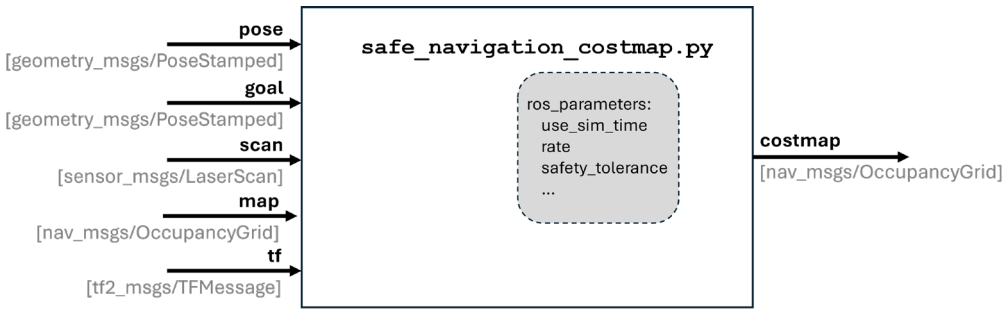
4TM00 Robot Motion Planning and Control
Assignment 2
Search-Based Path Planning & Safe Path Following

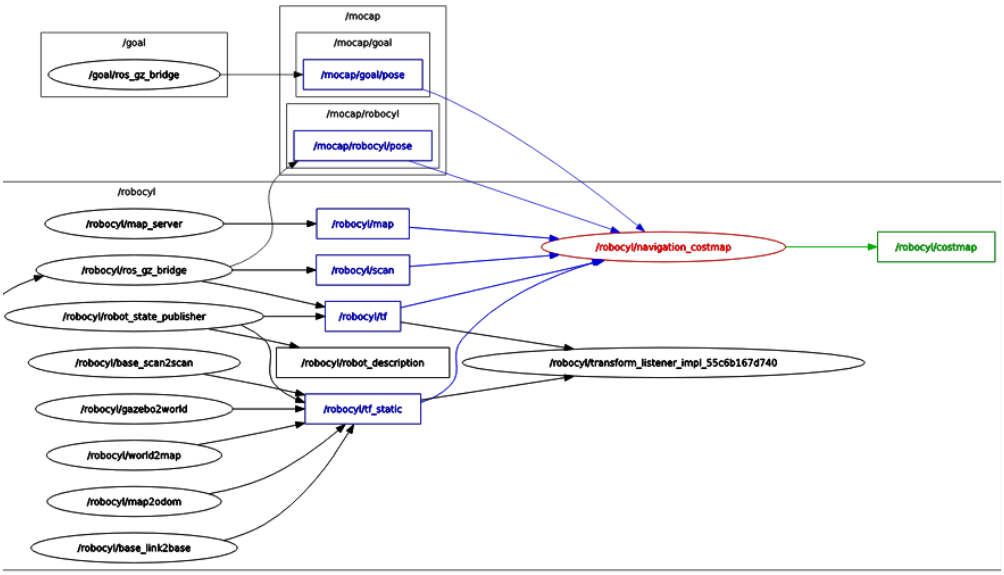
In this assignment, you will work with the RoboCyl mobile robot in Gazebo to reach a global goal position by following a minimum-cost navigation path over a given occupancy grid map, addressing three navigation tasks: safe navigation costmap design, search-based optimal path planning over costmaps, and safe path-following control around sensed obstacles. You may use any relevant software available in the 4TM00 course's GitLab repository.
A specific ROS package, *core_tue4tm00_assignment2*, is provided in the GitLab repository for this assignment. It includes template files for your ROS node scripts, parameter configurations, and demo launch files. Simply copy and paste *core_tue4tm00_assignment2* into your group's GitLab repository and rename it with your group name, e.g., *group0_tue4tm00_assignment2*. Please ensure that your final demo code runs properly with the 4TM00 course software.

Part 1) *(Safe Navigation Costmap)* In the first part of the assignment, you are asked to design a ROS node that receives an occupancy grid map and returns a safe navigation costmap for safe reference path planning. This may involve using the robot's scan measurements, pose, goal pose, and available coordinate transformations (i.e., tf) to ensure safety around obstacles. A schematic drawing of the ROS node and an example rqt_graph are provided below for your reference.

