Introducing Motion Planning

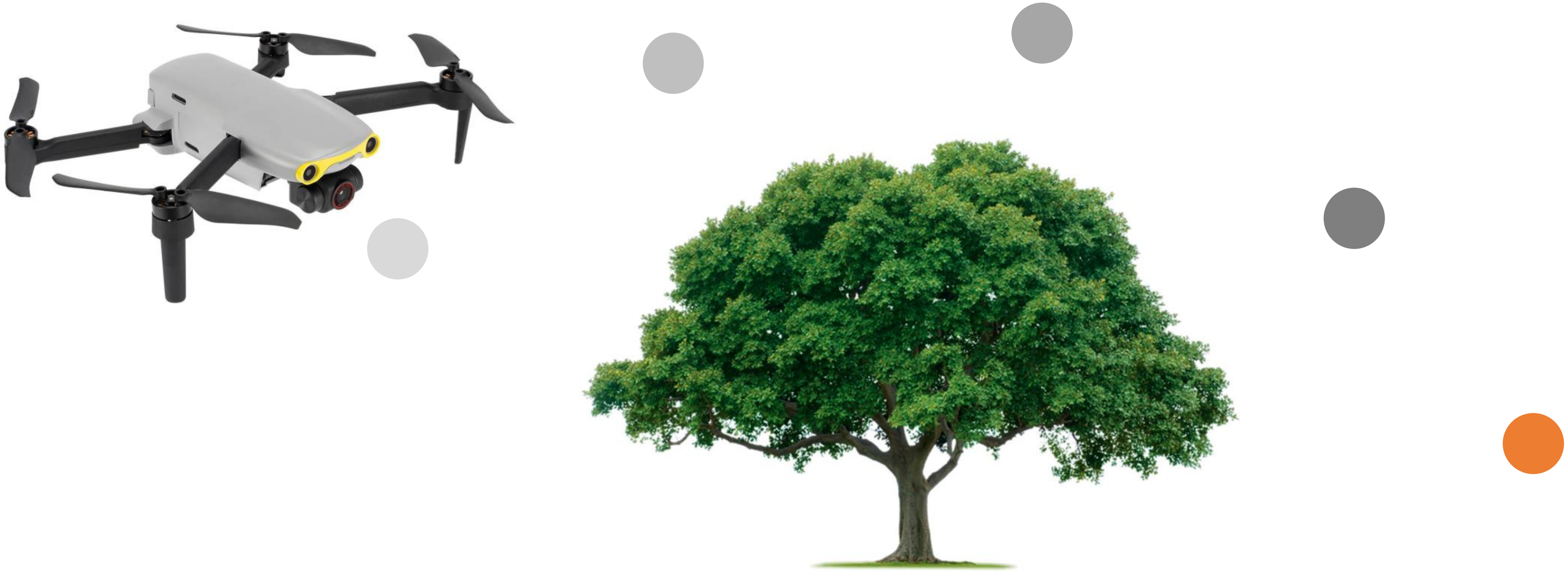
Reading: Modern Robotics 10.1 – 10.4, 10.6

Multi-Agent

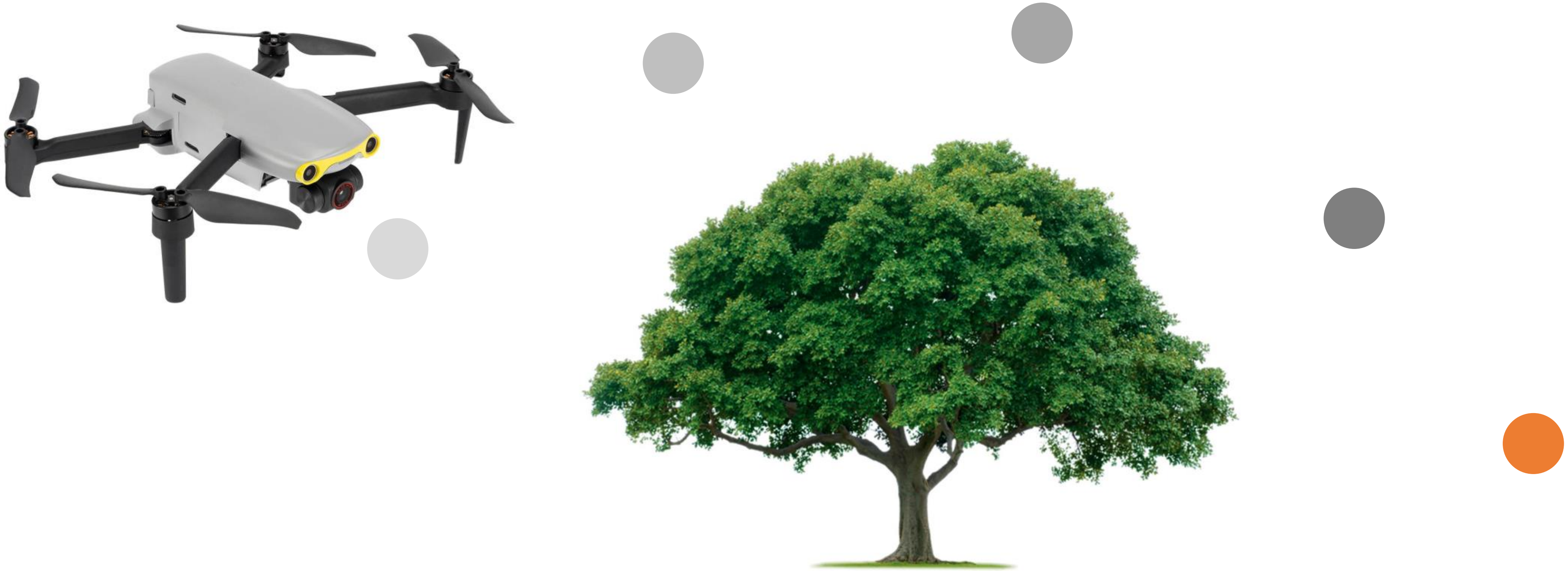
Hide and Seek

This Lecture

- What are some challenges for motion planning?
- Introducing basic motion planning algorithms



We use **motion planning** to find a continuous sequence of collision-free configurations from start to goal.



