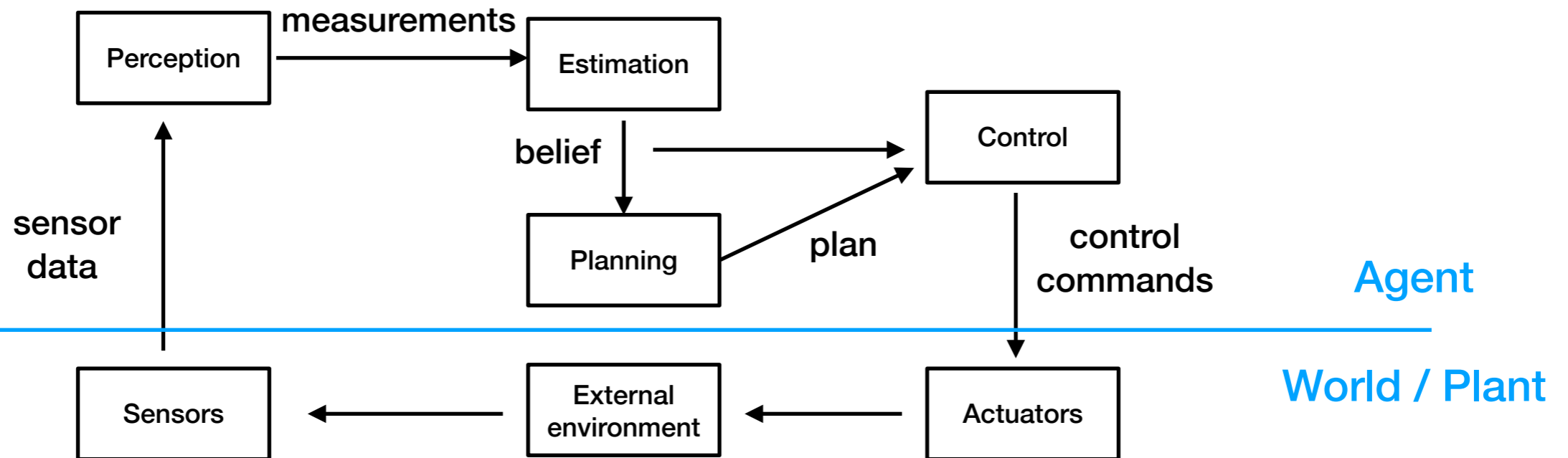




Robot Path Planning

Big picture



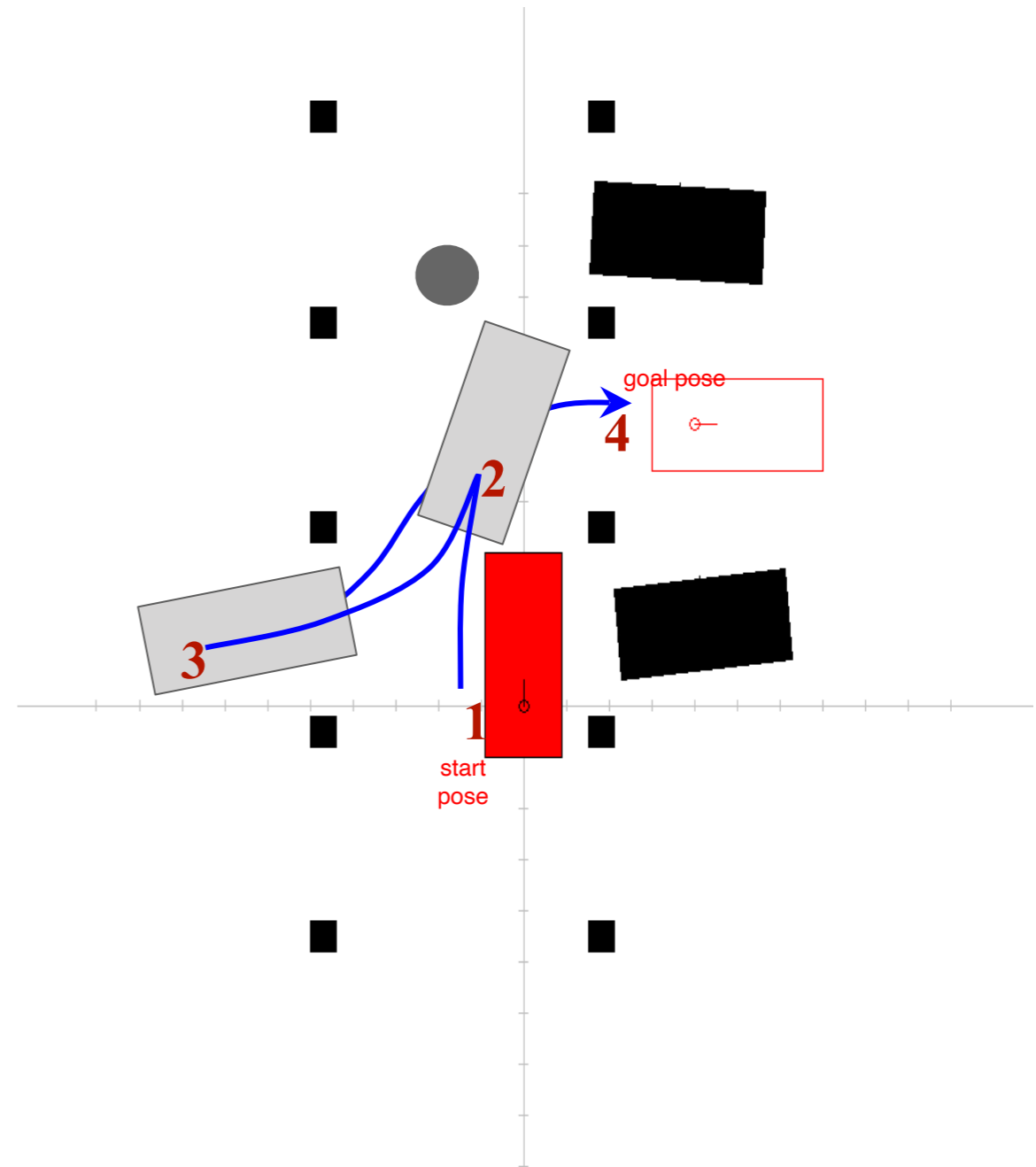
Today we are going to talk about planning

