Simulating TurtleBot3 AMR on Different Custom Worlds for the Purpose of Data Collection and Record

Prepared by Belkacem Bekkour

1 Running willowgarage.world with TurtleBot3:

For this purpose we have to write our **turtlebot3_willowgarage.launch** **custom launch file** for the **TurtleBot3 AMR**, which must include the right path for our **.world** file.

Terminal 0: Setup and Directory Navigation

Command 1: Attempt to launch Gazebo ROS

Description: This command tries to access the ‘launch‘ directory within Gazebo ROS. Since ‘launch‘ is a directory, it results in an error.

```
/opt/ros/noetic/share/gazebo_ros/launch
```

Output

```
bash: /opt/ros/noetic/share/gazebo_ros/launch: Is a
   directory
```

Command 2: Attempt to access TurtleBot3 Gazebo worlds

Description: This command tries to access the ‘worlds‘ directory for TurtleBot3 in Gazebo, but since it’s also a directory, it results in an error.

```
/opt/ros/noetic/share/turtlebot3_gazebo/worlds
```

Output

```
bash: /opt/ros/noetic/share/turtlebot3_gazebo/worlds:
   Is a directory
```

Command 3: List available Gazebo worlds

Description: Lists all available Gazebo world files in the specified directory.

```
ls /usr/share/gazebo-11/worlds
```

Output

```
actor_bvh.world
   nested_multilink_shape.world
actor.world                                 openal.world
animated_box.world                          ortho.world
animation_tension.world                     osrf_elevator.world
attach_lights.world                         pioneer2dx_camera.
   world
...
zephyr_demo.world
```

Command 4: Find ‘willowgarage.world‘ file

Description: Searches for the ‘willowgarage.world‘ file in the ‘/usr/share‘ directory.

```
find /usr/share -name "willowgarage.world"
```

Output

```
/usr/share/gazebo-11/worlds/willowgarage.world
```

Terminal 1: Source and Launch Simulation

Command 1: Source the setup file

Description: Sources the setup file to configure the environment for the project.

```
source ~/Desktop/process_LiDAR_data/project_garage/
    devel/setup.bash
```

Command 2: Set TurtleBot3 model to 'burger'

Description: Sets the TurtleBot3 model to 'burger' as an environment variable, which is required for launching specific simulations.

```
export TURTLEBOT3_MODEL=burger
```

Command 3: Launch TurtleBot3 in Willow Garage world

Description: Launches TurtleBot3 in a predefined Willow Garage environment using a custom launch file.

```
roslaunch turtlebot3_custom_launch
    turtlebot3_willowgarage.launch
```

Output

```
... logging to /home/hedi/.ros/log/4239abf8-a120-11ef
    -8a8e-4926e3b2f764/roslaunch-hedi-HP-ProDesk-600-G5
    -MT-319569.log
Checking log directory for disk usage. This may take a
     while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://hedi-HP-ProDesk-600-G5
    -MT:41981/

SUMMARY
========

PARAMETERS
```

```
 * /robot_description: <?xml version="1....
 * /rosdistro: noetic
 * /rosversion: 1.17.0
 * /use_sim_time: True

NODES
  /
    gazebo (gazebo_ros/gzserver)
    gazebo_gui (gazebo_ros/gzclient)
    spawn_urdf (gazebo_ros/spawn_model)

auto-starting new master
process[master]: started with pid [319593]
ROS_MASTER_URI=http://localhost:11311
...
[ INFO] [1731434555.399829344, 0.262000000]: DiffDrive
   (ns = //): Advertise odom on odom
[spawn_urdf-4] process has finished cleanly
log file: /home/hedi/.ros/log/4239abf8-a120-11ef-8a8e
   -4926e3b2f764/spawn_urdf-4*.log
```

Terminal 2: Teleoperation Control

Command 1: Set TurtleBot3 model to 'burger'

Description: Sets the TurtleBot3 model to 'burger', necessary for running specific teleop simulations.

```
export TURTLEBOT3_MODEL=burger
```

