

near $\pi/2$), for some values of θ_2 the second link of the arm collides with the obstacle. The region $\mathcal{Q}\mathcal{O}$ shown in Figure 5.2(b) was computed using a discrete grid on the configuration space. For each cell in the grid, a collision test was performed, and the cell was shaded when a collision occurred. This is only an approximate representation of $\mathcal{Q}\mathcal{O}$; however, for robots with revolute joints, exact representations are very expensive to compute, and therefore such approximate representations are often used for robots with a few degrees of freedom.

◊

Computing $\mathcal{Q}\mathcal{O}$ for the two-dimensional case of $\mathcal{Q} = \mathbb{R}^2$ and polygonal obstacles is straightforward, but, as can be seen from the two-link planar arm example, computing $\mathcal{Q}\mathcal{O}$ becomes difficult for even moderately complex configuration spaces. In the general case (for example, articulated arms or rigid bodies that can both translate and rotate), the problem of computing a representation of the configuration space obstacle region is intractable. One of the reasons for this complexity is that the size of the representation of the configuration space tends to grow exponentially with the number of degrees of freedom. This is easy to understand intuitively by considering the number of n -dimensional unit cubes needed to fill a space of size k . For the one-dimensional case k unit intervals will cover the space. For the two-dimensional case k^2 squares are required. For the three-dimensional case k^3 cubes are required, and so on. Therefore, in this chapter we will develop methods that avoid the construction of an explicit representation of $\mathcal{Q}\mathcal{O}$ or of $\mathcal{Q}_{\text{free}}$.

The path planning problem is to find a path from an initial configuration q_s to a final configuration q_f , such that the robot does not collide with any obstacle as it traverses the path. More formally, a collision-free path from q_s to q_f is a continuous map, $\gamma : [0, 1] \rightarrow \mathcal{Q}_{\text{free}}$, with $\gamma(0) = q_s$ and $\gamma(1) = q_f$. We will develop path planning methods that compute a sequence of discrete configurations (set points) in the configuration space. In Section 5.5 we will show how smooth trajectories can be generated from such a sequence of set points.

5.2 PATH PLANNING USING POTENTIAL FIELDS

As mentioned above, it is typically not feasible to build an explicit representation of $\mathcal{Q}\mathcal{O}$ or of $\mathcal{Q}_{\text{free}}$. An alternative is to develop a search algorithm that incrementally explores $\mathcal{Q}_{\text{free}}$ while searching for a path. One of the most popular strategies for exploring $\mathcal{Q}_{\text{free}}$ uses an **artificial potential field** to guide the search.

The basic idea behind the potential field approach is to treat the robot as a point particle in the configuration space under the influence of an artificial potential field U . The field U is constructed so that the robot is attracted to the final configuration q_f while being repelled from the boundaries of $\mathcal{Q}\mathcal{O}$. If possible, U is constructed so that there is a single global minimum of U at q_f and there are no local minima. Unfortunately it is typically difficult or even impossible to construct such a field.

In general, the field U is an additive field consisting of one component that attracts the robot to q_f and a second component that repels the robot from the boundary of $\mathcal{Q}\mathcal{O}$

$$U(q) = U_{\text{att}}(q) + U_{\text{rep}}(q)$$

Given this formulation path planning can be treated as an optimization problem, that is, the problem of finding the global minimum in U starting from initial configuration q_s . One of the simplest algorithms to solve this problem is gradient descent. In this case, the negative gradient of U can be considered as a generalized force acting on the robot in configuration space

$$\tau(q) = -\nabla U(q) = -\nabla U_{\text{att}}(q) - \nabla U_{\text{rep}}(q)$$

in which τ is a vector of joint torques (for a revolute arm). Allowing this force to act on the robot will cause it to move toward its goal configuration along the path of steepest descent of the potential function.

In general, it is difficult to construct a potential field directly on the configuration space, and even more difficult to compute the gradient of the field on the configuration space. The reasons for this include the difficulty of computing shortest distances to configuration space obstacles (a computation that is required when computing the value of the repulsive field, as we will see below) and the complex geometry of the configuration space. For this reason, we will define our potential fields directly on the workspace of the robot. In particular, for an n -link arm, we will define a potential field for each of the origins of the n DH frames (excluding the fixed, frame 0). These **workspace potential fields** will attract the origins of the DH frames to their goal locations while repelling them from obstacles. We will use these fields to define motions in the configuration space using the manipulator Jacobian matrix.

