

Configuration Spaces

EECS 398

Intro. to Autonomous Robotics

ME/EECS 567 ROB 510
Robot Modeling and Control

Fall 2018



autorob.org

more than meets the eye

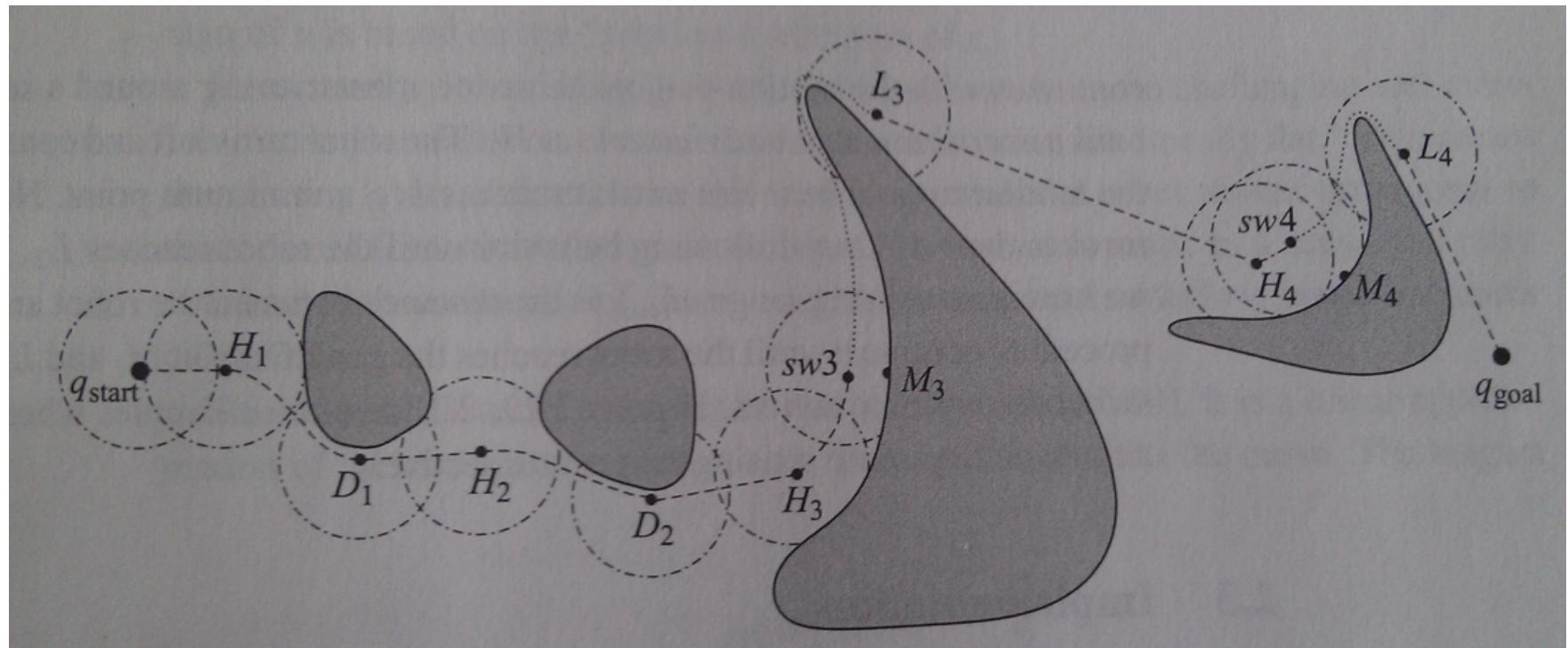
Configuration Spaces



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...

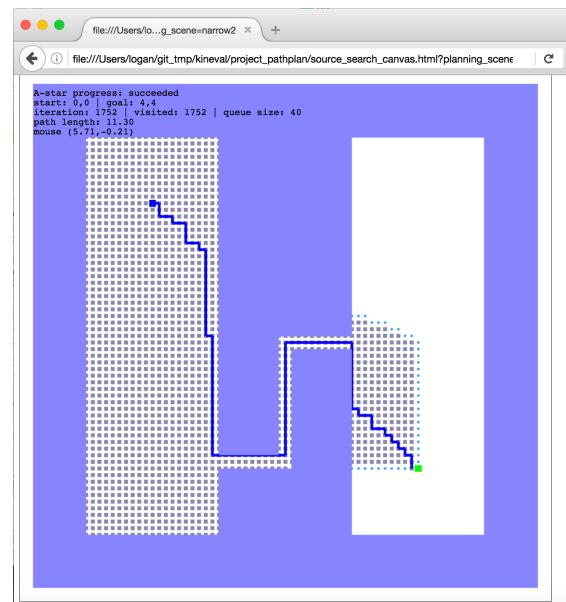
Approaches to motion planning

- 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



michigan ECE 398/567 ROB 510 - autorob.org

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?



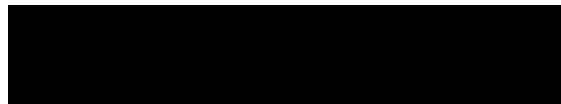
Configuration Space (or C-space)

- 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.

Let's 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?



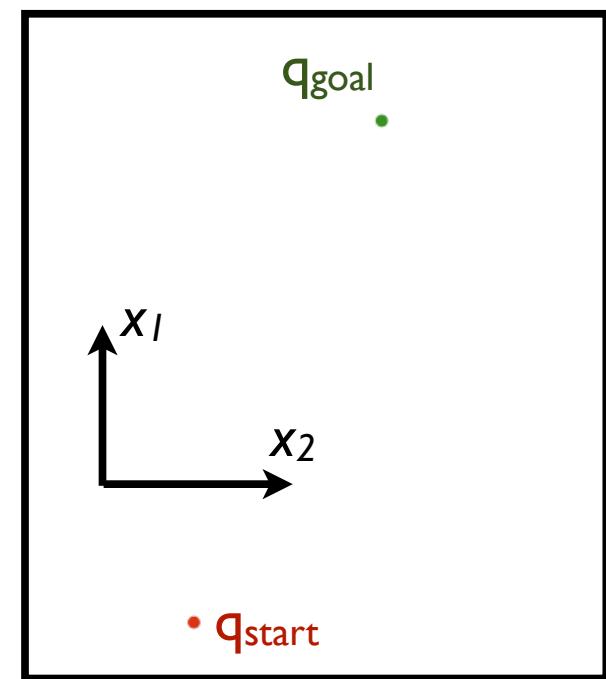
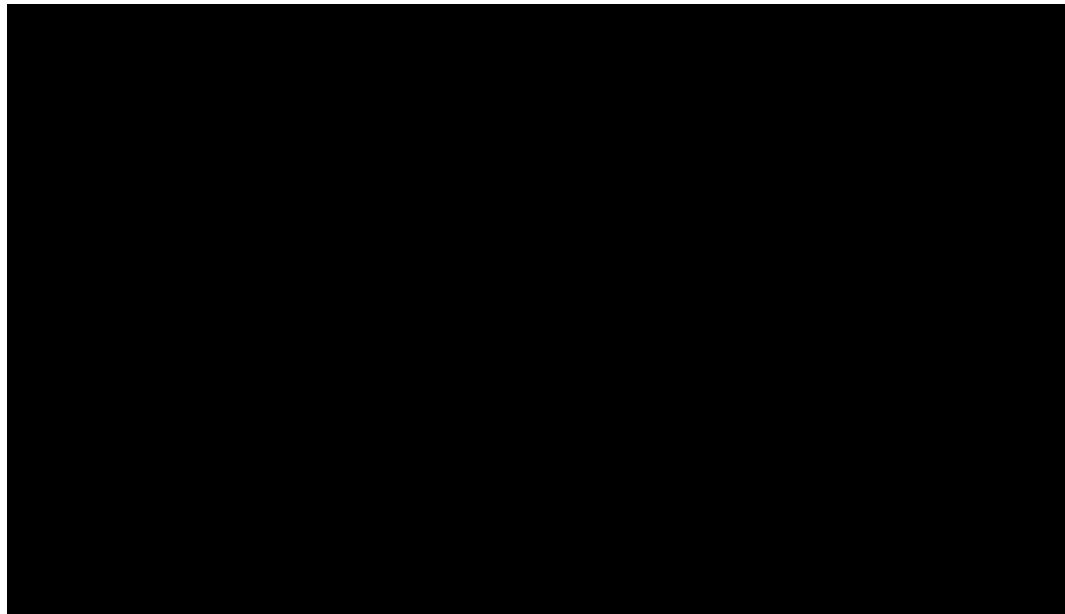
Configuration Space

- 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?

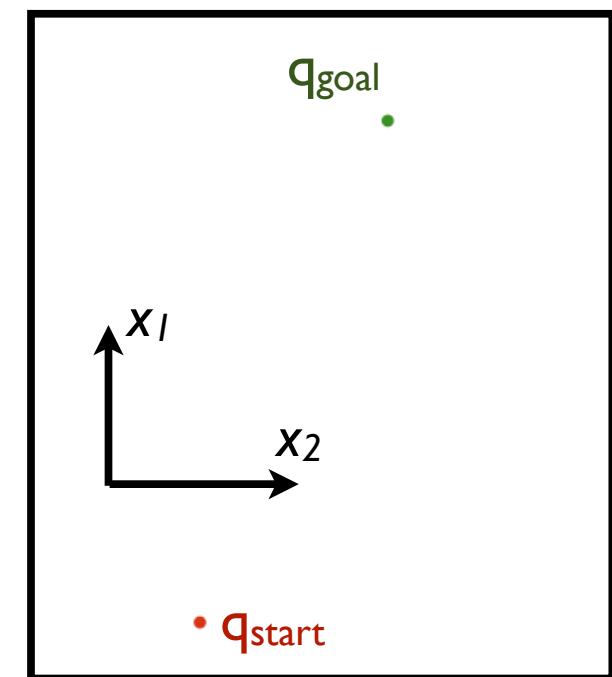


C-space examples

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

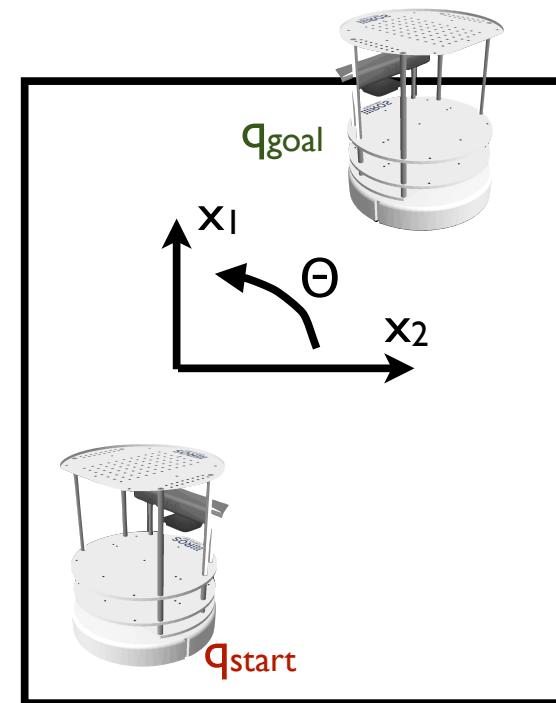
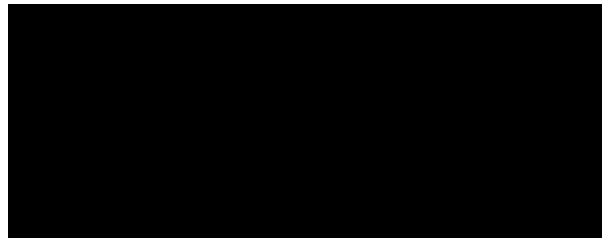
- DOFs: 2, $\{x_1, x_2\}$
- Number of poses is infinite
- C-space: \mathbb{R}^2

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



C-space examples

- What is the C-space of a Turtlebot?



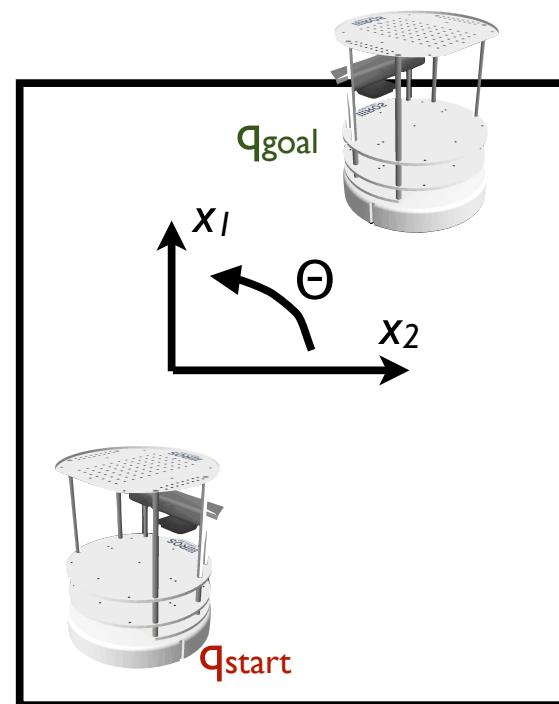
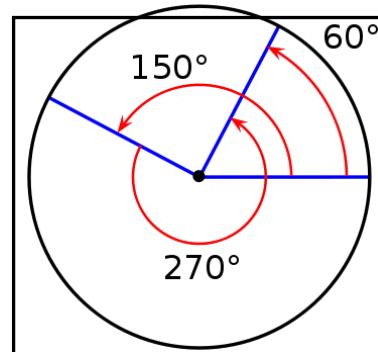
C-space examples

- 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

$S^1 \times S^1 \neq S^n$



C-space examples

- What is the C-space of a Turtlebot?

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

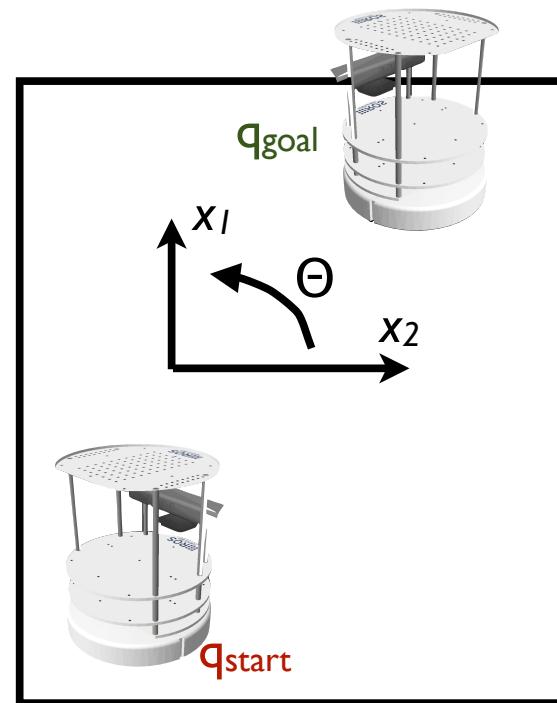
- C-space: $\mathbb{R}^2 \times S^1$

2D translation

rotation in 2D

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

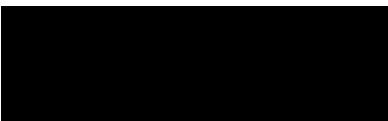
Group of homogeneous transformations in 2D



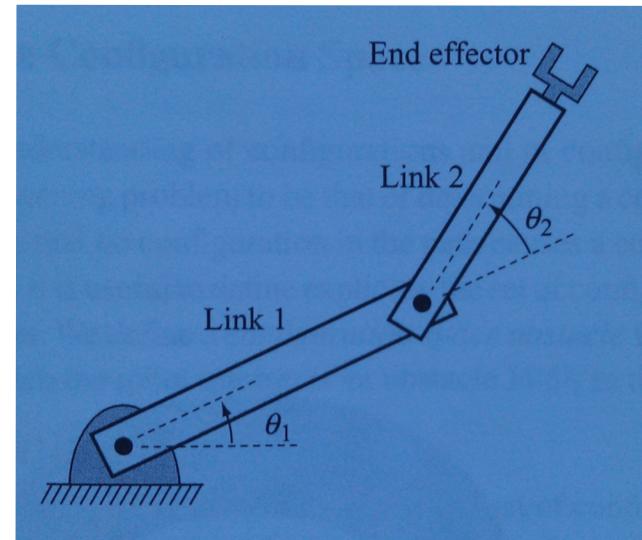
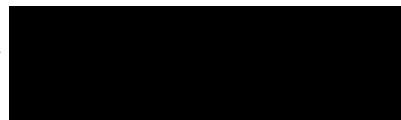
C-space examples

- What is the C-space of a planar arm with 2 rotational joints?

