Pacote ROS 2 com o Modelo de um Robô Manipulador

Walter Fetter Lages

fetter@ece.ufrgs.br

Universidade Federal do Rio Grande do Sul
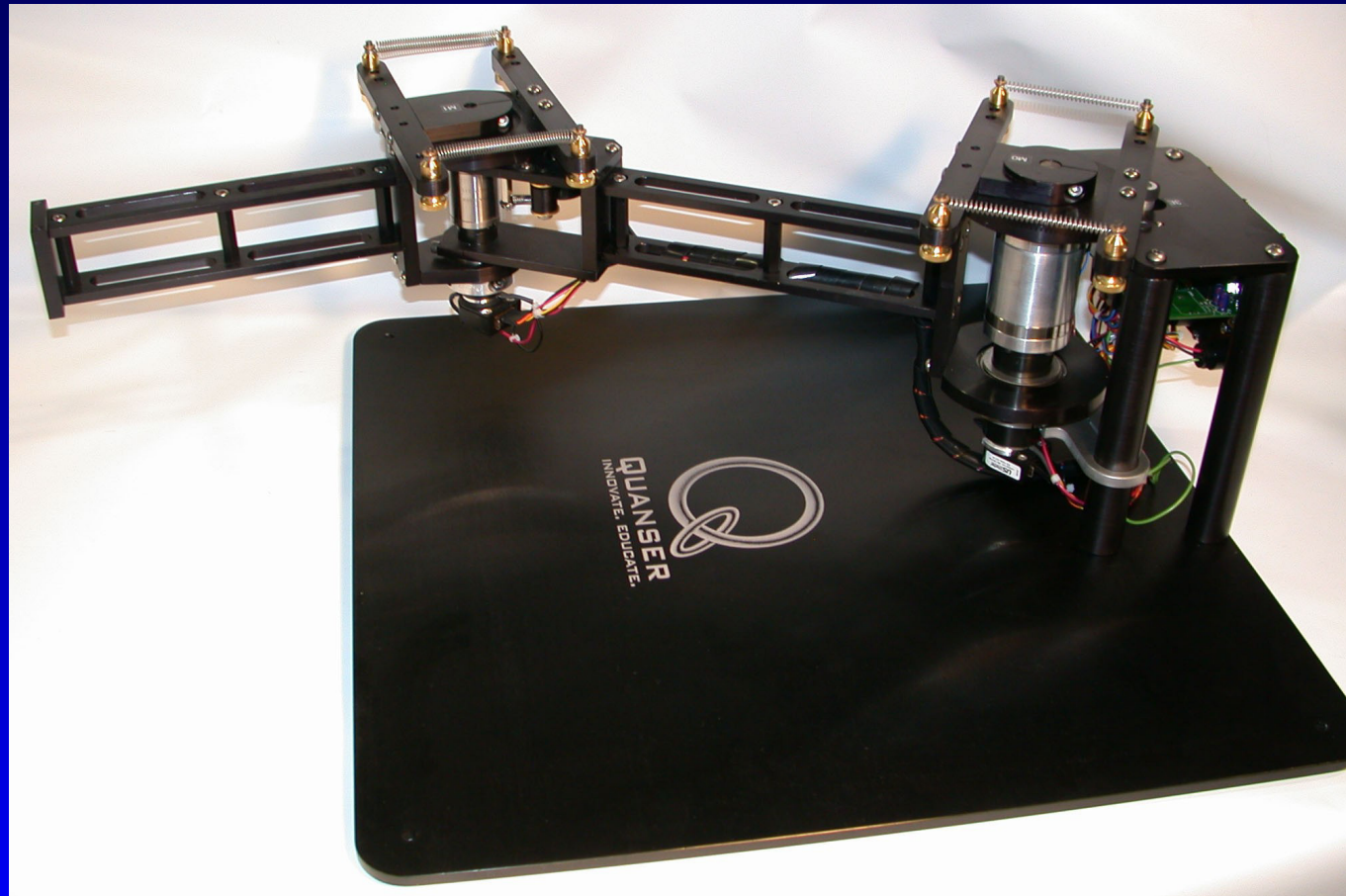
Escola de Engenharia