In the remainder of this section we will describe typical choices for the attractive and repulsive potential fields, how the manipulator Jacobian can be used to map these fields to configuration space motions, and a gradient descent algorithm that can be used to plan paths in this field.

5.2.1 The Attractive Field

To attract the robot to its goal configuration, we will define an attractive potential field $U_{\text{att},i}$ for o_i , the origin of the i^{th} DH frame. When all n origins reach their goal positions, the arm will have reached its goal configuration.

There are several criteria that the potential field $U_{\text{att},i}$ should satisfy. First, $U_{\text{att},i}$ should be monotonically increasing with the distance to o_i from its goal position. The simplest choice for such a field is a field that grows linearly with this distance, a so-called **conic well potential**. If we denote the position of the origin of the i^{th} DH frame by $o_i(q)$, then the conic well potential is given by

$$U_{\text{att},i}(q) = \|o_i(q) - o_i(q_f)\|$$

The gradient of such a field has unit magnitude everywhere but at the goal position where it is zero. This can lead to stability problems since there is a discontinuity in the attractive force at the goal position. We prefer a field that is continuously differentiable such that the attractive force decreases as o_i approaches its goal position. The simplest such field is a field that grows quadratically with distance

$$U_{\text{att},i}(q) = \frac{1}{2}\zeta_i\|o_i(q) - o_i(q_f)\|^2 \quad (5.1)$$

in which ζ_i is a parameter used to scale the effects of the attractive potential. This field is sometimes referred to as a **parabolic well potential**. The workspace attractive force for o_i is equal to the negative gradient of $U_{\text{att},i}$, which is given by (Problem 5-7)

$$F_{\text{att},i}(q) = -\nabla U_{\text{att},i}(q) = -\zeta(o_i(q) - o_i(q_f)) \quad (5.2)$$

For the parabolic well the attractive force for the origin of the i^{th} DH frame is a vector directed toward $o_i(q_f)$ with magnitude linearly related to the distance to $o_i(q_f)$ from $o_i(q)$.

Note that while this force converges linearly to zero as q approaches q_f , which is a desirable property, it grows without bound as q moves away from q_f . If q_s is very far from q_f , this may produce an initial attractive force that is very large. For this reason we may choose to combine the quadratic and conic potentials so that the conic potential attracts o_i when it is very distant from its goal position, and the quadratic potential attracts o_i when it is near its goal position. Of course it is necessary that the gradient be defined at the boundary between the conic and quadratic fields. Such a field can be

5.2. PATH PLANNING USING POTENTIAL FIELDS

defined by

$$U_{\text{att},i}(q) = \begin{cases} \frac{1}{2}\zeta_i\|o_i(q) - o_i(q_f)\|^2 & ; \|o_i(q) - o_i(q_f)\| \leq d \\ d\zeta_i\|o_i(q) - o_i(q_f)\| - \frac{1}{2}\zeta_id^2 & ; \|o_i(q) - o_i(q_f)\| > d \end{cases} \quad (5.3)$$

in which d is the distance that defines the transition from conic to parabolic well. In this case the workspace force for o_i is given by

$$F_{\text{att},i}(q) = \begin{cases} -\zeta_i(o_i(q) - o_i(q_f)) & : \|o_i(q) - o_i(q_f)\| \leq d \\ -d\zeta_i\frac{(o_i(q) - o_i(q_f))}{\|o_i(q) - o_i(q_f)\|} & : \|o_i(q) - o_i(q_f)\| > d \end{cases} \quad (5.4)$$

The gradient is well defined at the boundary of the two fields since at the boundary $d = \|o_i(q) - o_i(q_f)\|$ and the gradient of the quadratic potential is equal to the gradient of the conic potential $F_{\text{att},i}(q) = -\zeta_i(o_i(q) - o_i(q_f))$.