- DOFs:

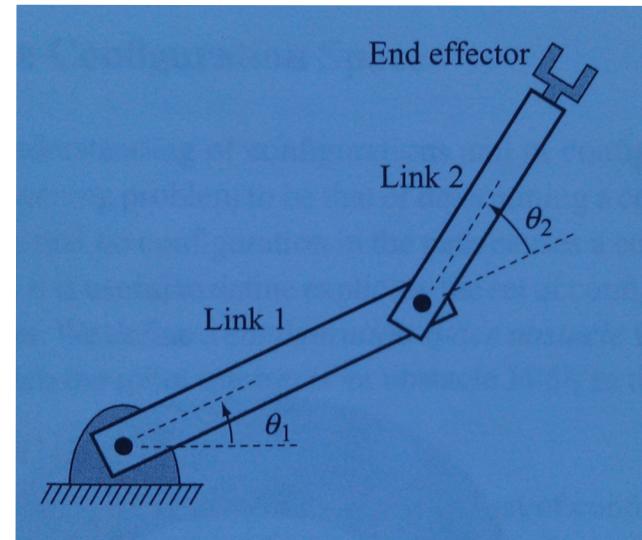
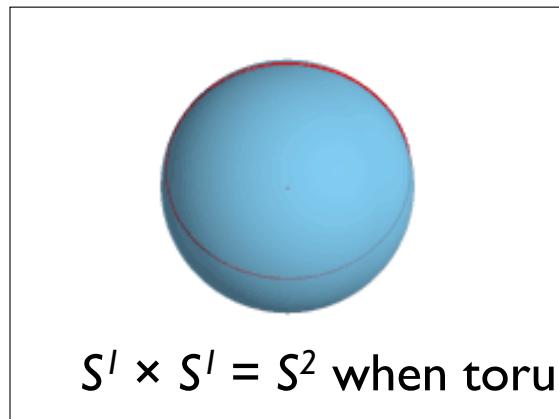


- C-space:

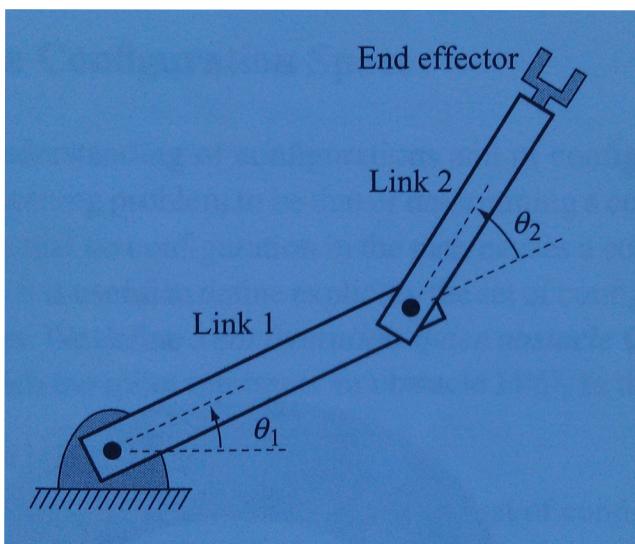


C-space examples

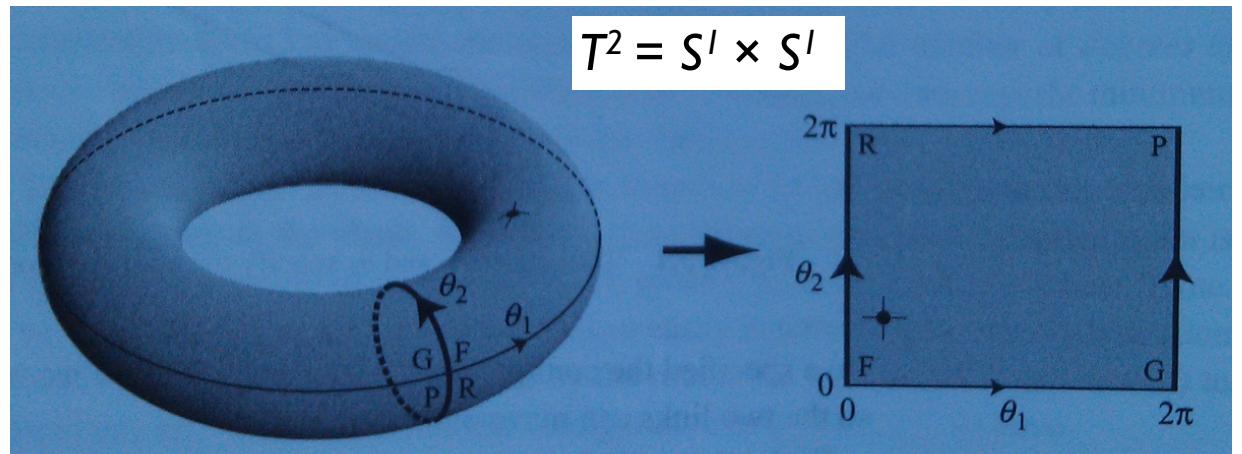
- What is the C-space of a planar arm with 2 rotational joints?
 - DOFs: 2, $\{\Theta_1, \Theta_2\}$
 - C-space: \mathbb{R}^2 or S^2 or $S^1 \times S^1$?



T^2 Torus Group



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

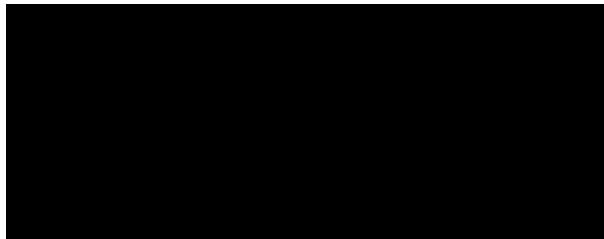


T^n is the torus group for an N-D rotational system

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

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?



Drone Racing League (2017) <https://www.youtube.com/watch?v=NpgGQCv642o>



IROS 2017 Autonomous Drone Racing Competition
<https://www.youtube.com/watch?v=y1DvYkPCnmM>



C-space examples

- What is the C-space of a quad rotor helicopter?
- DOFs: 6
- C-space: $SE(3)$,
- or $\mathbb{R}^3 \times SO(3)$

\nearrow \nearrow
3D translation 3D rotation

\leftarrow Group of homogeneous
transformations in 3D

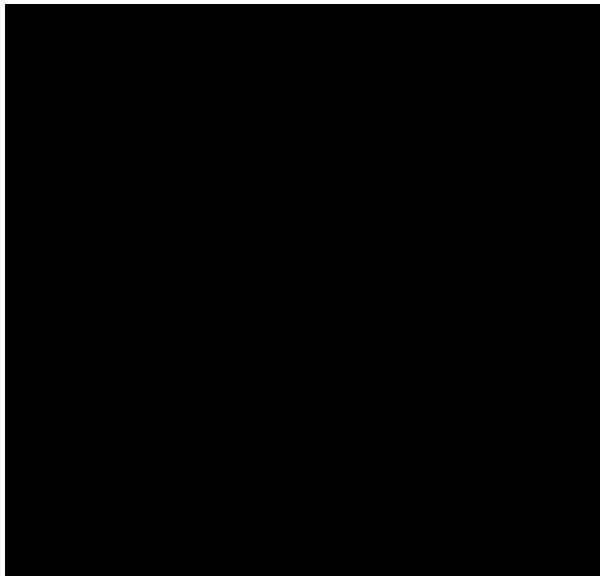


$SE(3)$ combines:
 \mathbb{R}^3 : 3D translation and
 $SO(3)$: 3D rotation

$$SO(3) = S^1 \times S^1 \times S^1$$

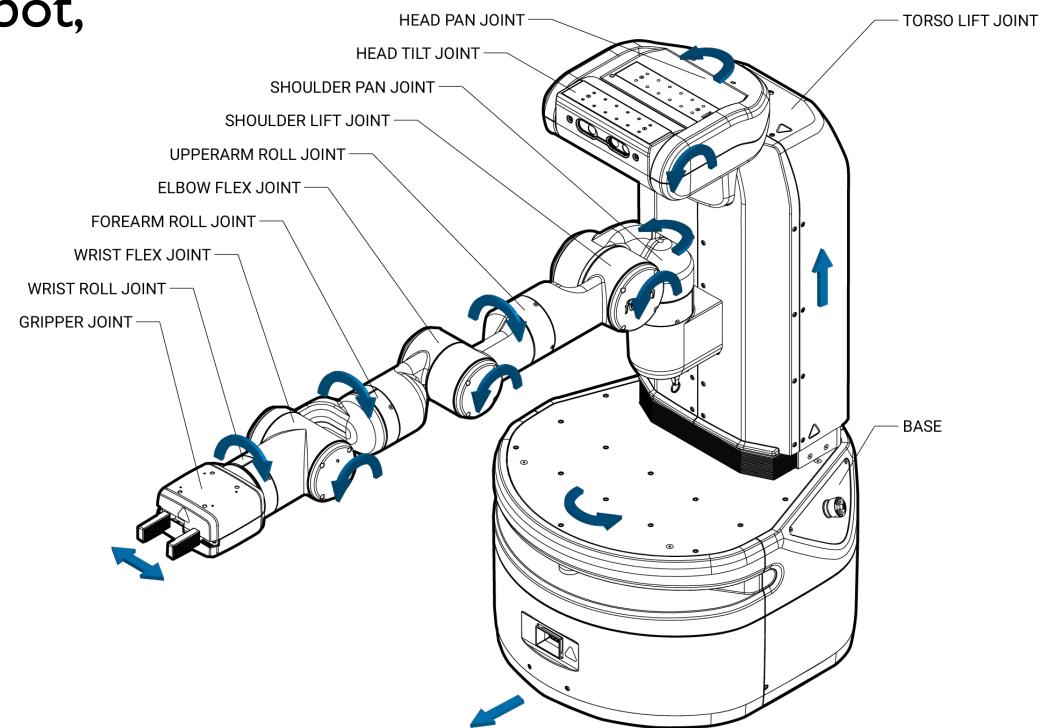
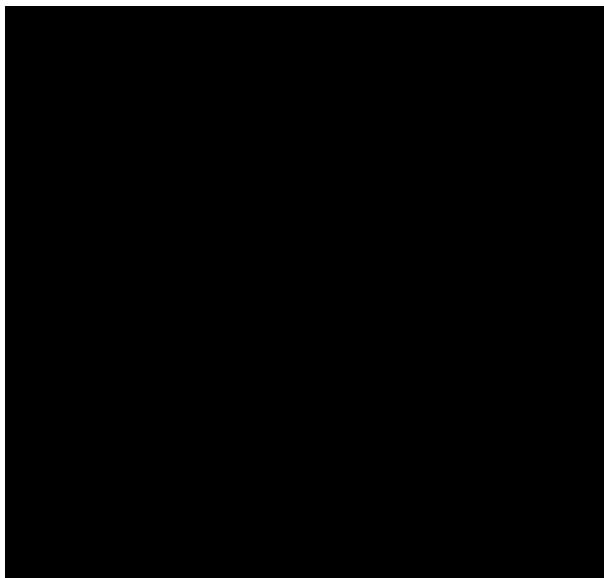
C-space examples

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



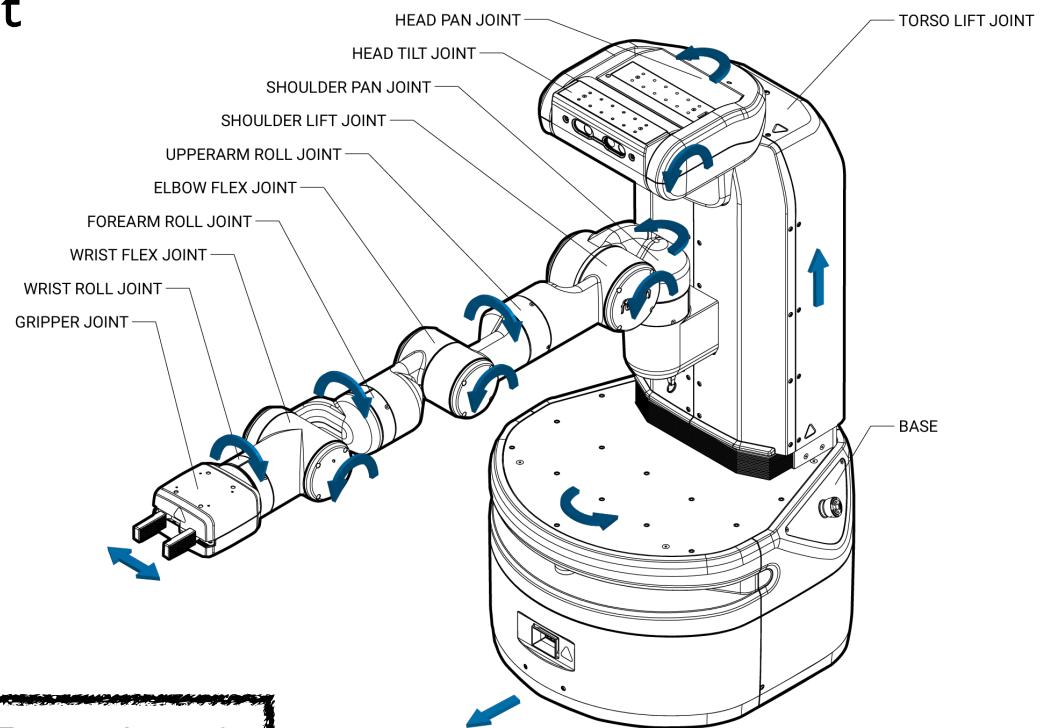
C-space examples

- 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: \mathcal{R}^1
 - 2 in neck: T^2

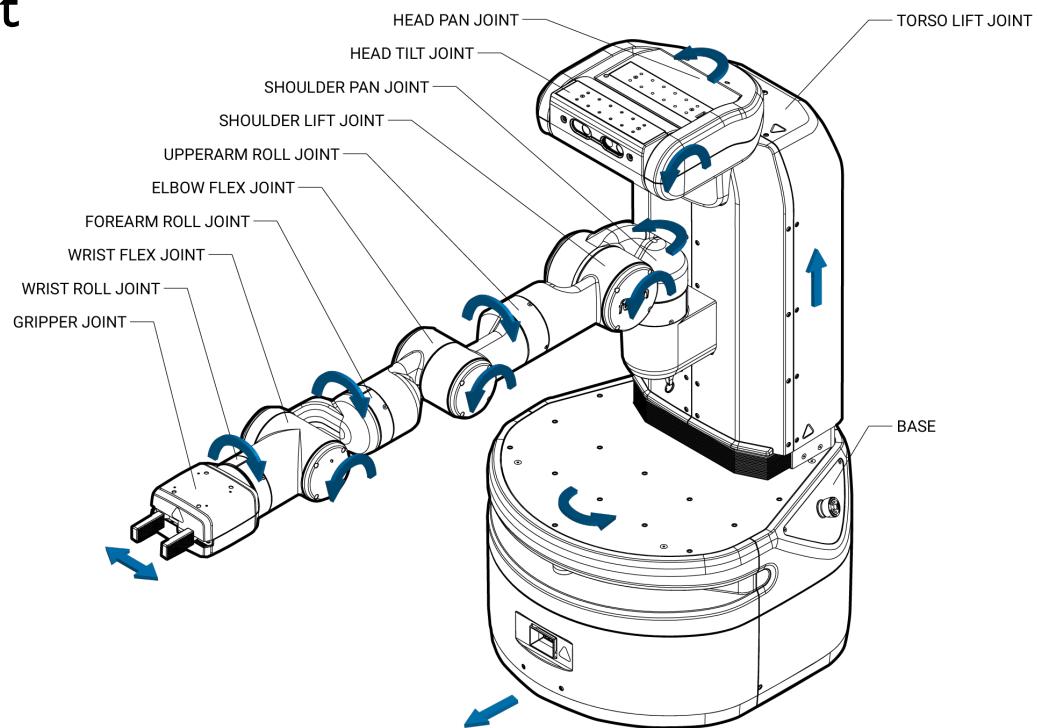


C-space: $SE(2) \times T^7 \times \mathcal{R}^1 \times T^2$

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: \mathcal{R}^1
- 2 in neck: T^2

