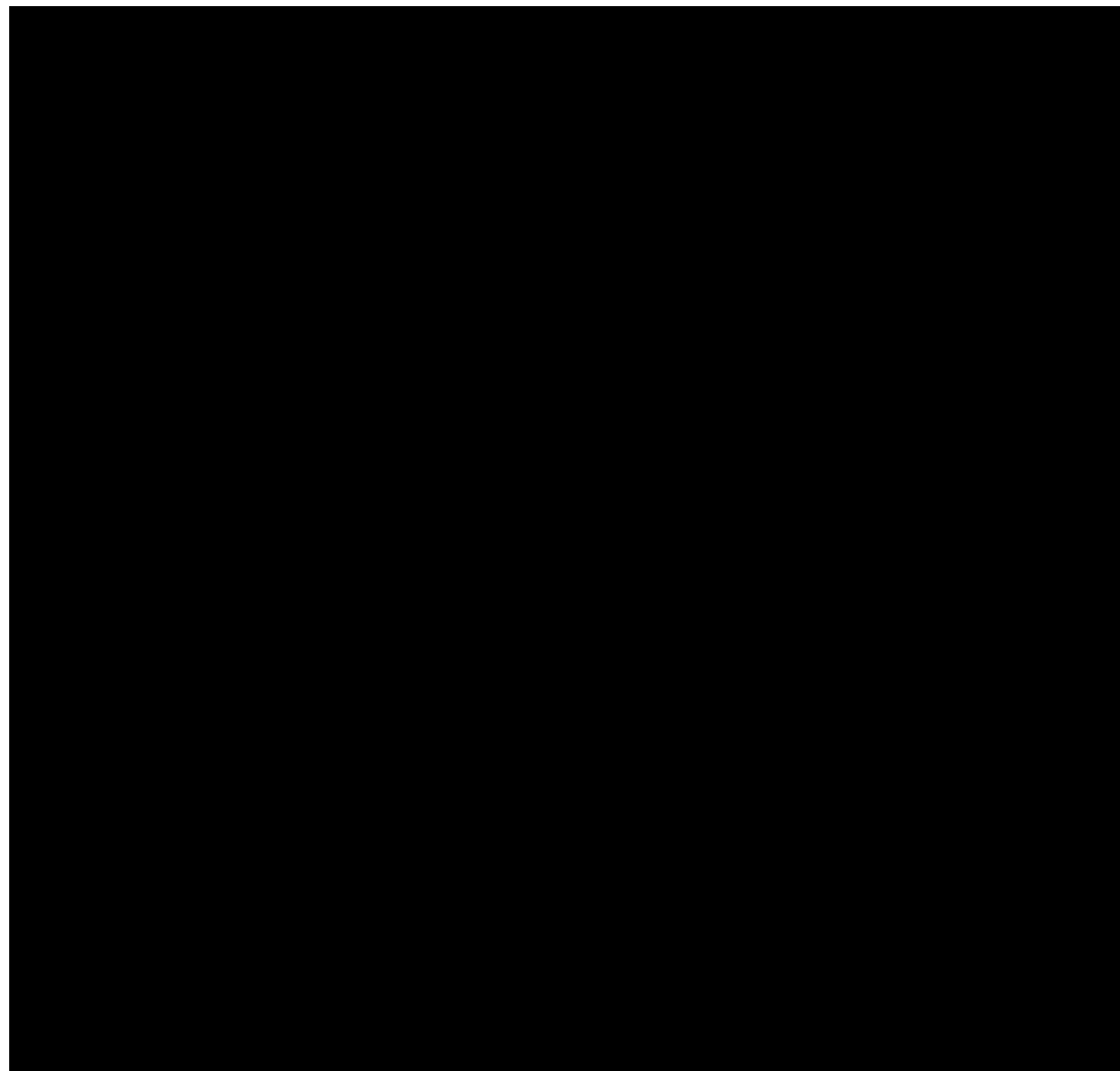
$\mathbb{R}^3$: 3D translation and

$SO(3)$: 3D rotation

$$SO(3) = S^I \times S^I \times S^I$$

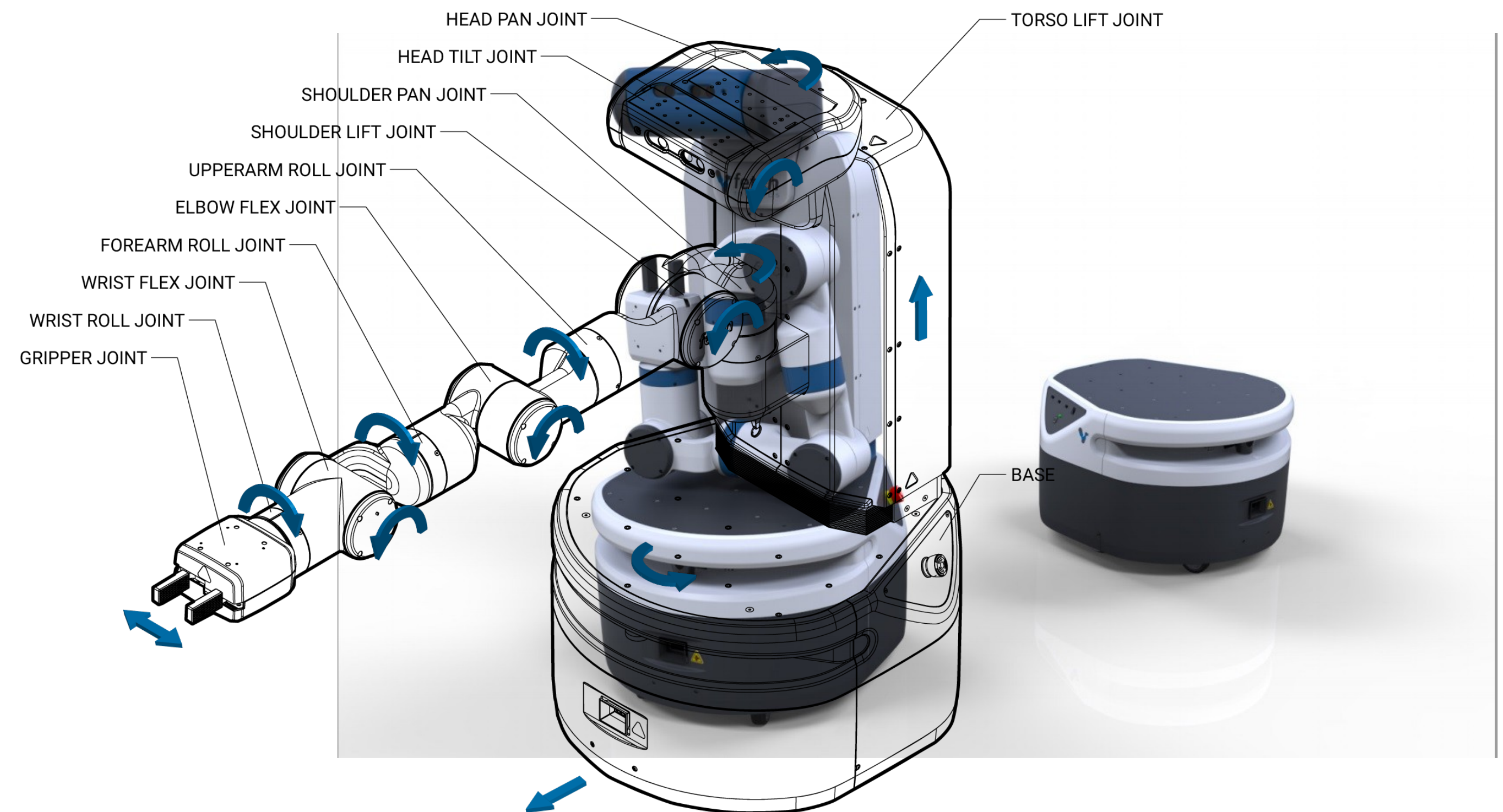
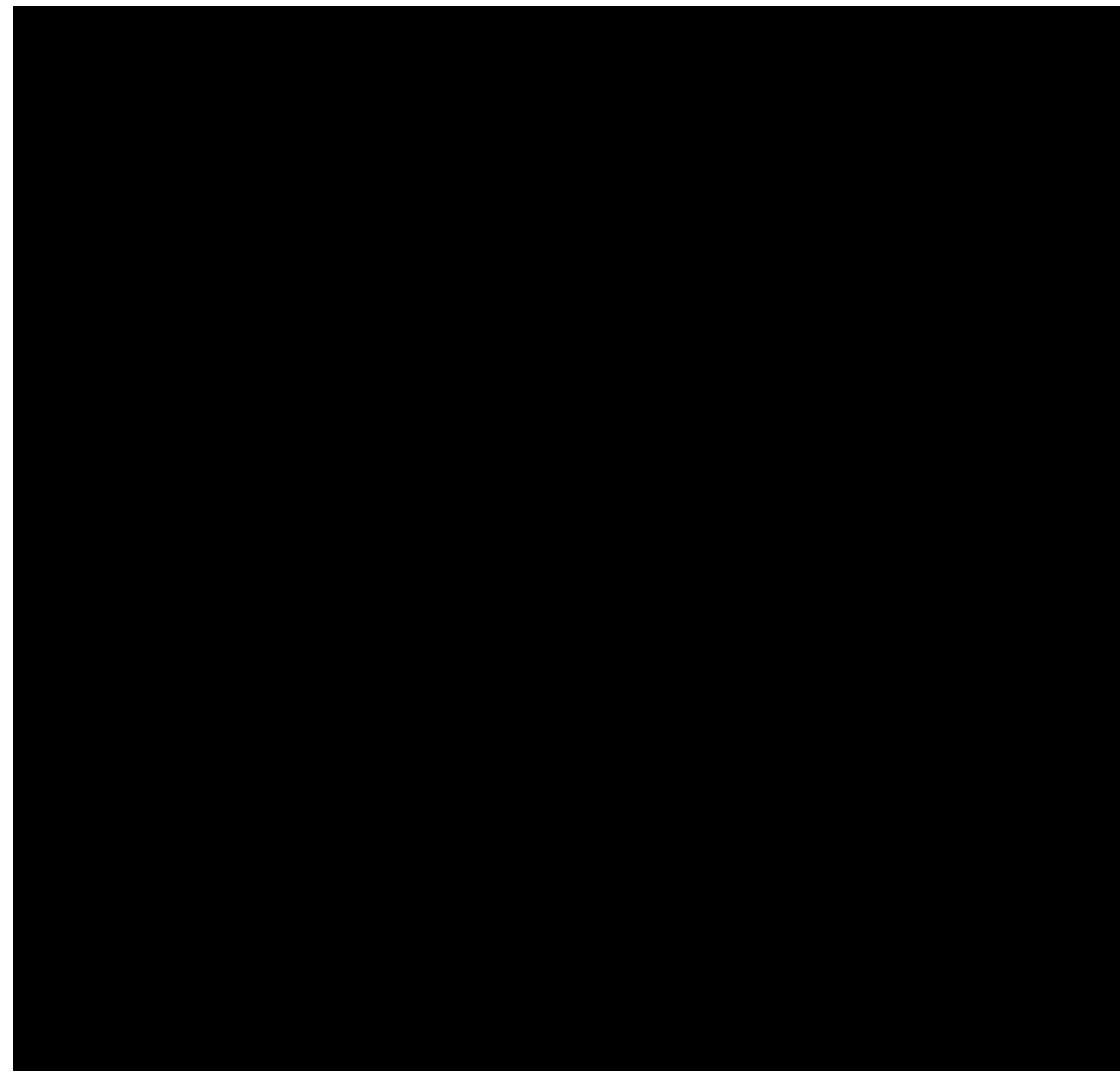
C-space examples

- What is the C-space of a Fetch robot, not including grippers?



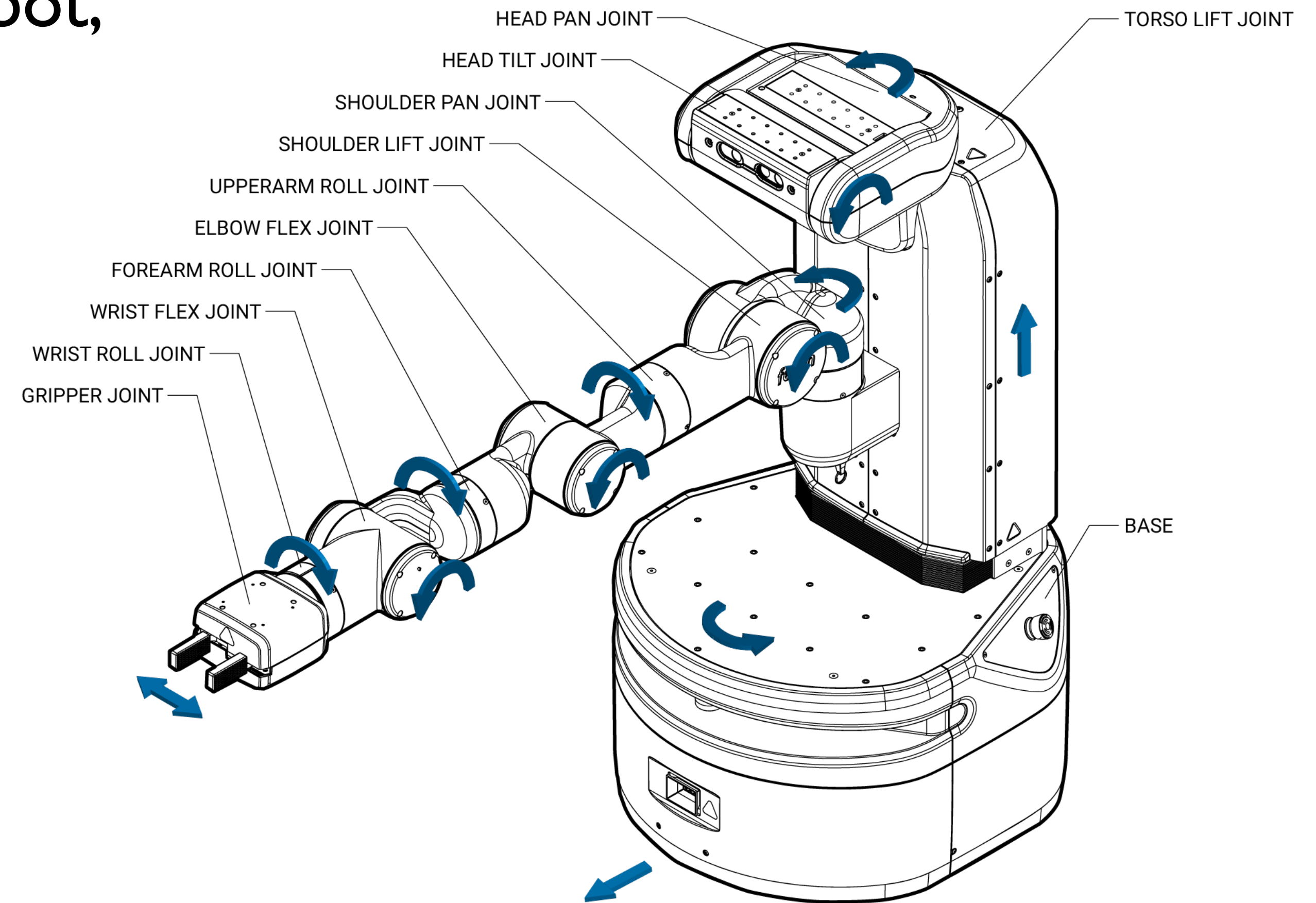
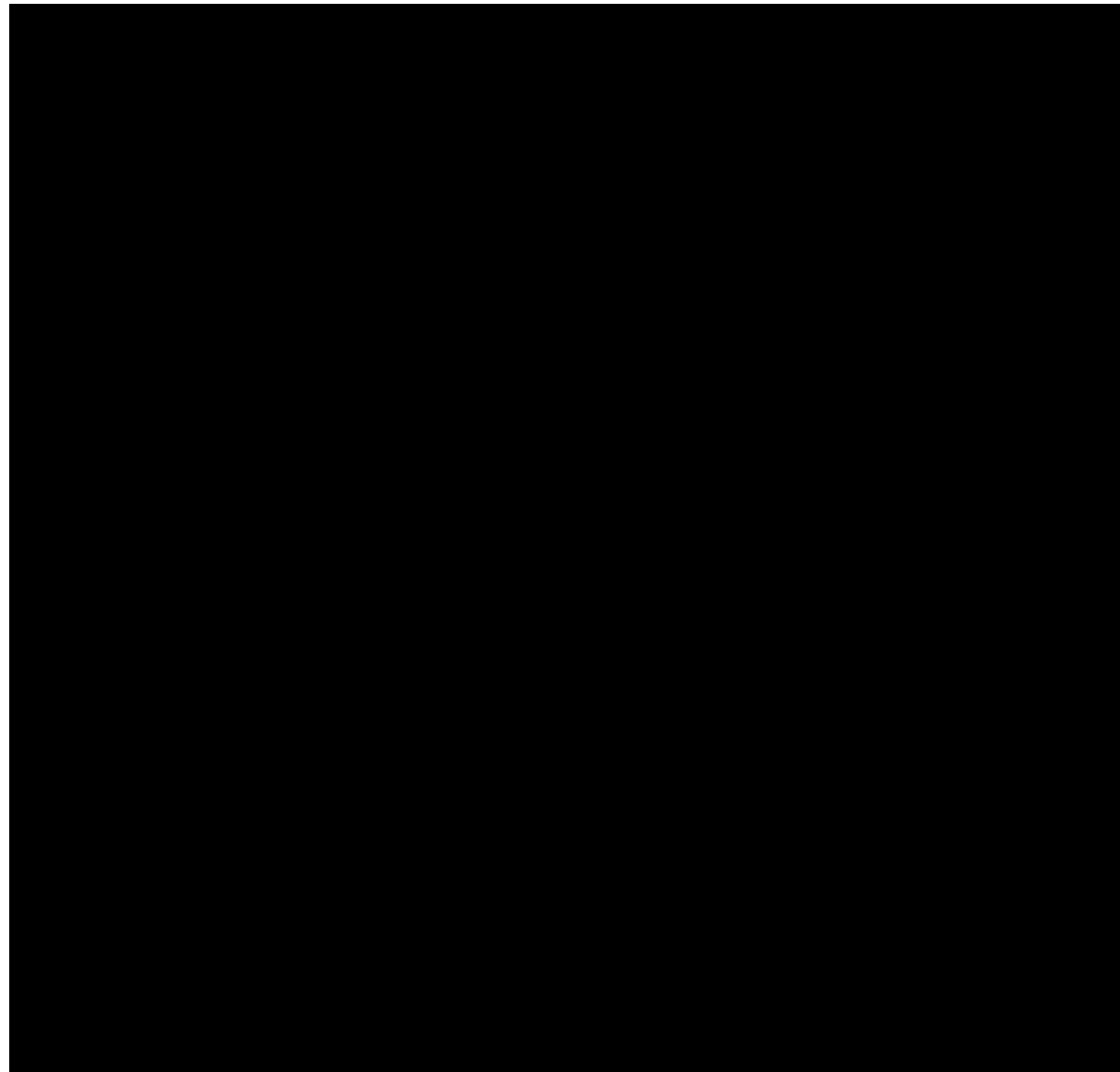
C-space examples

- What is the C-space of a Fetch robot, not including grippers?



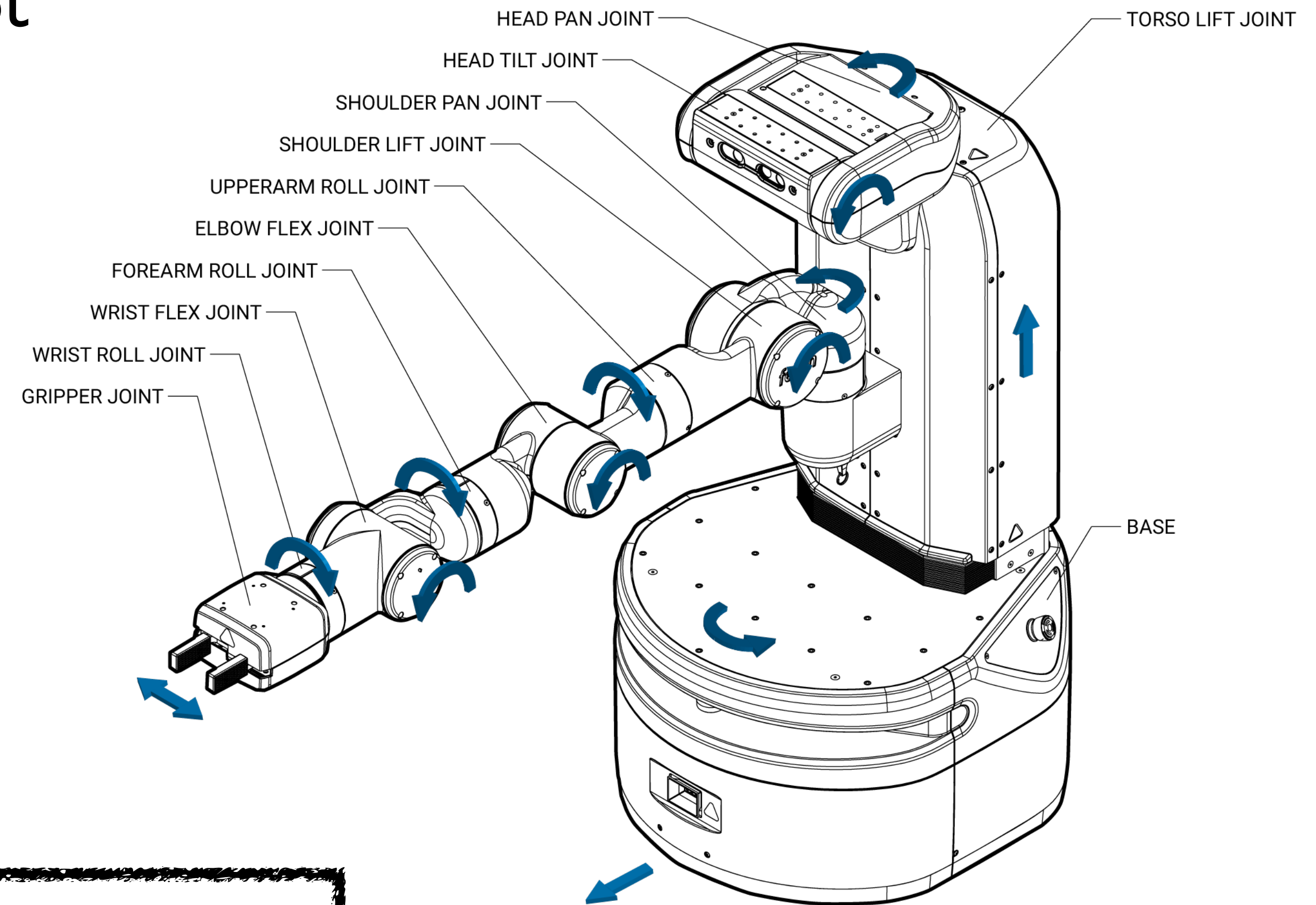
C-space examples

- What is the C-space of a Fetch robot, not including grippers?



C-space examples

- What is the C-space of a Fetch, not including grippers?
- DOFs: 13
 - 3 in base: $SE(2)$
 - 7 in arm: T^7
 - 1 in the spine: $\mathbb{R}^1$
 - 2 in neck: T^2

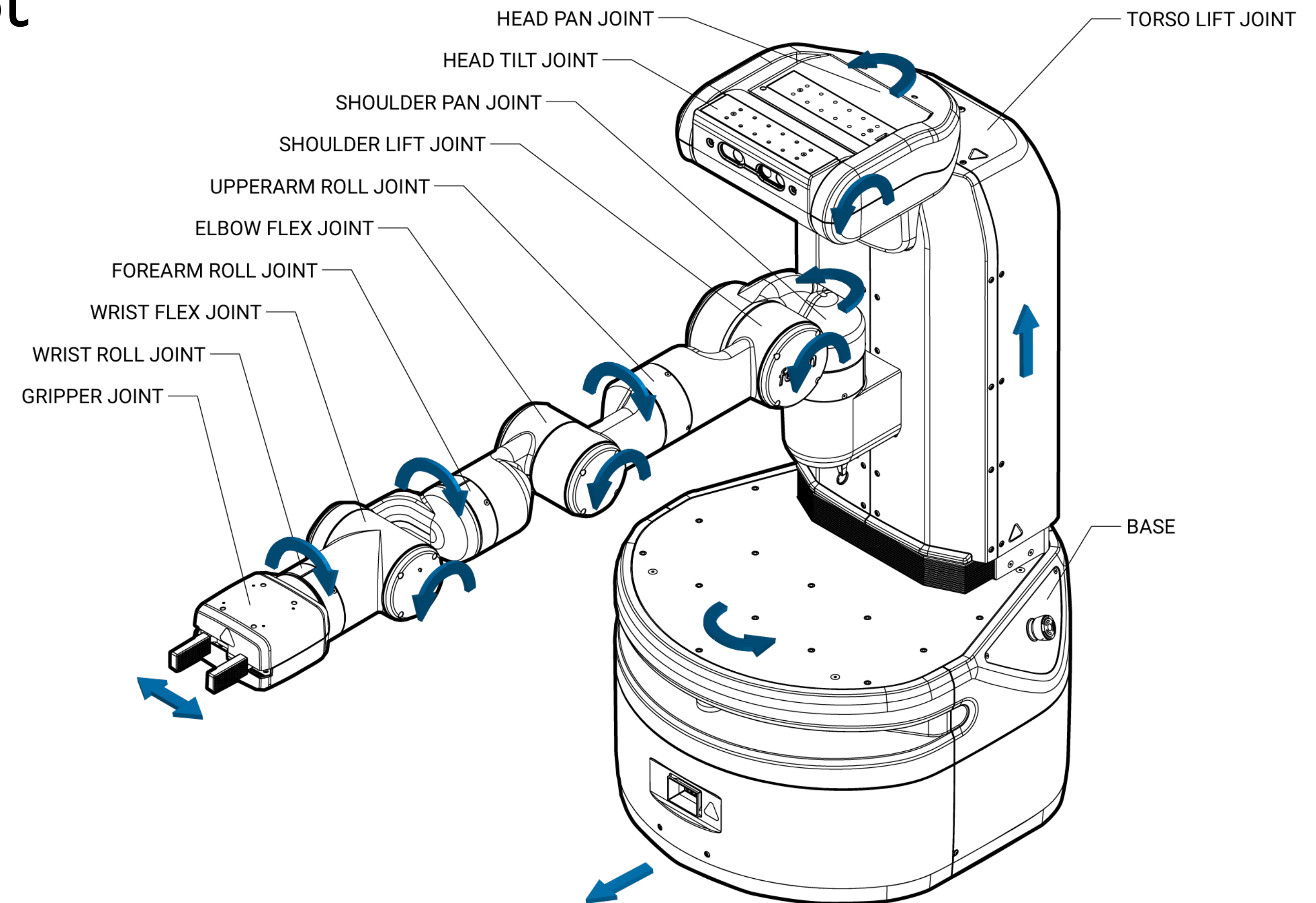


C-space: $SE(2) \times T^7 \times \mathbb{R}^1 \times T^2$

Did we get this wrong?