Example 5.3 Two-link Planar Arm

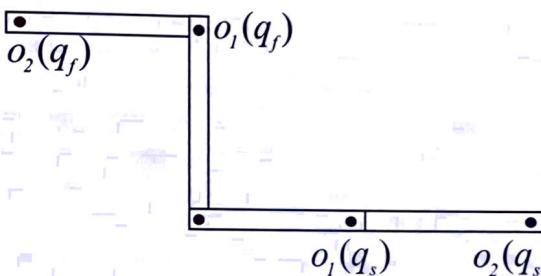


Figure 5.3: The initial configuration for the two-link arm is given by $\theta_1 = \theta_2 = 0$ and the final configuration is given by $\theta_1 = \theta_2 = \pi/2$. The origins for DH frames 1 and 2 are shown at both q_s and q_f .

Consider the two-link planar arm shown in Figure 5.3 with $a_1 = a_2 = 1$ and with initial and final configurations given by

$$q_s = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad q_f = \begin{bmatrix} \frac{\pi}{2} \\ \frac{\pi}{2} \end{bmatrix}$$

Using the forward kinematic equations for this arm (see Example 3.1) we obtain

$$o_1(q_s) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad o_1(q_f) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad o_2(q_s) = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \quad o_2(q_f) = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Using these coordinates for the origins of the two DH frames at their initial and goal configurations, assuming that d is sufficiently large, we obtain the attractive forces

$$\begin{aligned} F_{\text{att},1}(q_s) &= -\zeta_1(o_1(q_s) - o_1(q_f)) = \zeta_1 \begin{bmatrix} -1 \\ 1 \end{bmatrix} \\ F_{\text{att},2}(q_s) &= -\zeta_2(o_2(q_s) - o_2(q_f)) = \zeta_2 \begin{bmatrix} -3 \\ 1 \end{bmatrix} \end{aligned}$$

◊

5.2.2 The Repulsive Field

In order to prevent collisions between the robot and obstacles we will define a **workspace repulsive potential field** for the origin of each DH frame (excluding frame 0). There are several criteria that these repulsive fields should satisfy. They should repel the robot from obstacles, never allowing the robot to collide with an obstacle, and, when the robot is far away from an obstacle, that obstacle should exert little or no influence on the motion of the robot. One way to achieve this is to define a potential function whose value approaches infinity as the configuration approaches an obstacle boundary, and whose value decreases to zero at a specified distance from the obstacle boundary. Note that by defining repulsive potentials only for the origins of the DH frames we cannot ensure that collisions never occur (for example, the middle portion of a long link might collide with an obstacle), but it is fairly easy to modify the method to prevent such collisions as we will see below. For now, we will deal only with the origins of the DH frames.

We define ρ_0 to be the distance of influence of an obstacle. This means that an obstacle will not repel o_i if the distance from o_i to the obstacle is greater than ρ_0 . One potential function that meets the criteria described above is given by

$$U_{\text{rep},i}(q) = \begin{cases} \frac{1}{2}\eta_i \left(\frac{1}{\rho(o_i(q))} - \frac{1}{\rho_0} \right)^2 & ; \quad \rho(o_i(q)) \leq \rho_0 \\ 0 & ; \quad \rho(o_i(q)) > \rho_0 \end{cases} \quad (5.5)$$

5.2. PATH PLANNING USING POTENTIAL FIELDS

in which $\rho(o_i(q))$ is the shortest distance between o_i and any workspace obstacle. The workspace repulsive force is equal to the negative gradient of $U_{\text{rep},i}$. For $\rho(o_i(q)) \leq \rho_0$, this force is given by (Problem 5-11)

$$F_{\text{rep},i}(q) = \eta_i \left(\frac{1}{\rho(o_i(q))} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2(o_i(q))} \nabla \rho(o_i(q)) \quad (5.6)$$

in which the notation $\nabla \rho(o_i(q))$ indicates the gradient $\nabla \rho(x)$ evaluated at $x = o_i(q)$. If the obstacle region is convex and b is the point on the obstacle boundary that is closest to o_i , then $\rho(o_i(q)) = \|o_i(q) - b\|$, and its gradient is

$$\nabla \rho(x) \Big|_{x=o_i(q)} = \frac{o_i(q) - b}{\|o_i(q) - b\|} \quad (5.7)$$

that is, the unit vector directed from b toward $o_i(q)$.

