Pathing - Knowing where your robot is going

In the previous assignment, we got to move the TurtleBot around the factory. If you got everything to work, this should have been fun to watch!

In this next assignment, we will start using the autonomous navigation stack and see the robot moving by itself and we will visualize the plans that the TurtleBot is using to navigate.

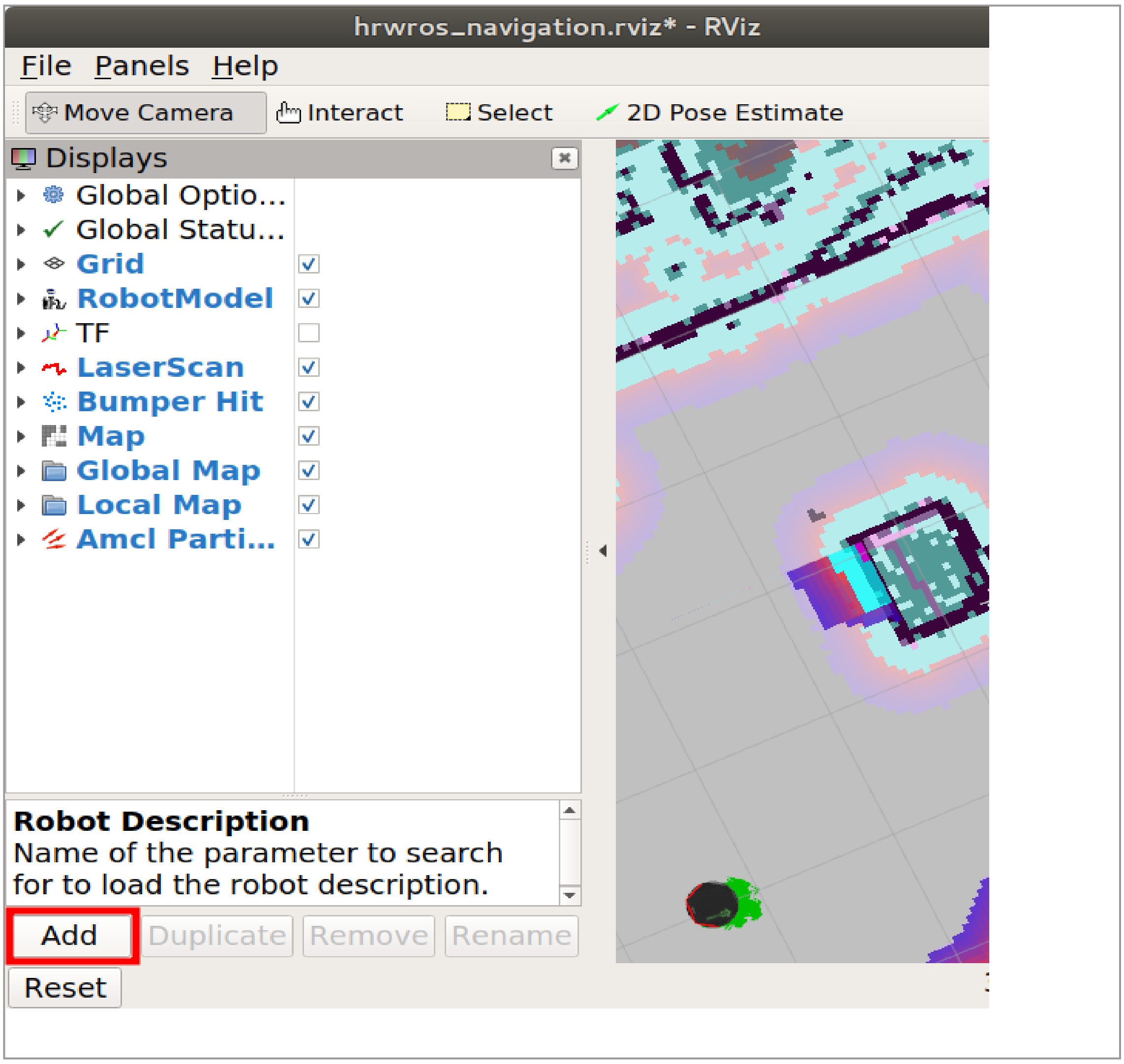
It is assumed here that Gazebo Simulations are still running, as the RViz and and the AMCL navigation. If you closed them, just relaunch them with the instructions of assignment 1.

Week 3 - Assignment 2 - part 1 (of 2) --- 2 Points

In this part you will visualize the Full Path planned by the Navigation stack in RViz by adding so-called path elements.

