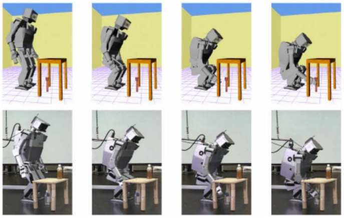
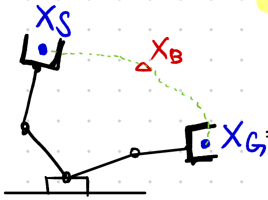
2024.09.23

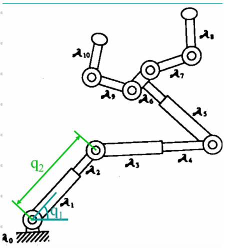


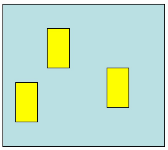



## <Motion Planning>

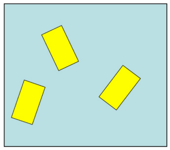
- Given start state $X_S$, goal state $X_G$ (and may be going through $X_B$)
  $\Rightarrow$ Find a sequence of control inputs that leads from $X_S$ to $X_G$, while avoiding obstacles.

## ☆ Degree of Freedom





Allowed to move only in *x* and *y*: 2DOF



Allowed to move in *x* and y and to rotate: 3DOF (*x*,*y*,θ)

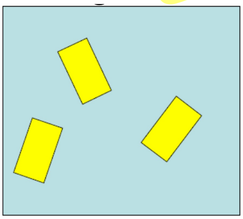
→ Defines geometric configuration of a robot.

→ Assume p DOFs, the geometric configuration A of a robot is defined with p variables

: $A(q)$, $q = (q_1, \dots q_p)$

→ Underactuated : # Actuator < # DoF.
(i.e. inverted pendulum)

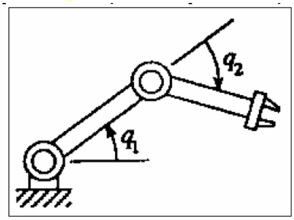
## ☆ Configuration Space



$q = (x,y,\theta)$
$\mathcal{C} = \Re^2 \times$ set of 2-D rotations

*Workspace*

(2 DOF: translation only, no rotation)

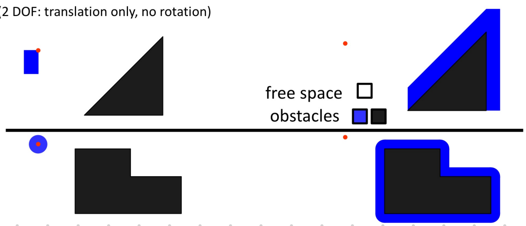


$q = (q_1,q_2)$
$\mathcal{C} =$ 2-D rotations x 2-D rotations

*Configuration Space*

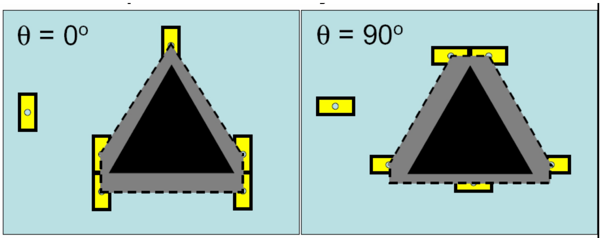
$\mathcal{C} = \{ q \mid q$ is a pose of robot$\}$
$q$ is a legal configuration of robot.
(no collision)

$\Rightarrow$ Defines a set of possible parameters (search space) and the set of allowed paths.




$\mathcal{C}_{free} = \{ q \mid A(q)$ does not intersect objects$\}$

→ For point robot $\Rightarrow \mathcal{C}_{free} = \mathbb{R}^2 - obs$.




→ C-Space $\in \mathbb{R}^3$ $(x, y, \theta)$
different obstacle expansion to each $\theta$.

→ Problem is reduced to point robot by expanding obstacles

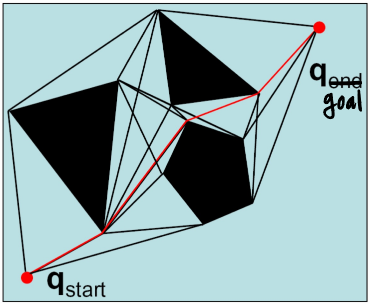
# Motion Planning : Classic Approaches

(1) Road Maps → Do not search whole space.

Find a small graph that stays on "road" while avoiding obs.

Find a path using that graph (roadmap)

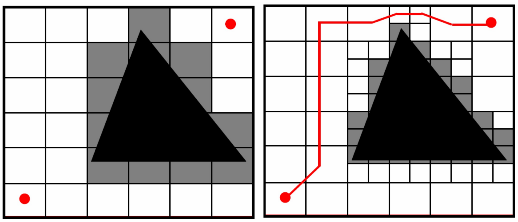
ex) Visibility graph $G = \{$unblocked lines between vertices of obs. $+ q_{start} + q_{goal}\}$




→ Node $P$ is linked to $P'$ if $P'$ is visible from $P$.

