Report – Homework 4

Students:

Matteo Langella

Raffaele Freschini

Nicola Caliendo

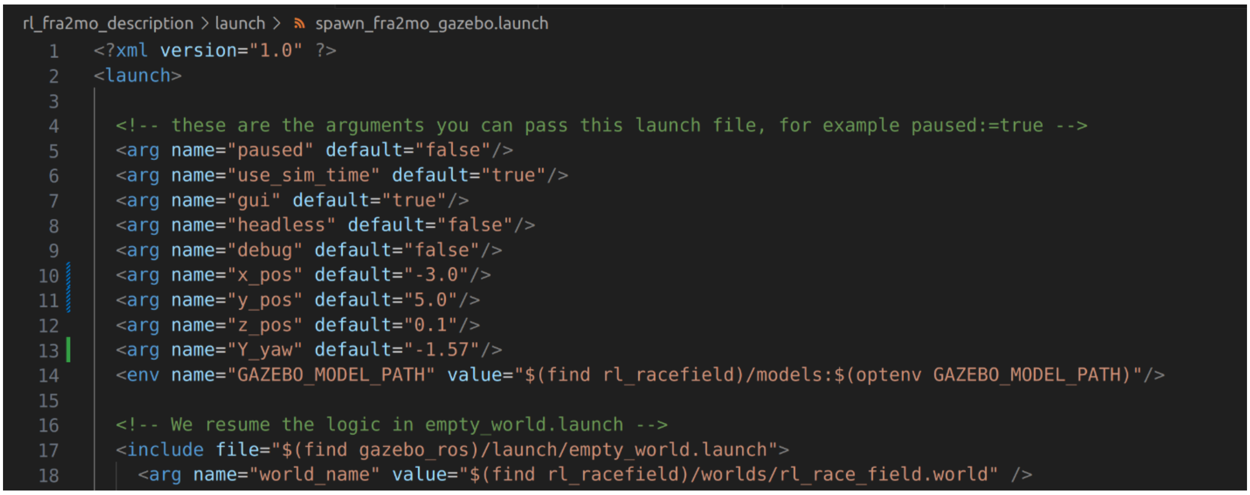
Control a mobile robot to follow a trajectory

1. Construct a gazebo world and spawn the mobile robot in a given pose

   a. Launch the Gazebo simulation and spawn the mobile robot in the world *“rl_racefield”* in the pose

$$x = -3, \qquad y = 5, \qquad yaw = -90\ deg$$

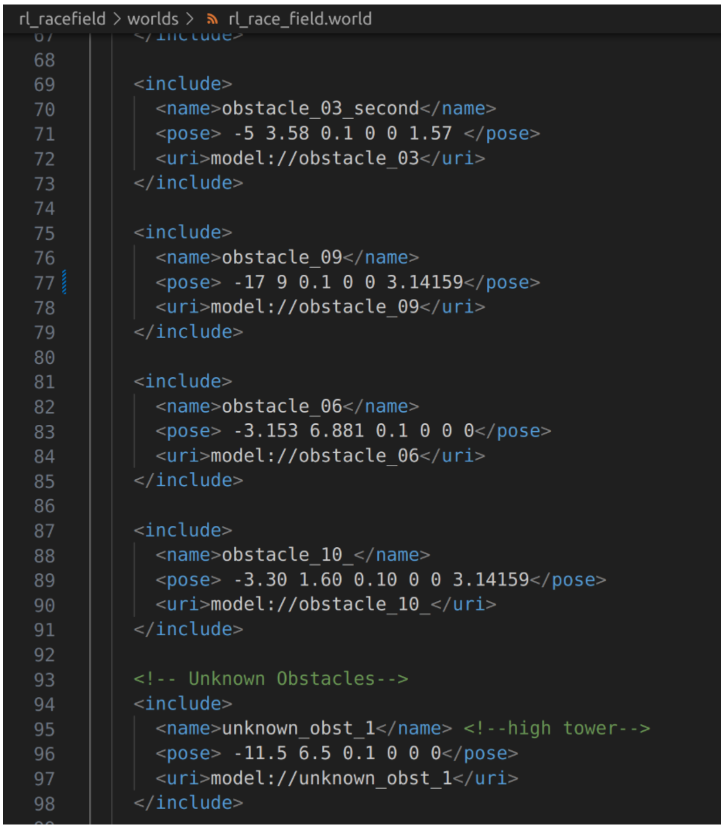
with respect to the map frame.



```xml
<?xml version="1.0" ?>
<launch>

  <!-- these are the arguments you can pass this launch file, for example paused:=true -->
  <arg name="paused" default="false"/>
  <arg name="use_sim_time" default="true"/>
  <arg name="gui" default="true"/>
  <arg name="headless" default="false"/>
  <arg name="debug" default="false"/>
  <arg name="x_pos" default="-3.0"/>
  <arg name="y_pos" default="5.0"/>
  <arg name="z_pos" default="0.1"/>
  <arg name="Y_yaw" default="-1.57"/>
  <env name="GAZEBO_MODEL_PATH" value="$(find rl_racefield)/models:$(optenv GAZEBO_MODEL_PATH)"/>

  <!-- We resume the logic in empty_world.launch -->
  <include file="$(find gazebo_ros)/launch/empty_world.launch">
    <arg name="world_name" value="$(find rl_racefield)/worlds/rl_race_field.world" />
```

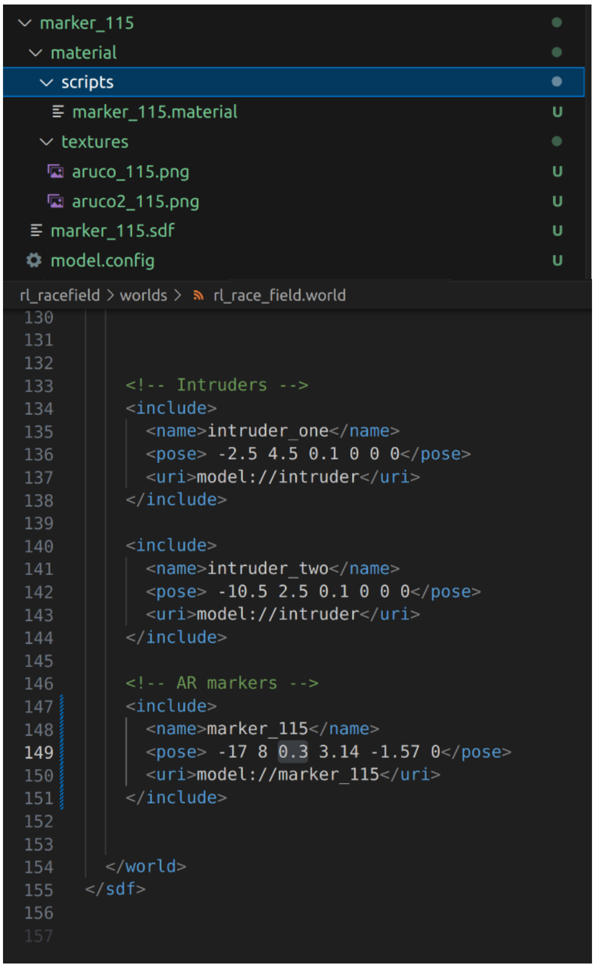
   b. Modify the world file of *“rl_racefield”* moving the obstacle 9 in position:

$$x = -17, \qquad y = 9, \qquad z = 0.1, \qquad yaw = 3.14$$



```xml
    </include>

    <include>
      <name>obstacle_03_second</name>
      <pose> -5 3.58 0.1 0 0 1.57 </pose>
      <uri>model://obstacle_03</uri>
    </include>

    <include>
      <name>obstacle_09</name>
      <pose> -17 9 0.1 0 0 3.14159</pose>
      <uri>model://obstacle_09</uri>
    </include>

    <include>
      <name>obstacle_06</name>
      <pose> -3.153 6.881 0.1 0 0 0</pose>
      <uri>model://obstacle_06</uri>
    </include>

    <include>
      <name>obstacle_10_</name>
      <pose> -3.30 1.60 0.10 0 0 3.14159</pose>
      <uri>model://obstacle_10_</uri>
    </include>

    <!-- Unknown Obstacles-->
    <include>
      <name>unknown_obst_1</name> <!--high tower-->
      <pose> -11.5 6.5 0.1 0 0 0</pose>
      <uri>model://unknown_obst_1</uri>
    </include>
```

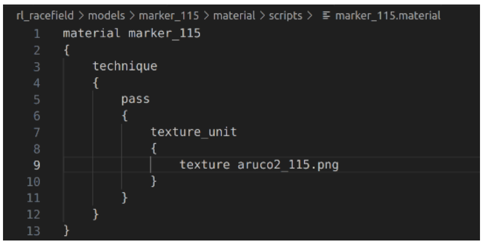
c. Place the *"ArUco"* marker number 115 on obstacle 9 in an appropriate position, such that it is visible by the mobile robot's camera when it comes in the proximity of the object.



```
marker_115
  material
    scripts
      marker_115.material          U
    textures
      aruco_115.png                U
      aruco2_115.png               U
  marker_115.sdf                   U
  model.config                     U
```

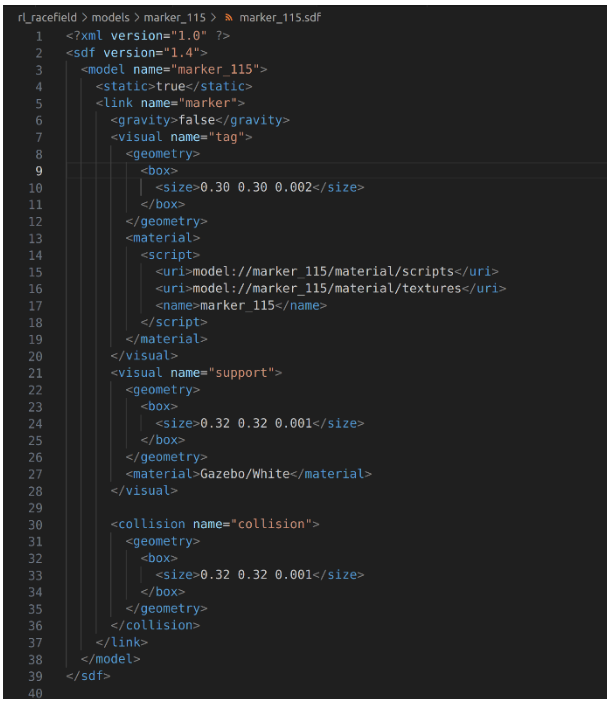
rl_racefield > worlds > rl_race_field.world

```



    <!-- Intruders -->
    <include>
      <name>intruder_one</name>
      <pose> -2.5 4.5 0.1 0 0 0</pose>
      <uri>model://intruder</uri>
    </include>

    <include>
      <name>intruder_two</name>
      <pose> -10.5 2.5 0.1 0 0 0</pose>
      <uri>model://intruder</uri>
    </include>

    <!-- AR markers -->
    <include>
      <name>marker_115</name>
      <pose> -17 8 0.3 3.14 -1.57 0</pose>
      <uri>model://marker_115</uri>
    </include>


  </world>
</sdf>
```



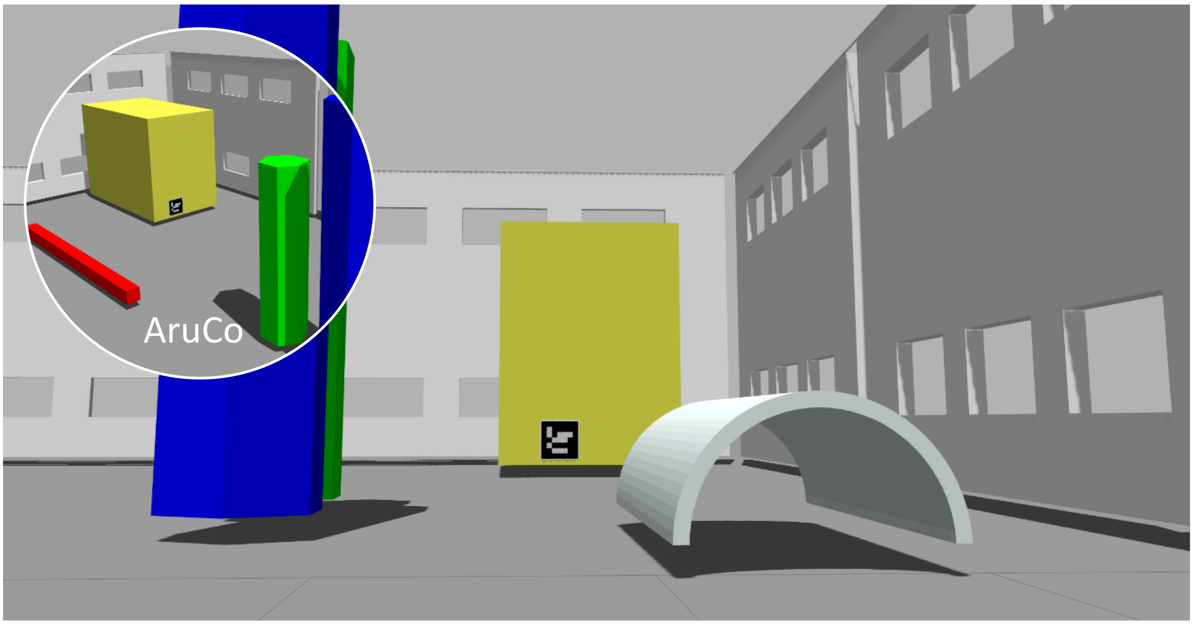
rl_racefield > models > marker_115 > material > scripts > marker_115.material

```
material marker_115
{
    technique
    {
        pass
        {
            texture_unit
            {
                texture aruco2_115.png
            }
        }
    }
}
```



rl_racefield > models > marker_115 > marker_115.sdf

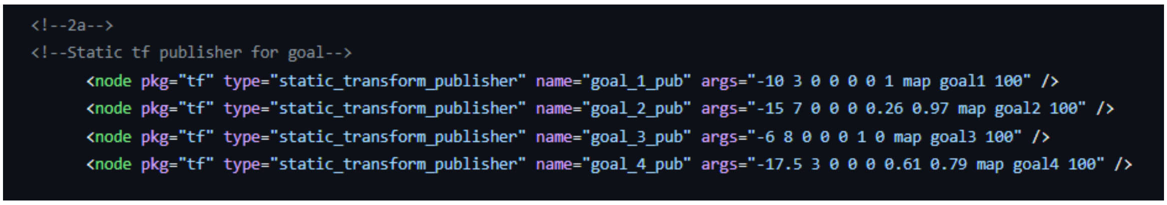
```
<?xml version="1.0" ?>
<sdf version="1.4">
  <model name="marker_115">
    <static>true</static>
    <link name="marker">
      <gravity>false</gravity>
      <visual name="tag">
        <geometry>
          <box>
            <size>0.30 0.30 0.002</size>
          </box>
        </geometry>
        <material>
          <script>
            <uri>model://marker_115/material/scripts</uri>
            <uri>model://marker_115/material/textures</uri>
            <name>marker_115</name>
          </script>
        </material>
      </visual>
      <visual name="support">
        <geometry>
          <box>
            <size>0.32 0.32 0.001</size>
          </box>
        </geometry>
        <material>Gazebo/White</material>
      </visual>

      <collision name="collision">
        <geometry>
          <box>
            <size>0.32 0.32 0.001</size>
          </box>
        </geometry>
      </collision>
    </link>
  </model>
</sdf>
```