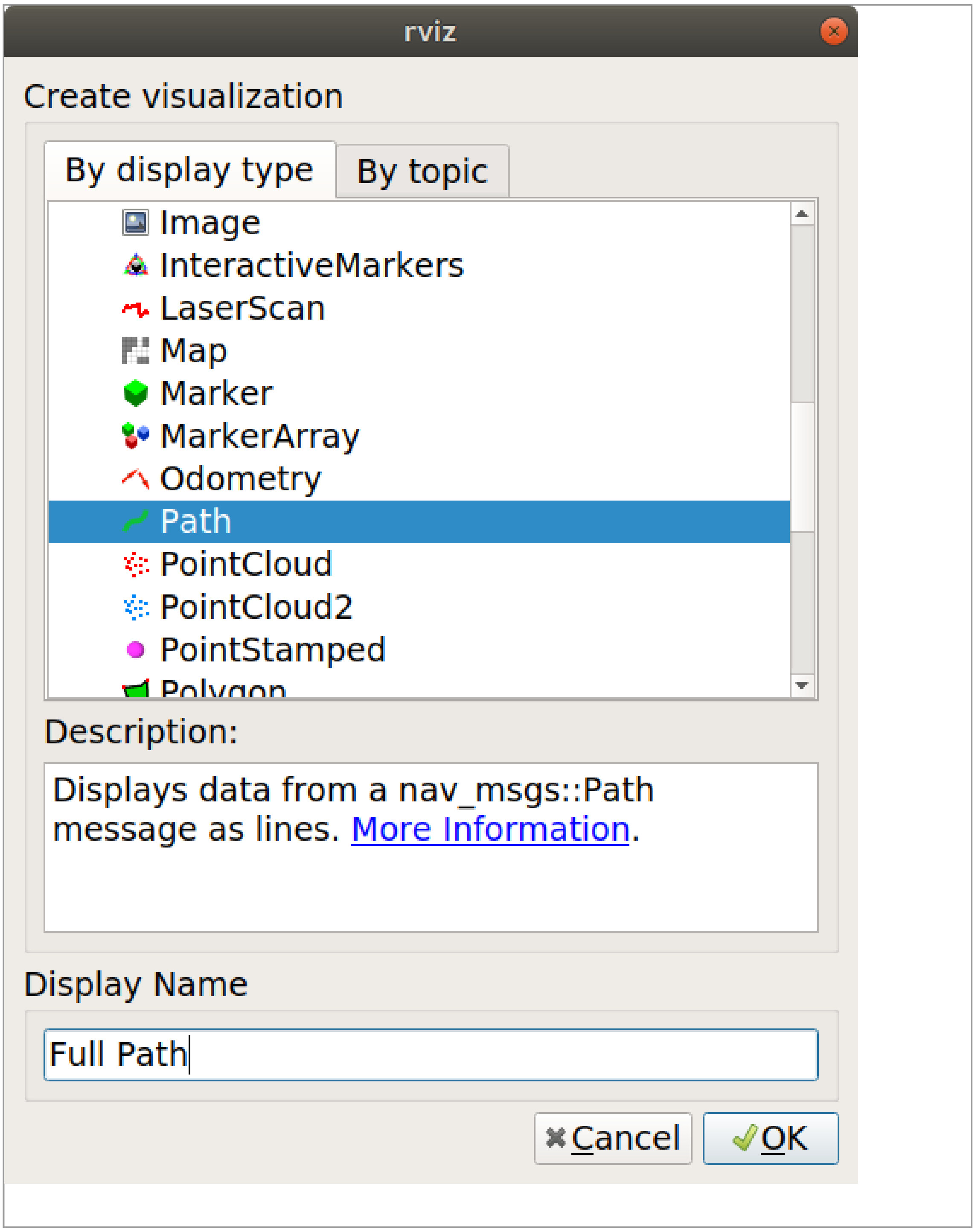
You can do that with the following steps:

Step1: At the bottom of the left panel in RViz, you can see a button marked 'Add'. Below is a screenshot with the Add button marked red:



Robot Description
Name of the parameter to search for to load the robot description.

Step2: After clicking on the Add button, select the Path display type like in the screenshot below, and change the display name to `Full Path`



Displays data from a nav_msgs::Path message as lines. More Information.

Full Path

Step 3: Point the Path display to the topic it will use to get the full path. Navigate to the topic field, and select `/move_base/NavfnROS/plan.`

- This topic will visualize the overall global path planned by the navigation stack using Dijkstra's algorithm.
- If you set a new navigation goal in RViz, a green line will appear, showing you the path the Turtlebot intends to take.

Step 4: To visualize this path, give the robot a goal with a long path, for example from near robot 1 to near robot 2, so it is clearly visible.

Remember to save the RVIZ configuration with `File -> Save config` **or** `Ctrl+S`

Week 3 - Assignment 2 - part 2 (of 2) --- 2 Points

We can now see the Full Path planned by toward the goal. But ROS navigation stack also provides more than just the 'pre-planned' full path.

The ROS navigation stack also provides two more plans: the first one, called Global Plan, is an implementation of the DWA local planner for planning around unknown and dynamic obstacles! That is, the path can also be dynamically modified, when previously unknown obstacles appear. For example, there is a new crate blocking its path, it will attempt to drive around it. The second one is called Local Plan, and it's a very short plan that corresponds to the path that the TurtleBot will follow in the next couple of seconds, and is used to actually compute the motion control of the wheels in that short period.

In this part of the assignment, you will visualize also create new path displays for those two plans, just follow these steps.

Step 1: Add a new path display and subscribe it to `/move_base/DWAPlannerROS/global_plan`. Change the color from green to blue (0; 0; 255), and the name to `Global Plan.`

- Notice how this `Global Plan` is different from the overall `Full Path` when the TurtleBot starts navigating.

Step 2: Add another path display, in this case for the `/move_base/DWAPlannerROS/local_plan`, Set its color to red (255; 0; 0) and name it `Local Plan`.

Step 3. Give the robot a new navigation goal and look at the displayed paths.

- You should now be able to visualize all three paths, full, global and local.
- You might need to zoom in a little to view the local plan, as it is very short.

Remember to save the RVIZ configuration with `File -> Save config` **or** `Ctrl+S`

This completes assignment 2. And only one last assignment to go!