Piano Mover's Problem



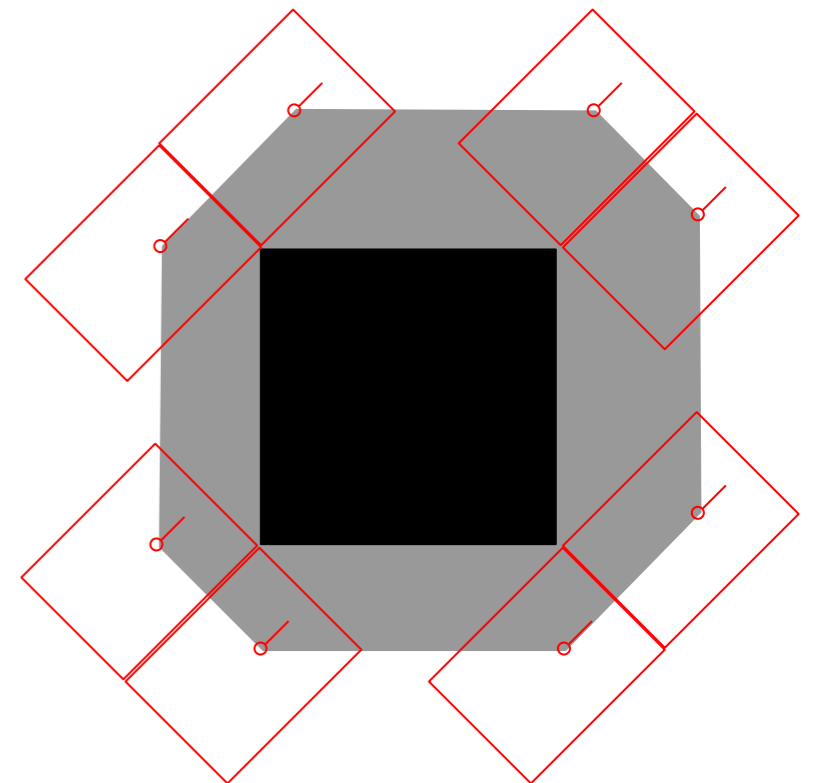
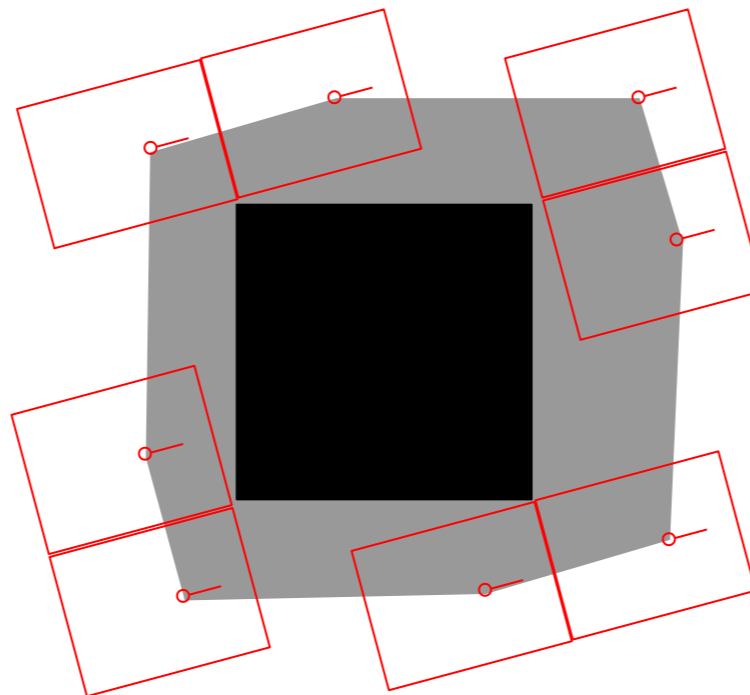
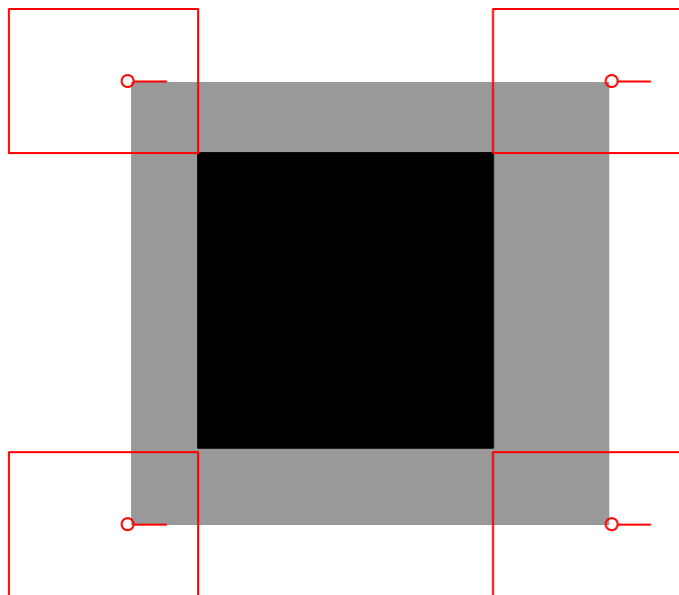
Informal definition of motion planning

- **Path Planning:** Find a feasible, collision-free path from given start pose to given destination pose.
- A desirable motion:
 - Starts at **current position**.
 - Ends at **goal position**.
 - Robot **does not collide** with obstacles
 - Respect **kinematic constraints**
 - Optimizes “**performance objectives**”