- What is the C-space of a Fetch, not including grippers?
- DOFs: 13
 - 3 in base: $SE(2)$
 - 7 in arm: ~~T^7~~
 - 1 in the spine: $\mathbb{R}^1$
 - 2 in neck: ~~T^2~~

Consider
joint limits

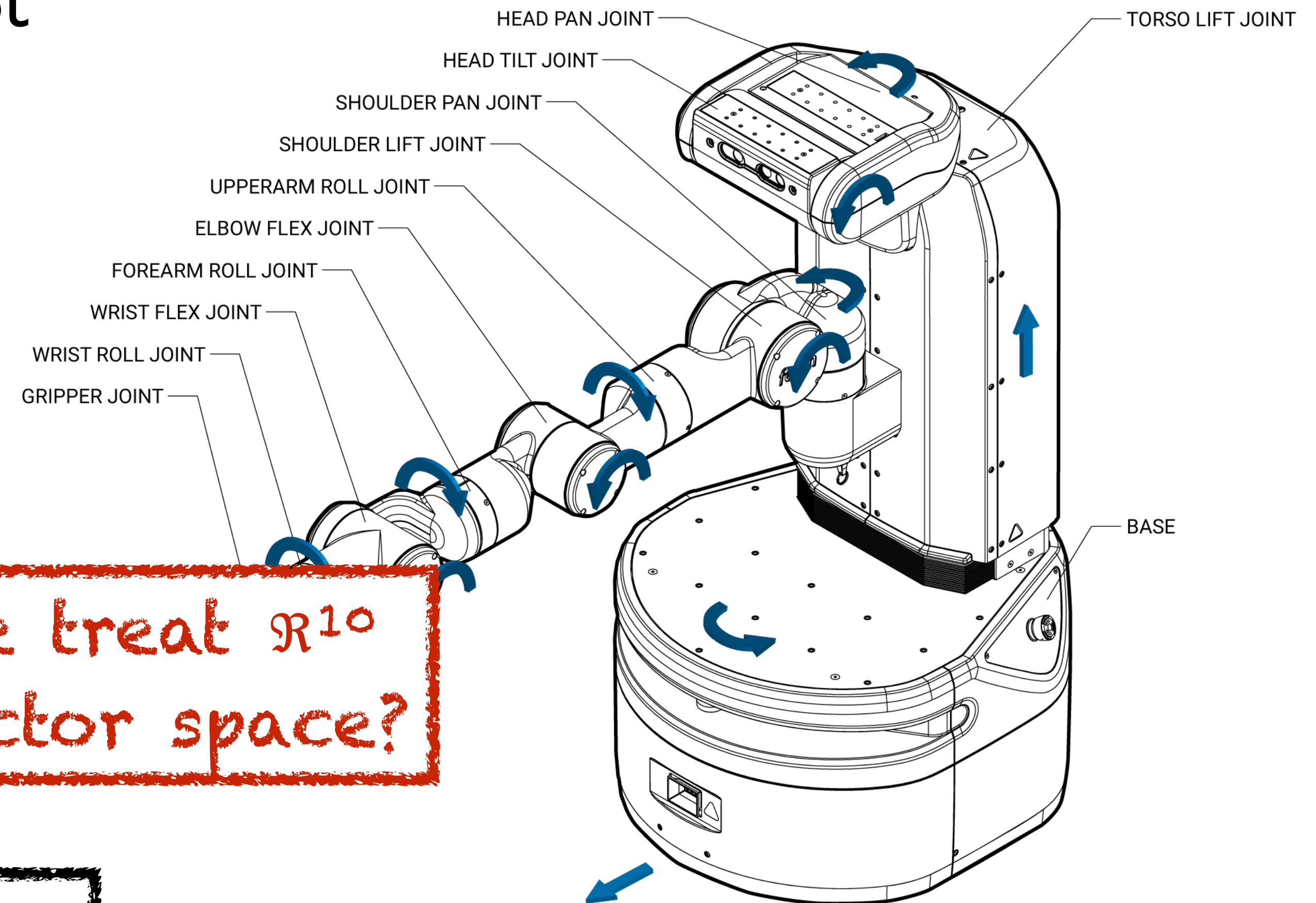


C-space with joint limits

- What is the C-space of a Fetch, not including grippers?
- DOFs: 13
 - 3 in base: $SE(2)$
 - 7 in arm: $\mathbb{R}^7$
 - 1 in the spine: $\mathbb{R}^1$
 - 2 in neck: $\mathbb{R}^2$

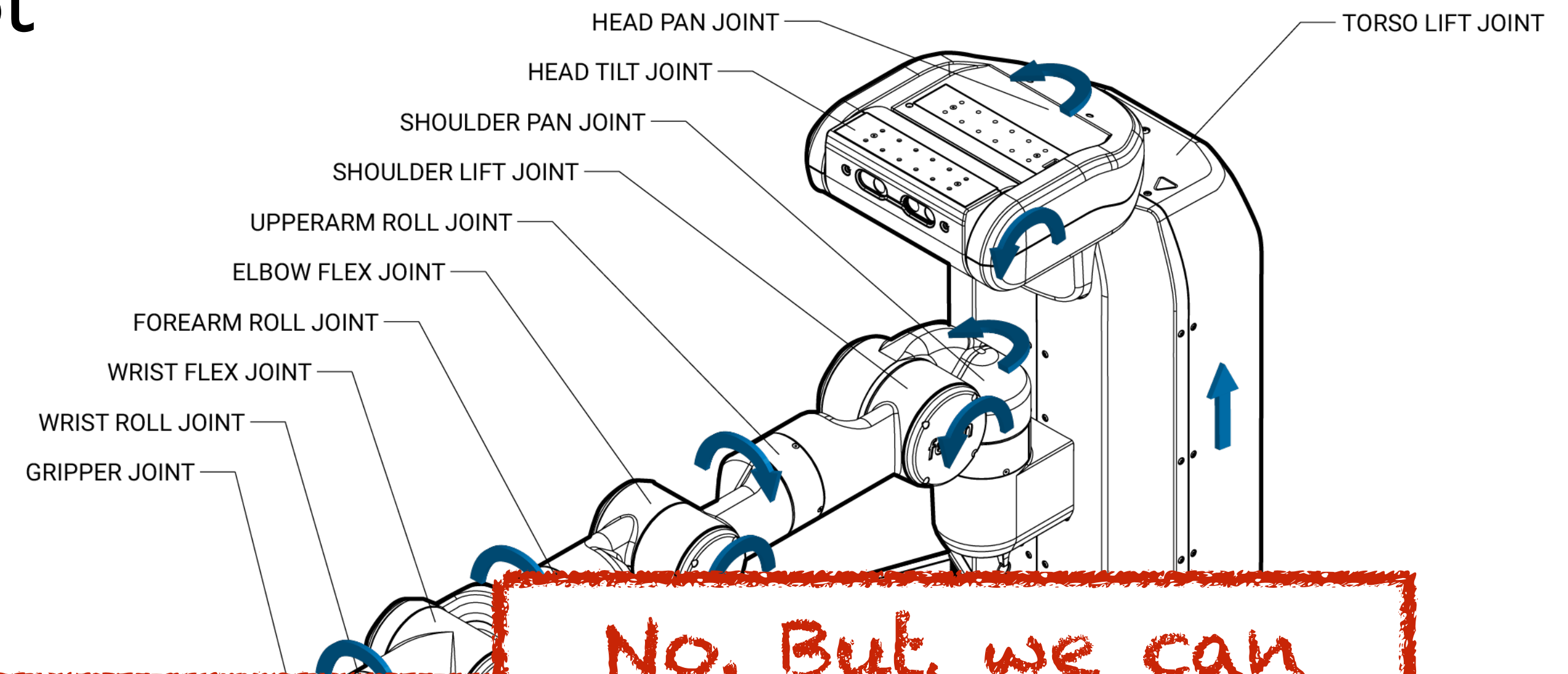
Can we treat $\mathbb{R}^{10}$ as a vector space?

C-space: $SE(2) \times \mathbb{R}^{10}$



C-space with joint limits

- What is the C-space of a Fetch, not including grippers?
- DOFs: 13
 - 3 in base: $SE(2)$
 - 7 in arm: $\mathbb{R}^7$
 - 1 in the spine: $\mathbb{R}^1$
 - 2 in neck: $\mathbb{R}^2$



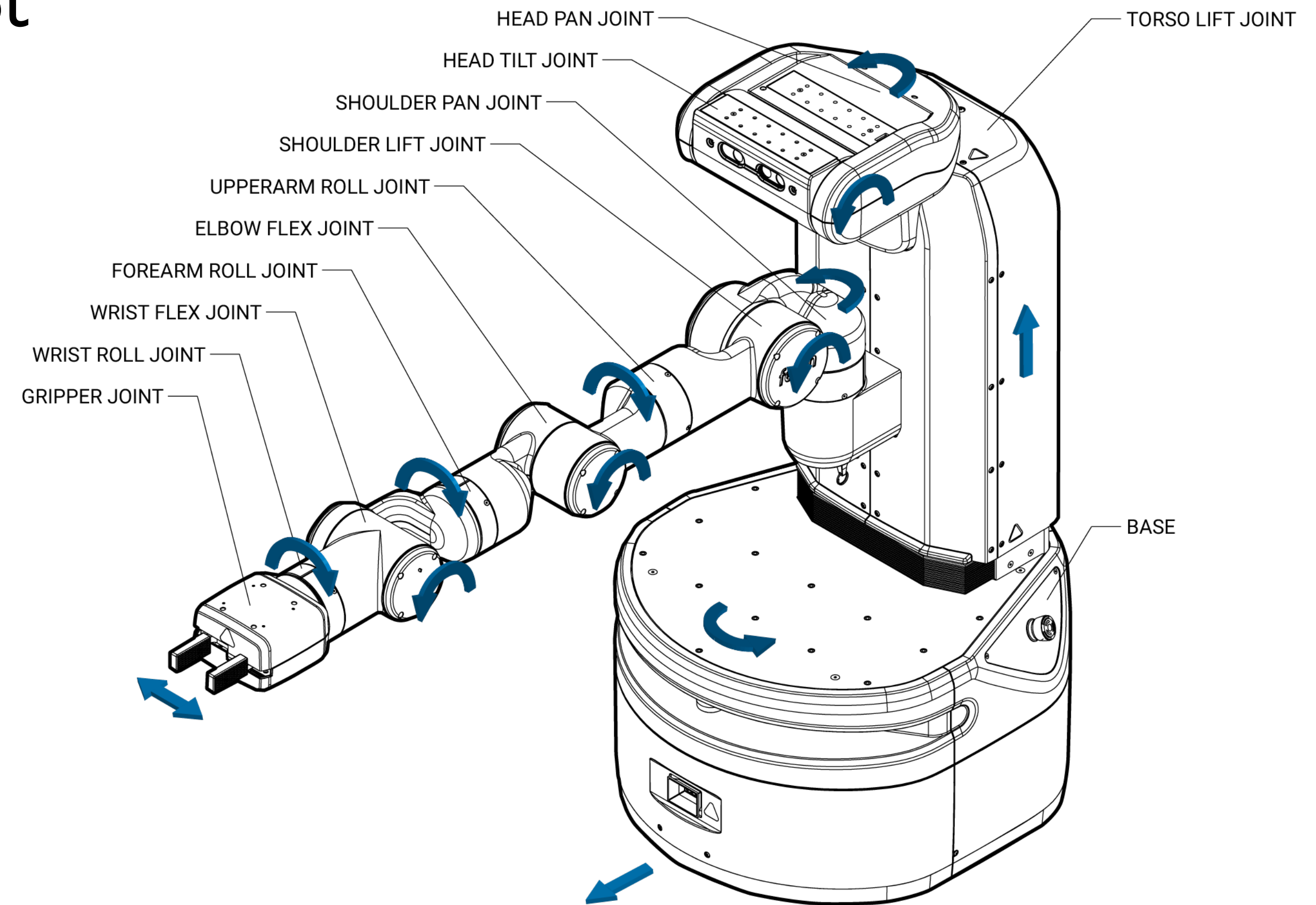
Can we treat $\mathbb{R}^{10}$ as a vector space?

No. But, we can sample C-space using vector operations

C-space: $SE(2) \times \mathbb{R}^{10}$

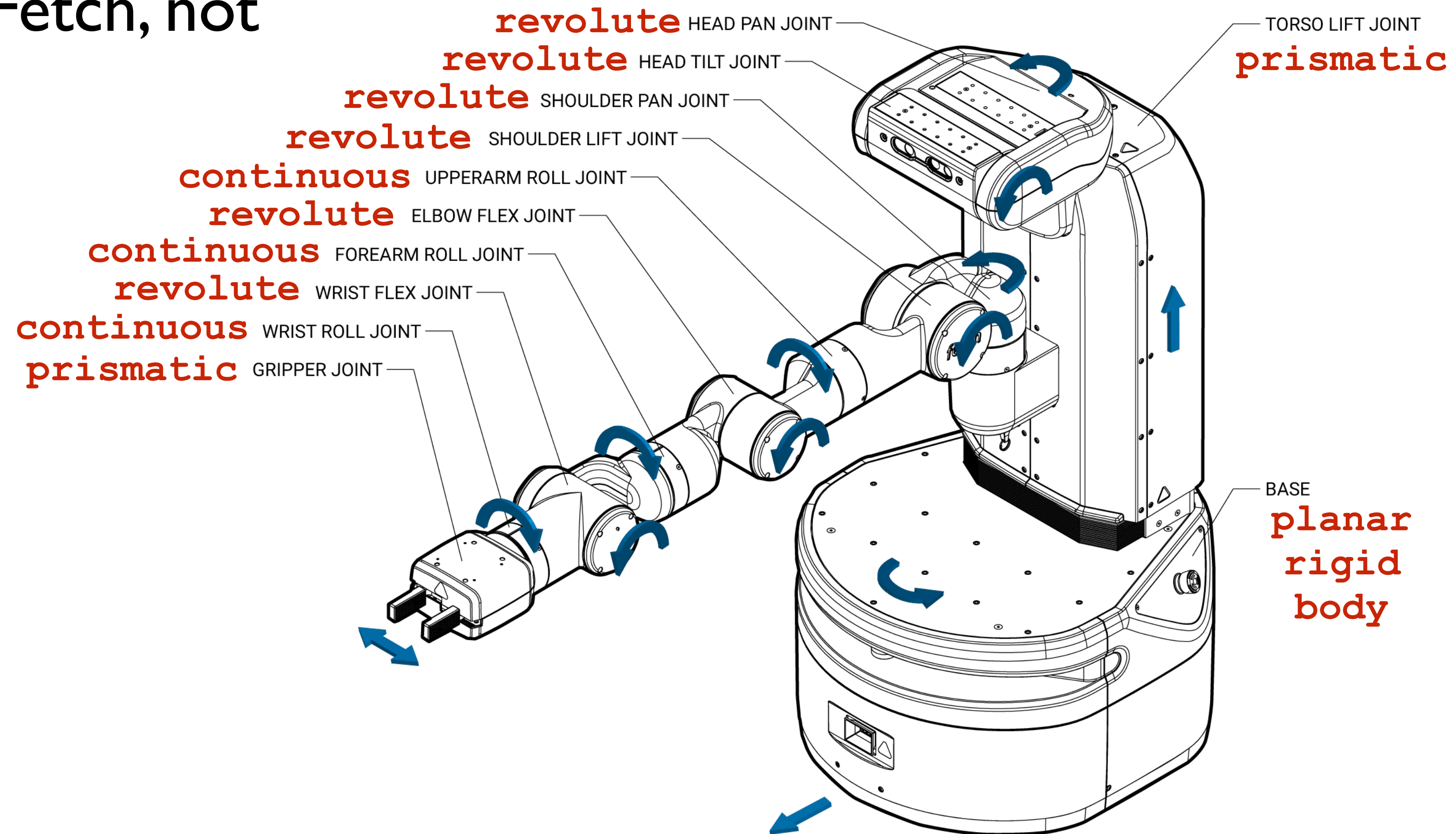
Still not quite right...

- What is the C-space of a Fetch, not including grippers?
- DOFs: 13
 - 3 in base: $SE(2)$
 - 7 in arm: ~~$\mathbb{R}^7$~~
 - 1 in the spine: $\mathbb{R}^1$
 - 2 in neck: $\mathbb{R}^2$

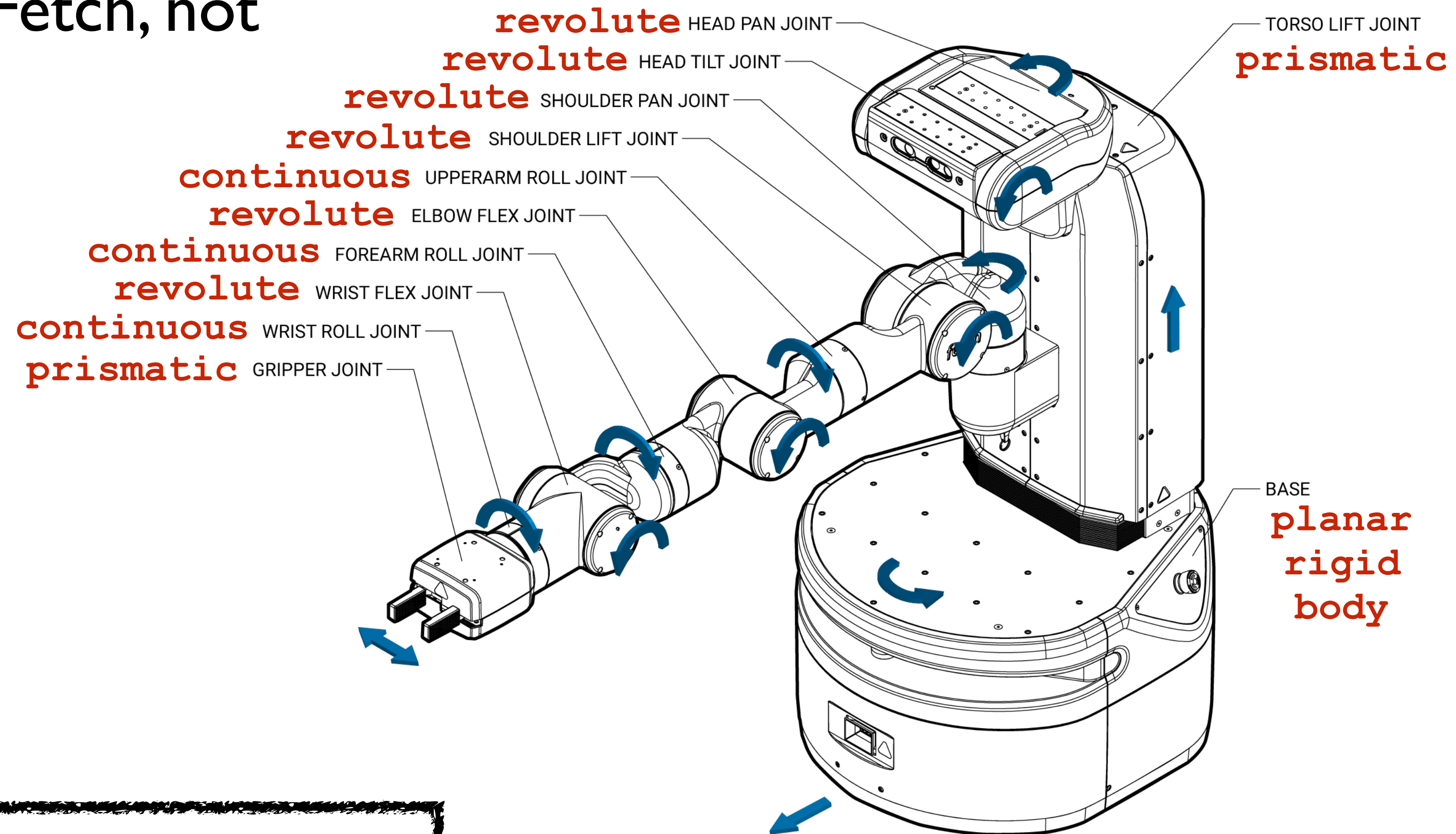


Still not quite right...

- What is the C-space of a Fetch, not including grippers?
- DOFs: 13
 - 3 in base: $SE(2)$
 - 7 in arm: ~~$\mathbb{R}^7$~~
 - 1 in the spine: $\mathbb{R}^1$
 - 2 in neck: $\mathbb{R}^2$



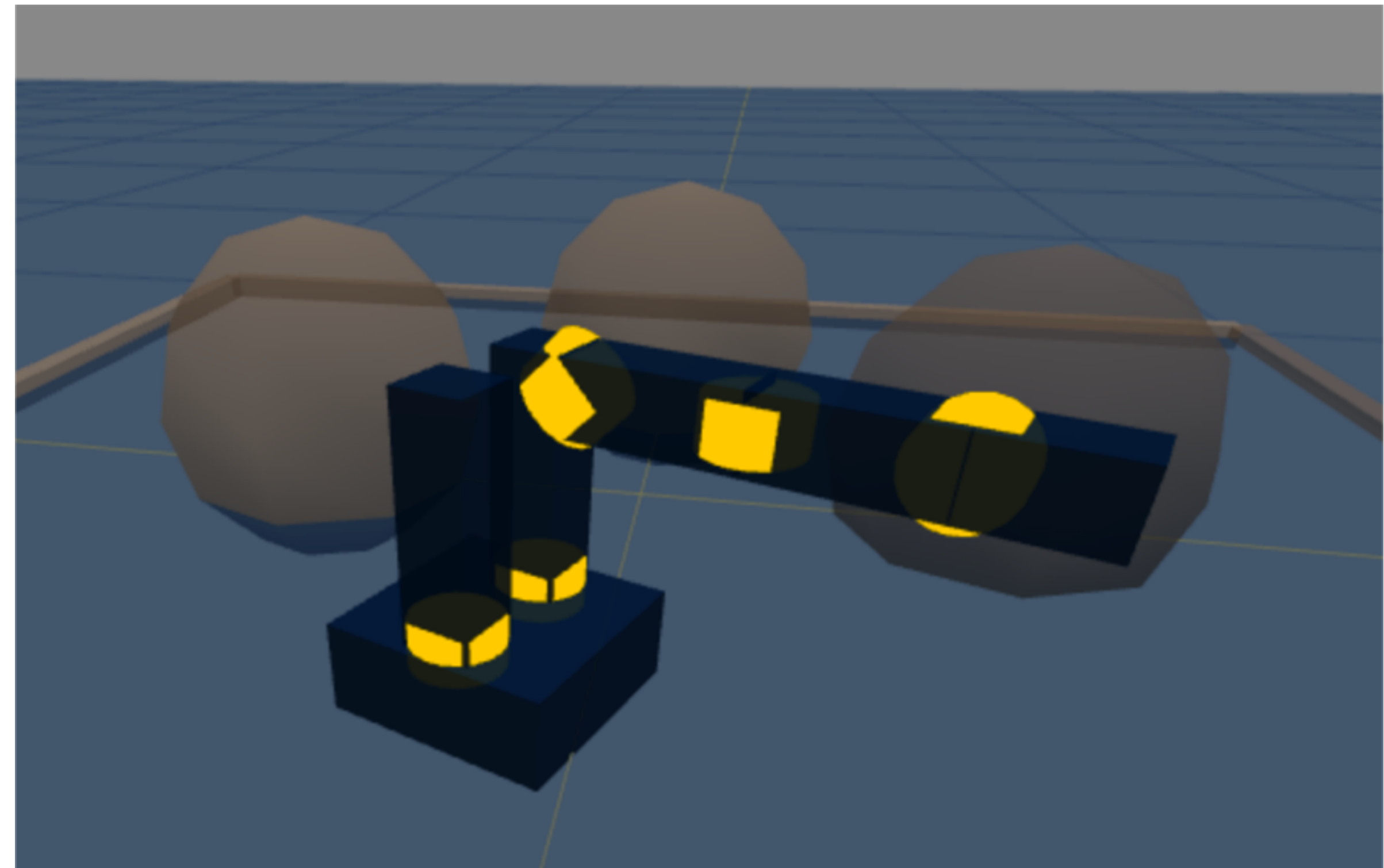
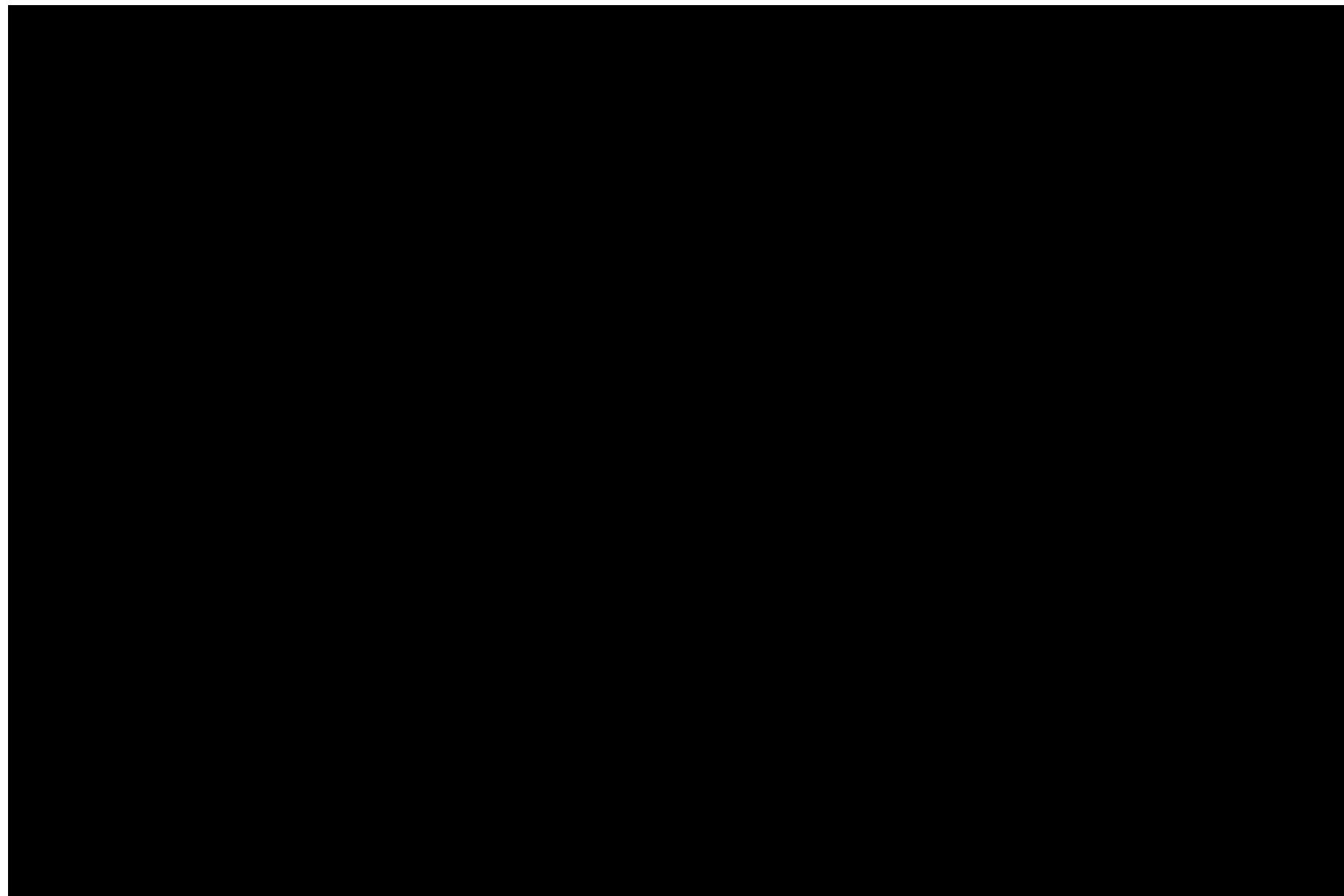
- What is the C-space of a Fetch, not including grippers?
- DOFs: 13
 - 3 in base: $SE(2)$
 - 3 continuous: T^3
 - 1 prismatic: $\mathbb{R}^1$
 - 6 revolute: $\mathbb{R}^6$



C-space: $SE(2) \times T^3 \times \mathbb{R}^7$

C-space examples

- What is the C-space of a MR2?

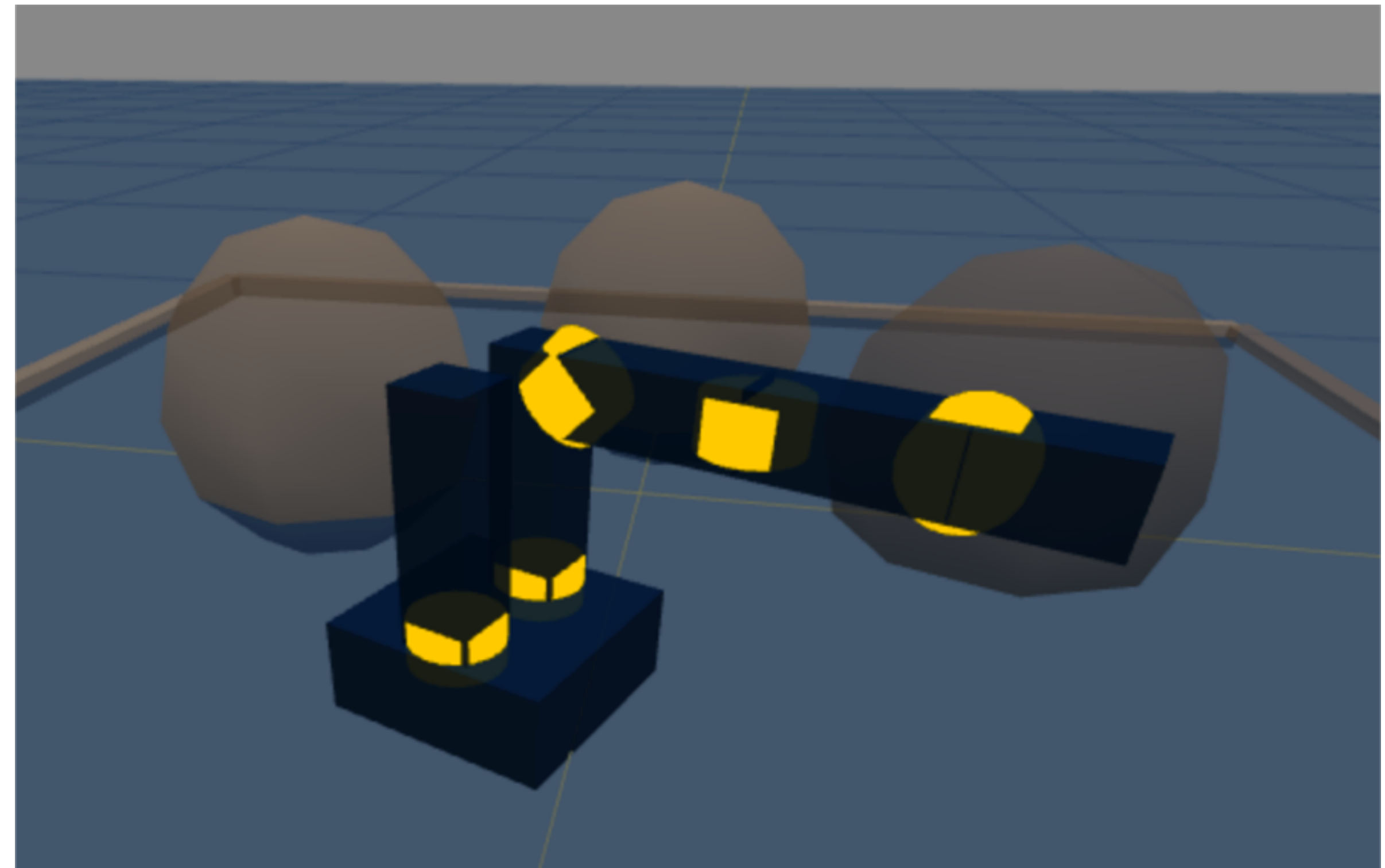


C-space examples

- What is the C-space of a MR2?
- DOFs: 8
 - 3 in base: $SE(2)$
 - 5 in arms: T^5

⊕ C-space: $SE(2) \times T^5$

Can we treat $SE(2) \times T^5$
as a vector space?