Consider
joint limits

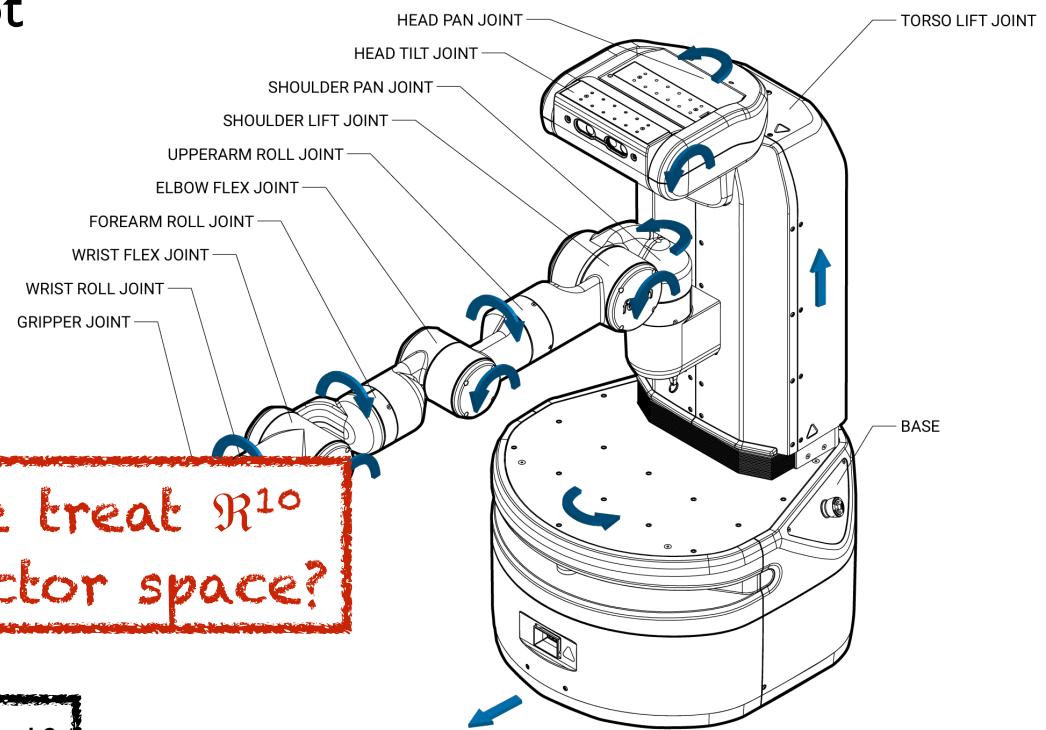


C-space with joint limits

- 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

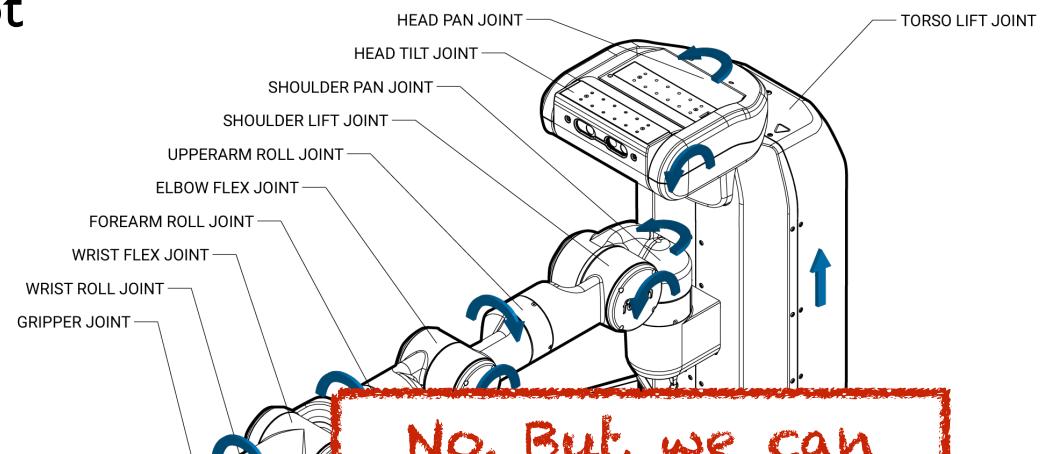
Can we treat \mathbb{R}^{10} as a vector space?

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



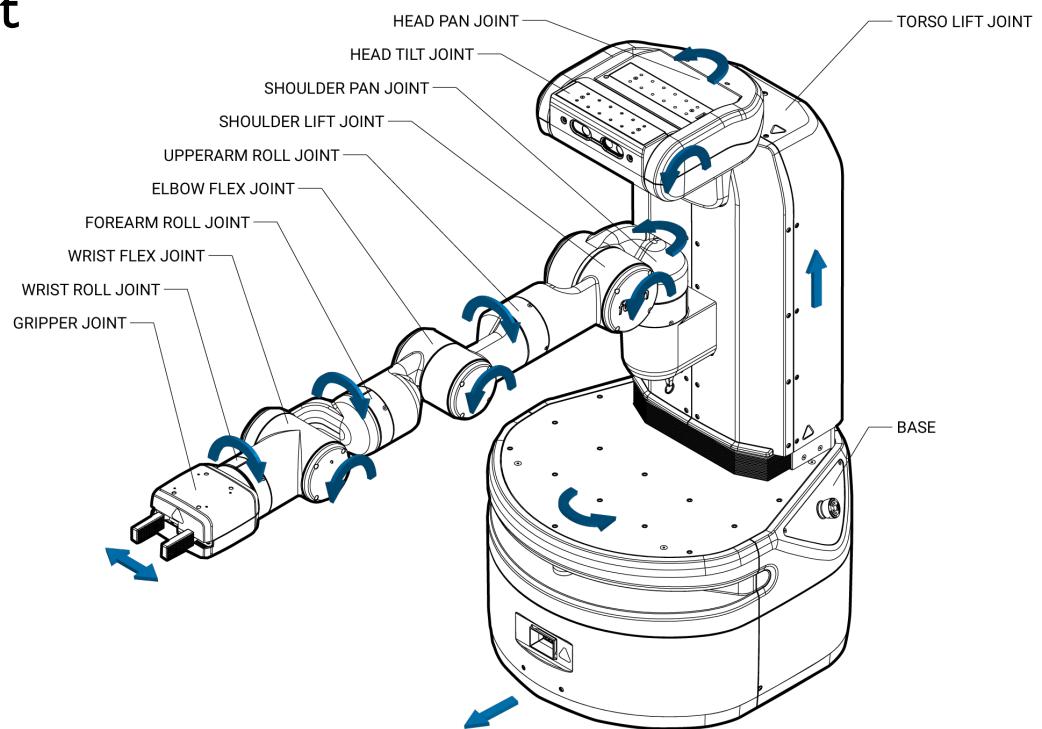
Can we treat \mathbb{R}^{10} as a vector space?

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

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

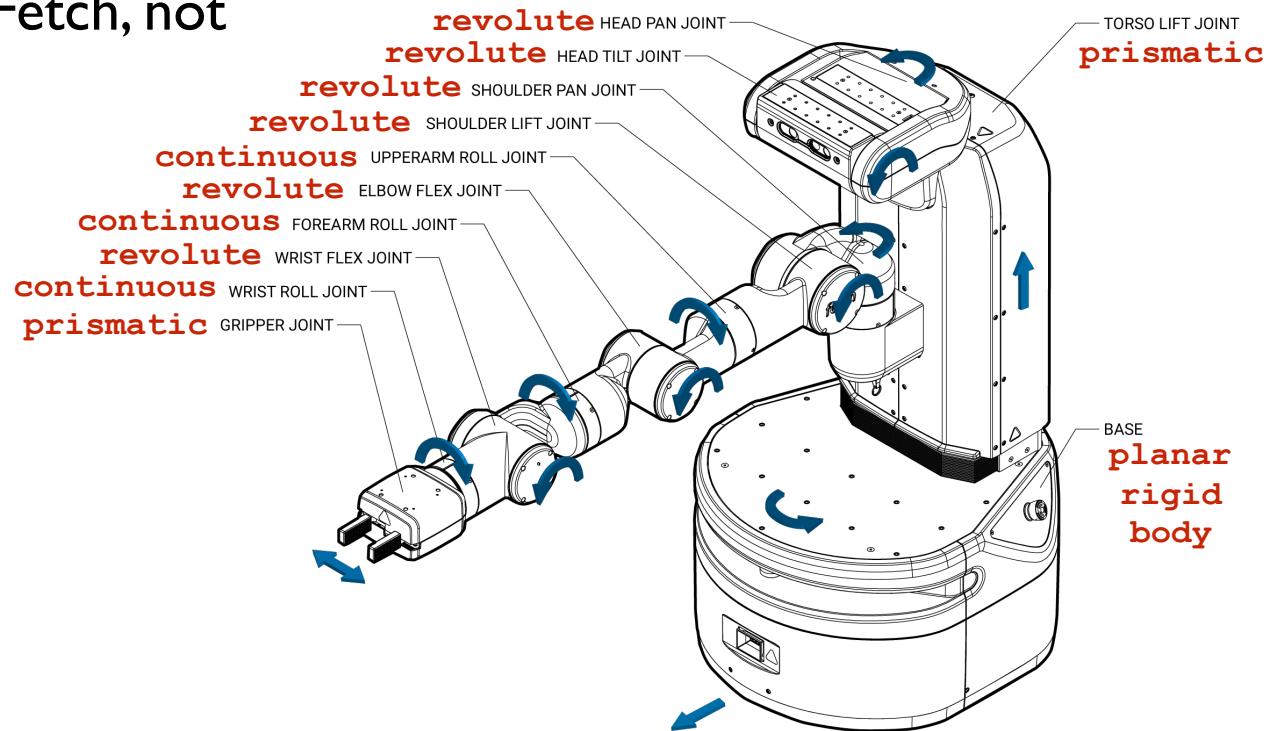
Still not quite right...

- 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



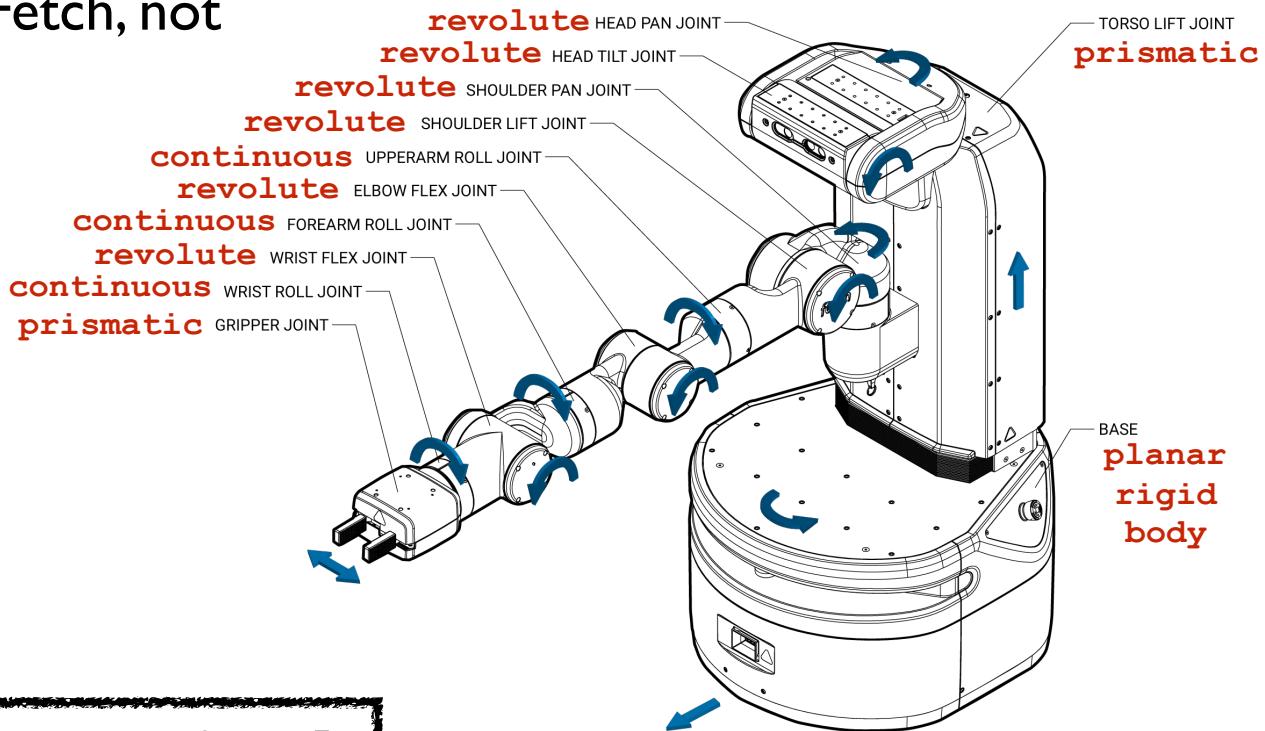
Still not quite right...

- 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



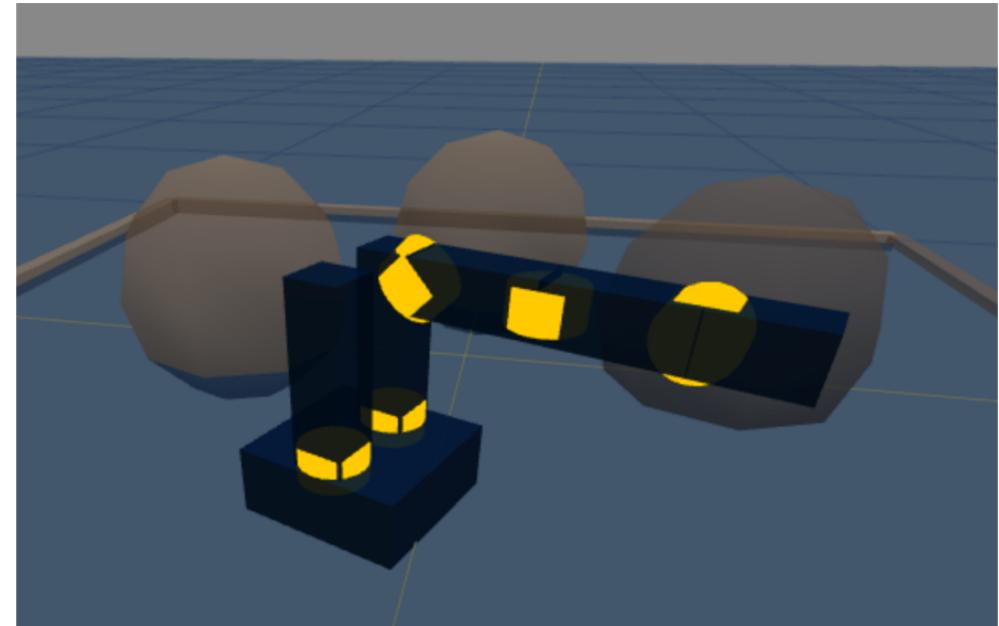
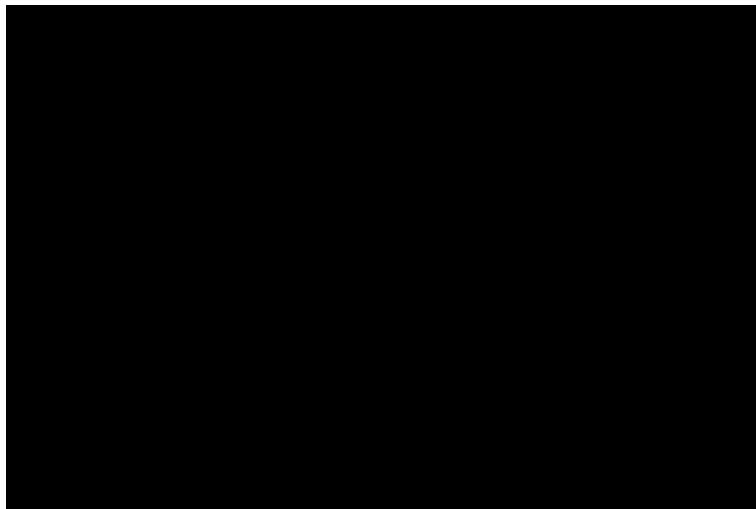
- What is the C-space of a Fetch, not including grippers?
- DOFs: 13
- 3 in base: $SE(2)$
- 3 continuous: T^3
- 1 prismatic: \mathbb{R}^1
- 6 revolute: \mathbb{R}^6

