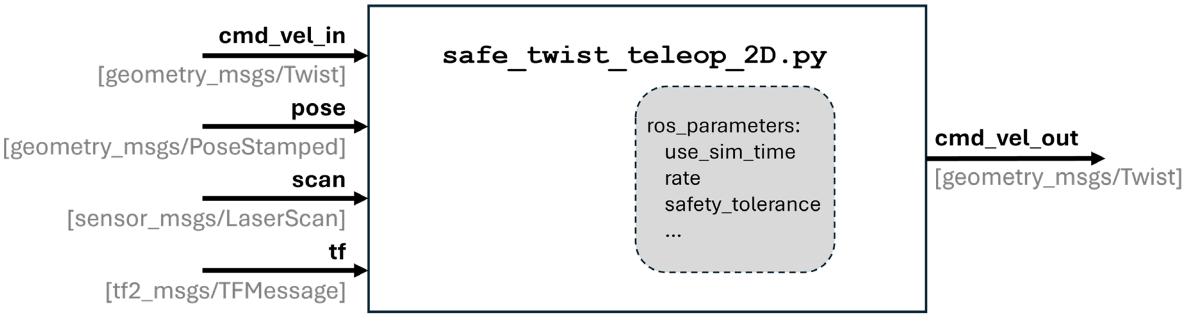
4TM00 Robot Motion Planning and Control
Assignment 1
Safe Robot Teleoperation & Reactive Robot Navigation

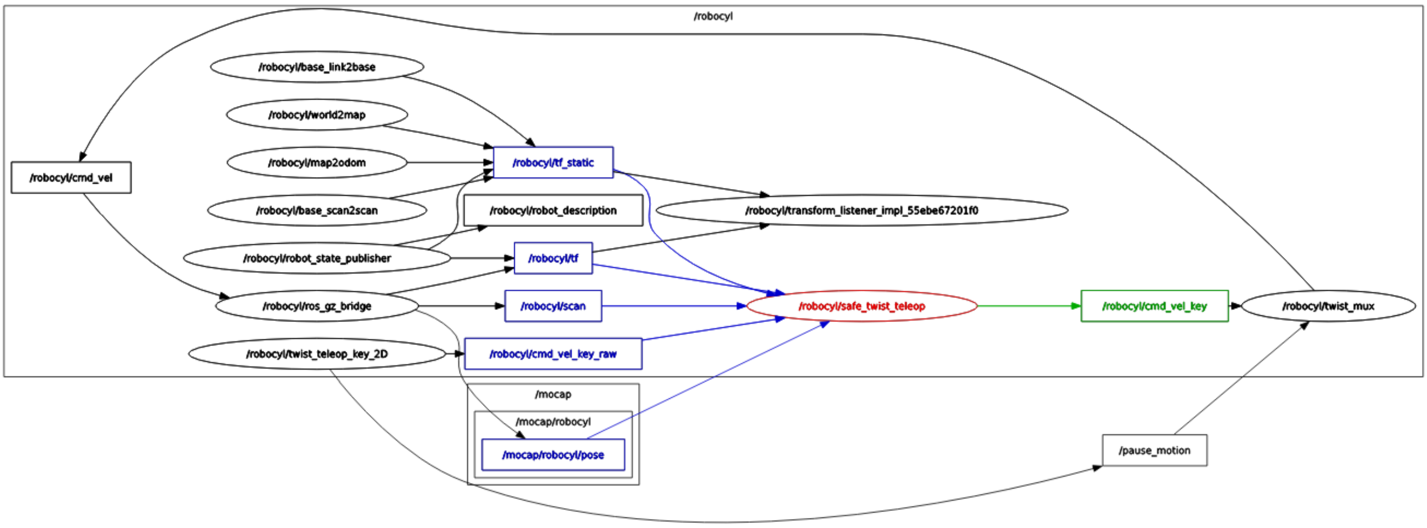
In this assignment, you will work with the RoboCyl mobile robot in Gazebo on two tasks: safe teleoperation and safe reactive navigation towards a goal position. You may use any relevant software available in the 4TM00 course's GitLab repository.
A specific ROS package, *core_tue4tm00_assignment1*, is 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_assignment1* into your group's GitLab repository and rename it with your group name, e.g., *group0_tue4tm00_assignment1*.

Part 1) *(Safe Robot Teleoperation)* In the first part of the assignment, you are ask to design a ROS node that receives original teleoperation control commands from a user as input and minimally modifies these commands using the robot's scan measurements, pose, and available coordinate transformations (i.e., tf) to ensure collision-free teleoperation around obstacles. In other words, if the requested control command is safe, the robot should execute it as is; otherwise, the command should be minimally modified or set to zero in the worst-case scenario to avoid collisions. A schematic drawing of the ROS node and an example rqt_graph is provided below for your reference.

