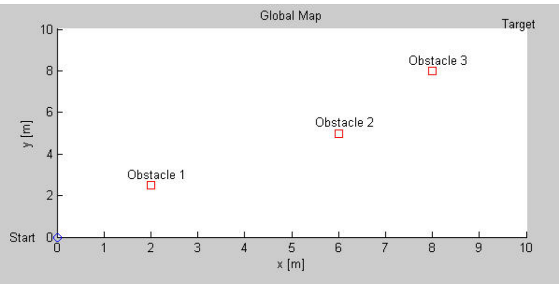
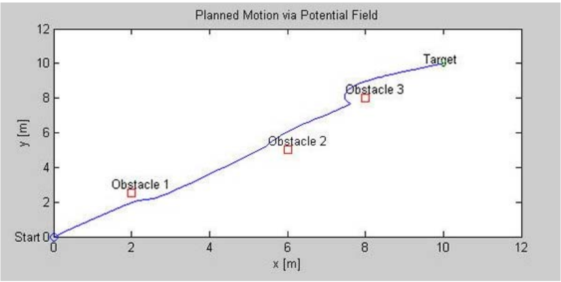
# Lecture XX – Path Planning I: Potential Fields

## XX Potential Fields

The left figure below depicts path planning where the robot must travel from start to target locations without colliding into obstacles. The right figure depicts a solution generated by the potential field based method. Well-tested, well understood and ubiquitous, this method serves as a good starting point to gather experience in autonomous robot navigation.



The robot must navigate from start position (0,0) to target location (10,10) without colliding into the three obstacles (squares).



The path generated by a potential field based method appears to work.

Founded upon attraction-repulsion theory in physics, the robot and obstacles are modeled as point particles. The *configuration space*, which is the environment the robot moves in, is modeled to be under the influence of a potential field $U$ with obstacles and the targets modeled as repulsive, $U_r$, and attractive, $U_a$, forces respectively. In other words

$$U(q) = U_a(q) + U_r(q) \tag{1}$$

where $q$ is the configuration vector, or the current robot location

$$q = \begin{bmatrix} x \\ y \end{bmatrix} \tag{2}$$

Recall that the gradient yields the direction of maximum increase. Thus, taking the gradient of the potential field, the robot should move away from obstacles and be drawn to the target. In other words

$$\dot{q} = -\nabla_q \{U_a(q) + U_r(q)\} \tag{3}$$

where the negative means to move in the direction opposite of maximum. What remains is to define suitable functions for $U_r$ and $U_a$.

### XX.1 Attractive Potential

Because a gradient is performed, a suitable $U_a$ must be differentiable across the entire configuration space. Additionally, $U_a$ should increase as the robot moves away from target. As such, one possibility is to use a parabolic well defined as

$$U_a = \frac{\eta_a}{2}\rho_t^{\,2}(q) \qquad (4)$$

where

$$\rho_t = \sqrt{(x_t - x)^2 + (y_t - y)^2} \qquad (5)$$

and

$\eta_a$ : attractive potential constant

$(x_t, y_t)$ : target location

$\rho_t$ : distance to target

**Problem XX.1:** Derive the gradient for the attractive potential defined by equations (4) and (5)

**Solution XX.1**: Recall that the gradient is a vector given by

$$\nabla_q U_a = \begin{pmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{pmatrix} \frac{1}{2}\eta_a \left\{ (x_t - x)^2 + (y_t - y)^2 \right\}$$

Consequently the $x$ component is $-\eta_a(x_t - x)$ and $y$ component is $-\eta_a(y_t - y)$. Hence the solution is

$$\nabla_q U_a = \begin{bmatrix} \eta_a(x - x_t) \\ \eta_a(y - y_t) \end{bmatrix} \qquad (6)$$

**QED**

### XX.2 Repulsive Potential

Like the attractive potential, $U_r$ must be differentiable throughout the configuration space but must decrease when the robot moves away from obstacles. A suitable candidate is

$$U_r(q) = \begin{cases} \frac{1}{2}\eta_r\left(\frac{1}{\rho} - \frac{1}{\rho_o}\right)^2 & \rho \le \rho_o \\ 0 & \rho > \rho_o \end{cases} \qquad (7)$$

Here

$$\rho = \sqrt{(y_o - y)^2 + (x_o - x)^2} \qquad (8)$$

Where

$\eta_r$ : repulsive potential constant

$(x_o, y_o)$ : obstacle location

$\rho$ : current robot-to-obstacle distance

$\rho_o$ : repulsive force maximum range

**Problem XX.2:** Derive the gradient for the repulsive field defined by (7) and (8)

**Solution XX.2:** Equations (7) and (8) combine to yield the following repulsive field

$$U_r(q) = \frac{1}{2}\eta_r\left(\frac{\rho_o - \rho}{\rho\rho_o}\right)^2 = \frac{\eta_r\left\{\rho_o - \left[(y_o - y)^2 + (x_o - x)^2\right]^{1/2}\right\}^2}{2\rho_o^{\,2}\left\{(y_o - y)^2 + (x_o - x)^2\right\}}$$

$$= \frac{\eta_r}{2\rho_o^{\,2}}\left\{\rho_o - \left[(y_o - y)^2 + (x_o - x)^2\right]^{1/2}\right\}^2\left\{(y_o - y)^2 + (x_o - x)^2\right\}^{-1}$$

