Skoltech

Skolkovo Institute of Science and Technology

L03: Configuration Space

Planning Algorithms in AI

Prof. Gonzalo Ferrer,

Skoltech, 7 November 2025

L03 Summary

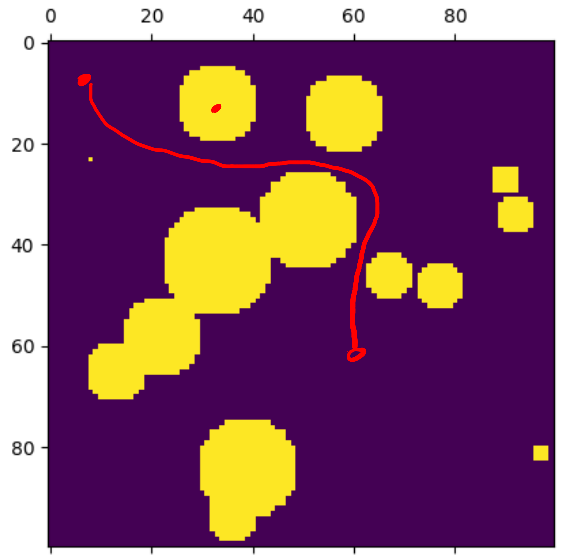
Material from LaValle 4.1-4.3, 5.1

- Workspace and Configuration space.
- Topology of the new spaces.
- Path connectivity.
- C-space of obstacles.
- Distances and Metric Space

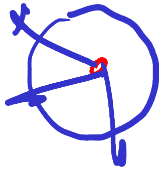
Workspace

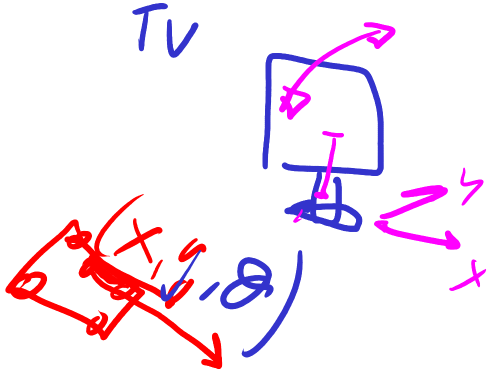
- Workspace corresponds to the physical world, usually R^2 for 2D or R^3 for 3D where the plan will be executed.
- Here, in this natural space, is where many of the problems we are defining need to be solved.
- It is natural to define obstacles or goals, and this is definitely the space where our intuition works well.
- We require this distinction since many planning problems live in the workspace but this is not the best way to solve them.

Example: 2D rectangle with some obstacles.



Configuration Space





- The *configuration space* (or C-space) corresponds to the degrees of freedom of the robot.
- It can be considered a special case of the *state space*.
- To solve the motion planning problem, one must solve it in the C-space.
- Now the problem is that some of these C-spaces, required to obtain a plan, can be more complex than the Euclidean space.
- We need to define proper tools to abstract the problem, regardless of how complicated the C-space is, in order to calculate plans (today's lecture).

Configuration Space. 2D point

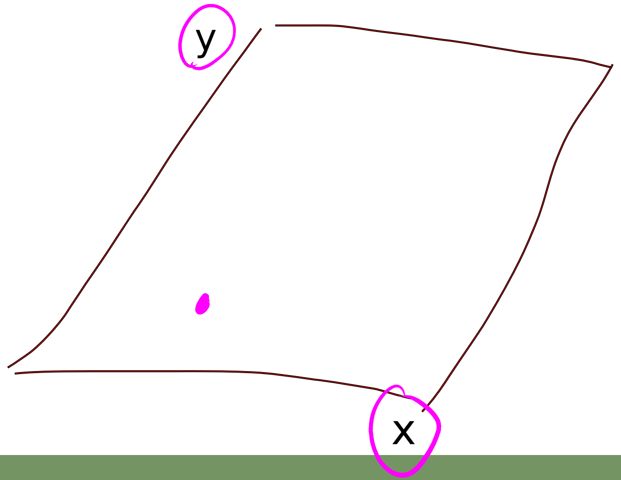
A 2D point is what we used on L02 to define the "escape the maze" planning problem.

We can formalize this configuration space as

$$q = [x, y]^\top \in \mathcal{C}. \quad \text{where } \mathcal{C} = \mathbb{R}^2$$

$\mathcal{C} \subset \mathcal{W}$

In fact, this configuration space is pretty simple to analyze since it is exactly equivalent to the workspace and where things are intuitive for us.



