12

Motion Planning

Solution to Problem 12.1

The configuration of the mobile manipulator is $\boldsymbol{q} = [\, x \quad y \quad \theta_0 \quad \theta_1 \quad \ldots \quad \theta_6 \,]^T$, where (x, y) are the Cartesian coordinates of the contact point of the wheel with the ground (equivalently, of the wheel centre), θ_0 is the orientation of the unicycle with respect to the x axis, and $\theta_1, \ldots, \theta_6$ are the manipulator joint variables. The configuration space is therefore $\mathcal{C} = \mathbb{R}^2 \times SO(2) \times \ldots \times SO(2)$ (with $SO(2)$ appearing 7 times), and has dimension 9.

Solution to Problem 12.2

Assume that the configuration $\boldsymbol{q}$ takes values in the subset $\mathcal{Q}$ of Fig. 12.1 — this can be obtained by computing the joint variables q_1 and q_2 modulo 2π). Given two configurations $\boldsymbol{q}_A = (q_{1,A}, q_{2,A})$, $\boldsymbol{q}_B = (q_{1,B}, q_{2,B})$ in $\mathcal{Q}$, define

$$\begin{aligned} \Delta_1 &= \min(|q_{1,A} - q_{1,B}|, 2\pi - |q_{1,A} - q_{1,B}|) \\ \Delta_2 &= \min(|q_{2,A} - q_{2,B}|, 2\pi - |q_{2,A} - q_{2,B}|) \end{aligned}$$

and let

$$d_3(\boldsymbol{q}_A, \boldsymbol{q}_B) = \sqrt{\Delta_1^2 + \Delta_2^2}.$$

This definition of configuration space distance clearly satisfies the requirement of the problem.

It can be easily shown that $d_3(\boldsymbol{q}_A, \boldsymbol{q}_B)$ coincides with $d_2(\boldsymbol{q}_A, \boldsymbol{q}_B)$ given by eq. (12.2) whenever the Euclidean distance between $\boldsymbol{q}_A$ and $\boldsymbol{q}_B$ is not larger than $\sqrt{2}\,\pi$.

Solution to Problem 12.3

A possible solution is shown in Fig. S12.1. In the first scene, the robot is triangular and there is a single square obstacle, while in the second the robot

