UiO : Department of Technology Systems
University of Oslo

12. Motion planning

Kim Mathiassen

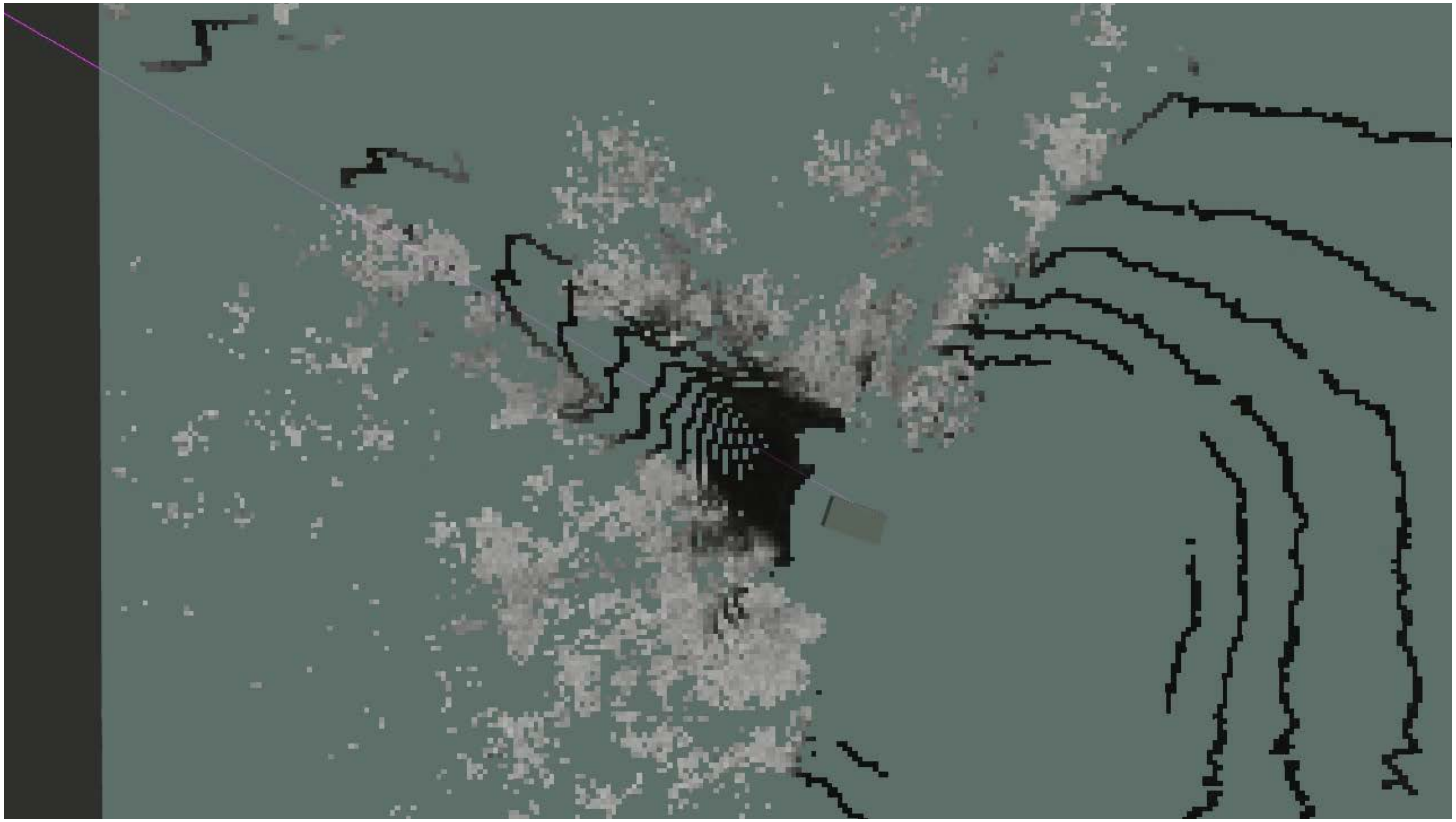
Lecture overview

- Motivation
- The canonical problem (12.1)
- Configuration space (12.2)
- Planning via retraction (12.3)
- Probabilistic planning (12.5)
- Planning via artificial potentials (12.6)

Motion planning

- Planning a trajectory in an environment with obstacles
- This task assumes we get a map of the environment
  - Whole map known in advance
  - Map get updated while the robot moves

Map generation using Lidar



Motion planning

12.1 The canonical problem

- We have a robot $\mathcal{B}$, which is either
  - A single rigid body (mobile robot)
  - A kinematic chain with fixed base (standard manipulator)
  - A kinematic chain with moving base (mobile robot with trailers or mobile manipulator)
- The robot moves in the Euclidian space $\mathcal{W} = \mathbb{R}^N$ with N = 2 or 3. This is called *workspace*
- The obstacles $\mathcal{O}_1, \ldots, \mathcal{O}_p$ are rigid objects in the workspace
- Assumed that the geometry of $\mathcal{B}, \mathcal{O}_1, \ldots, \mathcal{O}_p$ and the poses of $\mathcal{O}_1, \ldots, \mathcal{O}_p$ is known
- It is also assumed that $\mathcal{B}$ is *free-flying* i.e. without constraints

The motion planning problem

- Given an initial and a final posture of the robot ($\mathcal{B}$) in the workspace ($\mathcal{W}$)
- Find a path that drives the robot between the two postures, while avoiding collision
- Report failure if such a path does not exist
- Special case: *the piano movers' problem*
  - The robot is a single body moving in $\mathbb{R}^2$
  - Equivalent with moving a piano without lifting it

Issues with the canonical problem

- Assumes that robot is the only moving object
- Assumes advanced knowledge of obstacle geometry
- The free-flying hypothesis may not hold, because the robot may have non-holonomic constraints
- Manipulation and assembly problems are excluded since they involve contact between rigid bodies

- All these simplifications are introduces to make the problems simpler, but it still a difficult problem

- Many methods that solves the simplified version can be extended to solve more difficult versions of the problem

12.2 Configuration space

- An effective scheme for motion planning is representing the robot as a mobile point in an appropriate space
- Using a set of generalized coordinates of the mechanical system, whose values identifies the robot *configuration*
- The set of all possible configurations is called the configuration space $\mathcal{C}$;
- Generalized coordinates are of two types
  - Cartesian coordinates
  - Angular coordinates

Exampel configuration spaces

- The configuration of a polygonal mobile robot in $\mathcal{W} = \mathbb{R}^2$ is represented by a position (x,y) and an orientation
- The configuration space $\mathcal{C}$ is then $\mathbb{R}^2 \times SO(2)$ with a dimension 3
- For a polyhedral mobile robot in $\mathcal{W} = \mathbb{R}^3$ the configuration space $\mathcal{C}$ is $\mathbb{R}^3 \times SO(3)$ with a dimension of 6
- For a fixed base planar robot with n revolute joints, the configuration space is a subspace of $(\mathbb{R}^2 \times SO(2))^n$
- The subspace is limited by the number of constraints due to joints, i.e. 3n – 2n = n
- The planar kinematic chain imposes two holonomic constraints for each joint

Configuration space

- If n is the dimension of $\mathcal{C}$., then a configuration can be described by a vector $\boldsymbol{q} \in \mathbb{R}^n$
- The geometry of a configuration space is generally more complex than Euclidian space
- Consider a planar manipulator with two revolute joints
- The configuration space has dimension 2, and is represented by a subset of $\mathbb{R}^2$

$$\mathcal{Q} = \{\boldsymbol{q} = (q_1, q_2) : q_1 \in [0, 2\pi), q_2 \in [0, 2\pi)\}.$$