Motion planning is a **search problem**.
A motion planner is **complete** if the planner in finite time either finds a solution or correctly reports that there is none.

Challenges

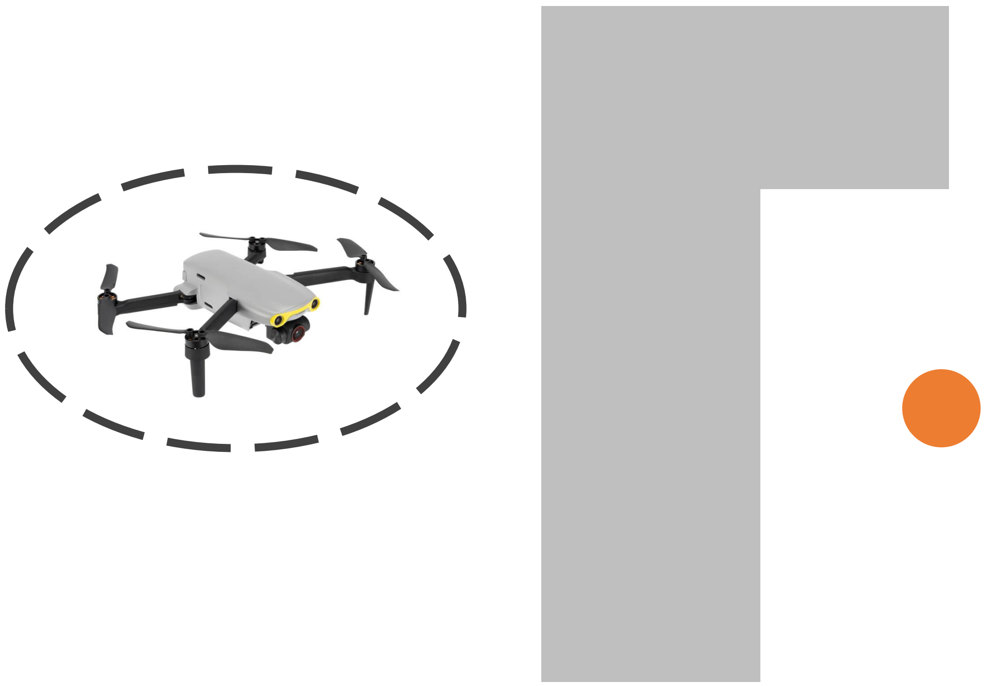
- How much of the environment can we accurately perceive with our sensors?
- How quickly can we replan when something changes?
- Does the robot have dynamic or kinematic constraints? (e.g., cars can’t move sideways)

Bug Algorithm

Program this robot to reach the goal

- Robot can sense nearby objects (i.e., knows when close to wall)
- Robot knows the direction of goal
- Robot does not know about the environment *a priori*

Your approach should work for an arbitrary number of obstacles, no matter what shape.

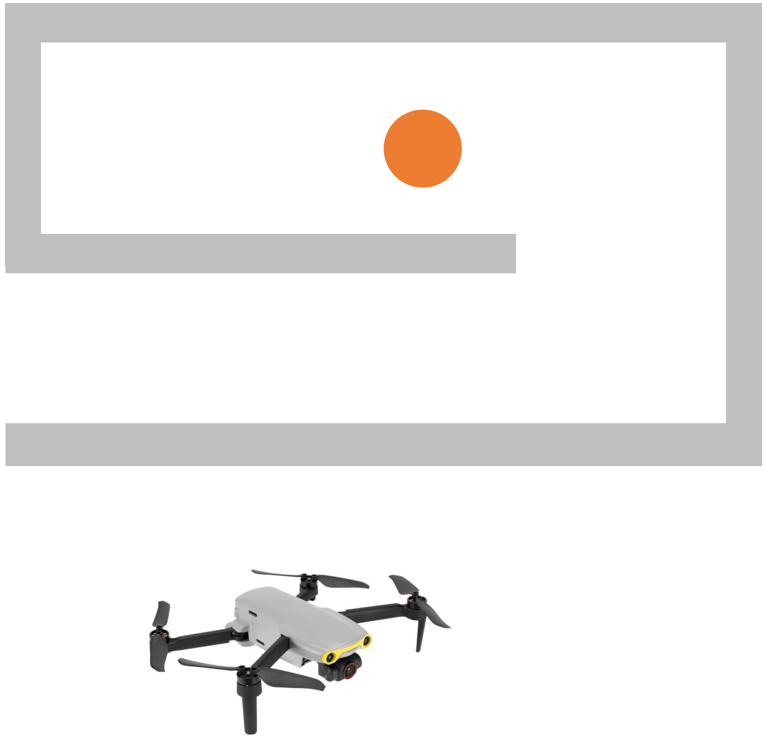


Bug Algorithm

Naïve Solution.