 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

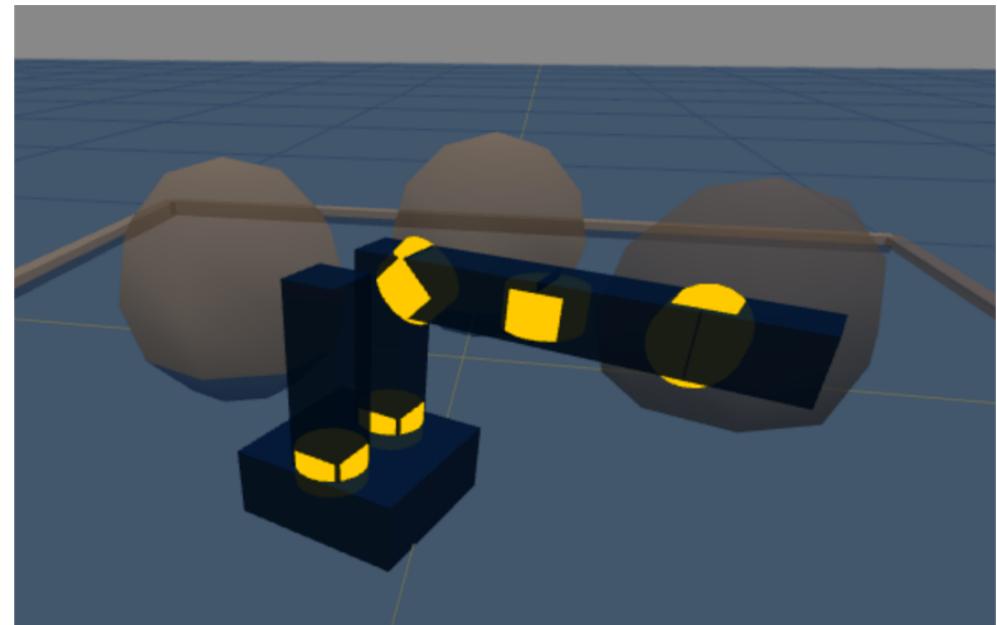
- What is the C-space of a MR2?
- DOFs: 14
 - 3 in base: $SE(2)$
 - 5 in arms: T^5



C-space: $SE(2) \times T^5$



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



C-space examples

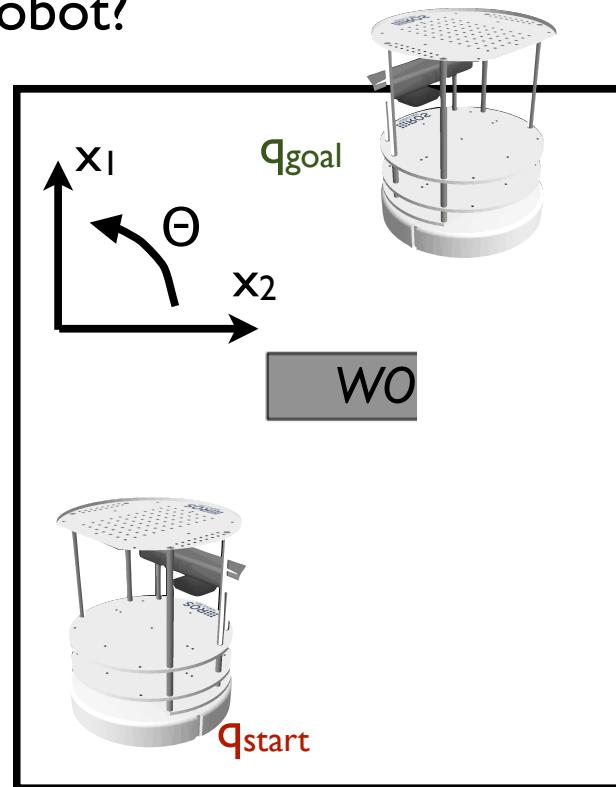
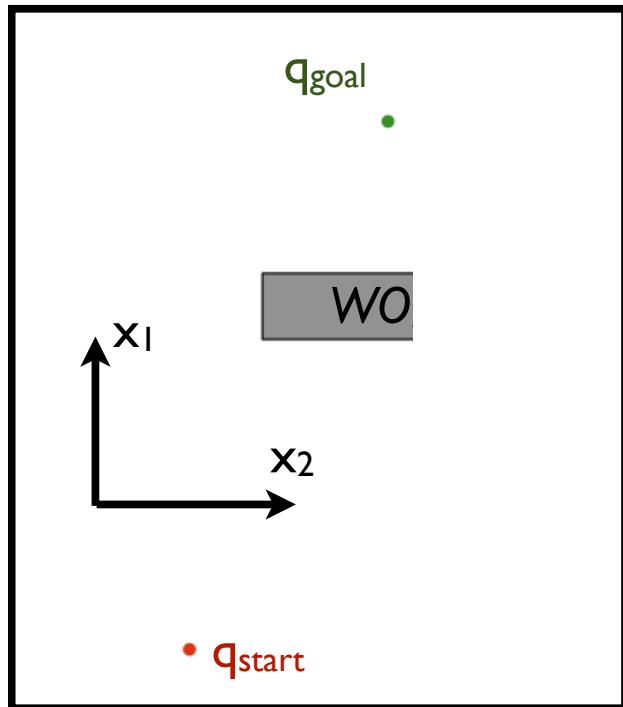
- What is the C-space of a Robonaut 2 on the International Space Station?
- What is the C-space of a PR2?



What about the robot's
physical geometry?

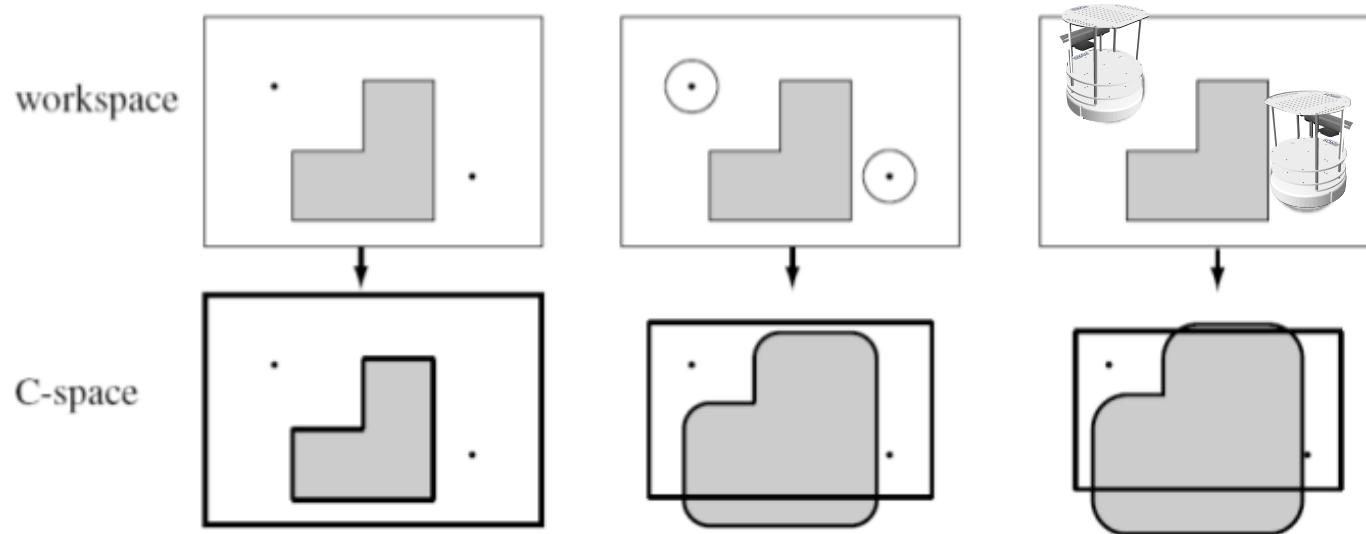
Configuration v. Workspaces

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



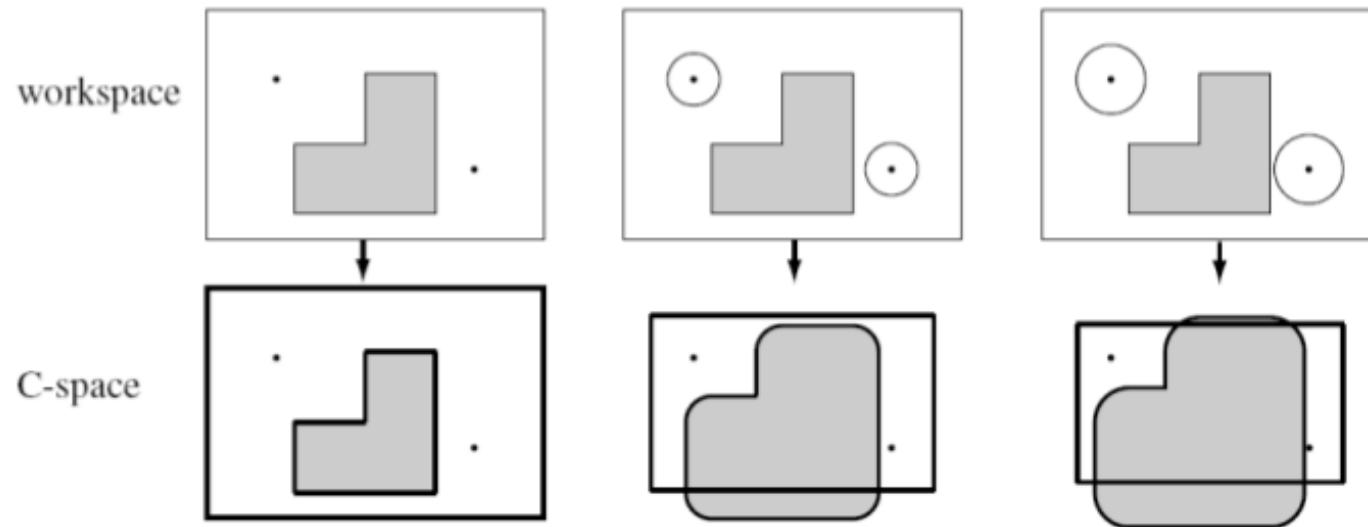
Robot Geometry

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

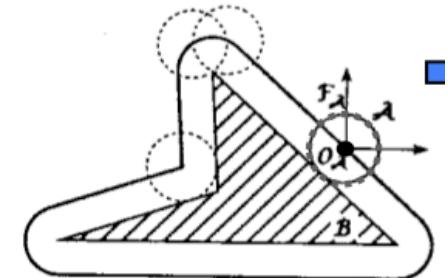


Robot Geometry

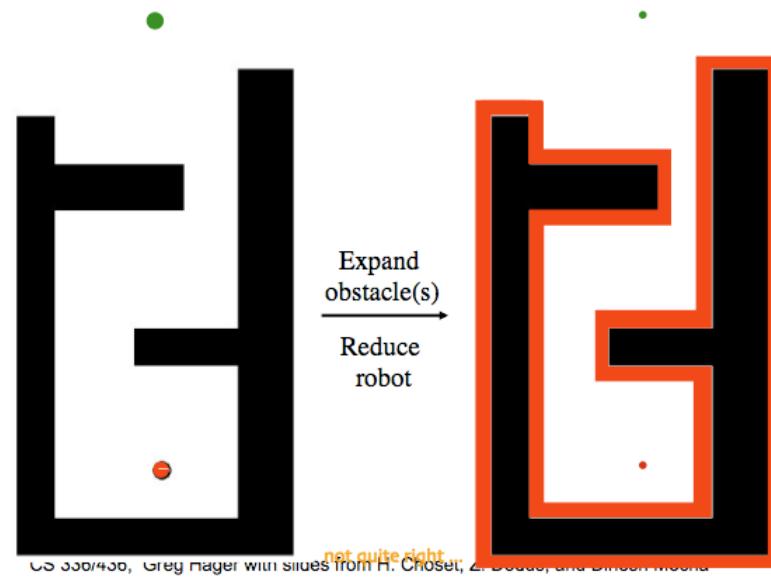
- Turtlebot is larger than a point, having a circular radius in the robot's planar workspace
- As this radius increases, the C-space shrinks



Conversion to point robot C-space

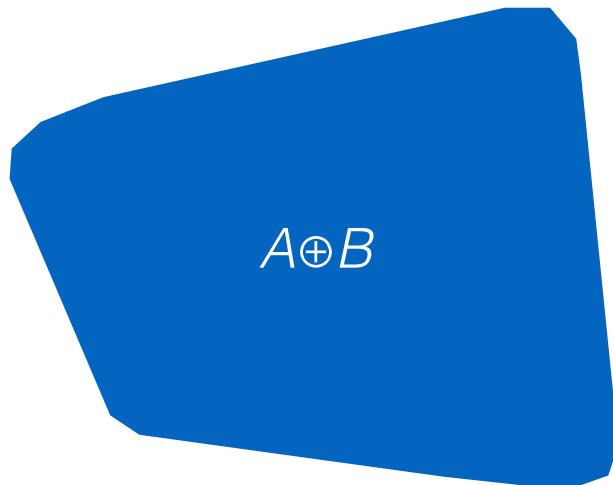
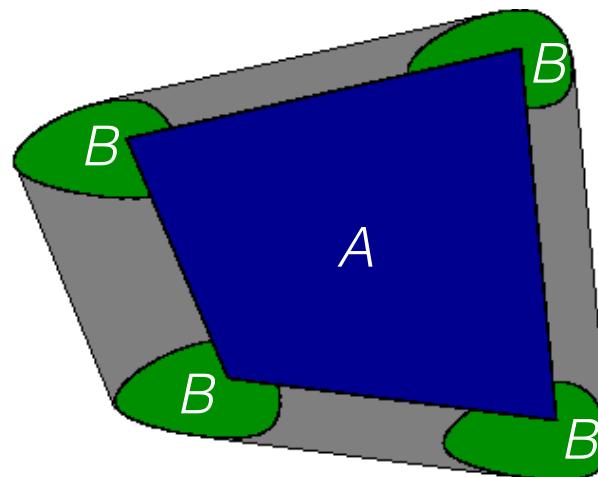


- Workspace for robot can be converted to point robot C-space
- Expand obstacles by tracing robot geometry along boundary
- Performed by Minkowski sum



Minkowski Sum (or morphological dilation)

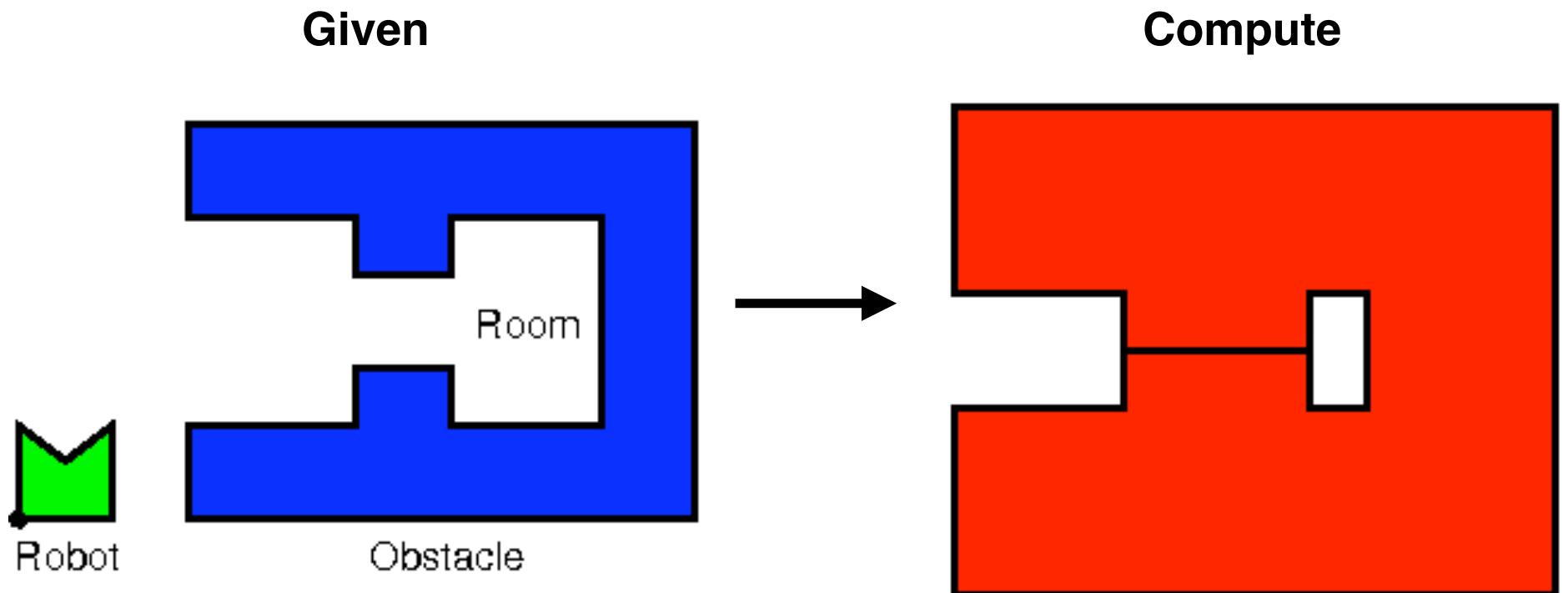
- $A \oplus B: \{a+b \mid a \in A, b \in B\}$
- The result of adding every element of A to every element of B, given two sets A and B in Euclidean space
- Similarly, Minkowski difference (morphological erosion)
- $A \ominus B: \{a-b \mid a \in A, b \in B\}$