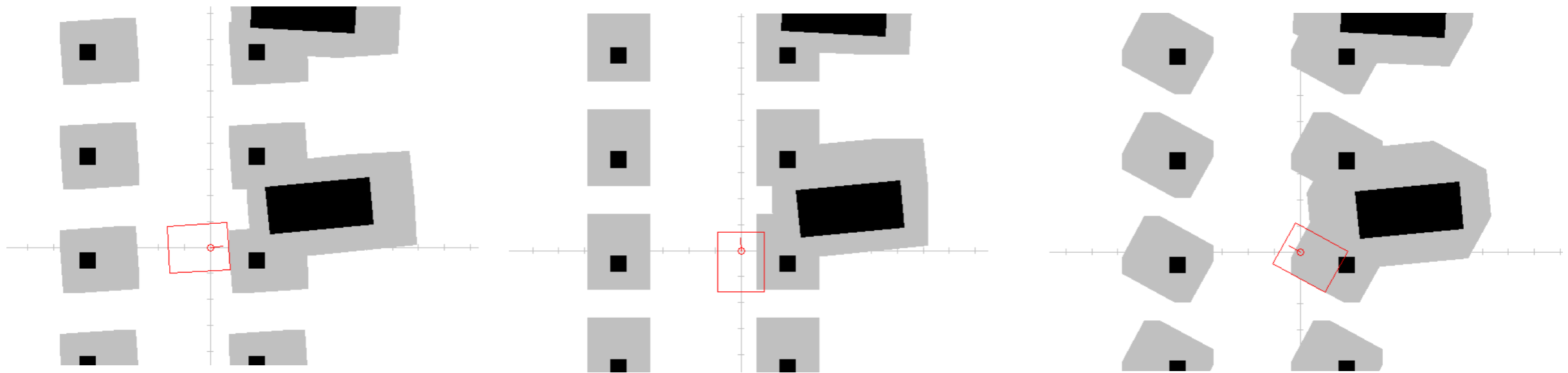
Example for Duckiebot-like robot

- Obstacle space
- W_{free}
- Free configuration space (C_{free})

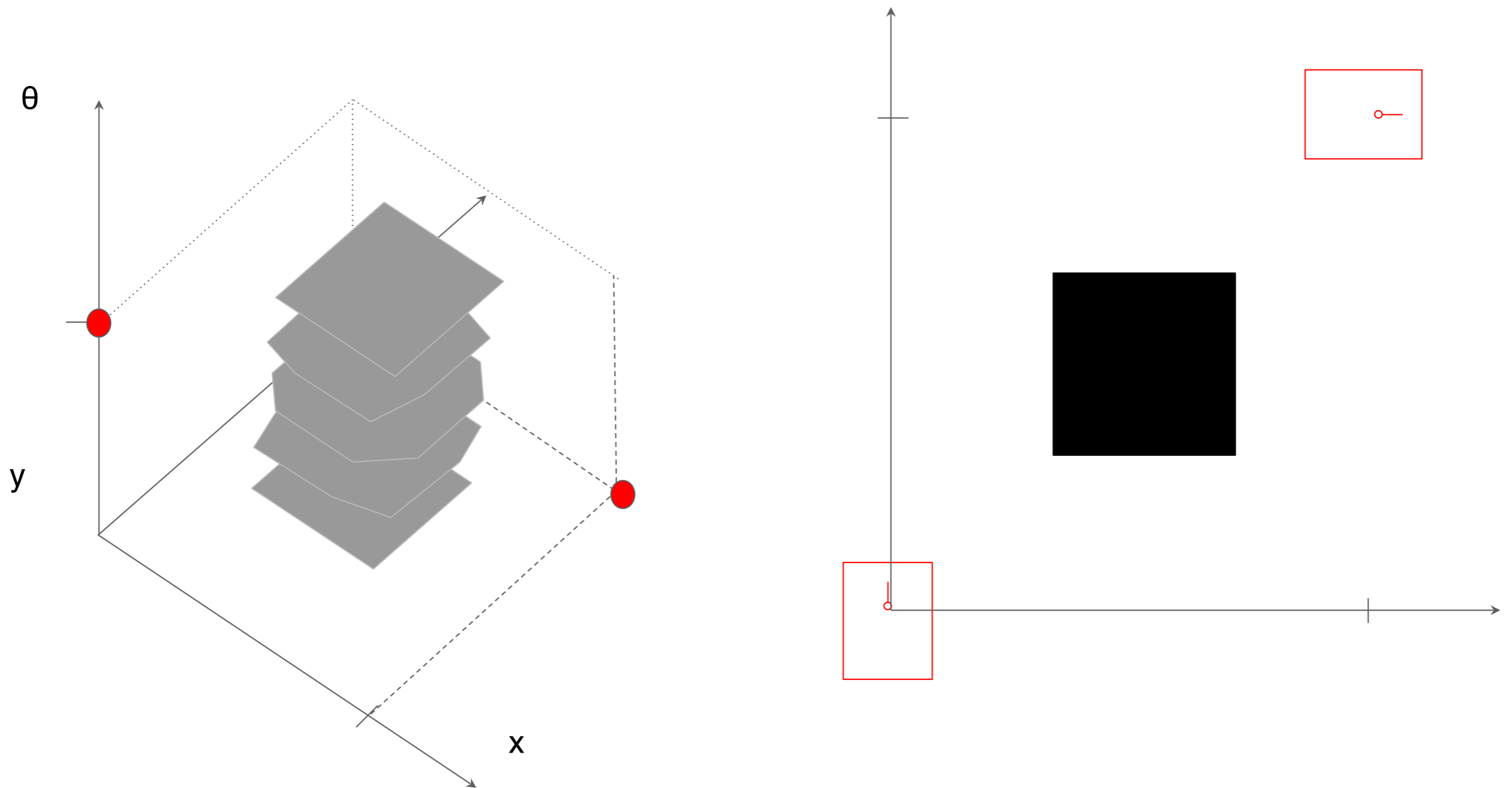


Example for Duckiebot-like robot

- Obstacle space
- W_{free}
- Free configuration space (C_{free})