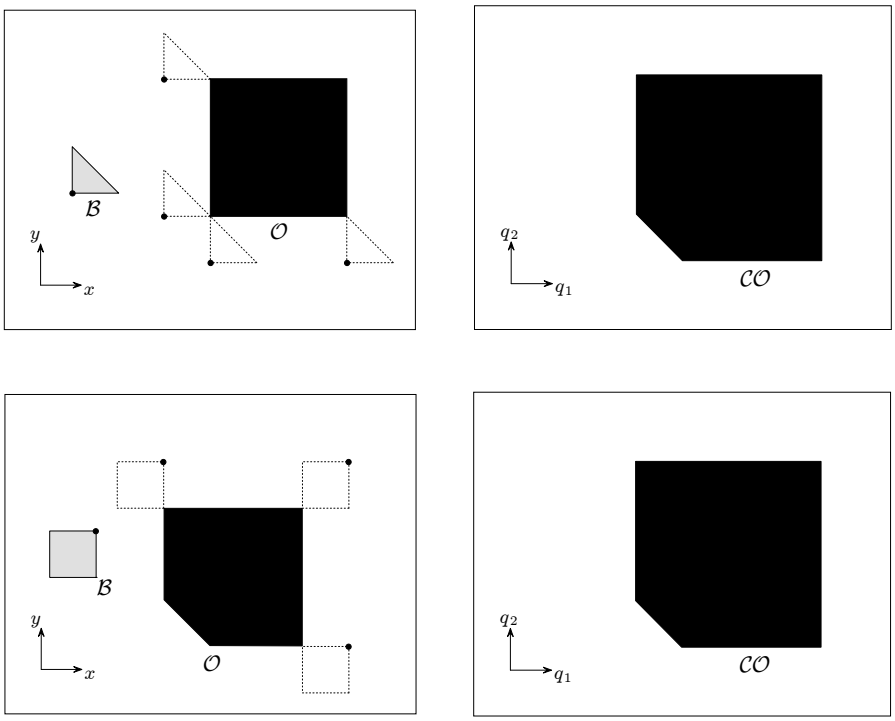

Fig. S12.1. Two different scenes (above/below) that result in the same $\mathcal{C}$-obstacle region. For each scene, *left*: the robot $\mathcal{B}$, the obstacle $\mathcal{O}$ and the growing procedure for building $\mathcal{C}$-obstacles, *right*: the configuration space $\mathcal{C}$ and the $\mathcal{C}$-obstacle $\mathcal{CO}$

is square and the obstacle is a pentagon. Under the assumption that the robots can freely translate without changing their orientation, the $\mathcal{C}$-obstacle region is exactly the same for the two scenes.

Solution to Problem 12.4

The problem can be solved by visual inspection of Fig. 12.4. For example, the following configurations of the 2R manipulator lie in the three disjoint components of $\mathcal{C}_{\text{free}}$:

$$\boldsymbol{q}_a = \begin{bmatrix} \pi/4 \\ \pi/2 \end{bmatrix} \quad \boldsymbol{q}_b = \begin{bmatrix} 4.12 \\ \pi/6 \end{bmatrix} \quad \boldsymbol{q}_c = \begin{bmatrix} 4.12 \\ 2.88 \end{bmatrix} \quad \text{[rad]}.$$

The manipulator posture and the position in $\mathcal{C}$ for each of these configurations is shown in Fig. S12.2.

Solution to Problem 12.5

Consider the lists $\boldsymbol{V} = \{V_1, \ldots, V_v\}$ and $\boldsymbol{S} = \{S_1, \ldots, S_s\}$ of all the vertices and sides of the given limited polygonal subset $\mathcal{K}$ of $\mathbb{R}^2$. A rough sketch of

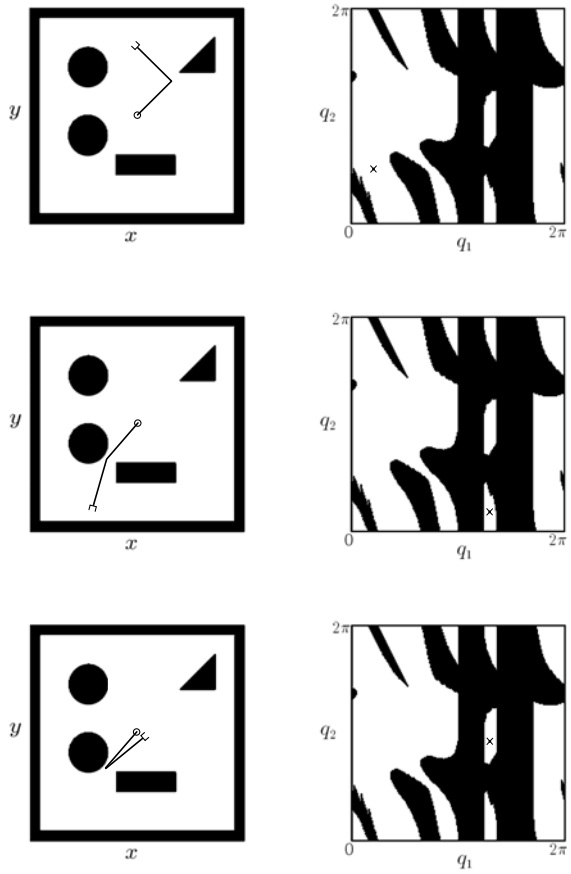

Fig. S12.2. Three configurations of the 2R manipulator that lie in disjoint components of $\mathcal{C}_{\text{free}}$: $\boldsymbol{q}_a$ (top), $\boldsymbol{q}_b$ (centre) and $\boldsymbol{q}_c$ (bottom). For each of them, *left*: the corresponding manipulator posture, *right*: the position in $\mathcal{C}$

a naive algorithm for computing the generalized Voronoi diagram of $\mathcal{K}$ is the following.