1. Head in straight line towards goal
2. If obstacle, turn left and follow until free to move towards goal again
3. Return to Step 1

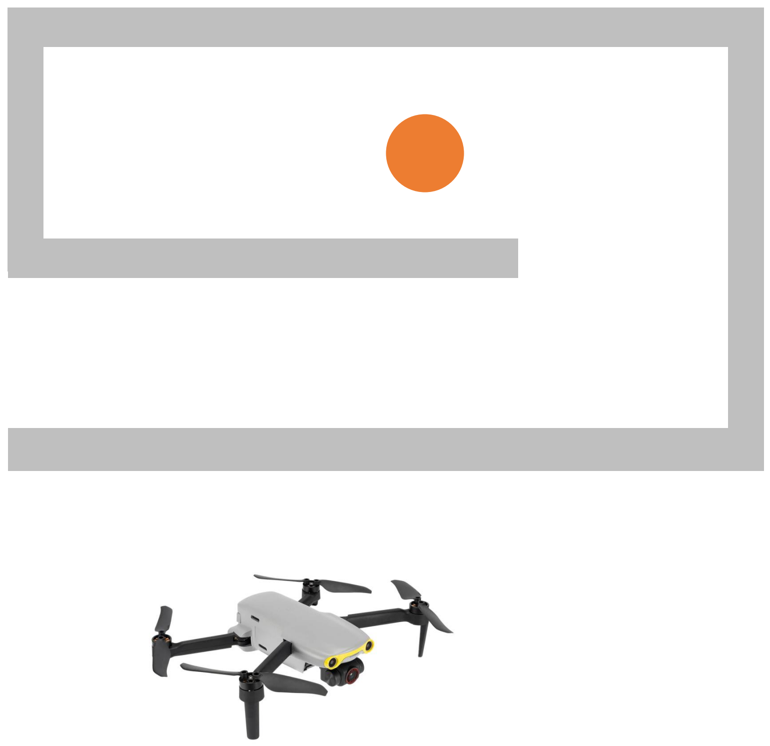
Is this motion planning algorithm complete?



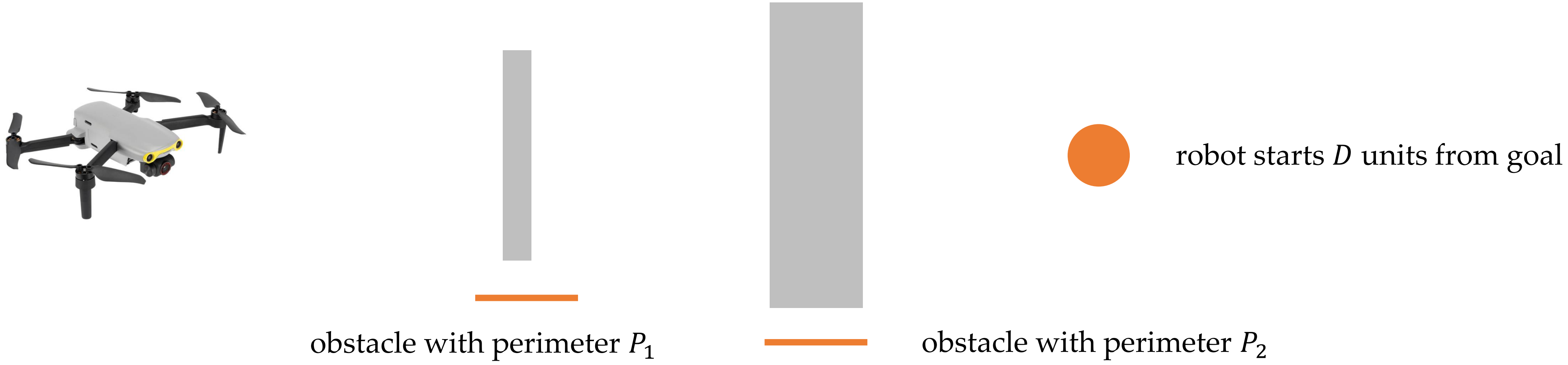
Bug Algorithm

Complete Solution.

1. Head in straight line towards goal
2. If obstacle, turn left and **circumnavigate the entire obstacle**
3. Return to point on obstacle closest to goal
4. Return to Step 1



Bug Algorithm

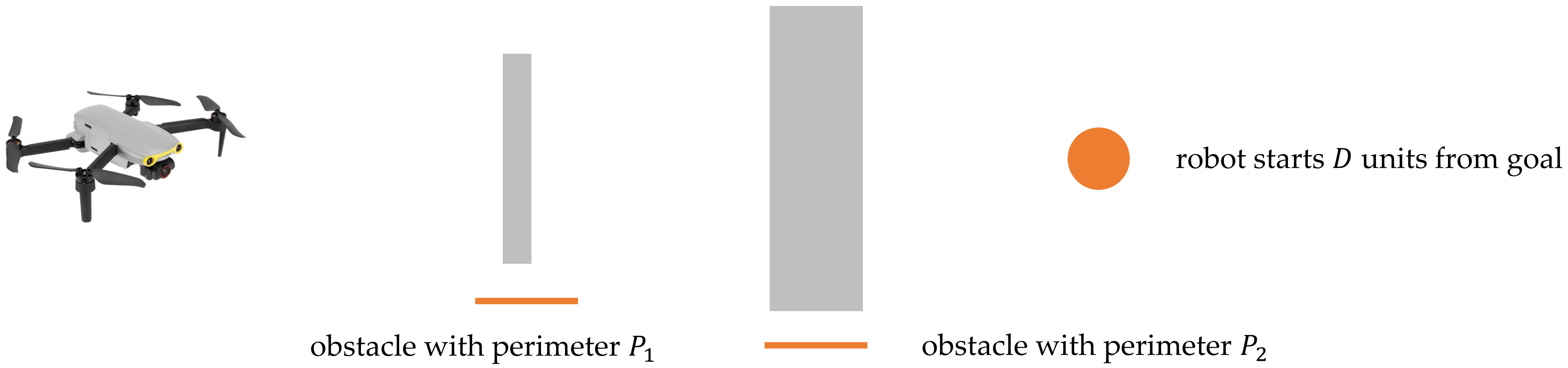


Consider an arbitrary environment with n obstacles…

What is the longest distance we might travel?