C-space examples

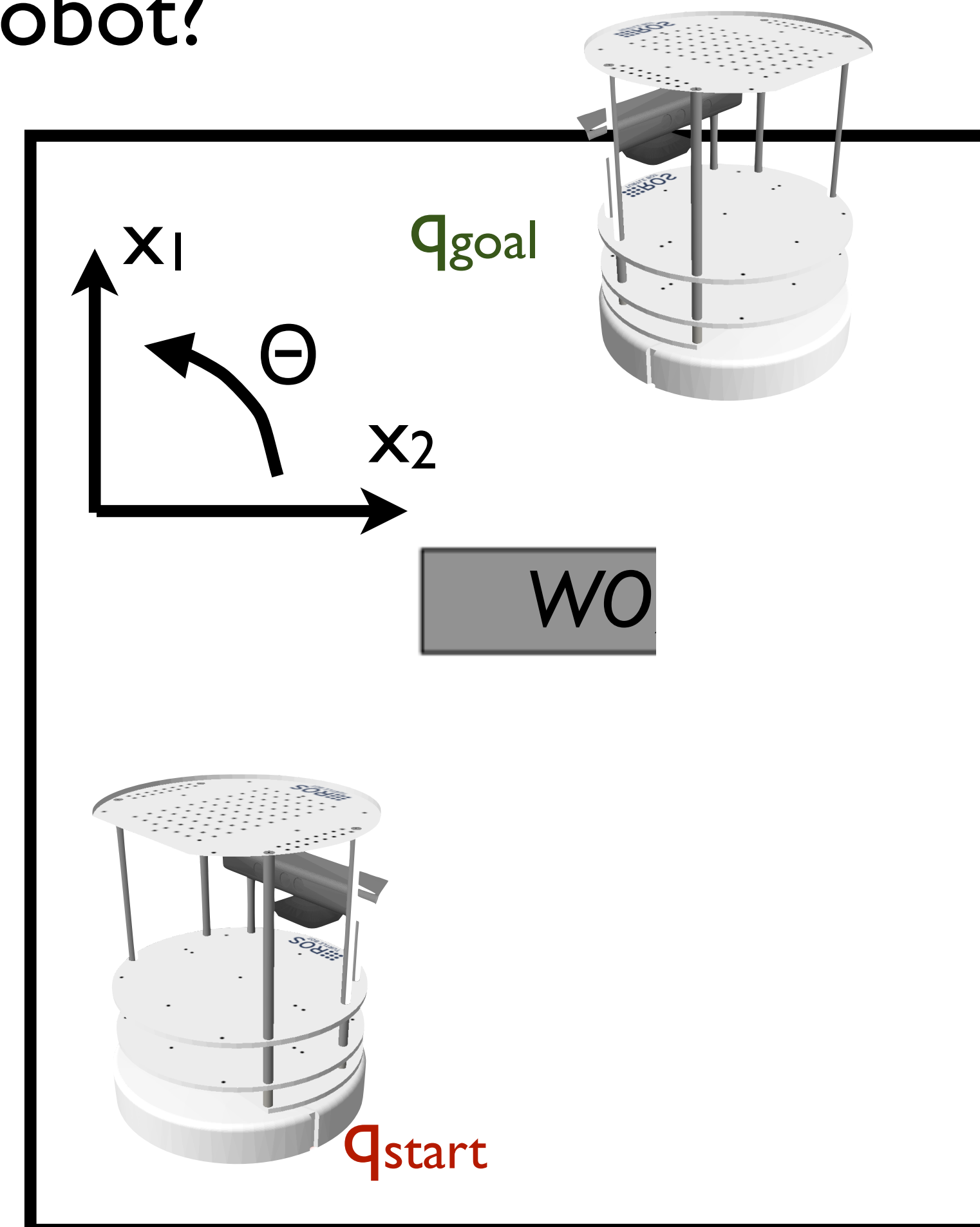
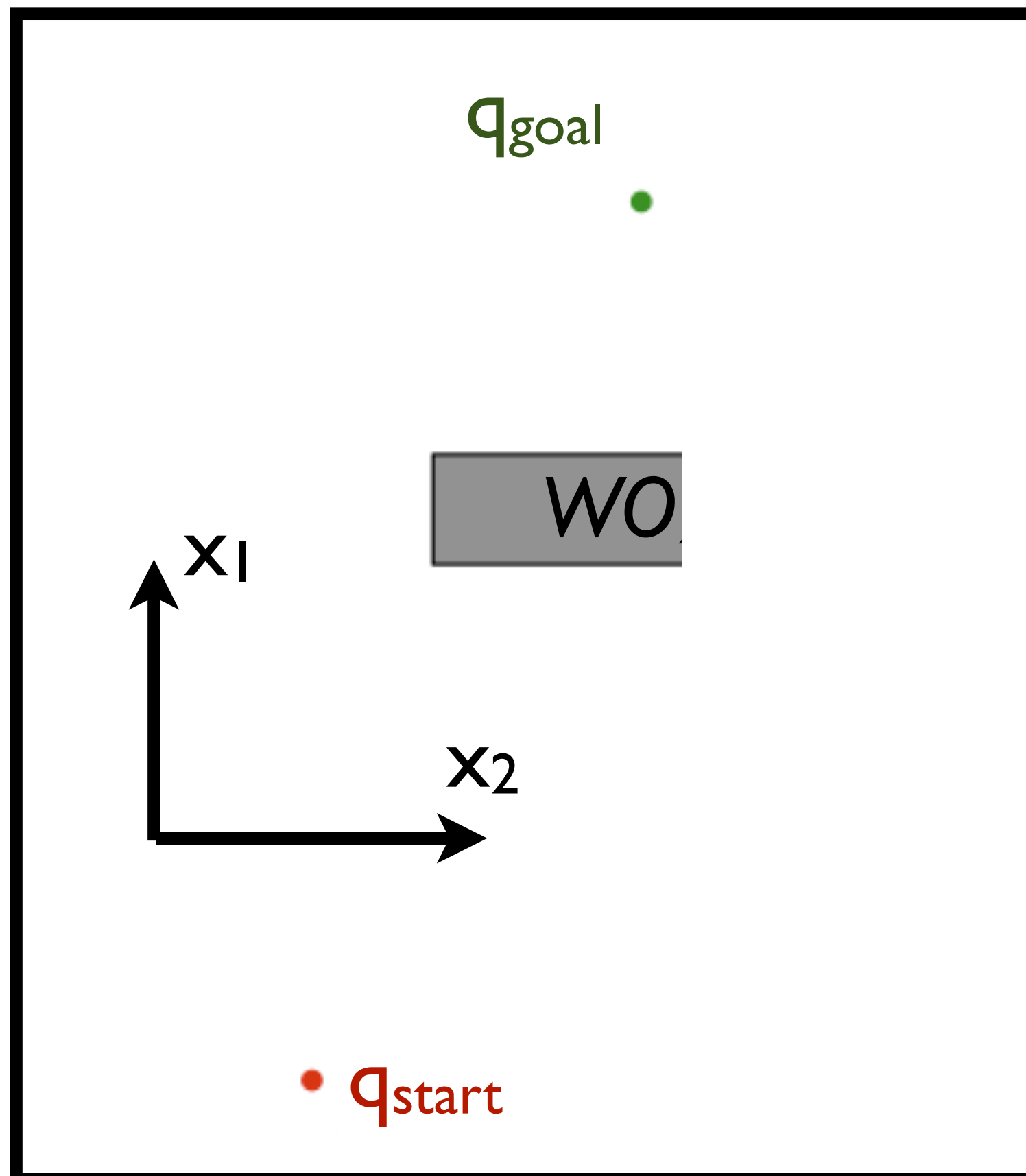
- What is the C-space of a Robonaut 2 on the International Space Station?
- What is the C-space of a PR2?



What about the robot's
physical geometry?

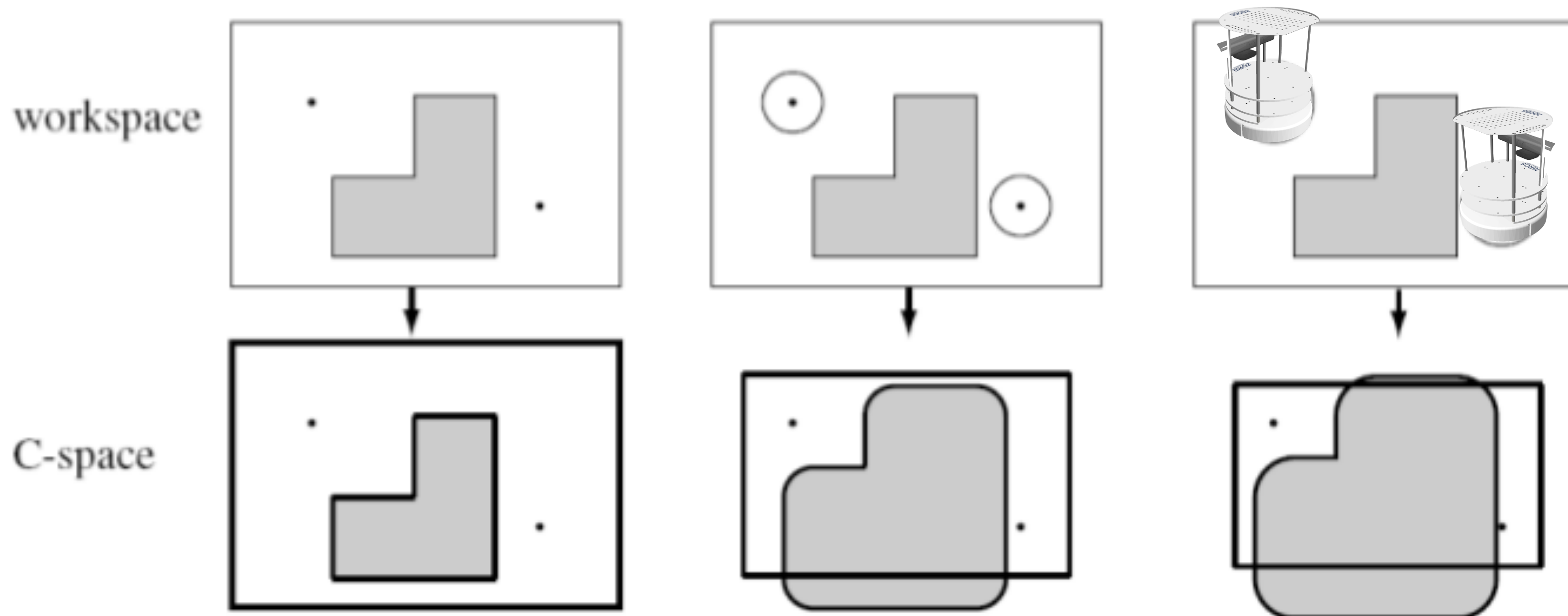
Configuration v. Workspaces

- Other than rotation, how is the Turtlebot different than the point robot?



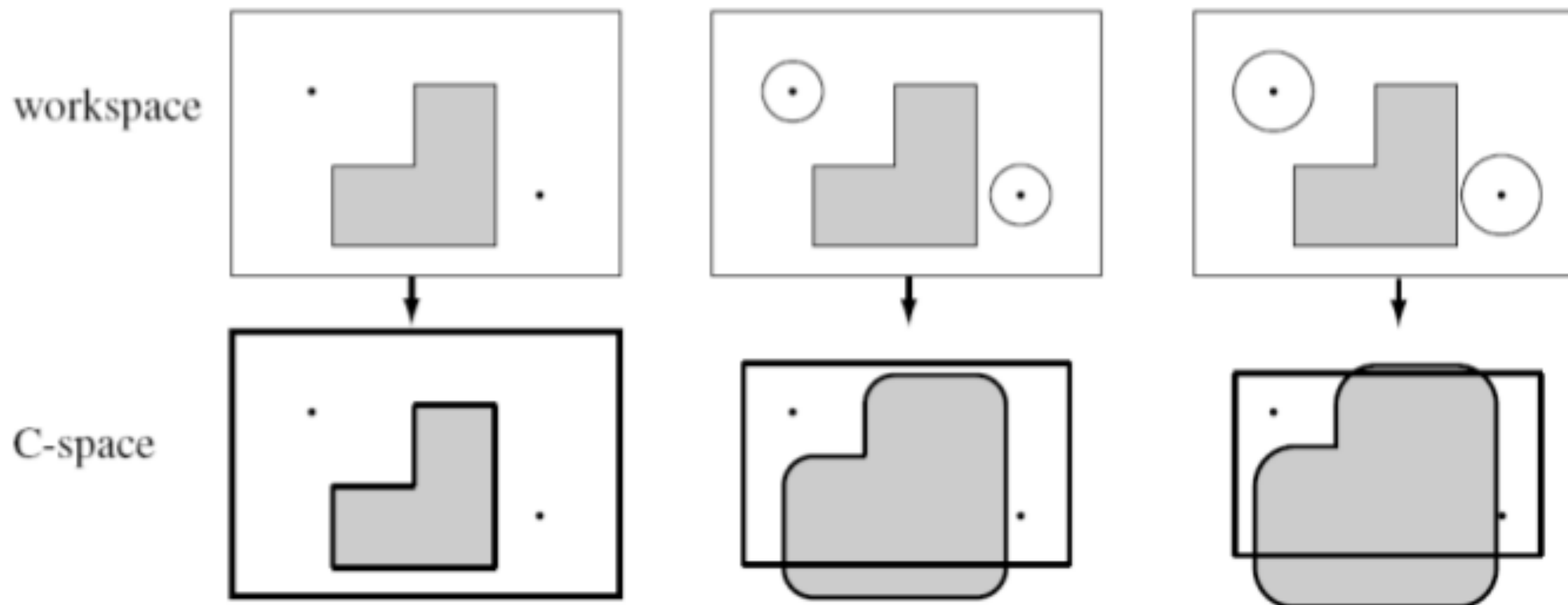
Robot Geometry

- Turtlebot is larger than a point, having a circular radius in the robot's planar workspace
- As this radius increases, the C-space shrinks

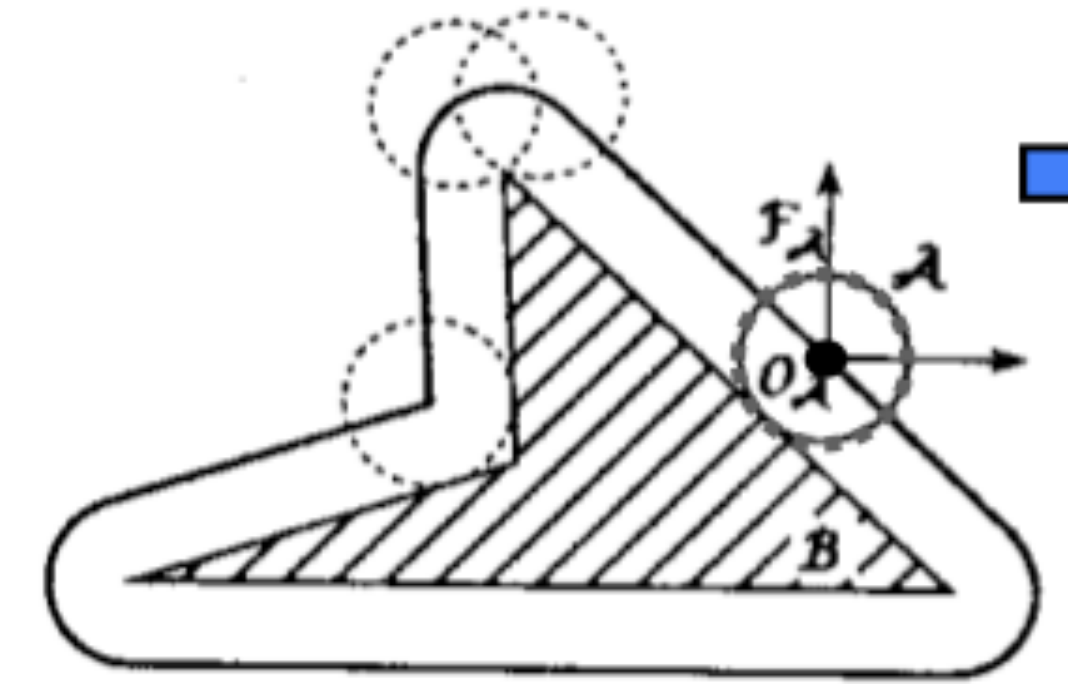


Robot Geometry

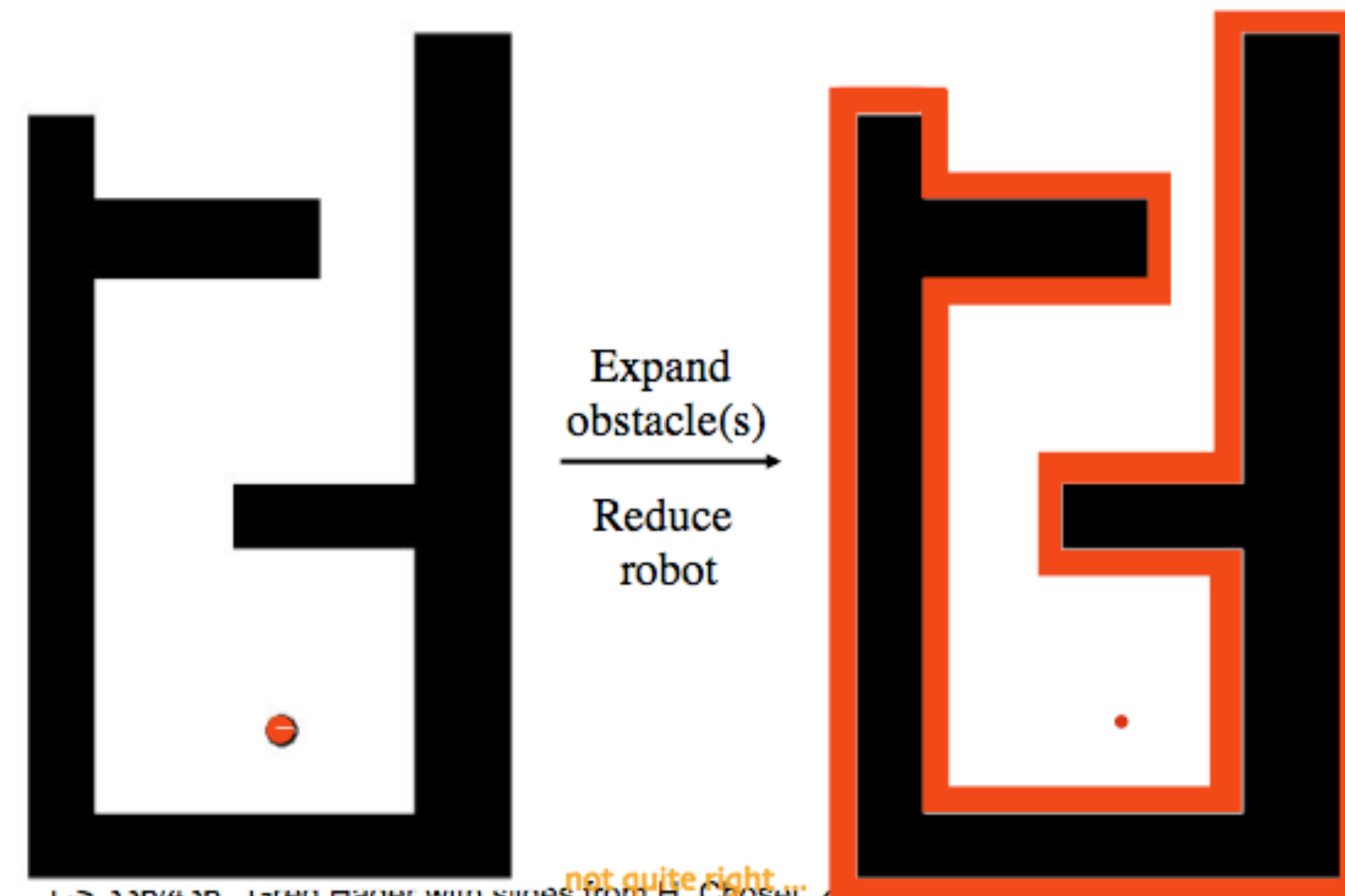
- Turtlebot is larger than a point, having a circular radius in the robot's planar workspace
- As this radius increases, the C-space shrinks



Conversion to point robot C-space



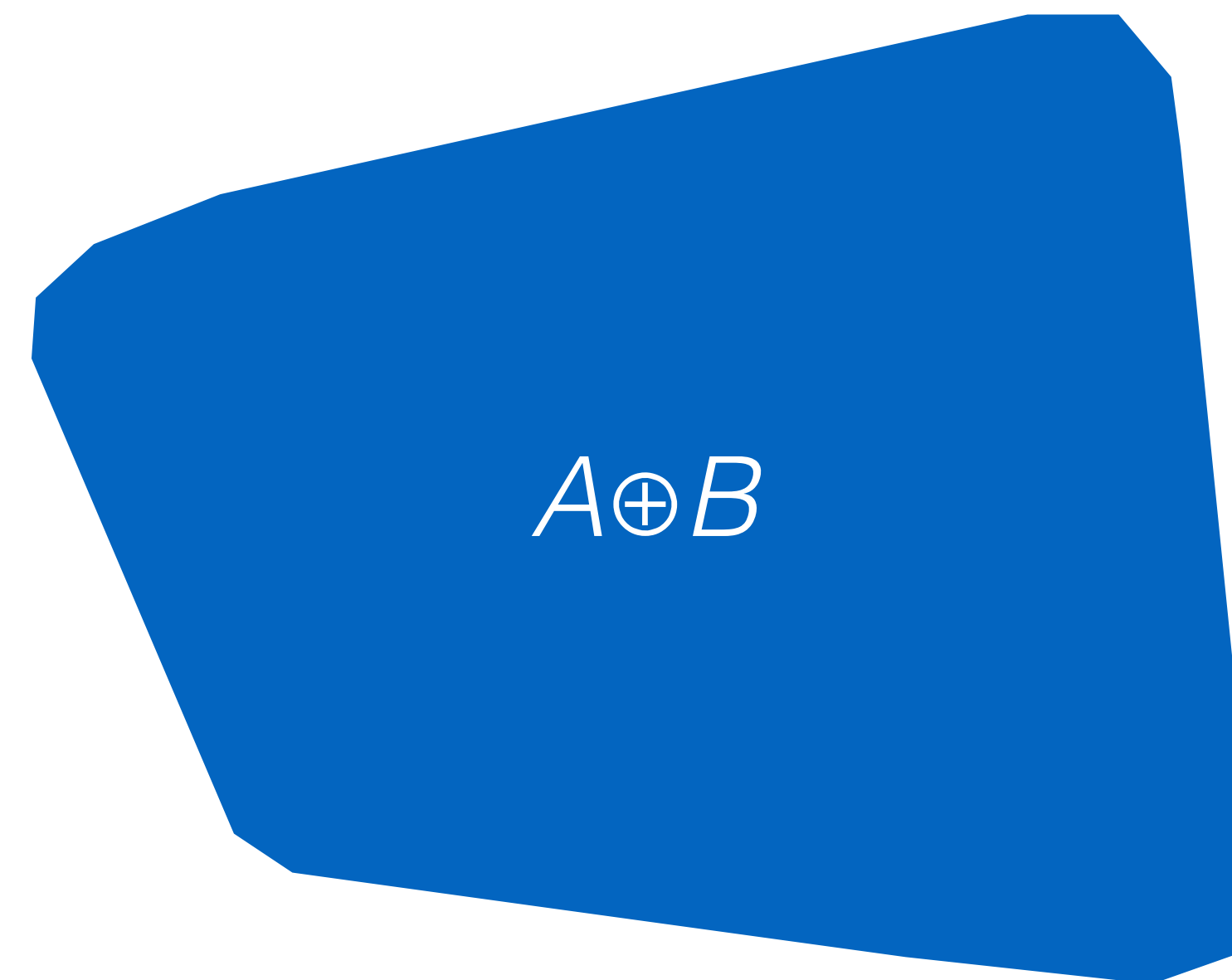
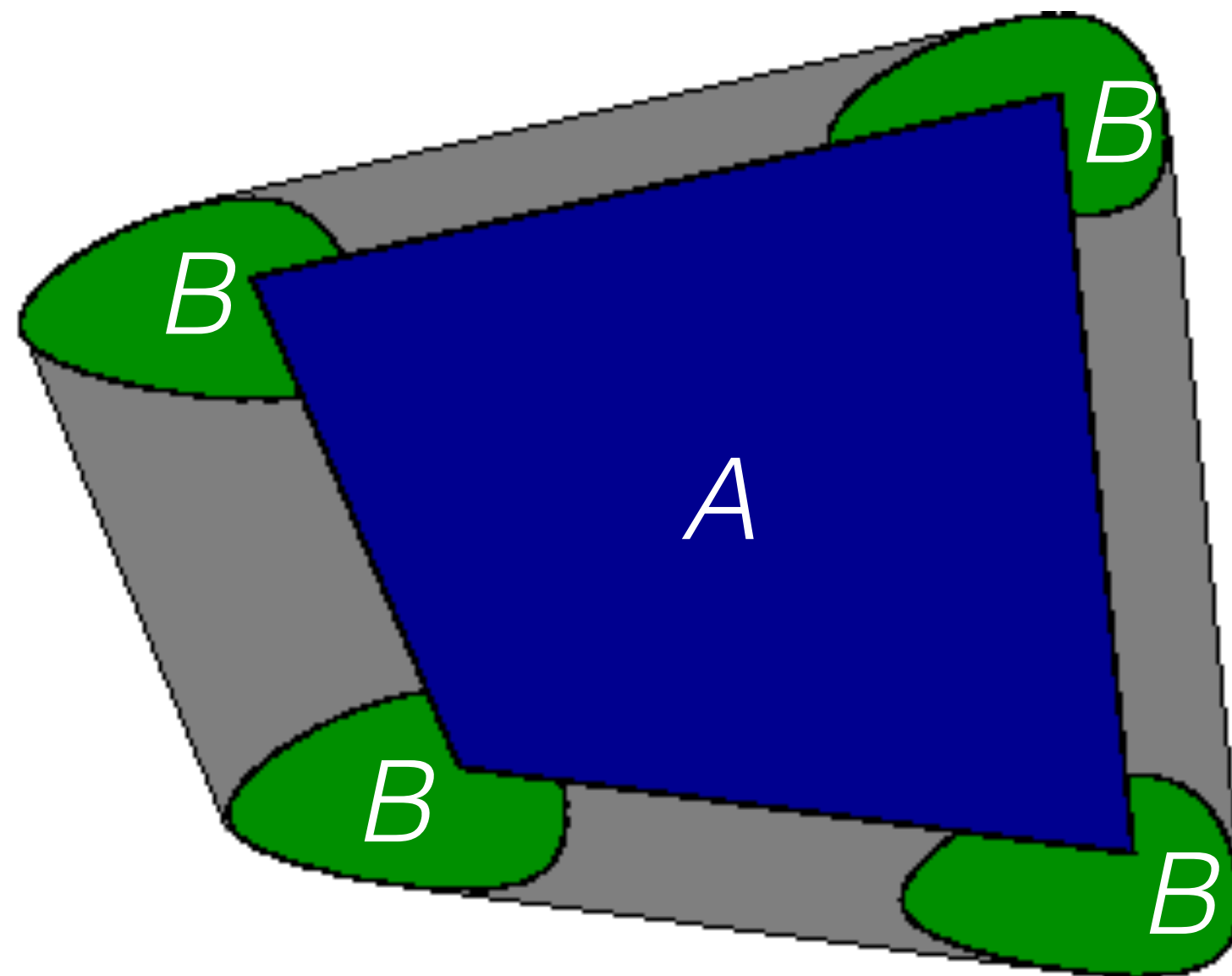
- Workspace for robot can be converted to point robot C-space
- Expand obstacles by tracing robot geometry along boundary
- Computable by Minkowski sum



