We are the group made by Giulio Vaccari (10582927), Sofia Trombini (10617477) and Fabio Visentin (10601276) and we are students from automation and control engineering.

The files inside the archive are:

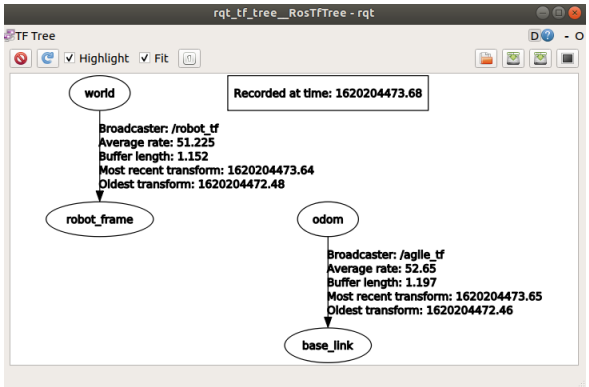
1. ***Bags*** folder →contains the bags provided.
2. ***Cfg*** folder → *parameter.cfg* that contains the configuration of the dynamic reconfigure server.
3. ***CMakeLists.txt*** → files that have to be compiled.
4. ***Launch*** folder → *robotics_first.launch* is the launch file that starts everything.
5. ***Msg*** folder → *CustomOdometry.msg* is the custom message & *MotorSpeed.msg* was provided to read the speeds from the bag.
6. ***Package.xml***
7. ***Src*** folder → *agile_tf.cpp* generates the tf of scout odom, *baseline_calculator.cpp* calculates the apparent baseline and the gear ratio, *tf_publisher.cpp* generates the tf of our odom, *odometry_functional.cpp* calculates the odom, does the dynamic reconfiguration and hosts the service to reset*, twist.cpp* synchronizes the messages of left and right front wheels.
8. ***Srv*** folder → *ResetToPose.srv* is the definition of the service to reset to a given pose.
9. ***Robotics_first.txt*** → it's this txt file where we describe our project.

ROS parameters are:

1. ***x0***, has initial value 0 and indicates the initial value of x.
2. ***y0****,* has initial value 0 and indicates the initial value of y.
3. ***theta0****,* has initial value 0 and indicates the initial value of the angle theta.

The structure of the TF tree:

1. world → robot_frame. It is referred to our odometry.
2. odom → base_link. It is referred to the scout odom.



The *CustomOdometry.msg* is the custom message. It contains *nav_msgs/Odometryodom* and *std_msgs/Stringmethod*.

We can start all the needed nodes with the command "*roslaunch robotics_first robotics_first.launch*". We can also start the single node with:

1. "*rosrun robotics_first twist",* that starts the synchronization between the left and right front wheels;
2. "*rosrun robotics_first agile_tf*", that starts the tf of the scout odom;

3. "*rosrun robotics_first baseline_calculator",* that starts the calculation of the apparent baseline and gear ratio;
4. *"rosrun robotics_first tf_publisher",* that starts the tf of our odom;
5. *"rosrun robotics_first odometry_functional",* that starts the calculation the odom, does the dynamic reconfiguration and hosts the service to reset.

The *ResetToPose.srv* allows us to reset position to a given one.

We did not provide the launch file of the baseline calculator, but we used it to calculate the baseline of the robot using the scout odom. To determine this value, we filtered the messages to have only the ones with the wheels moving in opposite directions and at almost the same speeds. If you want to see the baseline calculator launch file, please let us know.

In *twist.cpp* we decided to synchronise only the left and right front wheels, ignoring the messages of the rears because we noticed that they were very similar and to speed up the process.