

Navigation Stack

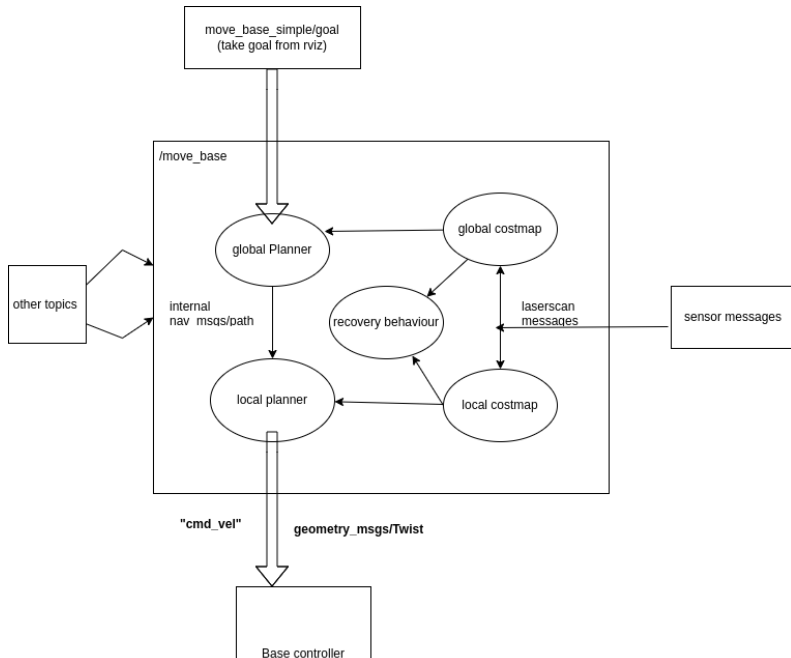
Prajwal Thakur

2022-11-03

Navigation Stack

- ▶ A 2D navigation stack that takes in information from odometry, sensor streams, and a goal pose and outputs safe velocity commands that are sent to a mobile base .

Overall picture



rqt_graph

- ▶ `rqt_graph` : `rqt_graph` provides a GUI plugin for visualizing the ROS computation graph (basically shows the relation between different nodes and topics)
- ▶ command : `roslaunch rqt_graph rqt_graph`
- ▶ more info at link : [roswiki_rqt_graph](#)

Notice

- ▶ topics represented by : rectangular / square box
- ▶ nodes represented by : ellipse shape
- ▶ topics subscribed by /move_base node (represented by the incoming arrows in the pic at slide2)
- ▶ topics published by /move_base node (represented by the out going arrows in the pic at slide2)

Move Base

- ▶ In ROS 1 we use the Move Base node to link together a global and local planner to accomplish its global navigation task.

How does the Planning work in ROS?

- ▶ In ROS we generate two types of planners, first, we generate a Global planner from the information obtained from the static environment second Local Planner.

Global Planner

- ▶ Global planner :
 - ▶ A new goal received by the move_base node is sent to the global planner. The global planner is in charge of finding a safe **path/waypoints** to the goal pose i.e. by avoiding obstacles. It does find a collision-free path from the start to the goal while skipping the dynamic constraints. This path is calculated before the robot starts moving, so it will not take into account the readings that the robot sensors are doing while moving towards the goal.
- ▶ more info at : [roswiki_global_planner](#)

Global Planner node publishes Only One topic :

- ▶ message type :

nav_msgs/Path Message

File: `nav_msgs/Path.msg`

Raw Message Definition

```
#An array of poses that represents a Path for a robot to follow
Header header
geometry_msgs/PoseStamped[] poses
```

Compact Message Definition

```
std_msgs/Header header
geometry_msgs/PoseStamped[] poses
```

Local Planner

- ▶ Local planner :
 - ▶ After the global planner has calculated some initial path to follow, the path is sent to a local planner. The local planner will execute a small segment of the global plan. Therefore given a global plan (provided by the global planner) and obsacle position the local planner will compute a kinematic and dynamic feasible path and send the velocity commands in order to move the robot.
- ▶ Unlike the global planner, the local planner considers the odometry and the laser data and chooses a collision-free local plan for the robot. So, the local planner can recompute the robot's path on the fly in order to keep the robot from striking objects, while still allowing it to reach its destination.

Global vs local Planner Demo

- ▶ Demo

Costmap

- ▶ A costmap is a grid map where each cell is assigned a specific value or cost: higher costs indicate a smaller distance between the robot and an obstacle. Path-finding is done by a planner which uses a series of different algorithms to find the shortest path while avoiding obstacles.

Cost map

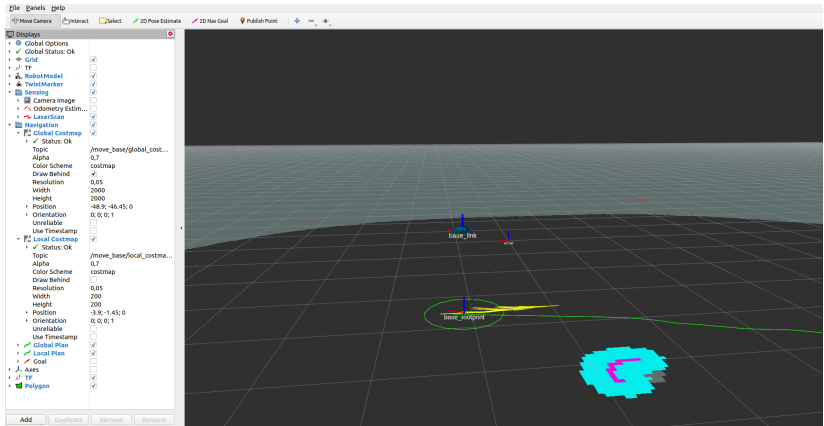
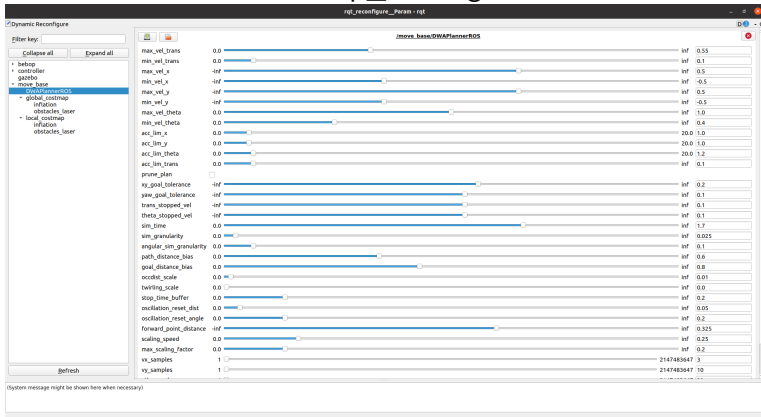


Figure 3: srrviz3

rqt_reconfigure

- ▶ command : `roslaunch rqt_reconfigure rqt_reconfigure`
- ▶ more info at link : [roswiki rqt_reconfigure](#)



rqt_reconfigure

- ▶ for testing the different set of parameters , you can change the parameters from rqt and observe the change in behavior of the robot in real time .
- ▶ changing in rqt changes the parameter in ros server
- ▶ rqt doesnt save the parameters automatically
- ▶ Once satisfied , note down the parameters and write in appropriate yaml files