CS 330/430, Greg Hager with slides from H. Choset, Z. Bouadi, and Lincoln Stein

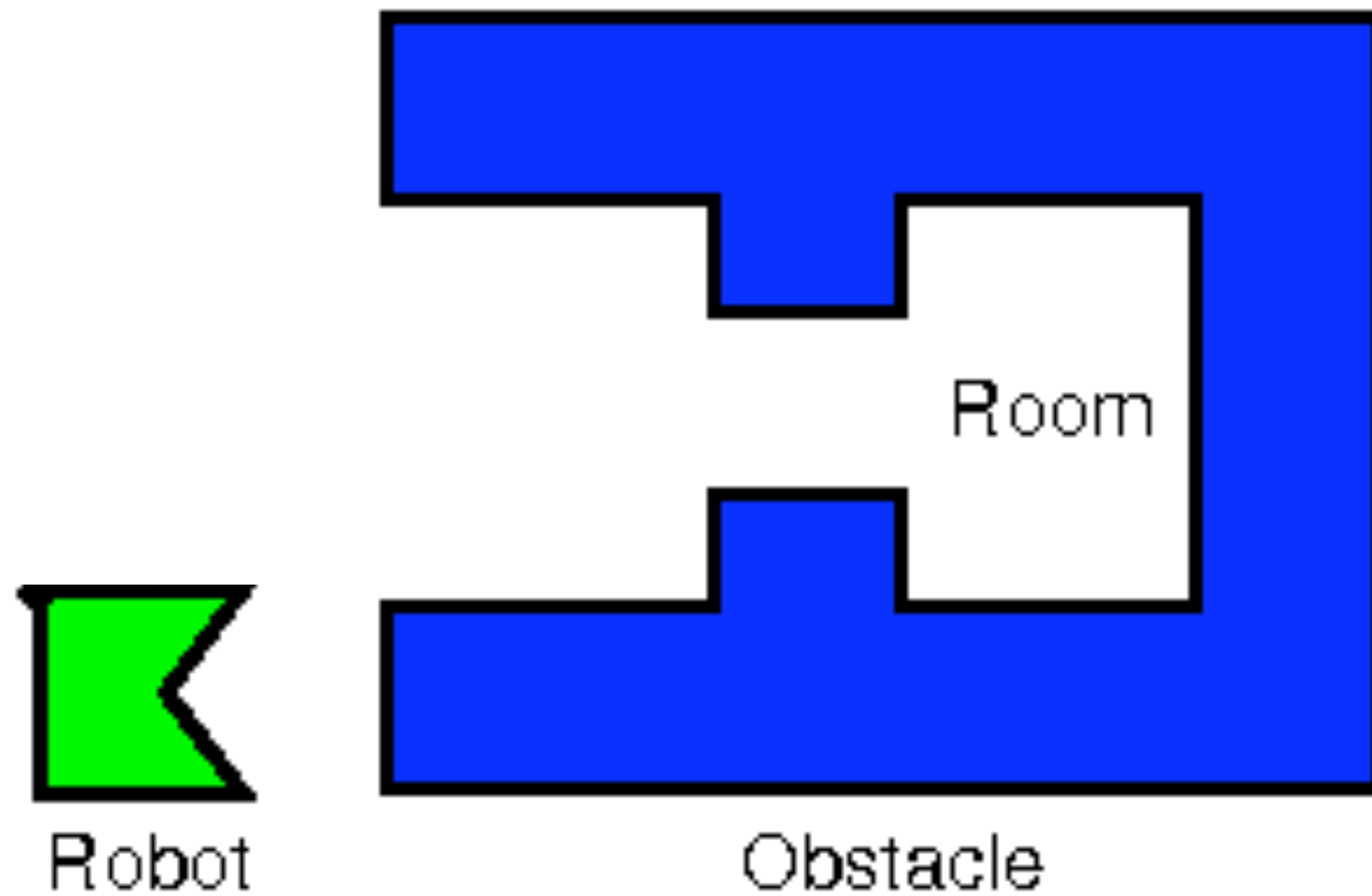
Minkowski Sum (or morphological dilation)

- $A \oplus B: \{a+b \mid a \in A, b \in B\}$
 - The result of adding every element of A to every element of B , given two sets A and B in Euclidean space
- Similarly, Minkowski difference (morphological erosion)
 - $A \ominus B: \{a-b \mid a \in A, b \in B\}$

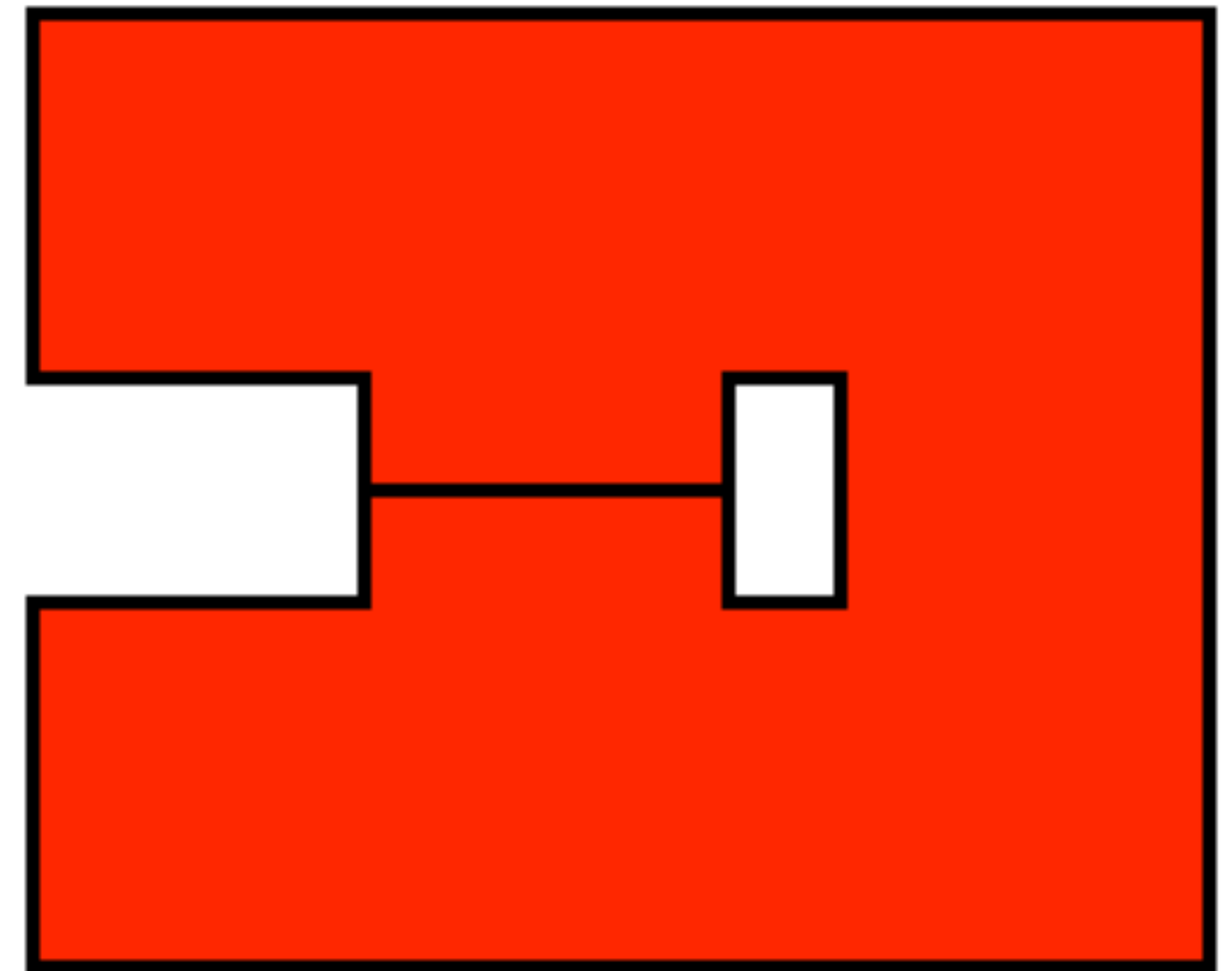


Minkowski Planning

Given

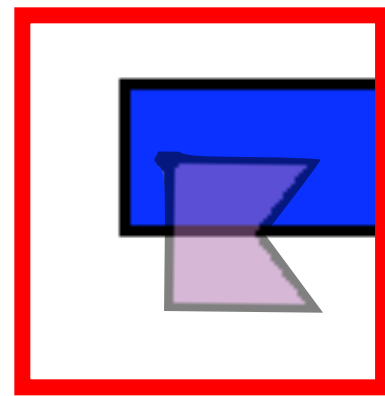


Compute

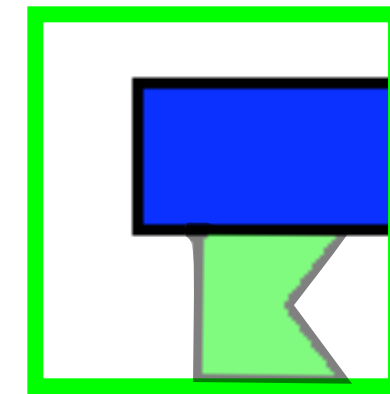


Minkowski sum

Minkowski Planning

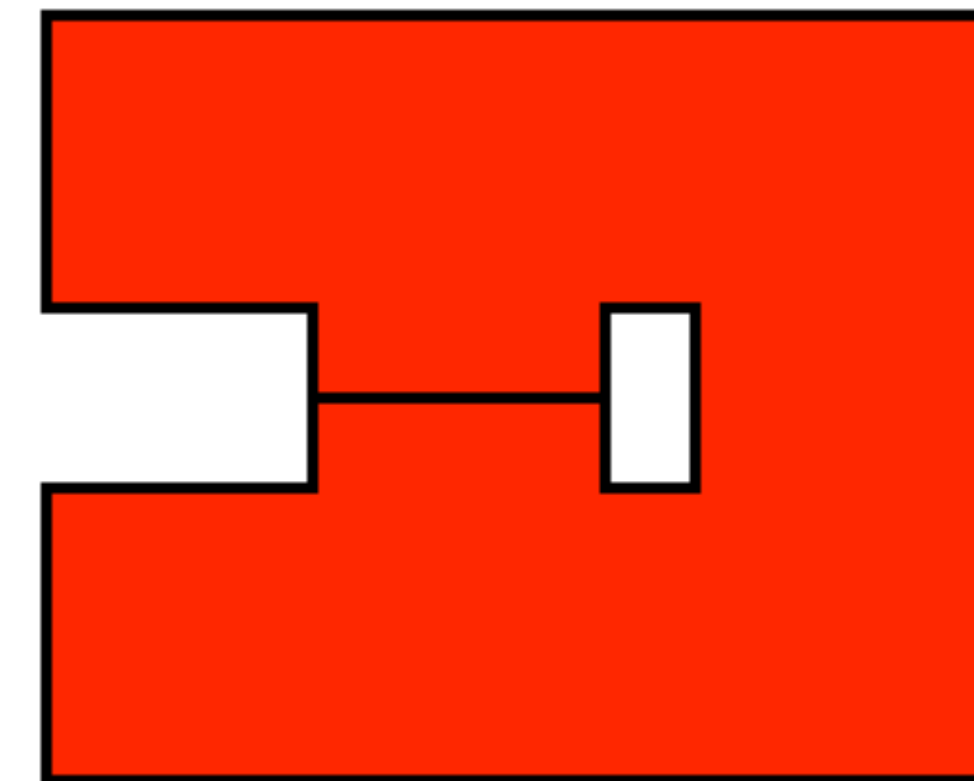
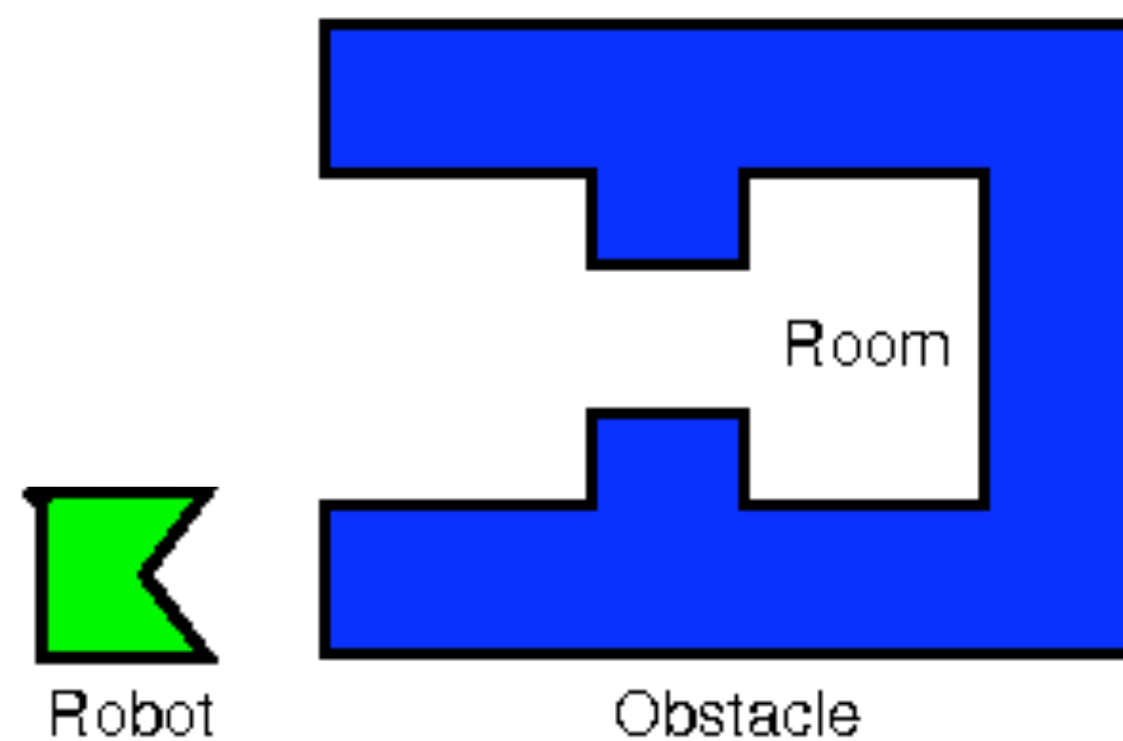


Occlude location if robot intersects obstacle

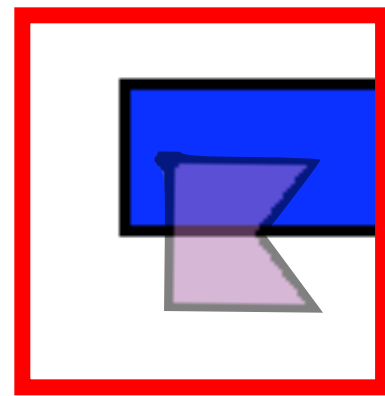


Leave location free if robot non-intersecting with object

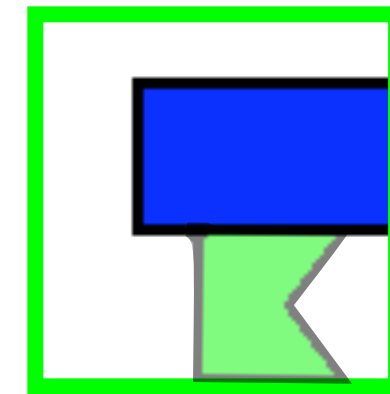
Minkowski sum: identify non intersecting robot locations



Minkowski Planning

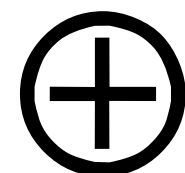


Occlude location if robot intersects obstacle

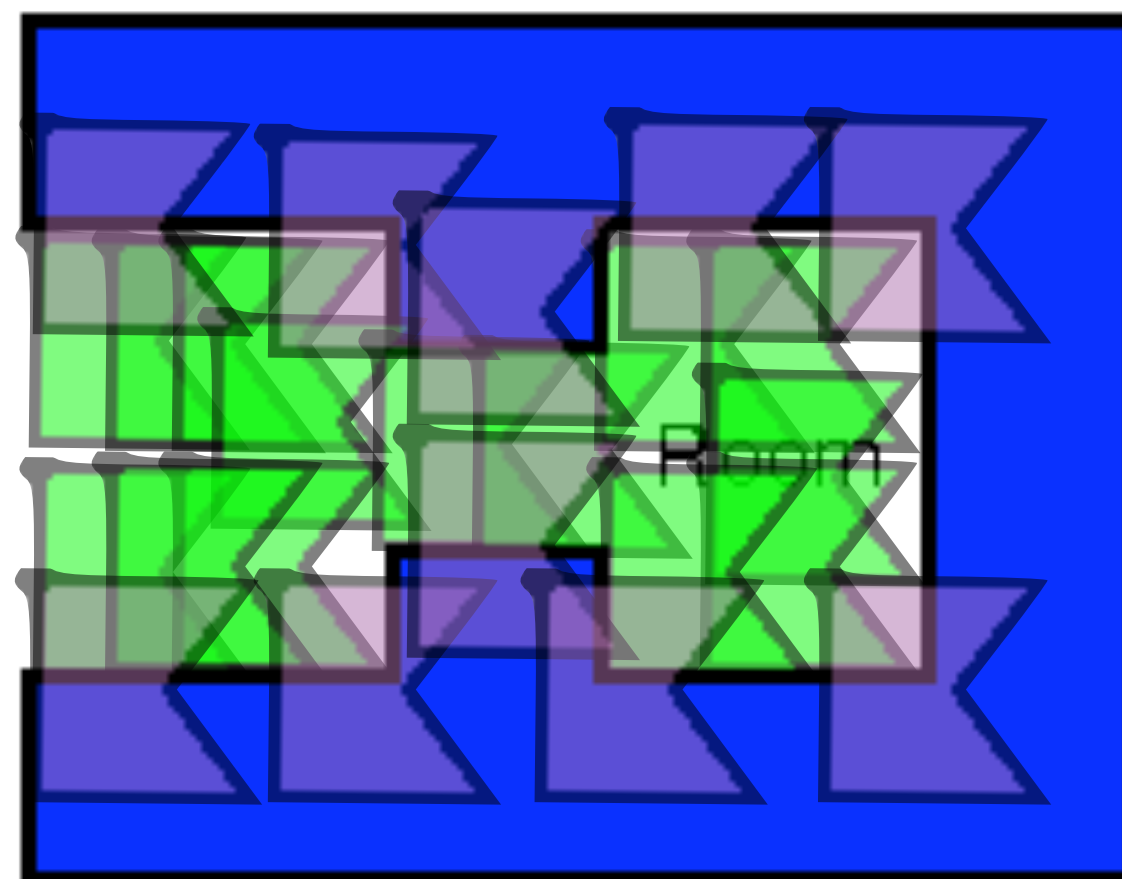


Leave location free if robot non-intersecting with object

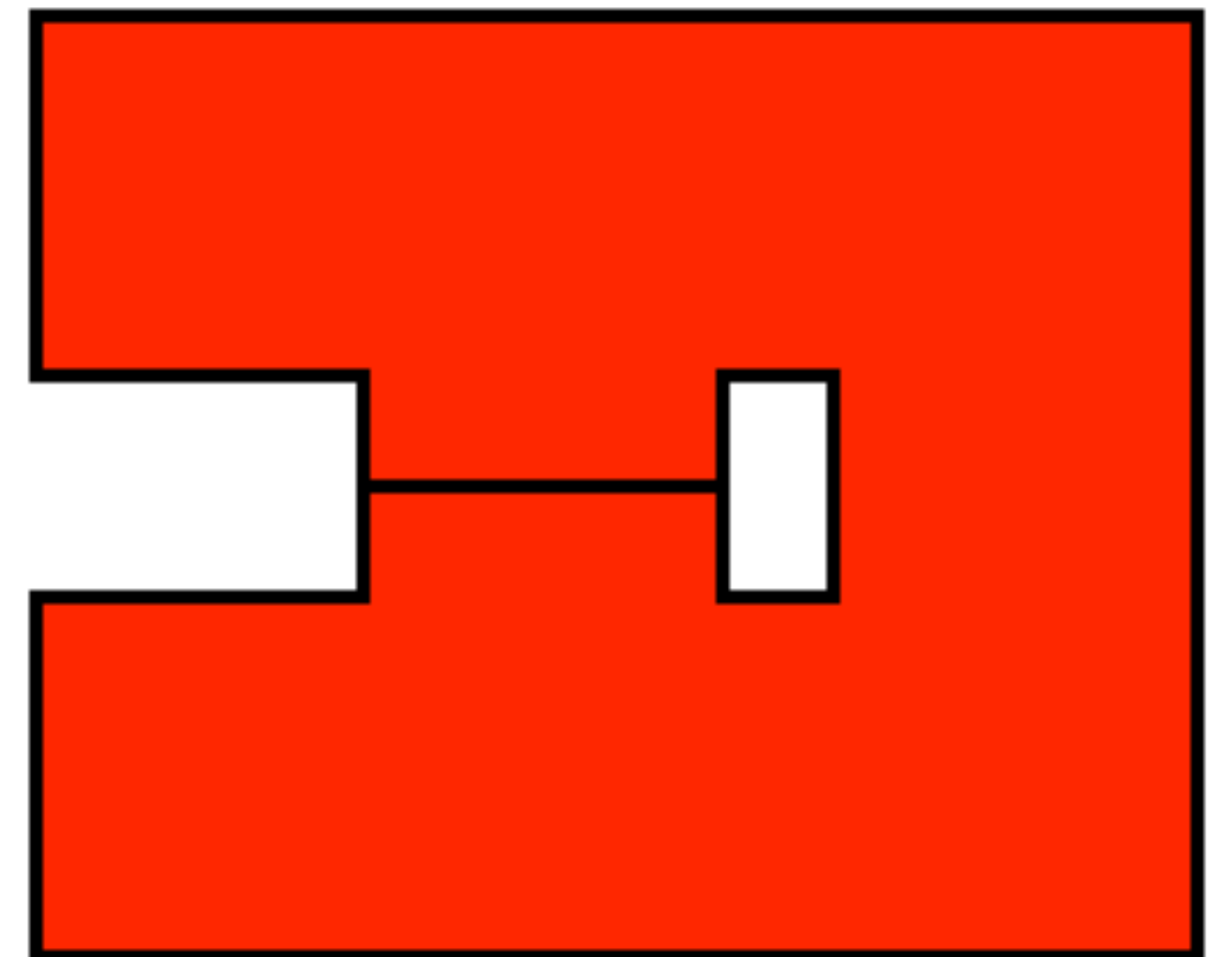
Minkowski sum: identify non intersecting robot locations



Robot geometry

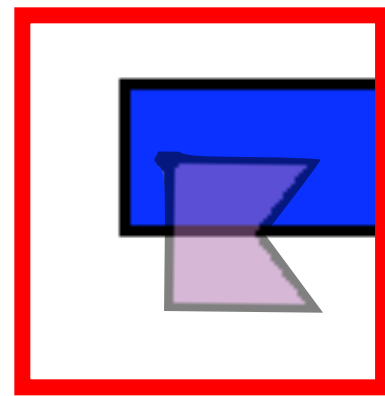


=

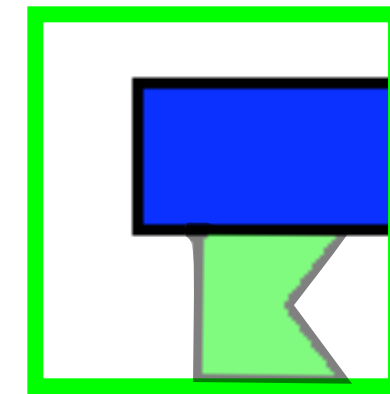


Space of valid paths defined by Minkowski sum
Michigan Robotics 367/320 - autorob.org

Minkowski Planning

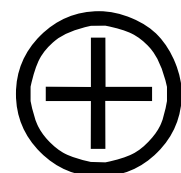


Occlude location if robot intersects obstacle

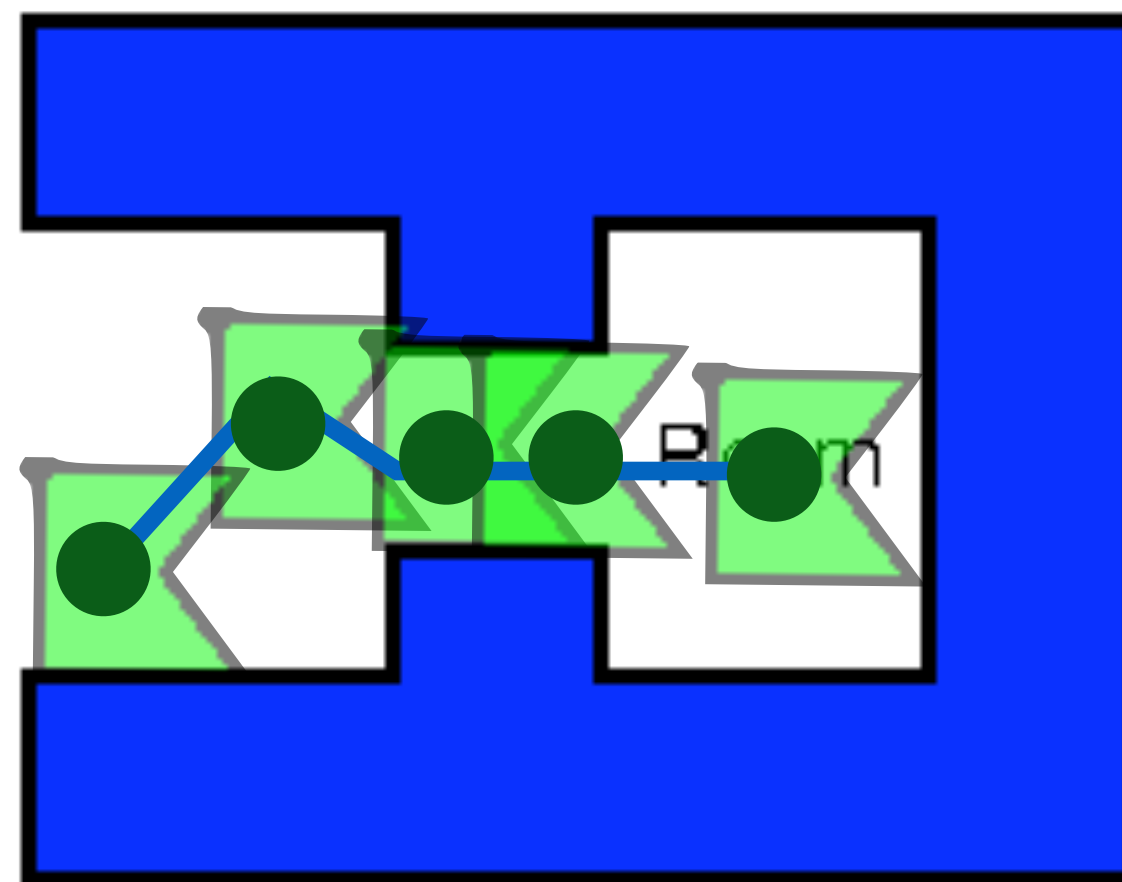


Leave location free if robot non-intersecting with object

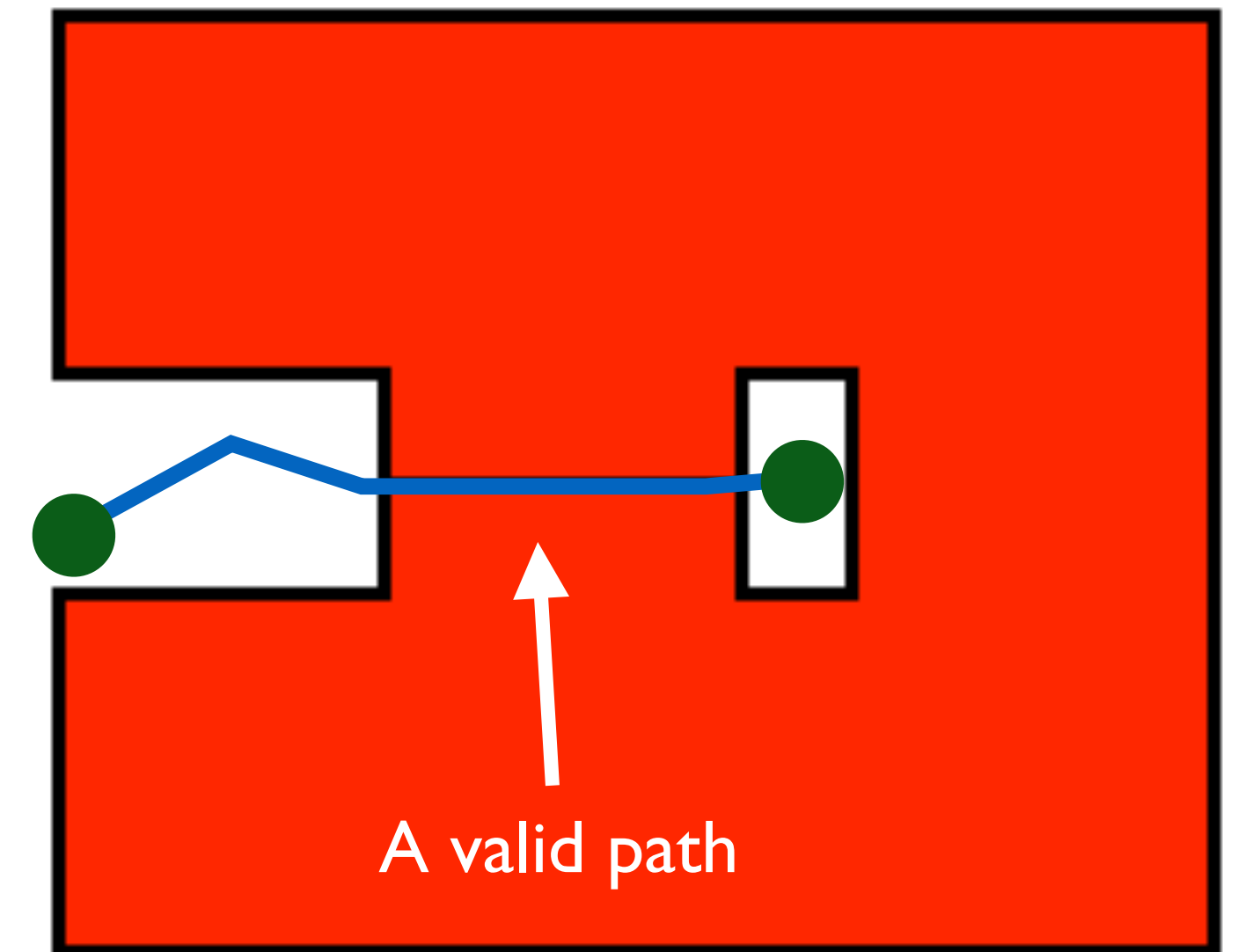
Minkowski sum: identify non intersecting robot locations



