Moving the Robot

Estimated time to completion: **10 minutes**

5.4 Differential Drive Plugin

This plugin taps into the Gazeo simulator's core elements and moves the links and joints accordingly. You could build your own, but that goes beyond the scope of this course. So, instead, you will use a premade plugin widely used for **differential drive robots**.

To add it, place this piece of code in the URDF.

In [ ]:

```
<gazebo>
  <plugin filename="libgazebo_ros_diff_drive.so" name="differential_drive_controller">

    <!-- wheels -->
    <left_joint>joint_left_wheel</left_joint>
    <right_joint>joint_right_wheel</right_joint>

    <!-- kinematics -->
    <wheel_separation>0.1</wheel_separation>
    <wheel_diameter>0.07</wheel_diameter>

    <!-- limits -->
    <max_wheel_torque>1.0</max_wheel_torque>
    <max_wheel_acceleration>2.0</max_wheel_acceleration>

    <!-- output -->
    <publish_odom>true</publish_odom>
    <publish_odom_tf>true</publish_odom_tf>

    <odometry_frame>odom</odometry_frame>
    <robot_base_frame>base_link</robot_base_frame>

  </plugin>
</gazebo>
```

This plugin has many features, so review them again:

- It publishes the odometry data, **in this case, in the /odom topic**.
- It publishes the **TF transforms** to visualize in RVIZ all the positions of the wheels.
- You set **the two joints corresponding to the two wheels for the differential drive**.
- You set some physical dimensions to calculate the differential drive parameters. That is the:
  - wheel_separation: In this case, it is the size of the box_bot chassis.
  - wheel_diameter: The cylinder diameter of the wheels.
- You also set the maximum forces for the wheels.
- AND VERY IMPORTANT: Set the `robot_base_frame`, which must be the root frame to which all the others are attached. Otherwise, this plugin will fail if you set it to something else, such as `chassis`.

Add these two elements and get your **box_bot** moving:

▶ Execute in Terminal 1

In [ ]:
```
cd ~/ros2_ws/src
```

In [ ]:
```
touch my_box_bot_gazebo/launch/spawn_robot_ros2_control.launch.xml
```

In [ ]:
```
touch my_box_bot_description/urdf/box_bot_physcal_control.urdf
```

In [ ]:
```
touch my_box_bot_description/launch/urdf_visualize_control.launch.py
```

spawn_robot_ros2_control.launch.xml

In [ ]:
```
<?xml version='1.0' ?>
<launch>
  <!-- Publish URDF file in robot_description topic -->
  <include file="$(find-pkg-share my_box_bot_description)/launch/urdf_visualize_control.launch.py"/>
  <!-- Read robot_description and spawn in gazebo running sim -->
  <include file="$(find-pkg-share my_box_bot_gazebo)/launch/spawn_robot_description.launch.py"/>
</launch>
```

