Mapping in ROS

Wyatt Newman
February, 2015

Introduction: Robot navigation with respect to a map requires creation of a map, plus a means to "localize" the robot with respect to the map (i.e. "you are here"). ROS provides some tools (ros nodes) that help with the creation of a map using the robot (simulated or real) to publish odometry and laserscans topics while exploring the environment. Initially, the exploration may be performed with user assistance, e.g. teleoperating the robot, or publishing incremental motion commands, or simply pushing the robot through the environment to be mapped. The process will be described here using our "cwruBot" model robot with Lidar. To help drive the robot through an environment, an interactive robot commander node is introduced.

Interactive Robot Commander Node: The program "interactive_robot_commander.cpp" in the package "example_robot_commander" illustrates a (primitive) means to command the robot to new positions and orientations. This process assumes that an interactive marker is being published (using: example_interactive_marker IM_example3, which publishes to topic /path_end/update).

To start up the system using cwruBot, in separate terminals, do:
1) start roscore
 1.5) rosparam set use_sim_time true
2) rosrun gazebo_ros gazebo
3) roslaunch cwru_urdf cwruBot.launch
4) insert a Gazebo model, e.g. the starting pen
5) rosrun example_interactive_marker IM_example3
6) rosrun example_robot_commander interactive_robot_commander
7) rosrun rviz rviz
Make sure an interactive-marker item is included in Rviz, with the topic set to /path_end/update.

One can now:
8) move the interactive marker
9) trigger motion with the command with: rosservice call trigger_path_goal 1

The interactive commander will cause the robot to move in three phases:
1) spin in place to face the marker coordinates
2) move forward towards the marker coordinates
3) spin in place to align with the marker orientations

Steps 8 and 9 may be repeated to move the robot around its environment.

Recording data to make a map: We may use "rosbag" to record data for a robot journey, which may be re-played and post-processed. For the purpose of map making, we will need to record the data published on topics /laser/scan and /tf (a topic that carries "transform" messages). This is performed by invoking:
10) rosbag record -O newMapData /laser/scan /tf
where "newMapData" is a chosen name for the file that will hold the recording.

While the logger is running, the robot should be controlled to explore its environment. This should use slow motions (particularly for rotation), and the robot should spin intermittently, and it should get

nearby scans of all surfaces of interest.

When the chosen exploration journey is done, the above processes can all be killed (including stopping the rosbag recording).

Converting a bag file to a map: Another ROS node can be used to convert exploration data into an occupancy map (in 2-D). To do so, we can replay the recorded journey while running the converter. Make sure "roscore" is still running (but not Gazebo). Then run:

11) rosrun gmapping slam_gmapping scan:=/laser/scan
    In the above, "laser/scan" is the name of the topic to which cwruBot publishes its laser data. For other systems (e.g. Jinx), choose the corresponding laser data topic.
12) play back your recording:
    rosbag play --clock newMapData
    In the above, "newMapData" is the exploration log performed in step 10. You will need to navigate to the corresponding directory to refer to this recording.

13) rosrun rviz rviz (w/ map topic displayed) (to watch progress of map building)
wait for playback to finish

14) save map:
rosrun map_server map_saver -f newMap
where "newMap" is the name for the new map being created (choose something more mnemonic).

Using a map:

15) repeat steps 1-7 above.
    (except this time run interactive_robot_commander_v2)
16) rosrun map_server map_server newMap.yaml (from the directory containing the map)
17) rosrun amcl amcl scan:=/laser/scan
18) add "map" visualization item to rviz, topic= /map
19) in rviz, set the fixed frame to "map"
20) add poseArray view on /particlecloud topic
21) click 2d Pose Estimate button in rviz and set initial pose estimate

Rviz will display its "cloud" of candidate pose estimates as red arrows. You can initialize the pose with the 2D Pose Estimate tool, and input motion commands with the 2D Nav Goal tool. If the correct transform from map coordinates to odom coordinates has been found, then Lidar data will align with the map's occupied cells.