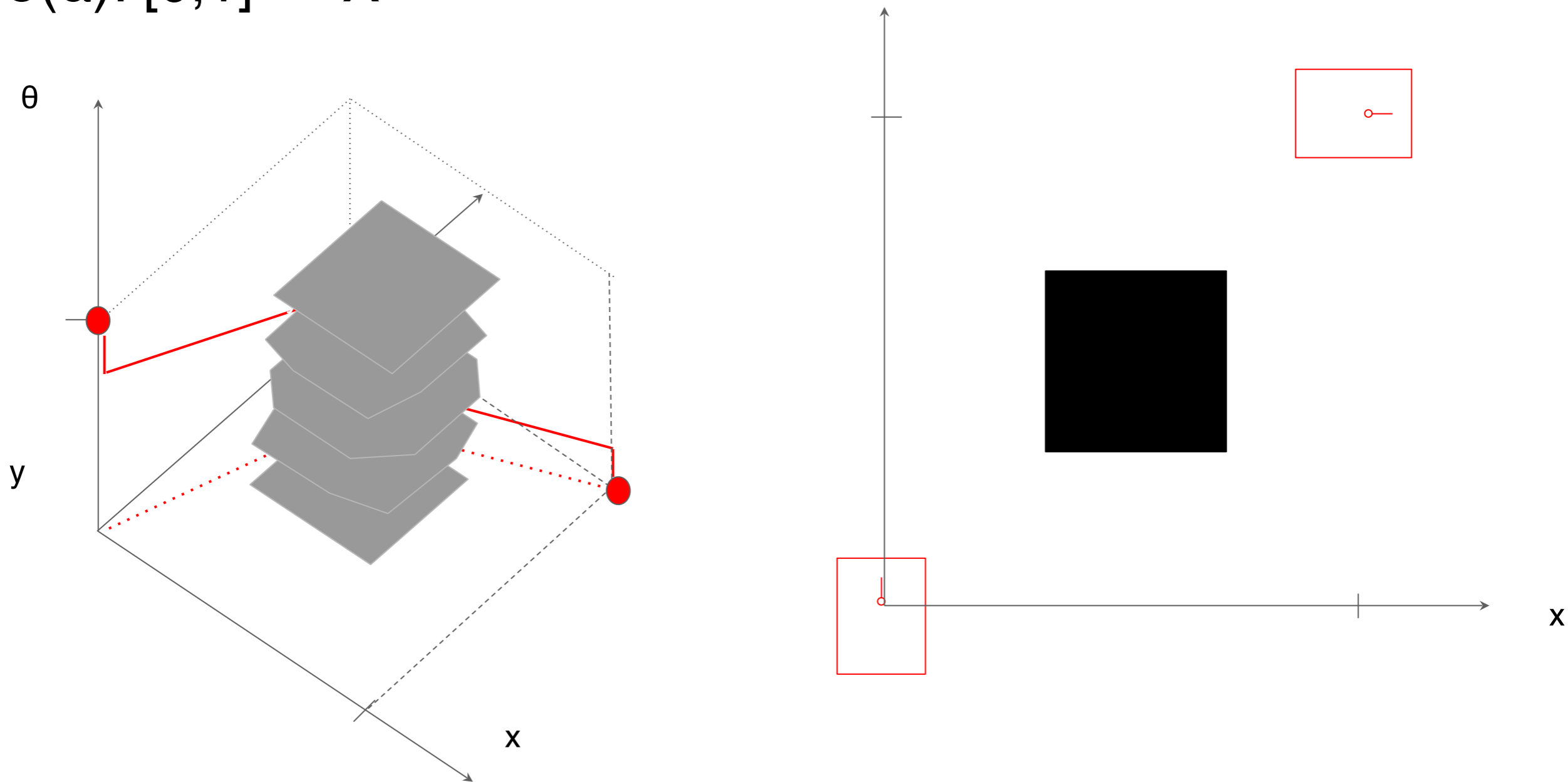


Path in configuration space