box_bot_physcal_control.urdf

In [ ]:

```
<?xml version="1.0"?>
<robot name="box_bot">

  <material name="red">
      <color rgba="1.0 0.0 0.0 1"/>
  </material>

  <material name="green_light">
      <color rgba="0.0 1.0 0.0 1"/>
  </material>

  <material name="green_dark">
    <color rgba="0.0 0.5 0.0 1"/>
  </material>

  <material name="blue">
      <color rgba="0.0 0.0 1.0 1"/>
  </material>

  <link name="base_link">
  </link>


  <!-- Body -->
  <link name="chassis">
    <visual>
      <geometry>
        <mesh filename="package://my_box_bot_description/meshes/cute_cube.dae" scale="0.1 0.1 0.1"/>
      </geometry>
    </visual>

    <collision>
      <geometry>
        <box size="0.1 0.1 0.1"/>
      </geometry>
    </collision>
```

```
  <inertial>
    <mass value="0.5"/>
    <origin rpy="0 0 0" xyz="0 0 0"/>
    <inertia ixx="0.0008333333333333335" ixy="0" ixz="0" iyy="0.0008333333333333335" iyz="0" izz="0.000833333333
  </inertial>

</link>

<joint name="base_link_joint" type="fixed">
  <origin rpy="0 0 0" xyz="0 0 0" />
  <parent link="base_link" />
  <child link="chassis" />
</joint>

<!-- Wheel Left -->
<link name="left_wheel">
    <visual>
      <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
      <geometry>
        <cylinder length="0.001" radius="0.035"/>
      </geometry>
      <material name="red"/>
    </visual>

    <collision>
      <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
      <geometry>
        <cylinder length="0.001" radius="0.035"/>
      </geometry>
    </collision>

    <inertial>
      <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
      <mass value="0.05"/>
      <inertia ixx="1.531666666666667e-05" ixy="0" ixz="0" iyy="1.531666666666667e-05" iyz="0" izz="3.0625000000
    </inertial>
```

```
</link>

<gazebo reference="left_wheel">
  <kp>10000000000000000000000000.0</kp>
  <kd>10000000000000000000000000.0</kd>
  <mu1>10.0</mu1>
  <mu2>10.0</mu2>
  <material>Gazebo/Green</material>
</gazebo>


<joint name="joint_left_wheel" type="continuous">
  <origin rpy="0 0 0" xyz="0 0.05 -0.025"/>
  <child link="left_wheel"/>
  <parent link="chassis"/>
  <axis rpy="0 0 0" xyz="0 1 0"/>
  <limit effort="10000" velocity="1000"/>
  <joint_properties damping="1.0" friction="1.0"/>
</joint>

<!-- Wheel Right -->
<link name="right_wheel">
    <visual>
      <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
      <geometry>
        <cylinder length="0.001" radius="0.035"/>
      </geometry>
      <material name="green"/>
    </visual>

    <collision>
      <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
      <geometry>
        <cylinder length="0.001" radius="0.035"/>
      </geometry>
    </collision>
```

```
    <inertial>
      <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
      <mass value="0.05"/>
      <inertia ixx="1.531666666666667e-05" ixy="0" ixz="0" iyy="1.531666666666667e-05" iyz="0" izz="3.0625000000
    </inertial>
</link>

<gazebo reference="right_wheel">
  <kp>10000000000000000000000000.0</kp>
  <kd>10000000000000000000000000.0</kd>
  <mu1>10.0</mu1>
  <mu2>10.0</mu2>
  <material>Gazebo/Orange</material>
</gazebo>

<joint name="joint_right_wheel" type="continuous">
  <origin rpy="0 0 0" xyz="0 -0.05 -0.025"/>
  <child link="right_wheel"/>
  <parent link="chassis"/>
  <axis rpy="0 0 0" xyz="0 1 0"/>
  <limit effort="10000" velocity="1000"/>
  <joint_properties damping="1.0" friction="1.0"/>
</joint>


<!-- Caster Wheel Front -->
<link name="front_yaw_link">
    <visual>
      <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
      <geometry>
        <cylinder length="0.001" radius="0.0045000000000000005"/>
      </geometry>
      <material name="blue"/>
    </visual>

    <collision>
      <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
```

```
      <geometry>
        <cylinder length="0.001" radius="0.0045000000000000005"/>
      </geometry>
    </collision>

    <inertial>
        <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
        <mass value="0.001"/>
        <inertia ixx="5.145833333333334e-09" ixy="0" ixz="0" iyy="5.145833333333334e-09" iyz="0" izz="1.01250000
    </inertial>

</link>

<joint name="front_yaw_joint" type="continuous">
  <origin rpy="0 0 0" xyz="0.04 0 -0.05" />
  <parent link="chassis" />
  <child link="front_yaw_link" />
  <axis xyz="0 0 1" />
  <limit effort="1000.0" velocity="100.0" />
  <dynamics damping="0.0" friction="0.1"/>
</joint>

  <gazebo reference="front_yaw_link">
      <material>Gazebo/Blue</material>
  </gazebo>



<link name="front_roll_link">
    <visual>
      <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
      <geometry>
        <cylinder length="0.001" radius="0.0045000000000000005"/>
      </geometry>
      <material name="red"/>
    </visual>
```

```
    <collision>
      <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
      <geometry>
        <cylinder length="0.001" radius="0.0045000000000000005"/>
      </geometry>
    </collision>

    <inertial>
        <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
        <mass value="0.001"/>
        <inertia ixx="5.145833333333334e-09" ixy="0" ixz="0" iyy="5.145833333333334e-09" iyz="0" izz="1.01250000
    </inertial>
</link>

<joint name="front_roll_joint" type="continuous">
  <origin rpy="0 0 0" xyz="0 0 0" />
  <parent link="front_yaw_link" />
  <child link="front_roll_link" />
  <axis xyz="1 0 0" />
  <limit effort="1000.0" velocity="100.0" />
  <dynamics damping="0.0" friction="0.1"/>
</joint>

  <gazebo reference="front_roll_link">
      <material>Gazebo/Red</material>
  </gazebo>


<link name="front_pitch_link">
  <visual>