→ Find shortest path in $G$.

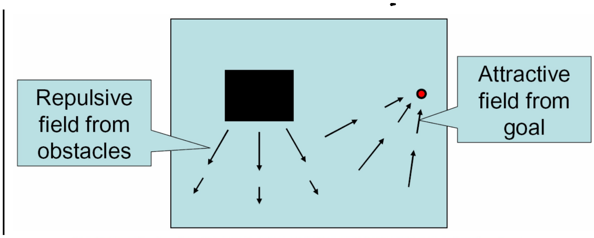
(2) Cell decomposition : Decompose the space into cells w/o obstacle.

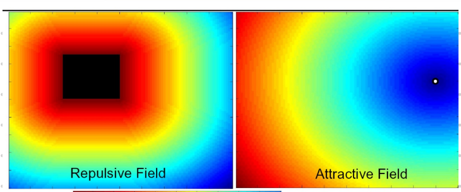


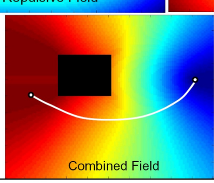
→ Define a discrete grid in C-space.

→ Find path through free cells using A*.

(3) Potential Fields.



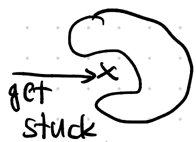

Move toward lowest potential
Steepest descent
(Best first search)
on potential field

$U_g(q) = d^2(q, q_{goal})$

$U_o(q) = 1/d^2(q, \text{obstacles})$

$U(q) = U_g(q) + \lambda U_o(q)$

☆ Has Local Minimum Problem




• $q_{goal}$

## Motion Planning : Optimization-based Approaches. (i.e. CHOMP, Trajopt)

- $\min_{u,x} \quad (X_T - X_G)^T (X_T - X_G)$

  s.t. $\quad x_{t+1} = f(x_t, u_t) \quad \forall t$

  $\quad u_t \in \mathcal{U}_t \ , \ x_t \in \underline{\mathcal{X}_t} , \ x_0 = x_s .$

  Can encode obs.

OR) $\min_{u,x} \quad \|u\|$

s.t. $\quad x_{t+1} = f(x_t, u_t) \quad \forall t$

$\quad u_t \in \mathcal{U}_t \ , \ x_t \in \mathcal{X}_t , \ x_0 = x_s , \ X_T = x_G .$

OR) $\min_{x_{1:T}} \quad \sum_t \|x_{t+1} - x_t\|^2 + \text{other costs}$

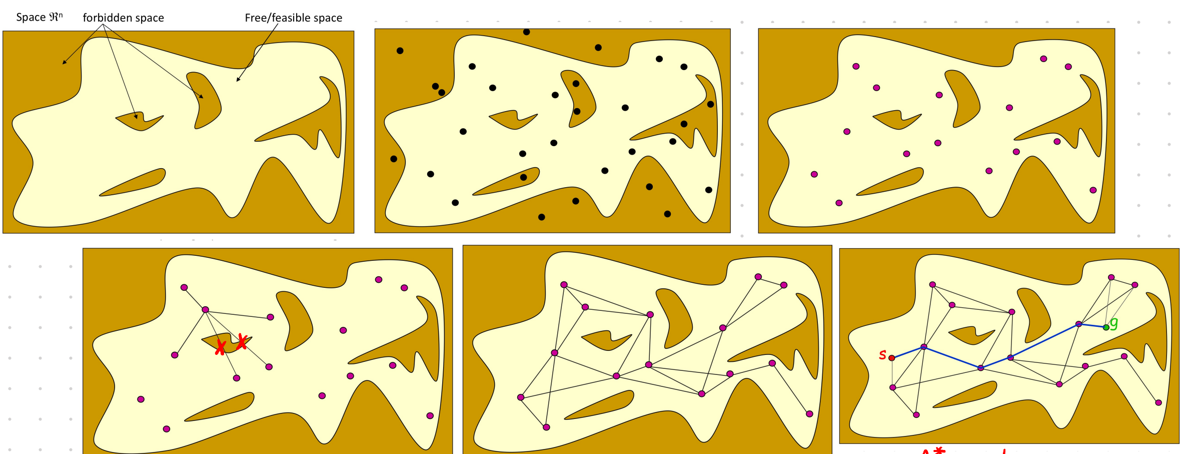
s.t. $x_0$ = start state, $x_T$ in goal set. Joint limits, no collision.

## Motion Planning : Sampling-based approaches.

- Continuous Space → Discretization → A* search.

※ Completely Describing & optimally exploring whole C-space is too hard and not necessary.

→ Try to find a "good" sampling in C-space.




↑ Probablistic Road Map (PRM)

i.e. A* search.