Configuration Space. The Revolute Joint (1R)

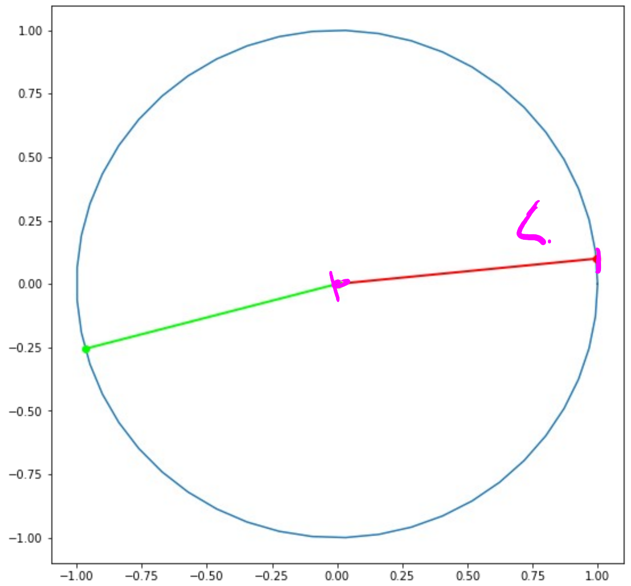
This space is also known as an "angle".

$$\alpha \in (-\pi, \pi] \subset \mathbb{R} \qquad\qquad \alpha = \alpha + k2\pi, \quad k \in \mathbb{Z}$$

There is an ambiguity in this representation, that is why the most natural representation for angles is the 1-spherical group:

$$\mathbb{S} \text{ is the 1-spherical group} \qquad \mathbb{S}^1 = \{(x, y) \in \mathbb{R}^2 \,|\, x^2 + y^2 = 1\}$$

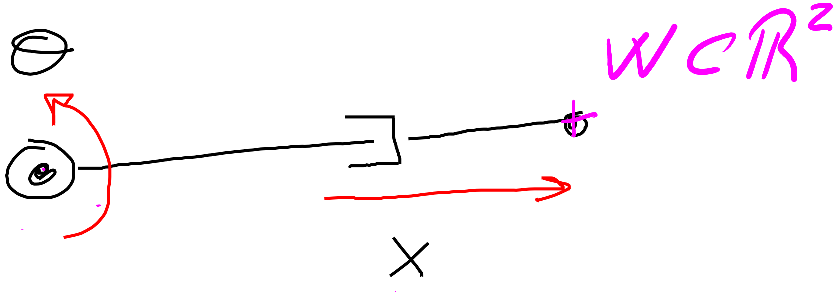
A **manifold** is a higher dimensional space can be represented locally by the Euclidean space.
In the case of angles, it is a 1-manifold.



Configuration Space. The RP manipulator

The RP manipulator consist of a *revolute* and a *prismatic* articulation.

Also known as a *kinematic chain*. It can be viewed as polar coordinates.

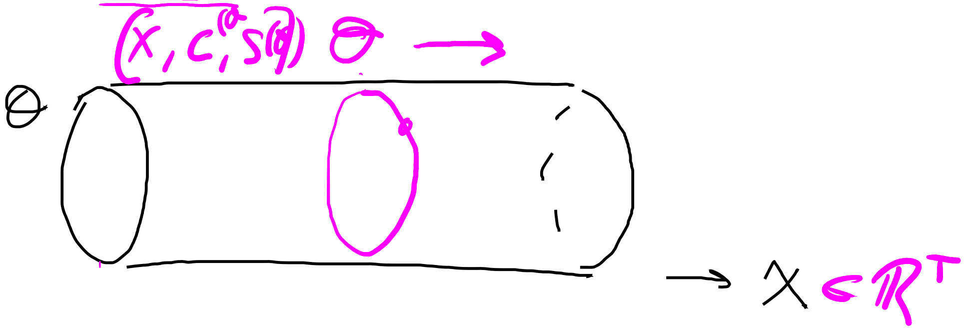


$\mathbb{S}$ is the 1-spherical group

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

More formally, the configuration space is the Cartesian product of 2 spaces:

$$q = [x, \theta]^\top \in \mathcal{C}, \quad \text{where } \mathcal{C} = \mathbb{R}^+ \times \mathbb{S}$$