Departamento de Sistemas Elétricos de Automação e Energia

ENG10052 Laboratório de Robótica

Robô Quanser 2DSFJ

- Possui juntas flexíveis
- Serão consideradas juntas rígidas

Pacote para Descrever o Robô

```
q2d_description/
├──CMakeLists.txt
├──config/
│   ├──controller_manager.yaml
│   └──gazebo.yaml
├──launch/
│   ├──display.launch.xml
│   ├──gazebo.launch.xml
│   └──q2d.launch.xml
├──meshes/
│   ├── .
│   │   .
│   └──collision/
│       └── .
│           .
├──urdf/
│   └──q2d.urdf
├──package.xml
├──rviz/
│   └──display.rviz
```

Criação do Pacote

source ~/colcon_ws/install/setup.bash

cd ~/colcon_ws/src

ros2 pkg create q2d_description

```
colcon_ws/
├─build/
├─install/
├─log/
└─src/
  └─q2d_description/
    ├─CMakeLists.txt
    └─package.xml
```

`package.xml`

- Editar o arquivo `package.xml` para preenchimento dos meta-dados do pacote
  - Descrição
  - Mantenedor
  - Licença
  - Dependências
  - Exportações

<export>

   <build_type>ament_cmake**</build_type>**

   <gazebo_ros gazebo_model_path="${prefix}/.."/>

</export>

---

cd q2d_description

nano package.xml

CMakeLists.txt

- Editar `CMakeLists.txt` para incluir a *tag*:

---

install(DIRECTORY config launch urdf meshes rviz

    DESTINATION share/${PROJECT_NAME}

)

Modelo em URDF

- `urdf/q2d.urdf`

```xml
<?xml version="1.0"?>
<robot name="q2d">

  <link name="origin_link"/>

  <link name="base_link">
    <inertial>
      <origin xyz="0.074571214 −0.00008502114 0.039600060500000"/>
      <mass value="6.9621272394"/>
      <inertia ixx="0.12983348950453222" ixy="0.0000414079177225"
        ixz="−0.03019200341025967" iyy="0.17857934382693791"
        iyz="0.00000393635720902" izz="0.25812143874621400" />
    </inertial>
  </inertial>
```

Modelo em URDF

```xml
<visual>
  <geometry>
    <mesh filename="package://q2d_description/meshes/base.STL"/>
  </geometry>
  <material name="base_material">
    <color rgba="0.55 0.45 0.45 1.0"/>
  </material>
</visual>

<collision>
  <geometry>
    <mesh filename="package://q2d_description/meshes/base.STL"/>
  </geometry>
</collision>
</link>
```

Modelo em URDF

```xml
<joint name="origin_joint" type="fixed">
  <parent link="origin_link"/>
  <child link="base_link"/>
</joint>


<link name="shoulder_active_link">
  <inertial>
    <origin xyz="0.0252456823 −0.0000002723 0.06470401873"/>
    <mass value="0.19730261508"/>
    <inertia ixx="0.00038685518702305" ixy="0.0000000055222416"
      ixz="−0.00000031340718614" iyy="0.00010241438913870"
      iyz="−0.00000000015426019" izz="0.00047879093657893" />
  </inertial>
```

```xml
<visual>
  <geometry>
    <mesh filename="package://q2d_description/meshes/shoulder_active.STL"/>
  </geometry>
  <material name="shoulder_active_material">
    <color rgba="0.55 0.55 0.45 1.0"/>
  </material>
</visual>

<collision>
  <geometry>
    <mesh filename="package://q2d_description/meshes/collision/shoulder_active.STL"/>
  </geometry>
</collision>
</link>
```

Modelo em URDF

```
<joint name="shoulder_active_joint" type="revolute">
  <parent link="base_link"/>
  <child link="shoulder_active_link"/>
  <origin xyz="0 0 0.1477"/>
  <axis xyz="0 0 1"/>
  <limit lower="−1.570796326794897" upper="1.570796326794897"
    velocity="2.27" effort="27.94" />
  <dynamics damping="29.7914" />
</joint>

<link name="shoulder_passive_link">
  <inertial>
    <mass value="1.26475817816"/>
    <origin xyz="0.1516344805 −0.0004828845 −0.00016382412"/>
    <inertia ixx="0.00346199967740929" ixy="−0.00010902049981923"
      ixz="−0.00401182173261703" iyy="0.03314904030482527"
      iyz="0.00005087359051462" izz="0.03113579694057124" />
  </inertial>
```

