Actividad 2

Control de movimiento usando odometría

En esta segunda actividad instalaremos las herramientas necesarias para las simulaciones que se realizarán durante el curso. En la primera actividad nos comenzamos a familiarizar con el robot con el que se realizarán actividades, en esta segunda instalaremos el software necesario para su realización.

2.1. Sistema operativo y compilador

Casi todos los modelos de robots de MobileRobots viene equipados con ordenadores que tiene instalado el sistema operativo Linux, concretamente la distribución Debian. Por lo que este SO y esta distribución son las preferibles para realizar las actividades y es para la que el equipo docente puede ofrecer mayor soporte. También está disponible para la distribución Ubuntu como paquete deb, por lo que también se puede usar esta distribución en lugar de Debian. Para otras distribuciones de Linux existen archivos comprimidos que es necesario compilar. El compilador usado es gcc, preferiblemente en versión 4.3 o superior.

En Windows también están disponibles los archivos necesarios. El compilador que hay que usar es Visual C++.

Para más información sobre estos temas Librería Aria.

2.2. Descarga e instalación

Hay que descargar la librería Aria, su última versión para Linux es la 2.9.4, para Debian de 64 bits se pueden usar sin problemas los paquetes de Ubuntu, desde la carpeta Software externo de Alf.

Para la instalación usando dpkg hay que tener permisos de administrador, ver comando sudo, el comando para instalar desde el directorio donde tengamos el fichero descargado es:

```
sudo dpkg - i libaria_2.9.4+ubuntu16_amd64.deb
```

Esto crea el directorio /usr/local/Aria. A continuación vamos a ese directorio y editamos, con permisos de superusuario, el fichero Makefile cambiando la línea 843 de

obj rust csharp csharpExamples \

a

obj \

Ahora compilamos Aria

La librería incluye una documentación interesante que además de detallar la API de la librería contiene en las secciones iniciales una explicación detallada de su funcionamiento que es muy recomendable leer.

También hay que descargar el simulador MobileSim, su última versión para Linux es la 0.7.3. La versión original presenta ciertos problemas y que han sido corregidos por el equipo docente, los cambios se describen en el fichero cambios.txt en la carpeta principal de MobileSim-src-0.7.3-new. Por esto es imprescindible descargar y compilar la nueva versión de MobileSim que podéis encontrar en la carpeta Software de los Documentos Públicos MobileSim-src-0.7.3-new.tar.gz. Para compilar e instalar seguid las instrucciones del fichero README.src.txt en la carpeta principal de MobileSim-src-0.7.3-new, compilaréis el programa usando la versión de Aria que ya tengáis instalada.

Una vez descargados e instalados la librería y el simulador, conviene hacer una prueba inicial. Lanzar el simulador y cargar un mundo (.map) de los contenidos en <directorio-Aria>examples, en Linux suele ser /usr/local/Aria/examples, o preferiblemente de los suministrados por el equipo docente.

A continuación lanzar el programa "demo" de ese mismo directorio. Probar a mover el robot usando las teclas de flecha y explorar otras posibilidades del programa.

En Linux los directorios donde se instalan los ejemplos de no tiene escritura permitida para un usuario normal (no root), por lo que conviene dar permisos de escritura al usuario normal.

2.3. Actividad evaluable

Con la opción en el simulador File->Export->Sequence of Frames, se puede exportar gráficamente la trayectoria seguida por el robot.

Utilizando el mundo suministrado por el equipo docente, cuadrado6x6.map, realizar las siguiente actividades:

1. Hacer que el robot describa un cuadrado de cuatro metros de lado. Colocar el robot inicialmente en una de las esquinas del cuadrado a describir. El control de la distancia recorrida se realizará ordenando al robot recorrer 4 m., girar 90 grados, etc, consultar la clase ArRobot de la librería.

   a) Probar primero sin errores de odometría. Para ello en el fichero <directorio-MobileSim>/PioneerRobotModels.world.inc cambiar (aprox. línea 36,38) odom_error [0.0 0.0 0.0] y localization "odom" por localization "gps". Con esta configuración el robot debe describir un cuadrado perfecto.

b) Volver al odom_error[0.01 0.01 0.020] original y localization "odom" y repetir el experimento ahora el cuadrado no será perfecto, sino que aparecerán cuadrados ligeramente desplazados.

2. Mover el robot en un círculo de dos metros de radio.

El robot es posible moverlo indicando las velocidades lineal y angular, o las velocidades de las ruedas de cada lado, ver la documentación de ArRobot. Las ecuaciones que relacionan ambas formas son las de dirección diferencial o con los métodos de ArRobot move(), setHeading(), isMoveDone() e isHeadingDone(), ver en Alf directMotionExample.cpp en la carpeta Documentos públicos/Software.

Posicionamiento de robot en un punto distinto del origen original del mapa. Para situar el robot en el punto que se desee del mapa hay que indicarlo en el mapa que se vaya a usar, en la línea que comienza con Cairn: RobotHome colocaremos la pose del robot. p. ej. con Cairn: RobotHome -2000 2000 0.000000 "" ICON "Home" (las longitudes en mm. y en ángulo en grados) situamos el robot a 2 m. a la izquierda (-2000) del centro y a 2 m. arriba del centro, con heading 0 grados (mira a la derecha). En PioneerRobotModels.world.inc podemos hacer coincidir el origen de la odometría con ese nuevo origen, localization_origin [-2 2 0] (ahora en metros y grados), con la orden anterior situamos el origen de la odometría en el nuevo Home.

2.4. Envío

Hay que enviar el código correspondiente a las dos actividades descritas en la sección anterior mediante la plataforma Alf.