Robot geometry



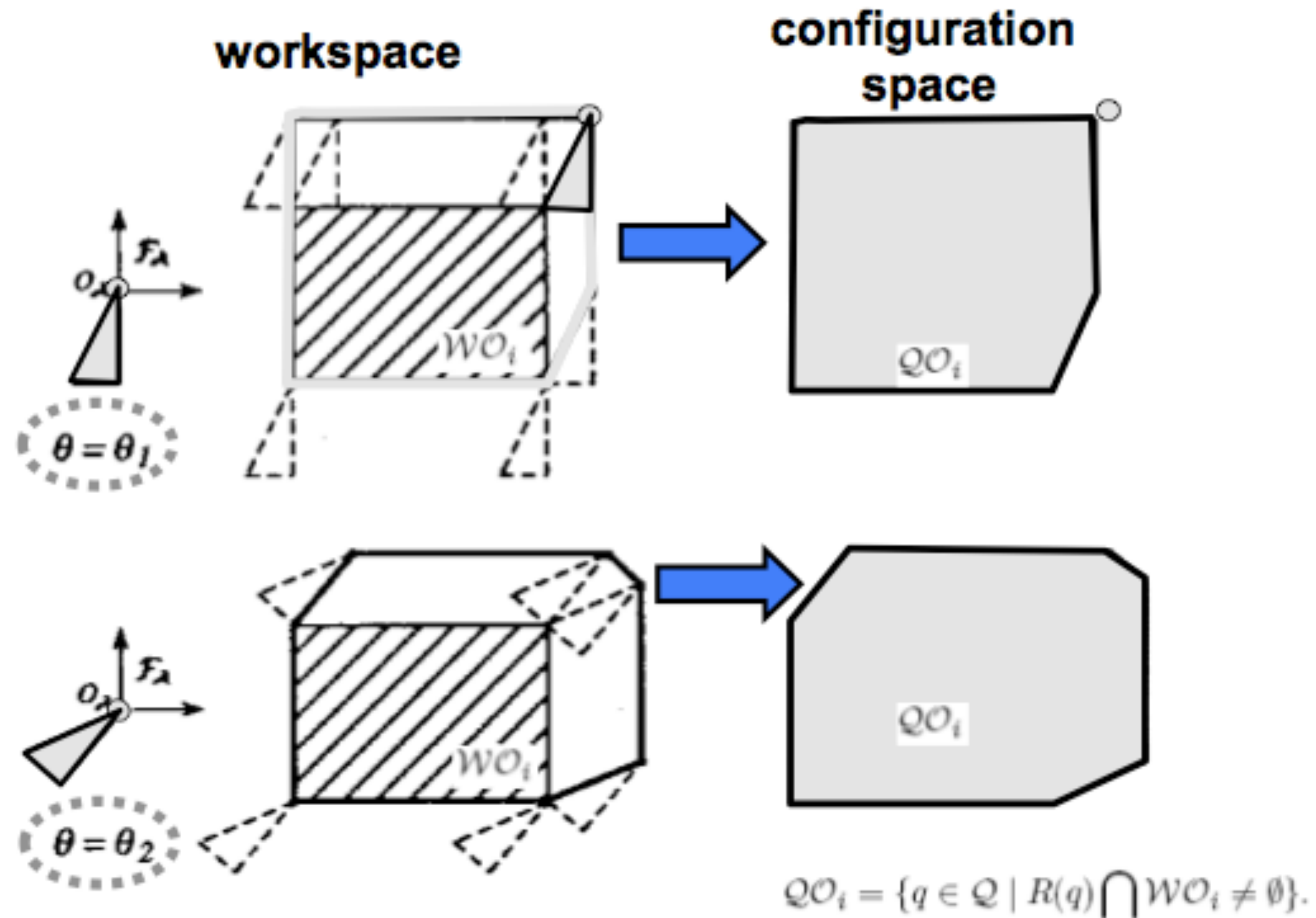
=



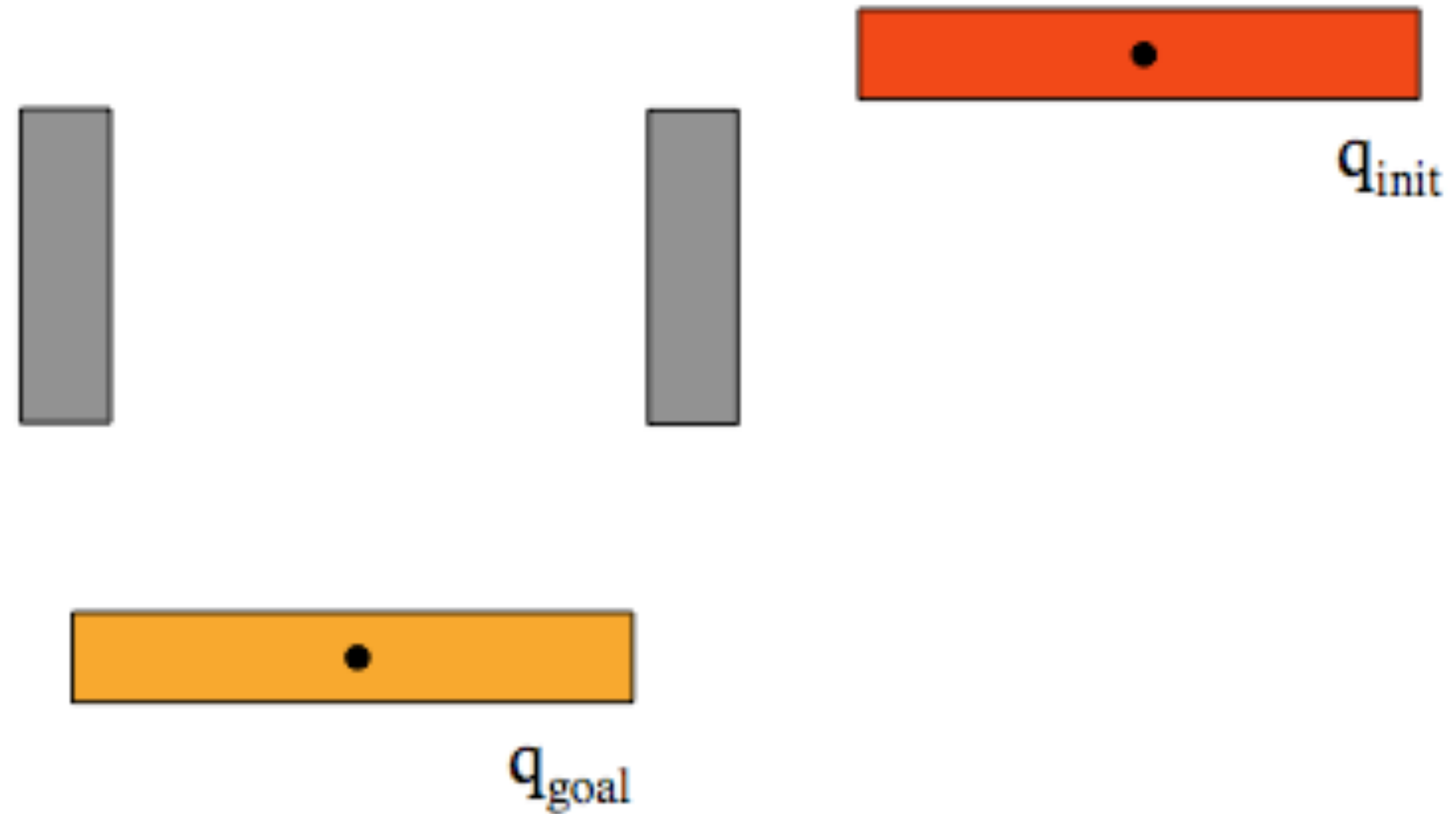
Space of valid paths defined by Minkowski sum
Michigan Robotics 367/320 - autorob.org

What does an obstacle look like
in configuration space?

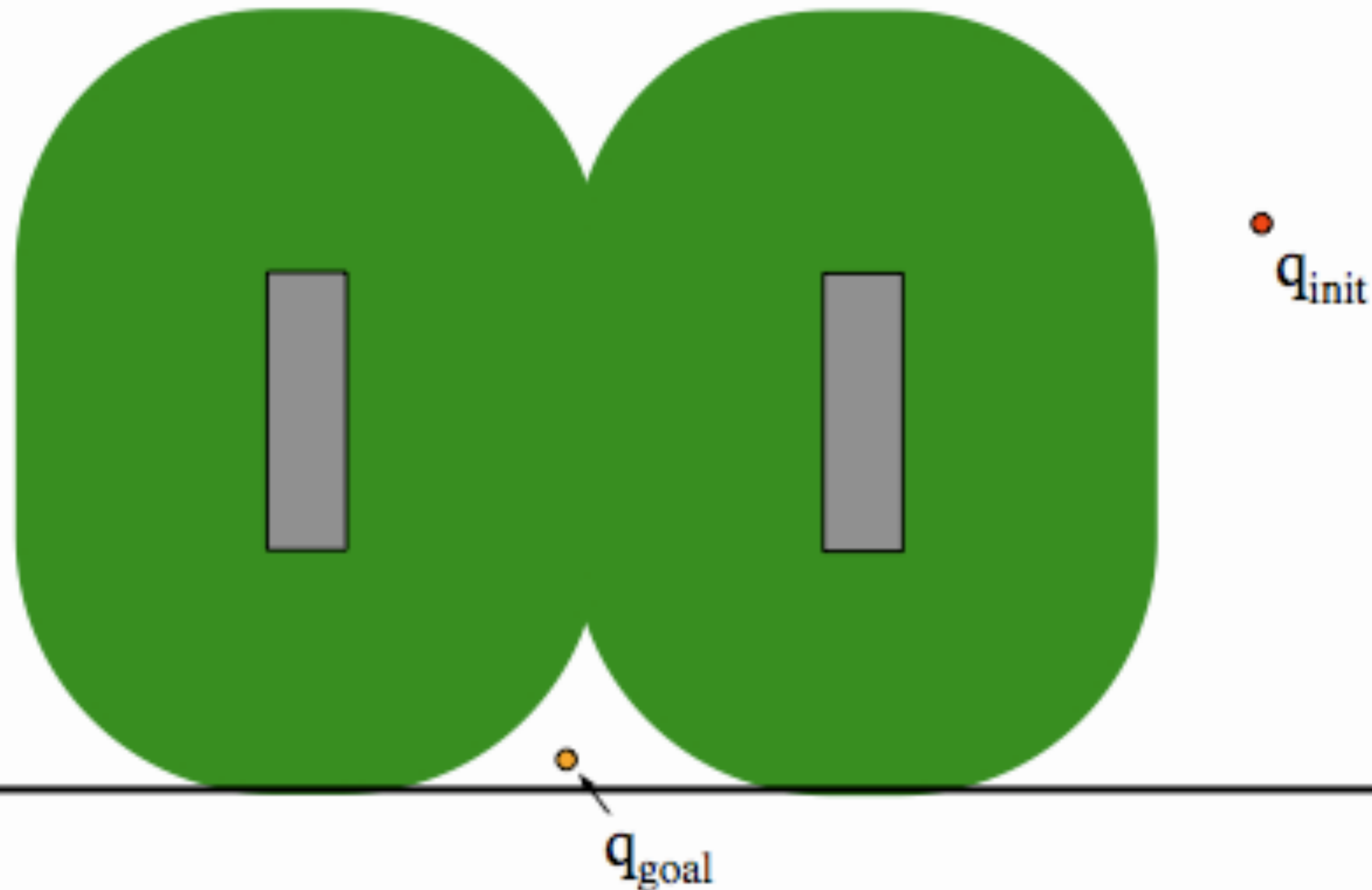
C-space depends on rotation



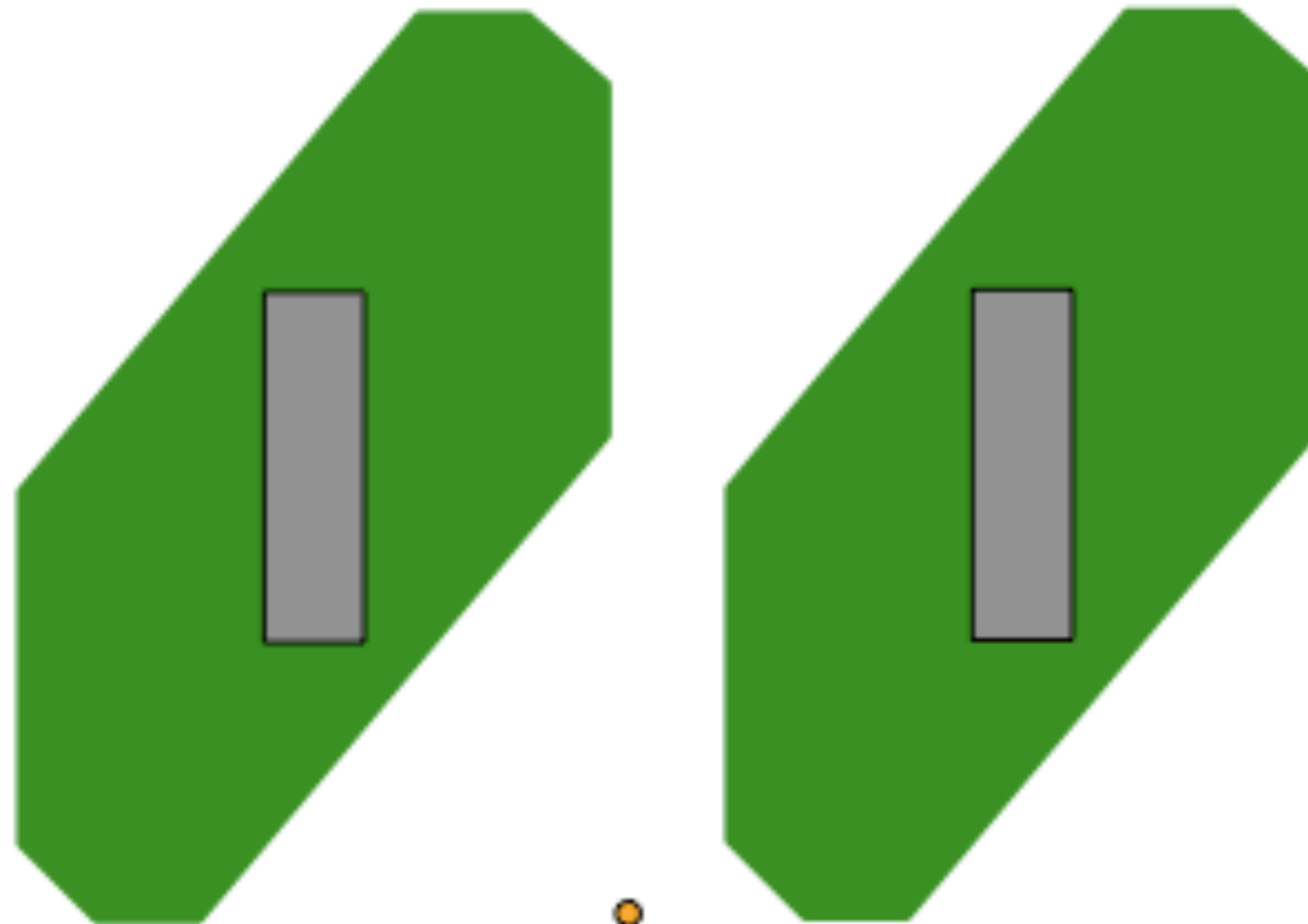
Consider this workspace...



C-space where obstacles are grown with all possible object positions and orientations



C-space where obstacles are grown with all possible object positions, orientation constrained to 45 degrees

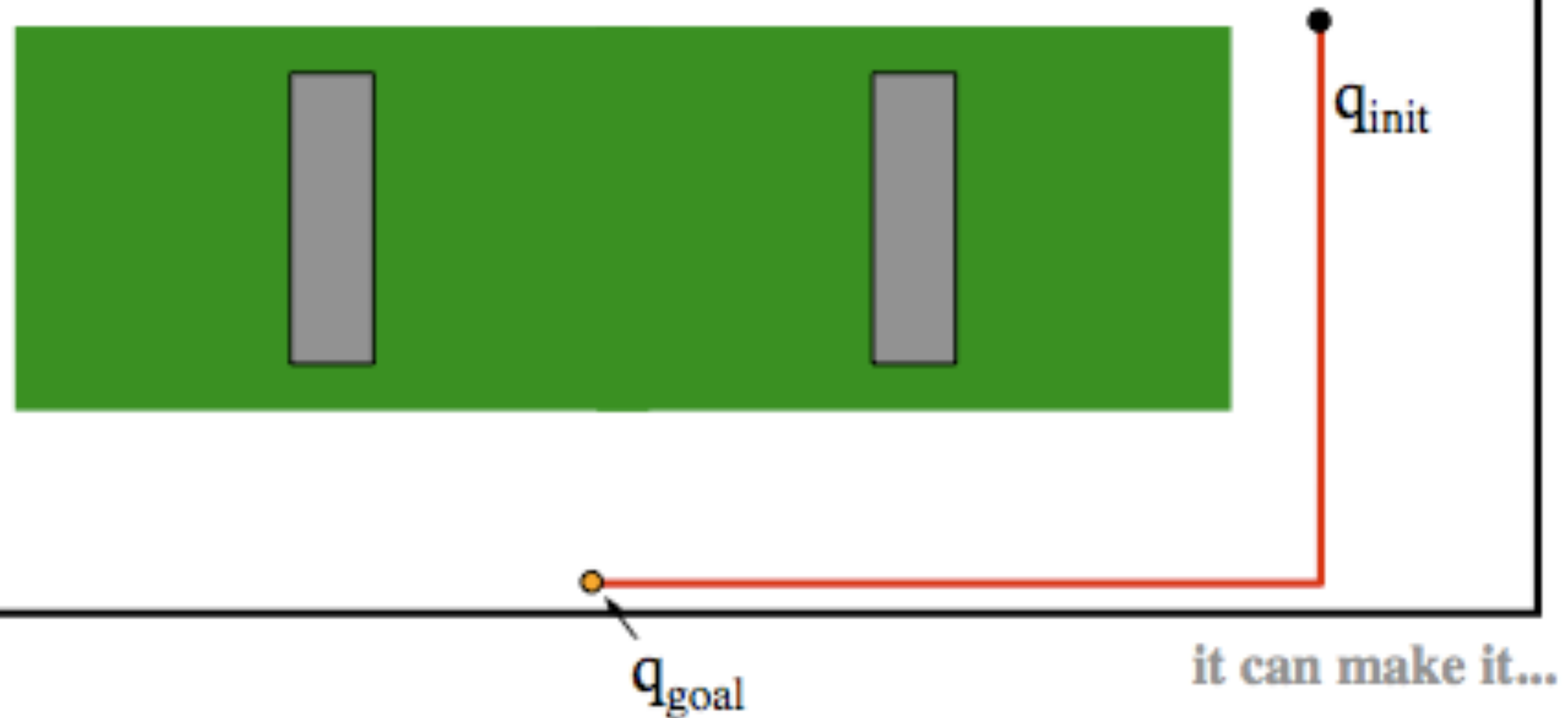


q_{init}

q_{goal}

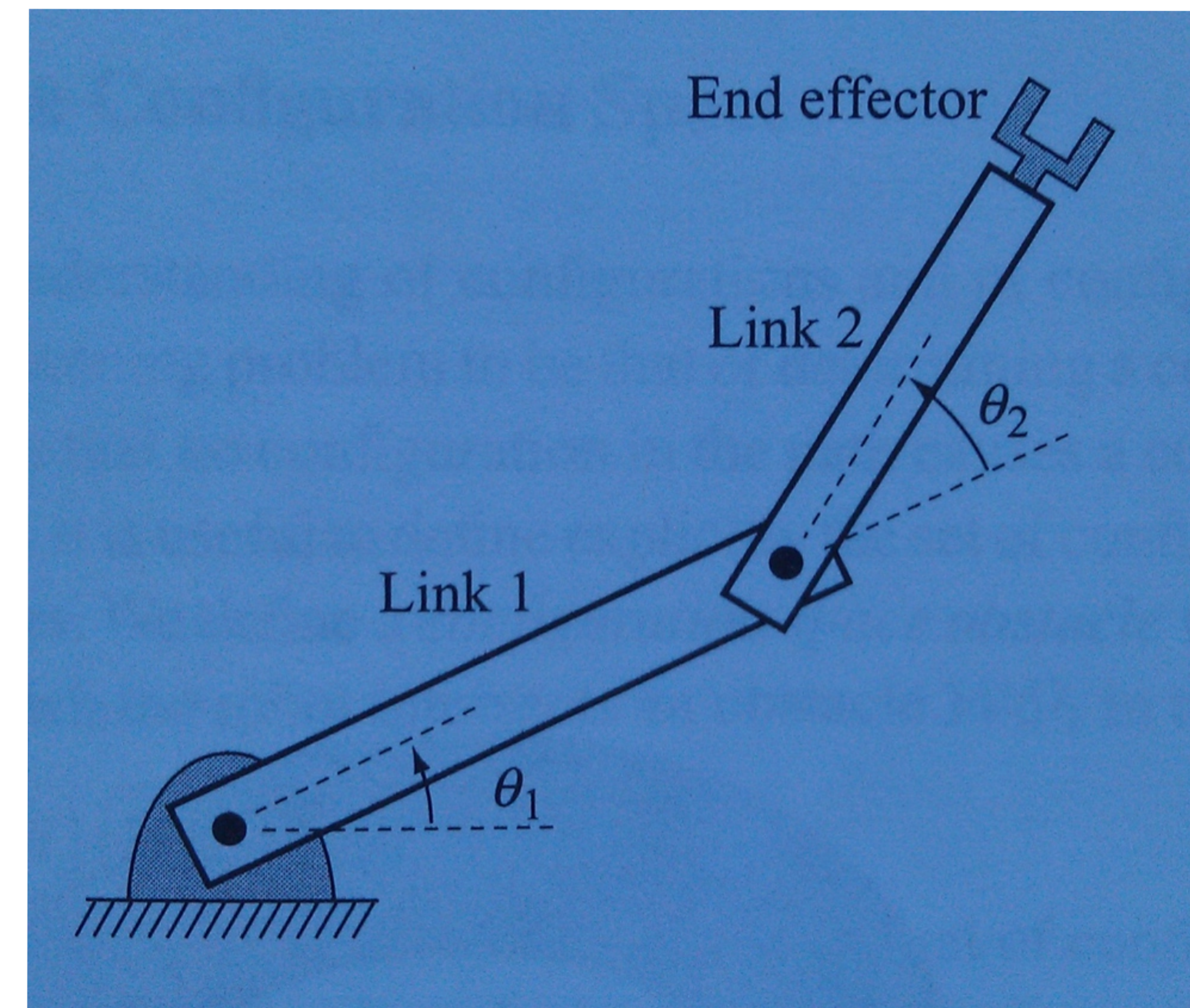
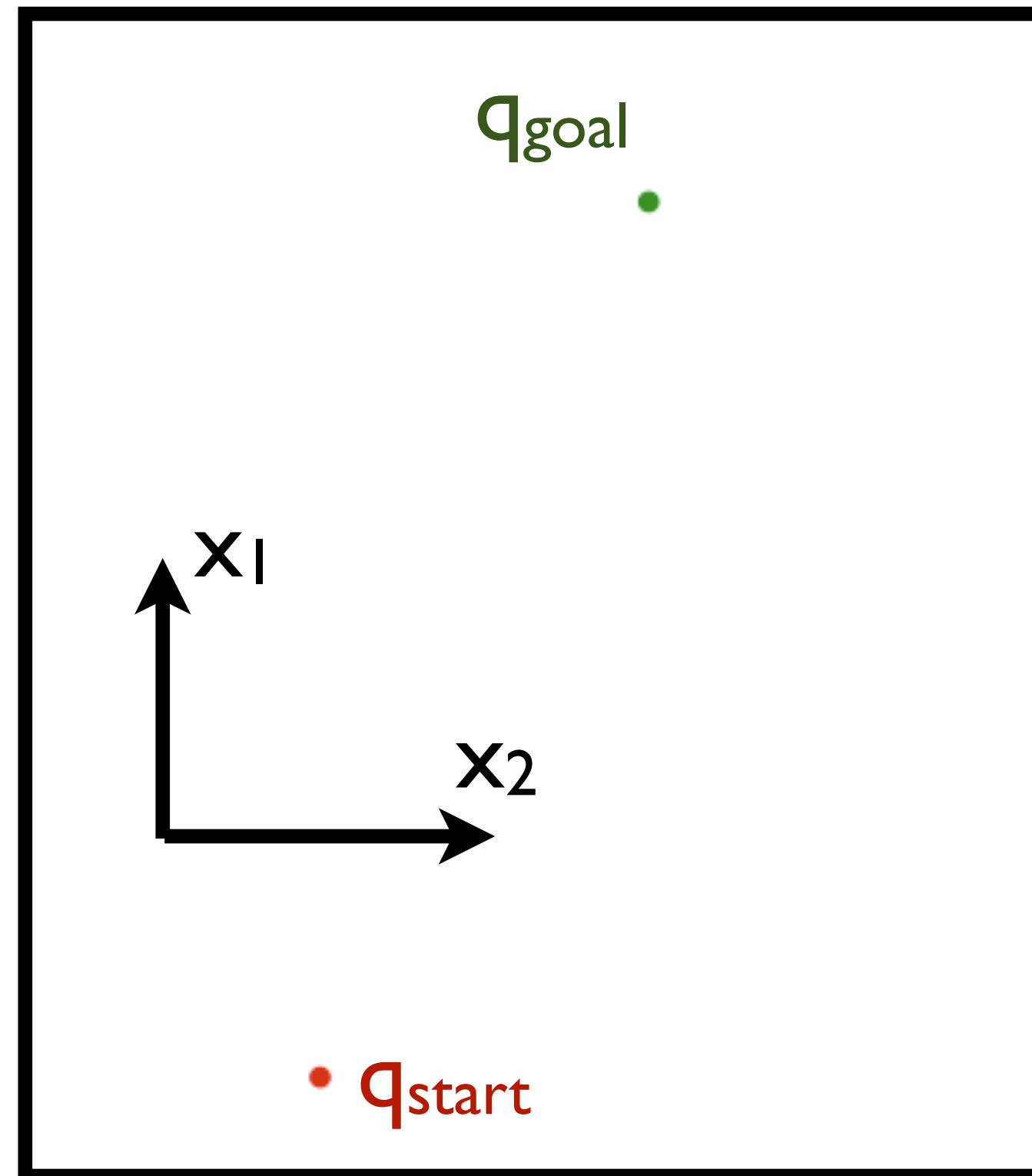
it depends...