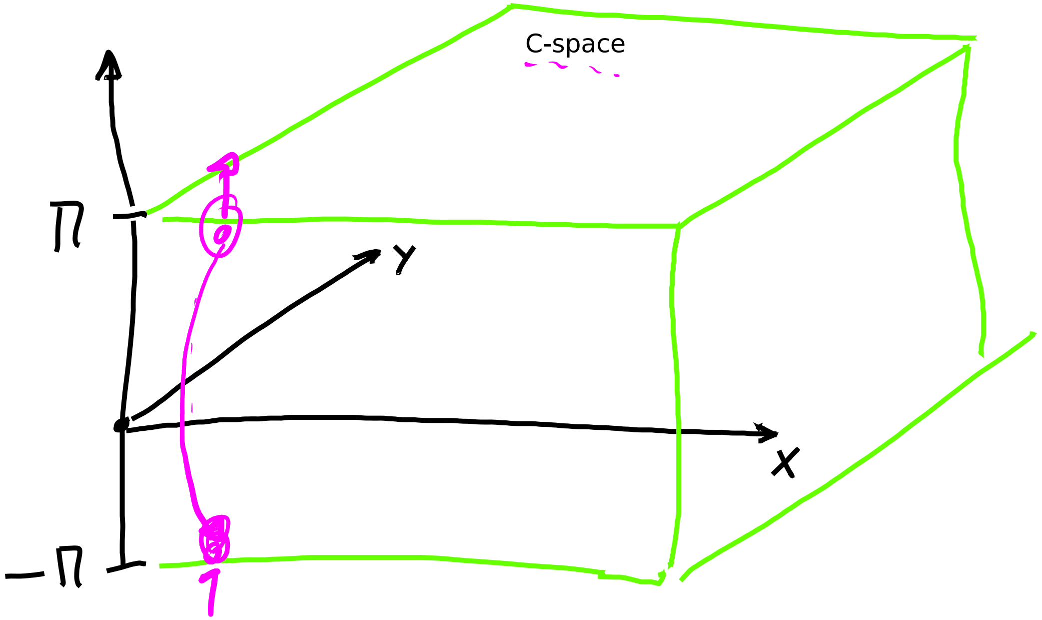


Configuration Space. 2D Pose

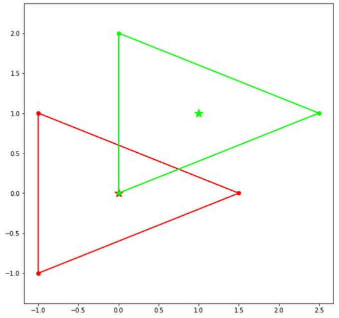
A robot as a rigid body consisting in position and orientation (2D pose)

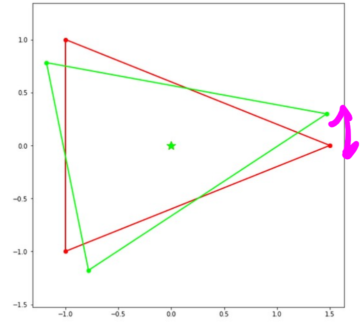
$$q = [x, y, \theta]^\top \in \mathcal{C}, \quad \text{where } \mathcal{C} = \mathbb{R}^2 \times \mathbb{S}$$

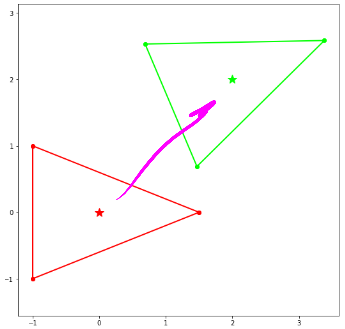
Let's take a look at the following drawing:



"How to connect q_i, q_j ?"







Configuration Space. 3D Pose

The robot is a 3D rigid body, consisting of a position and orientation:

$$q = [p, \theta]^\top \in \mathcal{C}, \quad \text{where } \mathcal{C} = \mathbb{R}^3 \times SO(3)$$

$$p = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

$$\in SO(2)$$

$$R(\theta) = \begin{pmatrix} c(\theta) & s(\theta) \\ -s(\theta) & c(\theta) \end{pmatrix}_{2\times 2}$$

Now the configuration space is a more involved space to operate.

Orientations in 3D are expressed by rotation matrices:

$$SO3 = \{R \in \mathbb{R}^{3\times 3} \,|\, RR^\top = I\,,\, det(R) = 1\}$$

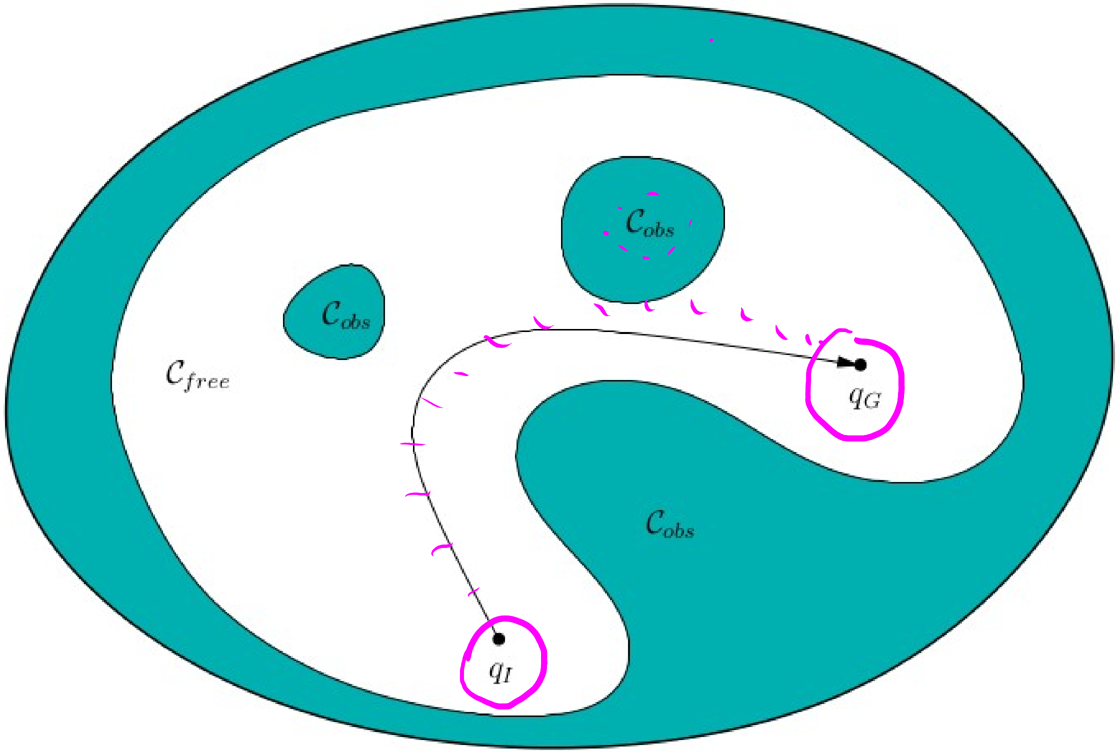
So how can be plan? We need two basic ingredients: **connectivity** and a well-defined **distance** function. (more later)