2. Place static *"tf"* acting as goals and get their pose to enable an autonomous navigation task

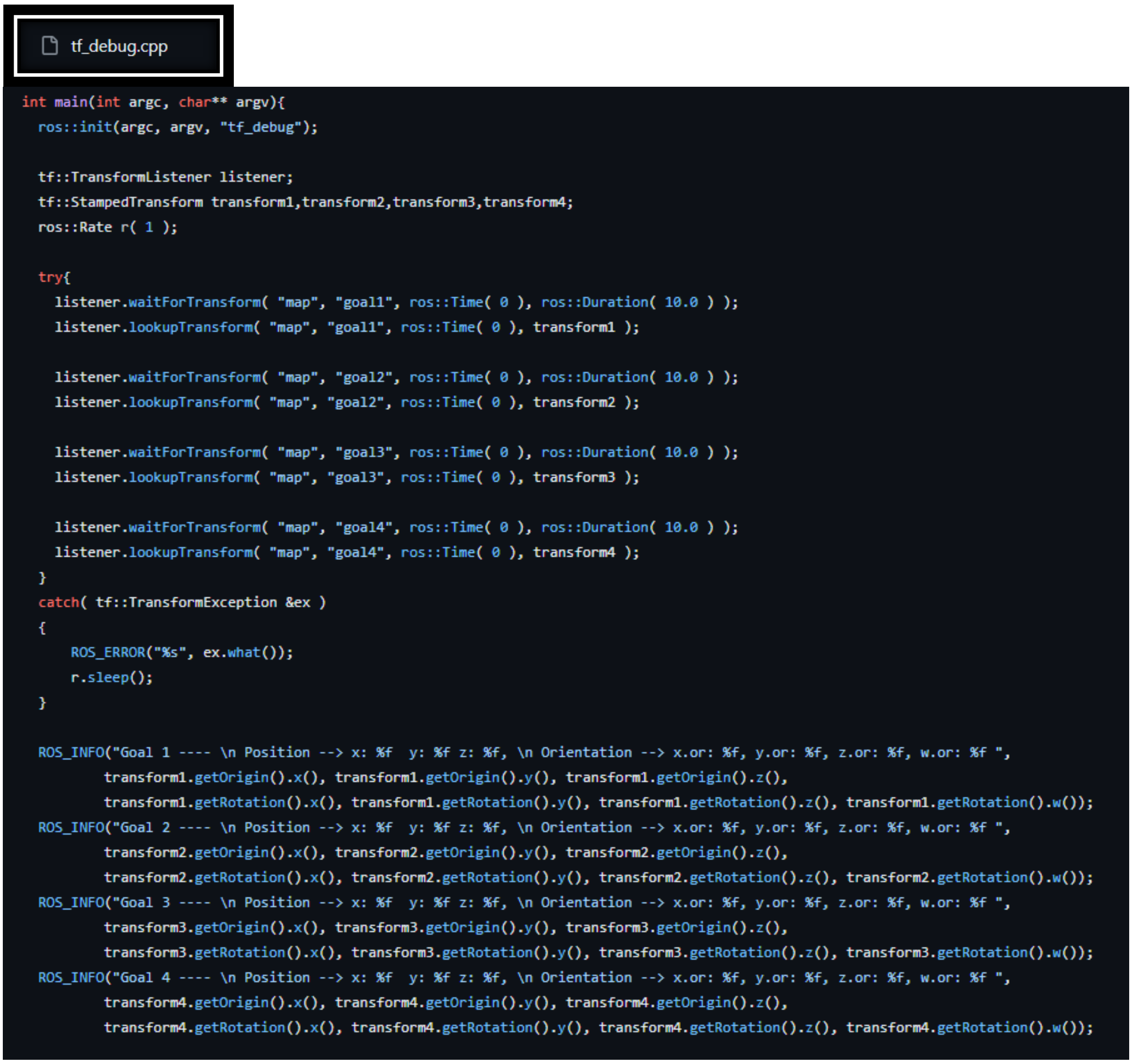
   a. Insert 4 static *"tf"* acting as goals in the following poses with respect to the map frame:

      - $Goal1: x = -10, \quad y = 3, \quad yaw = 0\ deg$
      - $Goal2: x = -15, \quad y = 7, \quad yaw = 30\ deg$
      - $Goal3: x = -6, \quad y = 8, \quad yaw = 180\ deg$
      - $Goal4: x = -17.5, \quad y = 3, \quad yaw = 75\ deg$

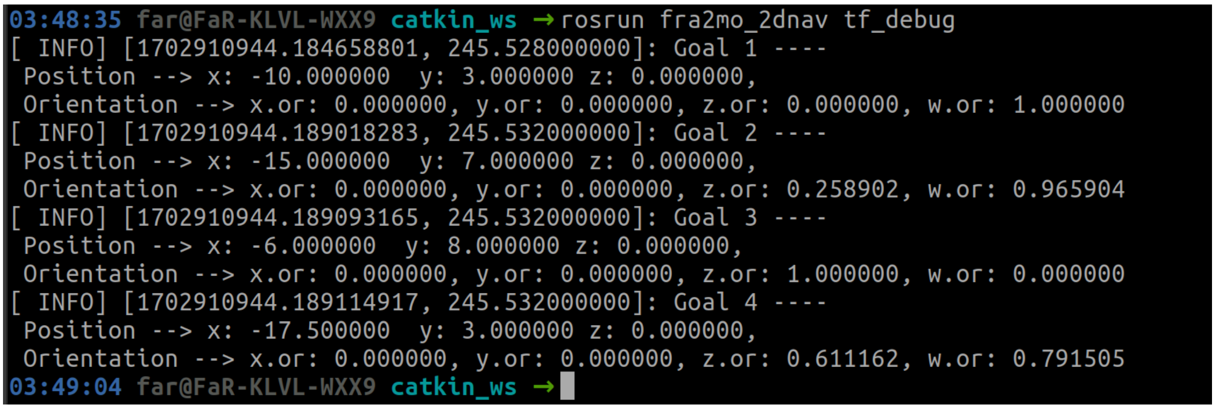


```
<!--2a-->
<!--Static tf publisher for goal-->
    <node pkg="tf" type="static_transform_publisher" name="goal_1_pub" args="-10 3 0 0 0 0 1 map goal1 100" />
    <node pkg="tf" type="static_transform_publisher" name="goal_2_pub" args="-15 7 0 0 0 0.26 0.97 map goal2 100" />
    <node pkg="tf" type="static_transform_publisher" name="goal_3_pub" args="-6 8 0 0 0 1 0 map goal3 100" />
    <node pkg="tf" type="static_transform_publisher" name="goal_4_pub" args="-17.5 3 0 0 0 0.61 0.79 map goal4 100" />
```

   b. Following the example code in *"fra2mo_2dnav/src/tf_nav.cpp"*, implement *"tf"* listeners to get target poses and print them to the terminal as debug.



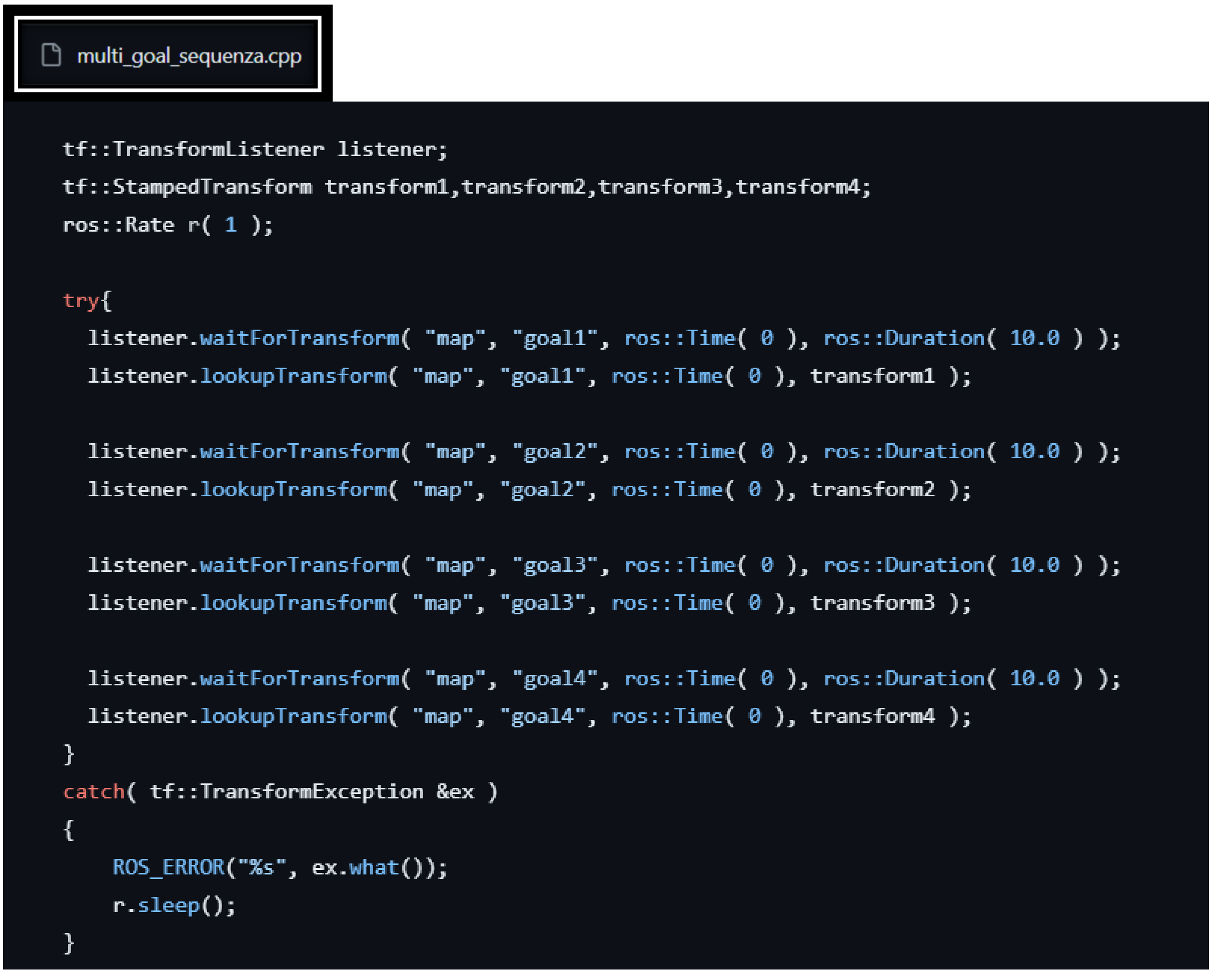
```
int main(int argc, char** argv){
  ros::init(argc, argv, "tf_debug");

  tf::TransformListener listener;
  tf::StampedTransform transform1,transform2,transform3,transform4;
  ros::Rate r( 1 );

  try{
    listener.waitForTransform( "map", "goal1", ros::Time( 0 ), ros::Duration( 10.0 ) );
    listener.lookupTransform( "map", "goal1", ros::Time( 0 ), transform1 );

    listener.waitForTransform( "map", "goal2", ros::Time( 0 ), ros::Duration( 10.0 ) );
    listener.lookupTransform( "map", "goal2", ros::Time( 0 ), transform2 );

    listener.waitForTransform( "map", "goal3", ros::Time( 0 ), ros::Duration( 10.0 ) );
    listener.lookupTransform( "map", "goal3", ros::Time( 0 ), transform3 );

    listener.waitForTransform( "map", "goal4", ros::Time( 0 ), ros::Duration( 10.0 ) );
    listener.lookupTransform( "map", "goal4", ros::Time( 0 ), transform4 );
  }
  catch( tf::TransformException &ex )
  {
      ROS_ERROR("%s", ex.what());
      r.sleep();
  }

  ROS_INFO("Goal 1 ---- \n Position --> x: %f  y: %f z: %f, \n Orientation --> x.or: %f, y.or: %f, z.or: %f, w.or: %f ",
          transform1.getOrigin().x(), transform1.getOrigin().y(), transform1.getOrigin().z(),
          transform1.getRotation().x(), transform1.getRotation().y(), transform1.getRotation().z(), transform1.getRotation().w());
  ROS_INFO("Goal 2 ---- \n Position --> x: %f  y: %f z: %f, \n Orientation --> x.or: %f, y.or: %f, z.or: %f, w.or: %f ",
          transform2.getOrigin().x(), transform2.getOrigin().y(), transform2.getOrigin().z(),
          transform2.getRotation().x(), transform2.getRotation().y(), transform2.getRotation().z(), transform2.getRotation().w());
  ROS_INFO("Goal 3 ---- \n Position --> x: %f  y: %f z: %f, \n Orientation --> x.or: %f, y.or: %f, z.or: %f, w.or: %f ",
          transform3.getOrigin().x(), transform3.getOrigin().y(), transform3.getOrigin().z(),
          transform3.getRotation().x(), transform3.getRotation().y(), transform3.getRotation().z(), transform3.getRotation().w());
  ROS_INFO("Goal 4 ---- \n Position --> x: %f  y: %f z: %f, \n Orientation --> x.or: %f, y.or: %f, z.or: %f, w.or: %f ",
          transform4.getOrigin().x(), transform4.getOrigin().y(), transform4.getOrigin().z(),
          transform4.getRotation().x(), transform4.getRotation().y(), transform4.getRotation().z(), transform4.getRotation().w());
```



```
03:48:35 far@FaR-KLVL-WXX9 catkin_ws →rosrun fra2mo_2dnav tf_debug
[ INFO] [1702910944.184658801, 245.528000000]: Goal 1 ----
 Position --> x: -10.000000  y: 3.000000 z: 0.000000,
 Orientation --> x.or: 0.000000, y.or: 0.000000, z.or: 0.000000, w.or: 1.000000
[ INFO] [1702910944.189018283, 245.532000000]: Goal 2 ----
 Position --> x: -15.000000  y: 7.000000 z: 0.000000,
 Orientation --> x.or: 0.000000, y.or: 0.000000, z.or: 0.258902, w.or: 0.965904
[ INFO] [1702910944.189093165, 245.532000000]: Goal 3 ----
 Position --> x: -6.000000  y: 8.000000 z: 0.000000,
 Orientation --> x.or: 0.000000, y.or: 0.000000, z.or: 1.000000, w.or: 0.000000
[ INFO] [1702910944.189114917, 245.532000000]: Goal 4 ----
 Position --> x: -17.500000  y: 3.000000 z: 0.000000,
 Orientation --> x.or: 0.000000, y.or: 0.000000, z.or: 0.611162, w.or: 0.791505
03:49:04 far@FaR-KLVL-WXX9 catkin_ws →
```