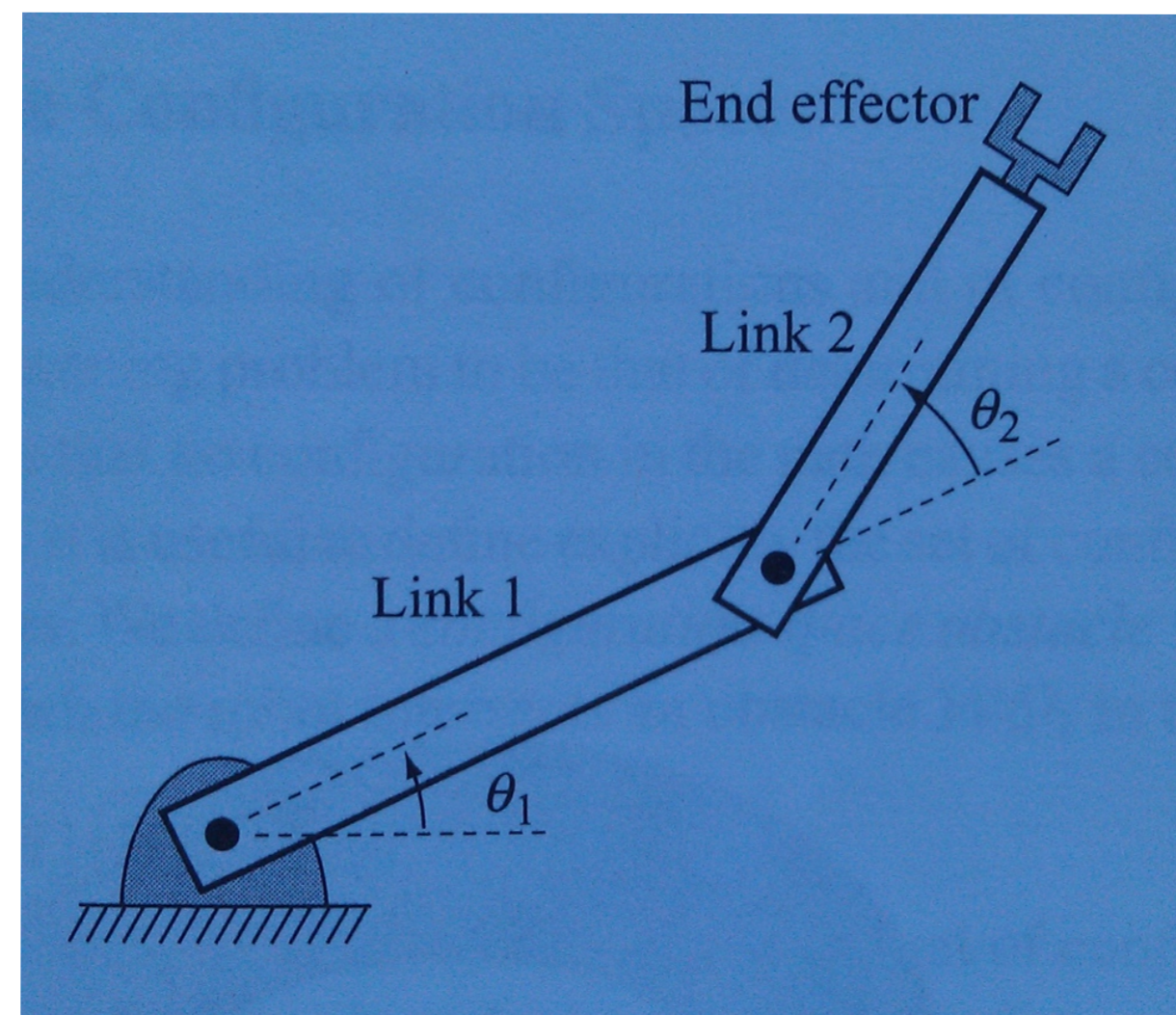
C-space where obstacles are grown with all possible object positions, orientation constrained to 0 degrees



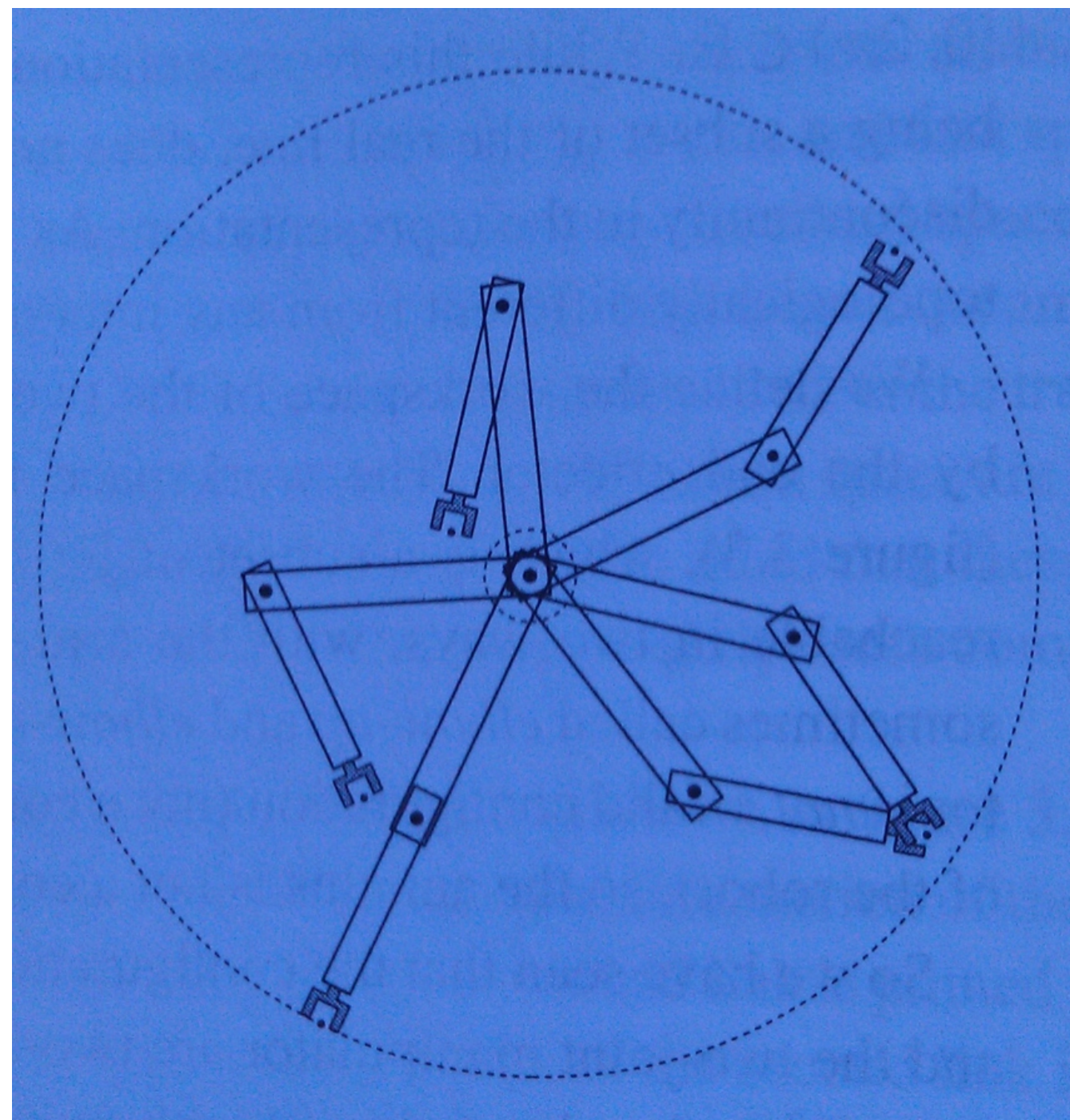
Configuration v. Workspaces

- Other than rotation and geometry, how is the 2-link arm different than the point robot?

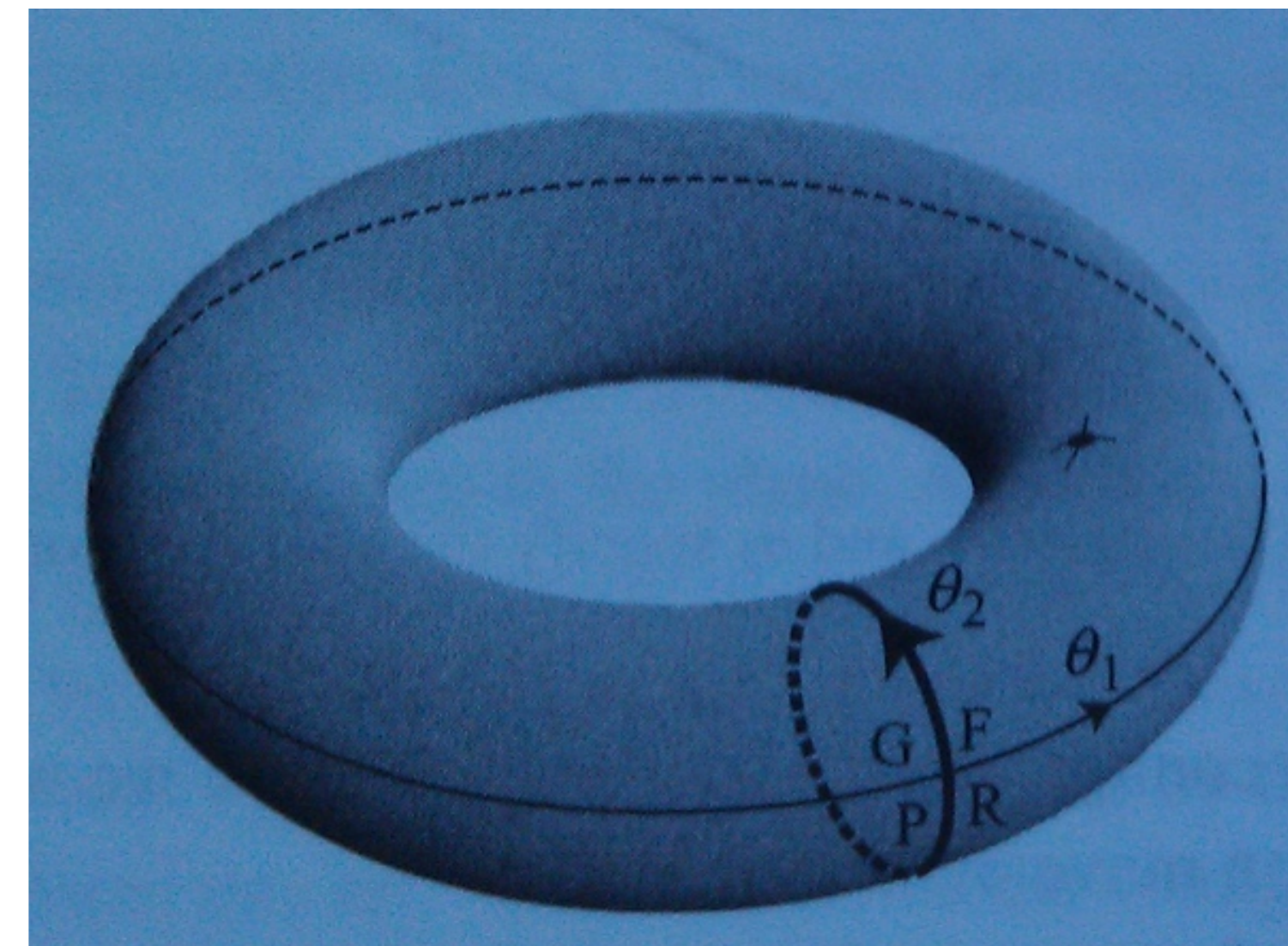




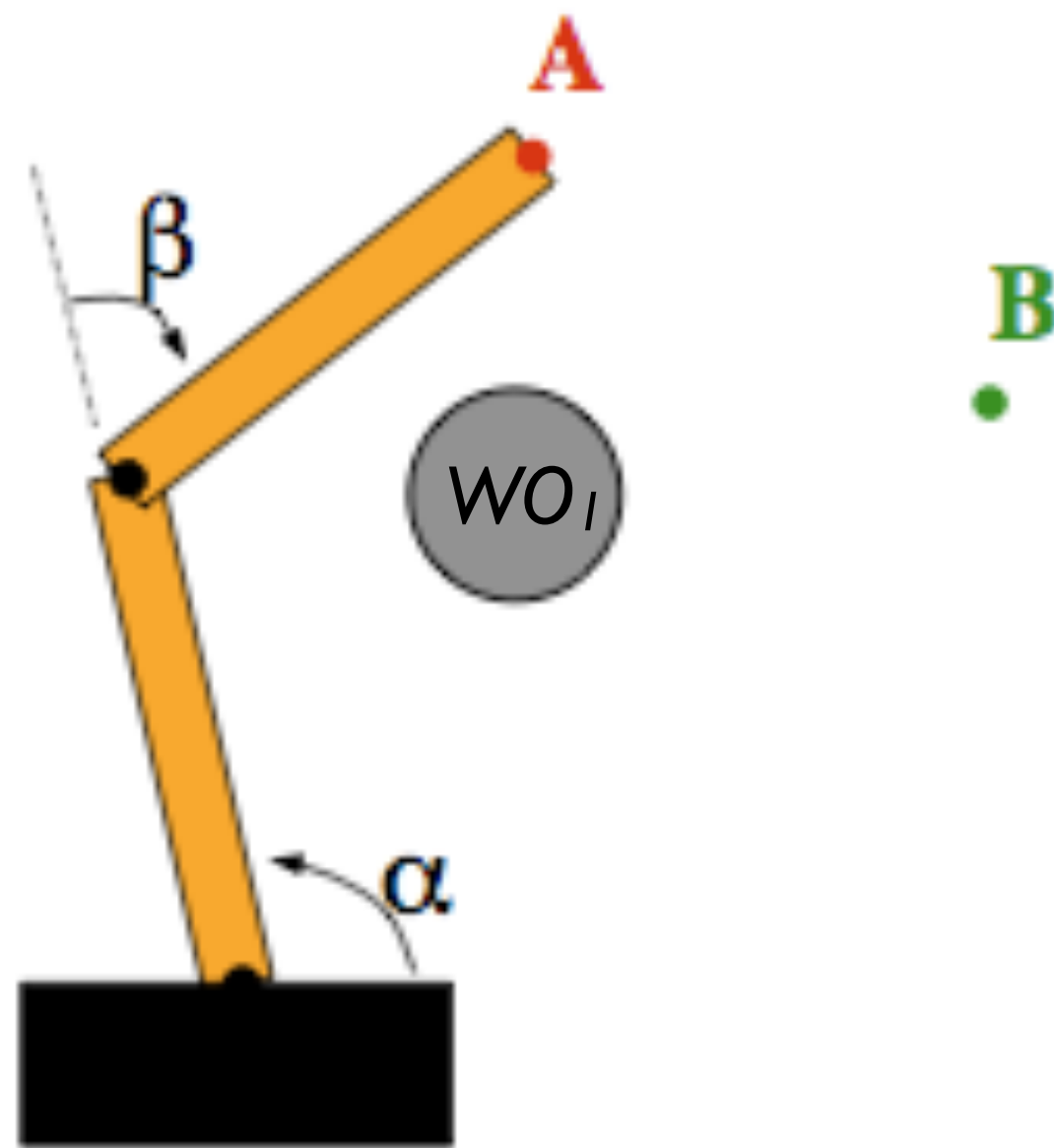
Workspace is w.r.t. end-effector position (x,y)



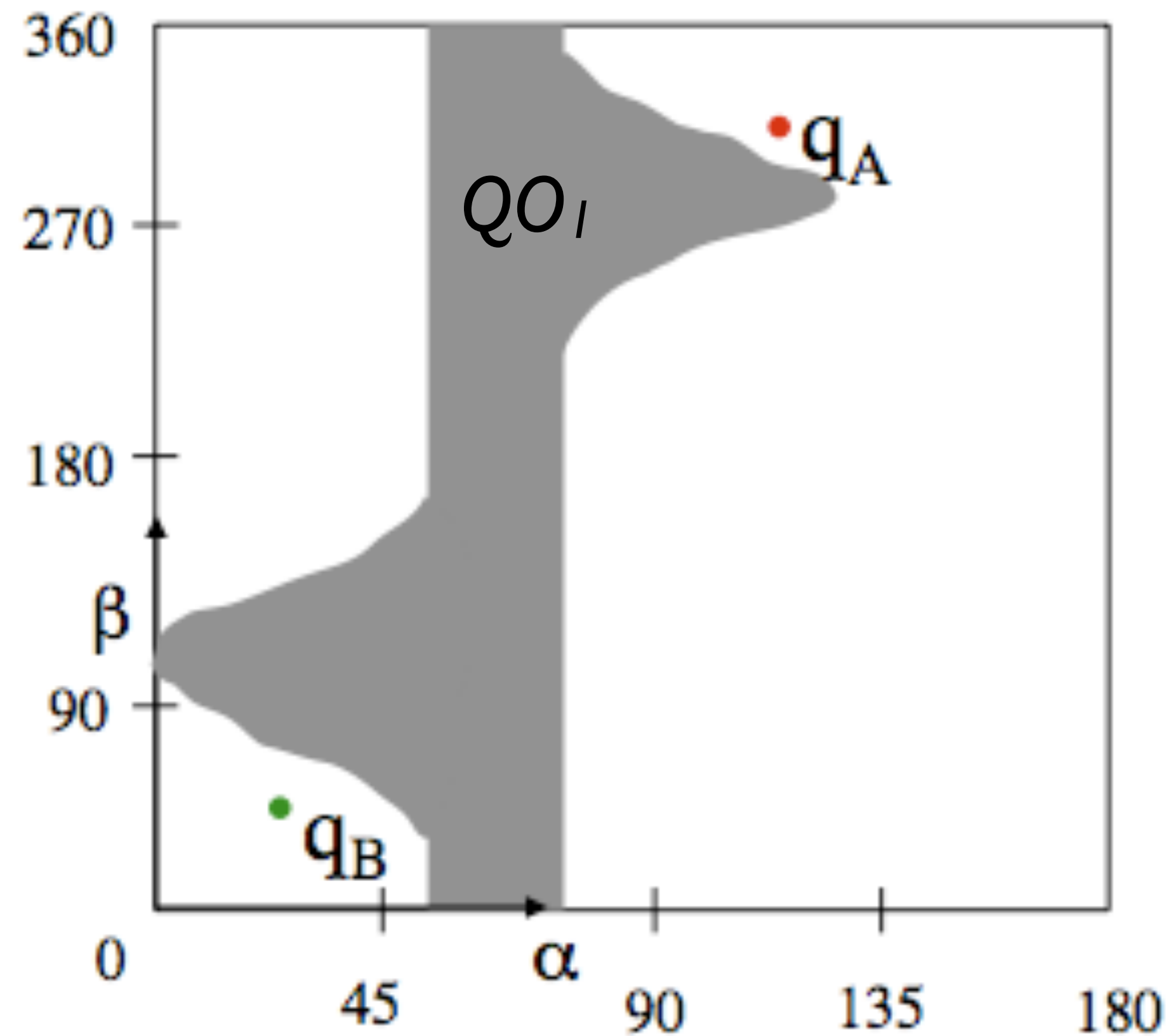
C-space is w.r.t. joint angles (θ_1, θ_2)



Obstacles in T²

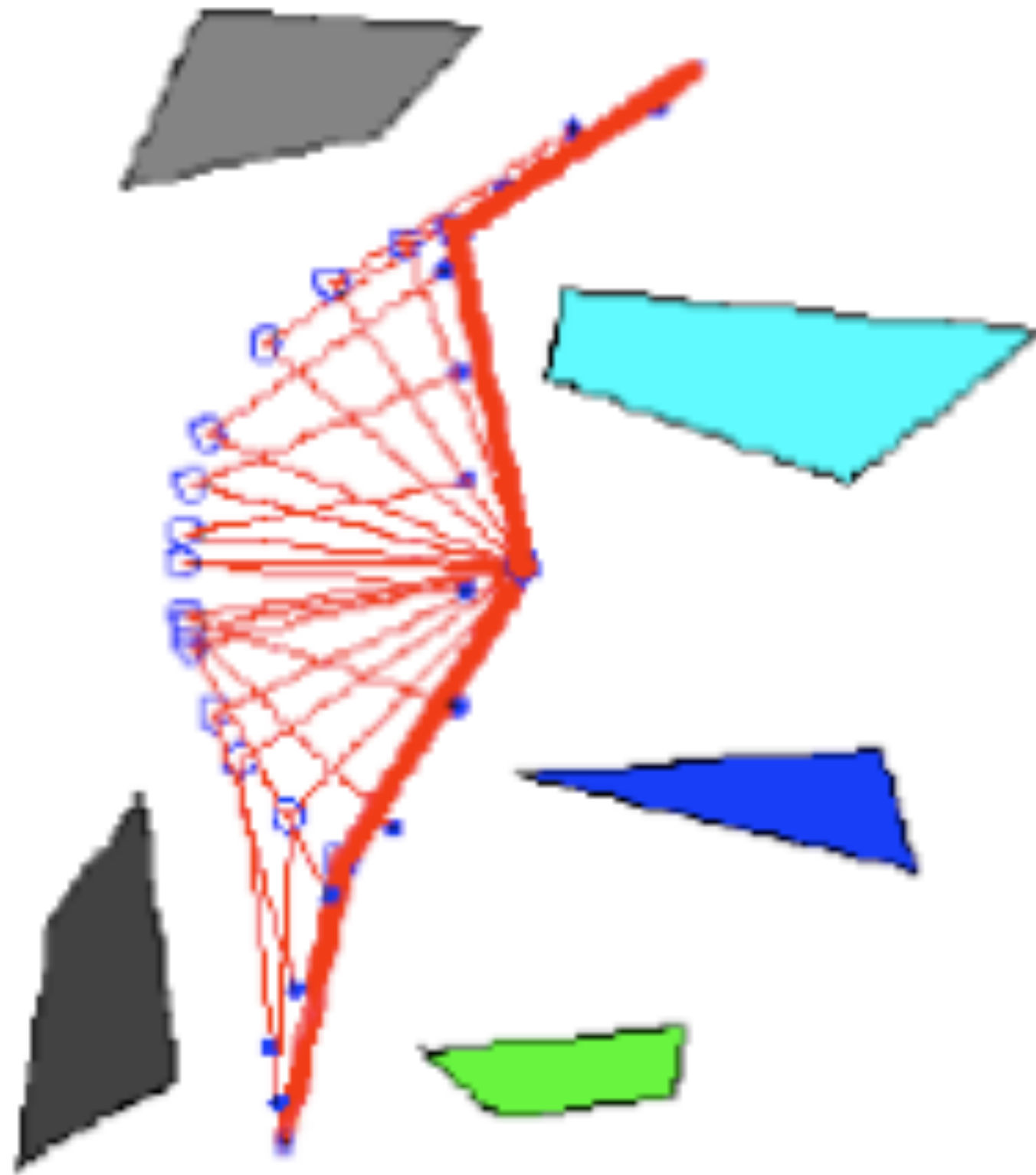


Circular obstacle in workspace

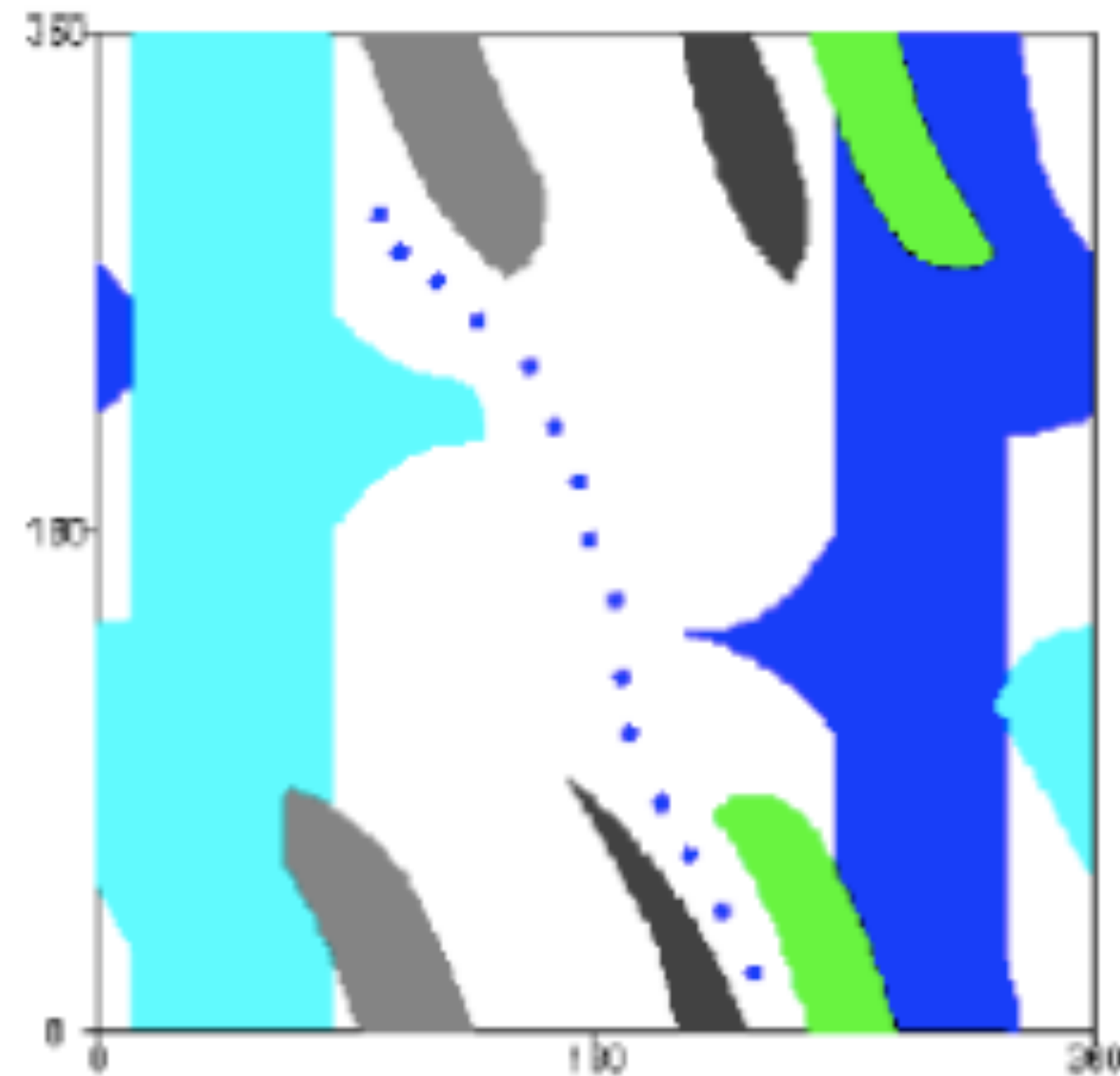


C-space representation

Path in T^2 with several obstacles



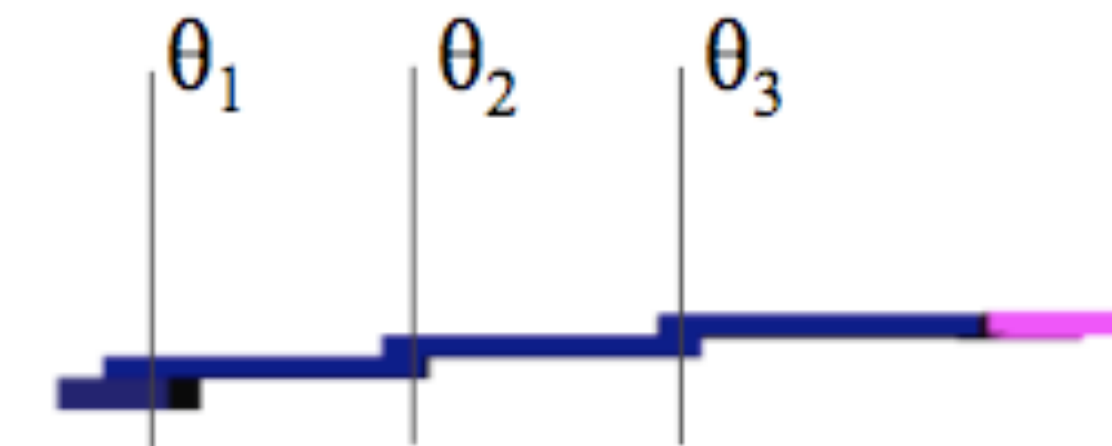
Arm navigation in workspace



C-space representation

C-space for 3-link arm

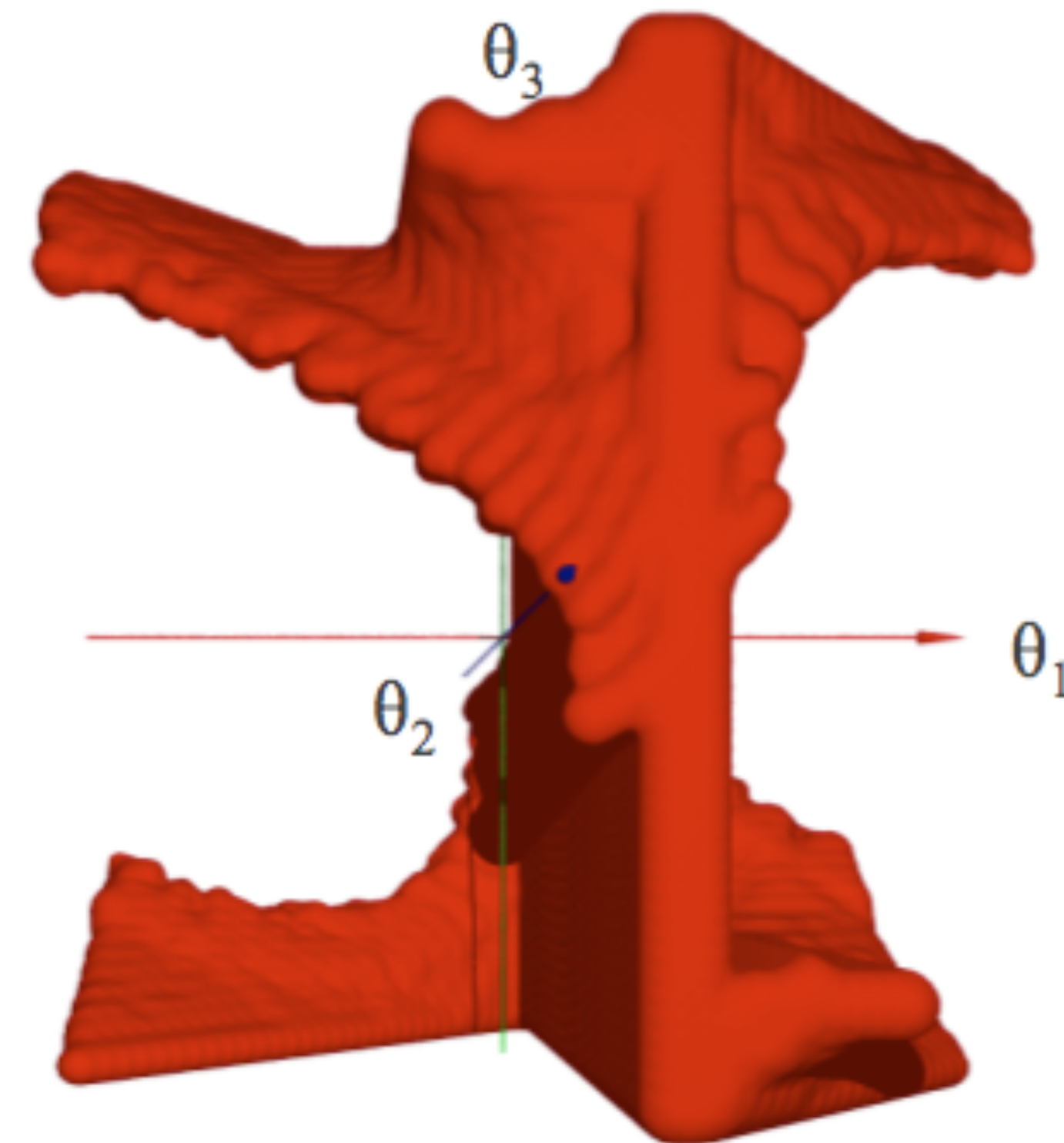
The Configuration Space (C-space)