- It can be visualized as a two dimensional surface in 3D space
- The correct expression of the space is

$$SO(2) \times SO(2)$$

Configuration space

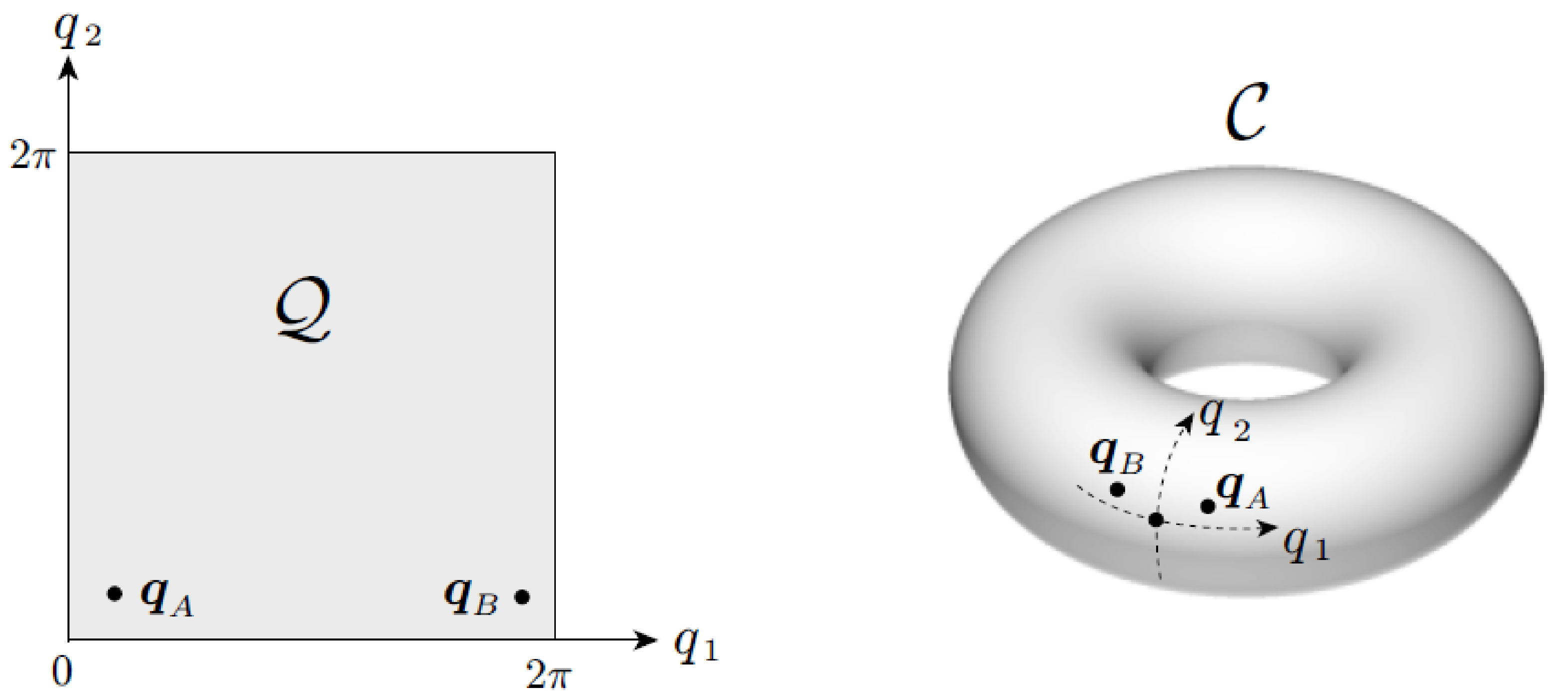


Fig. 12.1. The configuration space of a 2R manipulator; *left*: a locally valid representation as a subset of $\mathbb{R}^2$, *right*: a topologically correct representation as a two-dimensional torus

12.2.1 Distance

- Given a configuration $\boldsymbol{q}$
  - $\mathcal{B}(\boldsymbol{q})$ is the subset of the workspace occupied by the robot
  - $p(\boldsymbol{q})$ is the position of a point p on $\mathcal{B}$ in the workspace
- Intuition says that the distance between $\boldsymbol{q}_A$ and $\boldsymbol{q}_B$ should go to zero when the two regions $\mathcal{B}(\boldsymbol{q}_A)$ and $\mathcal{B}(\boldsymbol{q}_B)$ coincide
- The following distance measure satisfies this

$$d_1(\boldsymbol{q}_A, \boldsymbol{q}_B) = \max_{p \in \mathcal{B}} \|p(\boldsymbol{q}_A) - p(\boldsymbol{q}_B)\|, \tag{12.1}$$

- The measure is, however, cumbersome to compute

Distance

- Often the Euclidian norm is used

$$d_2(q_A, q_B) = \|q_A - q_B\|. \tag{12.2}$$

- This is only valid in Euclidian space, not necessarily in configuration space
- For instance for the two link planar robot the measure does not represent the correct distance on the torus
- A possible solution is to compute the difference in angle in the correct space

12.2.2 Obstacles

- In order to find paths that solve the canonical problem it is necessary to build «images» of the obstacles in configuration space
- Given an obstacle $\mathcal{O}_i \ (i = 1, \dots, p)$ in $\mathcal{W}$ its image in configuration space $\mathcal{C}$ is called $\mathcal{C}$-*obstacle*
- It is defined as

$$\mathcal{CO}_i = \{\boldsymbol{q} \in \mathcal{C} : \mathcal{B}(\boldsymbol{q}) \cap \mathcal{O}_i \neq \emptyset\}. \qquad (12.3)$$

- In other words $\mathcal{CO}_i$ is the subset of configurations that cause a collision between the robot $\mathcal{B}$ and the obstacle $\mathcal{O}_i$

Obstacles

- The union of all obstacles

$$\mathcal{CO} = \bigcup_{i=1}^{p} \mathcal{CO}_i \tag{12.4}$$

- Defines the $\mathcal{C}$*-obstacle region*,
- The *free configuration space* is defined as its complement

$$\mathcal{C}_{\text{free}} = \mathcal{C} - \mathcal{CO} = \{\boldsymbol{q} \in \mathcal{C} : \mathcal{B}(\boldsymbol{q}) \cap \left(\bigcup_{i=1}^{p} \mathcal{O}_i \right) = \emptyset\} \tag{12.5}$$

- A path in the configuration space is called *free* if it is entirely contained in $\mathcal{C}_{\text{free}}$

Obstacles

- Although $\mathcal{C}$ is a connected space, $\mathcal{C}_{\text{free}}$ may not be connected as a consequence of occlusions cause by $\mathcal{C}$-obstacles
- It is now possible to give a more compact formulation of the canonical problem
- Given a start configuration q_s and a goal configuration q_g the method should plan a free path between q_s and q_g if they belong to the same connected component
- If not the method should fail

12.2.3 Examples of obstacles

- Consider a point robot $\mathcal{B}$. (ex. 12.2)
  - The configuration is described by the coordinates of $\mathcal{B}$ in $\mathcal{W} = \mathbb{R}^N$
  - The configuration space $\mathcal{C}$ is the same as the workspace $\mathcal{W}$
  - $\mathcal{C}$-obstacles are copies of the obstacles in $\mathcal{W}$
- Consider a circular robot (ex 12.3)
  - The orientation of the robot is irrelevant for collision checking
  - Therefore the configuration space is a copy of the workspace
  - However $\mathcal{C}$-obstacles are no longer copies of the obstacles in $\mathcal{W}$
  - They must be obtained through a growing procedure