Command 2: Launch teleoperation for TurtleBot3

Description: Launches a teleoperation node for controlling TurtleBot3 via keyboard.

```
roslaunch turtlebot3_teleop turtlebot3_teleop_key.
   launch
```

Output

```
... logging to /home/hedi/.ros/log/4239abf8-a120-11ef
    -8a8e-
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://hedi-HP-ProDesk-600-G5
    -MT:42075/

SUMMARY
========

PARAMETERS
 * /model: burger
 * /rosdistro: noetic
 * /rosversion: 1.17.0

NODES
  /
    turtlebot3_teleop_keyboard (turtlebot3_teleop/
        turtlebot3_teleop_key)

ROS_MASTER_URI=http://localhost:11311

process[turtlebot3_teleop_keyboard-1]: started with
    pid [321122]

Control Your TurtleBot3!
---------------------------
Moving around:
        w
   a    s    d
        x

w/x : increase/decrease linear velocity (Burger : ~
    0.22, Waffle and Waffle Pi : ~ 0.26)
a/d : increase/decrease angular velocity (Burger : ~
    2.84, Waffle and Waffle Pi : ~ 1.82)
space key, s : force stop

CTRL-C to quit
```

Terminal 3: Launch Gazebo with RViz

Command 1: Set TurtleBot3 model to 'burger'

Description: Sets the TurtleBot3 model to ‘burger‘, which is used in the Gazebo RViz simulation.

```
export TURTLEBOT3_MODEL=burger
```

Command 2: Launch TurtleBot3 in Gazebo with RViz

Description: Launches TurtleBot3 in a Gazebo simulation environment along with RViz for visualization.

```
roslaunch turtlebot3_gazebo turtlebot3_gazebo_rviz.
   launch
```

Output

```
SUMMARY
========

PARAMETERS
 * /robot_description: <?xml version="1....
 * /robot_state_publisher/publish_frequency: 50.0
 * /robot_state_publisher/tf_prefix:
 * /rosdistro: noetic
 * /rosversion: 1.17.0

NODES
  /
    robot_state_publisher (robot_state_publisher/
       robot_state_publisher)
    rviz (rviz/rviz)

ROS_MASTER_URI=http://localhost:11311

process[robot_state_publisher-1]: started with pid
   [321493]
process[rviz-2]: started with pid [321494]
```

2 GMapping willowgarage.world with TurtleBot3

Terminal 1: Launch Gazebo with TurtleBot3 in Willow Garage World

Description: In this terminal, we start Gazebo with the Willow Garage world and TurtleBot3.

Command 1: Source the setup file

Description: Sources the setup file to configure the environment for the project.

```
source ~/Desktop/process_LiDAR_data/project_garage/
    devel/setup.bash
```

Command 2: Set TurtleBot3 model to ’burger’

Description: Sets the TurtleBot3 model to ‘burger‘ as an environment variable, which is required for launching specific simulations.

```
export TURTLEBOT3_MODEL=burger
```

Command 3: Launch TurtleBot3 in Willow Garage world

Description: Launches TurtleBot3 in a predefined Willow Garage environment using a custom launch file.

```
roslaunch turtlebot3_custom_launch
    turtlebot3_willowgarage.launch
```

Terminal 2: Run GMapping for SLAM

Description: This terminal runs the GMapping SLAM node to create a map as the TurtleBot3 navigates through the Willow Garage world.

Command 1: Source the setup file

Description: Sources the setup file to configure the environment for the project.

```
source ~/Desktop/process_LiDAR_data/project_garage/
    devel/setup.bash
```

Command 2: Set TurtleBot3 model to ’burger’

Description: Sets the TurtleBot3 model to ‘burger‘.

```
export TURTLEBOT3_MODEL=burger
```

Command 3: Start the GMapping SLAM node

Description: This command launches the GMapping SLAM node to process lidar data and build the occupancy grid map.

```
roslaunch turtlebot3_slam turtlebot3_slam.launch
   slam_methods:=gmapping
```