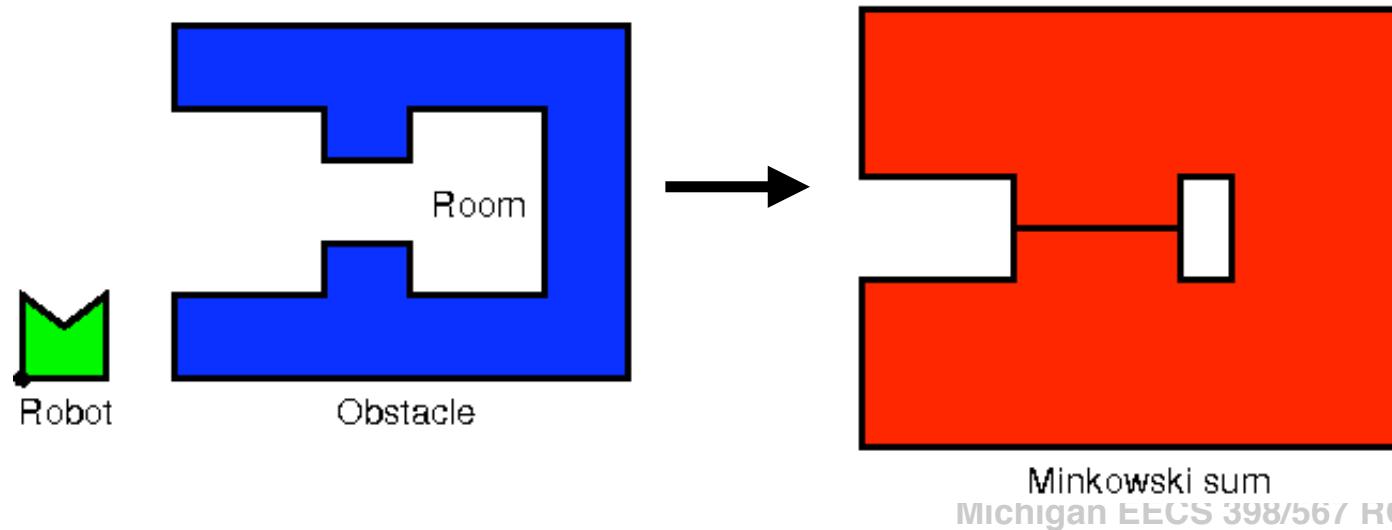
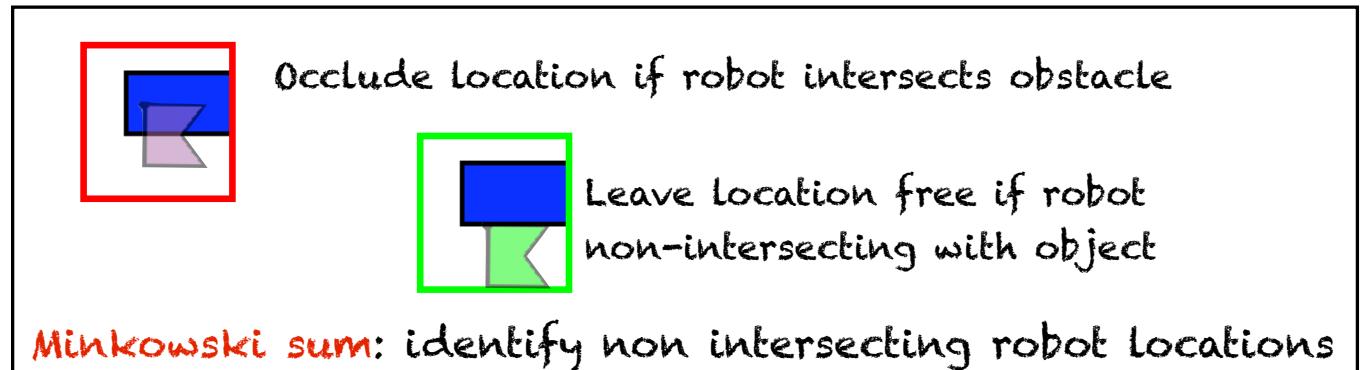
https://golem.ph.utexas.edu/category/2011/08/mixed_volume.html

Michigan EECS 398/567 ROB 510 - autorob.org

Minkowski Planning

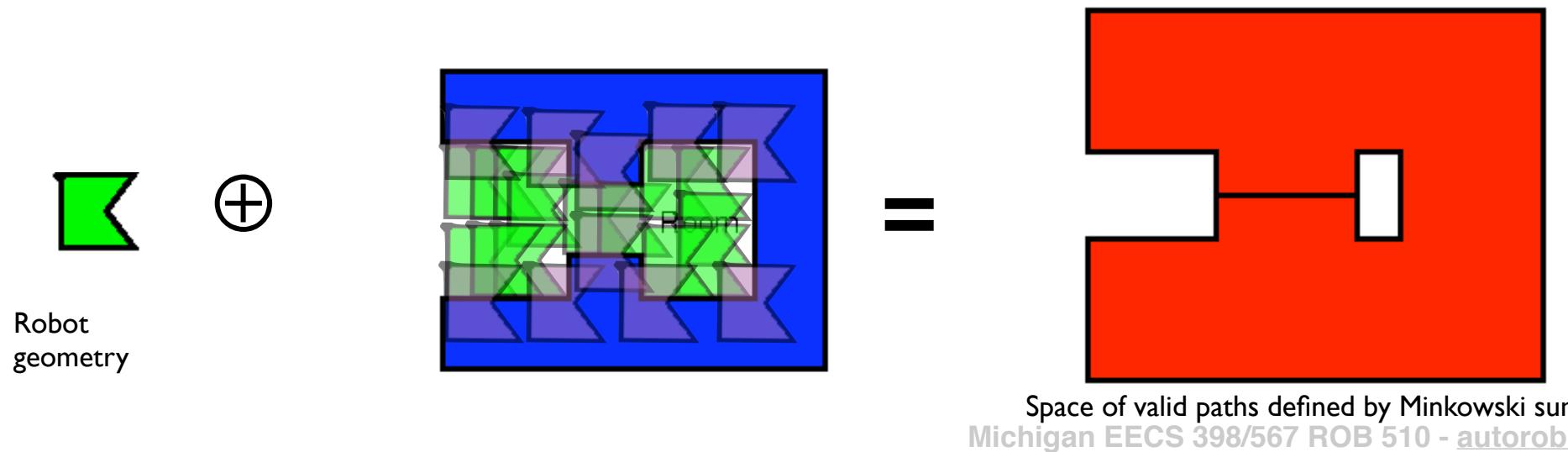
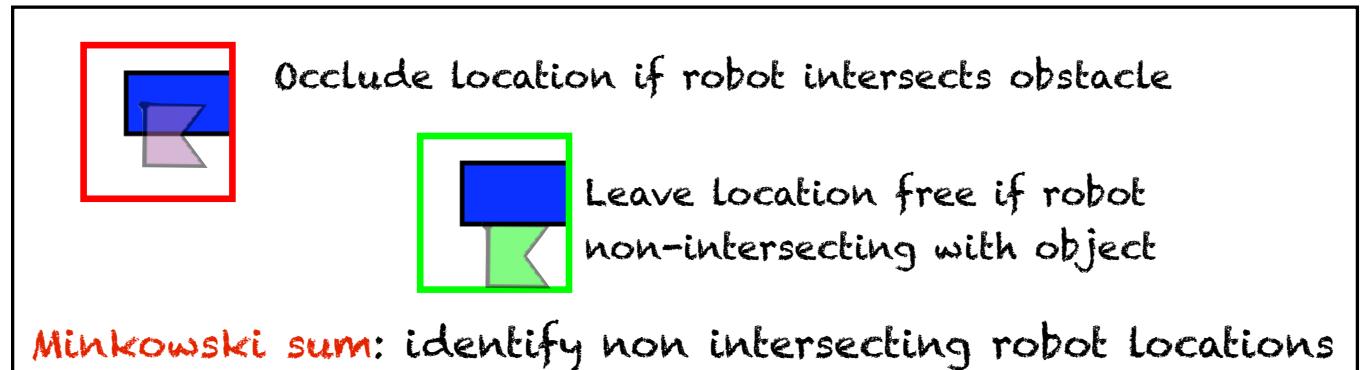


Minkowski Planning

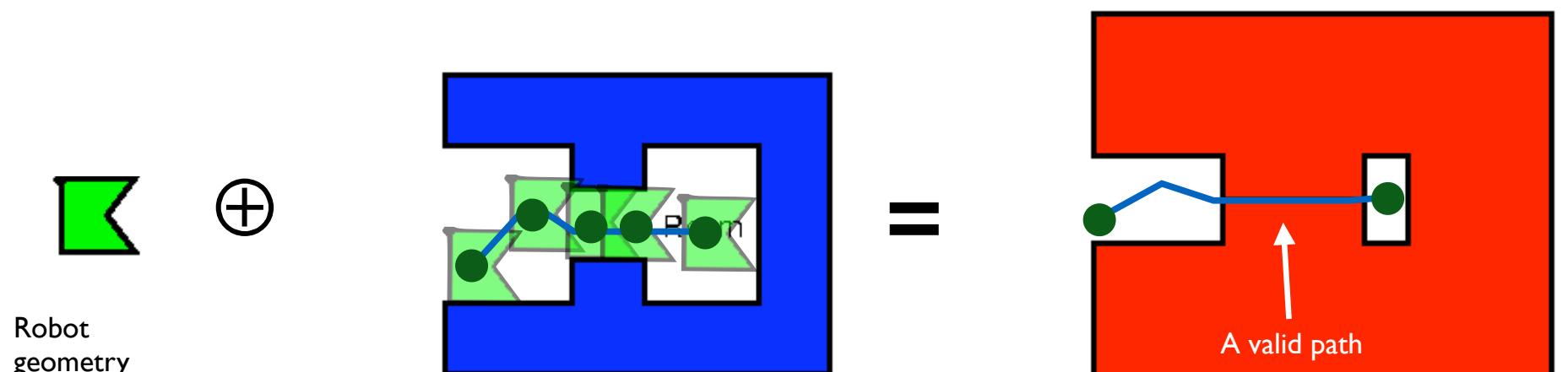
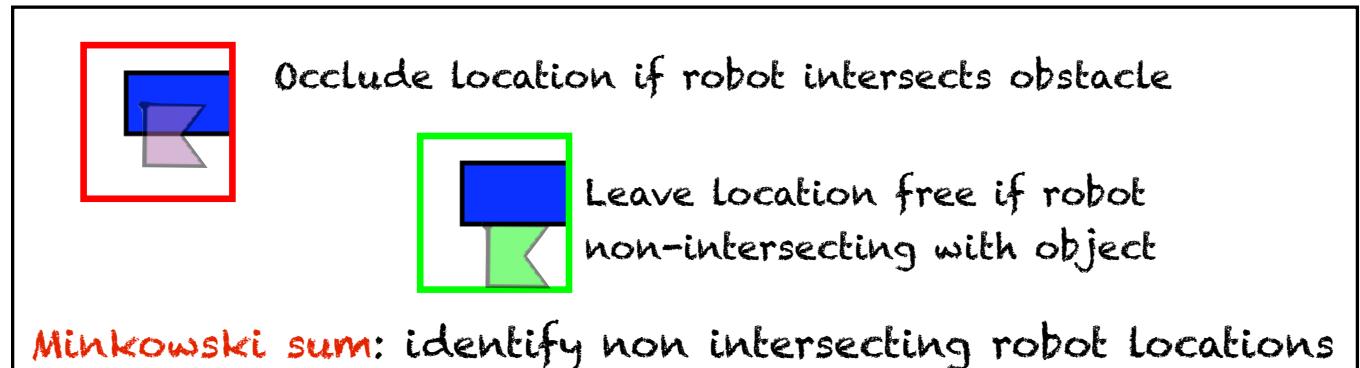


Minkowski sum
Michigan EECS 398/567 ROB 510 - autorob.org

Minkowski Planning



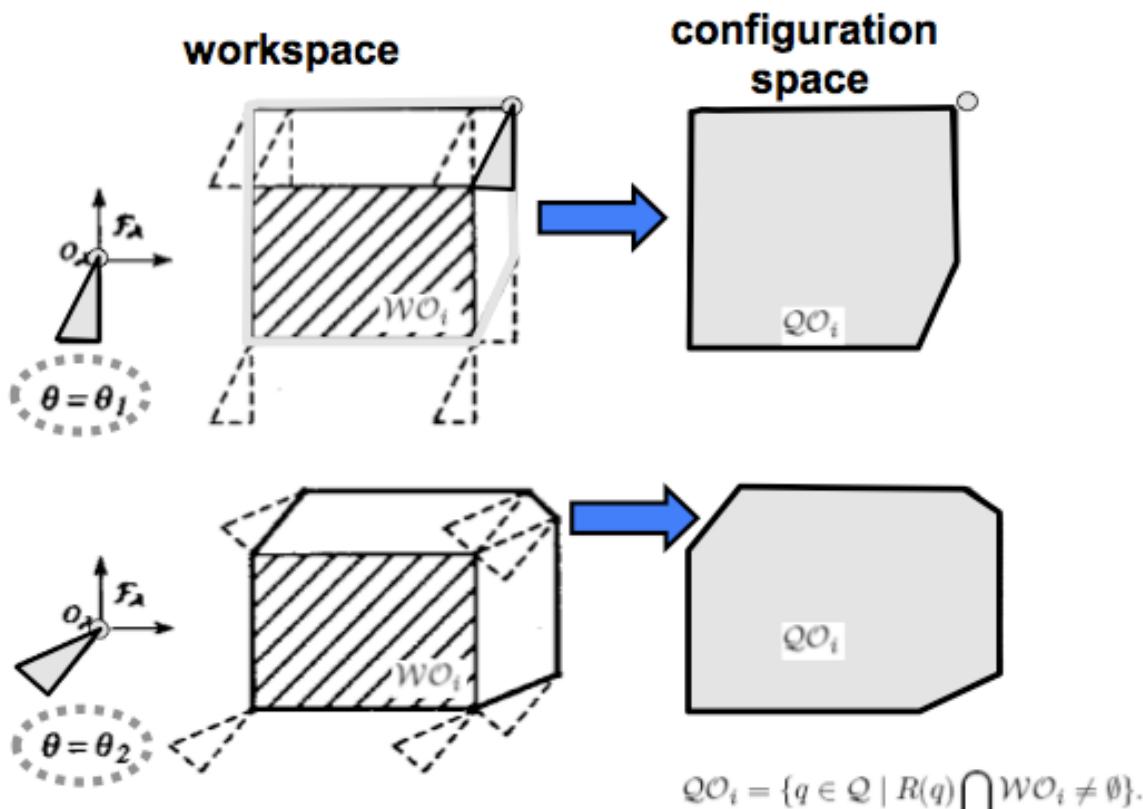
Minkowski Planning



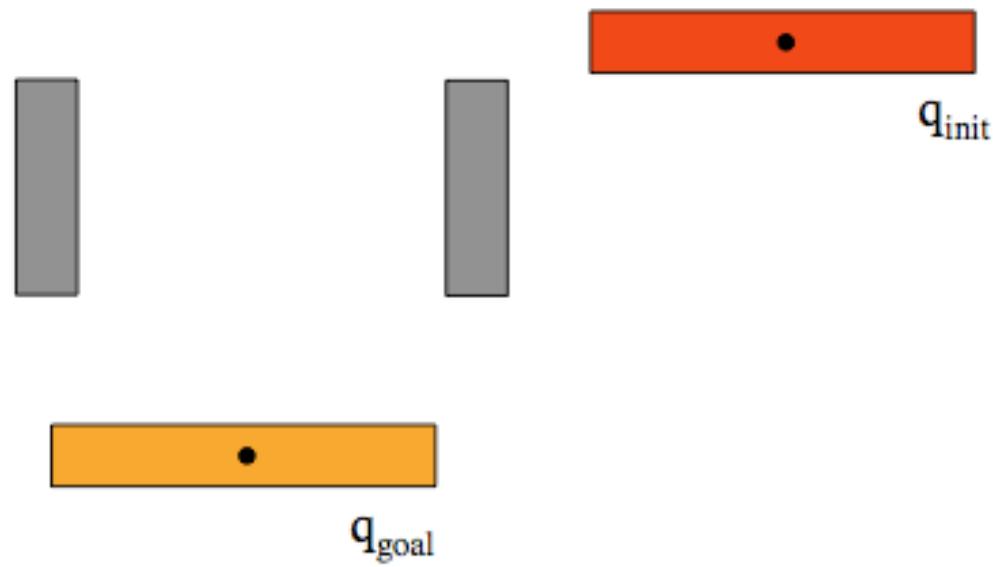
Space of valid paths defined by Minkowski sum
Michigan EECS 398/567 ROB 510 - autorob.org

What does an obstacle look like
in configuration space?