Example 12.3

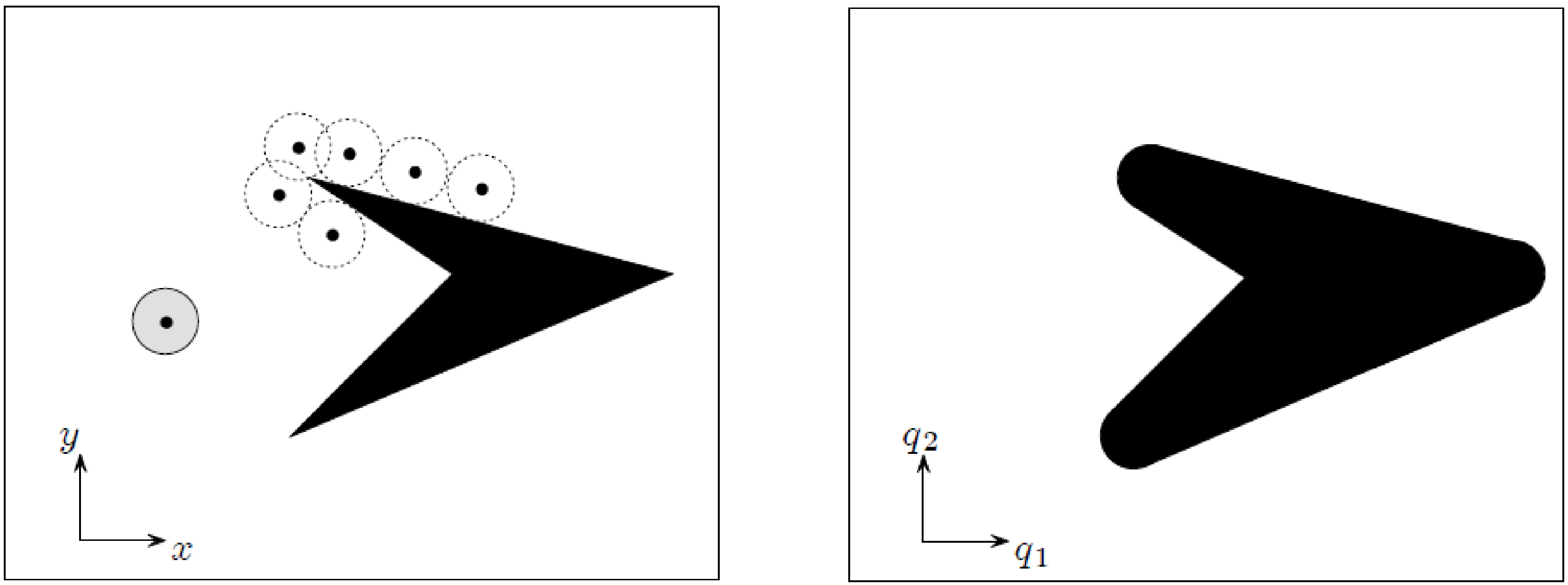


Fig. 12.2. $\mathcal{C}$-obstacles for a circular robot in $\mathbb{R}^2$; *left*: the robot $\mathcal{B}$, an obstacle $\mathcal{O}_i$ and the growing procedure for building $\mathcal{C}$-obstacles, *right*: the configuration space $\mathcal{C}$ and the $\mathcal{C}$-obstacle $\mathcal{CO}_i$

Examples of obstacles

- Consider a polynomial robot that is free to translate (ex. 12.4)
  - Configuration space is again a copy of the workspace
  - A growing procedure must be applied to the workspace obstacles in order to get the $\mathcal{C}$-obstacle
  - The resulting shape of the $\mathcal{C}$-obstacles depend of the representative point on the robot

Example 12.4

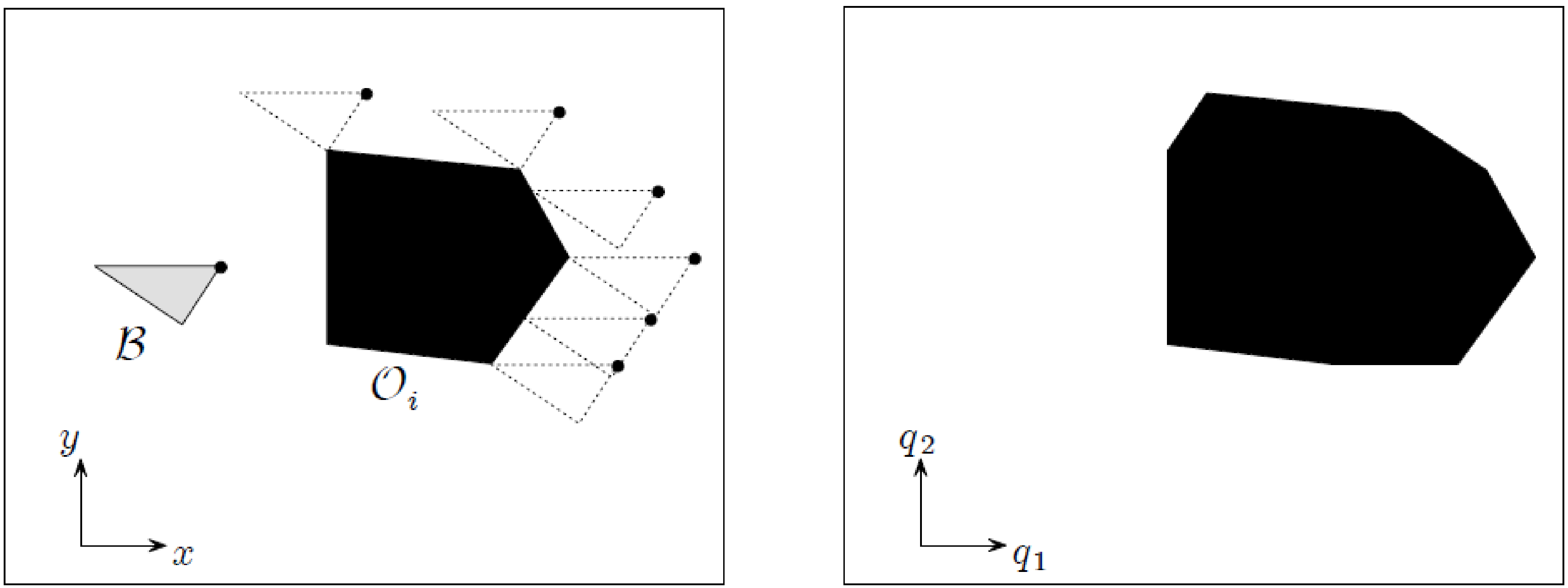
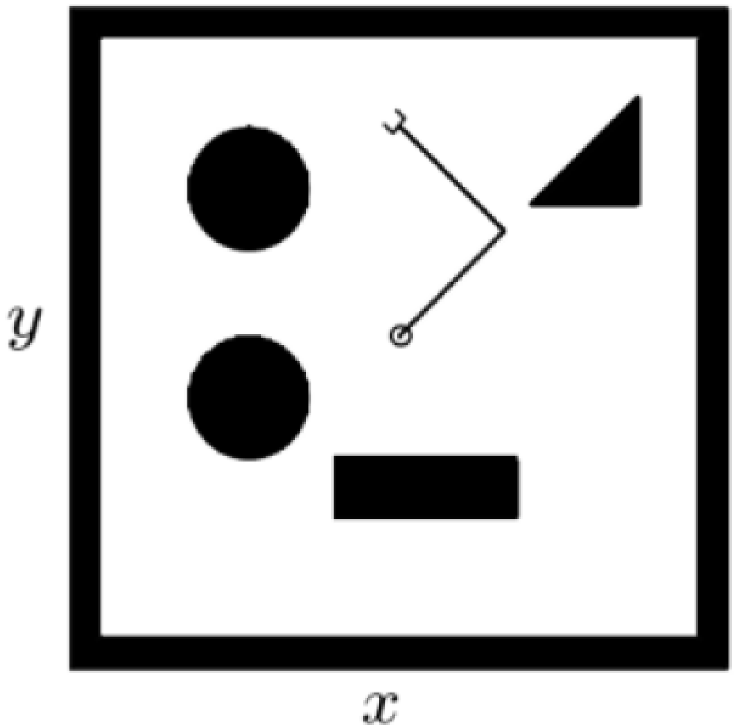


Fig. 12.3. $\mathcal{C}$-obstacles for a polygonal robot translating in $\mathbb{R}^2$; *left*: the robot $\mathcal{B}$, an obstacle $\mathcal{O}_i$ and the growing procedure for building $\mathcal{C}$-obstacles, *right*: the configuration space $\mathcal{C}$ and the $\mathcal{C}$-obstacle $\mathcal{CO}_i$

Examples of obstacles

- Consider a robot manipulator with n links (ex. 12.6)
  - There exists two kind of obstacles: self-collision and environment collision
  - Each link body must be checked for both obstacles using inverse kinematics
  - More complex than the previous cases