For more insights on 3D rotations and 3D poses, take a look at this material:

https://github.com/MobileRoboticsSkoltech/Perception-in-Robotics-course-T3-2021-Skoltech/blob/master/L13/RBT.pdf
https://www.youtube.com/watch?v=h-A8x10JKNI&list=PLRXYrdEUvBoBCjYdmuDqohvSTJIPPrBUT&index=13

Motivation for calculating the C-space

Once defined the new configuration space, we will build the equivalent to paths and obstacles in C and connect this to planning. And we will see that it will only require to consider points, rather than complicated objects in the workspace



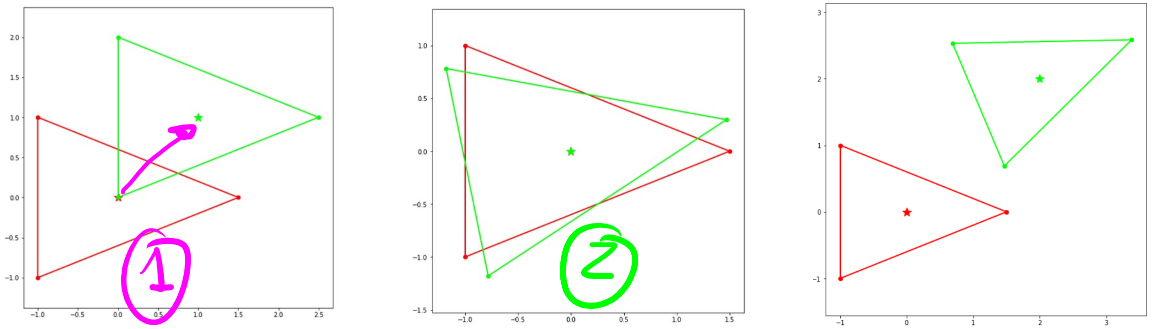
Path connectivity

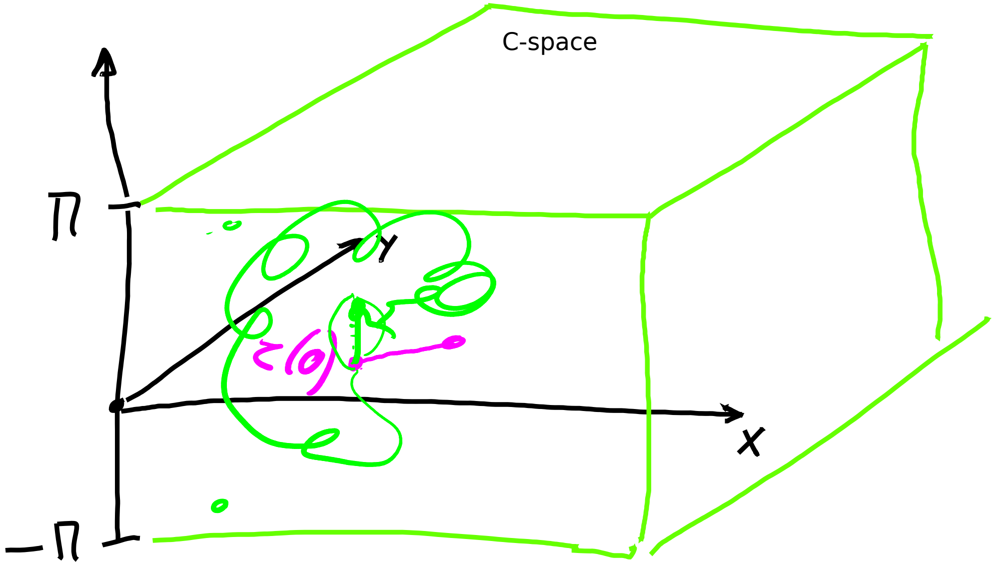
- A path is a continuous function such that all points belong to the space X:

$$\tau : [0, 1] \rightarrow X$$

- A space is said to be **path connected** if for any two states, there is a path connecting them:

$$\forall x_1, x_2 \in X, \quad \exists \tau \text{ such that } \tau(0) = x_1, \tau(1) = x_2$$