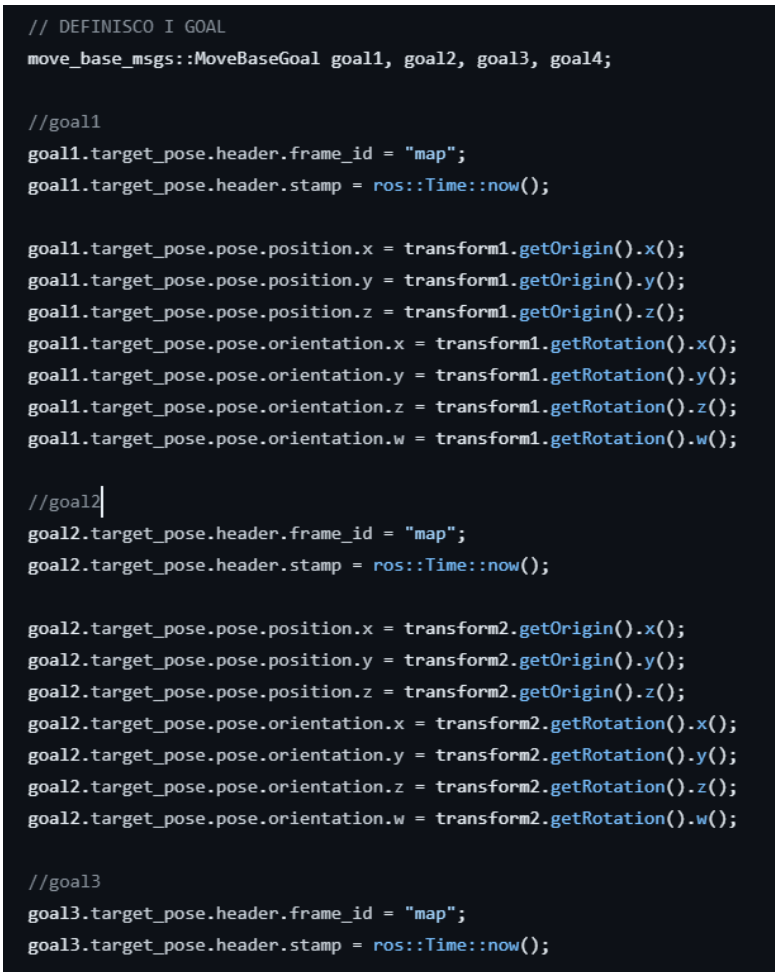
c. Using *"move_base"*, send goals to the mobile platform in a given order. Go to the next one once the robot has arrived at the current goal. The order of the explored goals must be Goal3→Goal4→Goal2→Goal1. Use the *"ActionClient"* communication protocol to get the feedback from *"move_base"*. Record a bag file of the executed robot trajectory and plot it as a result.

I created a *".cpp"* file that creates a listener for the static *"tf"*s that we published and uses them as goal for our mobile robot

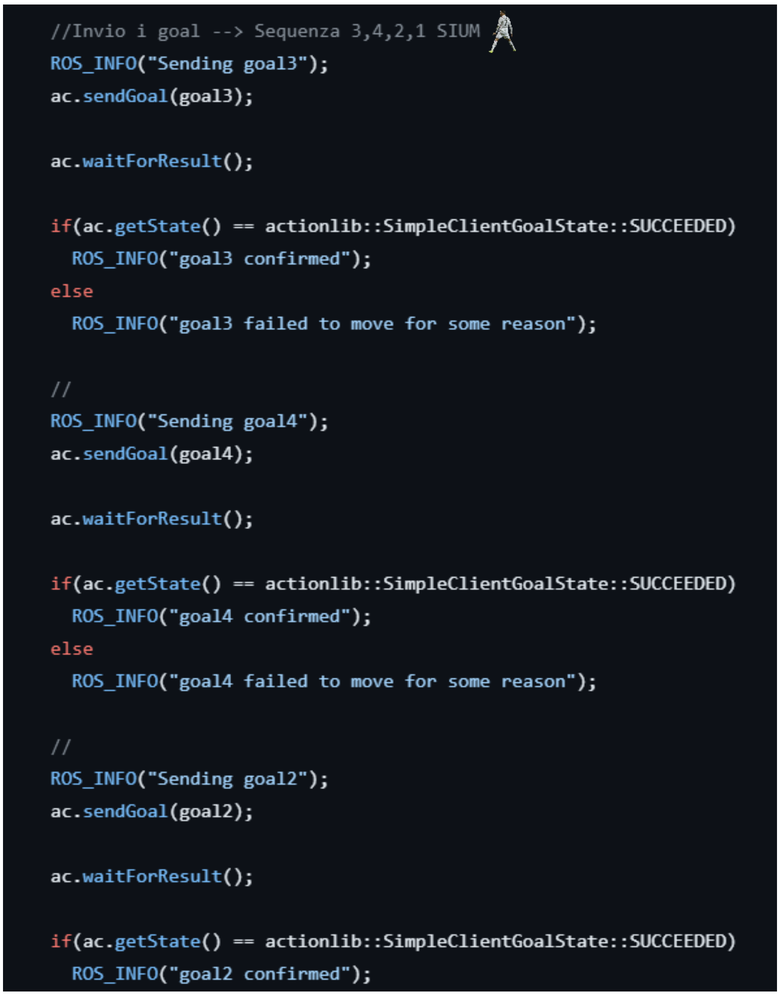


multi_goal_sequenza.cpp

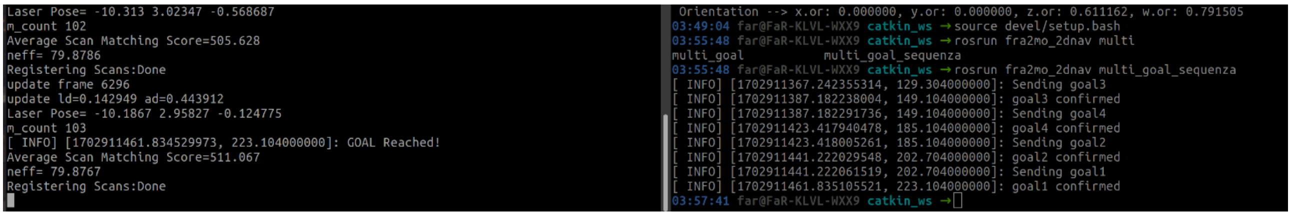
```
tf::TransformListener listener;
tf::StampedTransform transform1,transform2,transform3,transform4;
ros::Rate r( 1 );

try{
  listener.waitForTransform( "map", "goal1", ros::Time( 0 ), ros::Duration( 10.0 ) );
  listener.lookupTransform( "map", "goal1", ros::Time( 0 ), transform1 );

  listener.waitForTransform( "map", "goal2", ros::Time( 0 ), ros::Duration( 10.0 ) );
  listener.lookupTransform( "map", "goal2", ros::Time( 0 ), transform2 );

  listener.waitForTransform( "map", "goal3", ros::Time( 0 ), ros::Duration( 10.0 ) );
  listener.lookupTransform( "map", "goal3", ros::Time( 0 ), transform3 );

  listener.waitForTransform( "map", "goal4", ros::Time( 0 ), ros::Duration( 10.0 ) );
  listener.lookupTransform( "map", "goal4", ros::Time( 0 ), transform4 );
}
catch( tf::TransformException &ex )
{
    ROS_ERROR("%s", ex.what());
    r.sleep();
}
```



```
// DEFINISCO I GOAL
move_base_msgs::MoveBaseGoal goal1, goal2, goal3, goal4;

//goal1
goal1.target_pose.header.frame_id = "map";
goal1.target_pose.header.stamp = ros::Time::now();

goal1.target_pose.pose.position.x = transform1.getOrigin().x();
goal1.target_pose.pose.position.y = transform1.getOrigin().y();
goal1.target_pose.pose.position.z = transform1.getOrigin().z();
goal1.target_pose.pose.orientation.x = transform1.getRotation().x();
goal1.target_pose.pose.orientation.y = transform1.getRotation().y();
goal1.target_pose.pose.orientation.z = transform1.getRotation().z();
goal1.target_pose.pose.orientation.w = transform1.getRotation().w();

//goal2
goal2.target_pose.header.frame_id = "map";
goal2.target_pose.header.stamp = ros::Time::now();

goal2.target_pose.pose.position.x = transform2.getOrigin().x();
goal2.target_pose.pose.position.y = transform2.getOrigin().y();
goal2.target_pose.pose.position.z = transform2.getOrigin().z();
goal2.target_pose.pose.orientation.x = transform2.getRotation().x();
goal2.target_pose.pose.orientation.y = transform2.getRotation().y();
goal2.target_pose.pose.orientation.z = transform2.getRotation().z();
goal2.target_pose.pose.orientation.w = transform2.getRotation().w();

//goal3
goal3.target_pose.header.frame_id = "map";
goal3.target_pose.header.stamp = ros::Time::now();
```

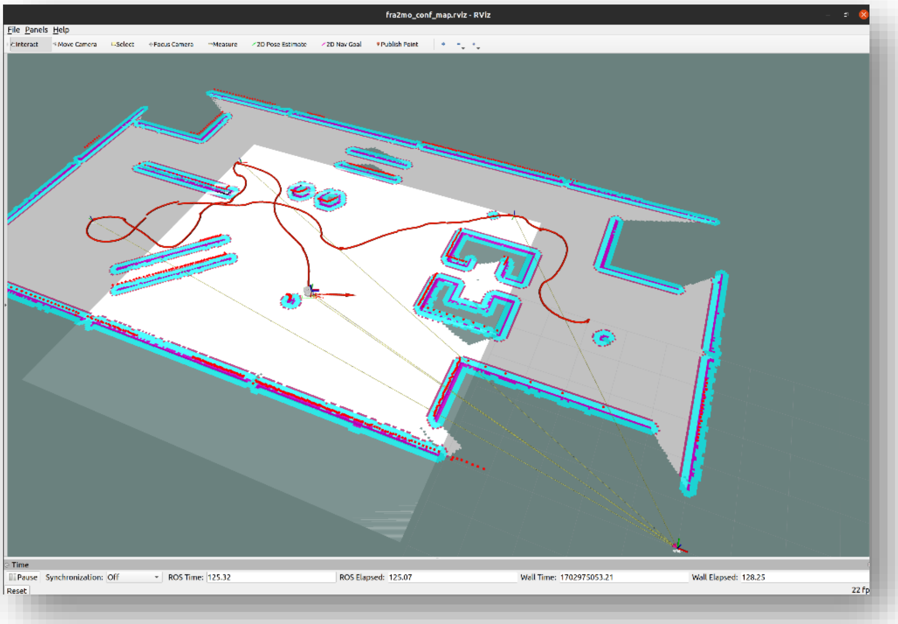


```
  //Invio i goal --> Sequenza 3,4,2,1 SIUM
  ROS_INFO("Sending goal3");
  ac.sendGoal(goal3);

  ac.waitForResult();

  if(ac.getState() == actionlib::SimpleClientGoalState::SUCCEEDED)
    ROS_INFO("goal3 confirmed");
  else
    ROS_INFO("goal3 failed to move for some reason");

  //
  ROS_INFO("Sending goal4");
  ac.sendGoal(goal4);

  ac.waitForResult();

  if(ac.getState() == actionlib::SimpleClientGoalState::SUCCEEDED)
    ROS_INFO("goal4 confirmed");
  else
    ROS_INFO("goal4 failed to move for some reason");

  //
  ROS_INFO("Sending goal2");
  ac.sendGoal(goal2);

  ac.waitForResult();

  if(ac.getState() == actionlib::SimpleClientGoalState::SUCCEEDED)
    ROS_INFO("goal2 confirmed");
```

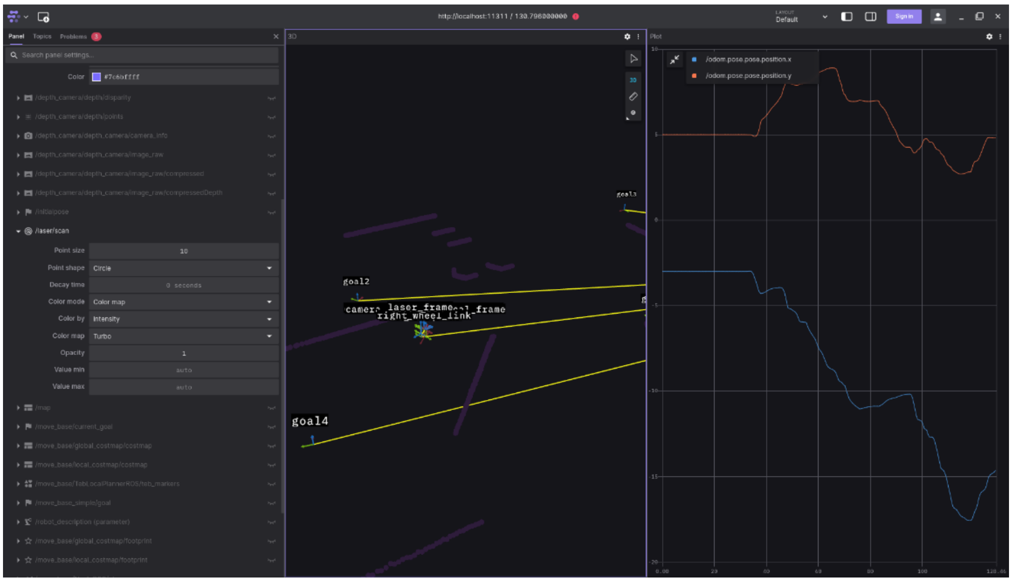


```
Laser Pose= -10.313 3.02347 -0.568687
m_count 102
Average Scan Matching Score=505.628
neff= 79.8786
Registering Scans:Done
update frame 6296
update ld=0.142949 ad=0.443912
Laser Pose= -10.1867 2.95827 -0.124775
m_count 103
[ INFO] [1702911461.834529973, 223.104000000]: GOAL Reached!
Average Scan Matching Score=511.067
neff= 79.8767
Registering Scans:Done
```

```
 Orientation --> x.or: 0.000000, y.or: 0.000000, z.or: 0.611162, w.or: 0.791505
03:49:04 far@FaR-KLVL-WXX9 catkin_ws → source devel/setup.bash
03:55:48 far@FaR-KLVL-WXX9 catkin_ws → rosrun fra2mo_2dnav multi
multi_goal           multi_goal_sequenza
03:55:48 far@FaR-KLVL-WXX9 catkin_ws → rosrun fra2mo_2dnav multi_goal_sequenza
[ INFO] [1702911367.242355314, 129.304000000]: Sending goal3
[ INFO] [1702911387.182238004, 149.104000000]: goal3 confirmed
[ INFO] [1702911387.182291736, 149.104000000]: Sending goal4
[ INFO] [1702911423.417940478, 185.104000000]: goal4 confirmed
[ INFO] [1702911423.418005261, 185.104000000]: Sending goal2
[ INFO] [1702911441.222029548, 202.704000000]: goal2 confirmed
[ INFO] [1702911441.222061519, 202.704000000]: Sending goal1
[ INFO] [1702911461.835105521, 223.104000000]: goal1 confirmed
03:57:41 far@FaR-KLVL-WXX9 catkin_ws →
```