The $x$ component of the gradient can be calculated using the chain rule

$$\frac{\partial U_r}{\partial x} = \frac{\partial U_r}{\partial \rho}\frac{\partial \rho}{\partial x}$$

First,

$$\frac{\partial U_r}{\partial \rho} = \frac{\partial}{\partial \rho}\left\{\frac{1}{2}\eta_r\left(\rho^{-1} - \frac{1}{\rho_o}\right)^2\right\}$$

$$= \eta_r\left(\rho^{-1} - \frac{1}{\rho_o}\right)\left(-\rho^{-2}\right) = -\eta_r\left(\frac{1}{\upsilon} - \frac{1}{\rho_o}\right)\frac{1}{\rho^2} \qquad (9)$$

Second,

$$\frac{\partial \rho}{\partial x} = \frac{\partial}{\partial x}\left\{(y_o - y)^2 + (x_o - x)^2\right\}^{1/2} = -\left\{(y_o - y)^2 + (x_o - x)^2\right\}^{-1/2}(x_o - x)$$

or, substituting with (8)

$$\frac{\partial \rho}{\partial x} = -\frac{(x_o - x)}{\rho} \qquad (10)$$

Combining (9) and (10) yields

$$\frac{\partial U_r}{\partial x} = \eta_r\left(\frac{1}{\rho} - \frac{1}{\rho_o}\right)\frac{(x_o - x)}{\rho^3} \qquad (11)$$

Likewise for the $y$ component, one can derive the following equations

$$\frac{\partial \rho}{\partial y} = -\frac{\left(y_o - y\right)}{\rho} \tag{12}$$

Combining (9) and (12) yields

$$\frac{\partial U_r}{\partial y} = \eta_r \left(\frac{1}{\rho} - \frac{1}{\rho_o}\right) \frac{\left(y_o - y\right)}{\rho^3} \tag{13}$$

Finally the gradient can be expressed using (11) and (13) as

$$\nabla_q U_r = \begin{cases} \dfrac{\eta_r \left(x_o - x\right)}{\rho^3} \left(\dfrac{1}{\rho} - \dfrac{1}{\rho_o}\right) \\ \dfrac{\eta_r \left(y_o - y\right)}{\rho^3} \left(\dfrac{1}{\rho} - \dfrac{1}{\rho_o}\right) \end{cases} \quad \text{when} \quad \rho \le \rho_o$$

$$\nabla_q U_r = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{when} \quad \rho > \rho_o \tag{14}$$

**QED**

### XX.3 Updating Robot Positions

Having derived the gradients of the attractive and attractive fields, the robot position within configuration space can be calculated. Given a sampling time $\Delta T$ and (3), the derivative can be written as a difference equation for sample $i$

$$\frac{q[i] - q[i-1]}{\Delta T} = -\nabla_q \left(U_a + U_r\right)$$

The update rule then becomes

$$q[i] = q[i-1] - \Delta T \nabla_q \left(U_a + U_r\right)$$

The $x$ components of both (6) and (14) yield an update equation for the robot's $x$ location

$$x[i] = \begin{cases} x[i-1] - \Delta T \eta_a \left(x[i-1] - x_t\right) - \dfrac{\Delta T \eta_r \left(x_o - x[i-1]\right)}{\rho^3} \left\{\dfrac{1}{\rho} - \dfrac{1}{\rho_o}\right\} & \text{when} \quad \rho \le \rho_o \\ x[i-1] - \Delta T \eta_a \left(x[i-1] - x_t\right) & \text{when} \quad \rho > \rho_o \end{cases} \tag{15}$$

The $y$ components of (6) and (14) yield

$$y[i]=\begin{cases} y[i-1]-\Delta T\eta_a\left(y[i-1]-y_t\right)-\dfrac{\Delta T\eta_r\left(y_o-y[i-1]\right)}{\rho^3}\left\{\dfrac{1}{\rho}-\dfrac{1}{\rho_o}\right\} & \text{when} \quad \rho\le\rho_o \\ y[i-1]-\Delta T\eta_a\left(y[i-1]-y_t\right) & \text{when} \quad \rho>\rho_o \end{cases} \tag{16}$$