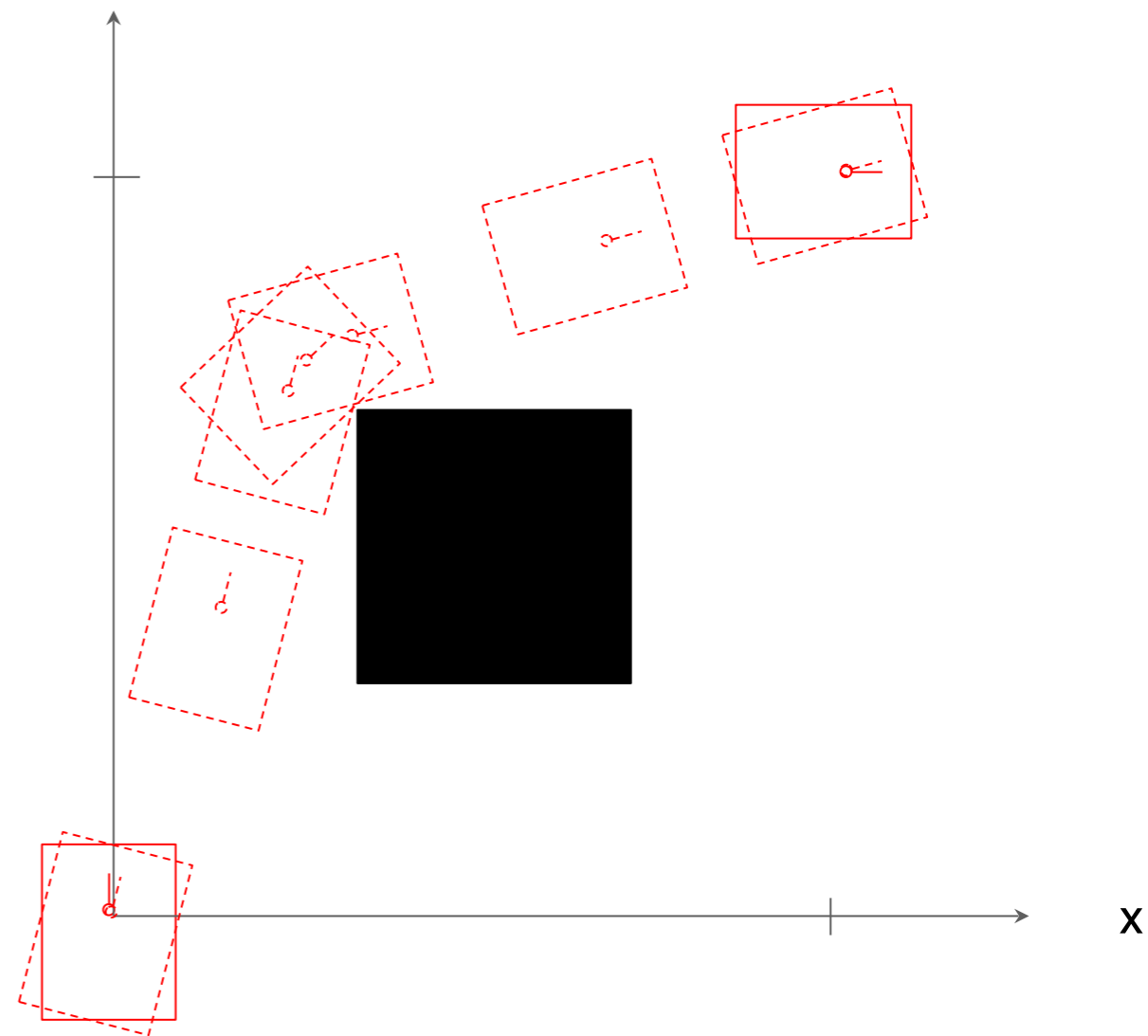
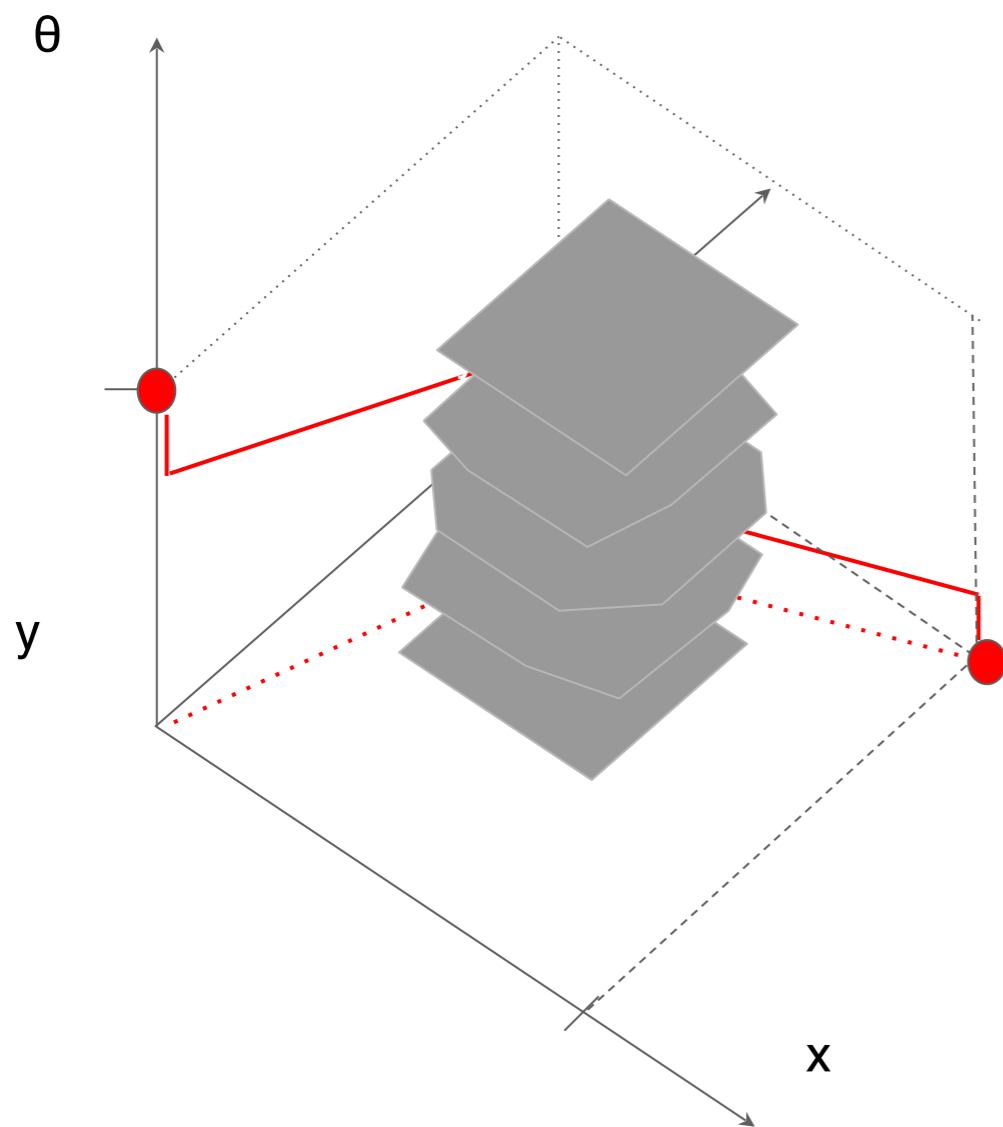
Path in configuration space