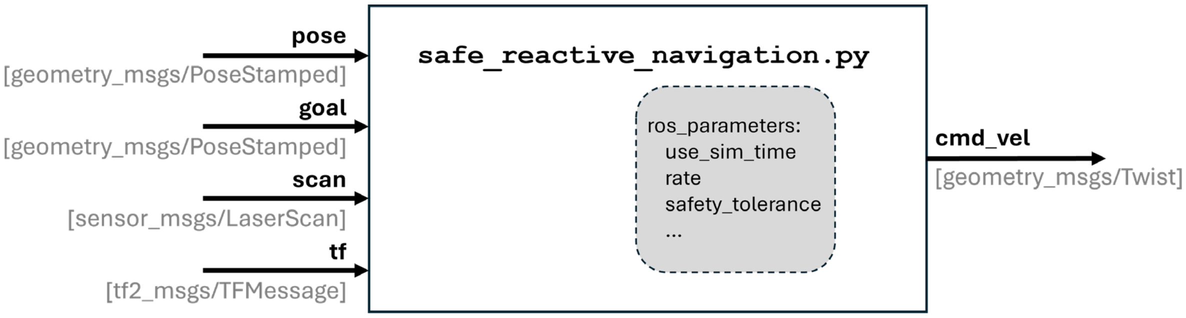


The ROS assignment package, *core_tue4tm00_assignment1*, includes:

- A ROS node script template: *scripts/safe_twist_teleop_2D.py*
- A parameter configuration file: *config/safe_twist_teleop_2D.yaml*
- Its interface launch file: *launch/safe_twist_teleop_2D.launch.py*
- A demo launch file: *launch/demo_safe_twist_teleop_2D.launch.py*

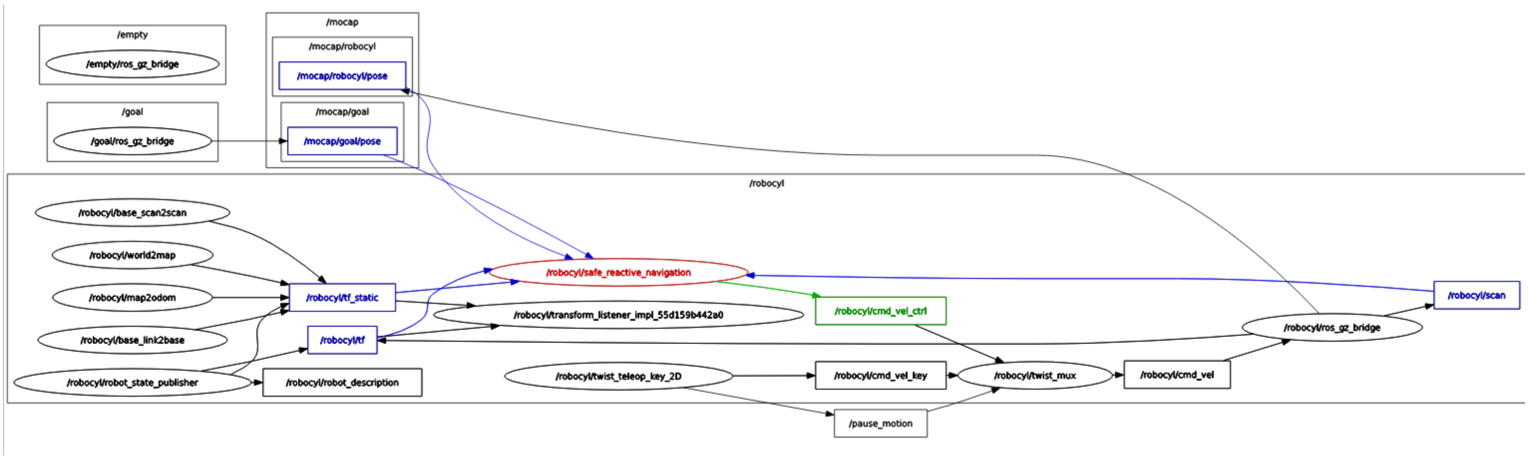


Part 2) (Safe Reactive Navigation) In the second part of the assignment, you are asked to design a ROS node that receives a goal position and navigates towards it while avoiding sensed obstacles. This could be achieved using the robot's scan measurements, pose, and available coordinate transformations (i.e., tf). As expected, without a map of the environment, it is not always possible to reach all given goal positions in a complex environment. Therefore, your objective is to use the instantaneous scan measurements as effectively as possible to move closer to the goal position without collisions. A schematic drawing and an example rqt_graph of the ROS node is provided below for your reference.



The ROS assignment package, *core_tue4tm00_assignment1*, includes:

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



For both 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.