1. Build all the equidistance curves as follows:
   - 1a. For each vertex-vertex pair (V_i, V_j), derive the equation of the line $L_{V_iV_j}$ passing through the midpoint of the segment V_iV_j and orthogonal to the segment itself.
   - 1b. For each vertex-side pair (V_i, S_j), denote by L_{S_j} the line containing S_j, and derive the equation of the parabola $P_{V_iS_j}$ having V_i as focus and L_{S_j} as directrix.
   - 1c. For each side-side pair (S_i, S_j), denote by L_{S_i} and L_{S_j} the lines containing S_i and S_j, respectively, and derive the equation of the bisectrix $L_{S_iS_j}$ of the angle formed by L_{S_i} and L_{S_j} which contains S_i and S_j.

2. For each possible pair of equidistance curves, compute the intersection points — at most one in the case of a line/line pair, at most two for a line/parabola or parabola/parabola pair. Each of these points is characterized by the two distances from the features (vertex/vertex, line/line or line/vertex) that generate the two intersecting equidistance curves.
3. Discard all intersection points that do not belong to $\mathcal{K}$.
4. Discard all intersection points for which the two characteristic distances are not equal.
5. For each remaining intersection point, compute its clearance, and discard all intersection points for which the clearance is not equal to the characteristic distance.
6. All the remaining intersection points are nodes of the generalized Voronoi diagram. Its arcs are the portions of the equidistance curves that are enclosed by these nodes.

Solution to Problem 12.6

Refer to the example shown in Fig. 12.7, and assume that the goal configuration $\boldsymbol{q}_g$ is moved somewhere in cell c_6, so that $c_s = c_3$ and $c_g = c_6$. The shortest channel joining c_s to c_g is clearly $\{c_3, c_2, c_6\}$. The path extraction procedure described in Sect. 12.4.1 would produce a broken line with the following 4 vertices: $\boldsymbol{q}_s$, the midpoint of the common boundary between c_3 and c_2, the midpoint of the common boundary between c_2 and c_6, and finally $\boldsymbol{q}_g$. Hence, the second of the 3 edges forming the path would lie completely on the boundary of cell c_2, and in particular would go through the leftmost vertex of the obstacle.

To solve this problem, the path extraction procedure can be modified by including additional vertices in the path. For example, in addition to $\boldsymbol{q}_s$, $\boldsymbol{q}_g$ and the midpoints of the common boundaries in the channel, one may include the centroids (or a generic internal point) of all the cells crossed by the channel, with the exception of c_s and c_g. In the above case, the resulting free path would be a broken line with 5 vertices: $\boldsymbol{q}_s$, the midpoint of the common boundary between c_3 and c_2, the centroid of cell c_2, the midpoint of the common boundary between c_2 and c_6, and finally $\boldsymbol{q}_g$.

Solution to Problem 12.7

The files with the solution can be found in Folder `12_7`; run `s12_7.m` to execute the program. The robot is a planar manipulator whose first two joints are prismatic (so that the base can move arbitrarily) followed by an arbitrary number of revolute joints. An admissible range can be specified for each joint variable; in particular, a fixed-base manipulator is specified by setting to zero the minimum and maximum values of the prismatic joint variables. The obstacles may be line segments, circles or polygons. The user must also input the start and goal configurations, the maximum number of configurations in