Modelo em URDF

```
<visual>
  <geometry>
    <mesh filename="package://q2d_description/meshes/
shoulder_passive.STL"/>
  </geometry>
  <material name="shoulder_passive_material">
    <color rgba="0.45 0.55 0.45 1.0"/>
  </material>
</visual>

<collision>
  <geometry>
    <mesh filename="package://q2d_description/meshes/collision/
shoulder_passive.STL"/>
  </geometry>
</collision>
</link>
```

Modelo em URDF

```
<joint name="shoulder_passive_joint" type="fixed">
  <parent link="shoulder_active_link"/>
  <child link="shoulder_passive_link"/>
</joint>


<link name="elbow_active_link">
  <inertial>
    <mass value="0.19712951877"/>
    <origin xyz="0.02548273493 −0.0000002263 0.05254513577"/>
    <inertia ixx="0.00038850510800265" ixy="0.0000000052121416"
      ixz="0.00000404728675587" iyy="0.00010146693248154"
      iyz="0.00000000002789435" izz="0.00048091942023028" />
  </inertial>
```

Modelo em URDF

```
<visual>
  <geometry>
    <mesh filename="package://q2d_description/meshes/elbow_active.
  STL"/>
  </geometry>
  <material name="elbow_active_material">
    <color rgba="0.45 0.55 0.55 1.0"/>
  </material>
</visual>

<collision>
  <geometry>
    <mesh filename="package://q2d_description/meshes/collision/
  elbow_active.STL"/>
  </geometry>
</collision>
</link>
```

Modelo em URDF

```
<joint name="elbow_active_joint" type="revolute">
  <parent link="shoulder_passive_link"/>
  <child link="elbow_active_link"/>
  <origin xyz="0.343 0 0"/>
  <axis xyz="0 0 1"/>
  <limit lower="-1.570796326794897" upper="1.570796326794897"
    velocity="23.08" effort="13.62" />
  <dynamics damping="2.6404" />
</joint>

<link name="elbow_passive_link">
  <inertial>
    <mass value="0.67529215765"/>
    <origin xyz="0.06204831581 0.0000013809 0.01489882531"/>
    <inertia ixx="0.00132247071698947" ixy="-0.0000000605403474"
      ixz="-0.00090893541574333" iyy="0.00774007102253750"
      iyz="0.0000000624688369" izz="0.00751638349361413" />
  </inertial>
```

Modelo em URDF

```
<visual>
  <geometry>
    <mesh filename="package://q2d_description/meshes/elbow_passive.
  STL"/>
  </geometry>
  <material name="elbow_passive_material">
    <color rgba="0.45 0.45 0.55 1.0"/>
  </material>
</visual>

<collision>
  <geometry>
    <mesh filename="package://q2d_description/meshes/collision/
  elbow_passive.STL"/>
  </geometry>
</collision>
</link>
```

Modelo em URDF

```xml
<joint name="elbow_passive_joint" type="fixed">
  <parent link="elbow_active_link"/>
  <child link="elbow_passive_link"/>
</joint>


<link name="tool_link"/>


<joint name="end_joint" type="fixed">
  <parent link="elbow_passive_link"/>
  <child link="tool_link"/>
  <origin xyz="0.267 0 0"/>
</joint>
```

Modelo em URDF

```
<transmission name="shoulder_active_transmission">
    <type>transmission_interface/SimpleTransmission</type>
    <joint name="shoulder_active_joint">
        <hardwareInterface>hardware_interface/EffortJointInterface</hardwareInterface>
    </joint>
    <actuator name="shoulder_motor">
        <mechanicalReduction>1</mechanicalReduction>
    </actuator>
</transmission>
```