Example 12.6



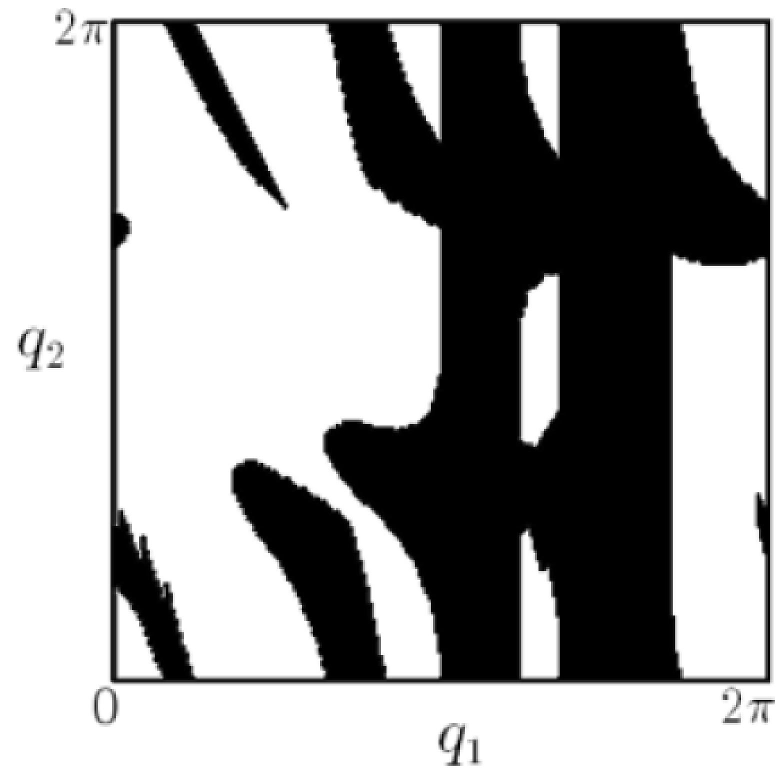
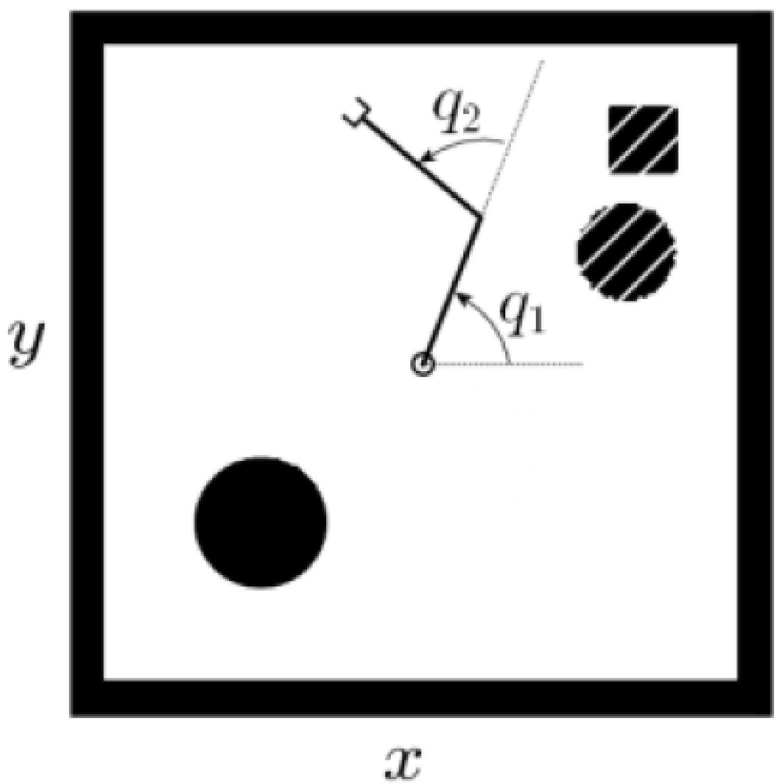
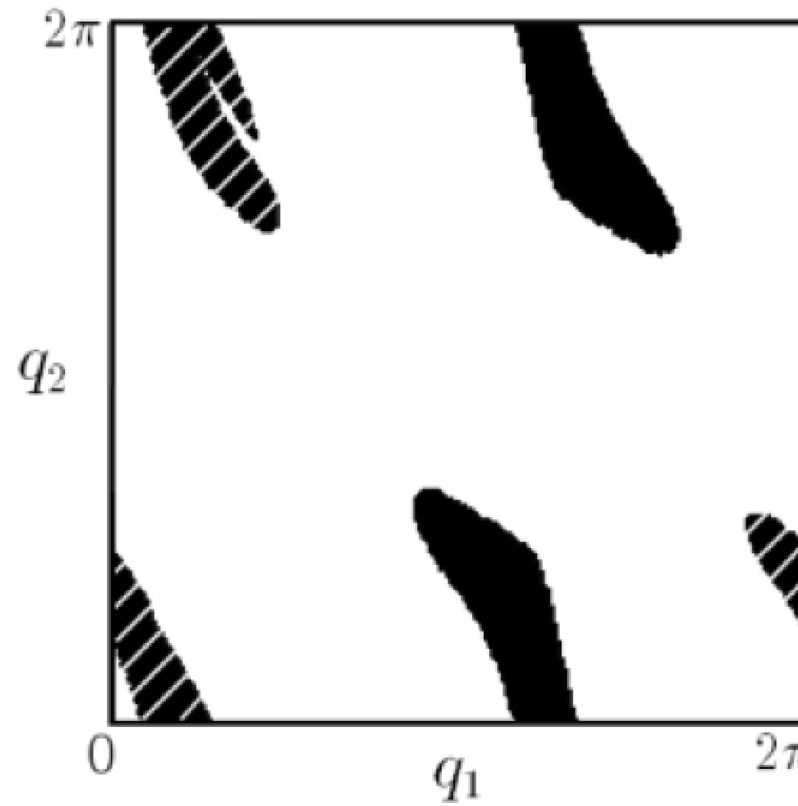


Fig. 12.4. $\mathcal{C}$-obstacles for a wire-frame 2R manipulator in two different cases; *left*: the robot and the obstacles in $\mathcal{W} = \mathbb{R}^2$, *right*: the configuration space $\mathcal{C}$ and the $\mathcal{C}$-obstacle region $\mathcal{CO}$

Example 12.6





12.3 Planning via Retraction

- The basic idea is to represent the free configuration space by the means of a roadmap
- The solution of a particular instance of a motion planning problem is to connect (retract) the start and goal configuration to the roadmap
- Depending on the type of roadmap and the retraction procedure, this general procedure leads to many different planning methods
- We will look at one method that assumes that $\mathcal{C}_{\text{free}}$ is a limited subspace of $\mathcal{C} = \mathbb{R}^2$ and is polygonal i.e. boundary consist only of line segments

Planning via retraction

- First we must generate a roadmap
- For each q in $\mathcal{C}_{\text{free}}$ let the clearance be defined as

$$\gamma(\boldsymbol{q}) = \min_{\boldsymbol{s} \in \partial\mathcal{C}_{\text{free}}} \|\boldsymbol{q} - \boldsymbol{s}\|, \tag{12.6}$$

- Where $\partial\mathcal{C}_{\text{free}}$ is the boundary of $\mathcal{C}_{\text{free}}$
- The clearance defines the minimum Euclidian distance between a configuration and an obstacle
- Any configuration q that has at least two points s that satisfies the equation

