

INSTRUCCIONES PARA INSTALAR Y EJECUTAR EL PAQUETE MI_MAPEO_STAGE

1) Descargar el fichero comprimido `mi_mapeo_stage.zip`. Descomprimirlo en el directorio `src` de un espacio de trabajo (que a partir de ahora llamaremos `<workspace>`).

2) Hacer `catkin_make` en `<workspace>`

3) Hacer `<workspace>/source devel/setup.sh`

4) Hacer `rospack find mi_mapeo_stage`, para comprobar que se ha actualizado la información de entorno de ROS.

5) Ejecutar `roslaunch mi_mapeo_stage mi_gmapping.launch`.

Este fichero launch lanza

- `stage`, un simulador 2D más ligero que `gazebo`, recibe como argumento el mundo en formato stage “corridor.world”, que se encuentra en el directorio `stage_config/worlds` del paquete `mi_mapeo_stage`.
- un nodo para hacer mapeo (`gmapping`),
- otro nodo para que el robot pueda navegar (`move_base`), que se explicará en la siguiente sesión
- el nodo `rviz` para visualización, que recibe como argumento el fichero de configuración `single_robot_markers.rviz` que se encuentra en el directorio `mi_mapeo_stage`.

6) Comprobar y observar que se levantan las ventanas de `stage` y de `rviz`. En PRADO hay un video de demostración, **video_mi_mapeo_stage**, en el que se visualiza este proceso.

7) Teleoperar el robot con el paquete `teleop-twist-keyboard`. Comprobar si está instalado el paquete con

```
rospack find teleop-twist-keyboard.
```

Si no está, instalarlo con el comando

```
sudo apt-get install ros-kinetic-teleop-twist-keyboard
```

Ejecutar `roslaunch teleop_twist_keyboard teleop_twist_keyboard.py`

y teleoperar el robot como hemos visto en clase. Observar cómo se va actualizando el mapa en `rviz` conforme movemos el robot.

8) A continuación ejecutaremos el nodo `ros` que viene implementado en el código fuente del paquete `mi_mapeo_stage` (directorio `src` del paquete). Este código fuente ya está compilado porque hemos ejecutado `catkin_make` previamente.

Ejecutar `roslaunch mi_mapeo_stage mi_mapeo_stage`

Este nodo tiene un comportamiento muy simple: en un bucle sin fin con una frecuencia de 0.2 Hz, escribe el mapa recibido en un fichero (grid.txt), muestra en pantalla la posición del robot recibida desde la odometría y espera hasta que en cada iteración se garantice el tiempo de ciclo. Observar cómo va cambiando la información sobre la posición del robot mostrada por este nodo a medida que teleoperamos el robot.

La descripción del código fuente se encuentra en el documento **explicación_código_fuente**.