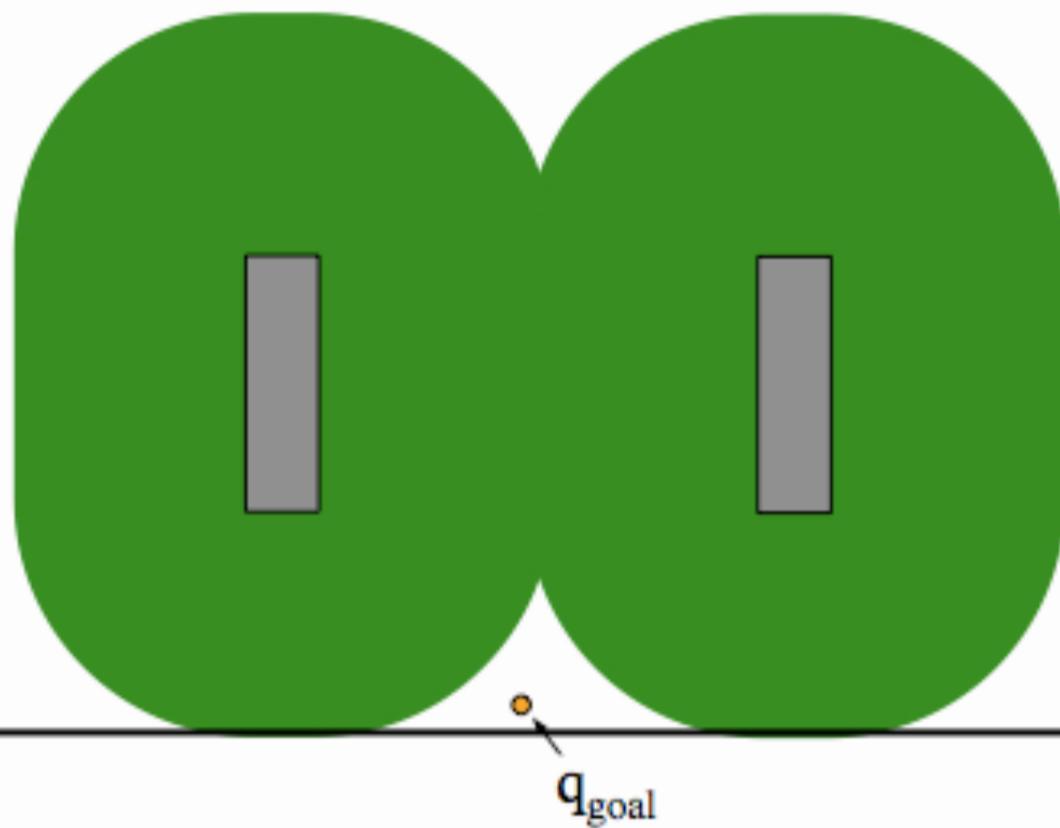
C-space depends on rotation

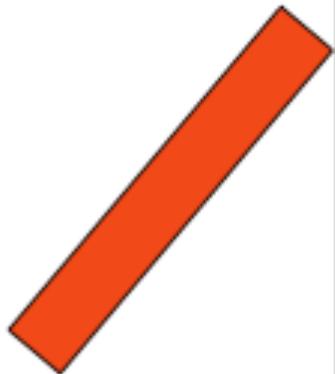


Consider this workspace...

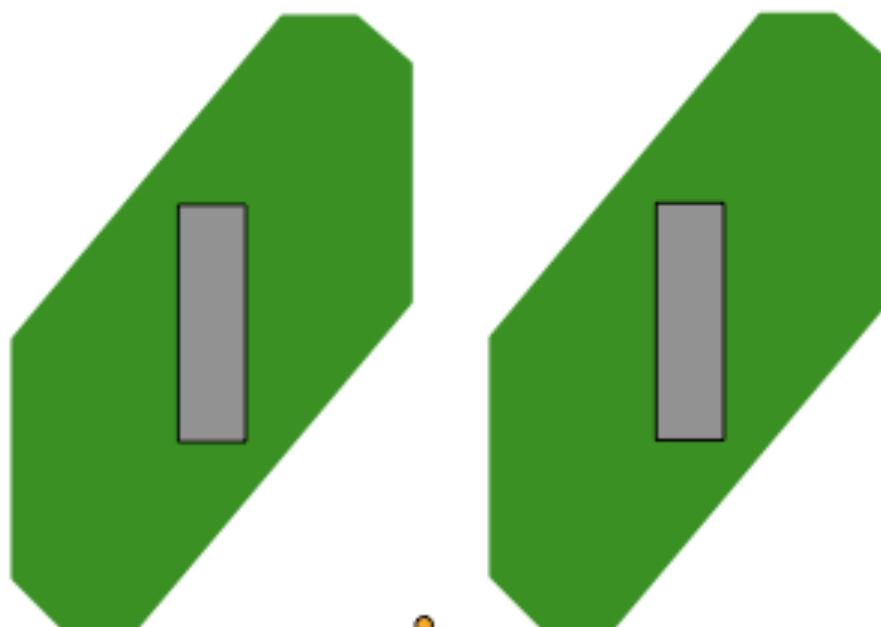


C-space over all possible object rotations





C-space with object constrained to 45 degree rotation

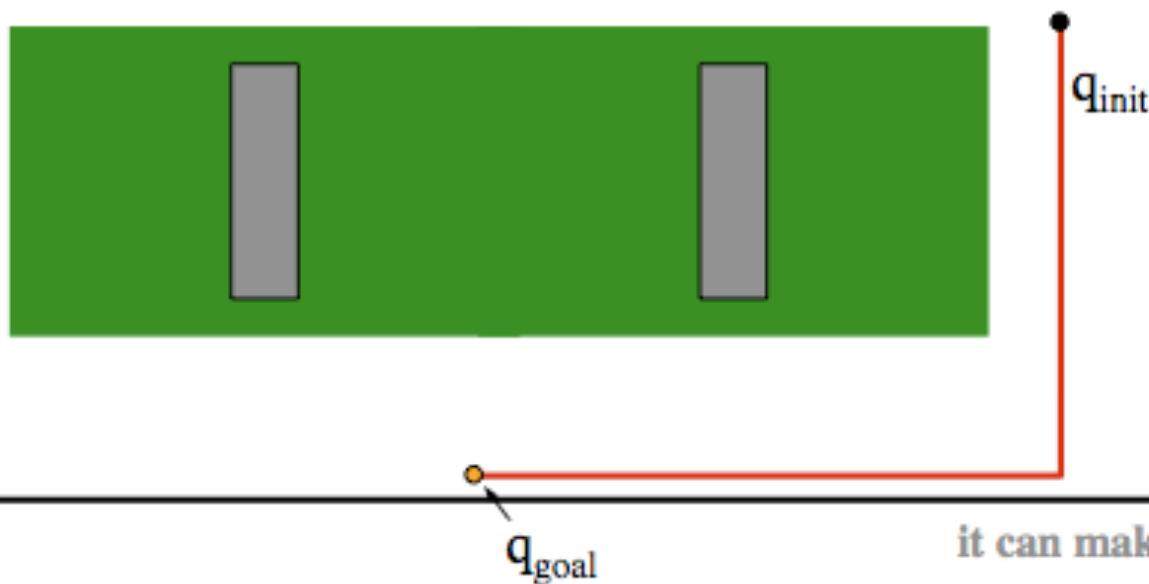


q_{goal}

q_{init}

it depends...

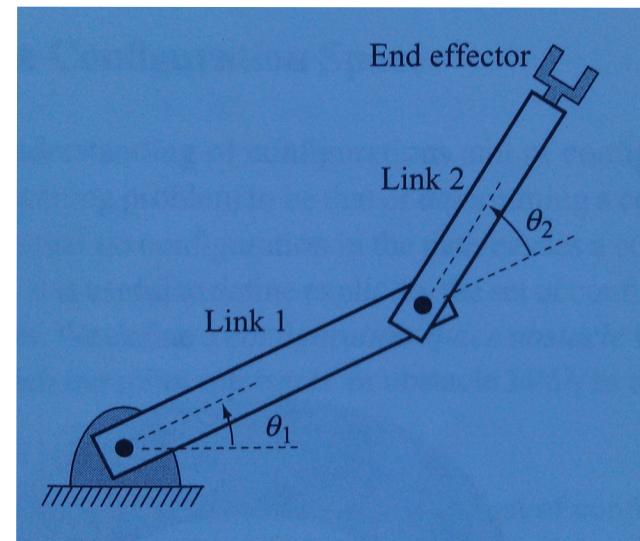
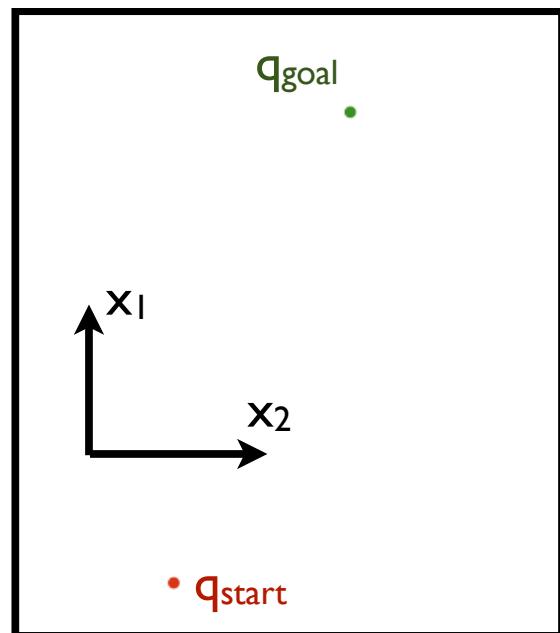
C-space with object constrained to zero degree rotation

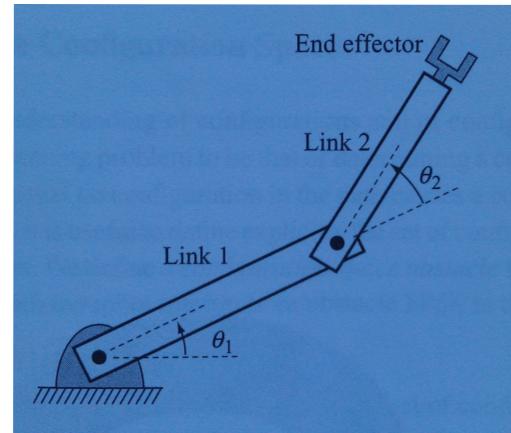


it can make it...

Configuration v. Workspaces

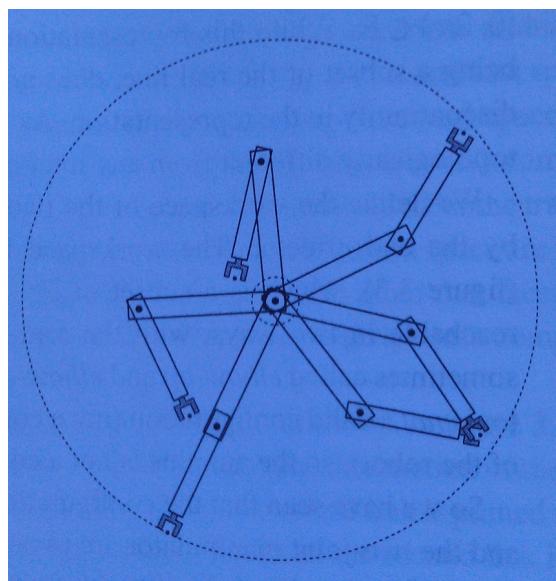
- Other than rotation and geometry, how is the 2-link arm different than the point robot?



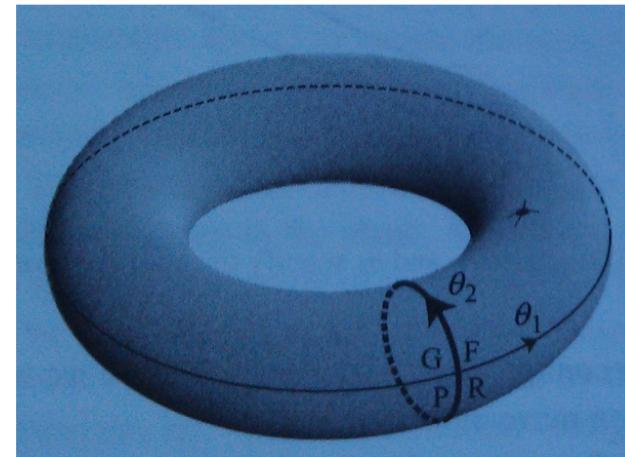


●
(x,y)

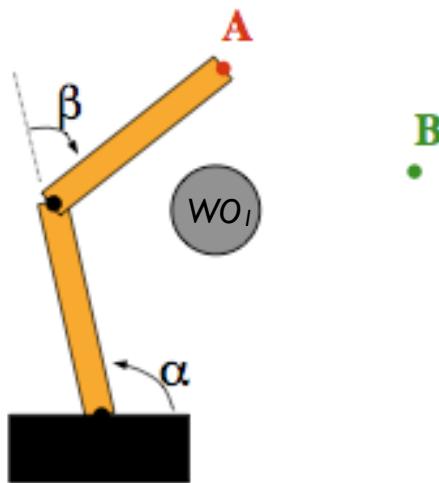
Workspace is w.r.t. end-effector position (x,y)



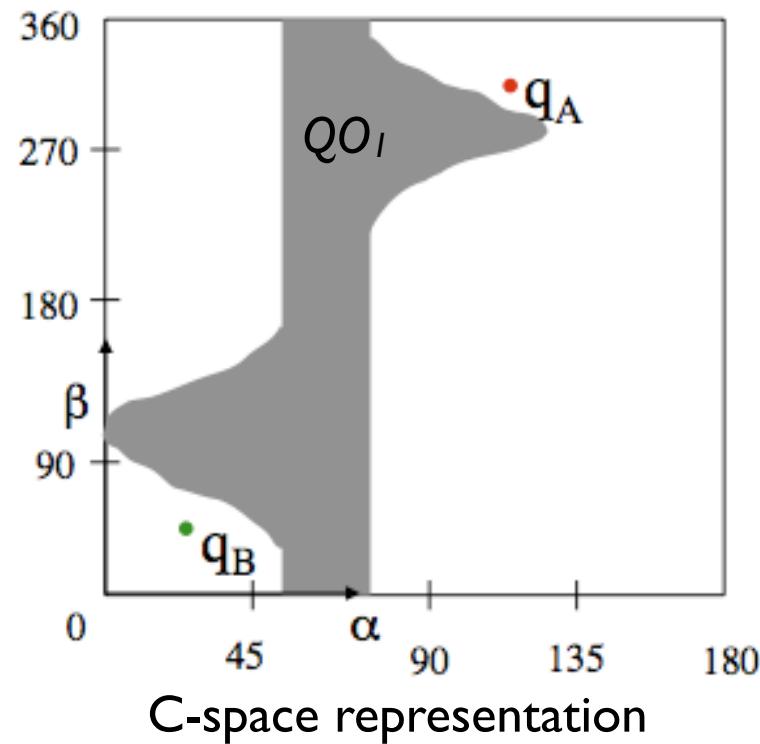
C-space is w.r.t. joint angles (Θ_1, Θ_2)



