Resumen Robótica Móvil - Comandos básicos

Comandos Habituales

- Hacer sources terminal

```
source /opt/ros/humble/setup.bash
export LDS_MODEL=LDS-01
export TURTLEBOT3_MODEL=burger
export ROS_LOCALHOST_ONLY=1
```

- Gazebo

```
# Buit
ros2 launch turtlebot3_gazebo empty_world.launch.py

# Casa
ros2 launch turtlebot3_gazebo turtlebot3_house.launch.py

# Si no va
source /usr/share/gazebo/setup.sh
```

- Mover robot

```
ros2 run turtlebot3_teleop teleop_keyboard
```

- rqt

```
rqt
```

- rviz

```
# Buscar configuració adecuada
rviz2 -d config.rviz
```

Robots físicos

```
ssh ubuntu@192.168.0.xxx     // password: turtlebot
source /opt/ros/humble/setup.bash
export ROS_LOCALHOST_ONLY=0
export ROS_DOMAIN_ID=30            # El que siga
export LDS_MODEL=LDS-01
export TURTLEBOT3_MODEL=burger
ros2 launch turtlebot3_bringup robot.launch.py
```

- (El resto se puede hacer en cualquier PC, importante el rosdomain y el localhost quitado. (Usando Ethernet wired: ver diapositiva)

Rosbags

```
ros2 bag record /clock /odom /tf /tf_static /scan
ros2 bag record -o <rosbag_name> /clock /odom /tf /tf_static /scan
ros2 bag info <rosbag_name>
ros2 bag play <rosbag_name>
ros2 topic echo /odom --no-arr
```

```
Topics habituals:
/clock /map /odom /scan /tf /tf_static
/amcl_pose /particle_cloud /robot_description
```

Buscar concrets per a Navigation...

Maps

2 archivos -> YAML y imagen (pixeles blancos...)

Blanco -> Libre
Negro -> Ocupado
Gris -> Desconocido

```
#./visualize_map.bash
rviz2 -d config_map.rviz

#./publish_map.bash
ros2 launch map_server.launch.py
```

ROS tf2 library and tools

ROS tf2 library and tools

```
ros2 run tf2_tools view_frames.py
ros2 run tf2_tools view_frames.py --output_filename my_frames.pdf
```

Escolta al topic `/tf`. Després de això, exportarà un PDF

```
ros2 run tf2_ros tf2_echo [source_frame] [target_frame]
```

```
At time 9532.274000000
- Translation: [x, y, z]
- Rotation: in Quaternion [qx, qy, qz, qw]
```

Que torna en temps real la transformació de `[source_frame]` a `[target_frame]`

```
ros2 run tf2_ros static_transform_publisher \
  --x x --y y --z z \
  --qx qx --qy qy --qz qz --qw qw \
  --frame-id source_frame \
  --child-frame-id target_frame
```

Mapping

Comandos Mapping - SLAM Methods

Ver: Resumen Robótica Móvil - Comandos básicos

- Saving map
  - `ros2 run nav2_map_server map_saver_cli -f ./map_name`
  - Por defecto se guarda en formato PGM, para ponerlo en imagen:
  - `ros2 run nav2_map_server map_saver_cli --fmt png -f ./map_name`
- Rosbag
  - `ros2 bag record /clock /map /map_updates /odom /robot_description /scan /scan_matched_points2 /submap_list /tf /tf_static`
  - Y para comprobarlo una vez grabado:
  - `rviz2 -d /opt/ros/humble/share/turtlebot3_cartographer/rviz/tb3_cartographer.rviz`
- Algoritmos en RViZ

```
ros2 launch turtlebot3_cartographer cartographer.launch.py use_sim_time:=True
```

```
ros2 launch slam_toolbox online_async_launch.py use_sim_time:=True
```

```
ros2 launch rtabmap_demos turtlebot3_scan.launch.py use_sim_time:=True
```

Navigation

Comandos Navigation

```
ros2 launch turtlebot3_navigation2 navigation2.launch.py \
use_sim_time:=True map:=Navigation/map9.yaml
```

```
ros2 bag record /clock /tf /tf_static /map /robot_description /scan /particle_cloud /plan /local_plan /waypoints
/mobile_base/sensors/bumper_pointcloud /global_costmap/costmap /global_costmap/costmap_updates
/global_costmap/voxel_marked_cloud /downsampled_costmap /downsampled_costmap_updates /local_costmap/costmap
/local_costmap/costmap_updates /local_costmap/published_footprint /local_costmap/voxel_marked_cloud
```

```
rviz2 -d /opt/ros/humble/share/nav2_bringup/rviz/nav2_default_view.rviz
```

Webots

```
sudo apt install ros-humble-webots-ros2
source /opt/ros/humble/setup.bash
export ROS_LOCALHOST_ONLY=1
export WEBOTS_HOME=$HOME/webots-R2023b
ros2 launch webots_ros2_turtlebot robot_launch.py
```

Otras cosas

Práctica Maps

```
launch_tb3_empty_sim.bash
# Importar model mapa
visualize_robot.bash
publish_map.bash
publish_transform.bash
# Teleop
```

Localization in simulation

!!! Cada bloque de código es una terminal, no ejecutar los scripts, hacer en todos los sources nada más entrar:

```
source /opt/ros/humble/setup.bash
export ROS_LOCALHOST_ONLY=1
export TURTLEBOT3_MODEL=burger
export ROS_DOMAIN_ID=92313
```

Obrir el turtlebot3 en un món buit

```
source /usr/share/gazebo/setup.sh

ros2 launch turtlebot3_gazebo \
     empty_world.launch.py
```

Afegir el mapa

Obrir el algoritme de localització

```
ros2 launch amcl.launch.py \
  use_sim_time:=True \
  map:=../TI_n1_edited.yaml
```

Obrir el rviz2 amb les configuracions

```
rviz2 -d config_amcl.rviz
```

Estimar la pose 2D del robot en el rviz2

Moure el robot un poc amb el teclat

```
ros2 run turtlebot3_teleop teleop_keyboard
```

Grabar el bagfile en els topics necessaris

```
ros2 bag record /clock /map /odom /scan /tf /tf_static /amcl_pose /particle_cloud /robot_description
```

Verificar tancant tot, i fent els 2 comandos:

```
rviz2 -d config_amcl.rviz
ros2 bag play...
```