$$\|\boldsymbol{q} - \boldsymbol{s}\| = \gamma(\boldsymbol{q})$$

- Is a part of the roadmap

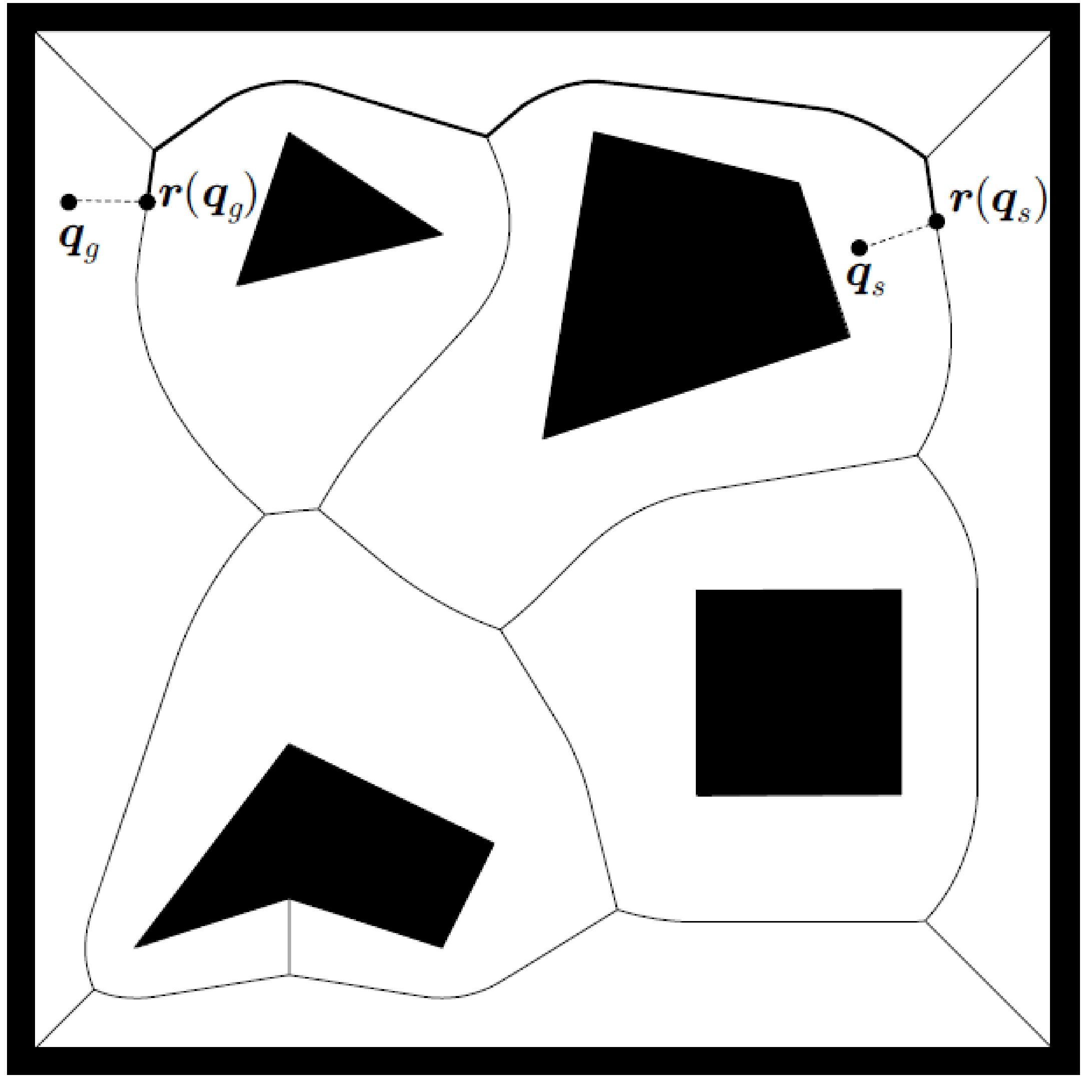


Fig. 12.5. An example of generalized Voronoi diagram and the solution of a particular instance of the planning problem, obtained by retracting $\boldsymbol{q}_s$ and $\boldsymbol{q}_g$ on the diagram. The solution path from $\boldsymbol{q}_s$ to $\boldsymbol{q}_g$ consists of the two *dashed* segments and the *thick* portion of the diagram joining them

Planning via retraction

- This generates a generalized Voronoi diagram
- This diagram will locally maximize the clearance

- The next step is finding the connection/retraction to the roadmap
- Consider a configuration q, there is only one point s on the boundary $\partial\mathcal{C}_{\text{free}}$ that satisfies the equation
$$\|\boldsymbol{q} - \boldsymbol{s}\| = \gamma(\boldsymbol{q})$$
- The gradient $\nabla\gamma(\boldsymbol{q})$ determines the steepest ascent for the clearance
- The point $\boldsymbol{r}(\boldsymbol{q})$ is the first intersection between the diagram and the line from q following $\nabla\gamma(\boldsymbol{q})$

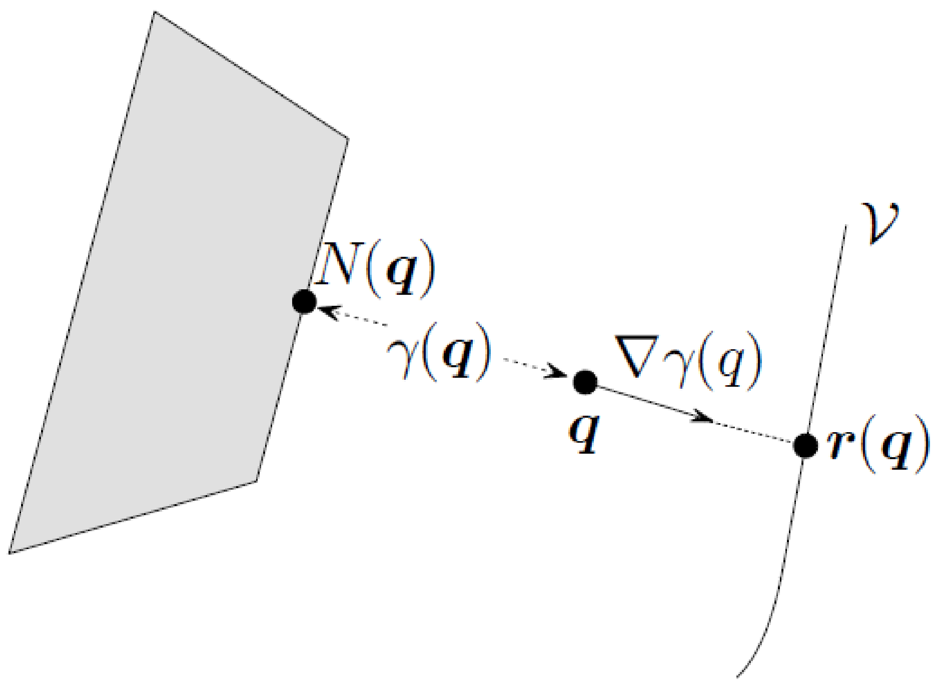


Fig. 12.6. The retraction procedure for connecting a generic configuration $\boldsymbol{q}$ in $\mathcal{C}_{\text{free}}$ to $\mathcal{V}(\mathcal{C}_{\text{free}})$

Planning via retraction

- General method
  - Build the generalized Voronai diagram
  - Compute the retractions $\boldsymbol{r}(\boldsymbol{q}_s)$ and $\boldsymbol{r}(\boldsymbol{q}_g)$
  - Search the graph to find the path between $\boldsymbol{r}(\boldsymbol{q}_s)$ and $\boldsymbol{r}(\boldsymbol{q}_g)$

- The method is guarantied to find a solution if one exists
- Once the roadmap has been built, the method can solve the problem quickly for other instances (queries)
- Useful for when the robot must repeatedly move in a static workspace
- It can be considered a *multi-query* method

12.5 Probabilistic planning

- Represent a class of method that efficient in solving high dimension configuration space problems
- They belong to the general family of sampling based methods
- The basic idea is to find a finite set of collision free configuration that adequately represents the connectivity of $\mathcal{C}_{\mathrm{free}}$
- Then these configurations are used to build a roadmap
- Two planners of this type will be presented
  - Probabilistic roadmap (PRM)
  - Bidirectional Rapid-exploring Random Tree (Bidirectional RRT)

12.5.1 Probabilistic roadmap (PRM)

- Incrementally build up a roadmap by selecting a random configuration q_{rand} from a uniform distribution
- Then q_{rand} is tested for collisions, and if it is found to be collision free it is added to the map
- After adding it to the map q_{rand} is connected to sufficiently near configurations through free local paths
- Nearness if often defined as Euclidian space distance
- A local planner finds a free local path between q_{rand} and a near configuration q_{near}
- Straight lines are often used between the points

Probabilistic roadmap (PRM)

- The incremental generation stops either when
  - A maximum number of iterations is reached
  - The number of connected components in the roadmap become smaller than a threshold
- Then one verifies that it is possible to solve the problem of connecting $q_s \text{ and } q_g$ to the same connected component
- If a solution cannot be found the roadmap must be improved
  - Having more iterations
  - Connecting unconnected components that are close to each other
- The method is efficient in finding a solution
- Repeated queries improves the roadmap

Probabilistic roadmap (PRM)

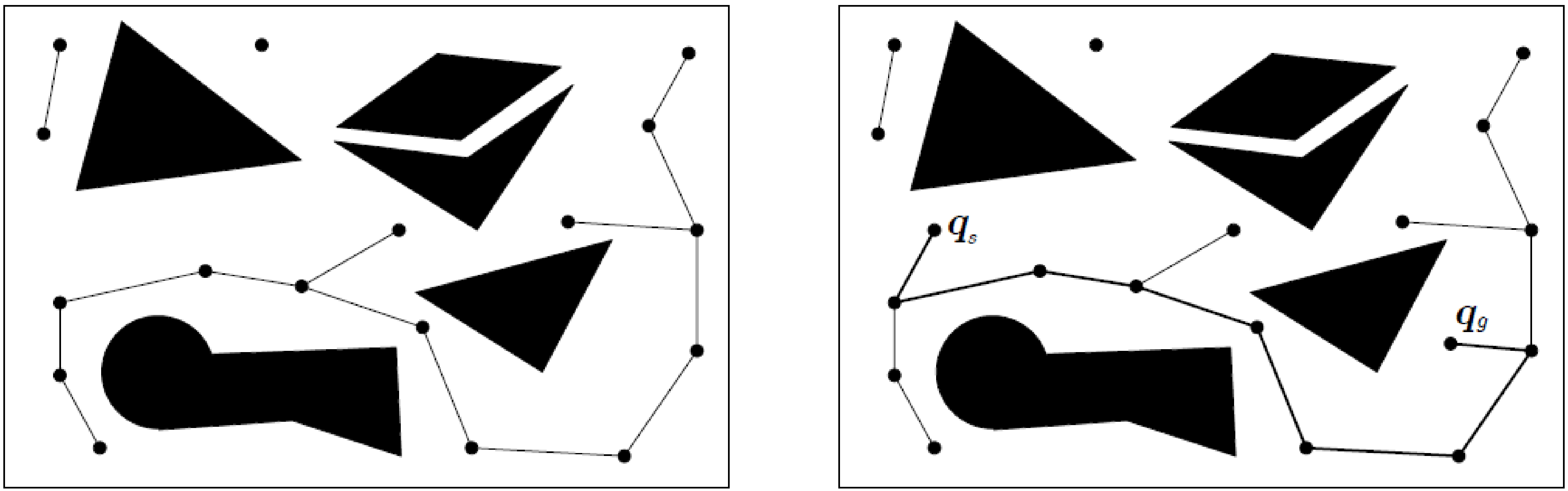


Fig. 12.9. A PRM in a two-dimensional configuration space (*left*) and its use for solving a particular planning problem (*right*)

