Implementation of a Robot Behavior Learning Simulator.

Kushagra Singh Bisen

Ecole des Mines de Saint Etienne

kushagrasingh.bisen@etu.emse.fr

June 1, 2021

Recap

- There was a discussion upon the log file and it's Implementation.
- I discussed about the path-planning algorithms and their usage in our work.

Today's Agenda

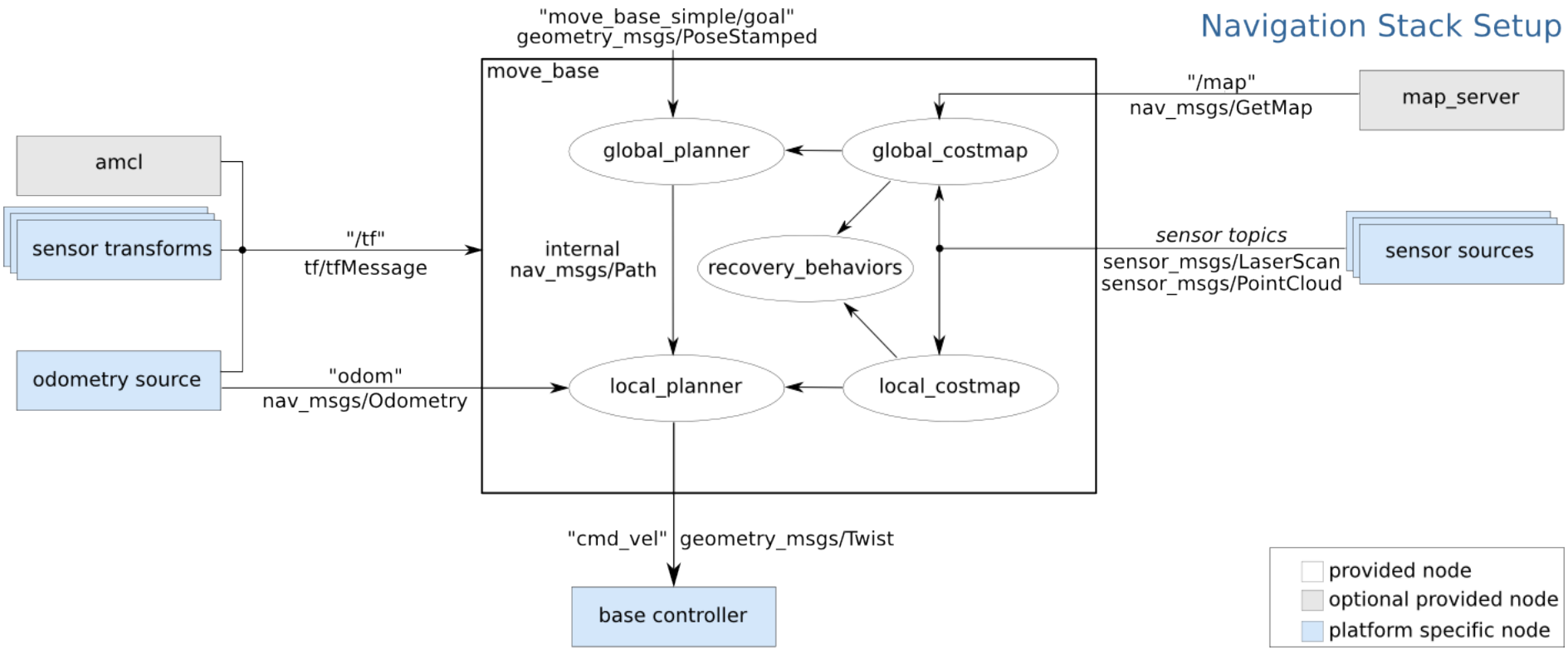
Today we will see how the complete path planning done completely by the robot. I implemented a Dijkstra path planning algorithm, in which I used the CostMap2D to decide to cost.

Navigation

Navigation in ROS is defined as process through which a robot is moved from the current node to the goal node. The step-by-step process for initiation of the planning is,

- Development of a World in Gazebo.
- SLAM (Localization of the Robot and the Mapping of the World).
- After SLAM is done, save the generated map. (Note : The map is generated as a PGM (portable graymap) file. The Map is very different to the World.)
- The Goal Node is given and the Robot uses the Dijkstra's Algorithm to find the shortest path towards the Goal.

Navigation Stack



Algorithm Implemented (Dijkstra)

As the algorithm is pretty famous in computer science for planning the path from a current node to the goal node. The algorithm is implemented in our case is,

- Initial Node has the g-cost of 0, and it is added to the open-list.
- Repeat the following while open-list is not empty:
  - Extract the node with the smallest g-cost from open-list and call it current-node
  - Mark it as visited by adding it into closed-list
- If the current-node is the GOAL, trace back the path. Otherwise, we fimd the neighbours of the current node (3*3 grid neighbours as described by Prof. Nida's Comments)
- If the node has already been visited, skip it.
- If the node has not been visited and not in the open-list, check if the cost is less than the current cost we have. If yes, update the cost and the parent node.
- If the node has not been visited, set the cost, set the parent node and add it to the open-list.

Initiating the Simulation.

Starting the World in Gazebo

roslaunch unit2_pp simulation_unit2.launch

Starting the Path Planning

roslaunch unit2_pp unit2_solution.launch

I will now proceed demonstrating the simulation.

Key Takeaways

- With the previous 'obstacle-avoidance' simulation, I was not able to do the grid based modelling of the scenario as it is required.
- Due to this approach, generating log file is easier (I hope). Because, as in the algorithm the node calculates the path successively of the neighbours. I also think I will be able to make the use of /twist/ topic in ROS to output the Rotation.
- I tried to use the inherant csv file generation by ROS log file, but our usecase is different thus I will have to write a seperte node to subscribe to specific topics and then log it into a file.
- I have a better way to model the World with the boundaries and squares being exactly described the way they should be. I will continue working with the present map that I made for the current log file simulation.
- The present simulation is ready to be pushed to the Gitlab repository of EMSE.

Thank you for your time.