Modelo em URDF

```
<transmission name="elbow_active_transmission">
  <type>transmission_interface/SimpleTransmission</type>
  <joint name="elbow_active_joint">
    <hardwareInterface>hardware_interface/EffortJointInterface</hardwareInterface>
  </joint>
  <actuator name="elbow_motor">
    <mechanicalReduction>1</mechanicalReduction>
  </actuator>
</transmission>
```

Modelo em URDF

```
<gazebo reference="base_link">
  <visual>
    <material>
      <ambient>0.55 0.45 0.45 1.0</ambient>
      <diffuse>0.55 0.45 0.45 1.0</diffuse>
      <specular>0.55 0.45 0.45 1.0</specular>
      <emissive>0.0 0.0 0.0 0.0</emissive>
    </material>
  </visual>
</gazebo>
```

Modelo em URDF

```
<gazebo reference="shoulder_active_link">
  <visual>
    <material>
      <ambient>0.55 0.55 0.45 1.0</ambient>
      <diffuse>0.55 0.55 0.45 1.0</diffuse>
      <specular>0.55 0.55 0.45 1.0</specular>
      <emissive>0.0 0.0 0.0 0.0</emissive>
    </material>
  </visual>
</gazebo>
```

Modelo em URDF

```
<gazebo reference="shoulder_passive_link">
   <visual>
      <material>
         <ambient>0.55 0.45 0.45 1.0</ambient>
         <diffuse>0.55 0.45 0.45 1.0</diffuse>
         <specular>0.55 0.45 0.45 1.0</specular>
         <emissive>0.0 0.0 0.0 0.0</emissive>
      </material>
   </visual>
</gazebo>
```

Modelo em URDF

```
<gazebo reference="elbow_active_link">
   <visual>
     <material>
        <ambient>0.45 0.55 0.55 1.0</ambient>
        <diffuse>0.45 0.55 0.55 1.0</diffuse>
        <specular>0.45 0.55 0.55 1.0</specular>
        <emissive>0.0 0.0 0.0 0.0</emissive>
     </material>
   </visual>
</gazebo>
```

Modelo em URDF

```
<gazebo reference="elbow_passive_link">
  <visual>
    <material>
      <ambient>0.45 0.45 0.45 1.0</ambient>
      <diffuse>0.45 0.45 0.45 1.0</diffuse>
      <specular>0.45 0.45 0.45 1.0</specular>
      <emissive>0.0 0.0 0.0 0.0</emissive>
    </material>
  </visual>
</gazebo>
```

Modelo em URDF

```xml
<ros2_control name="GazeboSystem" type="system">
  <hardware>
    <plugin>gazebo_ros2_control/GazeboSystem</plugin>
  </hardware>

  <joint name="shoulder_active_joint">
    <command_interface name="effort">
      <param name="min">−27.94</param>
      <param name="max">27.94</param>
    </command_interface>
    <state_interface name="position"/>
    <state_interface name="velocity"/>
    <state_interface name="effort"/>
  </joint>
```

Modelo em URDF

```xml
<joint name="elbow_active_joint">
  <command_interface name="effort">
    <param name="min">-13.62</param>
    <param name="max">13.62</param>
  </command_interface>
  <state_interface name="position"/>
  <state_interface name="velocity"/>
  <state_interface name="effort"/>
</joint>
</ros2_control>
```

```
<gazebo>
  <plugin filename="libgazebo_ros2_control.so" name="
  gazebo_ros2_control">
    <robot_param>robot_description</robot_param>
    <robot_param_node>robot_state_publisher</robot_param_node>
    <parameters>$(find q2d_description)/config/controller_manager.yaml
    </parameters>
  </plugin>
</gazebo>
</robot>
```

controller_manager.yaml

controller_manager:

    ros__parameters:

        update_rate: 1000

        use_sim_time: true

Modelo em URDF

- Editar o arquivo `urdf/q2d.urdf`

  ---

  cd ~/colcon_ws/src/q2d_description

  mkdir urdf

  cd urdf

  nano q2d.urdf

- Ou baixar o repositório `q2d`

  ---

  cd ~/colcon_ws/src

  git clone −b $ROS_DISTRO http://git.ece.ufrgs.br/q2d

  touch q2d/q2d_bringup/COLCON_IGNORE

  touch q2d/q2d_teleop/COLCON_IGNORE

Repositório q2d

```
q2d/
└─q2d_bringup/
│ └─CMakeLists.txt
│ └─package.xml
│ └─ ⋮
└─q2d_description/
│ └─CMakeLists.txt
│ └─package.xml
│ └─ ⋮
└─q2d_teleop/
  └─CMakeLists.txt
  └─package.xml
  └─ ⋮
```