$$\sigma(\alpha): [0, 1] \rightarrow X$$



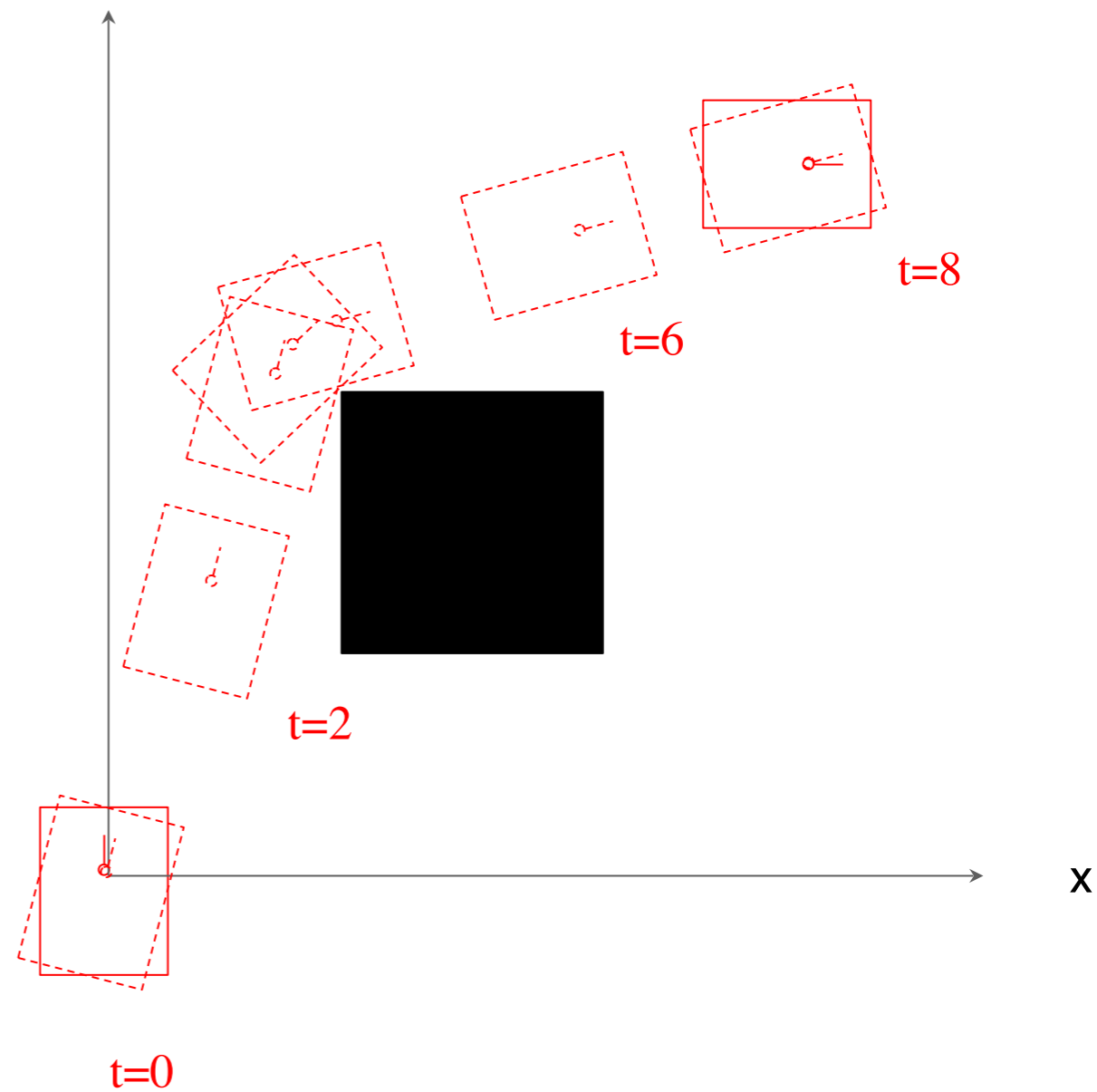
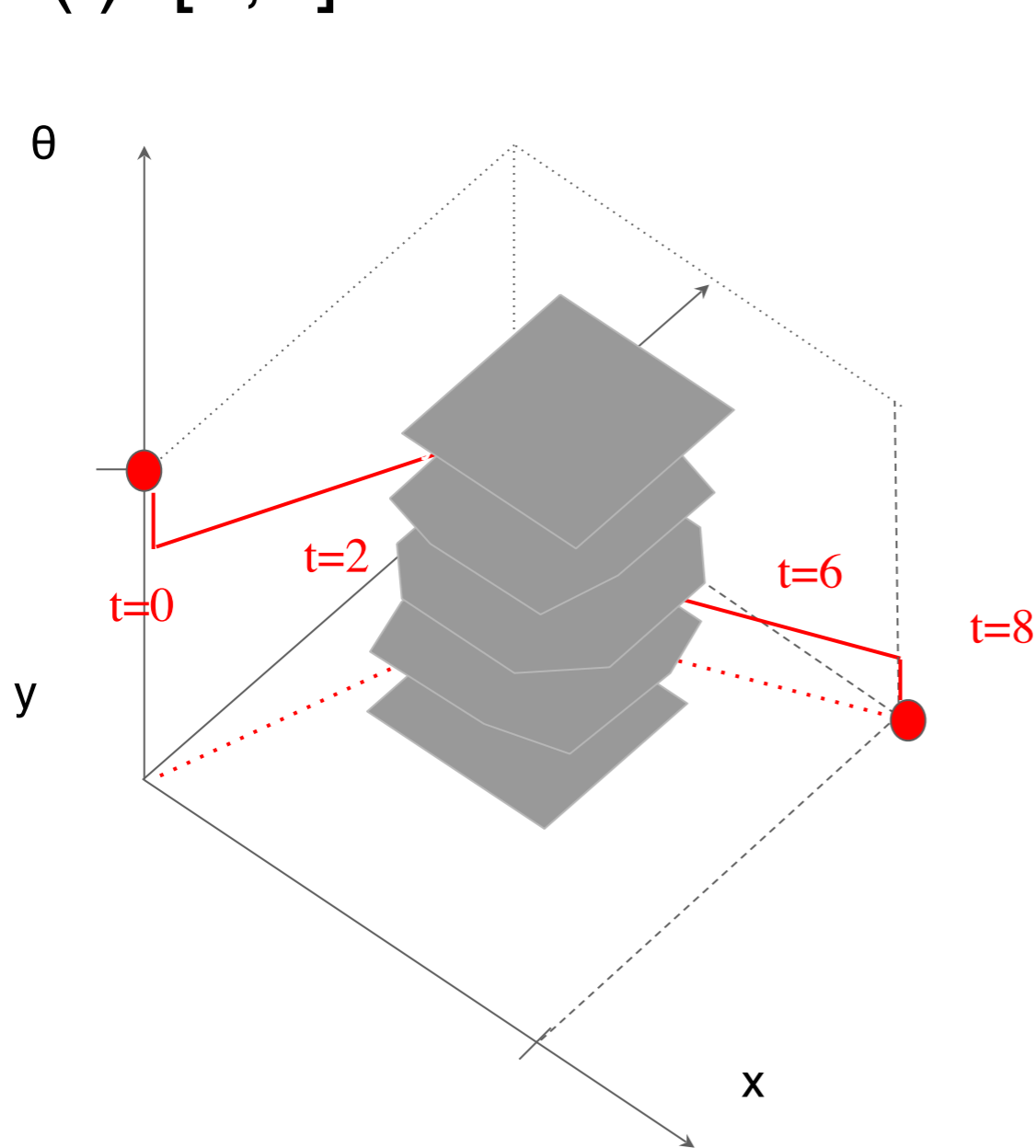
Path in configuration space

$$\sigma(\alpha): [0,1] \rightarrow X$$



Trajectory in configuration space

$$\pi(t): [0, T] \rightarrow X$$



Propriétés des algorithmes de planification

Un algorithme est :

Complet : s'il garantit de trouver un chemin réalisable ou de conclure que le problème n'est pas réalisable.

Optimal : s'il trouve le chemin optimal.