Example 5.4 Two-link Planar Arm

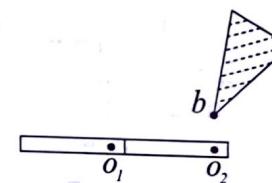


Figure 5.4: The obstacle shown repels o_2 , but is outside the distance of influence for o_1 . Therefore, it exerts no repulsive force on o_1 .

Consider the previous Example 5.3, with a single convex obstacle in the workspace as shown in Figure 5.4. Let $\rho_0 = 1$. This prevents the obstacle from repelling o_1 , which is reasonable since link 1 can never contact the obstacle. The nearest obstacle point to o_2 is the vertex b of the polygonal obstacle. Suppose that b has the coordinates $(2, 0.5)$. Then the distance from $o_2(q_s)$ to b is $\rho(o_2(q_s)) = 0.5$ and $\nabla \rho(o_2(q_s)) = [0, -1]^T$. The repulsive force at the initial configuration for o_2 is then given by

$$F_{\text{rep},2}(q_s) = \eta_2 \left(\frac{1}{0.5} - 1 \right) \frac{1}{0.25} \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \eta_2 \begin{bmatrix} 0 \\ -4 \end{bmatrix}$$

This force has no effect on joint 1, but causes joint 2 to rotate slightly in the clockwise direction, moving link 2 away from the obstacle.

◊

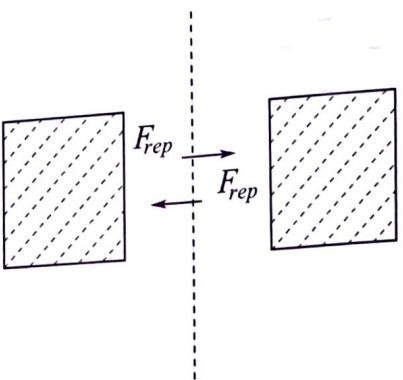


Figure 5.5: In this case the gradient of the repulsive potential given by Equation (5.6) is not continuous. In particular, the gradient changes discontinuously when o_i crosses the line midway between the two obstacles.

If the obstacle is not convex, then the distance function ρ will not necessarily be differentiable everywhere, which implies discontinuity in the force vector. Figure 5.5 illustrates such a case. Here the obstacle region is defined by two rectangular obstacles. For all configurations to the left of the dashed line the force vector points to the right, while for all configurations to the right of the dashed line the force vector points to the left. Thus, when o_i crosses the dashed line, a discontinuity in force occurs. There are various ways to deal with this problem. The simplest of these is merely to ensure that the regions of influence of distinct obstacles do not overlap.

As mentioned above, defining repulsive fields only for the origins of the DH frames does not guarantee that the robot cannot collide with an obstacle. Figure 5.6 shows an example where this is the case. In this figure o_1 and o_2 are very far from the obstacle and therefore the repulsive influence may not be great enough to prevent link 2 from colliding with the obstacle. To cope with this problem we can use a set of **floating repulsive control points** $o_{float,i}$ typically one per link. The floating control points are defined as points on the boundary of a link that are closest to any workspace obstacle. Obviously the choice of the $o_{float,i}$ depends on the configuration q . For the case shown in Figure 5.6, $o_{float,2}$ would be located near the center of link 2, thus repelling the robot from the obstacle. The repulsive force acting on $o_{float,i}$ is defined in the same way as for the other control points using Equation (5.6).

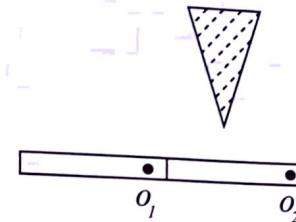


Figure 5.6: The repulsive forces exerted on the origins of the DH frames o_1 and o_2 may not be sufficient to prevent a collision between link 2 and the obstacle.

5.2.3 Mapping Workspace Forces to Joint Torques

We have shown how to construct potential fields in the robot's workspace that induce artificial forces on the origins o_i of the DH frames for the robot arm. In this section we describe how these forces can be mapped to joint torques.

As we derived in Chapter 4 using the principle of virtual work, if τ denotes the vector of joint torques induced by the workspace force F exerted at the end effector, then

$$J_v^T F = \tau \quad (5.8)$$

where J_v includes the top three rows of the manipulator Jacobian. We do not use the lower three rows, since we have considered only attractive and repulsive workspace forces, and not attractive and repulsive workspace torques. Note that for each o_i an appropriate Jacobian must be constructed, but this is straightforward given the techniques described in Chapter 4 and the A matrices for the arm. We denote the Jacobian for o_i by J_{o_i} .