    <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
    <geometry>
      <sphere radius="0.010"/>
    </geometry>
    <material name="green_dark"/>
  </visual>
```

```
    <collision>
      <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
      <geometry>
        <sphere radius="0.010"/>
      </geometry>
    </collision>

    <inertial>
        <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
        <mass value="0.001"/>
        <inertia ixx="4e-08" ixy="0" ixz="0" iyy="4e-08" iyz="0" izz="4e-08"/>
    </inertial>
  </link>

  <gazebo reference="front_pitch_link">
    <kp>10000000000000000000000000000.0</kp>
    <kd>10000000000000000000000000000.0</kd>
    <mu1>0.5</mu1>
    <mu2>0.5</mu2>
    <material>Gazebo/Purple</material>
  </gazebo>

  <joint name="front_pitch_joint" type="continuous">
    <origin rpy="0 0 0" xyz="0 0 0" />
    <parent link="front_roll_link" />
    <child link="front_pitch_link" />
    <axis xyz="0 1 0" />
    <limit effort="1000.0" velocity="100.0" />
    <dynamics damping="0.0" friction="0.1"/>
  </joint>

<!-- Caster Wheel Back -->
  <link name="back_yaw_link">
    <visual>
        <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
        <geometry>
          <cylinder length="0.001" radius="0.0045000000000000005"/>
```

```
      </geometry>
      <material name="blue"/>
    </visual>

    <collision>
      <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
      <geometry>
        <cylinder length="0.001" radius="0.0045000000000000005"/>
      </geometry>
    </collision>

    <inertial>
        <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
        <mass value="0.001"/>
        <inertia ixx="5.145833333333334e-09" ixy="0" ixz="0" iyy="5.145833333333334e-09" iyz="0" izz="1.01250000
    </inertial>
</link>

<joint name="back_yaw_joint" type="continuous">
  <origin rpy="0 0 0" xyz="-0.04 0 -0.05" />
  <parent link="chassis" />
  <child link="back_yaw_link" />
  <axis xyz="0 0 1" />
  <limit effort="1000.0" velocity="100.0" />
  <dynamics damping="0.0" friction="0.1"/>
</joint>

  <gazebo reference="back_yaw_link">
      <material>Gazebo/Blue</material>
  </gazebo>



<link name="back_roll_link">
    <visual>
      <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
      <geometry>
```

```
        <cylinder length="0.001" radius="0.0045000000000000005"/>
      </geometry>
      <material name="red"/>
    </visual>

    <collision>
      <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
      <geometry>
        <cylinder length="0.001" radius="0.0045000000000000005"/>
      </geometry>
    </collision>

    <inertial>
        <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
        <mass value="0.001"/>
        <inertia ixx="5.145833333333334e-09" ixy="0" ixz="0" iyy="5.145833333333334e-09" iyz="0" izz="1.01250000
    </inertial>
</link>

<joint name="back_roll_joint" type="continuous">
  <origin rpy="0 0 0" xyz="0 0 0" />
  <parent link="back_yaw_link" />
  <child link="back_roll_link" />
  <axis xyz="1 0 0" />
  <limit effort="1000.0" velocity="100.0" />
  <dynamics damping="0.0" friction="0.1"/>
</joint>

  <gazebo reference="back_roll_link">
      <material>Gazebo/Red</material>
  </gazebo>



<link name="back_pitch_link">
  <visual>
    <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
```

```
    <geometry>
      <sphere radius="0.010"/>
    </geometry>
    <material name="green_light"/>
  </visual>

  <collision>
    <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
    <geometry>
      <sphere radius="0.010"/>
    </geometry>
  </collision>

  <inertial>
      <origin rpy="0 1.5707 1.5707" xyz="0 0 0"/>
      <mass value="0.001"/>
      <inertia ixx="4e-08" ixy="0" ixz="0" iyy="4e-08" iyz="0" izz="4e-08"/>
  </inertial>
</link>

<gazebo reference="back_pitch_link">
  <kp>10000000000000000000000000.0</kp>
  <kd>10000000000000000000000000.0</kd>
  <mu1>0.5</mu1>
  <mu2>0.5</mu2>
  <material>Gazebo/Yellow</material>
</gazebo>

<joint name="back_pitch_joint" type="continuous">
  <origin rpy="0 0 0" xyz="0 0 0" />
  <parent link="back_roll_link" />
  <child link="back_pitch_link" />
  <axis xyz="0 1 0" />
  <limit effort="1000.0" velocity="100.0" />
  <dynamics damping="0.0" friction="0.1"/>
</joint>
```

```
<!-- PLUGINS -->

<!-- JOINT PUBLISHER -->
<gazebo>
  <plugin name="box_bot_joint_state" filename="libgazebo_ros_joint_state_publisher.so">
    <ros>
        <remapping>~/out:=joint_states</remapping>
    </ros>
    <update_rate>30</update_rate>

    <joint_name>joint_left_wheel</joint_name>
    <joint_name>joint_right_wheel</joint_name>
    <joint_name>front_yaw_joint</joint_name>
    <joint_name>back_yaw_joint</joint_name>
    <joint_name>front_roll_joint</joint_name>
    <joint_name>back_roll_joint</joint_name>
    <joint_name>front_pitch_joint</joint_name>
    <joint_name>back_pitch_joint</joint_name>

  </plugin>
</gazebo>

<!-- Differential drive -->
<gazebo>
  <plugin filename="libgazebo_ros_diff_drive.so" name="differential_drive_controller">

    <!-- wheels -->
    <left_joint>joint_left_wheel</left_joint>
    <right_joint>joint_right_wheel</right_joint>

    <!-- kinematics -->
    <wheel_separation>0.1</wheel_separation>
    <wheel_diameter>0.07</wheel_diameter>

    <!-- limits -->
    <max_wheel_torque>1.0</max_wheel_torque>
    <max_wheel_acceleration>2.0</max_wheel_acceleration>
```

```
      <!-- output -->
      <publish_odom>true</publish_odom>
      <publish_odom_tf>true</publish_odom_tf>

      <odometry_frame>odom</odometry_frame>
      <robot_base_frame>base_link</robot_base_frame>

    </plugin>
  </gazebo>



</robot>
```

