Técnicas de los Sistemas Inteligentes

Práctica 1 - Entrega 2

José Pimentel Mesones Lothar Soto Palma Francisco David Charte Luque
José Carlos Entrena Jiménez

Contents

1. Tareas para mejorar el comportamiento reactivo del robot respecto al uso de campos de potencial (navegación local sin mapa).

a. Conseguir aumentar el campo de visión. El campo de visión del robot simulado es de 270º, ¿cómo hacer para que el escaneo laser sea mayor que el actualmente implementado?

En el archivo *myPlannerLite.h* hemos cambiado el valor de las constantes MIN_SCAN_ANGLE_RAD y MAX_SCAN_ANGLE_RAD a -135.0/180*M_PI y 135.0/180*M_PI respectivamente, para aumentar el ángulo de visión.

b. Conseguir que no se demore tanto tiempo en detenerse cuando esté lo suficientemente cerca del objetivo. ¿Por qué tarda tanto en detenerse cuando está próximo al objetivo? ¿Cómo solucionarlo?

Cuando calculamos la componente atractiva y nos encontramos dentro del campo de atracción (representado por la componente *spread*), el módulo de la velocidad es inversamente proporcional a la distancia al objetivo, por lo que cuanto más nos acercamos a él, más lento se mueve el robot.

Para solucionar este comportamiento, nuestra idea ha sido hacer constante el módulo de la velocidad cuando la distancia al objetivo es menor que la mitad del *spread* del objetivo. Así, estamos muy próximos al objetivo, la velocidad es constante y no se va reduciendo conforme nos acercamos, evitando que tarde tanto tiempo alcanzar el objetivo y detenerse.

c. Conseguir que el robot no tenga "comportamiento suicida", es decir, cuando está cerca de un obstáculo acelera y choca con él. ¿Cuál es la causa de este comportamiento suicida? ¿Cómo evitarlo?

Al encontrarnos muy próximos a un obstáculo, el cálculo de la componente repulsiva dará como resultado una velocidad lineal muy alta, consecuencia de querer alejarnos cuanto antes de dicho obstáculo. Sin embargo, como tratamos la velocidad lineal y la velocidad angular por separado, al aumentar la velocidad lineal el robot no tiene tiempo de girar antes de chocarse con el obstáculo, dando la sensación de que el robot se lanza hacia él.

Como solución, hemos decidido hacer 0 la velocidad lineal del robot cuando se encuentra con un obstáculo y tiene que girar para evitarlo. De esta forma, permitimos al robot tomar la dirección deseada antes de moverse, evitando así lanzarnos hacia el obstáculo.

d. Conseguir que, una vez que el robot se queda atrapado en una esquina (en un mínimo local) , trate de salir de esta situación. ¿Cómo conseguirlo sin utilizar memoria (es decir, información de un mapa)?

e. Conseguir suprimir las oscilaciones del robot. ¿Cuál es la causa de las oscilaciones? ¿Cómo eliminarlas en la mayor medida posible?

2. Tareas para mejorar el comportamiento del robot mediante técnicas de navegación local con mapa.

a. Contemplar el uso de distintos mapas para la experimentación. Los mapas pueden generarse a partir de una imagen mediante el paquete mapserver. Ver explicación sobre manejo de mapas y mundos simulados en la documentación adjunta.

b. Usar dos costmaps en el cliente: uno local configurado para tener una ventana activa que se desplace con el robot y otro global para tener una información sobre el costmap del mapa completo.

c. Usar el costmap local para poder encontrar una trayectoria local segura desde la pose actual y que finalice en un punto (en el que no haya colisión) de la frontera del costmap lo más próximo posible al objetivo. Esto se llevará a cabo mediante un proceso de búsqueda heurística teniendo en cuenta las siguientes fuentes de información: la distancia al objetivo, costes de las celdas del costmap local local y muestras del escaneo láser.

d. Mejorar el comportamiento del cliente en los siguientes aspectos:

- Detectar que el robot está "atascado" (demasiado tiempo en una región sin avanzar al objetivo), usando información de "feedback".

- En caso de atasco cancelar goal actual.
- Determinar un nuevo goal para sacarlo del atasco (usando el proceso de búsqueda local).
- Enviar el nuevo goal, detectar que se ha alcanzado, y volver a enviar el goal original.

e. Mejorar el comportamiento del servidor para ajustarse a los nuevos servicios que le va a requerir el cliente.