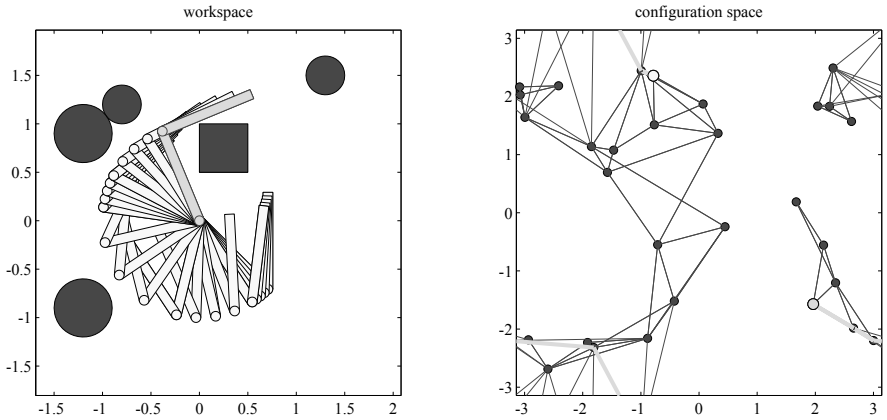

Fig. S12.3. Motion planning for a 2R planar manipulator via the PRM method; *left*: stroboscopic motion of the robot in the workspace, *right*: the PRM and the solution path (thick gray) in the configuration space

the roadmap, the maximum number of neighbours to which a new sample can be connected, and the maximum distance between neighbours.

The solution of a specific planning problem for a 2R fixed-base manipulator is reported in Fig. S12.3. The start and goal configurations are $\boldsymbol{q}_s = [-\pi/4 \quad 3\pi/4]^T$ and $\boldsymbol{q}_g = [5\pi/8 \quad -\pi/2]^T$ [rad,rad], respectively. Note that the configuration space, shown as a square in the figure, is correctly treated as a two-dimensional torus in the program. In particular, the configuration space distance is a generalization of the definition proposed in the solution to Problem 12.2.

Solution to Problem 12.8

The files with the solution can be found in Folder `12_8`; run `s12_8.m` to execute the program. The obstacles may be line segments, circles or polygons. The user must also input the start and goal configurations, the motion primitives (the discrete set of admissible constant values of the driving and steering velocities within the time interval Δ), the value of Δ and the maximum number of configurations in the roadmap.

Since the unicycle cannot reach arbitrary configurations in $\mathcal{C}_{\text{free}}$ with this planner, the program terminates whenever a configuration is reached that is sufficiently close to the desired goal; to this end, a user-selectable distance threshold is used. Moreover, to bias the search towards the goal, $\boldsymbol{q}_{\text{rand}}$ may be set to $\boldsymbol{q}_{\text{goal}}$ — rather than to a random configuration — with a probability that can be chosen arbitrarily.

The solution of a specific problem is reported in Fig. S12.4. The start and goal configurations are $\boldsymbol{q}_s = [-3 \quad 0 \quad 0]^T$ and $\boldsymbol{q}_g = [3 \quad -3.5 \quad 0]^T$ [m,m,rad], respectively. Note that the unicycle does not exactly reach $\boldsymbol{q}_g$, shown in gray

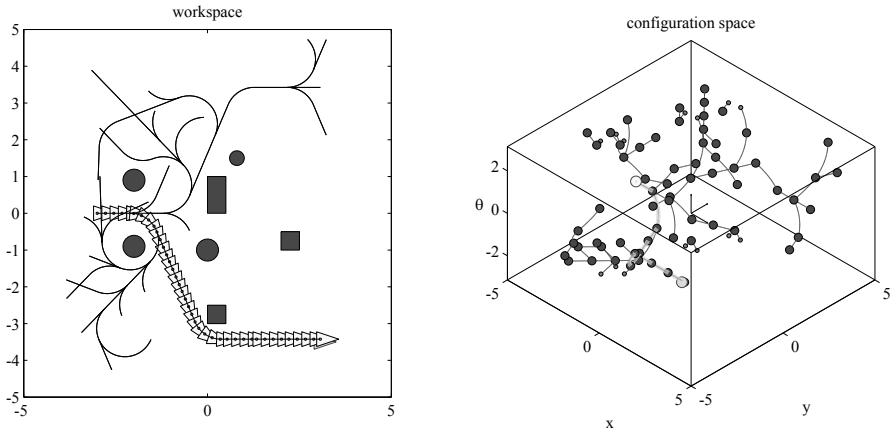

Fig. S12.4. Motion planning for a unicycle via the RRT method; *left*: stroboscopic motion of the robot and projection of the RRT on the workspace, *right*: the RRT and the solution path (thick gray) in the configuration space