Distance and Metric Space (LaValle 5.1)

$$\rho : X \times X \to \mathbb{R}^{0+}, \quad \forall a, b, c \in X$$

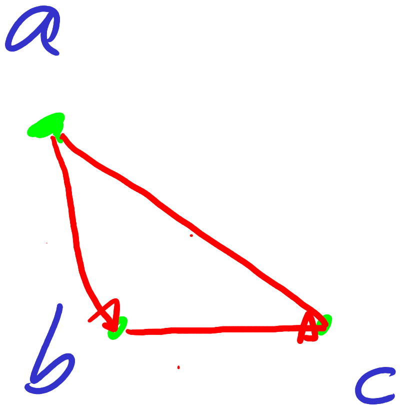
In order to match our intuition with the Euclidean distance in R^n, the function rho requires some axioms to be satisfied:

1) **Nonnegativity**: $\rho(a, b) \geq 0$

2) **Reflexivity**: $\rho(a, b) = 0 \iff a = b$

3) **Symmetry**: $\rho(a, b) = \rho(b, a)$

4) **Triangle inequality**: $\rho(a, b) + \rho(b, c) \geq \rho(a, c)$



Then, the *metric space* (X, ρ) is well defined.

L_p metrics (LaValle 5.1)

- The most important family of metrics over R^n:

$$\rho(x, x') = \left(\sum_{i=1}^{n} |x_i - x'_i|^p \right)^{1/p}$$

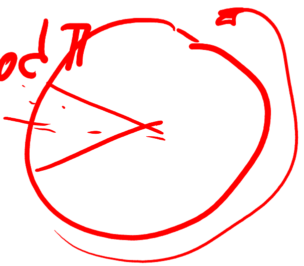
- L_2: The Euclidean metric.
- L_1: The Manhattan metric, a typical heuristic used for the A* in 2D

$|x_1 - x_1'| + |x_2 - x_2'|$

- L_{inf}: when *p* goes to infinity, then the distance is the maximum element.
- <u>Metric subspaces</u>: Any subspace *Y* of a metric space *(X,ρ)* becomes a metric space by restricting the domain of the distance to *Y*. This means, for any of the manifolds previously shown we can use L_p metric on R^n.

Cartesian product of spaces

$d(\alpha_1, \alpha_2) = L_1(\cdot) \bmod \pi$



The C-space is often constructed as the Cartesian product of spaces, so we want to obtain metrics for those spaces. The solution is very convenient:

Let (X, ρ_x) and (Y, ρ_y) be two metric spaces

Ex: $(x, y), \theta$

A new metric space (Z, ρ_z) where $Z = X \times Y$ $\rightarrow C$

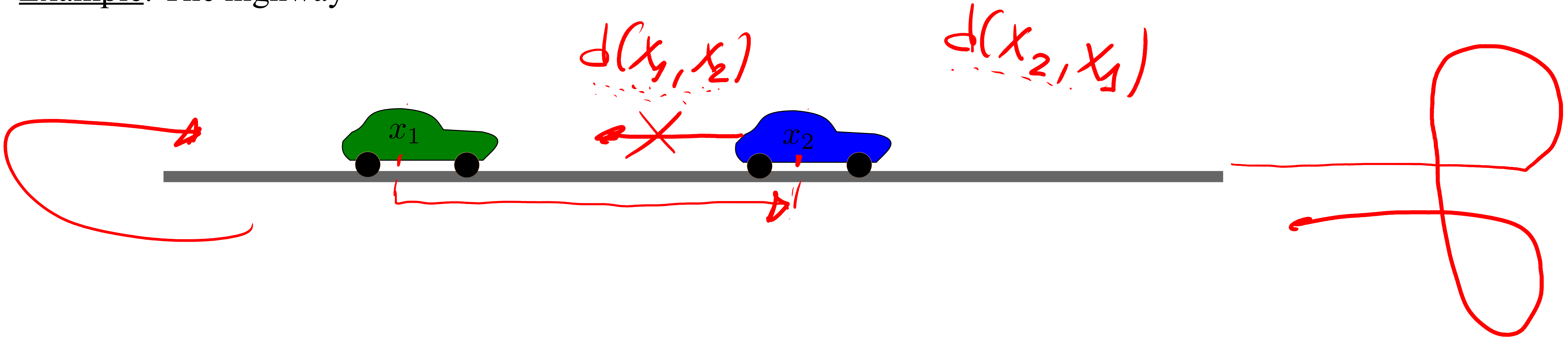
$$\boxed{\rho_z(z, z') = \rho_z(x, y, x', y') = c_1 \rho_x(x, x') + c_2 \rho_y(y, y')} \longrightarrow \rho_{L_2}(\ ) + c\,\rho_\theta(\cdot)$$

The weights must be positive, still there are infinite options...

Pseudo-Distances

- If some properties are not found, still we can use pseudo-distances for planning purposes, but the results might be non-desirable.