TOP
VIEW



workspace



C-space

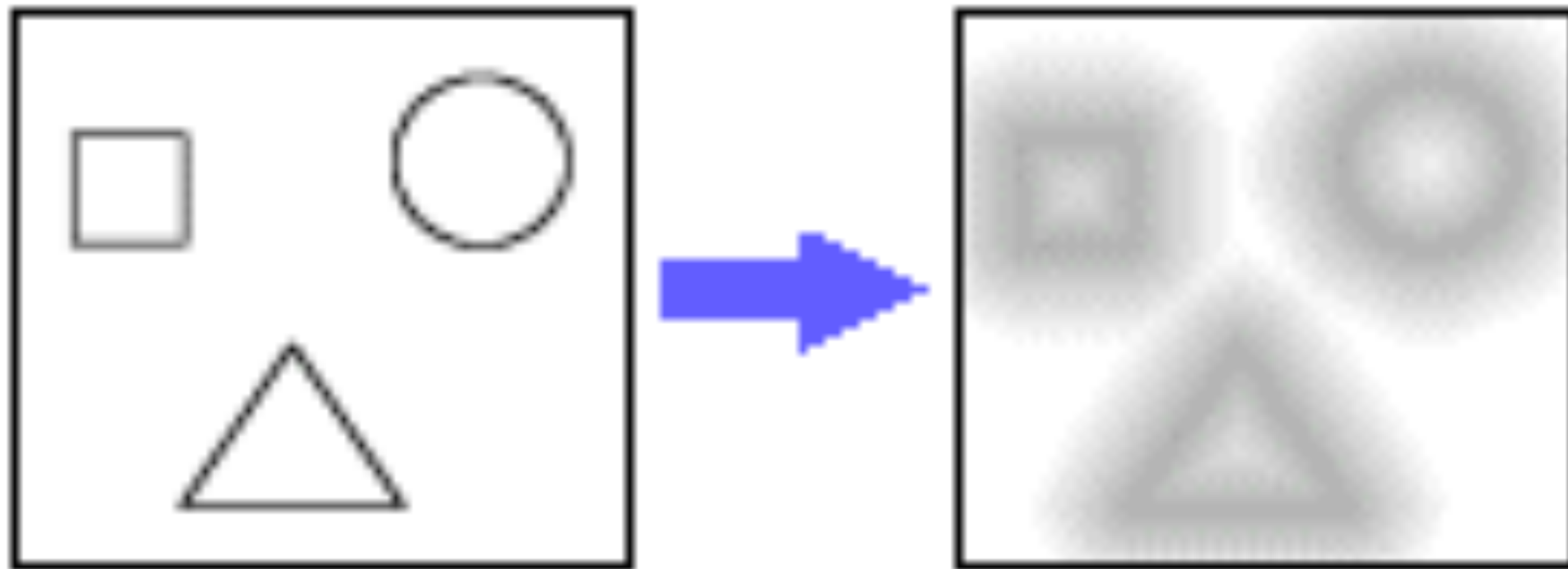
Generalizing graph search for robot configurations

Costmaps: Graph Search Revisited

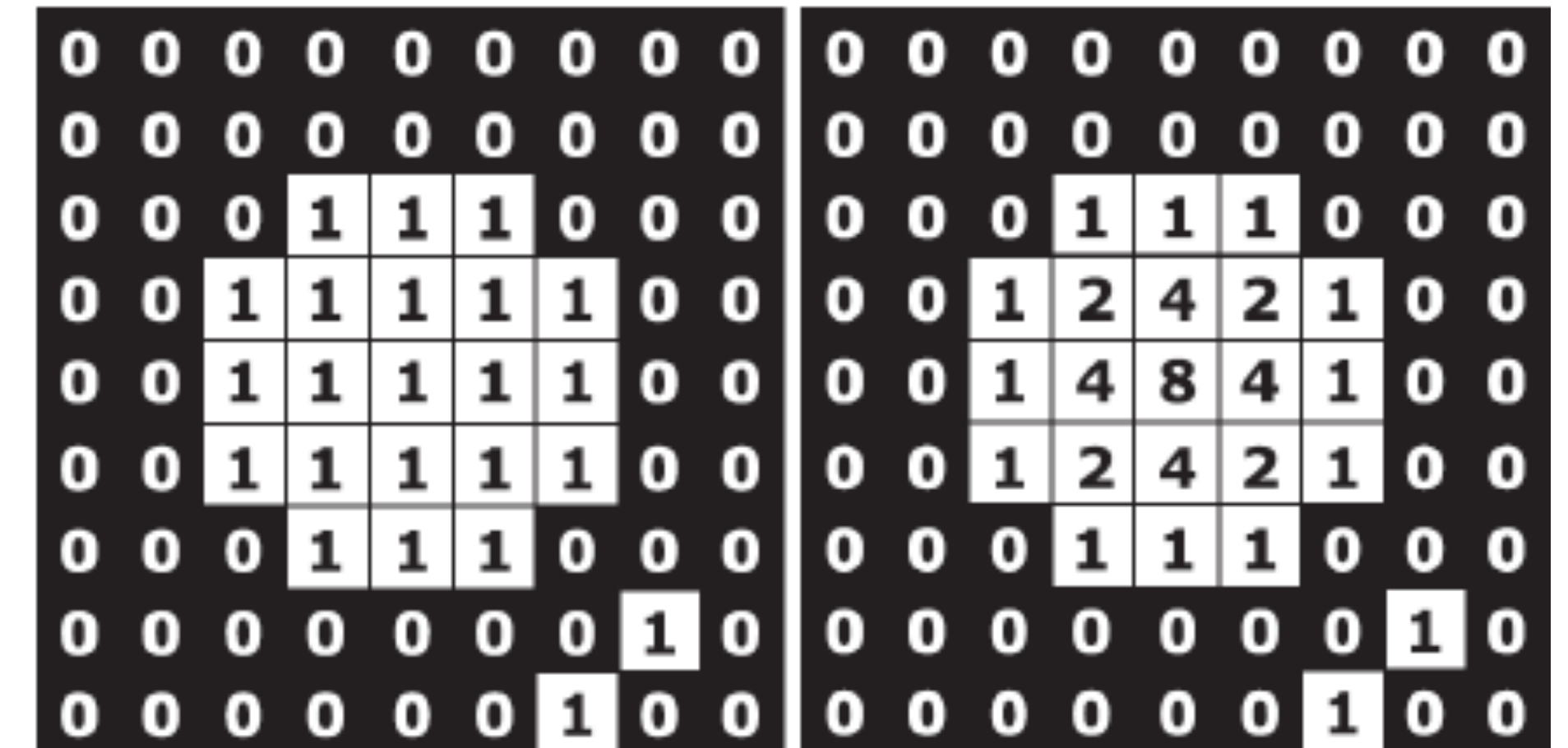
- Optimality: Path length vs. Path cost?
- **Costmap** provides weights on graph nodes based on cost factors:
 - Robot motion: joint limits, holonomicity, smoothness
 - Collisions and safety: distance from objects, trajectory predictions
 - Environmental conditions: traversability, slip

Distance Transform

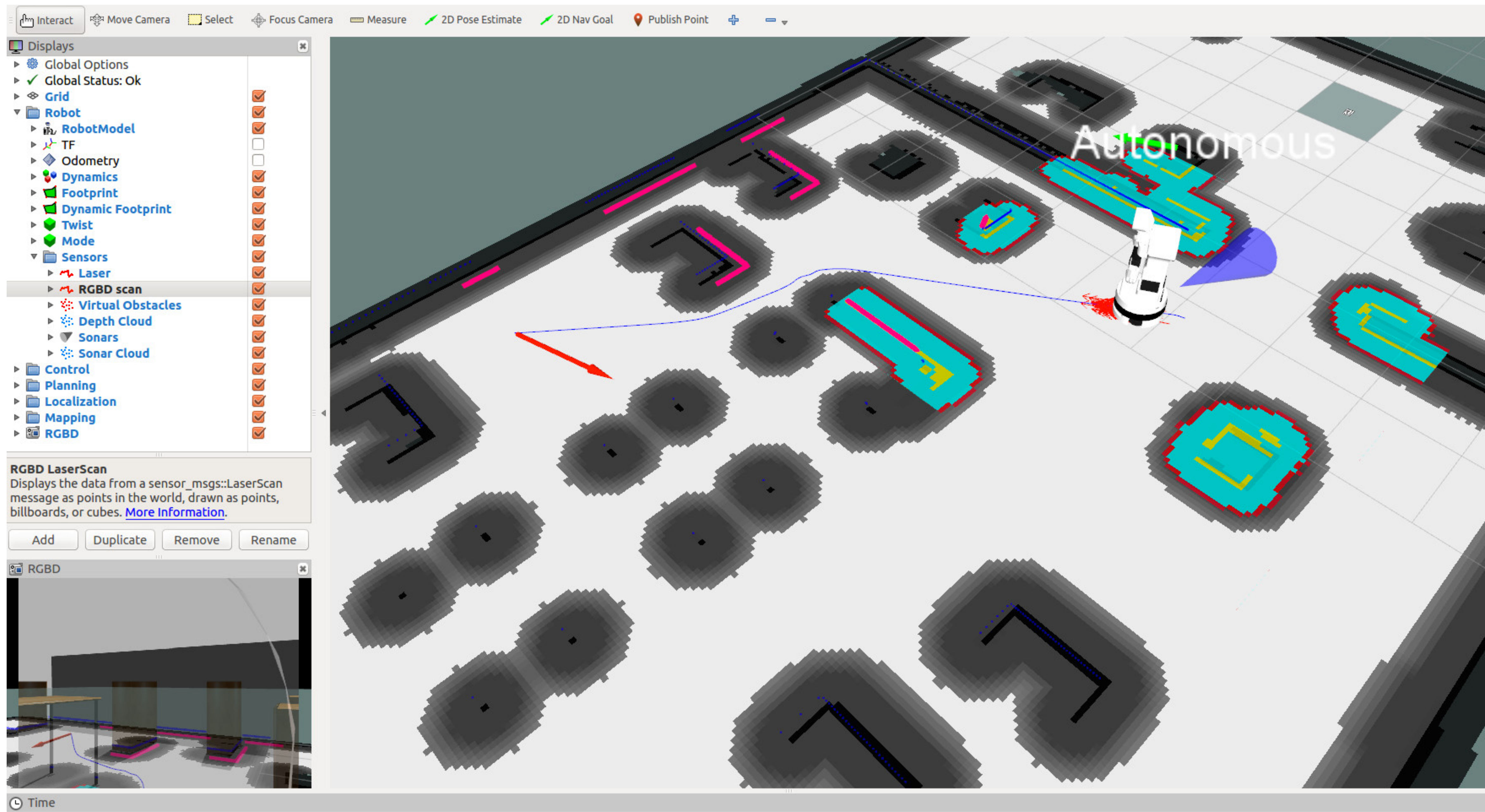
Compute distance of each grid cell to nearest obstacle boundary;
Weight grid cell cost higher if closer to a boundary



http://www.gavrila.net/Research/Chamfer_System/chamfer_basics2.gif



Nasonov and Krylov 2010
(zero indicates obstacle)



<https://blog.pal-robotics.com/blog/tiago-ros-simulation-tutorial-2-autonomous-robot-navigation/>

Search algorithm template

all nodes $\leftarrow \{\text{dist}_{\text{start}} \leftarrow \text{infinity}, \text{parent}_{\text{start}} \leftarrow \text{none}, \text{visited}_{\text{start}} \leftarrow \text{false}\}$

start_node $\leftarrow \{\text{dist}_{\text{start}} \leftarrow 0, \text{parent}_{\text{start}} \leftarrow \text{none}, \text{visited}_{\text{start}} \leftarrow \text{true}\}$

visit_list $\leftarrow \text{start_node}$

while visit_list $\neq$ empty && current_node $\neq$ goal

 cur_node $\leftarrow \text{highestPriority}(\text{visit_list})$

 visited_{cur_node} $\leftarrow$ true

for each nbr in not_visited(adjacent(cur_node))

if $\text{dist}_{\text{nbr}} > \text{dist}_{\text{cur_node}} + \text{distance}(\text{nbr}, \text{cur_node})$

 parent_{nbr} $\leftarrow$ current_node

$\text{dist}_{\text{nbr}} \leftarrow \text{dist}_{\text{cur_node}} + \text{distance}(\text{nbr}, \text{cur_node})$

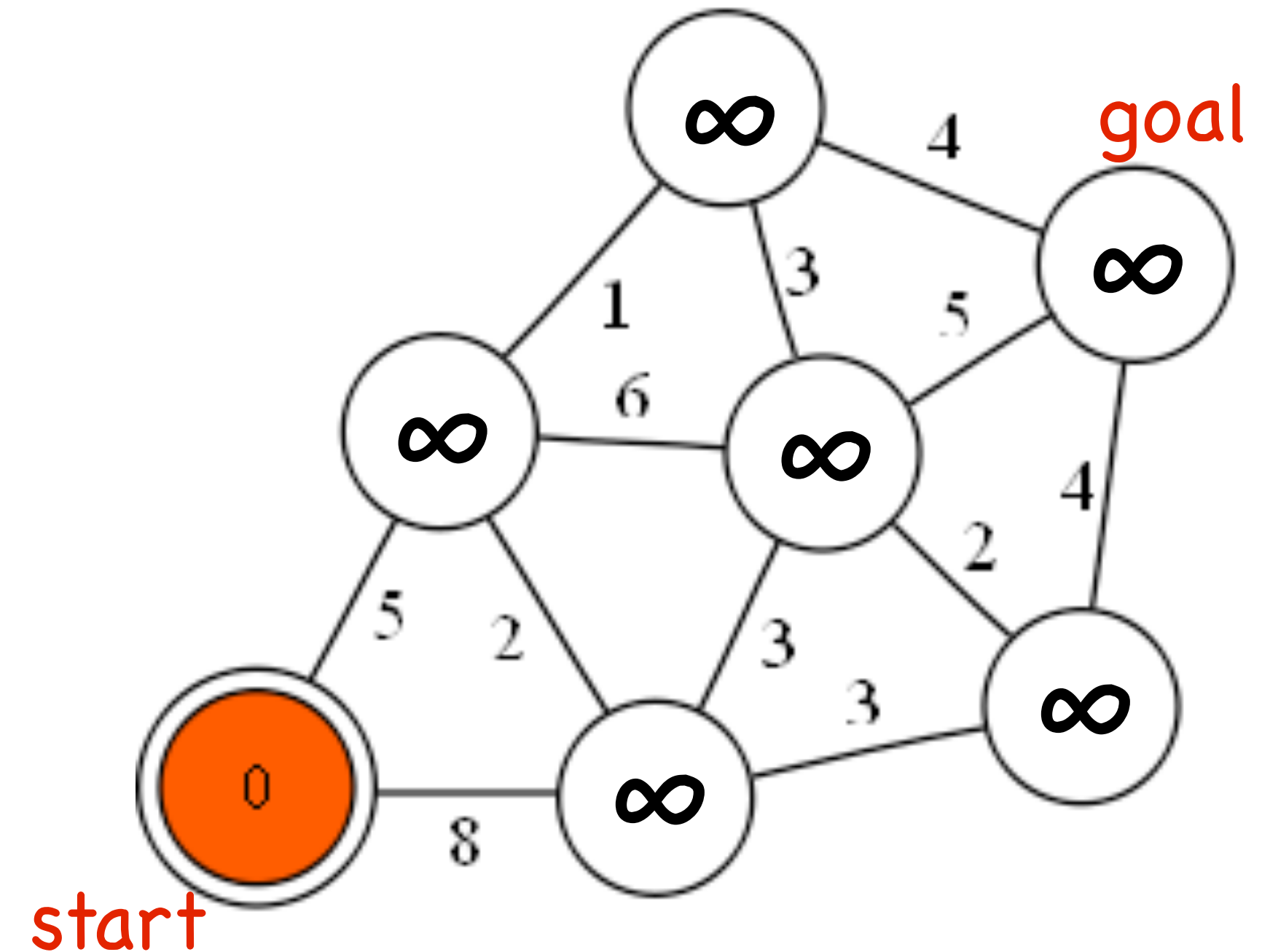
 add(nbr to visit_list)

end if

end for loop

end while loop

output $\leftarrow$ parent, distance



Search algorithm template

all nodes $\leftarrow \{\text{cost}_{\text{start}} \leftarrow \text{infinity}, \text{parent}_{\text{start}} \leftarrow \text{none}, \text{visited}_{\text{start}} \leftarrow \text{false}\}$

start_node $\leftarrow \{\text{cost}_{\text{start}} \leftarrow 0, \text{parent}_{\text{start}} \leftarrow \text{none}, \text{visited}_{\text{start}} \leftarrow \text{true}\}$

visit_list $\leftarrow \text{start_node}$

while visit_list $\neq$ empty && current_node $\neq$ goal

cur_node $\leftarrow \text{highestPriority}(\text{visit_list})$

visited_{cur_node} $\leftarrow$ true

for each nbr in not_visited(adjacent(cur_node))

if $\text{cost}_{\text{nbr}} > \text{cost}_{\text{cur_node}} + \text{cost}(\text{nbr})$

parent_{nbr} $\leftarrow$ current_node

$\text{cost}_{\text{nbr}} \leftarrow \text{cost}_{\text{cur_node}} + \text{cost}(\text{nbr})$

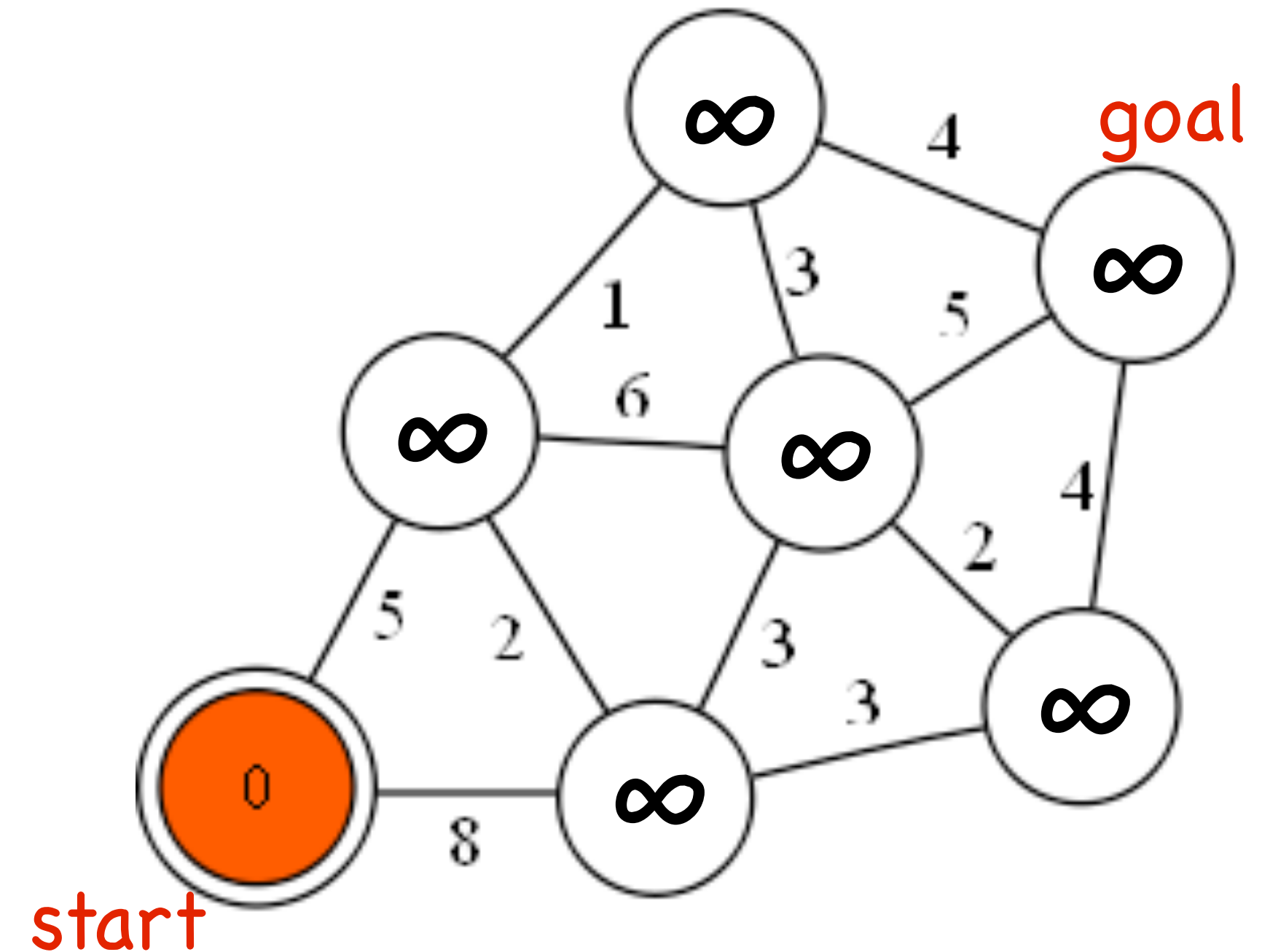
add(nbr to visit_list)

end if

end for loop

end while loop

output $\leftarrow$ parent, distance



A-star shortest path algorithm

all nodes $\leftarrow \{\text{cost}_{\text{start}} \leftarrow \text{infinity}, \text{parent}_{\text{start}} \leftarrow \text{none}, \text{visited}_{\text{start}} \leftarrow \text{false}\}$

start_node $\leftarrow \{\text{cost}_{\text{start}} \leftarrow 0, \text{parent}_{\text{start}} \leftarrow \text{none}, \text{visited}_{\text{start}} \leftarrow \text{true}\}$

visit_queue $\leftarrow \text{start_node}$

while (visit_queue $\neq$ empty) $\&\&$ current_node $\neq$ goal

 dequeue: cur_node $\leftarrow \text{f_score}(\text{visit_queue})$

 visited_{cur_node} $\leftarrow$ true

for each nbr in not_visited(adjacent(cur_node))

if $\text{cost}_{\text{nbr}} > \text{cost}_{\text{cur_node}} + \text{cost}(\text{nbr})$

 parent_{nbr} $\leftarrow$ current_node

$\text{cost}_{\text{nbr}} \leftarrow \text{cost}_{\text{cur_node}} + \text{cost}(\text{nbr})$

 f_score $\leftarrow \text{cost}_{\text{nbr}} + \text{line_distance}_{\text{nbr}, \text{goal}}$

 enqueue: nbr to visit_queue

end if

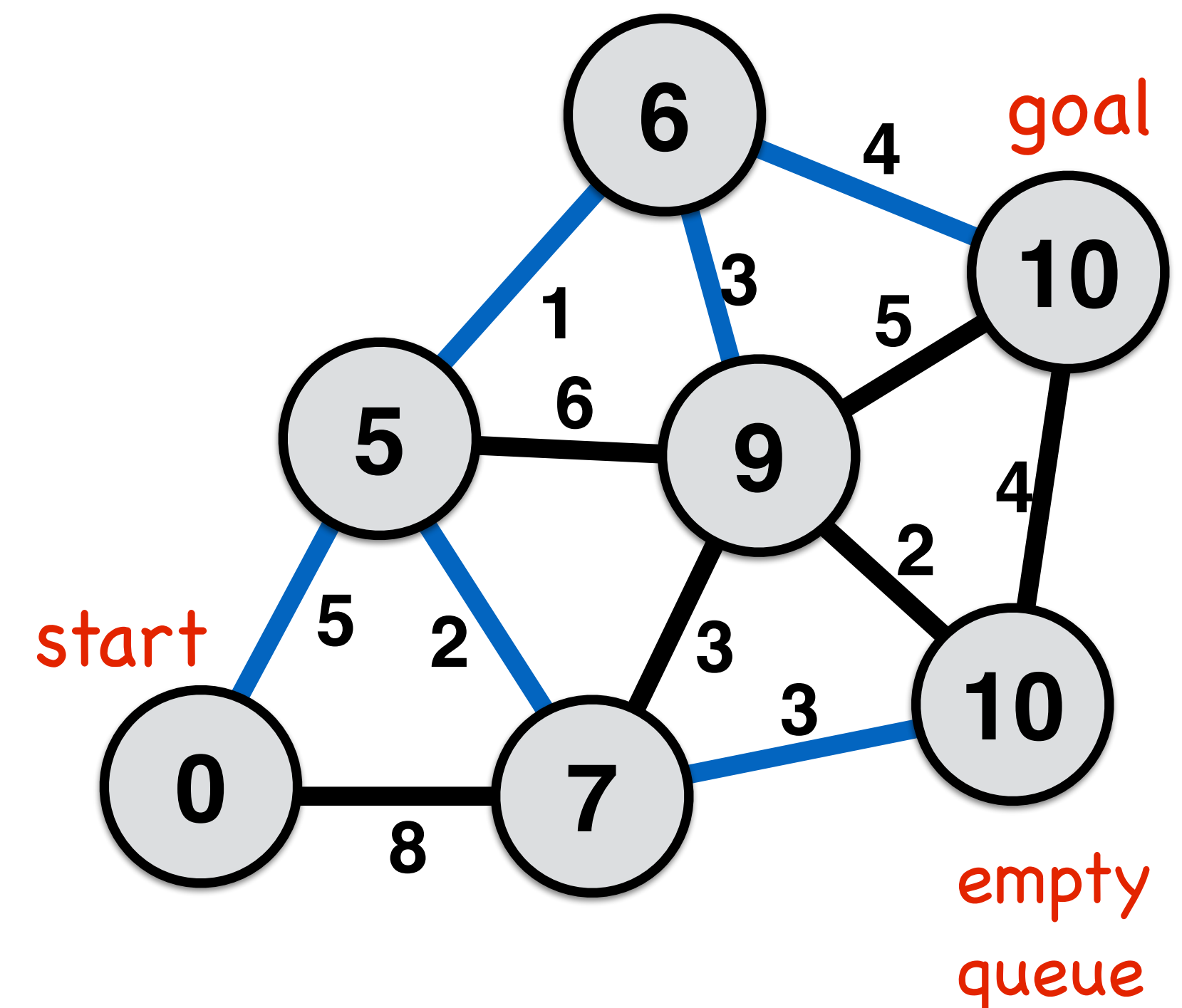
end for loop

end while loop

output $\leftarrow$ parent, distance

g_score:
cost from start

h_score:
optimistic cost to goal

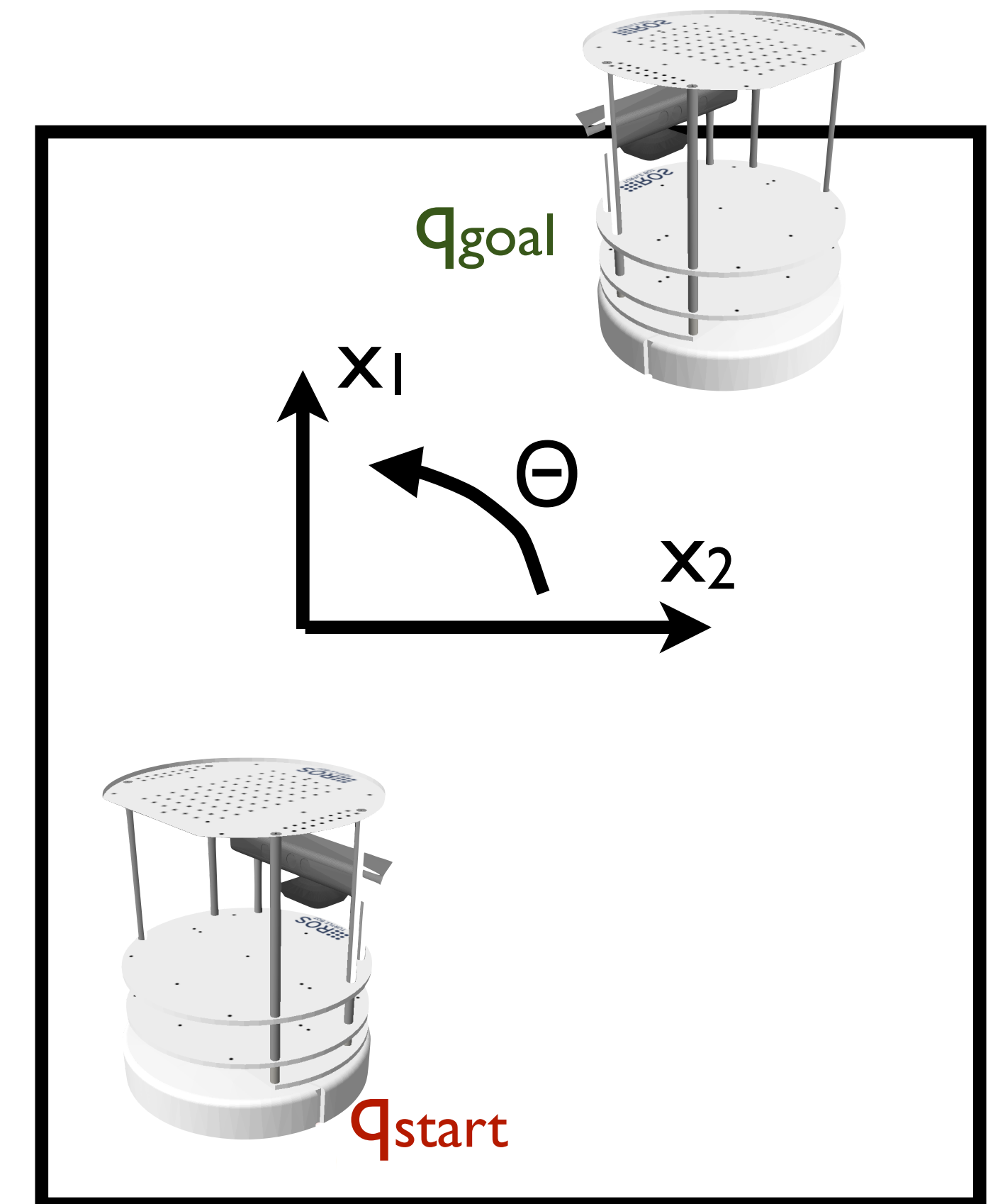


Can a robot move in any
direction instantaneously?