Synchronization: Off ROS Time: 125.32 ROS Elapsed: 125.07 Wall Time: 1702975053.21 Wall Elapsed: 128.25


Panel
Topics
Problems 3
Search panel settings...
/depth_camera/depth/disparity
/depth_camera/depth/points
/depth_camera/depth_camera/camera_info
/depth_camera/depth_camera/image_raw
/depth_camera/depth_camera/image_raw/compressed
/depth_camera/depth_camera/image_raw/compressedDepth
/initialpose
/laser/scan
/map
/move_base/current_goal
/move_base/global_costmap/costmap
/move_base/local_costmap/costmap
/move_base/TebLocalPlannerROS/teb_markers
/move_base_simple/goal
/robot_description (parameter)
/move_base/global_costmap/footprint
/move_base/local_costmap/footprint
3D
goal3
goal2
camera
laser_frame
right_wheel_link
goal4
http://localhost:11311 / 130.796000000
LAYOUT
Default
Sign in
Plot
/odom.pose.pose.position.x
/odom.pose.pose.position.y
10
5
0
-5
-10
-15
-20
0.00
20
40
60
80
100
128.46

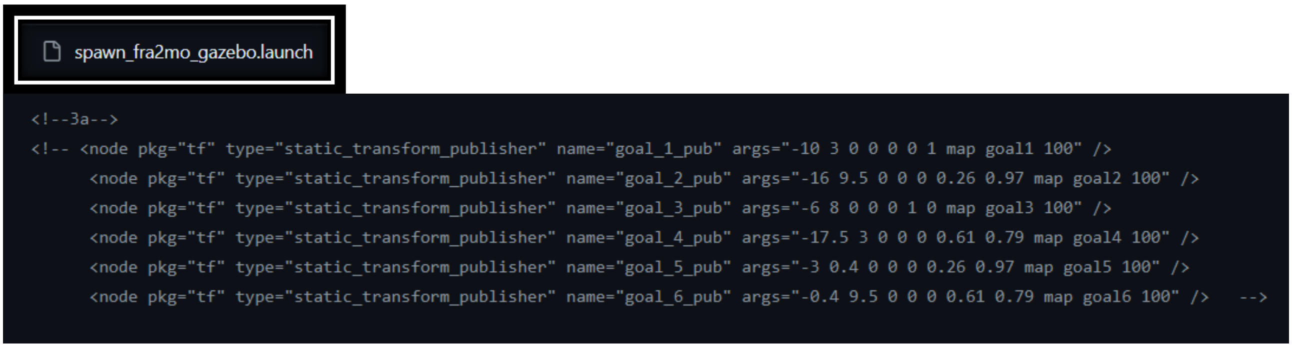
Color #7c6bffff

Point size 10
Point shape Circle
Decay time 0 seconds
Color mode Color map
Color by Intensity
Color map Turbo
Opacity 1
Value min auto
Value max auto

Instead of using a dedicated node to accomplish this goal, another possible solution would be to simply launch an instance of *“move_base”* and *“actionlib”* clients and use the *“Nav 2D Goal”* command inside of *“Rviz”* to manually plan the trajectory.

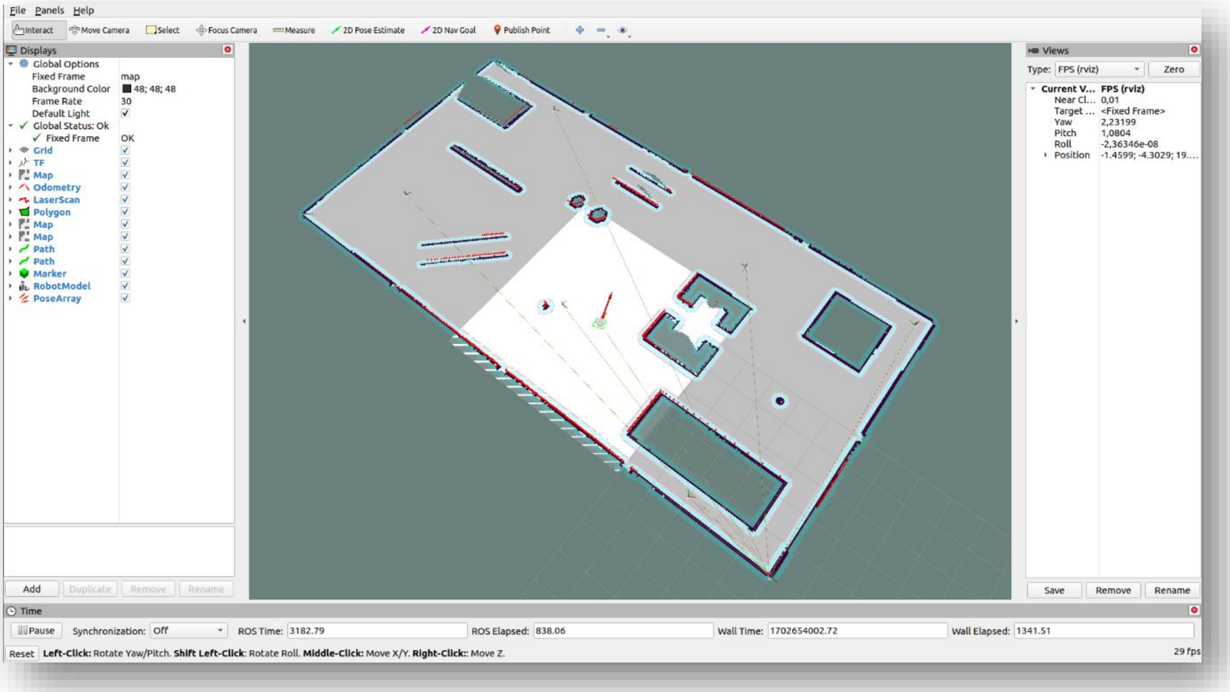
3. Map the environment tuning the navigation stack’s parameters

   a. Modify, add, remove, or change pose, the previous goals to get a complete map of the environment.



spawn_fra2mo_gazebo.launch

```
<!--3a-->
<!-- <node pkg="tf" type="static_transform_publisher" name="goal_1_pub" args="-10 3 0 0 0 0 1 map goal1 100" />
      <node pkg="tf" type="static_transform_publisher" name="goal_2_pub" args="-16 9.5 0 0 0 0.26 0.97 map goal2 100" />
      <node pkg="tf" type="static_transform_publisher" name="goal_3_pub" args="-6 8 0 0 0 1 0 map goal3 100" />
      <node pkg="tf" type="static_transform_publisher" name="goal_4_pub" args="-17.5 3 0 0 0 0.61 0.79 map goal4 100" />
      <node pkg="tf" type="static_transform_publisher" name="goal_5_pub" args="-3 0.4 0 0 0 0.26 0.97 map goal5 100" />
      <node pkg="tf" type="static_transform_publisher" name="goal_6_pub" args="-0.4 9.5 0 0 0 0.61 0.79 map goal6 100" />   -->
```



Global Options
Fixed Frame: map
Background Color: 48; 48; 48
Frame Rate: 30
Default Light: ✓
Global Status: Ok
Fixed Frame: OK

Type: FPS (rviz)
Current V... FPS (rviz)
Near Cl... 0,01
Target ... <Fixed Frame>
Yaw 2,23199
Pitch 1,0804
Roll -2,36346e-08
Position -1.4599; -4.3029; 19....

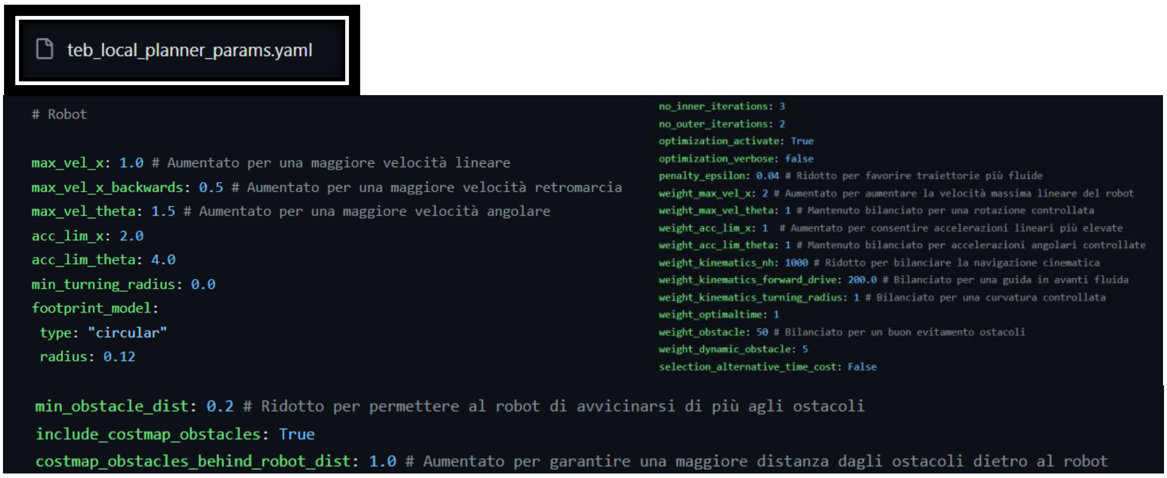
Synchronization: Off
ROS Time: 3182.79
ROS Elapsed: 838.06
Wall Time: 1702654002.72
Wall Elapsed: 1341.51

Left-Click: Rotate Yaw/Pitch. **Shift Left-Click**: Rotate Roll. **Middle-Click:** Move X/Y. **Right-Click:**: Move Z.

29 fps

b. Change the parameters of the planner and *“move_base”* and comment on the results you get in terms of robot trajectories.

This configuration gives better velocity performance



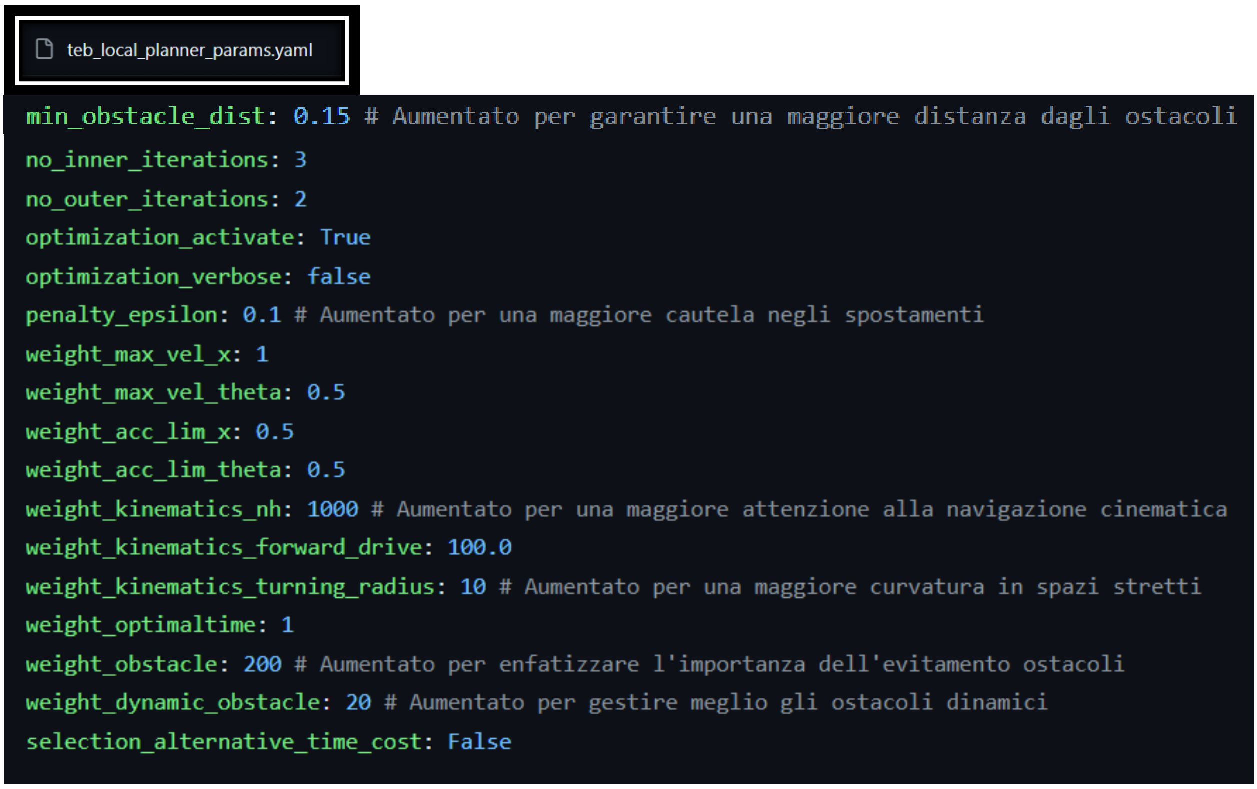
teb_local_planner_params.yaml

```yaml
# Robot

max_vel_x: 1.0 # Aumentato per una maggiore velocità lineare
max_vel_x_backwards: 0.5 # Aumentato per una maggiore velocità retromarcia
max_vel_theta: 1.5 # Aumentato per una maggiore velocità angolare
acc_lim_x: 2.0
acc_lim_theta: 4.0
min_turning_radius: 0.0
footprint_model:
 type: "circular"
 radius: 0.12
```

```yaml
no_inner_iterations: 3
no_outer_iterations: 2
optimization_activate: True
optimization_verbose: false
penalty_epsilon: 0.04 # Ridotto per favorire traiettorie più fluide
weight_max_vel_x: 2 # Aumentato per aumentare la velocità massima lineare del robot
weight_max_vel_theta: 1 # Mantenuto bilanciato per una rotazione controllata
weight_acc_lim_x: 1  # Aumentato per consentire accelerazioni lineari più elevate
weight_acc_lim_theta: 1 # Mantenuto bilanciato per accelerazioni angolari controllate
weight_kinematics_nh: 1000 # Ridotto per bilanciare la navigazione cinematica
weight_kinematics_forward_drive: 200.0 # Bilanciato per una guida in avanti fluida
weight_kinematics_turning_radius: 1 # Bilanciato per una curvatura controllata
weight_optimaltime: 1
weight_obstacle: 50 # Bilanciato per un buon evitamento ostacoli
weight_dynamic_obstacle: 5
selection_alternative_time_cost: False
```

```yaml
min_obstacle_dist: 0.2 # Ridotto per permettere al robot di avvicinarsi di più agli ostacoli
include_costmap_obstacles: True
costmap_obstacles_behind_robot_dist: 1.0 # Aumentato per garantire una maggiore distanza dagli ostacoli dietro al robot
```