urdf_visualize_control.launch.py

In [ ]:

```
import os

from ament_index_python.packages import get_package_share_directory
from launch import LaunchDescription
from launch.substitutions import Command
from launch_ros.actions import Node

# this is the function launch  system will look for
def generate_launch_description():

    ####### DATA INPUT ##########
    urdf_file = 'box_bot_physcal_control.urdf'
    #xacro_file = "box_bot.xacro"
    package_description = "my_box_bot_description"

    ####### DATA INPUT END ##########
    print("Fetching URDF ==>")
    robot_desc_path = os.path.join(get_package_share_directory(package_description), "urdf", urdf_file)

    # Robot State Publisher

    robot_state_publisher_node = Node(
        package='robot_state_publisher',
        executable='robot_state_publisher',
        name='robot_state_publisher_node',
        emulate_tty=True,
        parameters=[{'use_sim_time': True, 'robot_description': Command(['xacro ', robot_desc_path])}],
        output="screen"
    )

    # RVIZ Configuration
    rviz_config_dir = os.path.join(get_package_share_directory(package_description), 'rviz', 'urdf_vis.rviz')


    rviz_node = Node(
            package='rviz2',
            executable='rviz2',
```

```
            output='screen',
            name='rviz_node',
            parameters=[{'use_sim_time': True}],
            arguments=['-d', rviz_config_dir])

    # create and return launch description object
    return LaunchDescription(
        [
            robot_state_publisher_node,
            rviz_node
        ]
    )
```

If you launch the following command, you should get something similar to this:

▶ Execute in Terminal 1

In [ ]:
```
cd ~/ros2_ws
```

In [ ]:
```
colcon build
```

In [ ]:
```
source install/setup.bash
```

In [ ]:
```
ros2 launch my_box_bot_gazebo start_world.launch.py
```

▶ Execute in Terminal 2

In [ ]:
```
cd ~/ros2_ws
```

In [ ]:
```
source install/setup.bash
```

In [ ]:
```
ros2 launch my_box_bot_gazebo spawn_robot_ros2_control.launch.xml
```