in the stroboscopic plot. The configuration space, shown as a parallelepiped in the figure, is correctly treated as $\mathbb{R}^2 \times SO(2)$ in the program. In particular, the configuration space distance is an adaptation of the definition proposed in the solution to Problem 12.2.

Solution to Problem 12.9

Let

$$U_a(\boldsymbol{q}) = \begin{cases} \dfrac{1}{2}\, k_a \|\boldsymbol{e}(\boldsymbol{q})\|^2 & \text{if} \quad \|\boldsymbol{e}(\boldsymbol{q})\| \leq \rho \\ k_b \|\boldsymbol{e}(\boldsymbol{q})\| & \text{if} \quad \|\boldsymbol{e}(\boldsymbol{q})\| > \rho. \end{cases}$$

Continuity of the attractive force at the transition radius ρ is guaranteed by imposing

$$k_a \boldsymbol{e}(\boldsymbol{q}) = k_b \, \frac{\boldsymbol{e}(\boldsymbol{q})}{\|\boldsymbol{e}(\boldsymbol{q})\|} \qquad \text{for} \quad \|\boldsymbol{e}(\boldsymbol{q})\| = \rho,$$

i.e., $k_b = \rho \, k_a$.

Solution to Problem 12.10

Consider Fig. S12.5, and in particular the half-line that originates at $\boldsymbol{q}_g$ and passes through the centre of the obstacle. Visualize a point that travels along this half-line, coming from the point at infinity, where the total potential is infinity — because of the attractive potential. The total potential goes back to infinity when the point reaches the boundary of the obstacle — because of the repulsive potential. Therefore, there is a point on the line where U has a local minimum; call this point P.

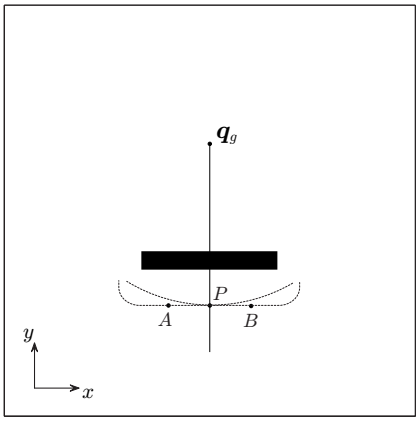

Fig. S12.5. The emergence of local minima due to the superposition of attractive and repulsive potentials

Now consider points A and B that are located along the same equipotential contour of the repulsive potential U_r as P. Since the attractive potential is higher at A and B than at C, P is a local minimum also in the direction of the line AB. In formulæ:

$$\left.\frac{\partial U}{\partial x}\right|_P = 0 \qquad \left.\frac{\partial U}{\partial y}\right|_P = 0,$$

which imply that P is a stationary point for U, i.e., the gradient of U is zero at P. To show that P is indeed a local minimum, one should consider the Hessian matrix of U

$$\begin{bmatrix} \dfrac{\partial^2 U}{\partial x^2} & \dfrac{\partial U}{\partial x \partial y} \\ \dfrac{\partial U}{\partial y \partial x} & \dfrac{\partial^2 U}{\partial y^2} \end{bmatrix}$$

and prove that it is positive definite in P. This can be verified by using the analytic expression of U. Note that the elements of the diagonal are certainly positive in P in view of the above arguments.