What is the shortest distance we might travel?

Bug Algorithm

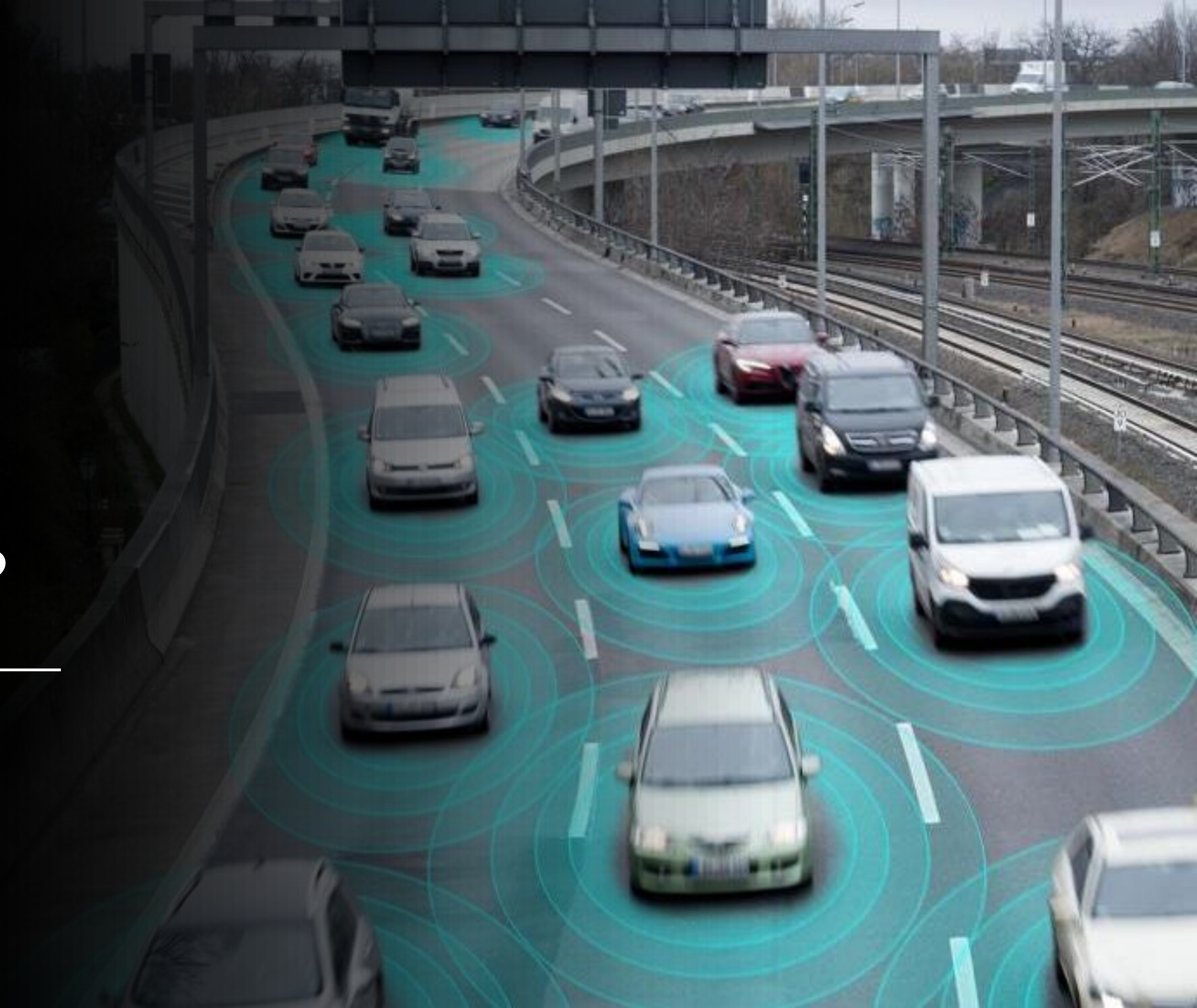


Upper bound. Longest possible distance:

$$D + \frac{3}{2}\sum_{i=1}^{n} P_i$$

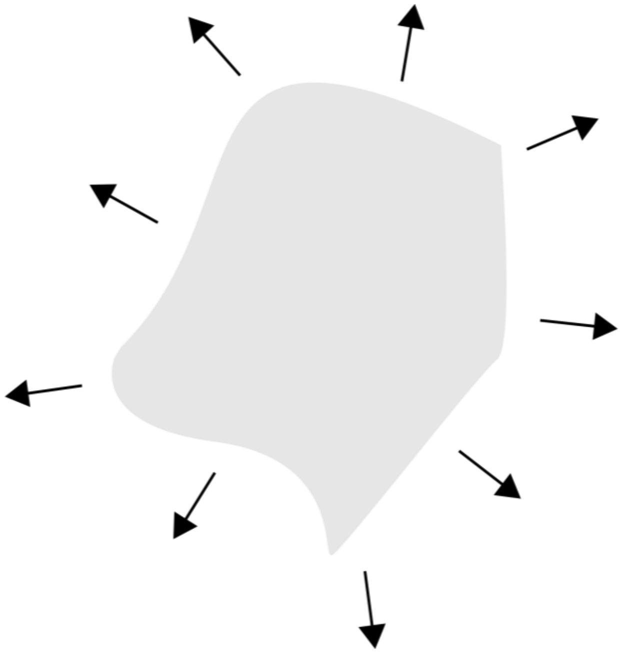
Lower bound.
Shortest possible distance is D

What if we **know** the **environment**?

