Resumen Robótica Móvil - Comandos básicos

Comandos Habituales

- Sources en cada terminal

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

- Compilación

```
colcon build --symlink-install --packages-select ...
source install/setup.bash
```

- 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

```
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
```

Per a comprimir

Usarem -a per a grabar tots els topics, en cas de ja haver gravat un rosbag anteriorment i volem grabar tot el que estava gravat. El tamany el diem en bytes, si volem que la compressió ocupe menys de 100MB aprox, haurem de fer que sense comprimir siga menys que 1GB aprox.

1 GB = 1,073,741,824 bytes.
1 MB = 1,048,576 bytes.

```
ros2 bag record -a --max-bag-size 1073741824
ros2 bag record -a -b 1073741824
```

Buscar topics concrets per a Navigation

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
```

Ara ja es poden fer ros2 bags, rviz, i tot

Maps

(Ver GitHub: 01-Worlds)

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

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

- **Resolution**
  Ver diapositiva

  In a map YAML file, the resolution is expressed in `[meters] / [pixel]`

  Example:

  - Paper size: A1, vertical (594 × 841 mm)
  - Scale: 1/125
  - Width image: 4678 px
  - Height image: 6623 px

  Resolution
  - horizontal: 0.594 * 125 / 4678
  - vertical: 0.841 * 125 / 6623

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

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

- `./publish_transform.bash` para ajustar que coindican map y odom

Práctica Maps

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

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]
```

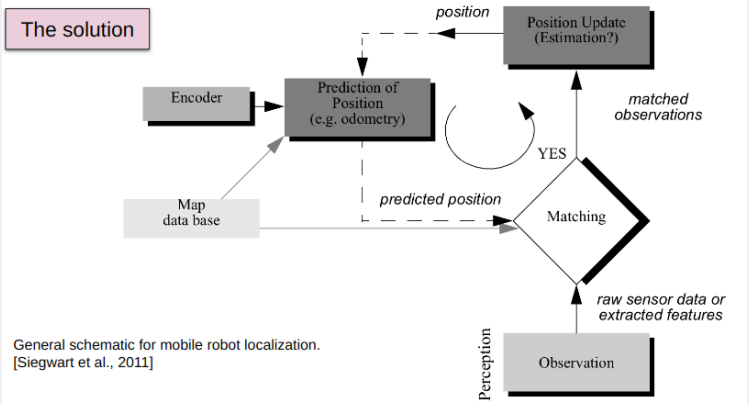
Que torna en temps real la transformació de `[source_frame]` a `[target_frame]`

```
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
```

Localization - AMCL

Consisteix en, donat un mapa ja existent, localitzar la posició del robot en aquest usant la combinació dels diferents sensors, evitant així el drift causat per sols utilitzar el /odom

The solution



General schematic for mobile robot localization.
[Siegwart et al., 2011]

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...
```

Mapping - SLAM methods

SLAM és localization and mapping, consisteix en crear un mapa de l'entorn al mateix temps que es calcula la posició del robot en aquest, usant la combinació de sensors.

- Comparison of SLAM methods with Gazebo - Google Docs
  - RTAB-Map para entornos 3D y donde se quiera precisión, se usa información RGB además de la lidar.
  - SLAM Toolbox para entornos 2D, es el más rápido y sencillo aunque puede ser menos preciso.
  - Cartographer es un punto medio, tanto 2D como 3D, y computacionalmente intermedio a los dos.

Comandos Mapping - SLAM Methods

- 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

El localization and mapping s'utilitza per a navegar en zones on els obstacles canvien constantment.
Nagivation consisteix en la planificació de rutes (global vs local)

Important actualitzar els arxius del root:

```
sudo sed -i 's/differential/nav2_amcl::DifferentialMotionModel/g' \
  /opt/ros/humble/share/turtlebot3_navigation2/param/burger.yaml

sudo sed -i 's/differential/nav2_amcl::DifferentialMotionModel/g' \
  /opt/ros/humble/share/turtlebot3_navigation2/param/waffle.yaml

sudo sed -i 's/differential/nav2_amcl::DifferentialMotionModel/g' \
  /opt/ros/humble/share/turtlebot3_navigation2/param/waffle_pi.yaml
```

```
turtlebot3_navigation2 / navigation2.launch.py
├── nav2_bringup / bringup_launch.py
│   ├── nav2_bringup / localization_launch.py
│   │   ├── nav2_map_server / map_server
│   │   ├── nav2_amcl / amcl
│   │   └── nav2_lifecycle_manager / lifecycle_manager
│   └── nav2_bringup / navigation_launch.py
│       ├── nav2_controller / controller_server
```

```
│       ├── nav2_planner / planner_server
│       ├── nav2_recoveries / recoveries_server
│       ├── nav2_bt_navigator / bt_navigator
│       ├── nav2_waypoint_follower / waypoint_follower
│       └── nav2_lifecycle_manager / lifecycle_manager
└── rviz2 -d nav2_bringup/rviz/nav2_default_view.rviz
```

Comandos Navigation

Con lo siguiente abrimos un mapa ya creado con mapping y podremos indicarle los objetivos al robot.

Primero estimaremos la pose 2D para tenerlo más o menos, y luego ya le daremos el objetivo, y se irá rectificando sobre la marcha.

```
ros2 launch turtlebot3_gazebo turtlebot3_house.launch.py

ros2 launch turtlebot3_navigation2 navigation2.launch.py \
use_sim_time:=True map:=/home/diego/Documents/Universidad/3rCurso/IR2121-ROBOTICA-MOBIL/10-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
```

```
ros2 run patrolling_task patrolling_task
```

Otras cosas

Veure posició robot:

- `ros2 topic echo /odom --no-arr`