Probabilistic roadmap (PRM)

- A *multi-query* method
- Only probabilistic complete, meaning the probability of finding a solution (if one exists) goes to one as the execution time goes to infinity
- An issue with the method is the narrow passages have low probability of being selected
- Using a non-uniform distribution may overcome this problem

12.5.2 Bidirectional Rapid-exploring Random Tree (Bidirectional RRT)

- A *single query* probabilistic method
- Does not compute a roadmap
- Generally faster for solving only one instance of the problem

- For each iteration a random sample of the configuration space q_{rand} is selected from a uniform distribution
- Then a configuration q_{near} already in the tree is found, which is close to q_{rand}
- A new candidate configuration q_{new} is found on the line between q_{rand} and q_{near}
- Collision checks are performed and q_{new} is added to the tree if there are no collisions

Bidirectional RRT

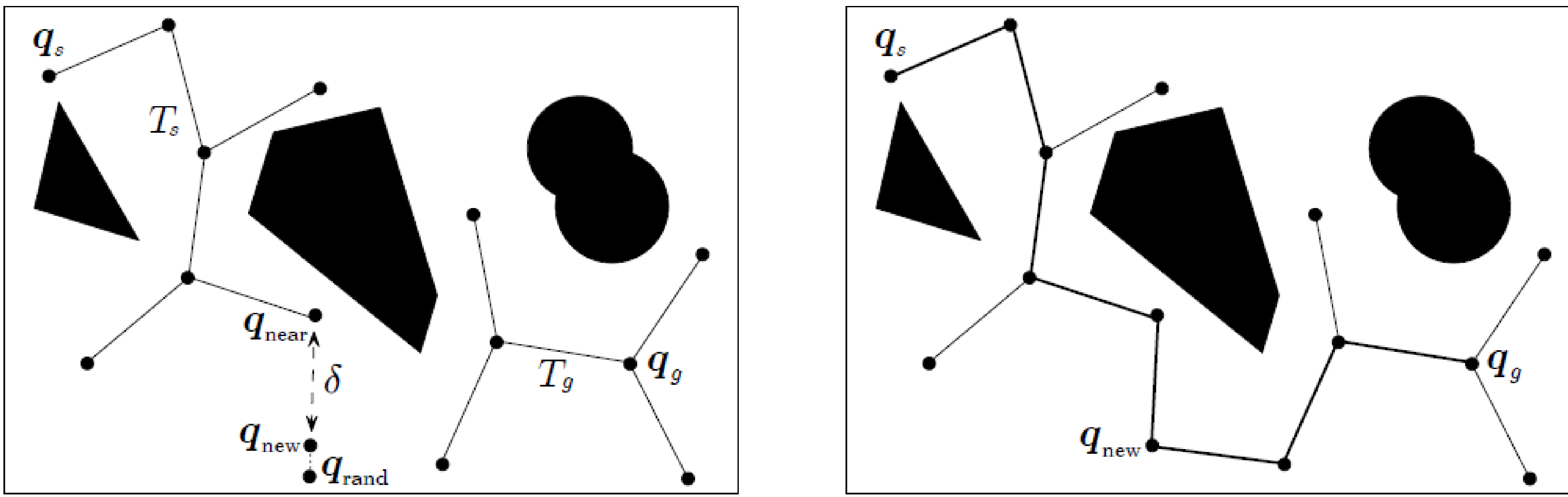


Fig. 12.10. The bidirectional RRT method in a two-dimensional configuration space *left*: the randomized mechanism for expanding a tree, *right*: the extension procedure for connecting the two trees

Bidirectional RRT

- The method quickly explores the configuration space
- In may be shown that the method is biased towards unexplored regions of the configuration space
- To speed up the search two trees are used, rooted in q_s and q_g
- After a certain number of iterations the method enters a phase where it tries to connect the two trees
- If this is unsuccessful, the exploration continues
- If it is successful a solution is found
- The method is probabilistic complete, as the PRM method

Bidirectional RRT for non-holonomic systems

- Generally the path found is not admissible for a non-holonomic mobile robot
- A simple yet general approach is to use motion primitives, i.e. a finite set of admissible local paths
- Each path is generated based on a specific choice of control inputs
- In the unicycle case the following inputs may be used

$$v = \bar{v} \qquad \omega = \{-\bar{\omega}, 0, \bar{\omega}\} \qquad t \in [0, \Delta] \tag{12.9}$$

- This results in three admissible paths

Bidirectional RRT for non-holonomic systems

- The new method is quite similar to the previously described method
- The difference is that once the near configuration q_{near} is identified on the graph the new configuration is generated by applying the motion primitives
- The new configurations must be checked for collisions before they are added to the tree

Bidirectional RRT for non-holonomic systems

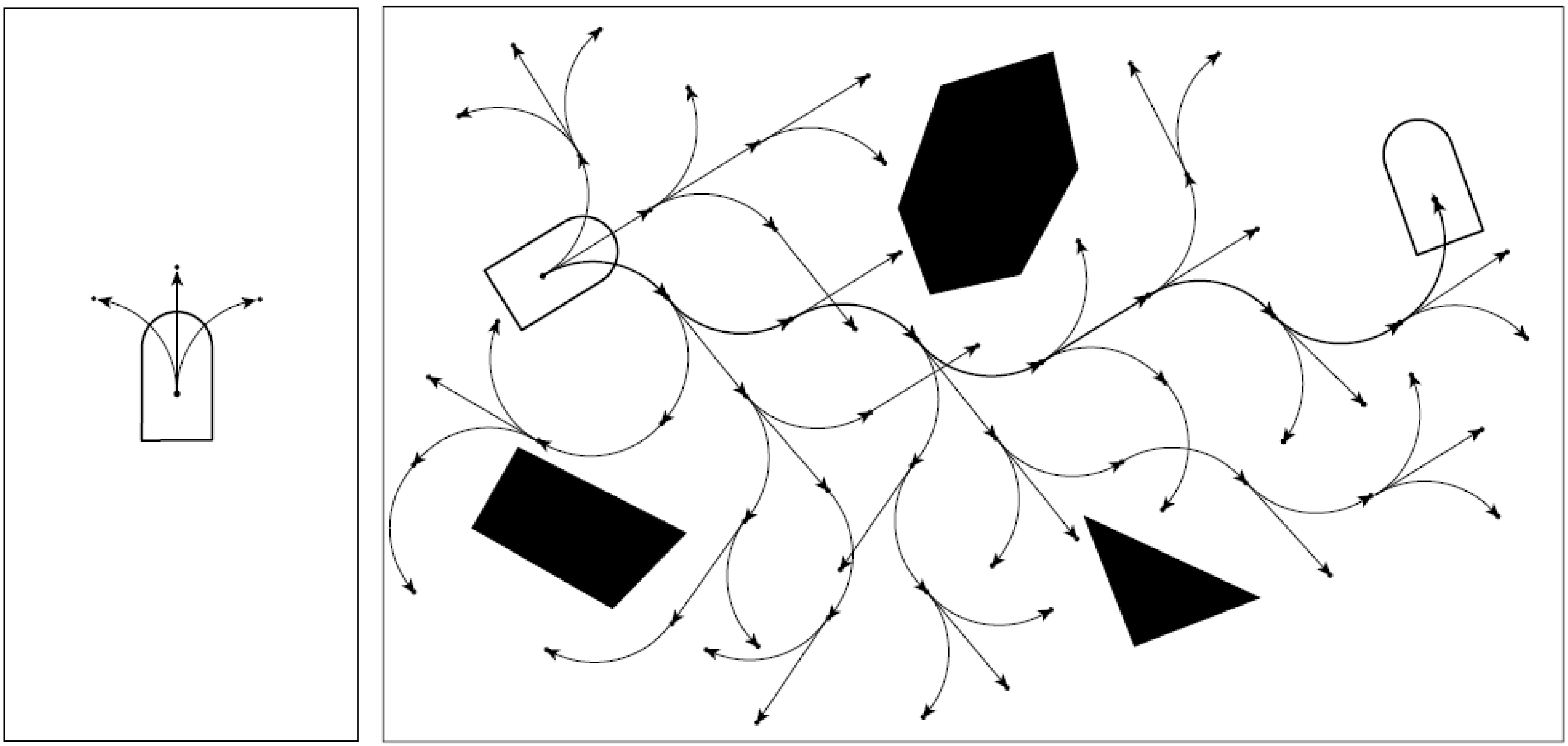


Fig. 12.11. RRT-based motion planning for a unicycle; *left*: a set of motion primitives, *right*: an example of RRT