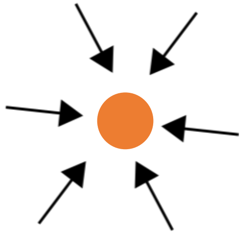


Potential Fields

Desired trajectory moves in direction of decreasing potential energy



Obstacles repel the robot (increasing potential energy)



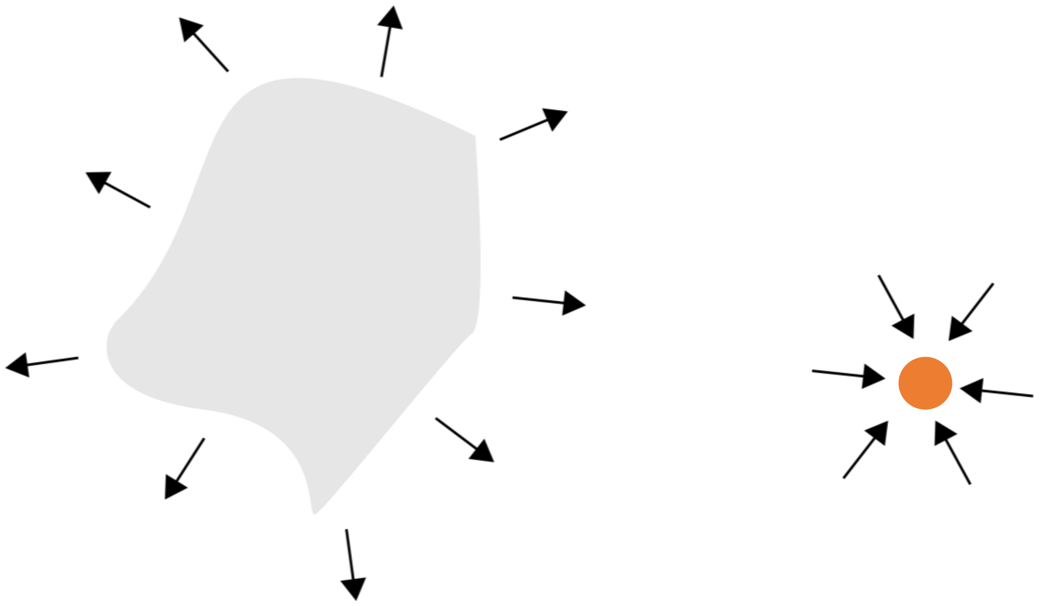
The goal attracts the robot (decreasing potential energy)

Potential Fields

Potential energy of environment:

$$U(\theta) = U_{att}(\theta) + U_{rep}(\theta)$$

- This potential energy is artificial
- We choose $U_{att}(\theta)$ and $U_{rep}(\theta)$
- Pick terms that are continuously differentiable (smooth)



Potential Fields

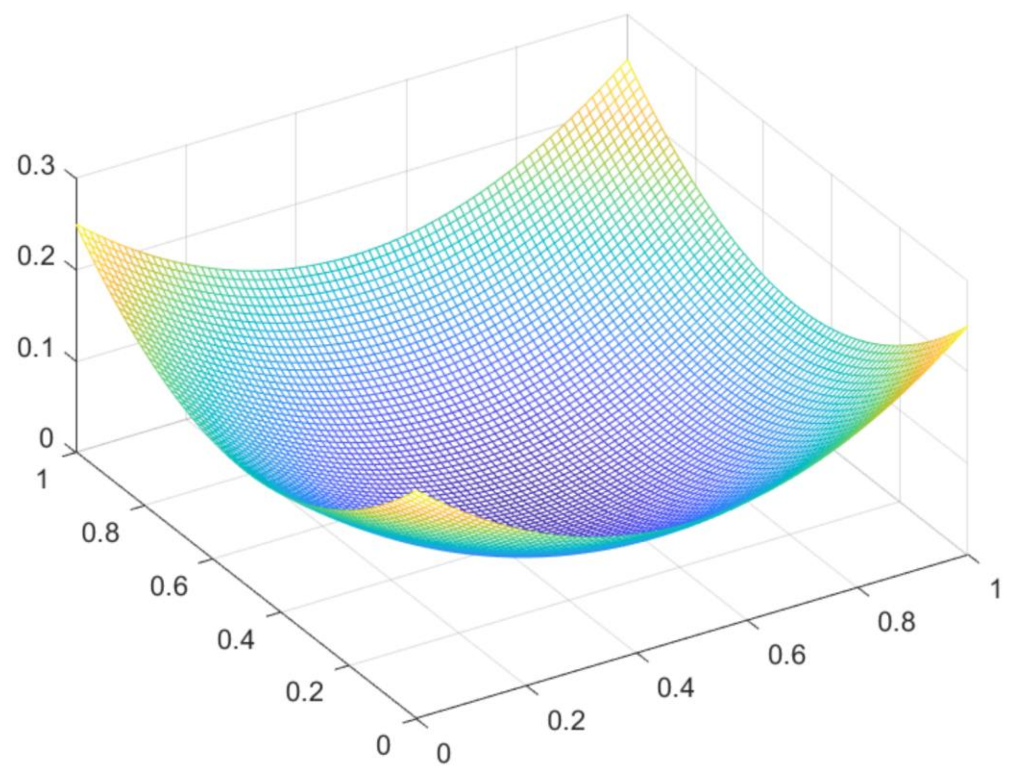
Potential energy of environment:

$$U(\theta) = U_{att}(\theta) + U_{rep}(\theta)$$

Attractive potential.

$$U_{att}(\theta) = \frac{1}{2}\beta\|g - \theta\|^2$$

g is goal position, $\beta > 0$ is parameter



Potential Fields

Potential energy of environment:

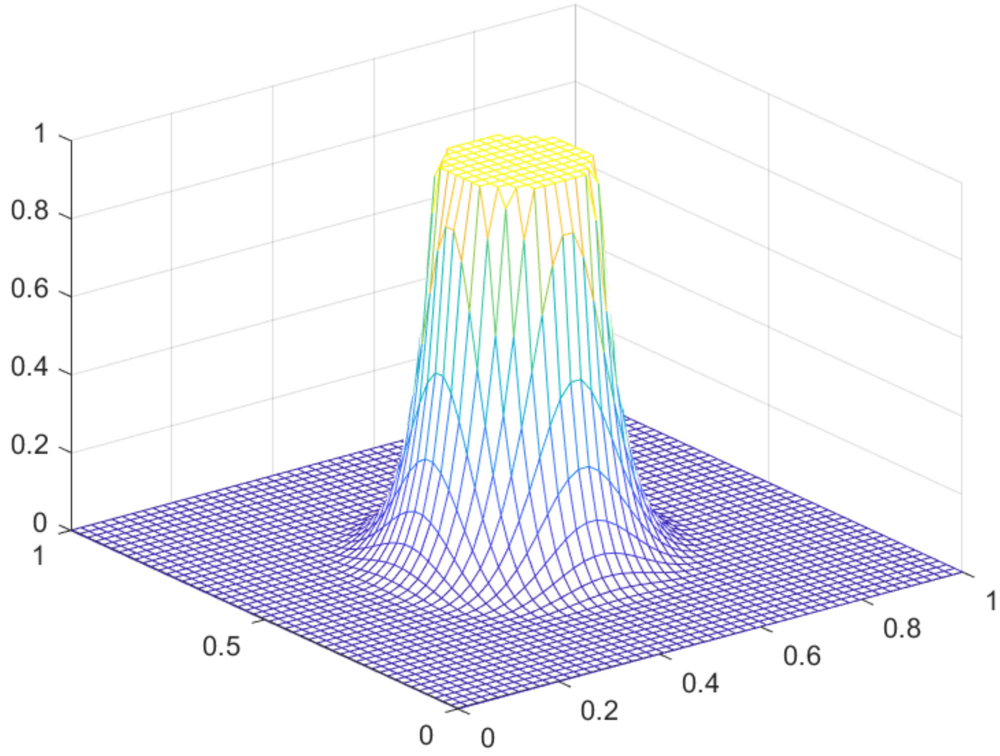
$$U(\theta) = U_{att}(\theta) + U_{rep}(\theta)$$

Repulsive potential.

$U_{rep}(\theta) = 0 \text{ if } \|c - \theta\| > r$

$U_{rep}(\theta) = \frac{1}{2}\gamma\left(\frac{1}{\|c-\theta\|} - \frac{1}{r}\right)^2 \text{ if } \|c - \theta\| \leq r$

c is center of obstacle, r is radius of obstacle,
$\gamma > 0$ is a parameter



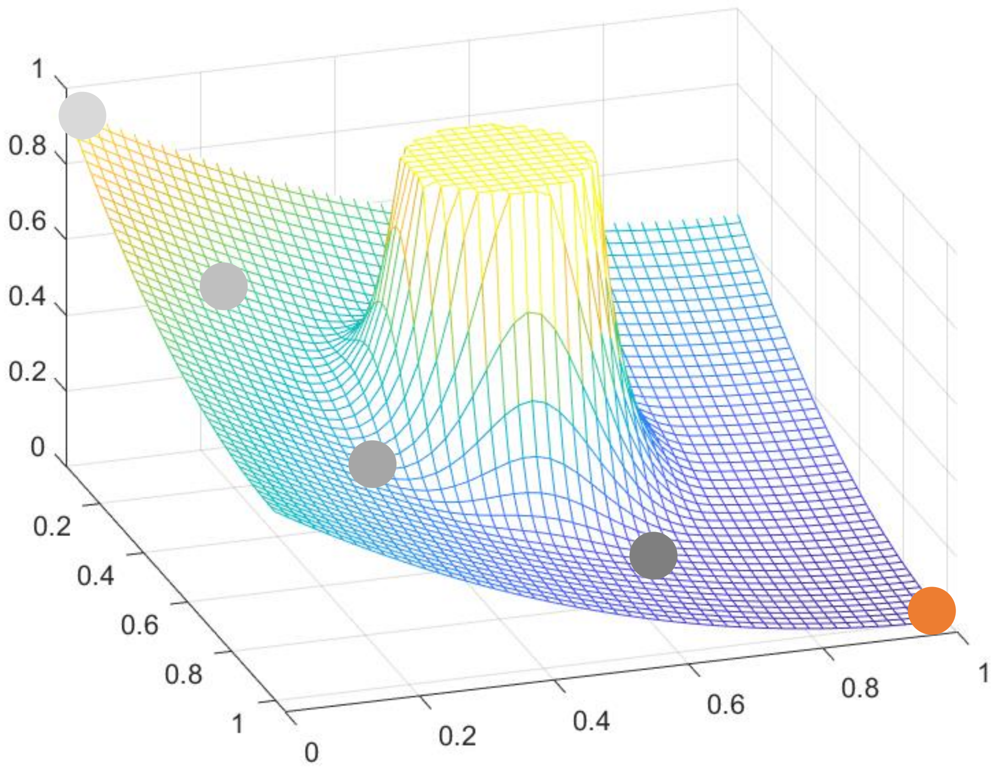
Potential Fields

Potential energy of environment:

$$U(\theta) = U_{att}(\theta) + U_{rep}(\theta)$$

Motion plan.

Move in direction of decreasing potential energy (**negative gradient**).