Output

```
... logging to /home/hedi/.ros/log/6239dbb2-a130-11ef
   -8a8e-3936d2e7f764/roslaunch-hedi-HP-ProDesk-600-G5
   -MT-329123.log
Checking log directory for disk usage. This may take a
    while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.
started roslaunch server http://hedi-HP-ProDesk-600-G5
   -MT:42981/
SUMMARY
========

PARAMETERS
 * /slam_methods: gmapping
 * /use_sim_time: True
 * /rosdistro: noetic
 * /rosversion: 1.17.0

NODES
  /
    slam_gmapping (turtlebot3_slam/turtlebot3_slam)

ROS_MASTER_URI=http://localhost:11311

process[slam_gmapping-1]: started with pid [329210]
```

Terminal 3: Teleoperation Control

Description: This terminal is used to control the TurtleBot3 manually via keyboard teleoperation.

Command 1: Source the setup file

Description: Sources the setup file to configure the environment for the project.

```
source ~/Desktop/process_LiDAR_data/project_garage/
   devel/setup.bash
```

Command 2: Set TurtleBot3 model to ’burger’

Description: Sets the TurtleBot3 model to ‘burger‘, necessary for running teleoperation.

```
export TURTLEBOT3_MODEL=burger
```

Command 3: Launch teleoperation for TurtleBot3

Description: Launches a teleoperation node for controlling TurtleBot3 via keyboard.

```
roslaunch turtlebot3_teleop turtlebot3_teleop_key.
   launch
```

Terminal 4: Visualize Mapping in RViz (Optional)

Description: This terminal is used to visualize the mapping process in RViz.

Command 1: Source the setup file

Description: Sources the setup file to configure the environment for the project.

```
source ~/Desktop/process_LiDAR_data/project_garage/
   devel/setup.bash
```

Command 2: Set TurtleBot3 model to ’burger’

Description: Sets the TurtleBot3 model to ‘burger‘, which is used in RViz for visualization.

```
export TURTLEBOT3_MODEL=burger
```

Command 3: Launch RViz to visualize the map

Description: Launches RViz to visualize the SLAM map as TurtleBot3 moves.

```
roslaunch turtlebot3_slam turtlebot3_slam_rviz.launch
```

Output

```
started roslaunch server http://hedi-HP-ProDesk-600-G5
   -MT:44875/

SUMMARY
========

PARAMETERS
 * /robot_description: <?xml version="1....
 * /rosdistro: noetic
 * /rosversion: 1.17.0

NODES
  /
    rviz (rviz/rviz)

ROS_MASTER_URI=http://localhost:11311

process[rviz-1]: started with pid [331213]
```

Terminal 5: Save the Occupancy Grid Map

Description: This terminal is used to save the generated SLAM map.

Command 1: Source the setup file

Description: Sources the setup file to configure the environment for the project.

```
source ~/Desktop/process_LiDAR_data/project_garage/
   devel/setup.bash
```

Command 2: Save the map

Description: This command saves the generated map files (‘willowgarage_map.pgm‘ and ”willowgarage_map.yaml”) to the Desktop.

```
rosrun map_server map_saver -f ~/Desktop/
   willowgarage_map
```

Output

```
[ INFO] [1731435550.123]: Saving map to ~/Desktop/
   willowgarage_map
[ INFO] [1731435551.456]: Map saved as
   willowgarage_map.pgm and willowgarage_map.yaml
```

2.1 NOTE:

- If you need to adjust the lidar range, modify the **URDF file**. Look for the tag `<gazebo reference="base_scan">` and change `<max>3.5</max>` under the `<range>` tag to, for example, `<max>15.5</max>`.

```
cd /opt/ros/noetic/share/turtlebot3_description/urdf
```

- To set the range for the occupancy grid in the GMapping Algorithm, you’ll need to create a custom **GMapping .launch** that includes the following:

```
maxRange: 15.5            # Match this to the lidar
   range
minimumScore: 200
delta: 0.05
maxUrange: 15.5           # Maximum usable range for
   mapping
map_update_interval: 2.0  # Update map every 2 seconds
```