Where

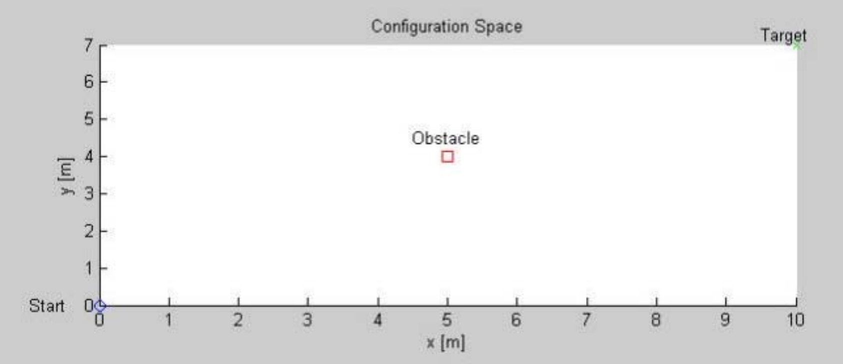
$$\rho=\sqrt{\left(y_o-y[i-1]\right)^2+\left(x_o-x[i-1]\right)^2}$$

XX.4 Algorithm Implementation

Potential field based path planning can be implemented in any computer language. Matlab has plot commands allowing results to be displayed graphically.

**Problem XX.3:** Given the robot's start and target locations are $(x_s, y_s)=(0,0)$ and $(x_t, y_t)=(10,7)$ respectively, plan a path using a potential field that avoids an obstacle located at $(x_o, y_o)=(5,4)$. Let the maximum range for the repulsive field $\rho_o=2$ and let the attractive and repulsive field constants respectively be $\eta_a=2$ and $\eta_r=1$, and sampling $\Delta T=0.1$.

**Solution XX.3:** Graphically, the problem is depicted in the figure below



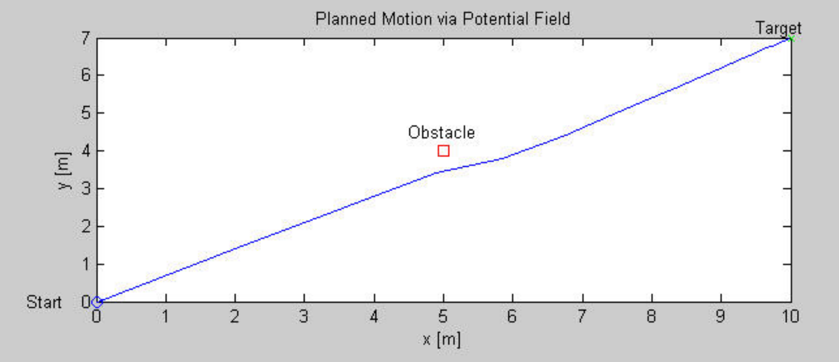
The M-file **`potField1_1.m`** yields the planned path and graphically displays the robot's trajectory. The code begins by initiating the variables that define the robot's start and target locations and the obstacle location. A `for` loop (see below) calculates equations (5), (8), (11), (13), (15) and (16). The variable `iterations` the maximum times the loop should cycle. If the configuration space is large or $\Delta T$ is small, then `iterations` should be increased to ensure the robot reaches the target.

```
for i=2:iterations  % Run algorithm for 100 sample periods
  t(i-1) = T*(i-1); % time in seconds
  % Distance to target (Eqn 5)
  rhoTarget(i-1) = sqrt(((yTarget-y(i-1))^2) + ((xTarget-x(i-1))^2));
  % Distance to an obstacle (Eqn 8)
  rho(i-1) = sqrt( ((yObstacle-y(i-1))^2) + ((xObstacle-x(i-1))^2) );
  % Calculate gradient (Eqns 11 and 13)
  if rho(i-1) < rho0
   ur_x = nRepulse*(xObstacle-x(i-1))*((1/rho(i-1))-1/rho0))/(rho(i-1)^3);
   ur_y = nRepulse*(yObstacle-y(i-1))*((1/rho(i-1))-(1/rho0))/(rho(i-1)^3);
  else
   ur_x = 0;
   ur_y = 0;
  end

  % Calculate new robot position (Eqns 15 and 16)
  x(i) = x(i-1) - T*nAttract*(x(i-1)-xTarget) - T*ur_x;
  y(i) = y(i-1) - T*nAttract*(y(i-1)-yTarget) - T*ur_y;

end
```

Conceivably, if a real robot were used, motor commands would be issued in the above `for` loop. Graphically, the trajectory traced out by the robot can be depicted; once the loop is completed, Matlab `plot` commands display the path. This result is given in the figure below.



**QED**

**Homework:** Given the robot's start and target locations are $(x_s, y_s) = (0,0)$ and $(x_t, y_t) = (10,10)$ respectively, plan a path using a potential field that avoids 3 obstacles. The locations are $(x_{o1}, y_{o1}) = (2,2.5)$, $(x_{o2}, y_{o2}) = (6,5)$ and $(x_{o3}, y_{o3}) = (8,8)$. Let the maximum range for the repulsive fields be $\rho_{o1} = \rho_{o2} = \rho_{o3} = 3$ and let the attractive and repulsive field constants respectively be $\eta_a = 2$ and $\eta_{r1} = \eta_{r2} = \eta_{r3} = 1$, and sampling $\Delta T = 0.1$.