Robótica I

Tareas 1 y 2. Tangent Bug

9 de mayo de 2020

Marco Antonio Esquivel Basaldua

I. Introducción

El algoritmo de plaeación de movimientos en robótica *Tangent Bug* sirve como una mejora al algoritmo *Bug2*, determinando el camino más corto a la meta usando un sensor, con rango de alcance, de orientación de 360 grados de resolución infinita. Este sensor se modela con la función de distancia $\rho : \mathbb{R}^2 \times S^1 \rightarrow \mathbb{R}$. Se considera que el robot es un punto localizado en $x \in \mathbb{R}^2$ con líneas de sensado emanando de él de forma radial. Para cada $\theta \in S^1$, el valor $\rho(x, \theta)$ es la distancia al obstáculo más cercano siguiendo la línea de sensado desde x con un ángulo θ. De manera más formal

$$\rho(x,\theta) = \min_{\lambda \in [0.\infty]} d(x, x + \lambda[cos\theta, sin\theta]^T),$$
$$tal\ que\ x + \lambda[cos\theta, sin\theta]^T \in \cup_i \mathcal{WO}_i$$

Ya que los sensores en la vida real tienen rango limitado, se define la función de distancia saturada denotada por $\rho_R : \mathbb{R}^2 \times S^1 \rightarrow \mathbb{R}$ que toma los valores de ρ cuando un obstáculo se encuentra dentro de su rango de sensado y vale infinito cuando las líneas de sensado son mayores a este rango, R, lo que significa que los obstáculos están fuera de del espacio de sensado. De forma más formal

$$\rho_R(x,\theta) = \begin{cases} \rho(x,\theta) & si \quad \rho(x,\theta) < R \\ \infty & si \quad e.o.c. \end{cases}$$

Justo como los algoritmos *Bug1* y *Bug2*, *Tangent Bug* itera entre dos comportamientos: ¨ ir a la meta¨ y ”seguimiento de frontera”. Estos comportamientos difieren de *Bug1* y *Bug2*. Aunque ¨ ir a la meta” direcciona el robot a la meta, este comportamiento puede tener una fase donde el robot sigue la frontera, y el ”seguimiento de frontera”puede tener una fase en la que no se sigue la frontera.
Para mayor información se puede consultar el capítulo 2 del libro ”Principles of Robot Motion, Theory, Algorithms and Implementations”de Howie Choset et al.

II. Algoritmo

El algoritmo *Tangent Bug* se presenta a continuación.

Input: A point robot with a range sensor
Output: A path to the q_{goal} or a conclusion no such path exists

1: **while** True **do**
2: **repeat**
3: Continuously move toward the point $n \in \{T, O_i\}$ which minimizes $d(x, n) + d(n, q_{\text{goal}})$
4: **until**
- the goal is encountered **or**
- The direction that minimizes $d(x, n) + d(n, q_{\text{goal}})$ begins to increase $d(x, q_{\text{goal}})$, i.e., the robot detects a “local minimum” of $d(\cdot, q_{\text{goal}})$.

5: Chose a boundary following direction which continues in the same direction as the most recent motion-to-goal direction.
6: **repeat**
7: Continuously update d_{reach}, d_{followed}, and $\{O_i\}$.
8: Continuously moves toward $n \in \{O_i\}$ that is in the chosen boundary direction.
9: **until**
- The goal is reached.
- The robot completes a cycle around the obstacle in which case the goal cannot be achieved.
- $d_{\text{reach}} < d_{\text{followed}}$

10: **end while**

III. Implementación

Para la implementación del algoritmo se hace uso del simulador *webots* en el que se simula el robot *e-puck* (figura 1). Este es un robot tipo disco de 7.4 cm de diámetro de diámetro y 4.5 cm de alto.
Cabe señalar que aunque el algoritmo *Tangent Bug* está pensado para un robot tipo punto con un sensor de rango de resolución infinita, en este caso se adapta el comportamiento del algoritmo para el robot *e-puck* que es un robot disco con sólo 8 rayos de sensado radiales. Entre estas modificaciones se encuentra el hacer que el robot rote en sitio para alinearse ya sea a la meta o a la frontera que debe seguir. También se debe de tomar en cuenta que no es posible que el robot pase por espacios menores a su radio, como lo haría un robot tipo punto.

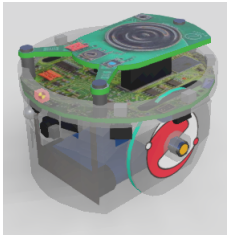

Figura 1. Robot *e-puck* usado en la implementación.

La ejecución del algoritmo se ejemplifica en un video sensillo en el que se observa como el robot *e-puck* alcanza la meta marcada con un punto rojo en un mapa con dos

obstáculos. El mapa utilizado se muestra en la figura 2 en la que se aprecia la ubicación inical del robot y la meta marcada por un círculo rojo.

