

Práctica 7

Evasión de obstáculos mediante campos potenciales artificiales

Laboratorio de Bio-Robótica
Robots Móviles y Agentes Inteligentes

Objetivos

- Implementar un comportamiento reactivo para evasión de obstáculos mediante campos potenciales artificiales.
- Determinar los parámetros de diseño para obtener un comportamiento satisfactorio.
- Probar la evasión de obstáculos tanto en simulación como experimentalmente.

1. Marco Teórico

1.1. Campos potenciales artificiales

Los campos potenciales artificiales son un método para evadir obstáculos de manera reactiva, es decir, no es necesario que dichos obstáculos estén en algún tipo de representación del espacio. En el caso de A*, por ejemplo, se pueden trazar rutas que evadan obstáculos pero estos deben estar definidos en el mapa de celdas de ocupación a partir del cual se calculan dichas rutas.

Los campos potenciales artificiales, como su nombre lo indica, son un método en el que se diseña una función potencial tal que, si el robot “desciende” por dicha función, seguirá una ruta que lo llevará al punto meta al mismo tiempo que evade los obstáculos a su paso.

Una función potencial es una función real diferenciable $U : R^n \rightarrow R$ que puede verse como una función de energía y que, por lo tanto, su gradiente $\nabla U(q)$, donde q es la posición del robot, representa fuerza.

El gradiente de la función, dado por

$$\nabla U(q) = \left[\frac{\partial U}{\partial q_1}, \dots, \frac{\partial U}{\partial q_n} \right] \quad (1)$$

es un vector que apunta en la dirección de máximo cambio de U . Si esta función potencial se diseña de modo que tenga un mínimo en la posición meta y máximos locales en la posición de cada obstáculo que se desee evadir, entonces se puede mover al robot mediante el algoritmo de descenso del gradiente y esto hará que el robot continúe moviéndose hasta encontrar un punto q^* en el cual $\nabla U(q^*) = 0$.

Los valores q que satisfacen $\nabla U(q^*) = 0$ son llamados puntos críticos y pueden ser máximos locales, mínimos locales o puntos silla. Para determinar la naturaleza de un punto crítico, se puede utilizar la matriz Hessiana, dada por

$$H(q) = \begin{bmatrix} \frac{\partial^2 U}{\partial q_1^2} & \cdots & \frac{\partial^2 U}{\partial q_1 \partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 U}{\partial q_n \partial q_1} & \cdots & \frac{\partial^2 U}{\partial q_n^2} \end{bmatrix} \quad (2)$$

Si $H(q)$ evaluada en q^* es no singular, significa que q^* es un punto crítico aislado. Si $H(q^*)$ es positiva definida, entonces q^* es un mínimo local, si es negativa definida, se trata de un máximo local, y en otro caso, q^* es un punto silla. Si $H(q^*)$ es singular, entonces significa que q^* no es un punto aislado, es decir, que la función potencial U es “plana” en la región que contiene a q^* .

En general, si el robot se mueve siempre en sentido contrario al gradiente, en algún momento llegará a un mínimo local. Puede suceder que la condición inicial del robot corresponda a un máximo local o a un punto silla, en cuyo caso el robot no se moverá, pues en estos puntos el gradiente es cero. Esta situación es muy improbable, además, cualquier perturbación hará que el robot “se mueva” de ese punto y entonces se dirigirá al mínimo local, es decir, los puntos silla y los máximos son puntos inestables.

1.2. Diseño mediante campos atractivos y repulsivos

Como se expuso previamente, la función potencial $U(q)$ debe diseñarse de modo que sea diferenciable, que tenga un mínimo en el punto meta, máximos locales en cada obstáculo, y todos sus puntos críticos deben ser aislados. Una forma sencilla de lograrlo es diseñando por separado dos campos: uno atractivo y otro repulsivo. La idea es que el punto meta debe generar una “fuerza atractiva” mientras que cada obstáculo debe generar una “fuerza repulsiva”.

Algunas veces no es necesario diseñar primero el campo potencial y luego obtener su gradiente, sino que se puede diseñar directamente la fuerza correspondiente al gradiente. La fuerza atractiva se debe diseñar de modo que siempre se aleje del punto meta y las fuerzas repulsivas siempre deben tener una dirección que apunte hacia el obstáculo que se desea evadir, además, las fuerzas repulsivas deben crecer conforme la distancia a dicho obstáculo disminuye y ser muy pequeñas o cero, si el robot está lo suficientemente lejos.

Es importante recordar que las fuerzas tanto atractiva como repulsiva corresponden al gradiente de la función potencial, es decir, representan la dirección de máximo cambio, sin embargo, lo que se desea es alcanzar un mínimo, por lo que al aplicar el algoritmo de descenso del gradiente, el robot se moverá en sentido contrario a estas fuerzas.

Para esta práctica, la fuerza atractiva que se utilizará está dada por

$$F_a = \zeta \frac{(q - q_g)}{\|q - q_g\|} \quad \zeta > 0 \quad (3)$$

donde q es la posición actual del robot y q_g es la posición a donde se desea llegar. Esta fuerza tiene una magnitud ζ constante pero su dirección apunta siempre en sentido contrario al punto meta.