12.6 Planning via artificial potentials

- All the previously presented methods are suitable for off-line motion planning, since they require a priori knowledge of the obstacles
- When a robot uses partial information of the environment, the computation of the motion planning problem must be done on-line
- An efficient on-line method is using artificial potential fields
- The robot moves according to a potential field consisting of
  - An attractive potential towards the goal
  - A repulsive potential away from the obstacles
- The artificial force generated by the field is the negative gradient of the field

12.6.1 Attractive potential

- The attractive potential is designed to guide the robot to the goal configuration
- One may use a parabolic function

$$U_{a1}(\boldsymbol{q}) = \frac{1}{2} k_a \boldsymbol{e}^T(\boldsymbol{q})\boldsymbol{e}(\boldsymbol{q}) = \frac{1}{2} k_a \|\boldsymbol{e}(\boldsymbol{q})\|^2, \qquad (12.10)$$

- Where e is the error
- The function is always positive and minimum in the goal configuration
- The attractive force is then

$$\boldsymbol{f}_{a1}(\boldsymbol{q}) = -\nabla U_{a1}(\boldsymbol{q}) = k_a \boldsymbol{e}(\boldsymbol{q}). \qquad (12.11)$$

Attractive potential

- Alternatively one can use a conical attractive potential

$$U_{a2}(\boldsymbol{q}) = k_a \|\boldsymbol{e}(\boldsymbol{q})\|. \tag{12.12}$$

- With the corresponding attractive force

$$\boldsymbol{f}_{a2}(\boldsymbol{q}) = -\nabla U_{a2}(\boldsymbol{q}) = k_a \frac{\boldsymbol{e}(\boldsymbol{q})}{\|\boldsymbol{e}(\boldsymbol{q})\|}, \tag{12.13}$$

- Which is constant. This is an advantage to get a bounded force when the error goes to infinity
- On the other hand the force is undefined for e = 0
- Possible to combine the two potentials, with a transition when $\|\boldsymbol{e}(\boldsymbol{q})\| = 1$

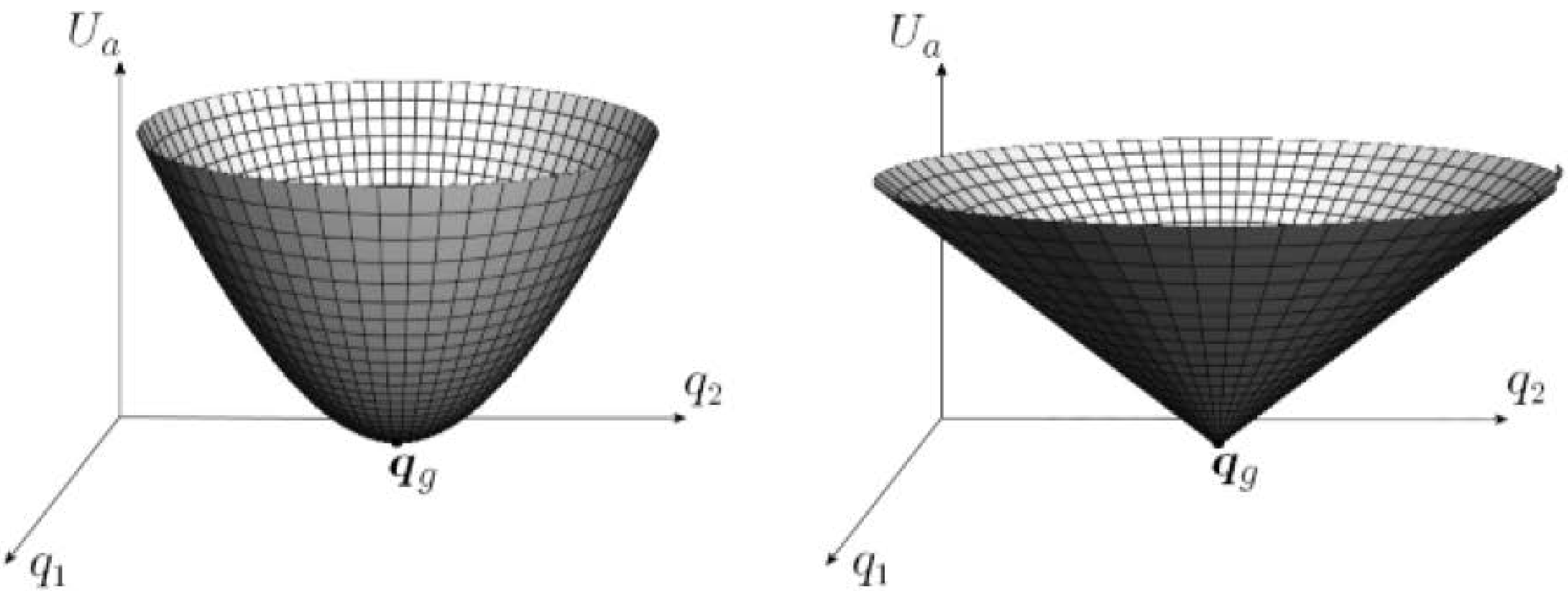


Fig. 12.12. The shape of the paraboloidal attractive potential U_{a1} (*left*) and of the conical attractive potential U_{a2} (*right*) in the case $\mathcal{C} = \mathbb{R}^2$, for $k_a = 1$

12.6.2 Repulsive potential

- The repulsive potential is added to prevent the robot colliding with obstacles
- The idea is to build a barrier potential in the vicinity of the obstacles
- Now assuming convex obstacles
- For each convex component $\mathcal{CO}_i$

$$U_{r,i}(\boldsymbol{q}) = \begin{cases} \dfrac{k_{r,i}}{\gamma}\left(\dfrac{1}{\eta_i(\boldsymbol{q})} - \dfrac{1}{\eta_{0,i}}\right)^{\gamma} & \text{if } \eta_i(\boldsymbol{q}) \le \eta_{0,i} \\ 0 & \text{if } \eta_i(\boldsymbol{q}) > \eta_{0,i}, \end{cases} \tag{12.14}$$

Repulsive potential

$$U_{r,i}(\boldsymbol{q}) = \begin{cases} \dfrac{k_{r,i}}{\gamma}\left(\dfrac{1}{\eta_i(\boldsymbol{q})} - \dfrac{1}{\eta_{0,i}}\right)^{\gamma} & \text{if } \eta_i(\boldsymbol{q}) \le \eta_{0,i} \\ 0 & \text{if } \eta_i(\boldsymbol{q}) > \eta_{0,i}, \end{cases} \tag{12.14}$$

- Where
  - $\eta_i(\boldsymbol{q})$ is the distance from the obstacle
  - $\eta_{0,i}$ is the range of influence
  - $\gamma = 2, 3$
- The potential is zero outside the range of influence
- The potential tends to infinity as the boundary of $\mathcal{CO}_i$ is approached