Probabilistiquement complet : si la probabilité que l'algorithme trouve une solution valide tend vers 1 avec le temps d'exécution.

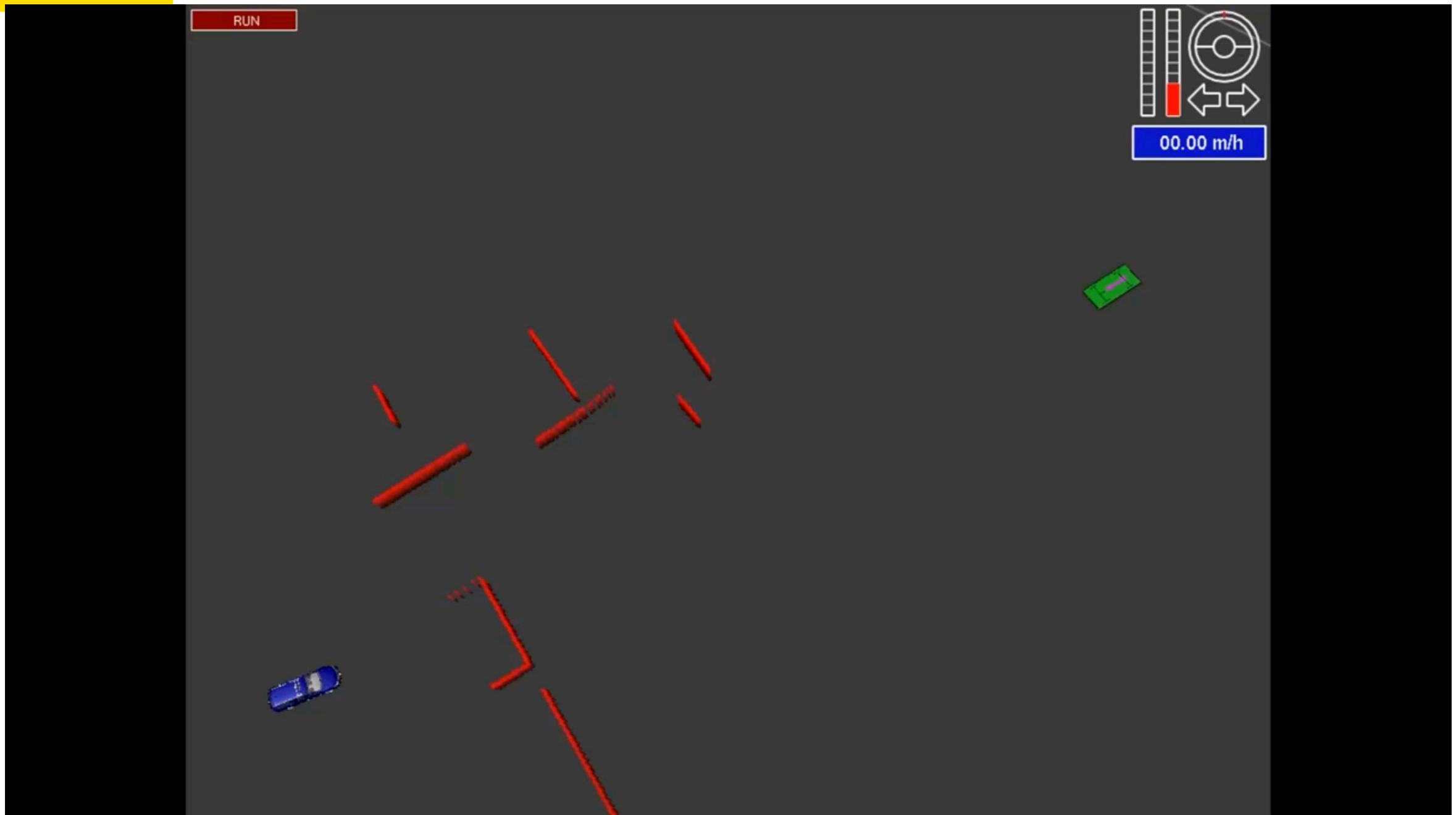
Asymptotement optimal : s'il renvoie une suite de solutions convergeant vers une solution optimale.

Délibératif : construit une représentation et planifie à partir de celle-ci. Contrairement aux algorithmes **réactifs** qui ne le font pas.

Multi-requêtes : la même représentation peut être utilisée pour résoudre plusieurs problèmes de planification.

