Potential Field Method

EN.530.663: Robot Motion Planning

Jin Seob Kim, Ph.D

Senior Lecturer, LCSR, ME dept., JHU

Spring 2024

Potential Field Method

- Incrementally "explores" free space (C_{free}) while searching for a path.
- Mimics a particle moving in a potential field'.
- Potential field in physics: attractive potential + repulsive potential
- Robot in C-space $\rightarrow$ a point $\Rightarrow$ treated as a particle under the influence of an artificial potential function $U(\mathbf{q})$.

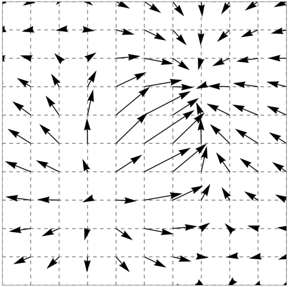
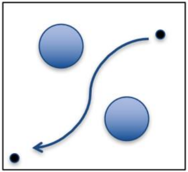
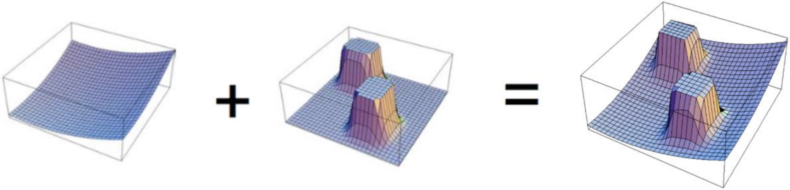


Figure: Example of a potential field [https://ximera.osu.edu/mooculus]

- Example:





attractive repulsive Total potential field

From cs.cmu.edu

Potential Field Method: Potential Function

- Potential function: a differentiable real-valued function

$$U : \mathbb{R}^n \to \mathbb{R} : \mathbf{q} \in \mathbb{R}^n \mapsto U(\mathbf{q})$$

- gradient:

$$\nabla U(\mathbf{q}) = \begin{pmatrix} \frac{\partial U}{\partial q_1} \\ \vdots \\ \frac{\partial U}{\partial q_n} \end{pmatrix}$$

- direction: locally maximally increasing direction.
- In physics: conservative force

$$\mathbf{F} = -\nabla U$$

Potential Field Method: Potential Function

- Now, artificial potential function

$$U : C_{free} \to \mathbb{R} : q \in C_{free} \mapsto U(q)$$
$$\to \mathbf{F}(q) = -\nabla U(q), \;\; q \in C_{free}$$

  $\mathbf{F}(q)$: force applied to the point.

- In physics, highly damped case (on inertial effect)
- Similar to the gradient descent optimization process
- The robot terminated the motion when it reaches a point q^* (critical point) where $\nabla U = \mathbf{0}$

Potential Field Method: Attractive Potential Function

- $U(q) = U_{att}(q) + U_{rep}(q)$
- Attractive potential function:

$$
\begin{aligned}
&U_{att}(q) \doteq \frac{1}{2}\zeta\,[\rho(q,\, q_G)]^2 \;\;,\;\; \zeta > 0 \\
&\text{if } q \in \mathbb{R}^n, \rho(q,\, q_G) = \|q - q_G\| \\
&\rightarrow \nabla U_{att}(q) = \frac{1}{2}\zeta\nabla\rho^2(q, q_G) = \zeta\rho(q,\, q_G)\nabla\rho(q,\, q_G) \\
&\therefore \nabla U_{att}(q) = \zeta(q - q_G)
\end{aligned}
$$

- The farther away q is from q_G, the bigger the magnitude $\rightarrow$ If initially too far away, it causes a numerical problem.

Potential Field Method: Attractive Potential Function

- Conic and quadratic potential function

$$U_{att}(q) \doteq \begin{cases} \frac{1}{2}\zeta\rho^2(q,\, q_G), & \rho(q,\, q_G) \leq \rho_G^* \\ \rho_G^*\,\zeta\,\rho(q,\, q_G) - \frac{1}{2}\zeta\left(\rho_G^*\right)^2, & \rho(q,\, q_G) > \rho_G^* \end{cases}$$

where ρ_G^*: threshold distance from the goal.

- Then, with $q \in \mathbb{R}^n$,

$$\nabla U_{att}(q) = \begin{cases} \zeta(q - q_G), & \rho(q,\, q_G) \leq \rho_G^* \\ \frac{\rho_G^*\,\zeta\,(q - q_G)}{\rho(q,\, q_G)}, & \rho(q,\, q_G) > \rho_G^* \end{cases}$$

Potential Field Method: Repulsive Potential Function

- Keeps the robot away from an obstacle.
- When the robot is sufficiently away from C_{obs}, U does not affect the robot motion.

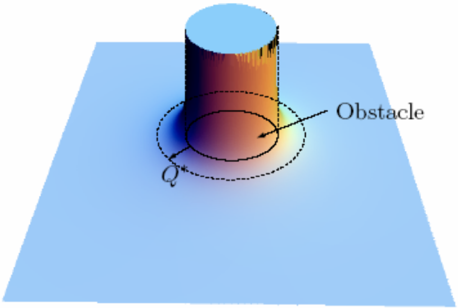


Figure: from cs.cum.edu

Potential Field Method: Repulsive Potential Function

- Usually,

$$U_{rep}(q) \doteq \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho(q)} - \frac{1}{Q^*}\right)^2, & \rho(q) \le Q^* \\ 0\,, & \rho(q) > Q^* \end{cases}$$

  where $\eta > 0$, $\rho(q) = \min_{q' \in C_{obs}} \rho(q,\, q')$.

