



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 2: Navegación local sin y con mapa.

1.1 Objetivo

La segunda entrega consiste en implementar en ROS técnicas de navegación local para un robot móvil usando el simulador Stage. Se desarrollarán técnicas de navegación sin mapa, basadas en campos de potencial, y con mapa, basadas en el uso de un costmap local y en una búsqueda heurística local sobre el entorno próximo del robot, en el espacio de posiciones libres del robot representadas en el costmap. Las técnicas de navegación estarán soportadas por una arquitectura cliente-servidor en la que

1. un nodo cliente enviará solicitudes de consecución de objetivos a un nodo servidor que las servirá
2. un nodo servidor recibirá las solicitudes y activará la navegación local del robot.

.

1.2 Tareas

El trabajo consistirá en modificar el código fuente proporcionado por el profesor (que incluye una versión básica de la arquitectura cliente-servidor y un navegador local de campos de potencial con un comportamiento básico). Los conceptos básicos de campos de potencial y el código fuente han sido explicados en la sesión de prácticas y las tareas a realizar para esta entrega son de dos tipos:



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 láser manejado sea mayor que el actualmente implementado?.
 - 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?.
 - c. Conseguir que el robot no tenga “comportamiento suicida”, e.d., cuando está cerca de un obstáculo acelera y choca con él. ¿Cuál es la causa del comportamiento suicida?. ¿Cómo evitarlo?.
 - d. Conseguir que, una vez que el robot se queda atrapado en una esquina (en un mínimo local), trate de salir de esa situación. ¿Cuál es la causa del mínimo local?. ¿Cómo conseguirlo sin utilizar memoria (e.d., 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:
 - i. Detectar que el robot está “atascado” (demasiado tiempo en una región sin avanzar al objetivo), usando información de “feedback”.
 - ii. En caso de atasco cancelar goal actual.
 - iii. Determinar un nuevo goal para sacarlo del atasco (usando el proceso de búsqueda local).



- iv. 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.

1.3 Metodología de trabajo

La idea es que se trabaje en equipo, repartiendo adecuadamente las tareas entre los miembros del equipo. Una posible organización de tareas entre miembros del equipo podría ser:

- 1 persona: 1.a 1.b 1.c 1.d y 1.e
- 1 persona: 2.b y 2.c
- 1 persona: 2.d y 2.e

Para las tareas 2.b y 2.c (uso de costmaps) se ha desarrollado una guía para la ayuda a la programación de costmaps en la arquitectura cliente-servidor.

Para las tareas 2d y 2.d (gestión de goals y detección de atascos) se ha desarrollado también una guía basada en ejercicios que deben realizarse para conseguir el objetivo de la tarea gradualmente.

1.4 Calificación del trabajo

Es obligatorio entregar al menos las tareas 1.a, 1.b y 1.c y una memoria en la que se describa claramente cómo se ha resuelto cada tarea. 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 el algoritmo usado e ilustrarlo con un ejemplo. Se valorará muy positivamente el uso de figuras o esquemas para apoyar la descripción. La descripción de las tareas 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 local sin mapa (hasta 5 puntos):
 - Si no se realizan las tareas 1.a, 1.b y 1.c, la entrega se puntúa con un 0.



- Si se realizan bien las tareas 1.a a 1.d, la puntuación de esta parte es un 4.
 - Si se realiza bien la tarea 1.e, la puntuación de esta parte es un 5.
- Calificación de navegación local con mapa (hasta 5 puntos)
 - Si se realizan las tareas 2.b a 2.e, la puntuación es 4 puntos.
 - Si se realizan bien las tareas 2.b y 2.c, la puntuación de esta parte es 2 puntos.
 - Si se realizan bien las tareas 2.d y 2.e, la puntuación de esta parte es 2 puntos
 - Si se realiza bien la tarea 2.a, a la puntuación obtenida en las otras tareas se suma 1 punto.

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 E2SubGrupo1_3` para identificar la Entrega 2 de los miembros del subgrupo de prácticas 1.3. En ese paquete tienen que estar bien implementados y colocados todos los ficheros fuente y los `CMakeLists.txt` necesarios para compilarlo.

1.6 Fecha de entrega

GrupoLunes: Jueves 31 de Marzo de 2016 a las 14:00.

GupoMartes: Jueves 31 de Marzo de 2016 a las 14:00.