Example 5.5 Two-link Planar Arm

Consider again the two-link arm of Example 5.3 with repulsive workspace forces as given in Example 5.4. The Jacobians that map joint velocities to linear velocities satisfy

$$\dot{o}_i = J_{o_i}(q) \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}$$

For the two-link arm the Jacobian matrix for o_2 is merely the Jacobian that we derived in Chapter 4, namely

$$J_{o_2}(q_1, q_2) = \begin{bmatrix} -s_1 - s_{12} & -s_{12} \\ c_1 + c_{12} & c_{12} \end{bmatrix} \quad (5.9)$$

The Jacobian matrix for o_1 is similar, but takes into account that motion of joint 2 does not affect the velocity of o_1 . Thus

$$J_{o_1}(q_1, q_2) = \begin{bmatrix} -s_1 & 0 \\ c_1 & 0 \end{bmatrix}$$

At $q_s = (0, 0)$ we have

$$J_{o_1}^T(q_s) = \begin{bmatrix} -s_1 & c_1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

and

$$J_{o_2}^T(q_s) = \begin{bmatrix} -s_1 - s_{12} & c_1 + c_{12} \\ -s_{12} & c_{12} \end{bmatrix} = \begin{bmatrix} 0 & 2 \\ 0 & 1 \end{bmatrix}$$

Using these Jacobians, we can easily map the workspace attractive and repulsive forces to joint torques. If we let $\zeta_1 = \zeta_2 = \eta_2 = 1$ we obtain

$$\tau_{\text{att},1}(q_s) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\tau_{\text{att},2}(q_s) = \begin{bmatrix} 0 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -3 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$\tau_{\text{rep},2}(q_s) = \begin{bmatrix} 0 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -4 \end{bmatrix} = \begin{bmatrix} -8 \\ -4 \end{bmatrix}$$

Example 5.6 A Polygonal Robot in the Plane

Artificial potential fields can also be used to plan the motions of a mobile robot moving in the plane. In this case the mobile robot is typically modeled as a polygon that can translate and rotate in the plane.

Consider the polygonal robot shown in Figure 5.7. The vertex a has coordinates (a_x, a_y) in the robot's local coordinate frame. Therefore, if the robot's configuration is given by $q = (x, y, \theta)$, the forward kinematic map for vertex a (that is, the mapping from $q = (x, y, \theta)$ to the global coordinates of the vertex a) is given by

$$a(x, y, \theta) = \begin{bmatrix} x + a_x \cos \theta - a_y \sin \theta \\ y + a_x \sin \theta + a_y \cos \theta \end{bmatrix} \quad (5.10)$$

The corresponding Jacobian matrix is given by

$$J_a(x, y, \theta) = \begin{bmatrix} 1 & 0 & -a_x \sin \theta - a_y \cos \theta \\ 0 & 1 & a_x \cos \theta - a_y \sin \theta \end{bmatrix}$$

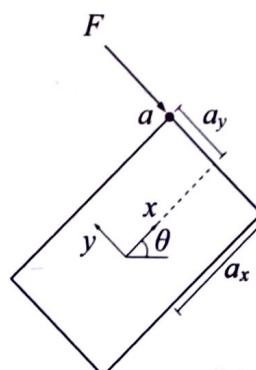


Figure 5.7: In this example, the robot is a polygon whose configuration can be represented as $q = (x, y, \theta)$, in which θ is the angle from the world x -axis to the x -axis of the robot's local frame. A force F is exerted on vertex a with local coordinates (a_x, a_y) .

Using the transpose of the Jacobian to map the workspace forces to generalized forces for the configuration space, we obtain

$$J_a^T(x, y, \theta) \begin{bmatrix} F_x \\ F_y \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ -F_x(a_x \sin \theta - a_y \cos \theta) + F_y(a_x \cos \theta - a_y \sin \theta) \end{bmatrix}$$

The bottom entry in this vector corresponds to the torque exerted about the origin of the robot frame.