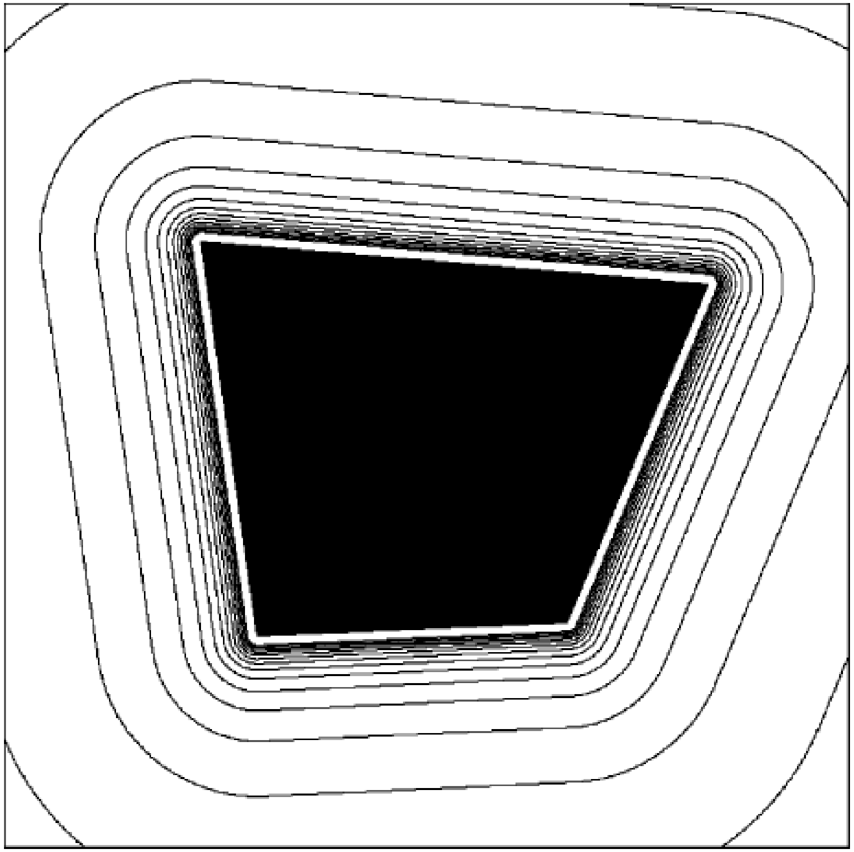


Fig. 12.13. The equipotential contours of the repulsive potential U_r in the range of influence of a polygonal $\mathcal{C}$-obstacle in $\mathcal{C} = \mathbb{R}^2$, for $k_r = 1$ and $\gamma = 2$

Repulsive potential

- The repulsive force is

$$\boldsymbol{f}_{r,i}(\boldsymbol{q}) = -\nabla U_{r,i}(\boldsymbol{q}) = \begin{cases} \dfrac{k_{r,i}}{\eta_i^2(\boldsymbol{q})}\left(\dfrac{1}{\eta_i(\boldsymbol{q})} - \dfrac{1}{\eta_{0,i}}\right)^{\gamma-1} \nabla \eta_i(\boldsymbol{q}) & \text{if } \eta_i(\boldsymbol{q}) \le \eta_{0,i} \\ 0 & \text{if } \eta_i(\boldsymbol{q}) > \eta_{0,i}. \end{cases} \tag{12.15}$$

- It is orthogonal to the contours of the potential

12.6.3 Total potential

- The total potential is obtained by superposition

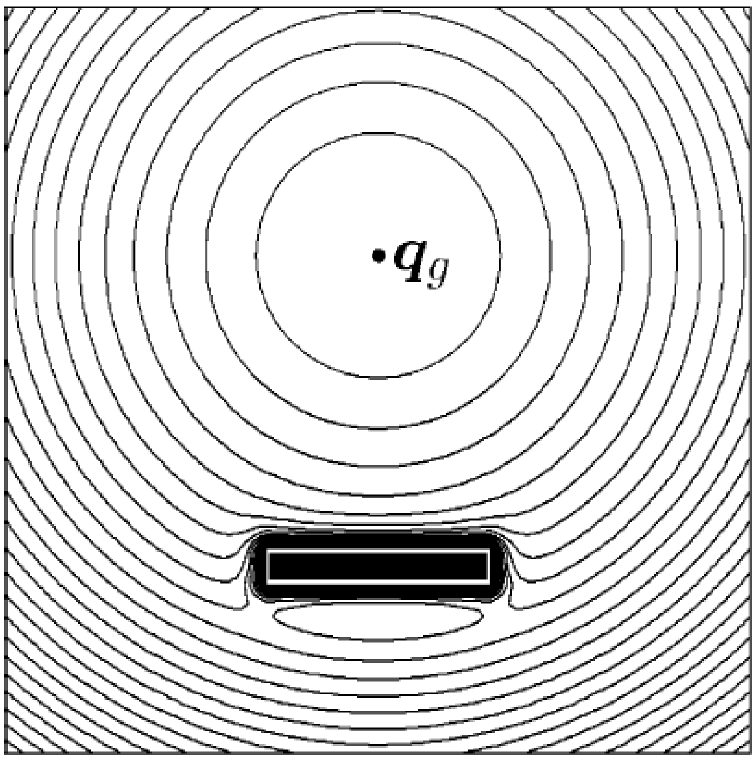
$$U_t(\boldsymbol{q}) = U_a(\boldsymbol{q}) + U_r(\boldsymbol{q}). \tag{12.17}$$

- The resulting force field is

$$\boldsymbol{f}_t(\boldsymbol{q}) = -\nabla U_t(\boldsymbol{q}) = \boldsymbol{f}_a(\boldsymbol{q}) + \sum_{i=1}^{p} \boldsymbol{f}_{r,i}(\boldsymbol{q}). \tag{12.18}$$

- U_t has a global minimum in $\boldsymbol{q}_g$, but there may be also some local minima where the force field is zero
- If all the obstacles are spheres the are only saddle points, and no local minima

Total potential



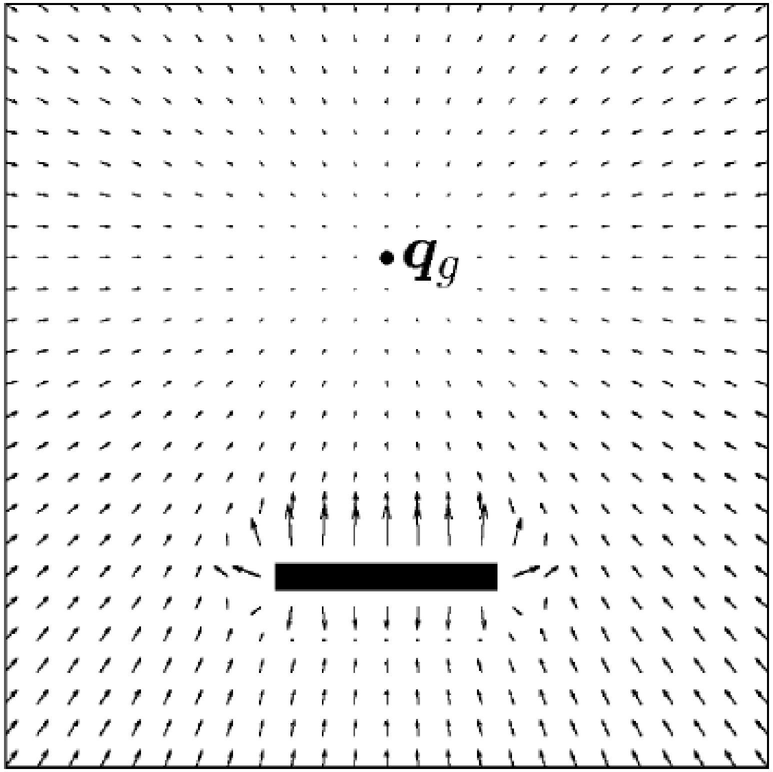


Fig. 12.14. The total potential in $\mathcal{C} = \mathbb{R}^2$ obtained by superposition of a hyperboloidic attractive potential and a repulsive potential for a rectangular $\mathcal{C}$-obstacle: *left*: the equipotential contours, *right*: the resulting force field

12.6.4 Planning techniques

- There are three different approaches for planning collision free motions on the basis of the artificial potential U_t

1. Force field represents generalized forces
$$\boldsymbol{\tau} = \boldsymbol{f}_t(\boldsymbol{q}), \tag{12.19}$$
2. The robot is a unit point mass
$$\ddot{\boldsymbol{q}} = \boldsymbol{f}_t(\boldsymbol{q}). \tag{12.20}$$
3. The force field is interpreted as desired velocity
$$\dot{\boldsymbol{q}} = \boldsymbol{f}_t(\boldsymbol{q}). \tag{12.21}$$

Planning techniques

- For on-line motion generation the force is evaluated at each time step, and the method is said to be *reactive* planning
- For off-line motion planning paths are generating by simulating the robot
- Generally (12.19) generates smoother paths, as the path is «filtered» through the robot dynamics
- On the other hand (12.21) is faster in generating motion corrections, and can therefore be considered safer
- (12.20) is an intermediate scheme
- (12.21) guaranties asymptotic stability (assuming no local minima)
- This is not true for the other two methods, they require a damping term

Lecture summary

- The canonical problem
- Configuration space
- Planning via retraction
- Probabilistic planning
  - Probabilistic roadmap (PRM)
  - Bidirectional Rapid-exploring Random Tree (Bidirectional RRT)
- Planning via artificial potentials