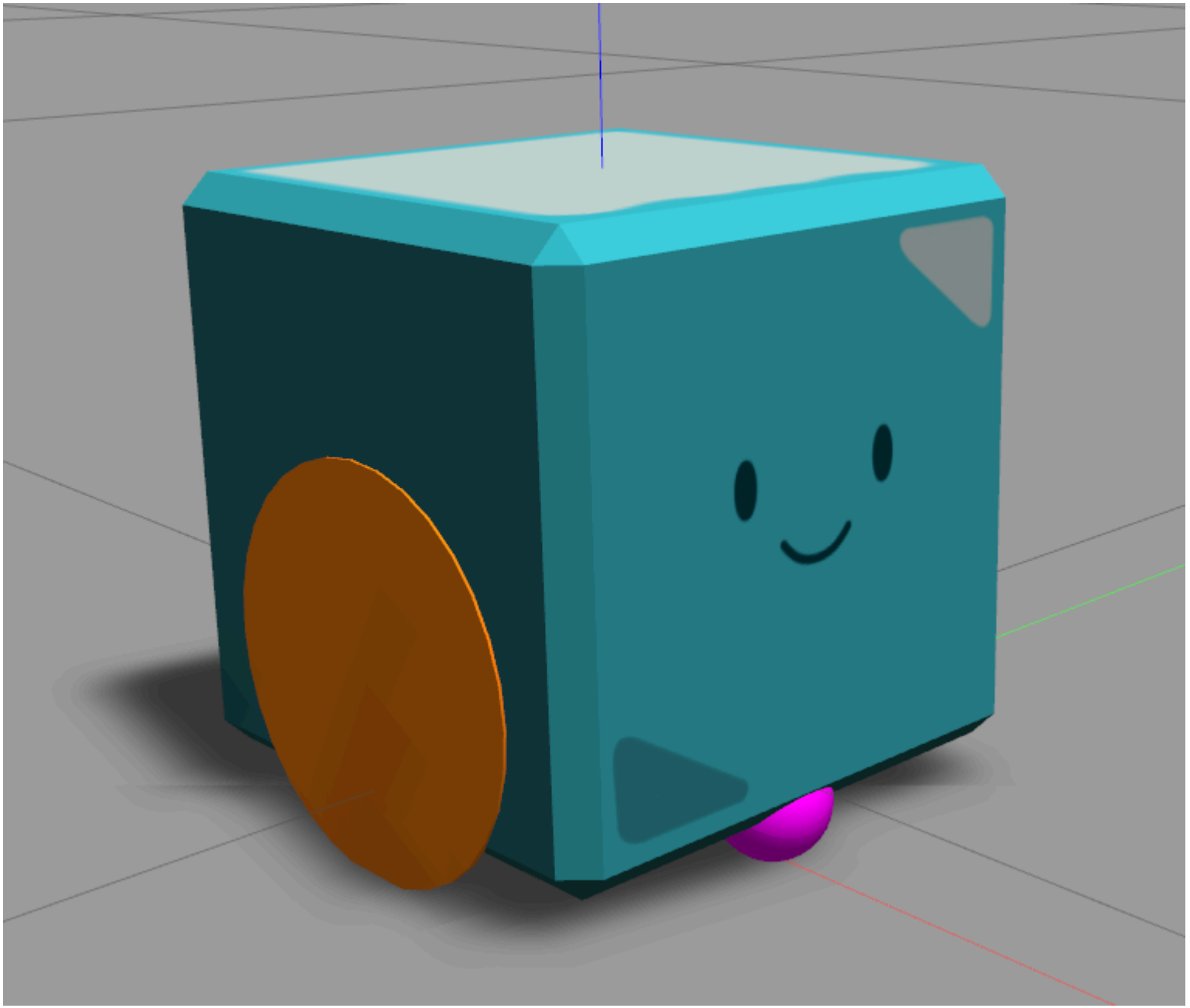
You should be able to move it using the **teleop_twist_keyboard** ROS 2 package:

▶ Execute in Terminal 3

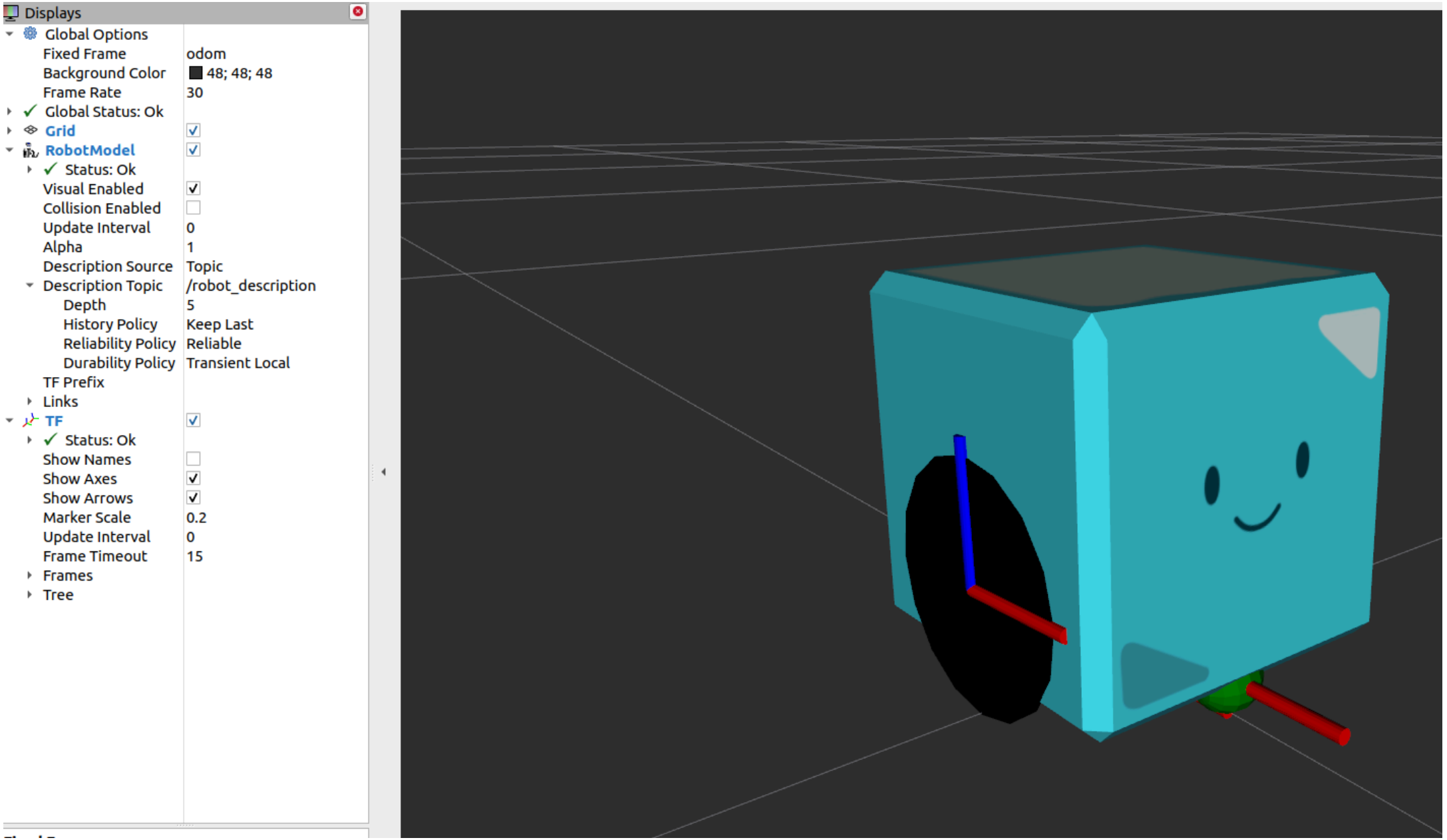
In [ ]:
```
cd ~/ros2_ws
```

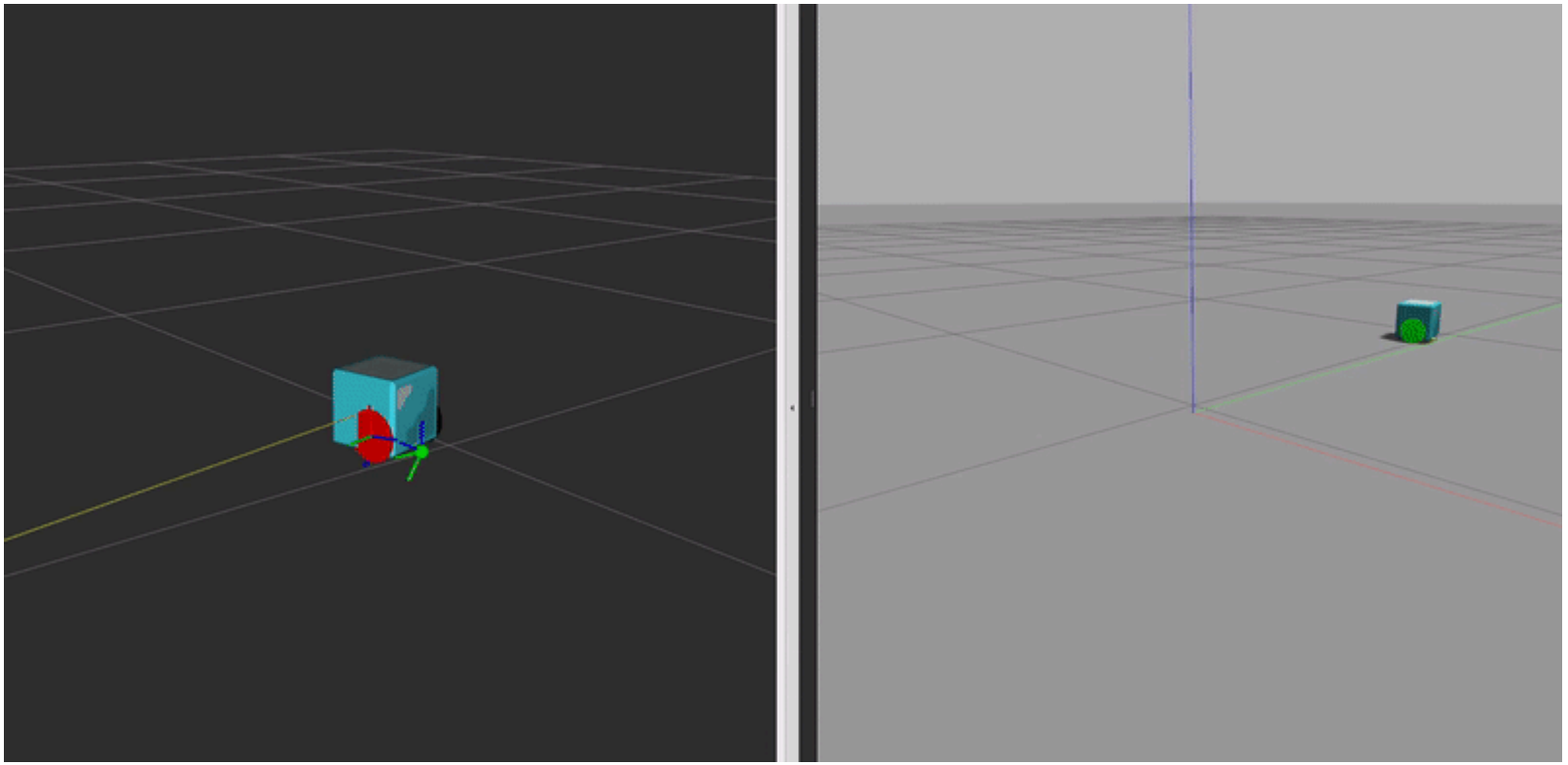
In [ ]:
```
source install/setup.bash
```

In [ ]:
```
ros2 run teleop_twist_keyboard teleop_twist_keyboard
```



You can change the fixed frame to **Odom** because now the **differential drive is publishing the TF** for the `Odom frame` , among others.


Displays
Global Options
Fixed Frame odom
Background Color 48; 48; 48
Frame Rate 30
Global Status: Ok
Grid
RobotModel
Status: Ok
Visual Enabled
Collision Enabled
Update Interval 0
Alpha 1
Description Source Topic
Description Topic /robot_description
Depth 5
History Policy Keep Last
Reliability Policy Reliable
Durability Policy Transient Local
TF Prefix
Links
TF
Status: Ok
Show Names
Show Axes
Show Arrows
Marker Scale 0.2
Update Interval 0
Frame Timeout 15
Frames
Tree



Here, you can see that the **Gazebo simulation** and **RVIZ2** can move using the **differential drive plugin** and that all the **TFs and Joint States** are being published correctly.

16/11/2023