La fuerza de repulsión está dada por:

$$F_r = \begin{cases} \eta \left(\sqrt{\frac{1}{\|q_{oi} - q\|} - \frac{1}{d_0}} \right) \frac{q_{oi} - q}{\|q_{oi} - q\|} & \text{si } \|q_{oi} - q\| < d_0 \\ 0 & \text{en otro caso} \end{cases} \quad (4)$$

donde q_{oi} es la posición del obstáculo i y d_0 es una distancia de influencia tal que, si el robot está a una distancia mayor que ésta del obstáculo i , entonces la fuerza de repulsión es cero. Nótese que la magnitud de esta fuerza se incrementa conforme el robot se acerca al objeto y la raíz cuadrada produce una transición suave cuando el robot pasa por d_0 .

La fuerza resultante se calcula como

$$F = F_a + \frac{1}{n} \sum_{i=1}^n F_r \quad (5)$$

es decir, la fuerza atractiva más el promedio de las fuerzas repulsivas producidas por cada obstáculo.

1.3. Movimiento por descenso del gradiente

Una vez diseñado el campo potencial y calculado el respectivo gradiente, sólo basta con mover al robot de acuerdo con el algoritmo 1. La constante $\alpha > 0$ debe ser lo suficientemente pequeña para evitar oscilaciones pero sin que ello conlleve un costo computacional muy alto.

Aunque en el algoritmo se indica que el robot seguirá moviéndose mientras la magnitud del gradiente sea mayor que cero, en una implementación real se fija una tolerancia y cuando la magnitud es menor que ésta, se detiene al robot.

Algoritmo 1: Descenso del gradiente para mover al robot a través de un campo potencial.

Datos: Posición inicial q_s , posición final q_g , posiciones q_{oi} de los obstáculos

Resultado: Secuencia de puntos $\{q_0, q_1, q_2, \dots\}$

$q_0 \leftarrow q_s$

$i \leftarrow 0$

mientras $\|\nabla U(q_i)\| > 0$ **hacer**

$q_{i+1} \leftarrow q_i - \alpha \nabla U(q_i)$

$i \leftarrow i + 1$

fin

Finalmente, para que el robot se mueva hacia cada punto q_i generado por el descenso del gradiente, se pueden utilizar las leyes de control implementadas en la práctica 6. Los pasos resultantes para evadir obstáculos mediante campos potenciales artificiales se muestran en el algoritmo 2.

1.4. El sensor láser Hokuyo-URG

La base móvil en la que se harán las pruebas cuenta con un sensor láser cuyas lecturas indican distancia hacia un objeto. Este sensor entrega una nube de puntos sobre un plano y cada punto está expresado en coordenadas polares. La figura ASDFASDFS muestra esquemáticamente las lecturas del sensor láser. El ángulo de cada lectura está expresado con respecto al robot por lo que para obtener la posición de cada punto con respecto al sistema absoluto, es necesario sumarle el ángulo del robot.

Las lecturas del láser son publicadas por un nodo en el tópico `/hardware/scan` de tipo `sensor_msgs::LaserScan`, el cual, de manera simplificada, tiene la siguiente estructura:

```
Header header
float32
```

Algoritmo 2: Evasión de obstáculos por campos potenciales.

Datos: Posición inicial q_s , posición final q_g , posiciones q_{oi} de los obstáculos

Resultado: Secuencia de puntos $\{q_0, q_1, q_2, \dots\}$

$q_0 \leftarrow q_s$

$k \leftarrow 0$

mientras $\|\nabla U(q_k)\| > tol$ **hacer**

$F_a \leftarrow \zeta \frac{(q_k - q_g)}{\|q_k - q_g\|}$

para $i \in [0, n]$ **hacer**

$F_{r_i} \leftarrow \begin{cases} \eta \left(\sqrt{\frac{1}{\|q_{oi} - q_k\|} - \frac{1}{d_0}} \right) \frac{q_{oi} - q_k}{\|q_{oi} - q_k\|} & \text{si } \|q_{oi} - q_k\| < d_0 \\ 0 & \text{en otro caso} \end{cases}$

fin

$F \leftarrow F_a + \frac{1}{n} \sum_{i=1}^n F_{r_i}$

$q_{k+1} \leftarrow q_k - \alpha \frac{F}{\|F\|}$

$e_\theta \leftarrow \text{atan2}(q_{k+1}^y - y_r, q_{k+1}^x - x_r) - \theta_r$

$v_l \leftarrow v_{max} e^{-\frac{e_\theta^2}{\alpha}} + \frac{L}{2} \omega_{max} \left(\frac{2}{1 + e^{-\frac{e_\theta}{\beta}}} - 1 \right)$

$v_r \leftarrow v_{max} e^{-\frac{e_\theta^2}{\alpha}} - \frac{L}{2} \omega_{max} \left(\frac{2}{1 + e^{-\frac{e_\theta}{\beta}}} - 1 \right)$

$k \leftarrow k + 1$

fin

2. Tareas

2.1. Prerrequisitos

Antes de continuar, actualice el repositorio y recompile:

```
cd ~/RoboticsCourses
git pull origin master
cd catkin_ws
catkin_make
```

2.2. Nodo que implementa los campos potenciales

3. Evaluación

- El código debe estar ordenado.
- **Importante:** Si el alumno no conoce su código, NO se contará la práctica.