



DECSAI

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada



Técnicas de los Sistemas Inteligentes.

Curso 2015-16.

Práctica 1: Robótica

Entrega 3: Navegación global.

1.1 Objetivo

La tercera entrega de la Práctica1 consiste en implementar en ROS técnicas de navegación global para un robot móvil usando el simulador Stage. El objetivo es implementar el algoritmo A*, a partir del código proporcionado por el profesor, como un plugin de ROS que pueda ser ejecutado como “global planner” en el paquete *Move_base* (ver transparencias de la sesión 4 para más información). Tener en cuenta que el algoritmo encontrará y devolverá un camino solución entre los puntos inicial y destino y, al estar integrado en *Move_base*, este camino se enviará automáticamente al planificador local de *move_base*, usando el resto de funciones de este paquete. Por esta razón, es importante conocer qué parámetros de configuración deben tocarse para afinar el comportamiento (ver transparencias de la Sesión 4 para más información).

1.2 Tareas

El trabajo consistirá en modificar el código fuente proporcionado por el profesor (que incluye una versión incompleta del algoritmo A* preparada para ser integrada como un plugin de “global planner” en el paquete *Move_base*) realizando las siguientes tareas:

1. Por un lado, debe extenderse la actual implementación de A* para obtener el camino más corto y más seguro desde la posición actual del robot hasta la posición dada como objetivo a través de Rviz. Entendemos como camino más seguro aquél que garantiza en todo paso que la distancia del robot a los obstáculos es la más amplia posible. En el código fuente se proveen funciones que calculan la heurística en función de la



DECSAI

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada



distancia pero no consideran la seguridad del camino. Como se observa en el código, A* usa como representación las celdas de un global costmap de ROS, a partir de los valores de este costmap puede extraerse información para construir una heurística que incorpore información sobre la seguridad de la ruta.

2. Por otro lado, se pide también mejorar el algoritmo para tratar de reducir el tiempo en que tarda en encontrar una solución. Este es un requisito importante porque la implementación está integrada en una arquitectura para la navegación de un robot en tiempo real. En cualquier caso, tener en cuenta que es posible configurar los tiempos de respuesta esperados para el planificador local y el planificador global del paquete `move_base` (ver transparencias de la sesión 4). Para intentar reducir el tiempo de búsqueda puede usarse cualquier técnica explicada en teoría (incluyendo modificación de pesos para encontrar una solución rápida aunque sea subóptima) o alguna de las técnicas descritas en un blog muy referenciado sobre A* para juegos (que apunta técnicas totalmente aplicables a nuestro problema en robótica) . <http://theory.stanford.edu/~amitp/GameProgramming/>
3. Además, deberá llevarse a cabo una experimentación en al menos tres mundos distintos en Stage : `willow_garage`, `mi-simplerooms` (ambos ya disponibles en el material de esta y otras entregas) y cualquier otro mundo definido a partir de un mapa **representando un laberinto**, como el ilustrado en la Guía para usar Mapas de la anterior entrega. En la experimentación se deberá mostrar por cada uno de los mapas:
 - a. una ilustración de un camino encontrado, el tiempo que ha tardado en encontrar la solución y otras estadísticas que se consideren necesarias para mostrar el esfuerzo realizado en la búsqueda del camino.
 - b. Una descripción del proceso de navegación llevado a cabo , incluyendo diferentes pantallas, o bien, un enlace a un vídeo YouTube en el que se muestre un episodio de navegación completo del robot.
 - c. Si se realiza además la tarea 2 es necesario realizar experimentos en los que se muestre cómo las técnicas implementadas mejorar el comportamiento del A* implementado en la tarea 1.

1.3 Metodología de trabajo

La idea es que se trabaje en equipo, repartiendo adecuadamente las tareas entre los miembros del equipo.

1.4 Calificación del trabajo



Es obligatorio entregar al menos **las tareas 1 y 3** y una memoria en la que se describa claramente cómo se ha resuelto cada tarea y que incluya la experimentación.. No hay que incluir código fuente en la memoria, hay que describir clara y sintéticamente cómo se ha resuelto cada tarea y, si fuera necesario, especificar las técnicas adicionales utilizadas. Se valorará muy positivamente el uso de figuras o esquemas para apoyar la descripción. La memoria para esta entrega tendrá una extensión máxima de 7 páginas. La primera página debe incluir obligatoriamente un resumen de máximo 150 palabras explicando resumidamente qué se ha hecho y cómo se ha hecho. Se añadirá una página de título en la que se incluirán los nombres de los miembros del grupo.

- Calificación de navegación global (hasta 10 puntos):
 - Si no se realizan las tareas 1 y 3, la entrega se puntúa con un 0.
 - La calificación del trabajo realizado en las tareas 1y 3 estará en el intervalo [0,6] y dependerá del grado de consecución de las tareas, de las heurísticas implementadas y de la experimentación realizada.
 - La tarea 2 se puntuará en el intervalo [0, 4] dependiendo de las técnicas utilizadas y de la experimentación realizada para comprobar su efectividad.

La memoria se calificará Bien, Regular o Mal, según los criterios de estructuración y buen formato, organización de ideas en la redacción (incluyendo estructura gramatical y ortografía), además de capacidad de síntesis, claridad y facilidad de comprensión de lo escrito en la descripción. Una memoria Regular supone restar entre 1 y 2 puntos a la calificación obtenida en las otras tareas, una memoria calificada como Mal supone restar entre 3 y 4 puntos, a criterio del profesor.

1.5 Material a entregar

Hay que entregar un fichero comprimido que contenga la memoria en pdf y con una estructura de directorios y ficheros según un paquete ros estándar. El paquete tiene que haberse realizado de manera que su nombre corresponda con la entrega y con el subgrupo de prácticas. Basta con hacer, por ejemplo, `catkin_create_package E3SubGrupo1_3` para identificar la Entrega 3 de los miembros del subgrupo de prácticas 1.3. En ese paquete tienen que estar bien implementados, organizados y colocados todos los ficheros fuente, los ficheros launch, los ficheros de configuraciónl y los CMakeLists.txt necesarios para compilarlo.



DECSAI

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada



1.6 Fecha de entrega

GrupoLunes: Lunes 18 de Abril de 2016 a las 14:00.

GupoMartes: Lunes 18 de Abril de 2016 a las 14:00.