- Then

$$\nabla U_{rep}(q) = \begin{cases} \eta\left(\frac{1}{Q^*} - \frac{1}{\rho(q)}\right)\frac{1}{\rho^2(q)}\nabla\rho(q)\,, & \rho(q) \le Q^* \\ 0\,, & \rho(q) > Q^* \end{cases}$$

- For convex obstacles

$$\nabla\rho(q) = \frac{q - c}{\rho(q,\, c)}$$

  where c: closest point on C_{obs} to q.

- Multiple C_{obs}'s: $U_{rep}(q) = \sum_{i=1}^{N} U_{rep,i}(q)$

Potential Field Method: Algorithm

- Input: A means to compute the gradient $\nabla U(q)$ at q.
- Output: A sequence of points $\{q(0), q(1), q(2), \cdots q(i), \cdots\}$
- Algorithm:
  - $q(0) = q_I$
  - $i = 0$
  - while $\nabla U(q) \neq 0$, d0:
  - $q(i+1) = q(i) - \alpha(i) \nabla U(q(i))$
  - $i = i + 1$
  - end

Potential Field Method

- Easy to implement.
- Mimics a particle moving in a potential field.
- Issue of being stuck in any of local minima.
- A major challenge: constructing C-space.
- Randomized Path Planner (RPP): follows the potential field method, and when stuck in local minima, it initializes random walks to escape local minima.
- Another:
  - Wave-front planner
  - Navigation function (only one minima)

Potential Field Approach for Rigid Bodies

- example: serial manipulator
- C-space: non-Euclidean.
- Here, treat "gradient" as "force" $\to$ establish a relation between a workspace force and a configuration space force.
- Let $\mathbf{f}$, $\mathbf{u}$ be the forces in $\mathcal{W}$, C, respectively. $\Rightarrow J^\top \mathbf{f} = \mathbf{u}$ (J: Jacobian)
- Pick control points $\{\mathbf{r}_j\}$ on the robot in $\mathcal{W} \to$ "pin down the robot".
- Then potential function:

$$U(q) = \sum_{j=1} U_{att,j}(q) + \sum_{i=1} \sum_{j=1} U_{rep,i,j}(q)$$

Potential Field Approach for Rigid Bodies

- For each $\mathbf{r}_j$:

$$U_{att,j}(q) = \begin{cases} \frac{1}{2}\zeta_j\,\rho^2(\mathbf{r}_j(q),\,\mathbf{r}_j(q_G)), & \rho(\mathbf{r}_j(q),\,\mathbf{r}_j(q_G)) \leq \rho_G^* \\ \rho_G^*\,\zeta_j\,\rho(\mathbf{r}_j(q),\,\mathbf{r}_j(q_G)) - \frac{1}{2}\zeta_j\,\left(\rho_G^*\right)^2, & \text{otherwise} \end{cases}$$

$$U_{rep,i,j}(q) \doteq \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho_i(\mathbf{r}_j(q))} - \frac{1}{Q^*}\right)^2, & \rho_i(\mathbf{r}_j(q)) \leq Q_i^* \\ 0\,, & \text{otherwise} \end{cases}$$

- ρ, ρ_i : distance in $\mathcal{W}$.
- $\rho_i(\mathbf{r}_j(q))$: shortest distance between $\mathbf{r}_j$ and obstacle Q_i.
- Q_i^*: workspace influence distance.

Potential Field Approach for Rigid Bodies

- The force:

$$
\begin{aligned}
\mathbf{u}(q) &= \sum_j \mathbf{u}_{att,j}(q) + \sum_i \sum_j \mathbf{u}_{rep,i,j}(q) \\
&= \sum_j J_j^\top(q)\mathbf{f}_{att,j}(q) + \sum_i \sum_j J_j^\top(q)\mathbf{f}_{rep,i,j}(q)
\end{aligned}
$$

- where $\mathbf{f}_{att,j}(q) = -\nabla_{\mathbf{r}} U_{att,j}, \quad \mathbf{f}_{rep,i,j}(q) = -\nabla_{\mathbf{r}} U_{rep,i,j}, \quad \dot{\mathbf{r}}_j = J_j(\mathbf{q})\dot{\mathbf{q}}$

- Then algorithm: $\dot{\mathbf{q}} = \mathbf{u}$

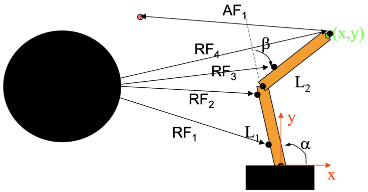


Figure: from cs.cmu.edu

Potential Field Method: Example

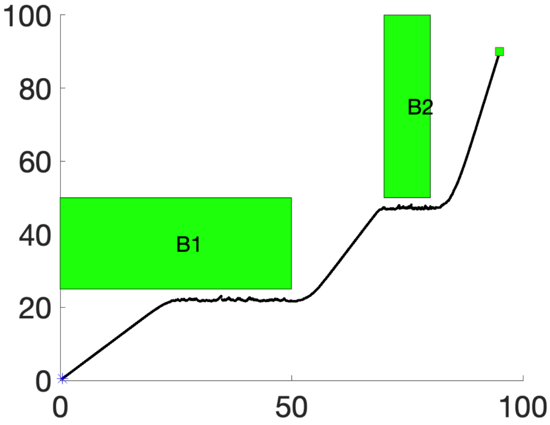


Figure: An example of potential field method (planar, point robot case)