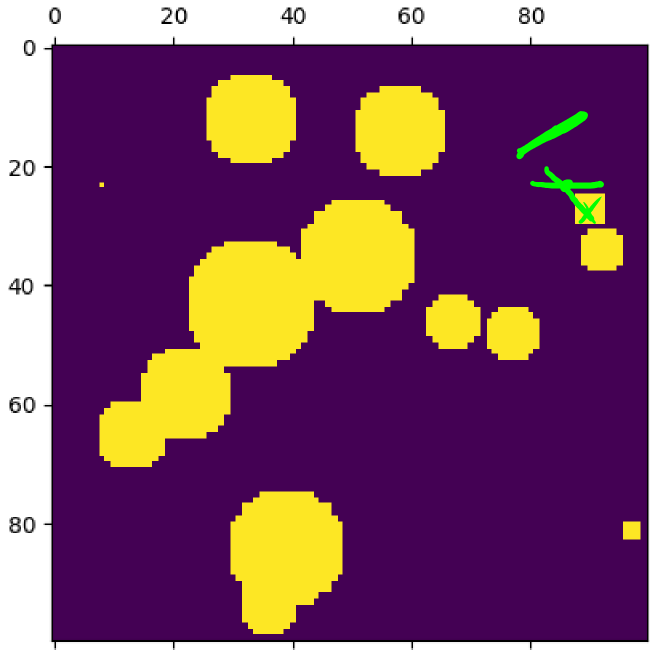
  Example: The highway



Obstacles in the C-space

- We have discussed about the C-space as if it was empty. What happens if there is an obstacle, which belongs to the Workspace?
- How does this interact with the C-space defined for the robot??