Gradient Descent

$\theta(0) \leftarrow \theta_{start}$

$t \leftarrow 0$

while $\|\nabla U(\theta(t))\| > \varepsilon$

$\quad \theta(t+1) \leftarrow \theta(t) - \underline{\alpha \nabla U}(\theta(t))$

$\quad t \leftarrow t+1$

$\alpha > 0$ is the learning rate

Output: a sequence of points $\{\theta(0), \theta(1), \dots, \theta(t)\}$

Gradient Descent

$\theta(0) \leftarrow \theta_{start}$

$t \leftarrow 0$

while $\|\nabla U(\theta(t))\| > \varepsilon$

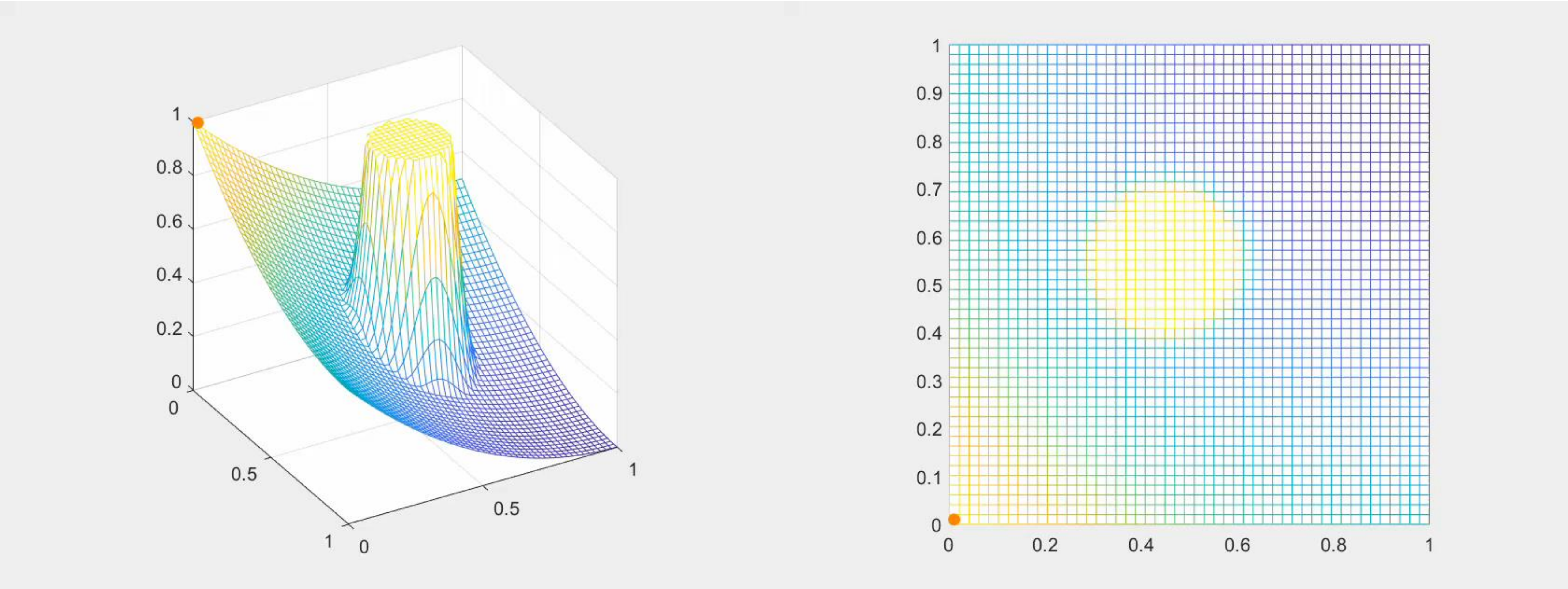
  $\theta(t+1) \leftarrow \theta(t) - \alpha \nabla U(\theta(t))$

  $t \leftarrow t + 1$

Tip. Often easier to compute gradient using numerical differentiation:

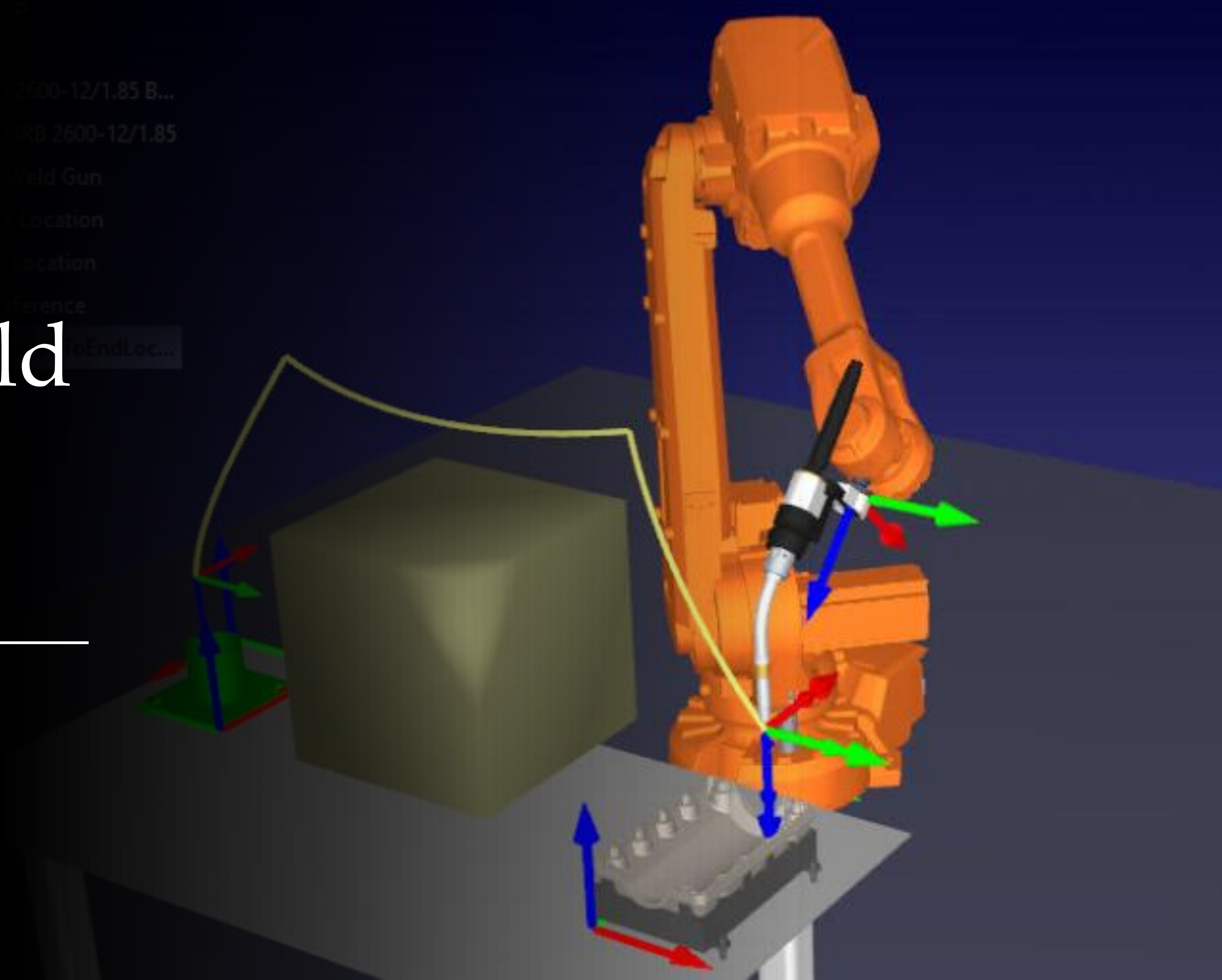
$$\nabla U = \begin{bmatrix} \frac{\partial U(\theta)}{\partial \theta_1} \\ \vdots \\ \frac{\partial U(\theta)}{\partial \theta_n} \end{bmatrix}, \qquad \frac{\partial U(\theta)}{\partial \theta_i} \approx \frac{U(\theta + \Delta\theta_i) - U(\theta)}{\Delta\theta_i}$$