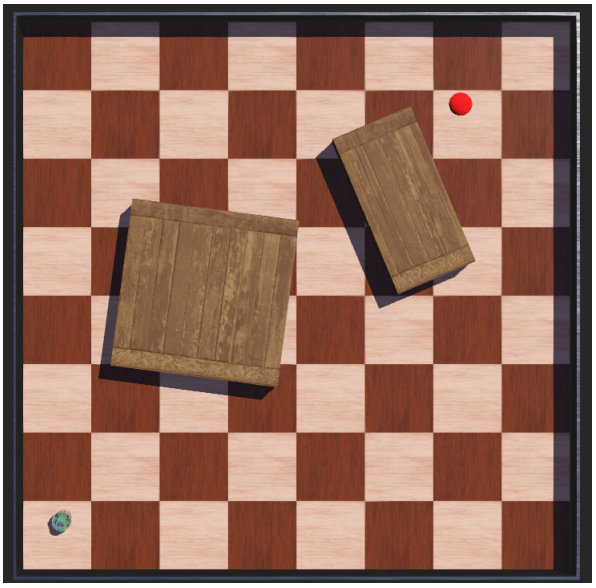

Figura 2. Mapa de prueba para el algoritmo *Tangent Bug*.

IV. Prueba de algortimo completo

Para probar que el algoritmo *Tangent Bug* es completo se consideran dos escenarios, uno en el que el robot llega al objetivo q_{goal} en un camino de distancia finita desde q_{start}y otro en el que no se encuentra tal camino a la meta.

Para el pimero de los escenarios, primero se considera el caso de un robot con un sensor de contacto ($R = 0$). En lo siguiente se denota D_{goal} como el disco centrado en q_{goal} y de radio $||q_{start} - q_{goal}||$. Usando un sensor de este tipo, existe una cota superior L_{max}

$$L_{max} = ||q_{start} - q_{goal}|| + \sum_{i \in \mathcal{I}} \Pi_i \times \#Minima_i$$

donde $\mathcal{I}$ es el índice de los obstáculos que intersectan el disco D_{goal}, Π_i es el perímetro del i-ésimo obstáculo y $\#Minima_i$ es el número de mínimos locales de $d(x, q_{goal})$ en D_{goal} a lo largo de la fontera del iésimo obstáculo. Ya que se trabaja con un número finito de obstáculos y de perímetro finito cada uno, por tanto con lugares finitos donde se generen mínimos locales en la función de distancia, se comprueba que $L_{max} < \infty$.
En el caso de un sensor con un rango de deteccion ilimitado ($R = \infty$), se hacen los siguientes dos supuestos. Sea $\hat{v}$ denota un vector unitario. Los nodos O_i (veáse la figura 3) satisfacen $(\widehat{O_i - x})(\widehat{q_{goal} - x}) > 0$, donde x es la posición actual del robot. Se asume que existe un valor γ positivo y mayor a cero tal que $(\widehat{O_i - x})(\widehat{q_{goal} - x}) > \gamma$. El segundo supuesto tiene que ver con la condición de dejar el comportameinto de "seguimiento de frontera". El nodo O_i que se busca para para dejar este comportamiento debe de estar dentro de un rango ρ, donde ρ es un valor finito y positivo. Bajo estos supuestos se define la cota superior

$$L_{max} = (1 + \frac{1}{\gamma})||q_{start} - q_{goal}|| + \sum_{i \in \mathcal{I}} (\Pi_i + \rho) \times \#Minima_i$$

Estas dos cotas descritas están definidas en los casos límite del rango de sensado, por lo que tambi'en se comprueban los casos en que $0 < R < \infty$¿

Para el segundo de los escenarios, se reporta que no se encuentra un camino a la meta cuando se recorre todo el borde de un obstáculo. Es decir, cuando se regresa al punto donde se hizo el cambio por última vez del comportamiento ¨ ir a la meta¨ a "segimiento de frontera".

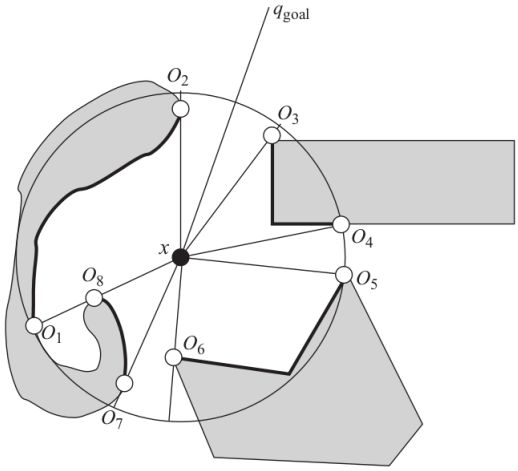

Figura 3. Puntos que indican discontinuidad debido a la pescencia de obstáculos.

Referencias

[1] H. Cheset et.al., "Principles of Robot Motion, Theory, Algorithms and Applications," MIT Press, pp. 23–31, 2005.

[2] Yoshiaki Shirai, "Robotics Research: The Eighth International Symposium", Springer, pp.113–116, 2012