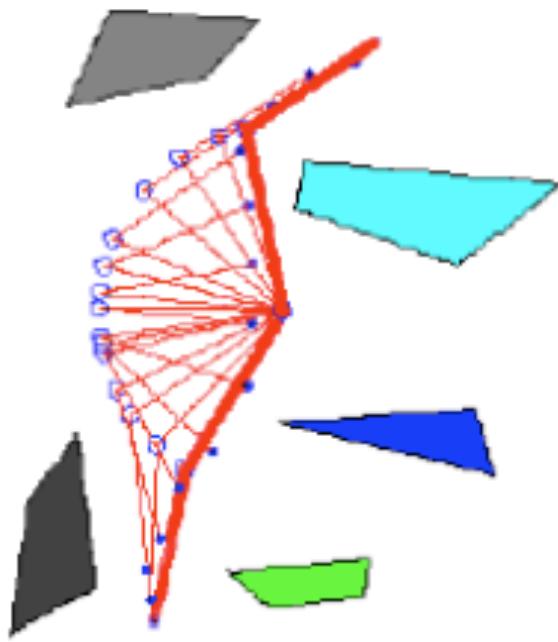
Obstacles in T^2



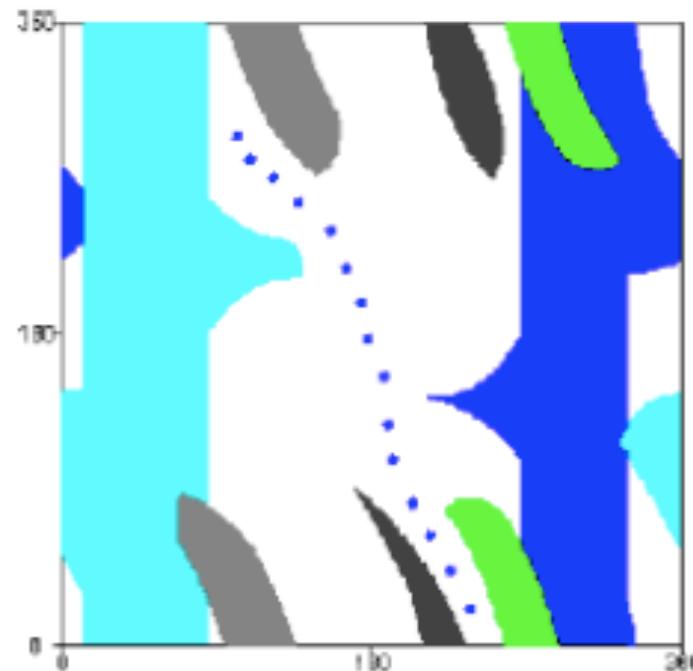
Circular obstacle in workspace



Path in T^2 with several obstacles



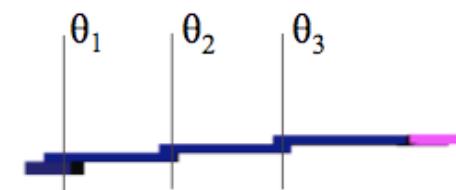
Arm navigation in workspace



C-space representation

C-space for 3-link arm

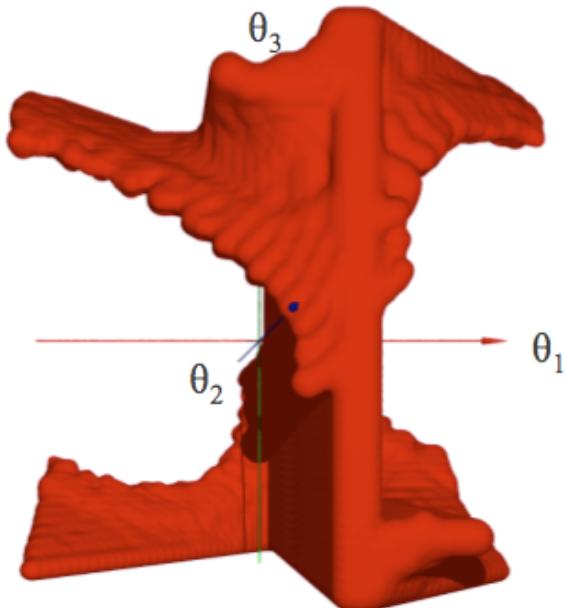
The Configuration Space (C-space)



TOP
VIEW



workspace



C-space

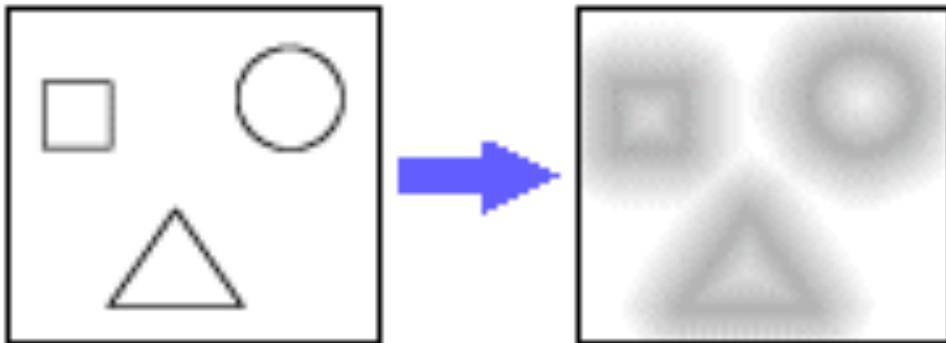
Generalizing graph search for robot configurations

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

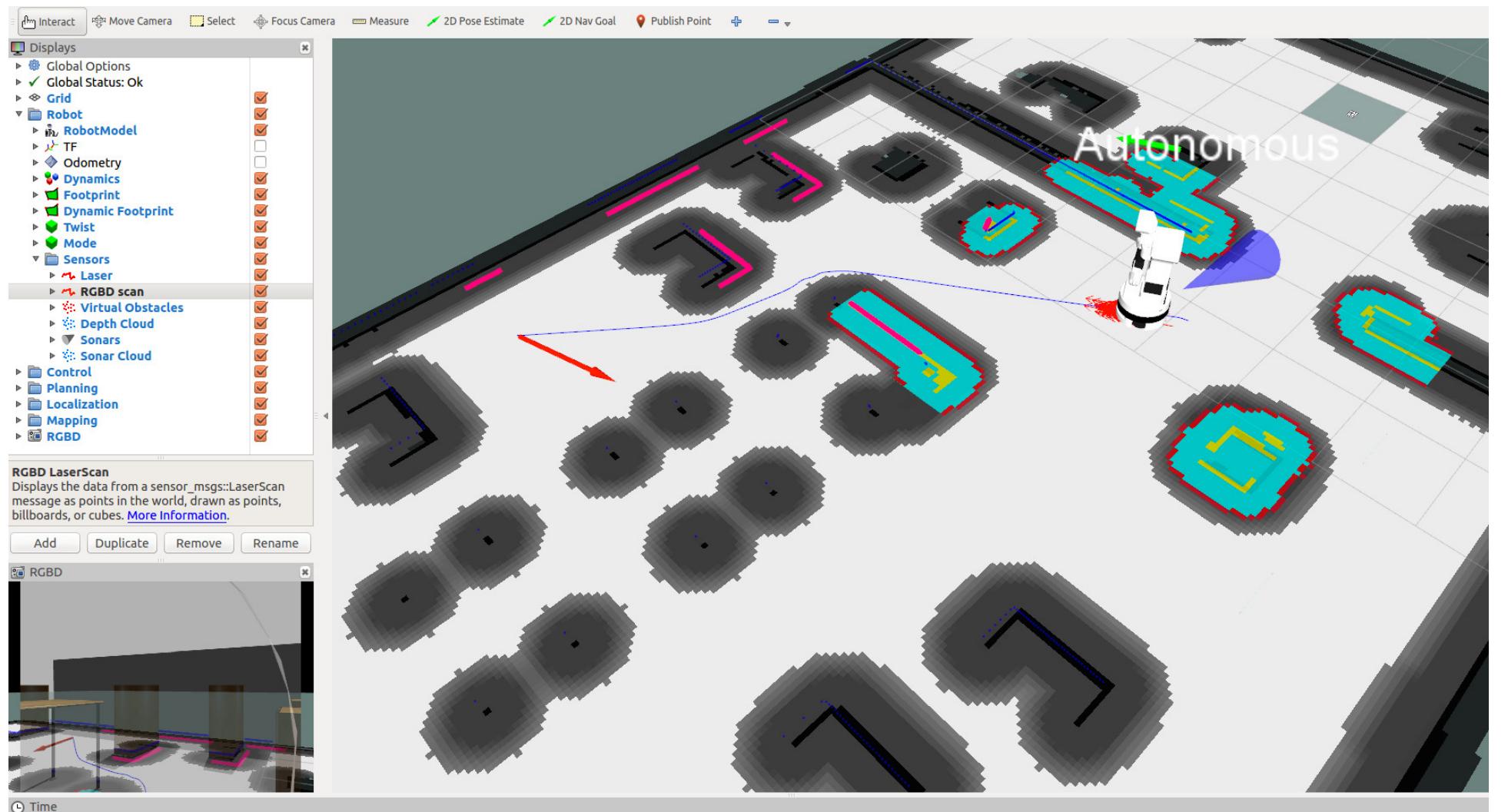


http://www.gavrila.net/Research/Chamfer_System/chamfer_basics2.gif

| | |
|-------------------|-------------------|
| 0 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 0 |
| 0 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 0 |
| 0 0 0 1 1 1 0 0 0 | 0 0 0 1 1 1 0 0 0 |
| 0 0 1 1 1 1 1 0 0 | 0 0 1 2 4 2 1 0 0 |
| 0 0 1 1 1 1 1 0 0 | 0 0 1 4 8 4 1 0 0 |
| 0 0 1 1 1 1 1 0 0 | 0 0 1 2 4 2 1 0 0 |
| 0 0 0 1 1 1 0 0 0 | 0 0 0 1 1 1 0 0 0 |
| 0 0 0 0 0 0 0 1 0 | 0 0 0 0 0 0 0 1 0 |
| 0 0 0 0 0 0 1 0 0 | 0 0 0 0 0 0 1 0 0 |

Nasonov and Krylov 2010
(zero indicates obstacle)

Michigan EECS 398/567 ROB 510 - autorob.org

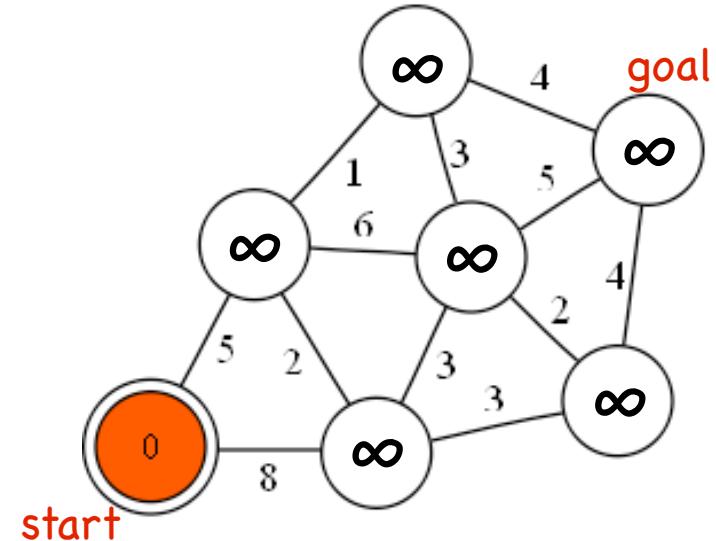


<https://blog.pal-robotics.com/blog/tiago-ros-simulation-tutorial-2-autonomous-robot-navigation/>

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  $\leftarrow \text{start\_node}$ 
```

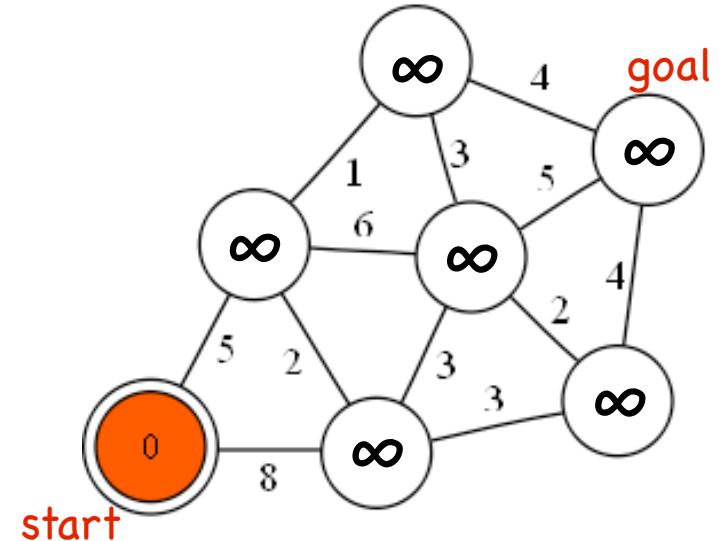
```
while visit_list != empty && current_node != goal
    cur_node  $\leftarrow \text{highestPriority}(\text{visit\_list})$ 
    visitedcur_node  $\leftarrow \text{true}$ 
    for each nbr in not_visited(adjacent(cur_node))
        add(nbr to visit_list)
        if distnbr > distcur_node + distance(nbr,cur_node)
            parentnbr  $\leftarrow \text{current\_node}$ 
            distnbr  $\leftarrow \text{dist}_{\text{cur\_node}} + \text{distance}(\text{nbr}, \text{cur\_node})$ 
        end if
    end for loop
end while loop
output  $\leftarrow \text{parent, distance}$ 
```



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 != empty && current_node != goal
    cur_node  $\leftarrow \text{highestPriority}(\text{visit\_list})$ 
    visitedcur_node  $\leftarrow \text{true}$ 
    for each nbr in not_visited(adjacent(cur_node))
        add(nbr to visit_list)
        if costnbr > costcur_node + cost(nbr)
            parentnbr  $\leftarrow \text{current\_node}$ 
            costnbr  $\leftarrow \text{cost}_{\text{cur\_node}} + \text{cost}(\text{nbr})$ 
        end if
    end for loop
end while loop
output  $\leftarrow \text{parent, distance}$ 
```