Output: a sequence of points $\{\theta(0), \theta(1), \dots, \theta(t)\}$



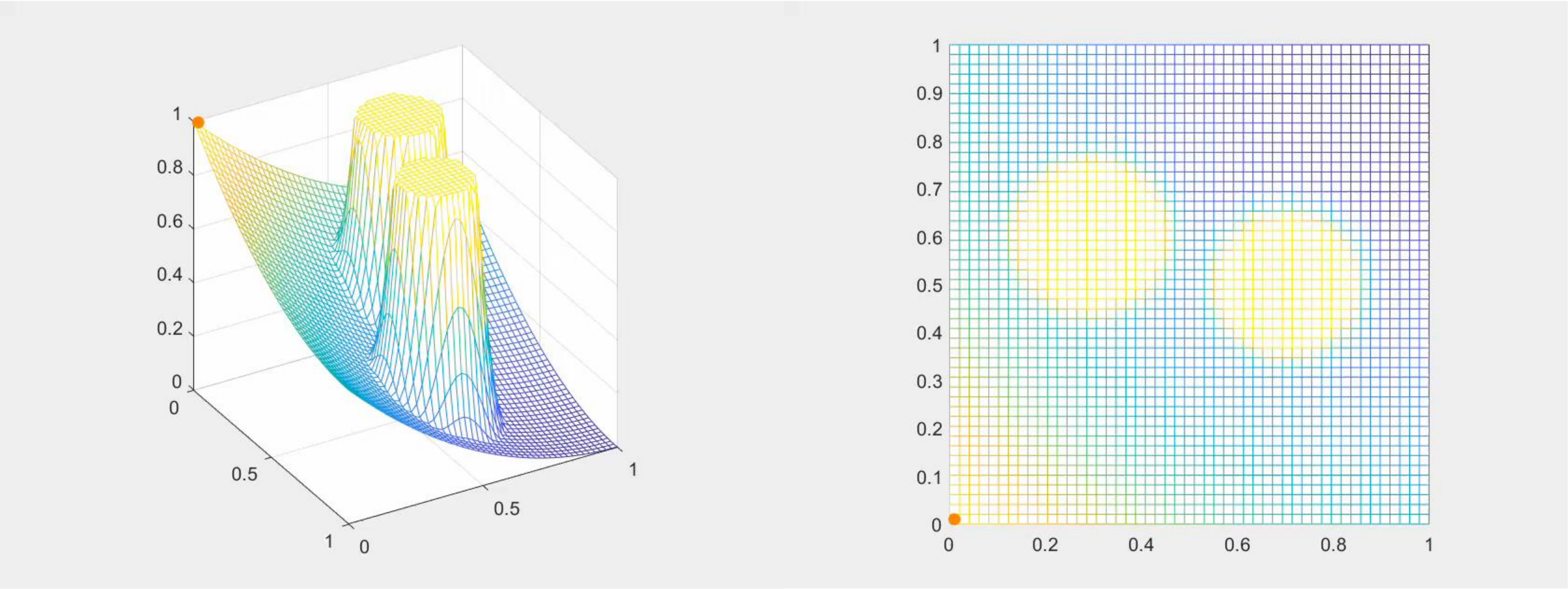
1
0.8
0.6
0.4
0.2
0
0
0.5
1
0
0.5
1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1

Is the potential field planner **complete**?



Gradient Descent

Gradient descent will converge to a **local minimum** of potential energy,
but is not guaranteed to converge to the global minimum.



1
0.8
0.6
0.4
0.2
0
0
0.5
1
0
0.5
1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1

This Lecture

- What are some challenges for motion planning?
- Introducing basic motion planning algorithms

Next Lecture

- How do ensure the motion planner is complete?
- What do state-of-the-art motion planning algorithms look like?