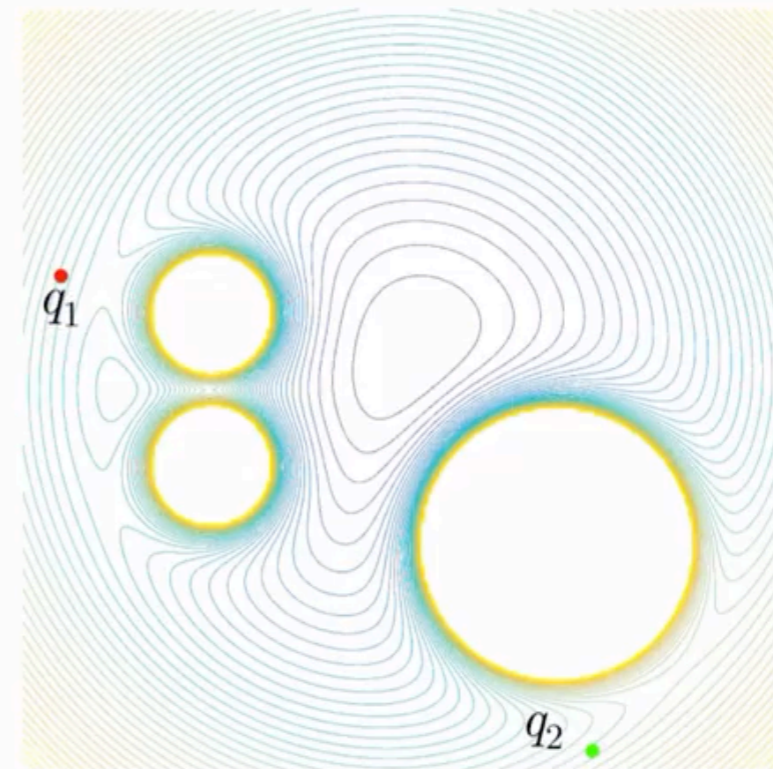
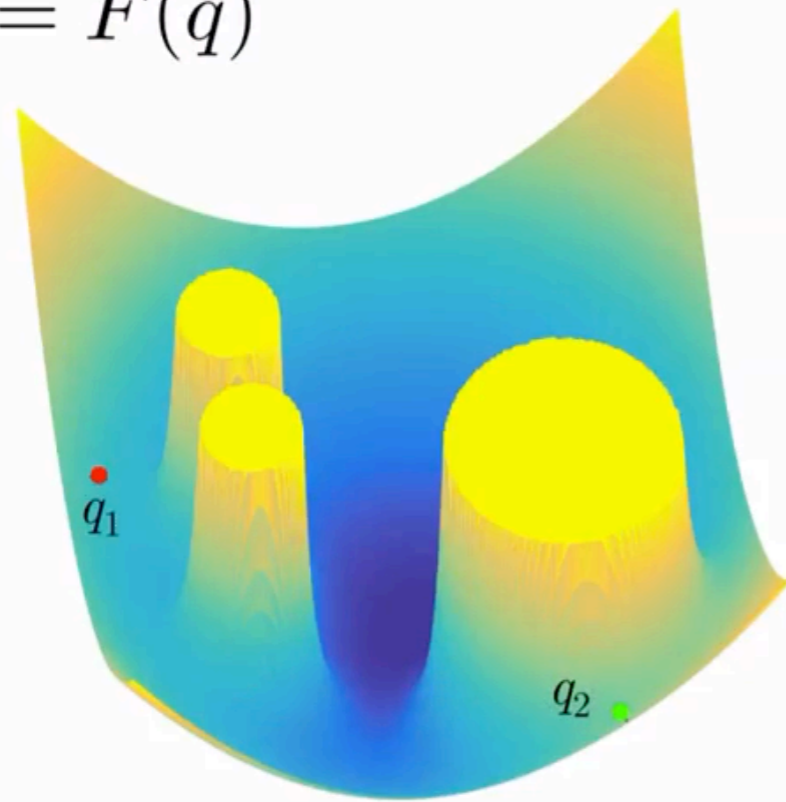
Requête unique : la représentation doit être reconstruite pour chaque problème de planification.

Motion primitives example



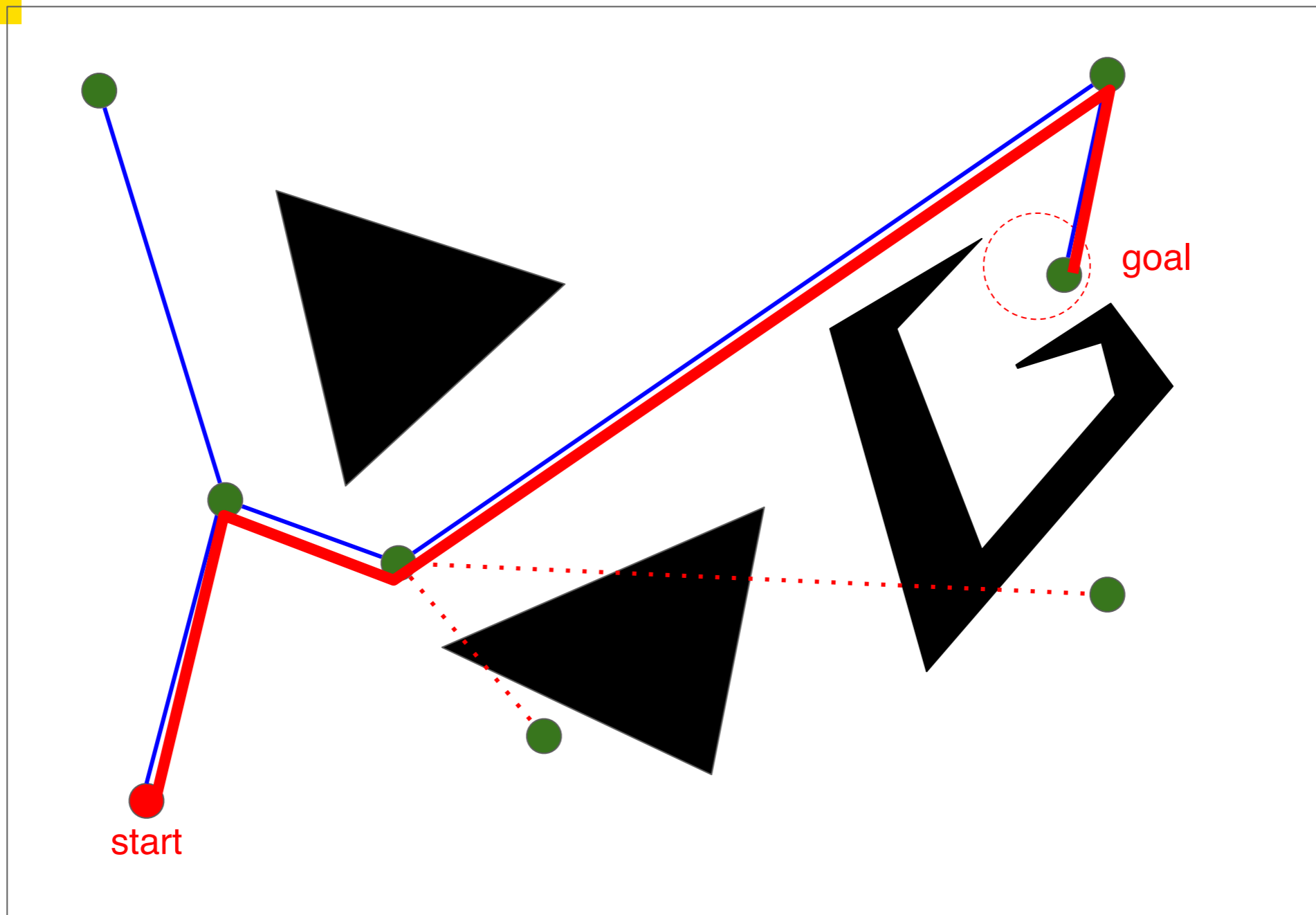
Potential Fields

$$\dot{q} = F(q)$$



Basin of Attraction

Rapidly-exploring Random Tree (RRT)



Optimal Rapidly-exploring Random Tree (RRT*)