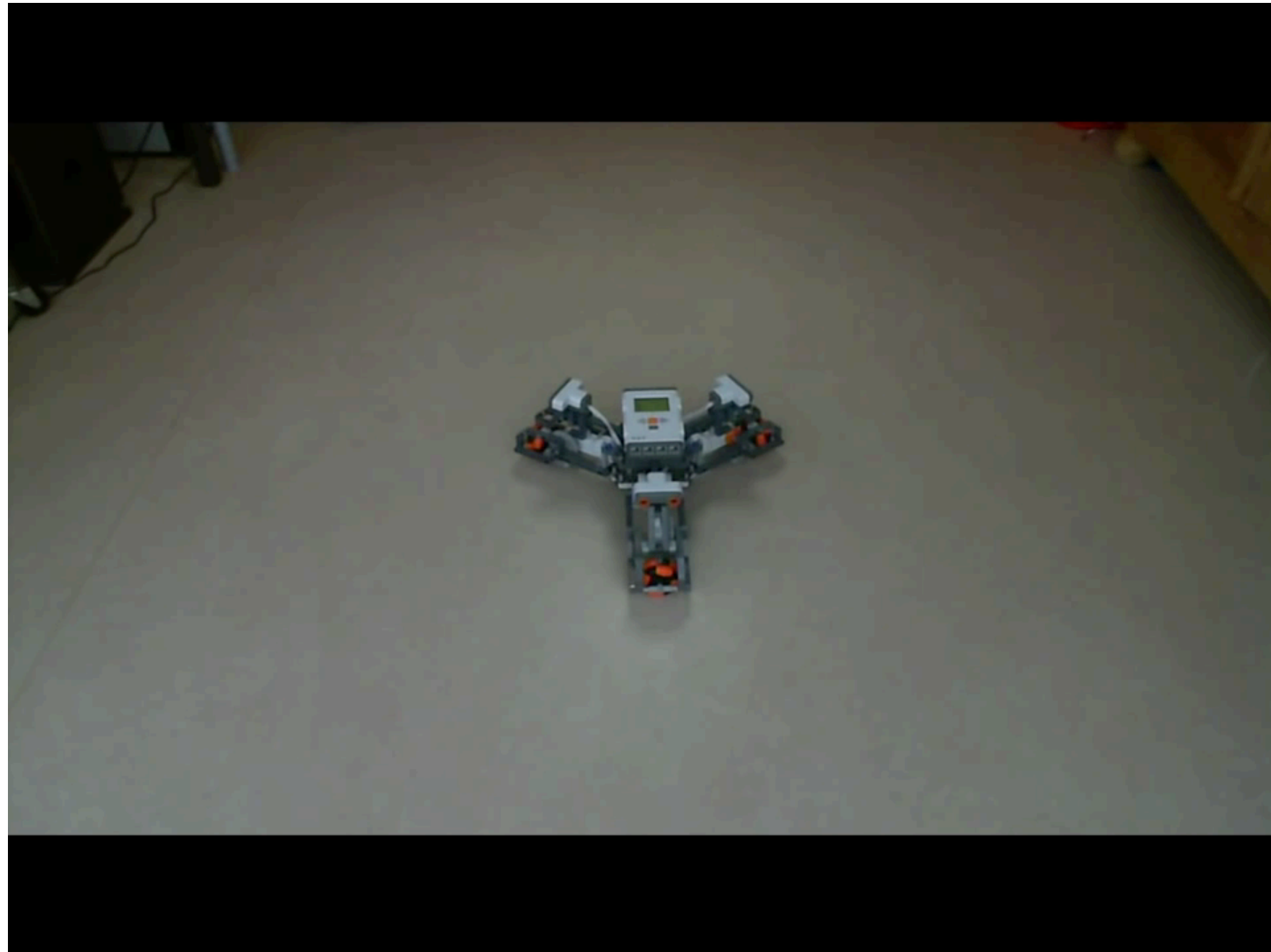
Holonomicity

- Does the Turtlebot have 2 DOFs, instead of 3?
- The Turtlebot can only move along 2 axes
 - linear: forward/backward
 - angular: turning

Turtlebot is nonholonomic



Holonomicity



<https://www.youtube.com/watch?v=c-IEjVsoiGo>

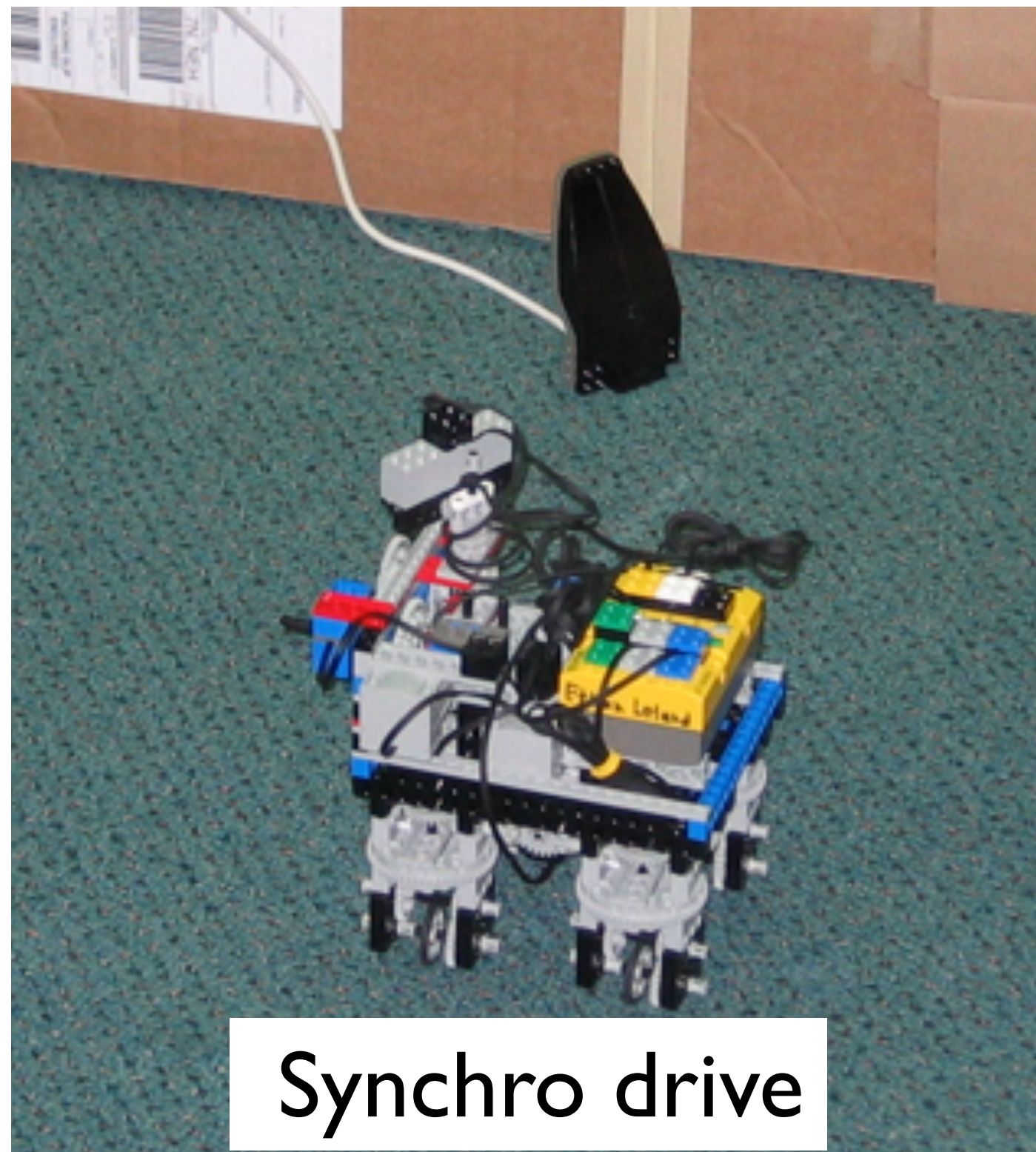


<https://www.youtube.com/watch?v=1ak17mdRg5I&t=75s>

- A robot is holonomic if it can change its pose instantaneously to move in all directions
- Otherwise, the robot is nonholonomic

Holonomic mobile robot systems

Omni-wheel drive



Synchro drive

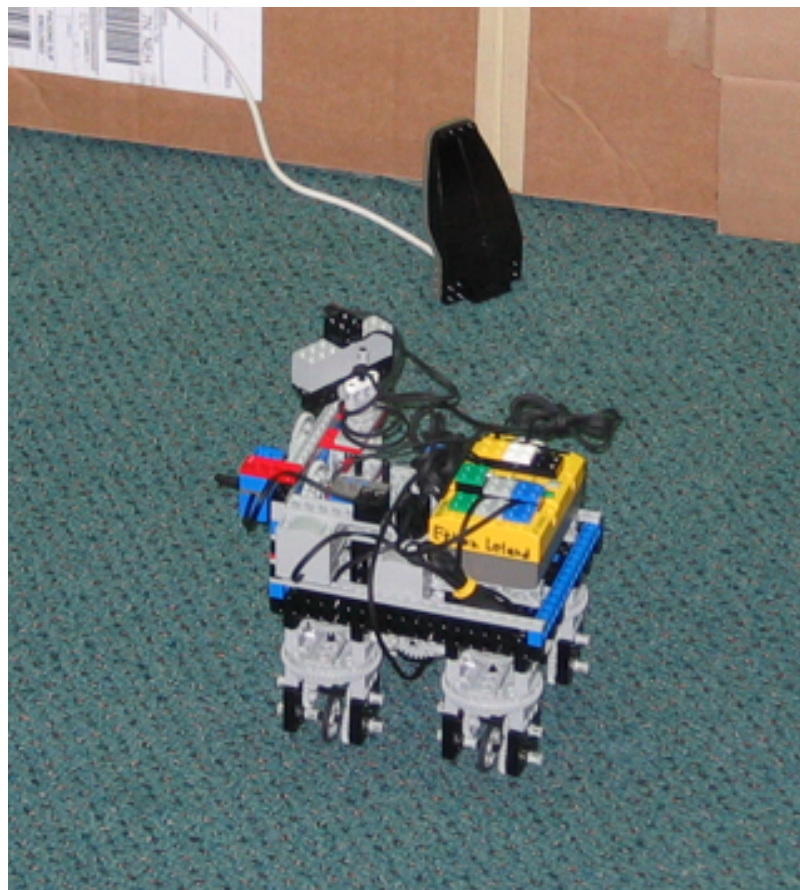


Mecanum wheels



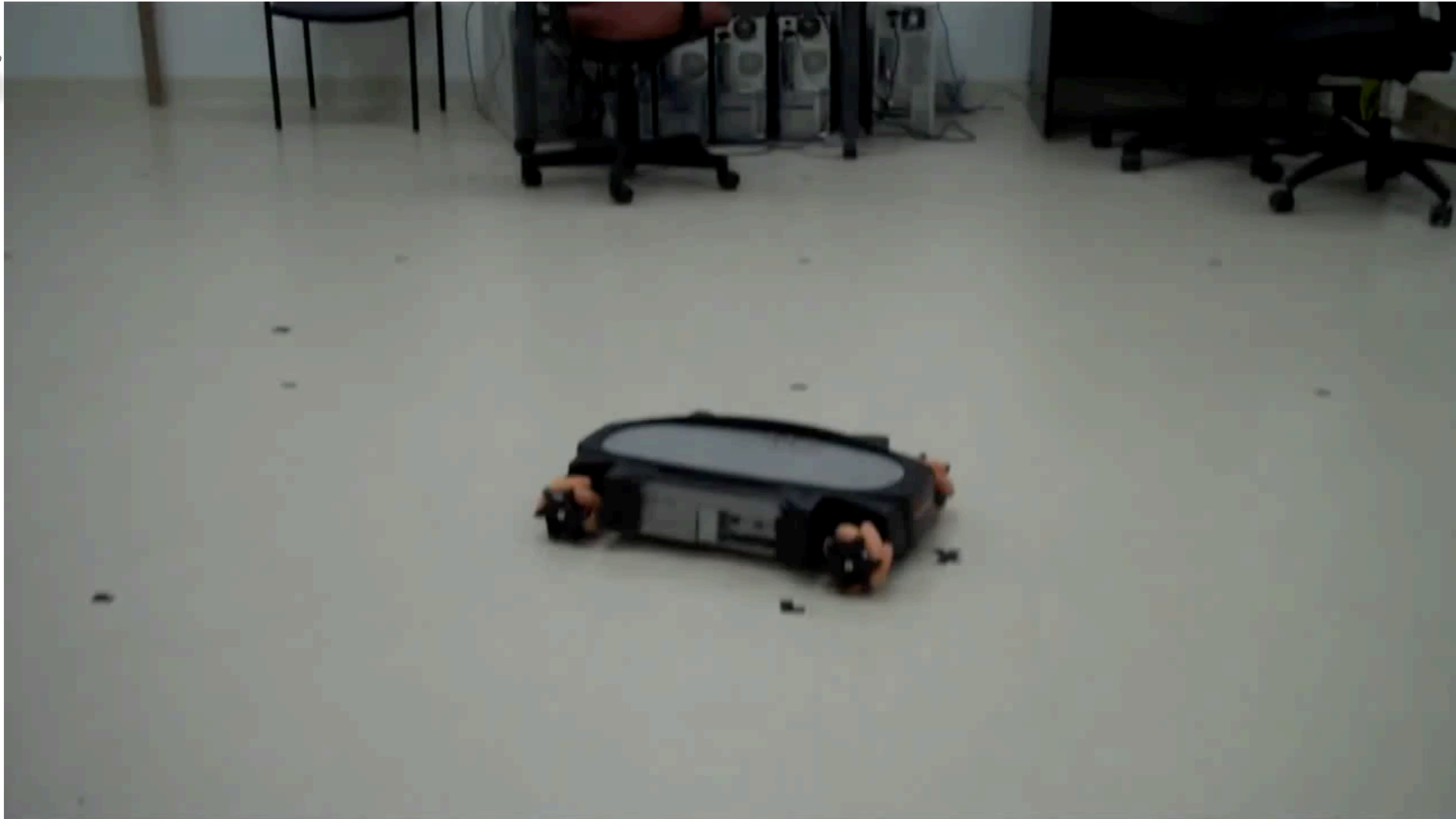
Killough platform

Synchro Drive



markclego, <https://www.youtube.com/watch?v=THdu6QD8Roc>

KUKA YouBot with Mecanum wheels

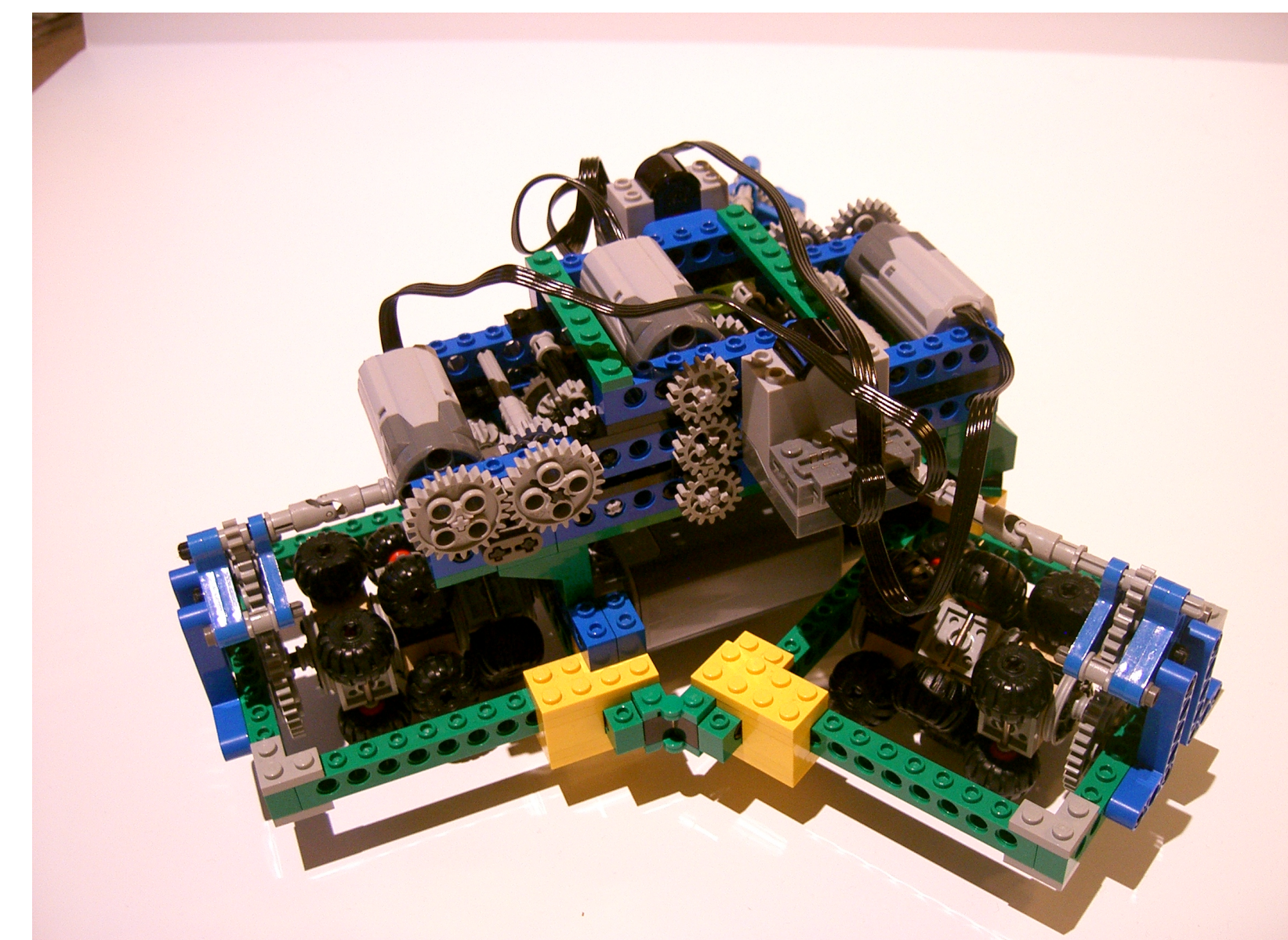


Me teleoperating KUKA YouBot from my house via a web browser, http://youtu.be/sVvRiy0AM_w

DJI Robomaster Racing

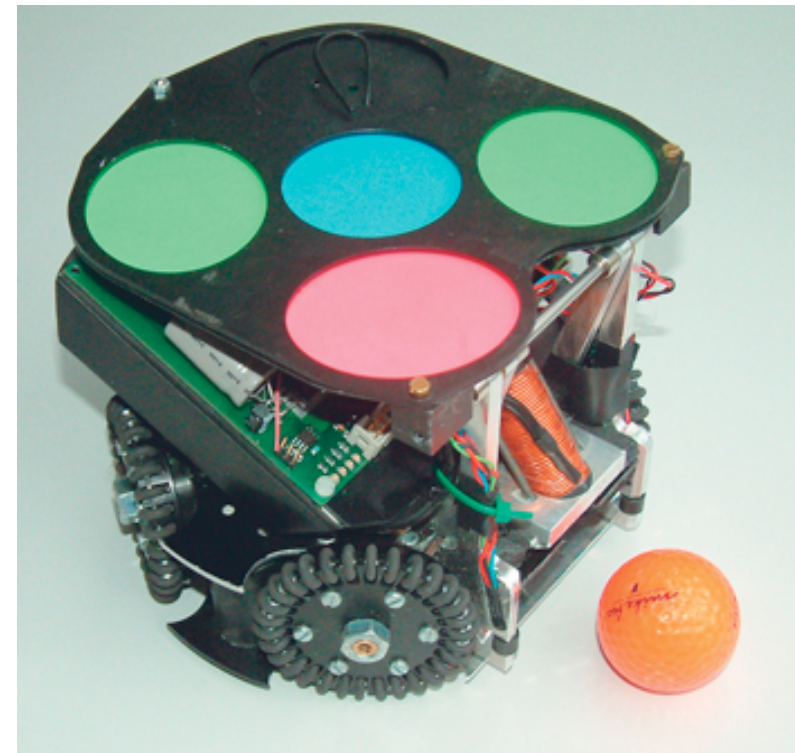


Killough platform



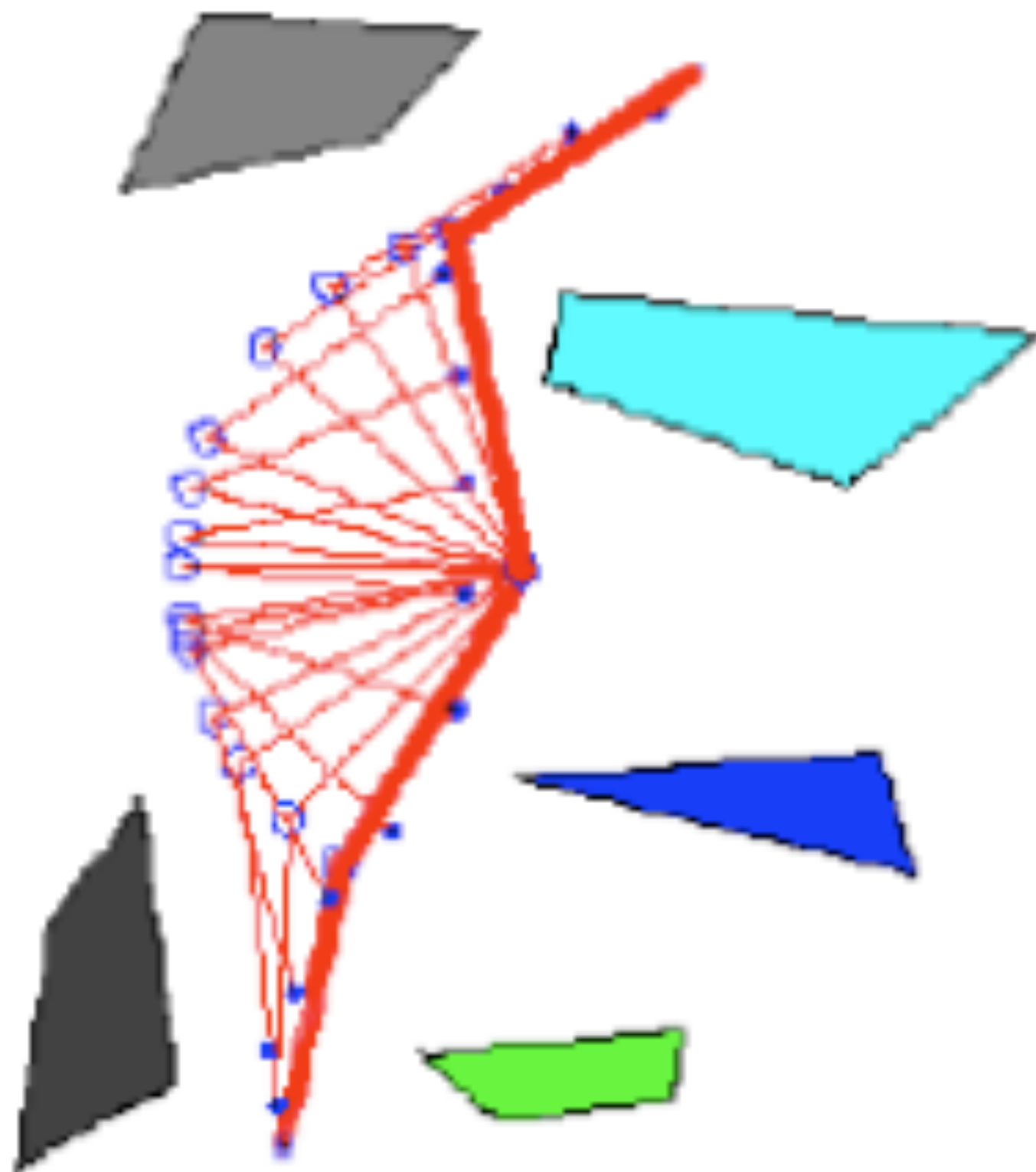
robotthoughts.com; http://technicbricks.blogspot.com/2008/08/going-to-all-places-in-all-directions_29.html

Recommended: D'Andrea on Omni-drive

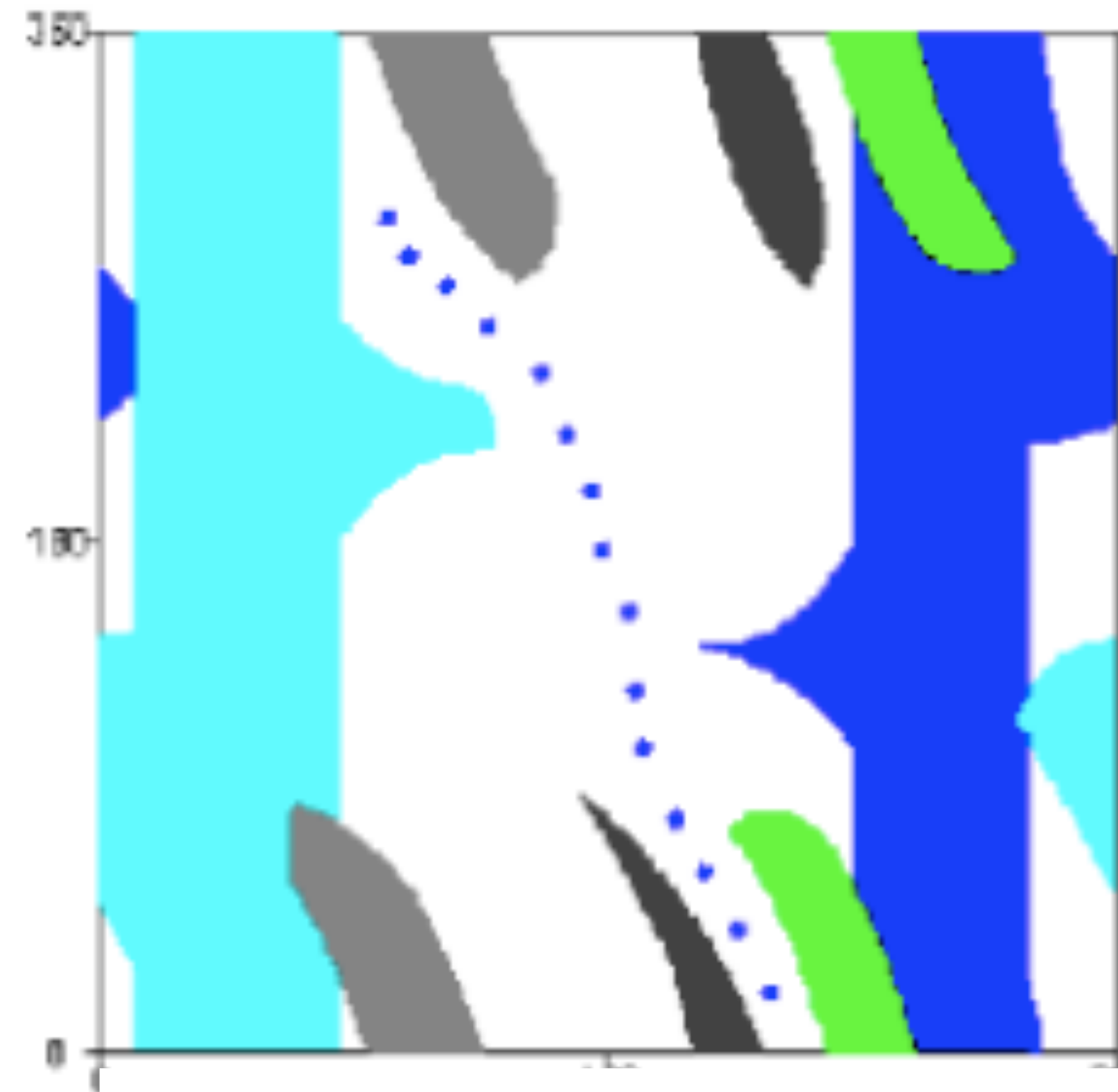


https://www.youtube.com/watch?v=p_WI-C-ORso

How do we search arbitrary C-spaces?



Arm navigation in workspace



C-space representation

How build graphs in arbitrary C-spaces?

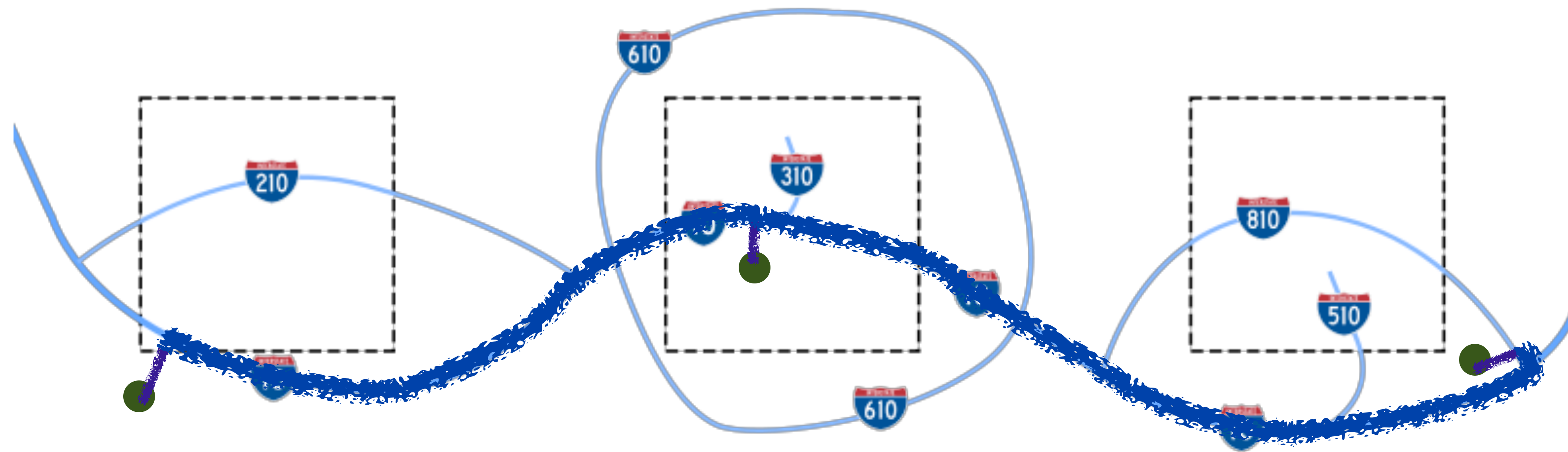
Next class:

Planning with Sampling Roadmaps

CITY A

CITY B

CITY C



Building a path to goal