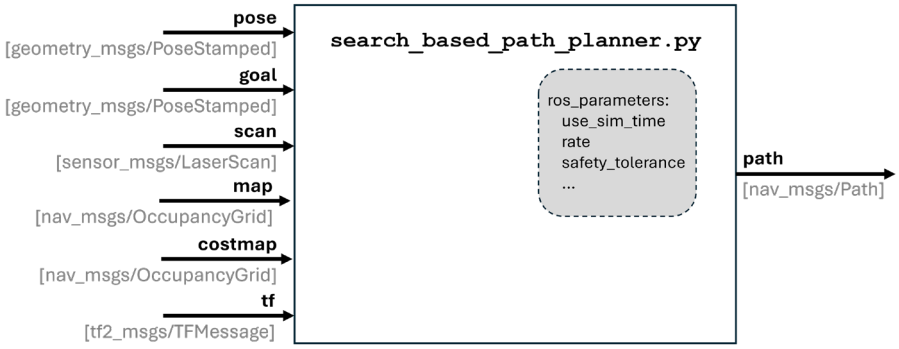


The ROS assignment package, *core_tue4tm00_assignment2*, includes:

- A ROS node script template: *scripts/safe_navigation_costmap.py*
- A parameter configuration file: *config/safe_navigation_costmap.yaml*
- Its interface launch file: *launch/safe_navigation_costmap.launch.py*
- A demo launch file: *launch/demo_safe_navigation_costmap.launch.py*

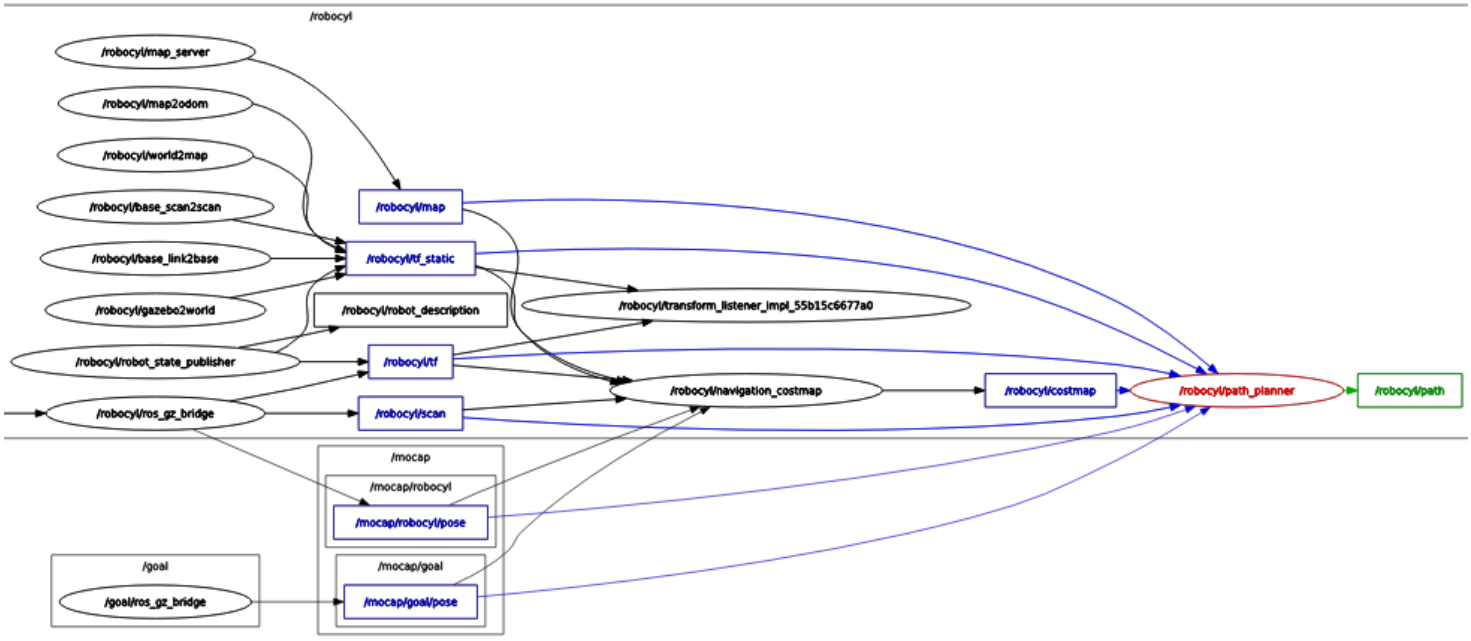


Part 2) *(Search-Based Path Planning)* In the second part of the assignment, you are asked to design a ROS node that receives a goal position and generates a safe navigation path over an occupancy grid map. The path should guide the robot toward the goal while avoiding obstacles and minimizing navigation cost. This this may involve using the robot's scan measurements, pose, costmap, and available coordinate transformations (i.e., tf). It is important to note that it may not always be possible to reach all given goal positions in a complex environment, and executing reference paths that are too close to obstacles can be challenging. Therefore, your objective is to find a reference path with adequate clearance from obstacles; otherwise, indicate that no safe navigation path exists. A schematic drawing and an example *rqt_graph* of the ROS node are provided below for your reference.