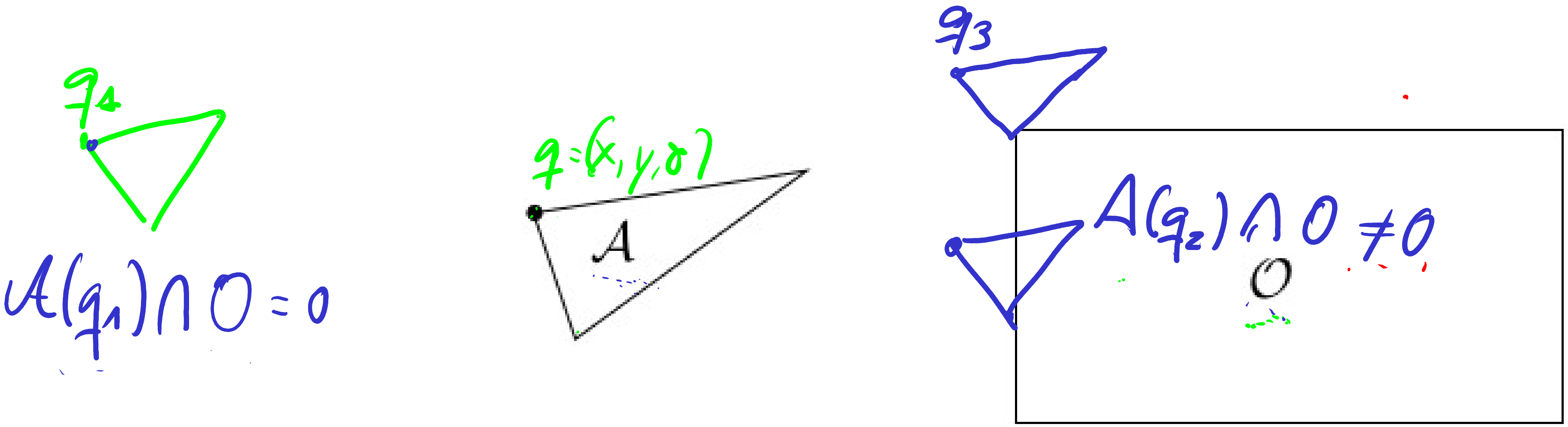
W



$$q = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

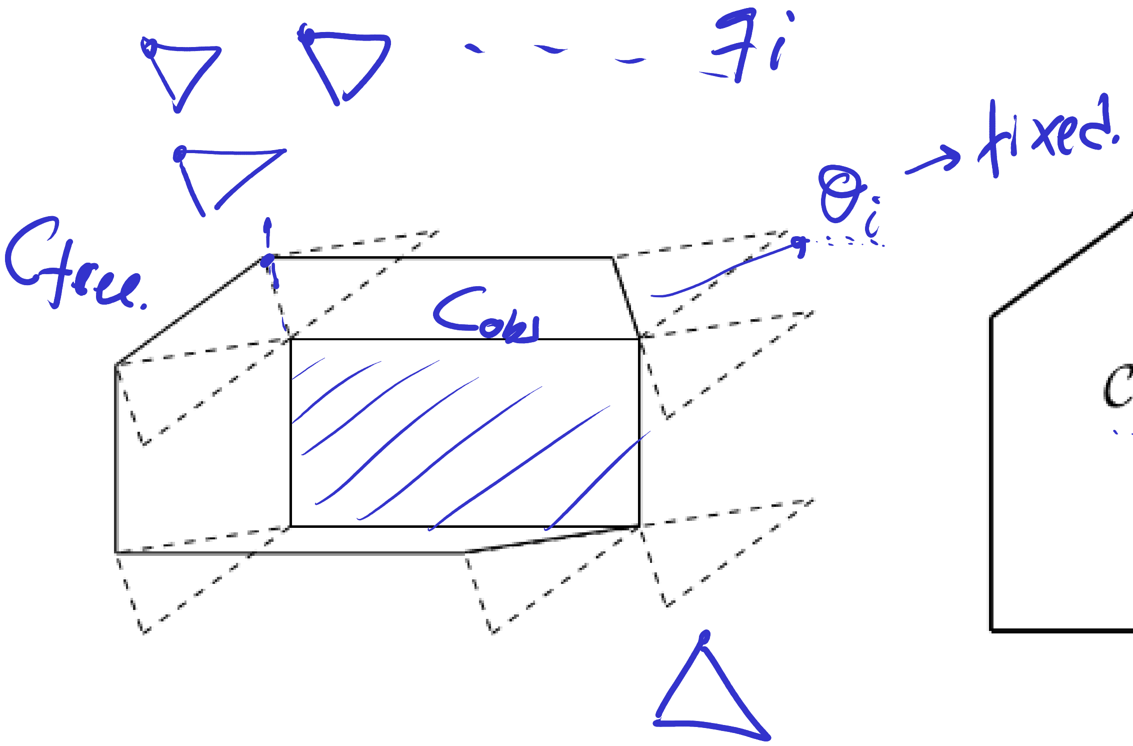
Obstacles in the C-space

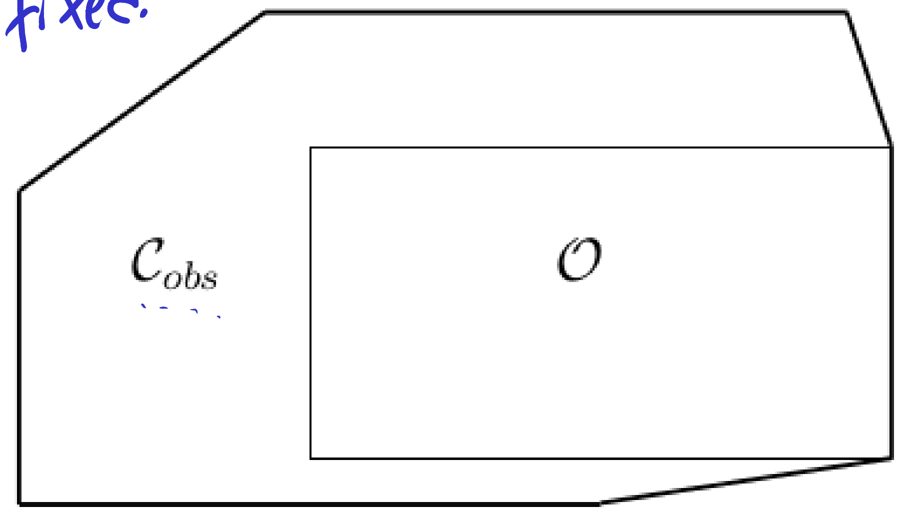
- Suppose the Workspace contains an obstacle region: $\mathcal{O} \subset \mathcal{W}$
- A rigid robot exists in the workspace, such that: $\mathcal{A}(q) \subset \mathcal{W}$ and it has some configuration $q = [x, y, \theta]$, where $q \in \mathcal{C}$
- Then, the **obstacle region**, $\mathcal{C}_{obs} \subseteq \mathcal{C}$, is defined as $\mathcal{C}_{obs} = \{q \in \mathcal{C} \mid \mathcal{A}(q) \cap \mathcal{O} \neq 0\}$



The Obstacle Region in practice

- A very simple approach is to check explicitly for collisions on all configurations, with 2D convolutions of the object against the obstacle.