The total artificial joint torque acting on the arm is the sum of the artificial joint torques that result from all attractive and repulsive potentials

$$\tau(q) = \sum_i J_{o_i}^T(q) F_{\text{att},i}(q) + \sum_i J_{o_i}^T(q) F_{\text{rep},i}(q) \quad (5.11)$$

It is essential that we add the joint torques and *not* the workspace forces. In other words, we must use the Jacobians to transform forces to joint torques before we combine the effects of the potential fields. For example, Figure 5.8 shows a case in which two workspace forces F_1 and F_2 , act on opposite corners of a rectangle. It is easy to see that $F_1 + F_2 = 0$, but that the combination of these forces produces a pure torque about the center of the rectangle.

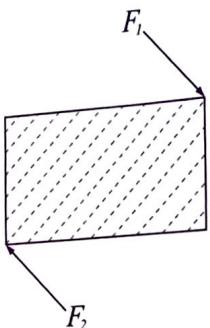


Figure 5.8: This example illustrates why forces must be mapped to the configuration space before they are combined. The two forces illustrated in the figure are vectors of equal magnitude in opposite directions. Vector addition of these two forces produces zero net force, but there is a net torque induced by these forces.

Example 5.7 Two-link planar arm

Consider again the two-link planar arm of Example 5.3, with joint torques as determined in Example 5.5. In this case the total joint torque induced by the attractive and repulsive workspace potential fields is given by

$$\begin{aligned}\tau(q_s) &= \tau_{\text{att},1}(q_s) + \tau_{\text{att},2}(q_s) + \tau_{\text{rep},2}(q_s) \\ &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \begin{bmatrix} -8 \\ -4 \end{bmatrix} = \begin{bmatrix} -5 \\ -3 \end{bmatrix}\end{aligned}$$

These joint torques have the effect of causing each joint to rotate in a clockwise direction, away from the goal, due to the close proximity of o_2 to the obstacle. By choosing a smaller value for η_2 , this effect can be overcome. \diamond

5.2.4 Gradient Descent Planning

Gradient descent is a well known approach for solving optimization problems. The idea is simple. Starting at the initial configuration, take a small step in the direction of the negative gradient (which is the direction that decreases the potential as quickly as possible). This gives a new configuration, and the process is repeated until the final configuration is reached. More formally, we can define a gradient descent algorithm as follows.

5.2. PATH PLANNING USING POTENTIAL FIELDS

Gradient Descent Algorithm

```

1.  $q^0 \leftarrow q_s, i \leftarrow 0$ 
2. IF  $\|q^i - q_f\| > \epsilon$ 
    $q^{i+1} \leftarrow q^i + \alpha^i \frac{\tau(q^i)}{\|\tau(q^i)\|}$ 
    $i \leftarrow i + 1$ 
ELSE return  $< q^0, q^1, \dots, q^i >$ 
3. GO TO 2

```

In this algorithm the notation q^i is used to denote the value of q at the i^{th} iteration (not the i^{th} component of the vector q) and the final path consists of the sequence of iterates $< q^0, q^1, \dots, q^i >$. The value of the scalar α^i determines the step size at the i^{th} iteration; it is multiplied by the unit vector in the direction of the resultant force. It is important that α^i be small enough that the robot is not allowed to “jump into” obstacles while being large enough that the algorithm does not require excessive computation time. In motion planning problems the choice for α^i is often made on an ad hoc or empirical basis, perhaps based on the distance to the nearest obstacle or to the goal. A number of systematic methods for choosing α^i can be found in the optimization literature.

It is unlikely that we will ever exactly satisfy the condition $q^i = q_f$ and for this reason the algorithm terminates in line 2 when q^i is sufficiently near the goal configuration q_f . We choose ϵ to be a sufficiently small constant, based on the task requirements.

There are a number of design choices that must be made when using this algorithm.

ζ_i in Equation (5.4) controls the relative influence of the attractive potential for control point o_i . It is not necessary that all of the ζ_i be set to the same value. Typically we assign a larger weight to one of the o_i than to the others, producing a “follow the leader” type of motion, in which the leader o_i is quickly attracted to its final position and the robot then reorients itself so that the other o_i reach their final positions.