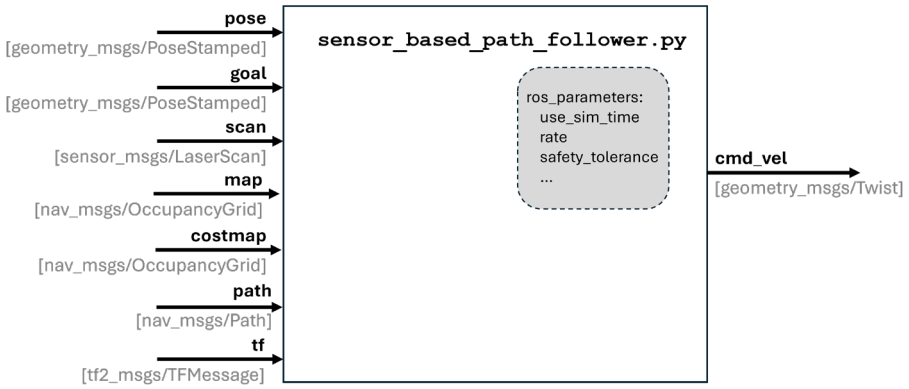


The ROS assignment package, *core_tue4tm00_assignment2*, includes:

- A ROS node script template: *scripts/search_based_path_planner.py*
- A parameter configuration file: *config/search_based_path_planner.yaml*
- Its interface launch file: *launch/search_based_path_planner.launch.py*
- A demo launch file: *launch/demo_search_based_path_planner.launch.py*

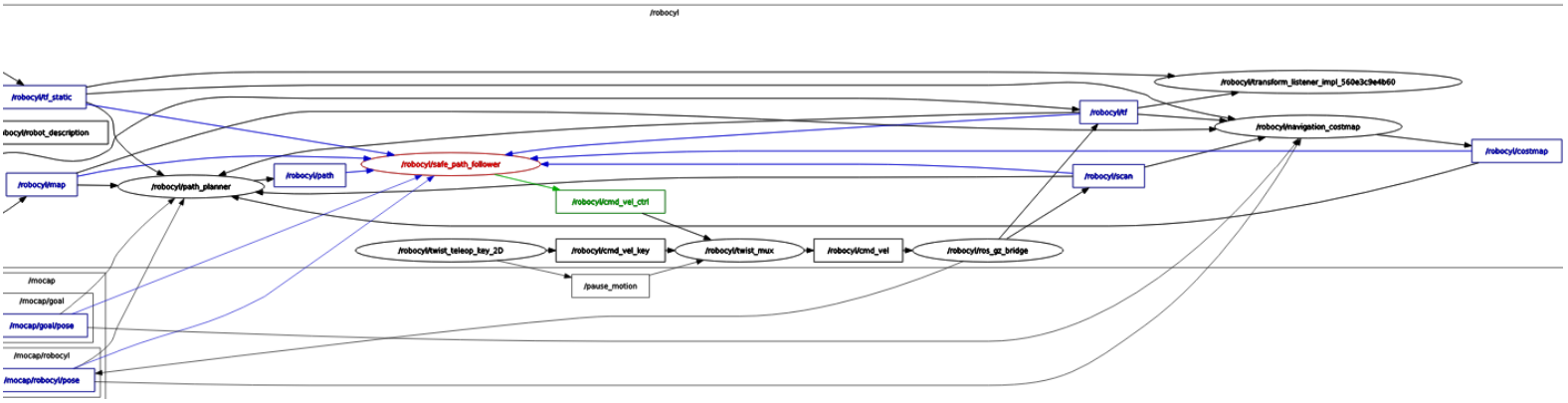


Part 3) *(Sensor-Based Path Following)* In the third part of the assignment, you are asked to design a ROS node that receives a reference path toward a global goal position and generates command velocities to safely follow it while avoiding sensed obstacles. This can be achieved using the robot's scan measurements, pose, goal pose, the occupancy map of the environment, the navigation costmap, and available coordinate transformations (i.e., tf). It is important to note that it may not be possible to safely follow all reference paths from start to end in complex environments. Therefore, your objective is to follow the path as much as possible. If the path cannot be followed safely, you may indicate that replanning is needed (e.g., by printing a message on the terminal). A schematic drawing and an example rqt_graph of the ROS node are provided below for your reference.



The ROS assignment package, *core_tue4tm00_assignment2*, includes:

- A ROS node script template: *scripts/sensor_based_path_follower.py*
- A parameter configuration file: *config/sensor_based_path_follower.yaml*
- Its interface launch file: *launch/sensor_based_path_follower.launch.py*
- A demo launch file: *launch/demo_sensor_based_path_follower.launch.py*



For all parts, please clearly describe your design approach and reasoning in the report. Your solution approach should be clear, concise, and justified. Please systematically demonstrate the pros and cons of your design in your report and presentation. Ensure that any figures and videos included are neat (e.g., with captions, proper labels, readable axes, and adequately thick lines). Provide all the necessary information to reproduce your results. The report should be concise and to the point. Your code should be clean, well-commented, and as modular as possible to ensure reusability by you and other. Please make sure that your code runs properly with the 4TM00 course software.