Solution to Problem 12.11

Denote by (x, y) the Cartesian coordinates of the robot, by (x_i, y_i) the Cartesian coordinates of the centre C_i of the i-th circular obstacle, and by ρ_i its radius. The attractive potential is

$$U_a(x, y) = \frac{1}{2}\, k_a (x^2 + y^2),$$

while the repulsive potentials ($\gamma = 2$) are

$$U_{r,i}(x,y) = \begin{cases} \dfrac{k_{r,i}}{2}\left(\dfrac{1}{\eta_i(x,y)} - \dfrac{1}{\eta_{0,i}}\right)^2 & \text{if } \eta_i(x,y) \le \eta_{0,i} \\ 0 & \text{if } \eta_i(x,y) > \eta_{0,i}, \end{cases}$$

for $i = 1, 2, 3$, with $\eta_i(x,y) = \sqrt{(x-x_i)^2 + (y-y_i)^2} - \rho_i$. The total potential is

$$U_t(x,y) = U_a(x,y) + \sum_{i=1}^{3} U_{r,i}(x,y),$$

while the total force is

$$\boldsymbol{f}(x,y) = \boldsymbol{f}_a(x,y) + \sum_{i=1}^{3} \boldsymbol{f}_{r,i}(x,y)$$

with

$$\boldsymbol{f}_a(x,y) = -k_a \begin{bmatrix} x \\ y \end{bmatrix}$$

and

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

In view of the above expression for $\eta_i(x,y)$, one has

$$\nabla\eta_i(x,y) = \begin{bmatrix} \dfrac{x-x_i}{\sqrt{(x-x_i)^2 + (y-y_i)^2}} \\ \dfrac{y-y_i}{\sqrt{(x-x_i)^2 + (y-y_i)^2}} \end{bmatrix}.$$

The actual numerical expressions depend on the choice of the constants k_a and $k_{r,i}$, for $i = 1, 2, 3$, as well as of the obstacle range of influence $\eta_{0,i}$.

Assuming that the ranges of influence of the obstacles do not overlap, the saddle points of the total potential will be clearly located along the lines that join the origin of the reference frame (the goal) with the centres of the circular obstacles, in the 'shadow zone' of each obstacle. In particular, the saddle points are found by solving the equations

$$-k_a \begin{bmatrix} x \\ y \end{bmatrix} = \frac{k_{r,i}}{\eta_i^2(x,y)}\left(\frac{1}{\eta_i(x,y)} - \frac{1}{\eta_{0,i}}\right)\nabla\eta_i(x,y) \qquad i = 1, 2, 3,$$

each representing the balance between the attractive force and the i-th repulsive force.

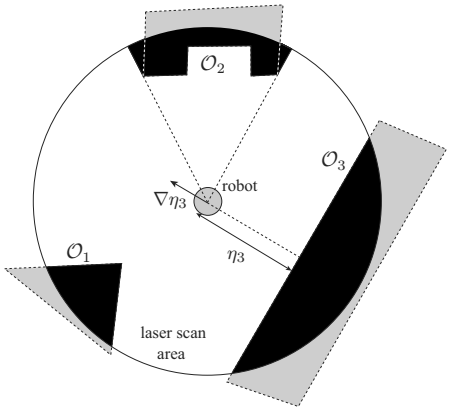

Fig. S12.6. On-line planning based on artificial potential fields for a circular mobile robot equipped with a rotating laser range finder.

Solution to Problem 12.12

In principle, there are two main approaches for on-line motion planning based on sensor data.

- The first approach is to store the data coming from the sensors in a map — i.e., a representation of the portion of the environment that the robot has visited so far — and use it to plan in an incremental fashion (deliberative planning).
- The second approach is to react only to the obstacles that are currently perceived of the robot, without any form of memory (reactive planning).

The artificial potential field technique naturally lends itself to a reactive implementation, as briefly discussed below.