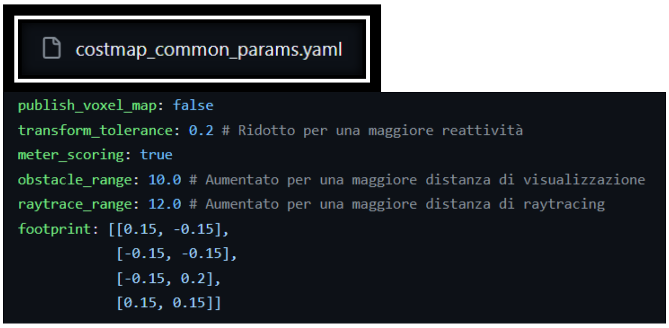
This configuration betters the obstacle avoidance capabilities and the ability of the robot to navigate narrow spaces



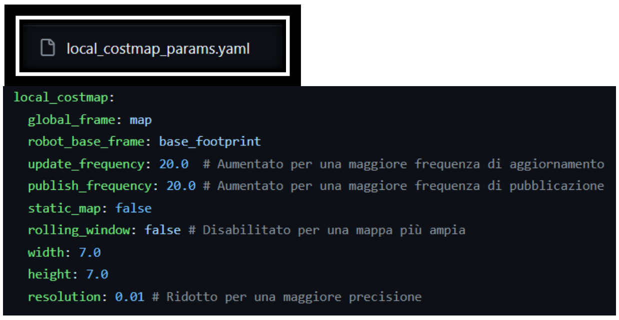
teb_local_planner_params.yaml

```yaml
min_obstacle_dist: 0.15 # Aumentato per garantire una maggiore distanza dagli ostacoli
no_inner_iterations: 3
no_outer_iterations: 2
optimization_activate: True
optimization_verbose: false
penalty_epsilon: 0.1 # Aumentato per una maggiore cautela negli spostamenti
weight_max_vel_x: 1
weight_max_vel_theta: 0.5
weight_acc_lim_x: 0.5
weight_acc_lim_theta: 0.5
weight_kinematics_nh: 1000 # Aumentato per una maggiore attenzione alla navigazione cinematica
weight_kinematics_forward_drive: 100.0
weight_kinematics_turning_radius: 10 # Aumentato per una maggiore curvatura in spazi stretti
weight_optimaltime: 1
weight_obstacle: 200 # Aumentato per enfatizzare l'importanza dell'evitamento ostacoli
weight_dynamic_obstacle: 20 # Aumentato per gestire meglio gli ostacoli dinamici
selection_alternative_time_cost: False
```

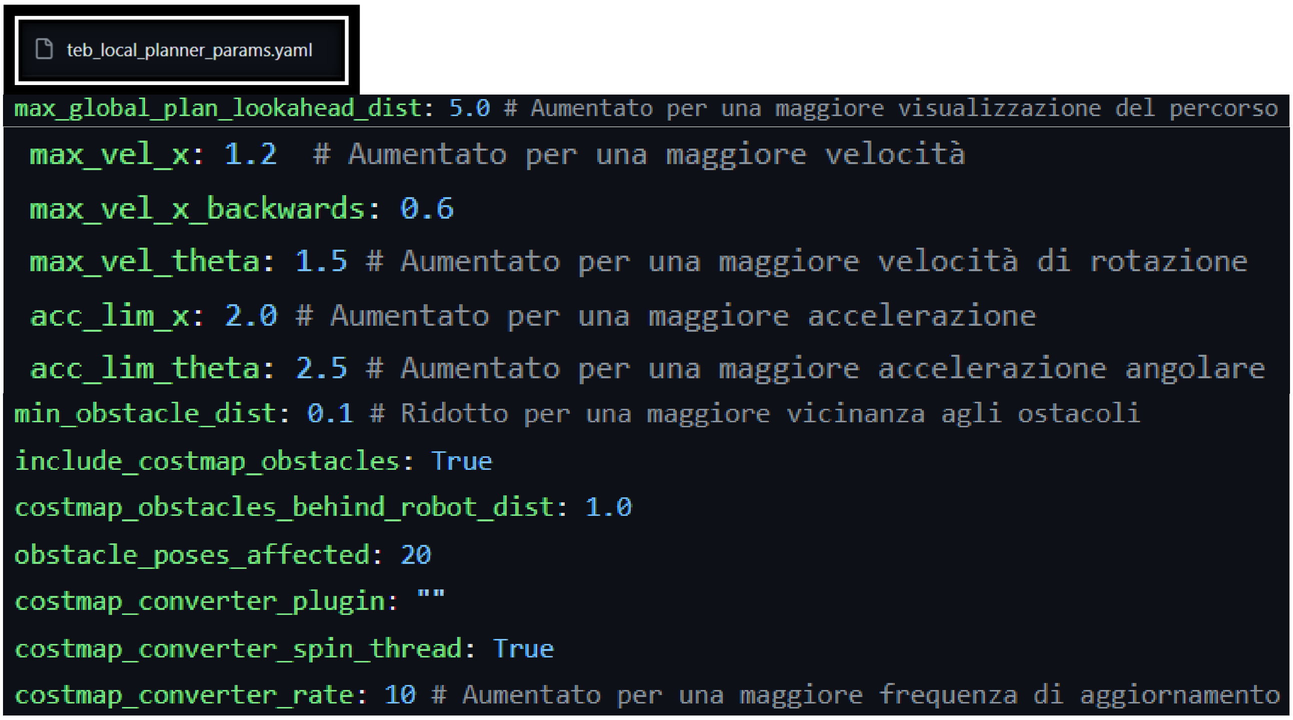
This configuration ensures a complete exploration of the environment.



```
publish_voxel_map: false
transform_tolerance: 0.2 # Ridotto per una maggiore reattività
meter_scoring: true
obstacle_range: 10.0 # Aumentato per una maggiore distanza di visualizzazione
raytrace_range: 12.0 # Aumentato per una maggiore distanza di raytracing
footprint: [[0.15, -0.15],
            [-0.15, -0.15],
            [-0.15, 0.2],
            [0.15, 0.15]]
```

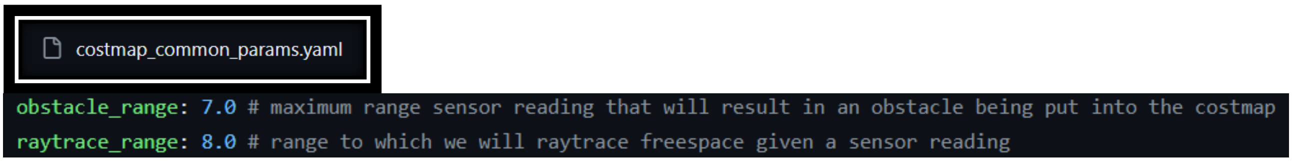


```
local_costmap:
  global_frame: map
  robot_base_frame: base_footprint
  update_frequency: 20.0  # Aumentato per una maggiore frequenza di aggiornamento
  publish_frequency: 20.0 # Aumentato per una maggiore frequenza di pubblicazione
  static_map: false
  rolling_window: false # Disabilitato per una mappa più ampia
  width: 7.0
  height: 7.0
  resolution: 0.01 # Ridotto per una maggiore precisione
```

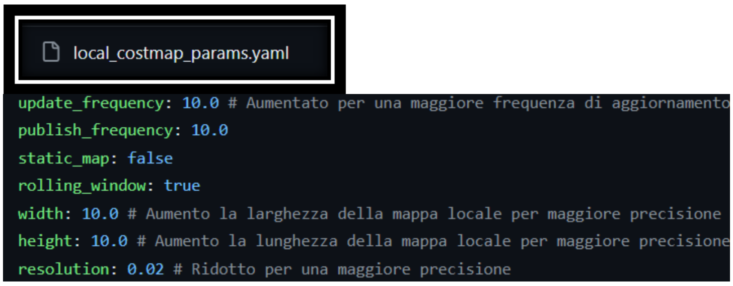


```
max_global_plan_lookahead_dist: 5.0 # Aumentato per una maggiore visualizzazione del percorso
 max_vel_x: 1.2  # Aumentato per una maggiore velocità
 max_vel_x_backwards: 0.6
 max_vel_theta: 1.5 # Aumentato per una maggiore velocità di rotazione
 acc_lim_x: 2.0 # Aumentato per una maggiore accelerazione
 acc_lim_theta: 2.5 # Aumentato per una maggiore accelerazione angolare
min_obstacle_dist: 0.1 # Ridotto per una maggiore vicinanza agli ostacoli
include_costmap_obstacles: True
costmap_obstacles_behind_robot_dist: 1.0
obstacle_poses_affected: 20
costmap_converter_plugin: ""
costmap_converter_spin_thread: True
costmap_converter_rate: 10 # Aumentato per una maggiore frequenza di aggiornamento
```

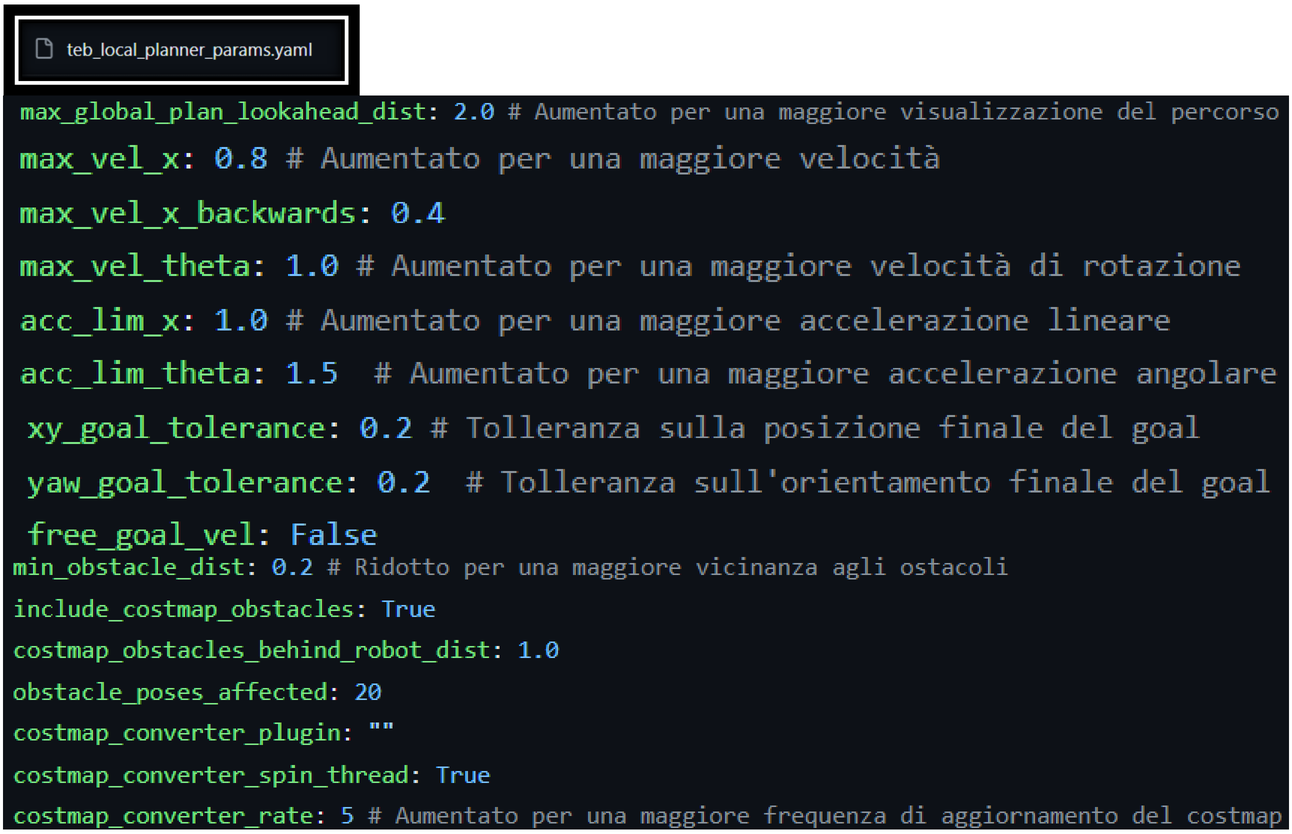
This configuration guarantees motion smoothness, balancing the obstacle avoidance capabilities and velocity performances



```
obstacle_range: 7.0 # maximum range sensor reading that will result in an obstacle being put into the costmap
raytrace_range: 8.0 # range to which we will raytrace freespace given a sensor reading
```



