

ROS Navigation

ROBOTICS PROJECT AA 2016/2017

Emiliano Gagliardi
Emanuele Ghelfi

August 4, 2017

1 Package structure

The directory `willy3_2dnav` is the main directory and contains the following folders:

- `launch`: contains the launch files.
- `nav_stack_conf`: contains the configuration files for the navigation stack.
- `map`: contains the map generated with `gmapping`. The generated maps are inside the folder “gen” (generated). The perfect map is inside the `map` folder.
- `robot_conf/models` contains the `sdf` file for the description of the robot and the sensor.

1.1 Launch files

Inside the folder `launch` there are two launch files:

- `launch_tf.launch` is the launch file that spawns the `willy3` model and publishes the transformation between the laser sensor and the base link using the node `static_transform_publisher`.
- `willy3_nav.launch` is the launch file required to configure the move base by assigning the path of the configuration files and remapping the topic `scan` to `laser_scan`. This launch file includes also the `amcl_diff.launch` for the navigation and the map server.

1.2 Navigation stack configuration

The file inside `nav_stack_conf` directory are used by the `move_base` node. These files are taken by the ros tutorial and adjusted for the “willy3” robot. The robot is assumed circular with radius of 0.4 since it hasn’t a particular shape and the obstacles are inflated by 0.4. The robot is not holonomic since it’s a differential drive.

2 Installation

The directory `willy3_2dnav` must be linked inside the `catkin_ws` workspace and the `robot_conf/models` must be linked inside the `.gazebo/models` folder:

```
cd ~/catkin_ws/src
ln -s path_to_ros-navigation-folder/willy3_2dnav
cd ..
catkin_make
cd ~/.gazebo/models
ln -s path_to_ros-navigation-folder/willy3_2dnav/robot_conf/models/willy3
ln -s path_to_ros-navigation-folder/willy3_2dnav/robot_conf/models/hokuyo_ros
```

3 Mapping

The world willowgarage is large and complicated. In order to map it we've used two different approaches.

In the first approach we've driven the robot around the world using rviz to visualize the creation of the map.

Unfortunately the resulting map has a shifted part but with more time available the resulting map should be better.



Figure 1: Map of willowgarage

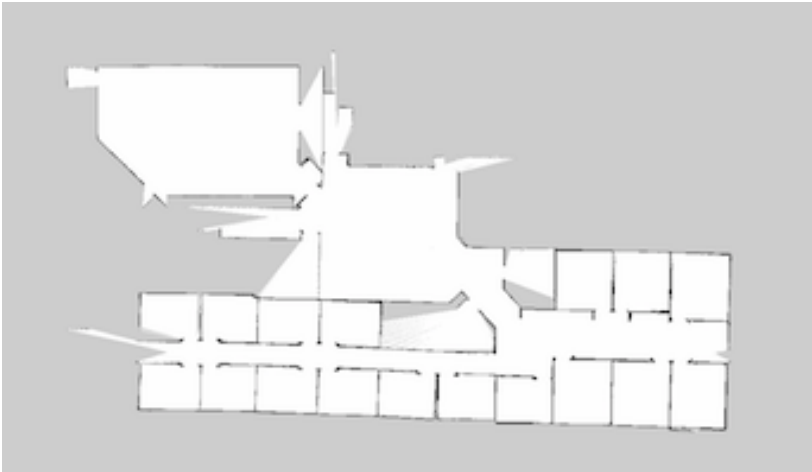
The commands in order to run the mapping process are:

```
roslaunch willy3_2dnav launch_tf.launch
roslaunch gmapping slam_gmapping scan:=laser_scan
roslaunch joy joy_node # for the use of the joystick
roslaunch teleop_twist_joy teleop_node
roslaunch rviz rviz # map visualization
```

After having obtained a good map:

```
roslaunch map_server map_saver -f <map_name>
```

The other approach is using the navigation stack coupled with gmapping for driving the robot in an autonomous way. The robot can only use the local planner since the map is not completed. Unfortunately this process is slower than the other and we haven't obtained a complete map (since it requires a lot of time).



The commands are:

```
roslaunch willy3_2dnav launch_tf.launch
roslaunch willy3_2dnav willy3_nav.launch
roslaunch gmapping slam_gmapping scan:=laser_scan
roslaunch rviz rviz # map visualization
```

4 Navigation

For the navigation part the commands are:

```
roslaunch willy3_2dnav launch_tf.launch
roslaunch willy3_2dnav willy3_nav.launch
roslaunch rviz rviz # map visualization
```

Using rviz it's possible to open the rviz configuration file inside the main folder that contains the topics already configured.

By sending goals using the rviz graphical interface the robot should move to the desired location.

5 Reference

- http://wiki.ros.org/slam_gmapping/Tutorials/MappingFromLoggedData
- <http://wiki.ros.org/navigation/Tutorials/RobotSetup>
- Github repository: <https://github.com/EmilianoGagliardiEmanueleGhelfi/ros-navigation>