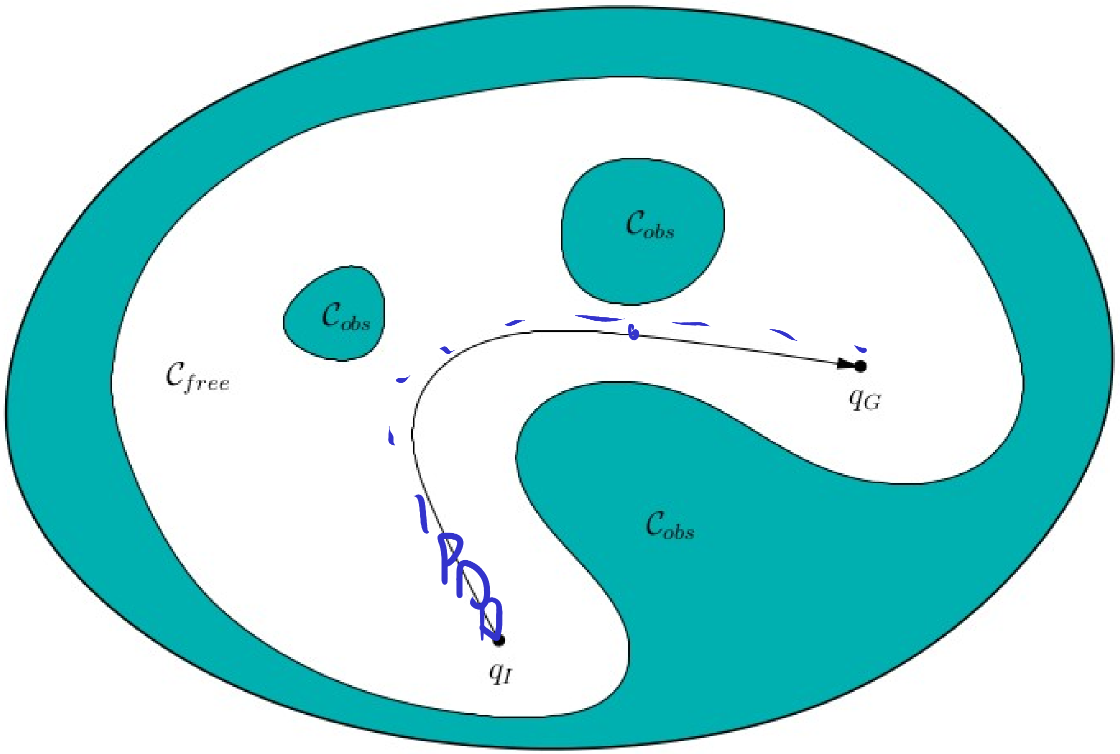




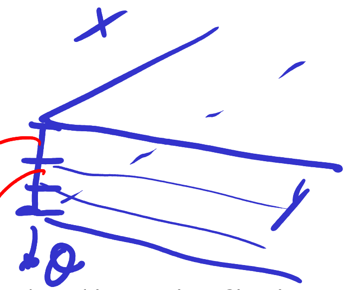
There are many other methods, some we will discuss in future lectures and others are in LaValle for reference.

Obstacles in the C-space

Once defined the obstacle region, one simply applies any planning algorithm to the free configuration space $\mathcal{C}_{free} = \mathcal{C} \setminus \mathcal{C}_{obs}$



The Obstacle Region in practice

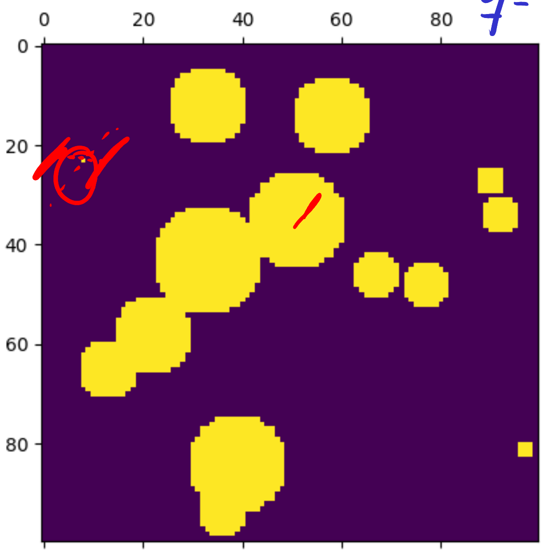


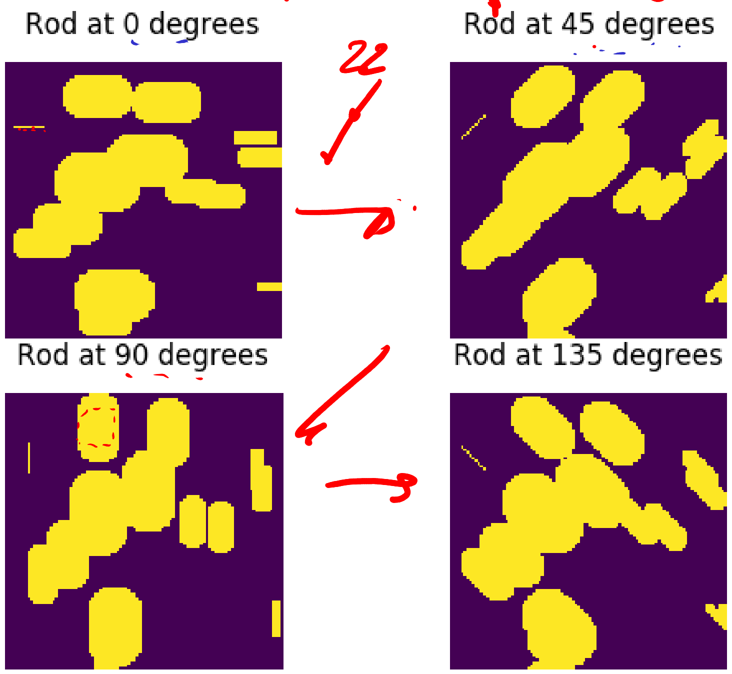
- How can this be practical at all? There are infinite different configurations, including infinite orientations.
- Recall L02 on discrete planning, we can discretize the C-space and calculate the obstacle region in this augmented space and apply graph search.

slices

$q = (x, y, \theta)$ $\theta = 0$

W





22

$\theta = 45$

$\theta = 90$

$\theta = 135$

$\theta = \cancel{\frac{3\pi}{2}}$

all equal. We remove some simmetries