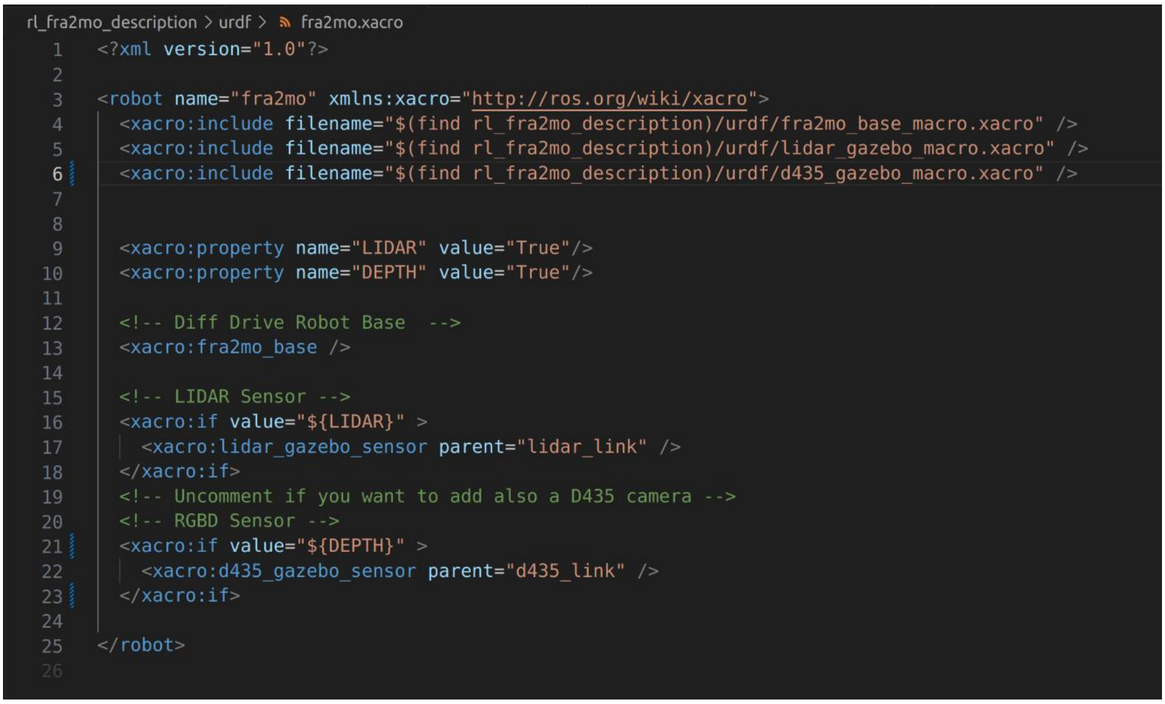
```
update_frequency: 10.0 # Aumentato per una maggiore frequenza di aggiornamento
publish_frequency: 10.0
static_map: false
rolling_window: true
width: 10.0 # Aumento la larghezza della mappa locale per maggiore precisione
height: 10.0 # Aumento la lunghezza della mappa locale per maggiore precisione
resolution: 0.02 # Ridotto per una maggiore precisione
```



```yaml
max_global_plan_lookahead_dist: 2.0 # Aumentato per una maggiore visualizzazione del percorso
max_vel_x: 0.8 # Aumentato per una maggiore velocità
max_vel_x_backwards: 0.4
max_vel_theta: 1.0 # Aumentato per una maggiore velocità di rotazione
acc_lim_x: 1.0 # Aumentato per una maggiore accelerazione lineare
acc_lim_theta: 1.5  # Aumentato per una maggiore accelerazione angolare
xy_goal_tolerance: 0.2 # Tolleranza sulla posizione finale del goal
yaw_goal_tolerance: 0.2  # Tolleranza sull'orientamento finale del goal
free_goal_vel: False
min_obstacle_dist: 0.2 # Ridotto per una maggiore vicinanza agli ostacoli
include_costmap_obstacles: True
costmap_obstacles_behind_robot_dist: 1.0
obstacle_poses_affected: 20
costmap_converter_plugin: ""
costmap_converter_spin_thread: True
costmap_converter_rate: 5 # Aumentato per una maggiore frequenza di aggiornamento del costmap
```

4. Vision-based navigation of the mobile platform

a. Run *"ArUcoROS"* node using the robot camera: bring up the camera model and uncomment it in that *"fra2mo.xacro"* file of the mobile robot description *"rl_fra2mo_description"*.



rl_fra2mo_description > urdf > fra2mo.xacro

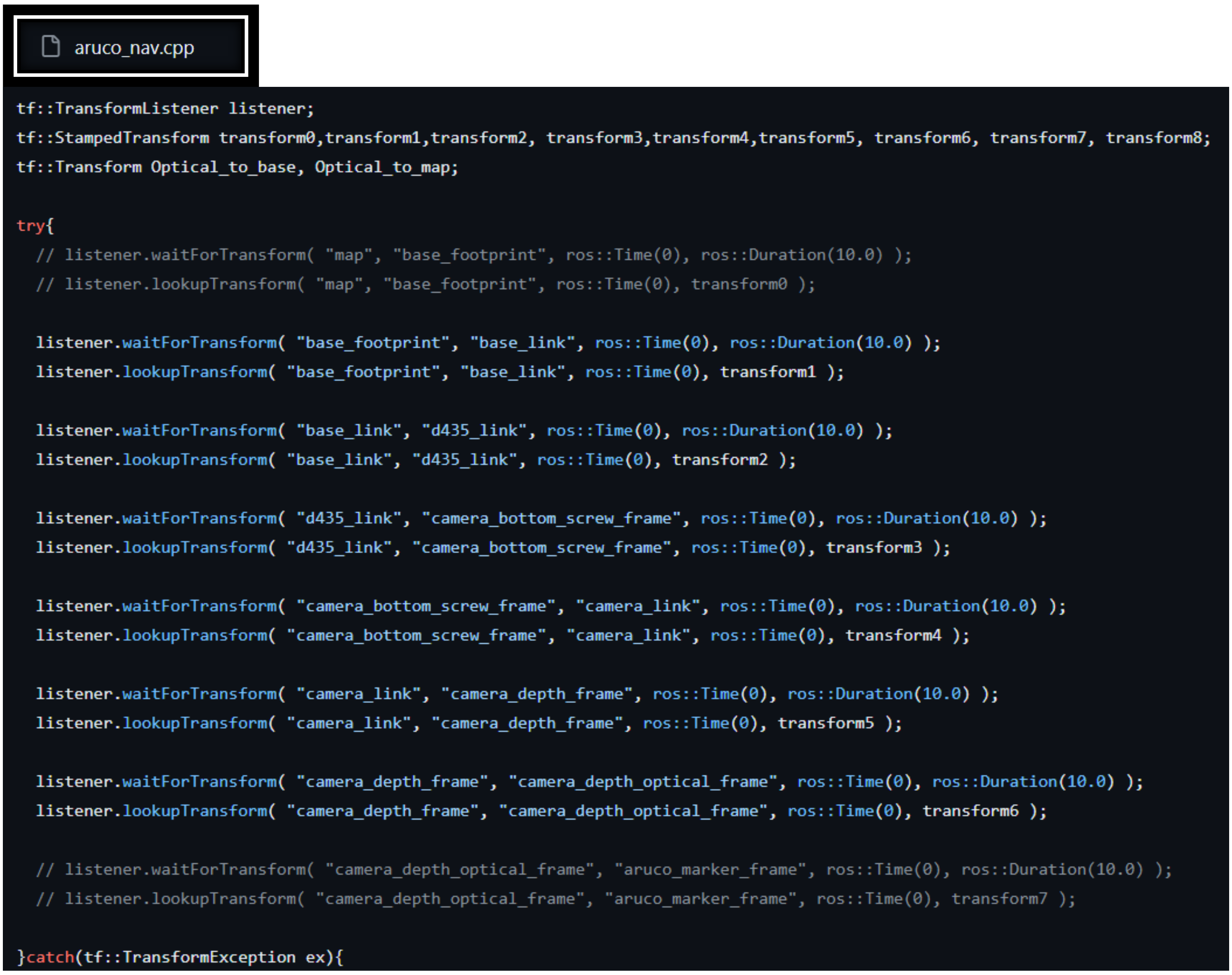
```xml
<?xml version="1.0"?>

<robot name="fra2mo" xmlns:xacro="http://ros.org/wiki/xacro">
  <xacro:include filename="$(find rl_fra2mo_description)/urdf/fra2mo_base_macro.xacro" />
  <xacro:include filename="$(find rl_fra2mo_description)/urdf/lidar_gazebo_macro.xacro" />
  <xacro:include filename="$(find rl_fra2mo_description)/urdf/d435_gazebo_macro.xacro" />


  <xacro:property name="LIDAR" value="True"/>
  <xacro:property name="DEPTH" value="True"/>

  <!-- Diff Drive Robot Base  -->
  <xacro:fra2mo_base />

  <!-- LIDAR Sensor -->
  <xacro:if value="${LIDAR}" >
    <xacro:lidar_gazebo_sensor parent="lidar_link" />
  </xacro:if>
  <!-- Uncomment if you want to add also a D435 camera -->
  <!-- RGBD Sensor -->
  <xacro:if value="${DEPTH}" >
    <xacro:d435_gazebo_sensor parent="d435_link" />
  </xacro:if>

</robot>

```

b. Implement a 2D navigation task following this logic

- Send the robot in the proximity of obstacle 9.
- Make the robot look for the *"ArUco"* marker. Once detected, retrieve its pose with respect to the map frame.
- Set the following pose as next goal for the robot

$$x = x_m + 1, \qquad y = y_m$$

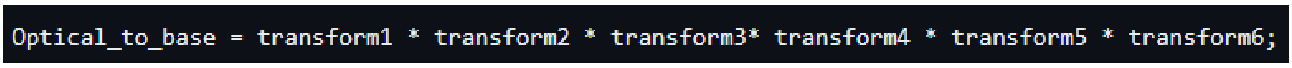
Where x_m, y_m are the marker coordinates.



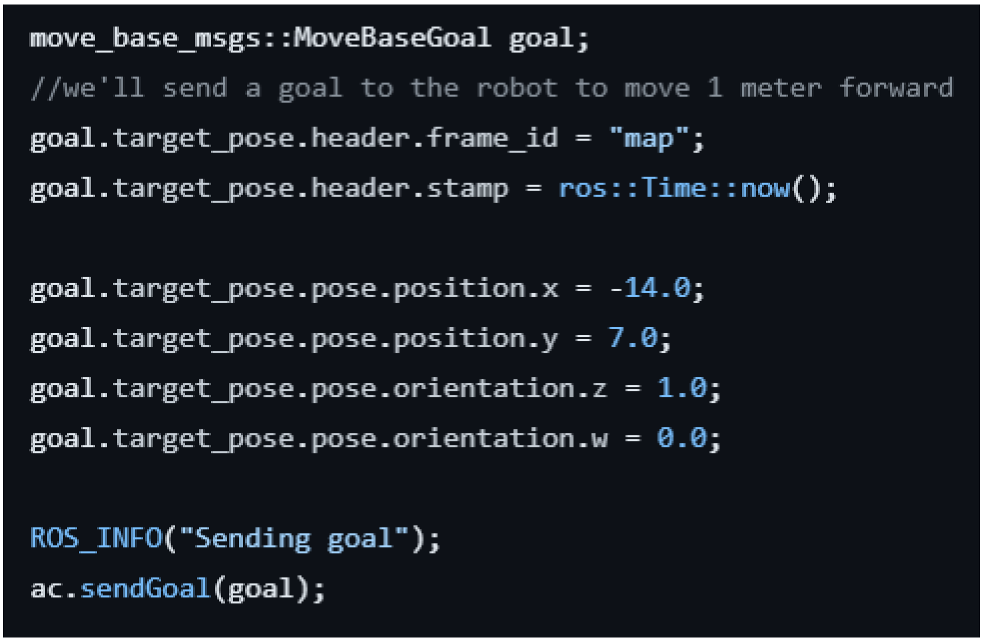
aruco_nav.cpp

```
tf::TransformListener listener;
tf::StampedTransform transform0,transform1,transform2, transform3,transform4,transform5, transform6, transform7, transform8;
tf::Transform Optical_to_base, Optical_to_map;

try{
  // listener.waitForTransform( "map", "base_footprint", ros::Time(0), ros::Duration(10.0) );
  // listener.lookupTransform( "map", "base_footprint", ros::Time(0), transform0 );

  listener.waitForTransform( "base_footprint", "base_link", ros::Time(0), ros::Duration(10.0) );
  listener.lookupTransform( "base_footprint", "base_link", ros::Time(0), transform1 );

  listener.waitForTransform( "base_link", "d435_link", ros::Time(0), ros::Duration(10.0) );
  listener.lookupTransform( "base_link", "d435_link", ros::Time(0), transform2 );

  listener.waitForTransform( "d435_link", "camera_bottom_screw_frame", ros::Time(0), ros::Duration(10.0) );
  listener.lookupTransform( "d435_link", "camera_bottom_screw_frame", ros::Time(0), transform3 );

  listener.waitForTransform( "camera_bottom_screw_frame", "camera_link", ros::Time(0), ros::Duration(10.0) );
  listener.lookupTransform( "camera_bottom_screw_frame", "camera_link", ros::Time(0), transform4 );

  listener.waitForTransform( "camera_link", "camera_depth_frame", ros::Time(0), ros::Duration(10.0) );
  listener.lookupTransform( "camera_link", "camera_depth_frame", ros::Time(0), transform5 );

  listener.waitForTransform( "camera_depth_frame", "camera_depth_optical_frame", ros::Time(0), ros::Duration(10.0) );
  listener.lookupTransform( "camera_depth_frame", "camera_depth_optical_frame", ros::Time(0), transform6 );

  // listener.waitForTransform( "camera_depth_optical_frame", "aruco_marker_frame", ros::Time(0), ros::Duration(10.0) );
  // listener.lookupTransform( "camera_depth_optical_frame", "aruco_marker_frame", ros::Time(0), transform7 );

}catch(tf::TransformException ex){
```

With this listener I get all the transformations from the camera frame down to the *"base_footprint"* frame going backwards in the *"tf"* tree.