Refer to the situation shown in Fig. S12.6. The workspace obstacles $\mathcal{O}_1$, $\mathcal{O}_2$ and $\mathcal{O}_3$ are depicted in black for the portion falling inside the laser scan area, and in gray outside. Note how the perceived obstacle is larger than the actual obstacle for $\mathcal{O}_2$, due to the fact that the laser range finder is a visibility-based sensor.

Consider first the case of a point robot. Since the goal is known, the attractive potential can be built as in the off-line case — this is true provided that the robot is equipped with a localization module. To build the repulsive potential, it is reasonable to define the range of influence $\eta_{0,i}$ of the obstacles to be smaller than the maximum measurable range R — if the robot does not detect the obstacle, it does not react to it. A typical choice is $\eta_{0,i} = R$, for all i. As seen in Sect. 12.6.2, the perceived obstacle region must be decomposed in convex components to guarantee continuity of the repulsive potential field; in the case of Fig. S12.6, only $\mathcal{O}_2$ needs to undergo this procedure. At this point, the repulsive force produced by each convex component is computed as

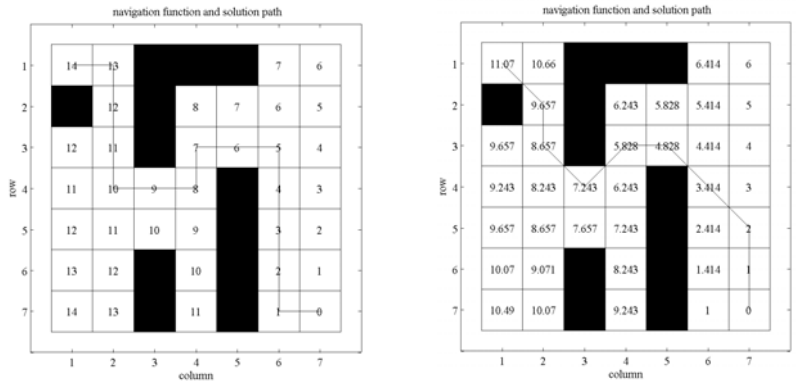

Fig. S12.7. The assigned occupancy grid, the navigation function and the solution path from cell $(1, 1)$ to cell $(7, 7)$; *left*: using 1-adjacency, *right*: using 2-adjacency

in (12.15), with η_i, $\nabla\eta_i$ directly obtained from the range scan profile — see the example in Fig. S12.6.

In the case of a robot with a finite nonzero radius ρ, one simply uses $\eta_i - \rho$ in place of η_i in (12.15), and obtains $\nabla\eta_i$ as in the point robot case.

It is easy to realize that this reactive version of the artificial potential method behaves exactly as the off-line version, in the sense that the resulting motion of the robot is the same. Clearly, this is true as long as the environment is continuously scanned during the motion and the above procedure for computing the potential field can be efficiently implemented, so as to be executed within the duration of the motion control cycle.

Solution to Problem 12.13

The files with the solution can be found in Folder `12_13`. The user is required to input the binary gridmap representing the configuration space, the start and goal cells, and the adjacency type (1- or 2-adjacency). The cells are assumed to be squares with unit side. During the expansion of the wavefront that originates at $\boldsymbol{q}_g$, the increase of the navigation function between adjacent cells (cost-to-go) is set equal to the Euclidean distance between the cell centres. The solution path is then found by following the steepest descent from the goal cell.

For illustration, Fig. S12.7 shows the results obtained on a (7×7) gridmap, with $(1, 1)$ as the start cell and $(7, 7)$ as the goal cell. The path found using 2-adjacency is shorter as expected. It is easy to realize that even when the solution paths are in the same homotopy class — i.e., one can be continuously deformed into the other — the 1-adjacency path may be longer than the 2-adjacency path by a factor of as much as $\sqrt{2}$. On the other hand, 2-adjacency produces paths that may graze the $\mathcal{C}$-obstacles; as a consequence, it is recommended that these are slightly enlarged before planning.