- To create a modified launch file that uses custom parameters for GMapping, follow these steps. This new launch file will set each required parameter explicitly and launch TurtleBot3 in your custom environment.
  - **Navigate to Your Workspace:** Go to the launch directory in your workspace or create one if it doesn't exist:

    ```
    cd ~/Desktop/process_LiDAR_data/project_garage/src/
       turtlebot3_custom_launch/launch
    ```

  - **Create the New Launch File:** Open a new under the following path **process_LiDAR_data/project_garage/src/turtlebot3_custom_launch** file named `turtlebot3_gmapping_custom.launch`:

    ```
    nano turtlebot3_gmapping_custom.launch
    ```

  - **Add the Following Launch Code:** Paste the following XML code into `turtlebot3_gmapping_custom.launch`:

    ```
            <launch>
              <!-- Arguments -->
              <arg name="model" default="burger" doc="
                 model␣type␣[burger,␣waffle,␣waffle_pi]"/>

              <!-- TurtleBot3 Bringup -->
              <include file="$(find␣turtlebot3_bringup)/
                 launch/turtlebot3_remote.launch">
                <arg name="model" value="$(arg␣model)" />
              </include>

              <!-- GMapping Node with Custom Parameters --
                 >
              <node pkg="gmapping" type="slam_gmapping"
                 name="slam_gmapping" output="screen">
                <param name="scan" value="/scan"/>
                <param name="maxRange" value="15.5"/>
                <param name="maxUrange" value="15.5"/>
                <param name="delta" value="0.05"/>
                <param name="linearUpdate" value="0.1"/>
                <param name="angularUpdate" value="0.05"/>
                <param name="map_update_interval" value="
                   2.0"/>
                <param name="lskip" value="1"/>
              </node>

              <!-- RViz (optional) -->
    ```

```
          <node pkg="rviz" type="rviz" name="rviz"
             args="-d␣$(find␣turtlebot3_slam)/rviz/
             turtlebot3_gmapping.rviz" />
        </launch>
```

- This file launches:
  - **TurtleBot3 bringup:** Starts up the basic TurtleBot3 nodes.
  - **GMapping:** Configured with custom parameters for maximum range, map resolution, and update settings.
  - **RViz:** Automatically opens RViz with the GMapping configuration to visualize the map.
- **Save and Close the File:** Press `Ctrl + X`, then `Y`, and `Enter` to save and exit.
- **Running the New Launch File:** Follow these steps to launch TurtleBot3 with GMapping using the new parameters.
  - **Terminal 1: Launch Gazebo with Willow Garage World**

    ```
    source ~/Desktop/process_LiDAR_data/project_garage/
       devel/setup.bash
    ```

    ```
    export TURTLEBOT3_MODEL=burger
    ```

    ```

    roslaunch turtlebot3_custom_launch
       turtlebot3_willowgarage.launch
    ```

    This terminal launches the Willow Garage world with TurtleBot3 in Gazebo.
  - **Terminal 2: Launch GMapping with Custom Parameters**

    ```
    source ~/Desktop/process_LiDAR_data/project_garage/
       devel/setup.bash
    ```

    ```
    export TURTLEBOT3_MODEL=burger
    ```

    ```

    roslaunch turtlebot3_custom_launch
       turtlebot3_gmapping_custom.launch
    ```

    This terminal launches GMapping with your specified parameters, as well as RViz for visualizing the map.
  - **Terminal 3: Teleoperation Control (Optional)** — Use this terminal to control the TurtleBot3 manually.

```
source ~/Desktop/process_LiDAR_data/project_garage/
   devel/setup.bash
```

```
export TURTLEBOT3_MODEL=burger
```

```
roslaunch turtlebot3_teleop turtlebot3_teleop_key.
   launch
```

- **Terminal 4: Save the map** This will genrate a **.pgm** file followed by the **.yaml** configuration file for the full occupancy grid map

```

source ~/Desktop/process_LiDAR_data/project_garage/
   devel/setup.bash
```

```

rosrun map_server map_saver -f ~/Desktop/
   willowgarage_map
```