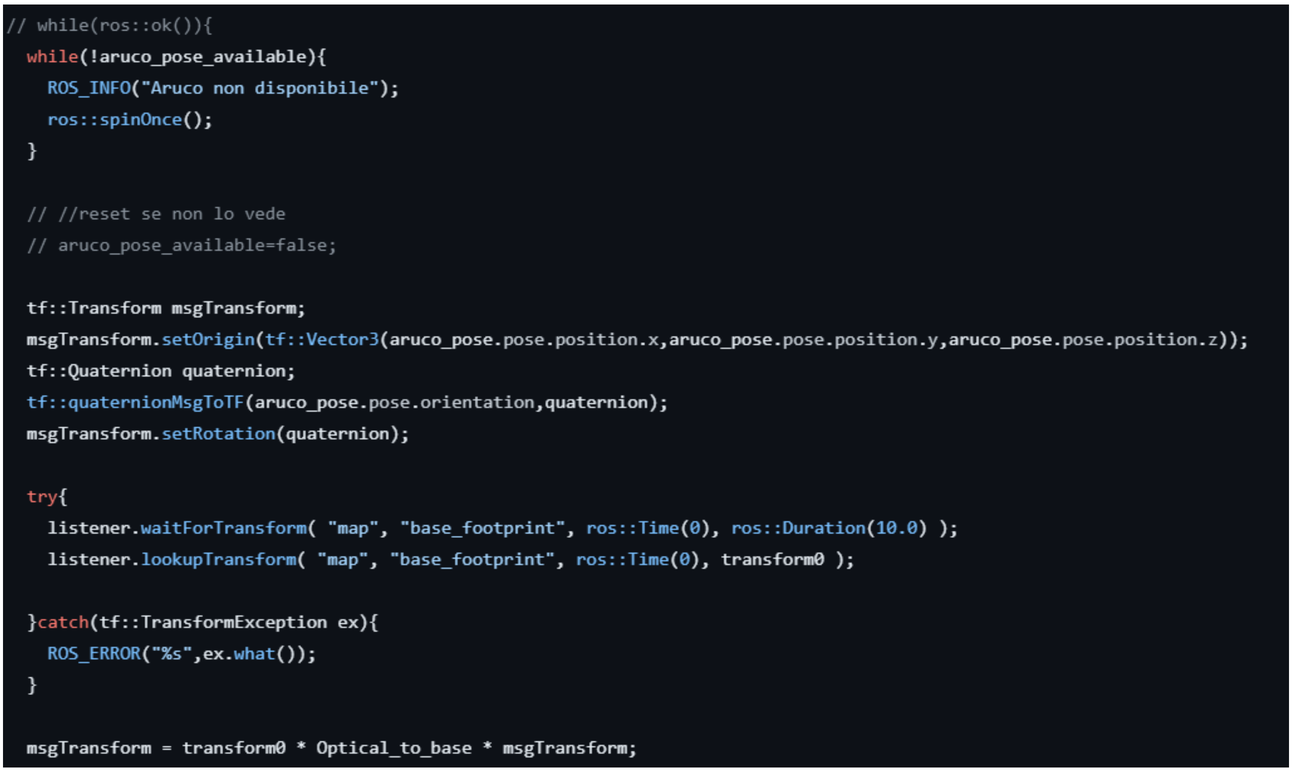


```
Optical_to_base = transform1 * transform2 * transform3* transform4 * transform5 * transform6;
```



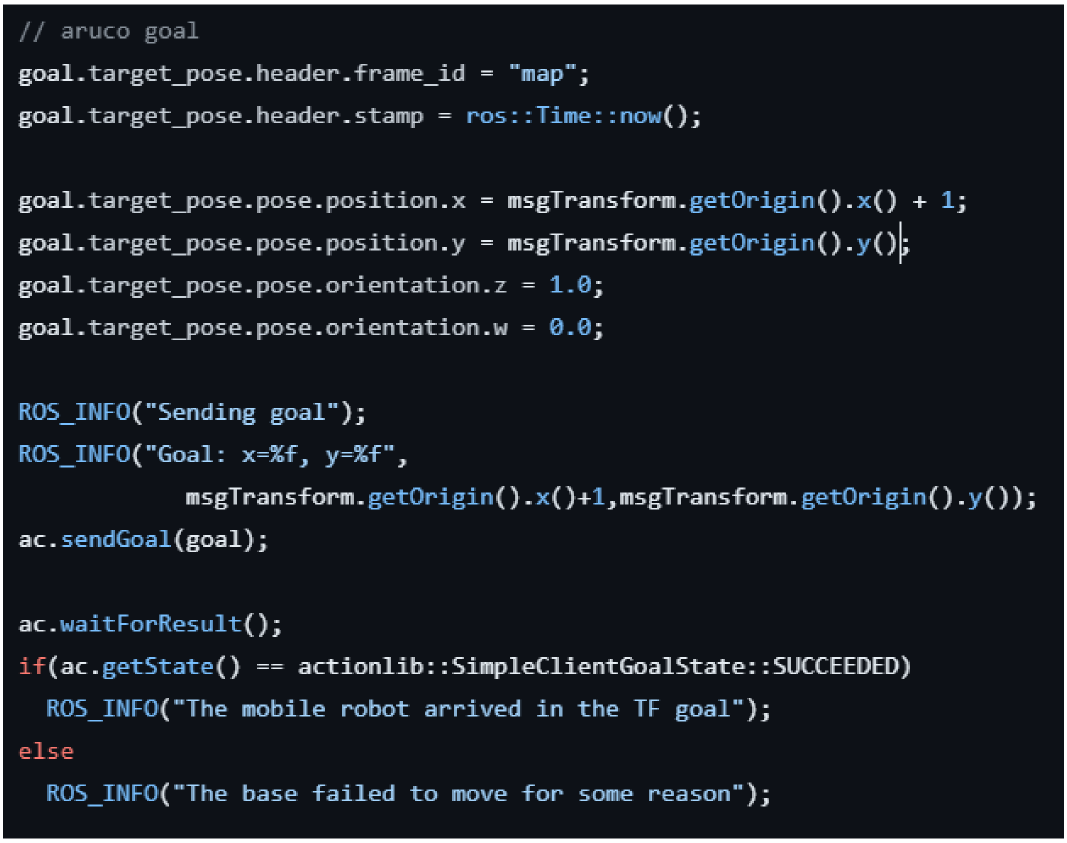
```
move_base_msgs::MoveBaseGoal goal;
//we'll send a goal to the robot to move 1 meter forward
goal.target_pose.header.frame_id = "map";
goal.target_pose.header.stamp = ros::Time::now();

goal.target_pose.pose.position.x = -14.0;
goal.target_pose.pose.position.y = 7.0;
goal.target_pose.pose.orientation.z = 1.0;
goal.target_pose.pose.orientation.w = 0.0;

ROS_INFO("Sending goal");
ac.sendGoal(goal);
```

I define a new goal in proximity of obstacle 9 to make the robot detect the *"ArUco"* marker



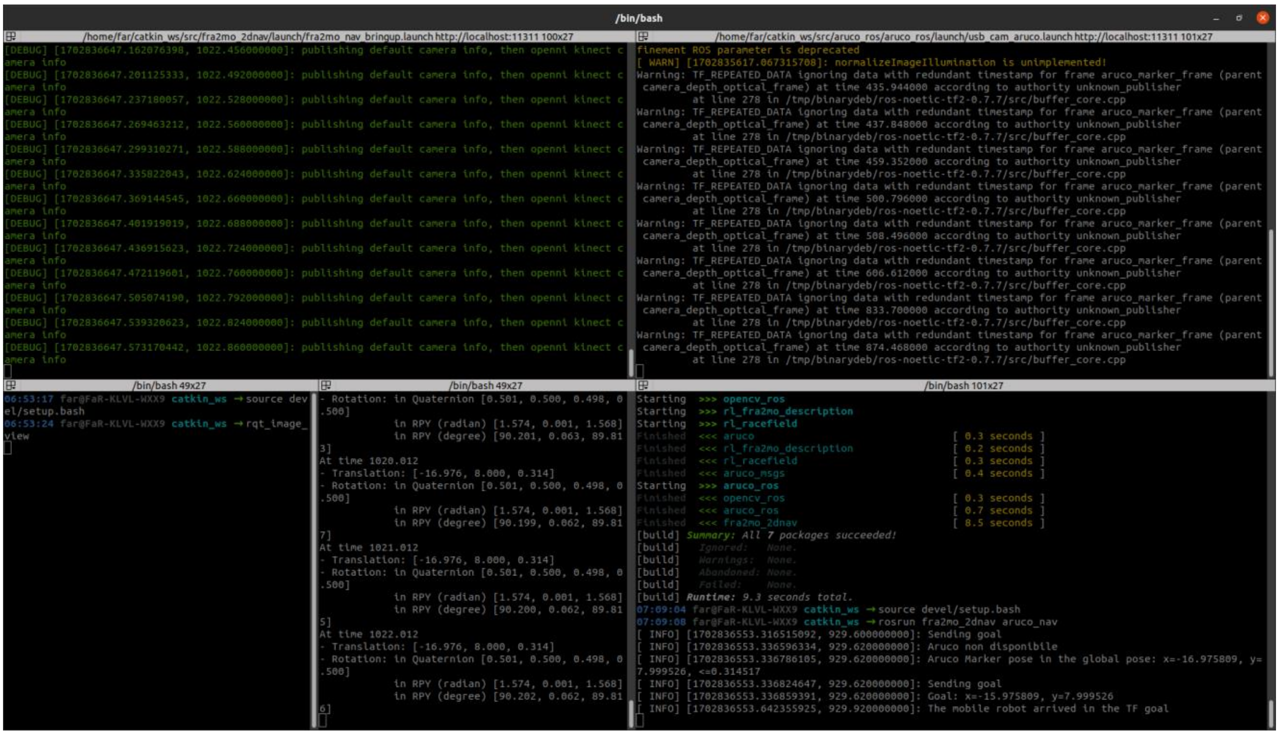
```
// while(ros::ok()){
  while(!aruco_pose_available){
    ROS_INFO("Aruco non disponibile");
    ros::spinOnce();
  }

  // //reset se non lo vede
  // aruco_pose_available=false;

  tf::Transform msgTransform;
  msgTransform.setOrigin(tf::Vector3(aruco_pose.pose.position.x,aruco_pose.pose.position.y,aruco_pose.pose.position.z));
  tf::Quaternion quaternion;
  tf::quaternionMsgToTF(aruco_pose.pose.orientation,quaternion);
  msgTransform.setRotation(quaternion);

  try{
    listener.waitForTransform( "map", "base_footprint", ros::Time(0), ros::Duration(10.0) );
    listener.lookupTransform( "map", "base_footprint", ros::Time(0), transform0 );

  }catch(tf::TransformException ex){
    ROS_ERROR("%s",ex.what());
  }

  msgTransform = transform0 * Optical_to_base * msgTransform;
```

When the *"ArUco"* pose becomes available, I setup another listener to compute the transform between the *"base_footprint"* and the *"map"* frame.



```
// aruco goal
goal.target_pose.header.frame_id = "map";
goal.target_pose.header.stamp = ros::Time::now();

goal.target_pose.pose.position.x = msgTransform.getOrigin().x() + 1;
goal.target_pose.pose.position.y = msgTransform.getOrigin().y();
goal.target_pose.pose.orientation.z = 1.0;
goal.target_pose.pose.orientation.w = 0.0;

ROS_INFO("Sending goal");
ROS_INFO("Goal: x=%f, y=%f",
            msgTransform.getOrigin().x()+1,msgTransform.getOrigin().y());
ac.sendGoal(goal);

ac.waitForResult();
if(ac.getState() == actionlib::SimpleClientGoalState::SUCCEEDED)
  ROS_INFO("The mobile robot arrived in the TF goal");
else
  ROS_INFO("The base failed to move for some reason");
```

After retrieving the pose of the marker, I can set another goal that fulfills the required specifications.



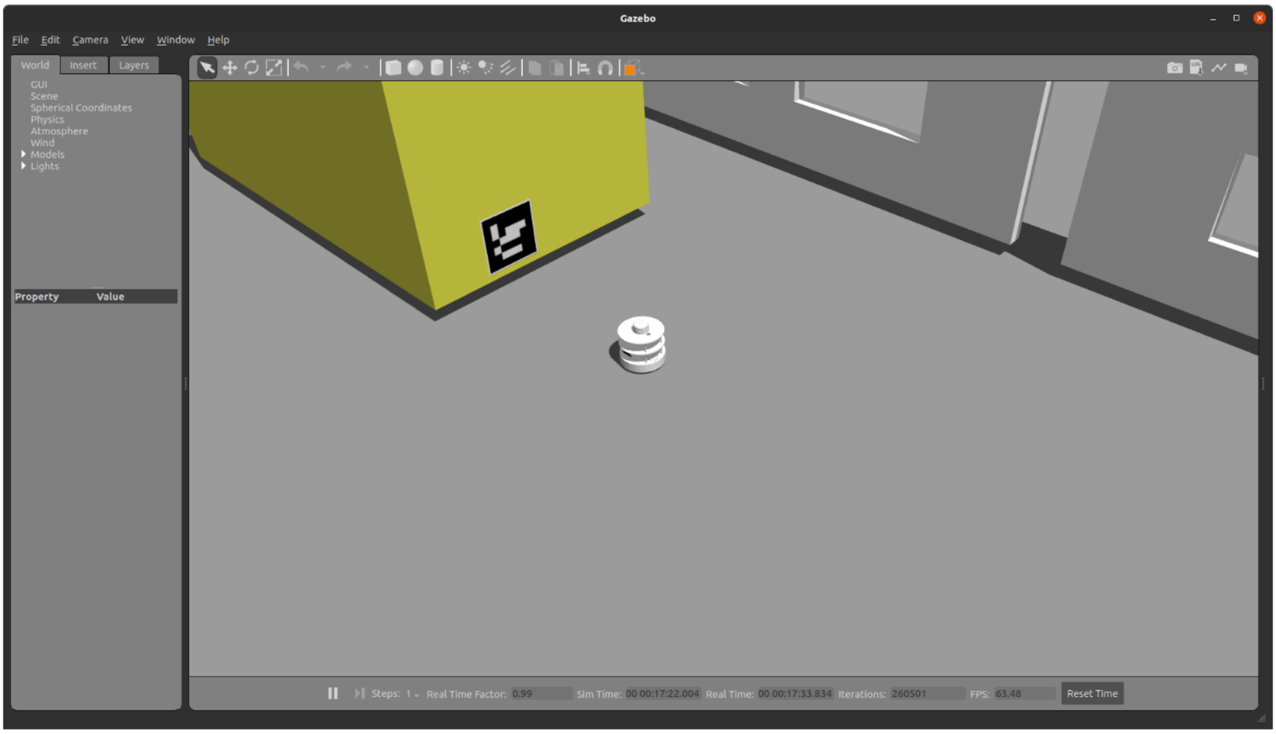
```
/home/far/catkin_ws/src/fra2mo_2dnav/launch/fra2mo_nav_bringup.launch http://localhost:11311 100x27
[DEBUG] [1702836647.162076398, 1022.456000000]: publishing default camera info, then openni kinect c
amera info
[DEBUG] [1702836647.201125333, 1022.492000000]: publishing default camera info, then openni kinect c
amera info
[DEBUG] [1702836647.237180057, 1022.528000000]: publishing default camera info, then openni kinect c
amera info
[DEBUG] [1702836647.269463212, 1022.560000000]: publishing default camera info, then openni kinect c
amera info
[DEBUG] [1702836647.299310271, 1022.588000000]: publishing default camera info, then openni kinect c
amera info
[DEBUG] [1702836647.335822043, 1022.624000000]: publishing default camera info, then openni kinect c
amera info
[DEBUG] [1702836647.369144545, 1022.660000000]: publishing default camera info, then openni kinect c
amera info
[DEBUG] [1702836647.401919019, 1022.688000000]: publishing default camera info, then openni kinect c
amera info
[DEBUG] [1702836647.436915623, 1022.724000000]: publishing default camera info, then openni kinect c
amera info
[DEBUG] [1702836647.472119601, 1022.760000000]: publishing default camera info, then openni kinect c
amera info
[DEBUG] [1702836647.505074190, 1022.792000000]: publishing default camera info, then openni kinect c
amera info
[DEBUG] [1702836647.539320623, 1022.824000000]: publishing default camera info, then openni kinect c
amera info
[DEBUG] [1702836647.573170442, 1022.860000000]: publishing default camera info, then openni kinect c
amera info
```

```
/home/far/catkin_ws/src/aruco_ros/aruco_ros/launch/usb_cam_aruco.launch http://localhost:11311 101x27
finement ROS parameter is deprecated
[ WARN] [1702835617.067315708]: normalizeImageIllumination is unimplemented!
Warning: TF_REPEATED_DATA ignoring data with redundant timestamp for frame aruco_marker_frame (parent
 camera_depth_optical_frame) at time 435.944000 according to authority unknown_publisher
         at line 278 in /tmp/binarydeb/ros-noetic-tf2-0.7.7/src/buffer_core.cpp
Warning: TF_REPEATED_DATA ignoring data with redundant timestamp for frame aruco_marker_frame (parent
 camera_depth_optical_frame) at time 437.848000 according to authority unknown_publisher
         at line 278 in /tmp/binarydeb/ros-noetic-tf2-0.7.7/src/buffer_core.cpp
Warning: TF_REPEATED_DATA ignoring data with redundant timestamp for frame aruco_marker_frame (parent
 camera_depth_optical_frame) at time 459.352000 according to authority unknown_publisher
         at line 278 in /tmp/binarydeb/ros-noetic-tf2-0.7.7/src/buffer_core.cpp
Warning: TF_REPEATED_DATA ignoring data with redundant timestamp for frame aruco_marker_frame (parent
 camera_depth_optical_frame) at time 500.796000 according to authority unknown_publisher
         at line 278 in /tmp/binarydeb/ros-noetic-tf2-0.7.7/src/buffer_core.cpp
Warning: TF_REPEATED_DATA ignoring data with redundant timestamp for frame aruco_marker_frame (parent
 camera_depth_optical_frame) at time 508.496000 according to authority unknown_publisher
         at line 278 in /tmp/binarydeb/ros-noetic-tf2-0.7.7/src/buffer_core.cpp
Warning: TF_REPEATED_DATA ignoring data with redundant timestamp for frame aruco_marker_frame (parent
 camera_depth_optical_frame) at time 606.612000 according to authority unknown_publisher
         at line 278 in /tmp/binarydeb/ros-noetic-tf2-0.7.7/src/buffer_core.cpp
Warning: TF_REPEATED_DATA ignoring data with redundant timestamp for frame aruco_marker_frame (parent
 camera_depth_optical_frame) at time 833.700000 according to authority unknown_publisher
         at line 278 in /tmp/binarydeb/ros-noetic-tf2-0.7.7/src/buffer_core.cpp
Warning: TF_REPEATED_DATA ignoring data with redundant timestamp for frame aruco_marker_frame (parent
 camera_depth_optical_frame) at time 874.468000 according to authority unknown_publisher
         at line 278 in /tmp/binarydeb/ros-noetic-tf2-0.7.7/src/buffer_core.cpp
```