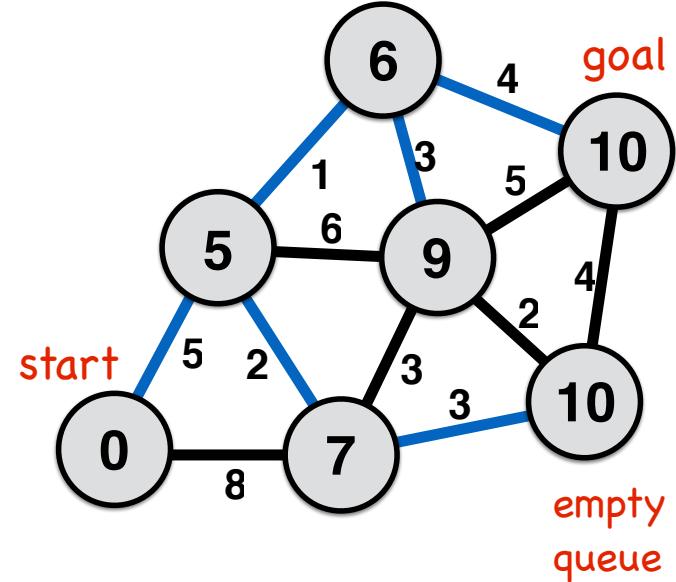
A-star shortest path algorithm

```
all nodes  $\leftarrow \{cost_{start} \leftarrow \text{infinity}, parent_{start} \leftarrow \text{none}, visited_{start} \leftarrow \text{false}\}$ 
start_node  $\leftarrow \{cost_{start} \leftarrow 0, parent_{start} \leftarrow \text{none}, visited_{start} \leftarrow \text{true}\}$ 
visit_queue  $\leftarrow \text{start\_node}$ 

while (visit_queue != empty) && current_node != goal
    dequeue: cur_node  $\leftarrow f\_score(\text{visit\_queue})$ 
    visitedcur_node  $\leftarrow \text{true}$ 
    for each nor in not_visited(adjacent(cur_node))
        enqueue: nbr to visit_queue
        if costnbr > costcur_node + cost(nbr)
            parentnbr  $\leftarrow \text{current\_node}$ 
            costnbr  $\leftarrow cost_{cur\_node} + cost(nbr)$ 
            f_score  $\leftarrow cost_{nbr} + line\_distance_{nbr,goal}$ 
        end if
    end for loop
end while loop
output  $\leftarrow \text{parent, distance}$ 
```

g_score: cost from start

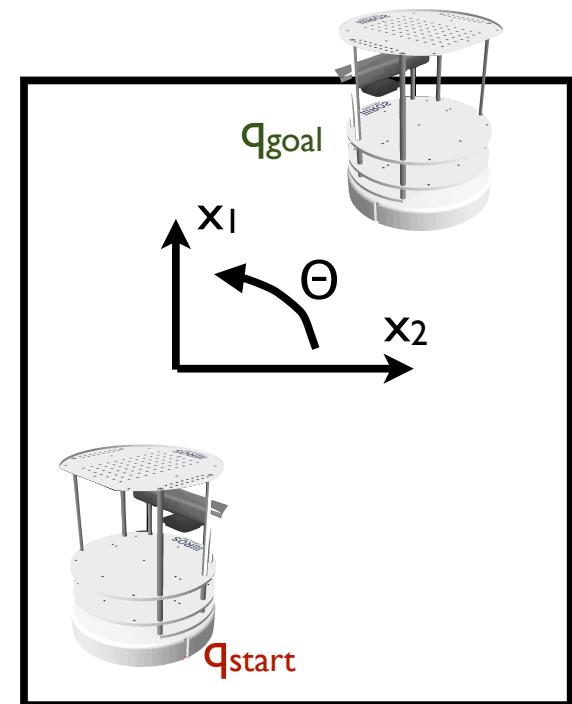
h_score: optimistic cost to goal



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



The value of marketing



Original TurtleBot

(Discontinued)



TurtleBot 2 Family

TurtleBot 2



TurtleBot 2i



TurtleBot Euclid



TurtleBot 2e

TurtleBot 3 Family

Burger



Waffle

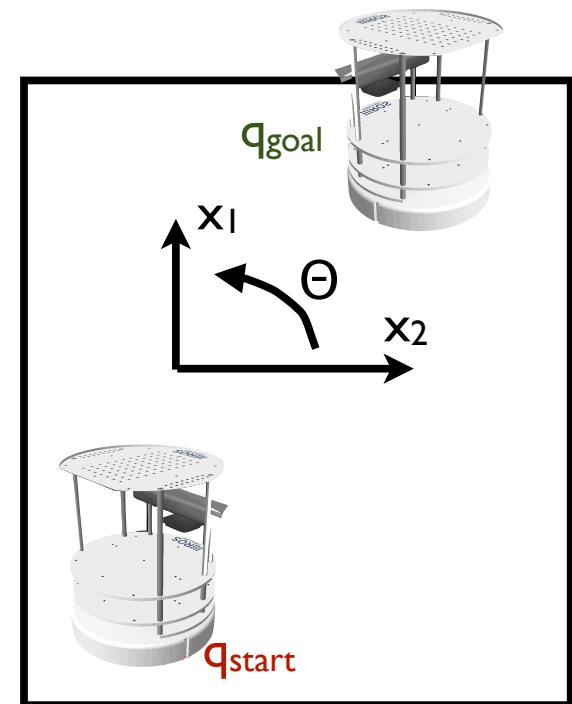


Holonomicity

- 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-lEjVsoiGo>

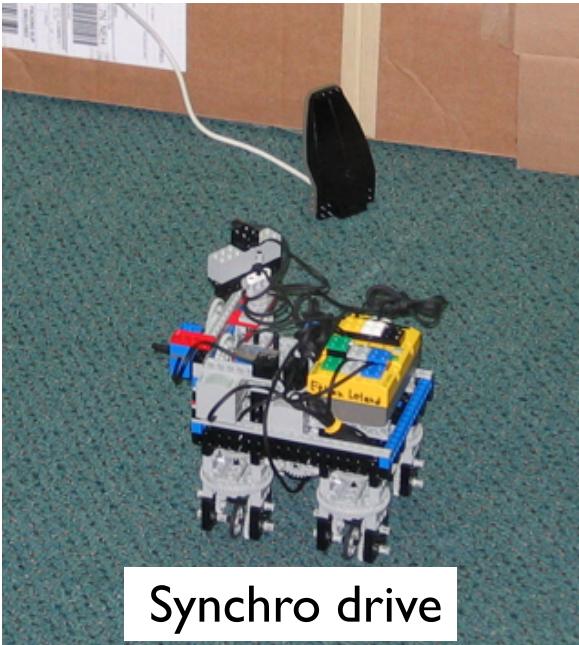


<https://www.youtube.com/watch?v=1ak17mdRg5I&t=75s>

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

Holonomic systems

Omni-wheel drive



Synchro drive

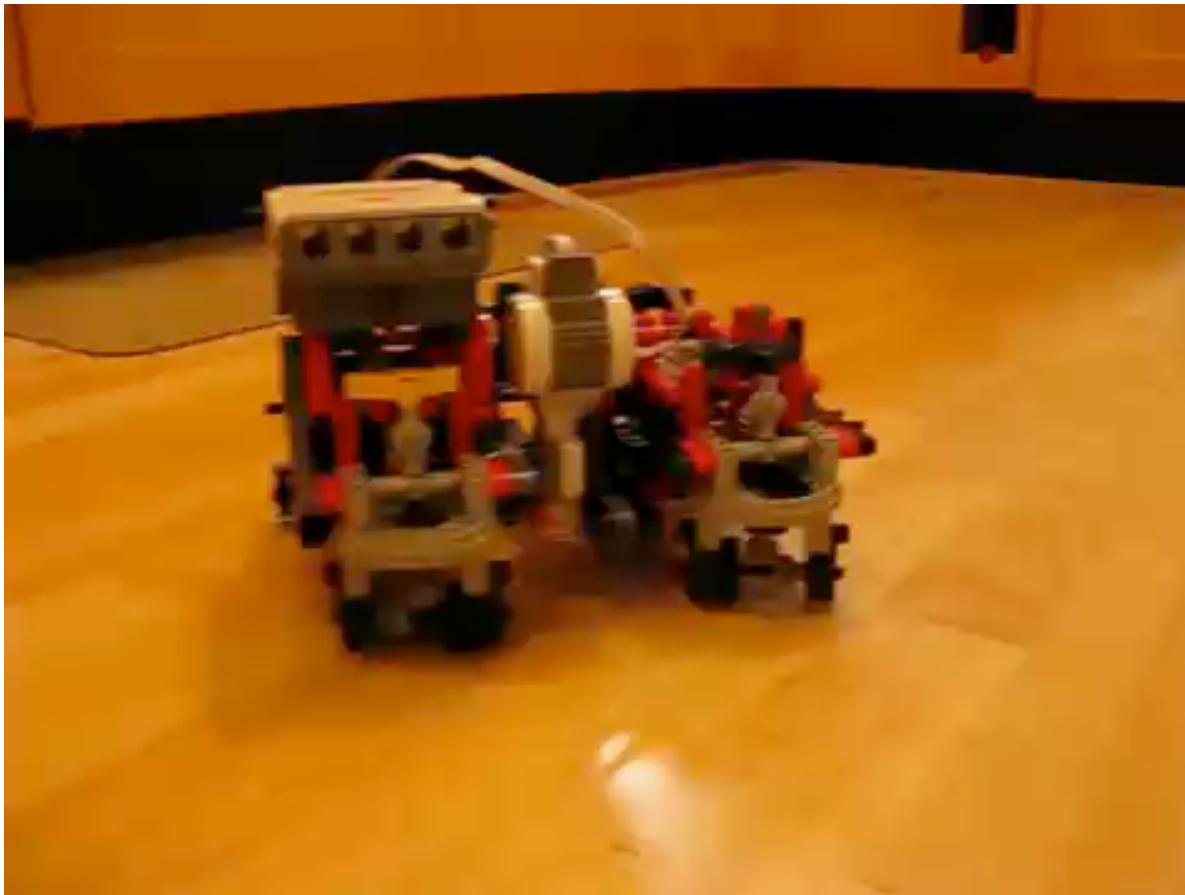


Killough platform



Mecanum wheels

Synchro Drive



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

Michigan EECS 398/567 ROB 510 - autorob.org

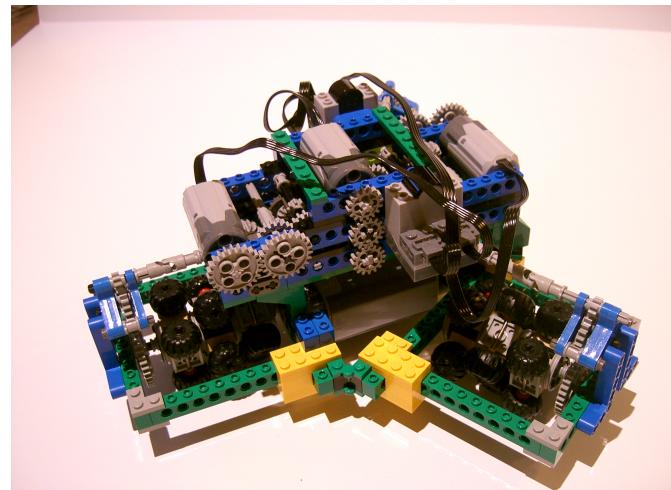
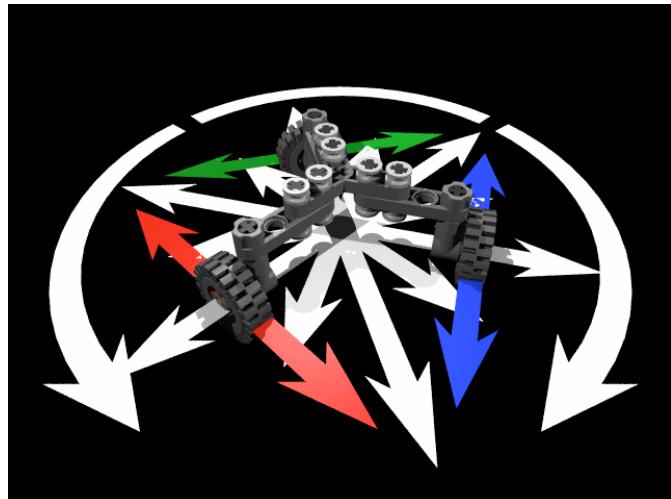
KUKA YouBot with Mecanum wheels



Me teleoperating KUKA YouBot from my house via a web browser, http://youtu.be/sWrRiy0AM_w

Michigan EECS 598/567 ROB 510 - autorob.org

Killough platform



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

monogram LEGO Technic autorob.org

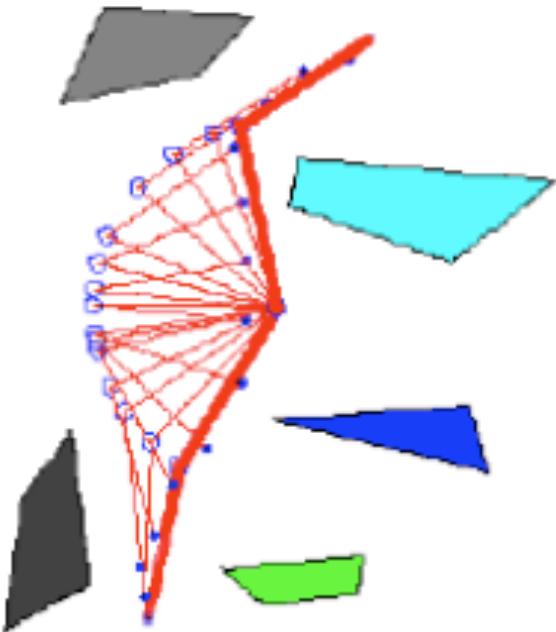
Recommended: D'Andrea on Omni-drive



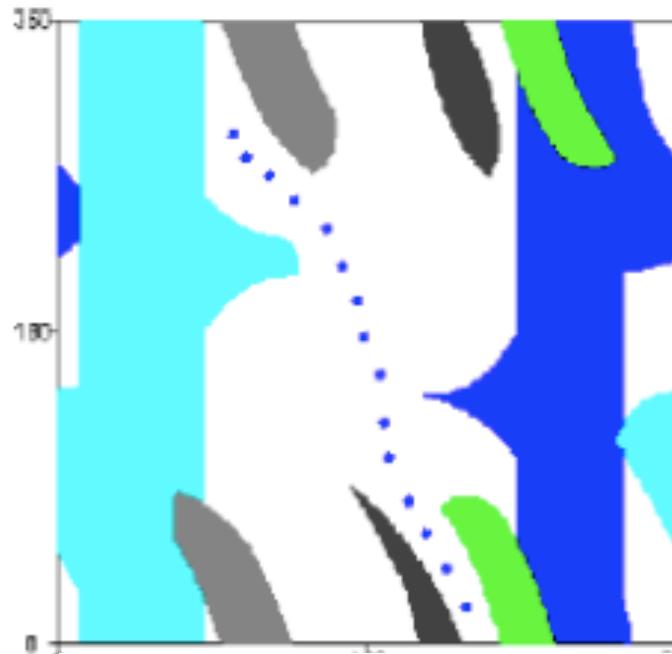
https://www.youtube.com/watch?v=p_WI-C-ORso

Michigan EECS 398/567 ROB 510 - autorob.org

How do we search arbitrary C-spaces?



Arm navigation in workspace

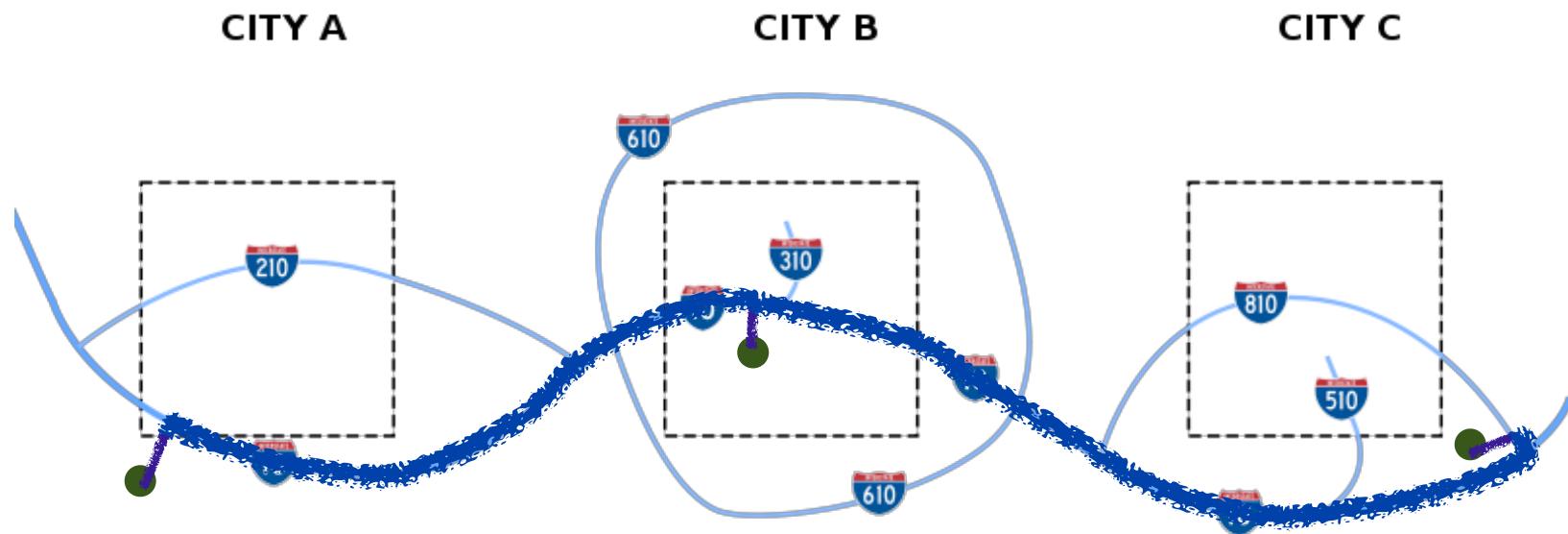


C-space representation

How build graphs in arbitrary C-spaces?

Next class:

Planning with Sampling Roadmaps



Building a path to goal