η_i in Equation (5.6) controls the relative influence of the repulsive potential for o_i . As with the ζ_i it is not necessary that all of the η_i be set to the same value. In particular, we typically set the value of η_i to be much smaller for obstacles that are near the goal position of the robot (to avoid having these obstacles repel the robot from the goal).

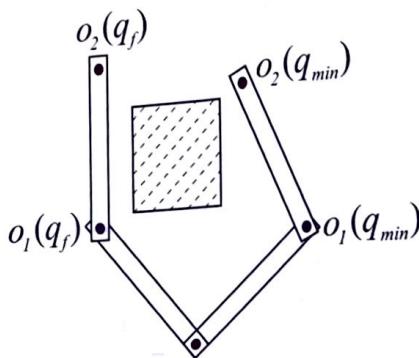


Figure 5.9: The configuration q_{\min} is a local minimum in the potential field. At q_{\min} the attractive force exactly cancels the repulsive force and the planner fails to make further progress.

ρ_0 in Equation (5.6) defines the distance of influence for obstacles. As with the η_i we can define a distinct value of ρ_0 for each obstacle. In particular, we do not want any obstacle's region of influence to include the goal position of any repulsive control point. We may also wish to assign distinct values of ρ_0 to the obstacles to avoid the possibility of overlapping regions of influence for distinct obstacles.

The problem that plagues all gradient descent algorithms is the possible existence of local minima in the potential field. For appropriate choice of α^i , it can be shown that the gradient descent algorithm is guaranteed to converge to a minimum in the field, but there is no guarantee that this minimum will be the global minimum. In our case this implies that there is no guarantee that this method will find a path to q_f . An example of this situation is shown in Figure 5.9. We will discuss ways to deal with this below in Section 5.3.

5.3 ESCAPING LOCAL MINIMA

As noted above, one problem that plagues artificial potential field methods for path planning is the existence of local minima in the potential field. In the case of articulated manipulators the resultant field U is the sum of many local minima. This problem has long been known in the optimization community, where probabilistic methods such as simulated annealing have been developed to cope with it. Similarly, the robot path planning community

5.3. ESCAPING LOCAL MINIMA

has developed what are known as **randomized methods** to deal with this and other problems.

The first method we discuss for escaping local minima combines gradient descent with randomization. This approach uses gradient descent until the planner finds itself stuck in a local minimum, and then uses a random walk to escape the local minimum. The algorithm is a slight modification of the gradient descent algorithm of Section 5.2.4:

1. $q^0 \leftarrow q_s, i \leftarrow 0$
2. IF $\|q^i - q_f\| > \epsilon$

$$q^{i+1} \leftarrow q^i + \alpha^i \frac{\tau(q^i)}{\|\tau(q^i)\|}$$
 $i \leftarrow i + 1$
ELSE return $< q^0, q^1, \dots, q^i >$
3. IF stuck in a local minimum
execute a random walk, ending at q'
 $q^{i+1} \leftarrow q'$
4. GO TO 2

The two new problems that must be solved are determining when the planner is stuck in a local minimum and defining the random walk. Typically, a heuristic is used to recognize a local minimum. For example, if several successive q^i lie within a small region of the configuration space, it is likely that there is a nearby local minimum (for example, if for some small positive ϵ_m we have $\|q^i - q^{i+1}\| < \epsilon_m$, $\|q^i - q^{i+2}\| < \epsilon_m$, and $\|q^i - q^{i+3}\| < \epsilon_m$ then assume q^i is near a local minimum).

Defining the random walk requires a bit more care. One approach is to simulate Brownian motion. The random walk consists of t random steps. A random step from $q = (q_1, \dots, q_n)$ is obtained by randomly adding a small fixed constant to each q_i ,

$$q_{\text{random-step}} = (q_1 \pm v_1, \dots, q_n \pm v_n)$$

with v_i a fixed small constant and the probability of adding $+v_i$ or $-v_i$ equal to 1/2 (that is, a uniform distribution). Without loss of generality, assume that $q = 0$. We can use probability theory to characterize the behavior of the random walk consisting of t random steps. In particular, the probability density function for $q' = (q_1, \dots, q_n)$ is given by

$$p_i(q_i, t) \approx \frac{1}{v_i \sqrt{2\pi t}} \exp\left(-\frac{q_i^2}{2v_i^2 t}\right) \quad (5.12)$$