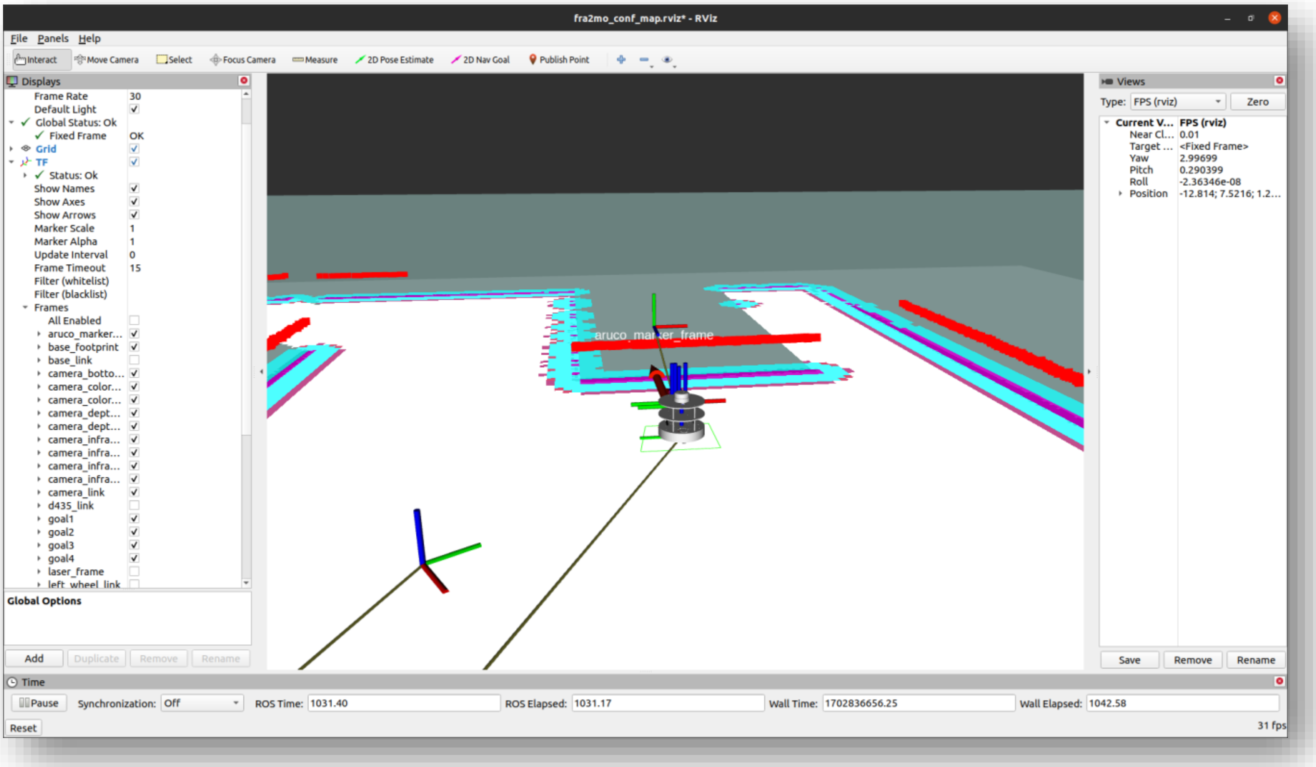
```
/bin/bash 49x27
06:53:17 far@FaR-KLVL-WXX9 catkin_ws →source dev
el/setup.bash
06:53:24 far@FaR-KLVL-WXX9 catkin_ws →rqt_image_
view
```

```
/bin/bash 49x27
- Rotation: in Quaternion [0.501, 0.500, 0.498, 0
.500]
            in RPY (radian) [1.574, 0.001, 1.568]
            in RPY (degree) [90.201, 0.063, 89.81
3]
At time 1020.012
- Translation: [-16.976, 8.000, 0.314]
- Rotation: in Quaternion [0.501, 0.500, 0.498, 0
.500]
            in RPY (radian) [1.574, 0.001, 1.568]
            in RPY (degree) [90.199, 0.062, 89.81
7]
At time 1021.012
- Translation: [-16.976, 8.000, 0.314]
- Rotation: in Quaternion [0.501, 0.500, 0.498, 0
.500]
            in RPY (radian) [1.574, 0.001, 1.568]
            in RPY (degree) [90.200, 0.062, 89.81
5]
At time 1022.012
- Translation: [-16.976, 8.000, 0.314]
- Rotation: in Quaternion [0.501, 0.500, 0.498, 0
.500]
            in RPY (radian) [1.574, 0.001, 1.568]
            in RPY (degree) [90.202, 0.062, 89.81
6]
```

```
/bin/bash 101x27
Starting  >>> opencv_ros
Starting  >>> rl_fra2mo_description
Starting  >>> rl_racefield
Finished  <<< aruco                                    [ 0.3 seconds ]
Finished  <<< rl_fra2mo_description                    [ 0.2 seconds ]
Finished  <<< rl_racefield                             [ 0.3 seconds ]
Finished  <<< aruco_msgs                               [ 0.4 seconds ]
Starting  >>> aruco_ros
Finished  <<< opencv_ros                               [ 0.3 seconds ]
Finished  <<< aruco_ros                                [ 0.7 seconds ]
Finished  <<< fra2mo_2dnav                             [ 8.5 seconds ]
[build] Summary: All 7 packages succeeded!
[build]   Ignored:   None.
[build]   Warnings:  None.
[build]   Abandoned: None.
[build]   Failed:    None.
[build] Runtime: 9.3 seconds total.
07:09:04 far@FaR-KLVL-WXX9 catkin_ws →source devel/setup.bash
07:09:08 far@FaR-KLVL-WXX9 catkin_ws →rosrun fra2mo_2dnav aruco_nav
[ INFO] [1702836553.316515092, 929.600000000]: Sending goal
[ INFO] [1702836553.336596334, 929.620000000]: Aruco non disponibile
[ INFO] [1702836553.336786105, 929.620000000]: Aruco Marker pose in the global pose: x=-16.975809, y=
7.999526, <=0.314517
[ INFO] [1702836553.336824647, 929.620000000]: Sending goal
[ INFO] [1702836553.336859391, 929.620000000]: Goal: x=-15.975809, y=7.999526
[ INFO] [1702836553.642355925, 929.920000000]: The mobile robot arrived in the TF goal
```



Steps: 1 Real Time Factor: 0.99 Sim Time: 00 00:17:22.004 Real Time: 00 00:17:33.834 Iterations: 260501 FPS: 63.48



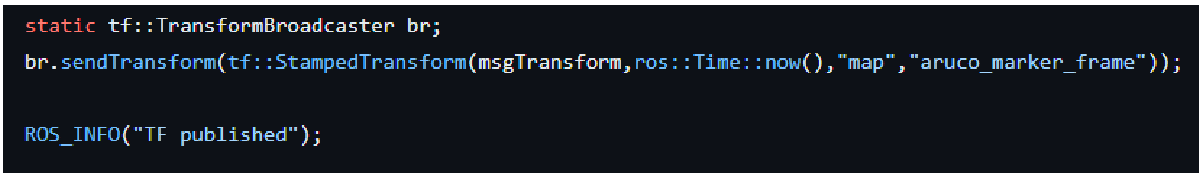
Frame Rate	30
Default Light	☑
✓ Global Status: Ok	
✓ Fixed Frame	OK
Grid	☑
TF	☑
✓ Status: Ok	
Show Names	☑
Show Axes	☑
Show Arrows	☑
Marker Scale	1
Marker Alpha	1
Update Interval	0
Frame Timeout	15
Filter (whitelist)	
Filter (blacklist)	
Frames	
All Enabled	☐
aruco_marker...	☑
base_footprint	☑
base_link	☐
camera_botto...	☑
camera_color...	☑
camera_color...	☑
camera_dept...	☑
camera_dept...	☑
camera_infra...	☑
camera_infra...	☑
camera_infra...	☑
camera_infra...	☑
camera_link	☑
d435_link	☐
goal1	☑
goal2	☑
goal3	☑
goal4	☑
laser_frame	☐
left_wheel_link	☐

Type: FPS (rviz)

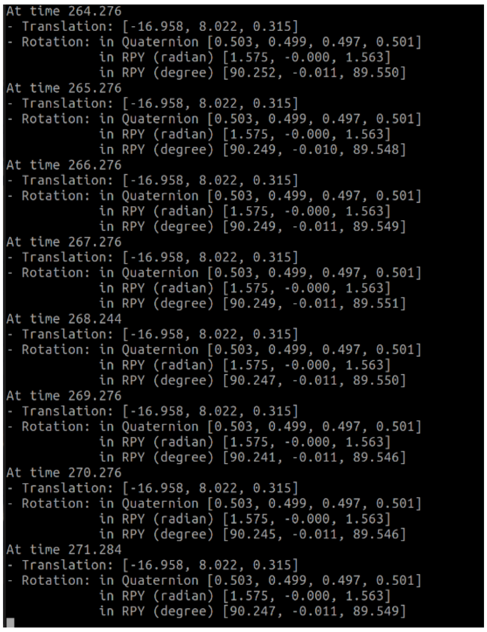
Current V...	FPS (rviz)
Near Cl...	0.01
Target ...	<Fixed Frame>
Yaw	2.99699
Pitch	0.290399
Roll	-2.36346e-08
Position	-12.814; 7.5216; 1.2...

Synchronization: Off ROS Time: 1031.40 ROS Elapsed: 1031.17 Wall Time: 1702836656.25 Wall Elapsed: 1042.58

c. Publish the *"ArUco"* pose as *"tf"*



```
static tf::TransformBroadcaster br;
br.sendTransform(tf::StampedTransform(msgTransform,ros::Time::now(),"map","aruco_marker_frame"));

ROS_INFO("TF published");
```



```
At time 264.276
- Translation: [-16.958, 8.022, 0.315]
- Rotation: in Quaternion [0.503, 0.499, 0.497, 0.501]
            in RPY (radian) [1.575, -0.000, 1.563]
            in RPY (degree) [90.252, -0.011, 89.550]
At time 265.276
- Translation: [-16.958, 8.022, 0.315]
- Rotation: in Quaternion [0.503, 0.499, 0.497, 0.501]
            in RPY (radian) [1.575, -0.000, 1.563]
            in RPY (degree) [90.249, -0.010, 89.548]
At time 266.276
- Translation: [-16.958, 8.022, 0.315]
- Rotation: in Quaternion [0.503, 0.499, 0.497, 0.501]
            in RPY (radian) [1.575, -0.000, 1.563]
            in RPY (degree) [90.249, -0.011, 89.549]
At time 267.276
- Translation: [-16.958, 8.022, 0.315]
- Rotation: in Quaternion [0.503, 0.499, 0.497, 0.501]
            in RPY (radian) [1.575, -0.000, 1.563]
            in RPY (degree) [90.249, -0.011, 89.551]
At time 268.244
- Translation: [-16.958, 8.022, 0.315]
- Rotation: in Quaternion [0.503, 0.499, 0.497, 0.501]
            in RPY (radian) [1.575, -0.000, 1.563]
            in RPY (degree) [90.247, -0.011, 89.550]
At time 269.276
- Translation: [-16.958, 8.022, 0.315]
- Rotation: in Quaternion [0.503, 0.499, 0.497, 0.501]
            in RPY (radian) [1.575, -0.000, 1.563]
            in RPY (degree) [90.241, -0.011, 89.546]
At time 270.276
- Translation: [-16.958, 8.022, 0.315]
- Rotation: in Quaternion [0.503, 0.499, 0.497, 0.501]
            in RPY (radian) [1.575, -0.000, 1.563]
            in RPY (degree) [90.245, -0.011, 89.546]
At time 271.284
- Translation: [-16.958, 8.022, 0.315]
- Rotation: in Quaternion [0.503, 0.499, 0.497, 0.501]
            in RPY (radian) [1.575, -0.000, 1.563]
            in RPY (degree) [90.247, -0.011, 89.549]
```

Using the *"tf echo"* command

Github repositories

Matteo Langella: https://github.com/matteolangella/RL-23-24-HW4.git
Raffaele Freschini: https://github.com/RFreschini/RL-23-24.git
Nicola Caliendo: https://github.com/NicoStayCali/Homework_RL