'Compilar' e Reconfigurar o Ambiente

cd ~/colcon_ws

colcon build −−symlink−install

source ~/colcon_ws/install/setup.bash

Pacote `joint_state_publisher_gui`

- Interface gráfica para obter e publicar as variáveis de junta

---

sudo apt install ros−$ROS_DISTRO−joint−state−publisher−gui

Nodo `joint_state_publisher_gui`

- Mostra janela com *sliders* e permite alterar as variáveis de junta

- Publica as variáveis de junta no tópico `/joint_states`

- Não se está simulando a dinâmica do robô
  - É como mover as juntas com a mão

- Permite movimentar o robô no Rviz sem usar controlador

- Não serve para movimentar o robô no Gazebo
  - No Gazebo é necessário usar controlador, pois a dinâmica é simulada

Arquivos de *Launch*

- Arquivos de *launch* servem para lançar nodos
- No ROS 1 são arquivos em XML
- No ROS 2 podem ser em XML, Python ou YAML
  - Em XML e YAML nem todas as funcionalidades do ROS 2 são suportadas
- São executados com o comando `ros2 launch`

---

ros2 launch q2d_description display.launch.xml use_gui:=true

---

q2d.launch.xml

- Lança o nodo `robot_state_publisher`
    - Publica as transformações entre os sistemas de coordenadas do robô no tópico `/tf`
    - Publica a descrição do robô no tópico `/robot_description`

\<launch\>

  \<arg name="use_sim_time" **default**="false"/\>

  \<node name="robot_state_publisher" **pkg**="robot_state_publisher"

    exec="robot_state_publisher"\>

   \<param name="robot_description" **value**="$(command 'xacro $(

   find−pkg−share q2d_description)/urdf/q2d.urdf')" **type**="str"/\>

   \<param name="use_sim_time" **value**="$(var use_sim_time)"/\>

  \</node\>

\</launch\>

display.launch.xml

```xml
<launch>

  <arg name="use_gui" default="false"/>
  <node pkg="tf2_ros" exec="static_transform_publisher" name="
    q2d_origin_publisher" args="0 0 0 0 0 0 1 map origin_link" />
  <node if="$(var use_gui)" name="joint_state_publisher" pkg="
    joint_state_publisher_gui" exec="joint_state_publisher_gui" />
  <include file="$(find-pkg-share q2d_description)/launch/q2d.
    launch.xml"/>
  <node name="rviz" pkg="rviz2" exec="rviz2" args="-d $(find-
    pkg-share q2d_description)/rviz/display.rviz" />
</launch>
```
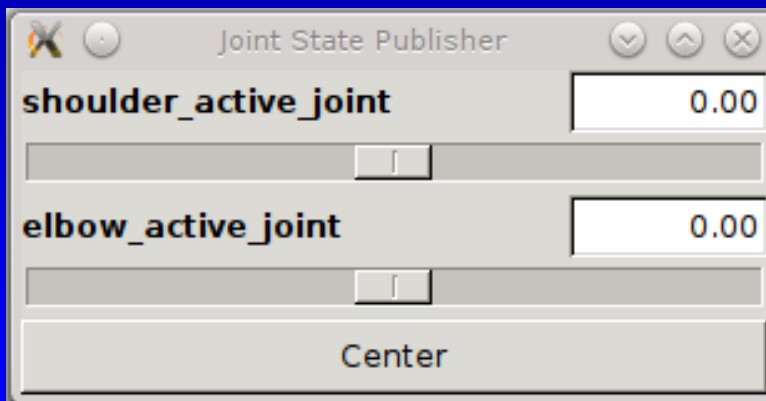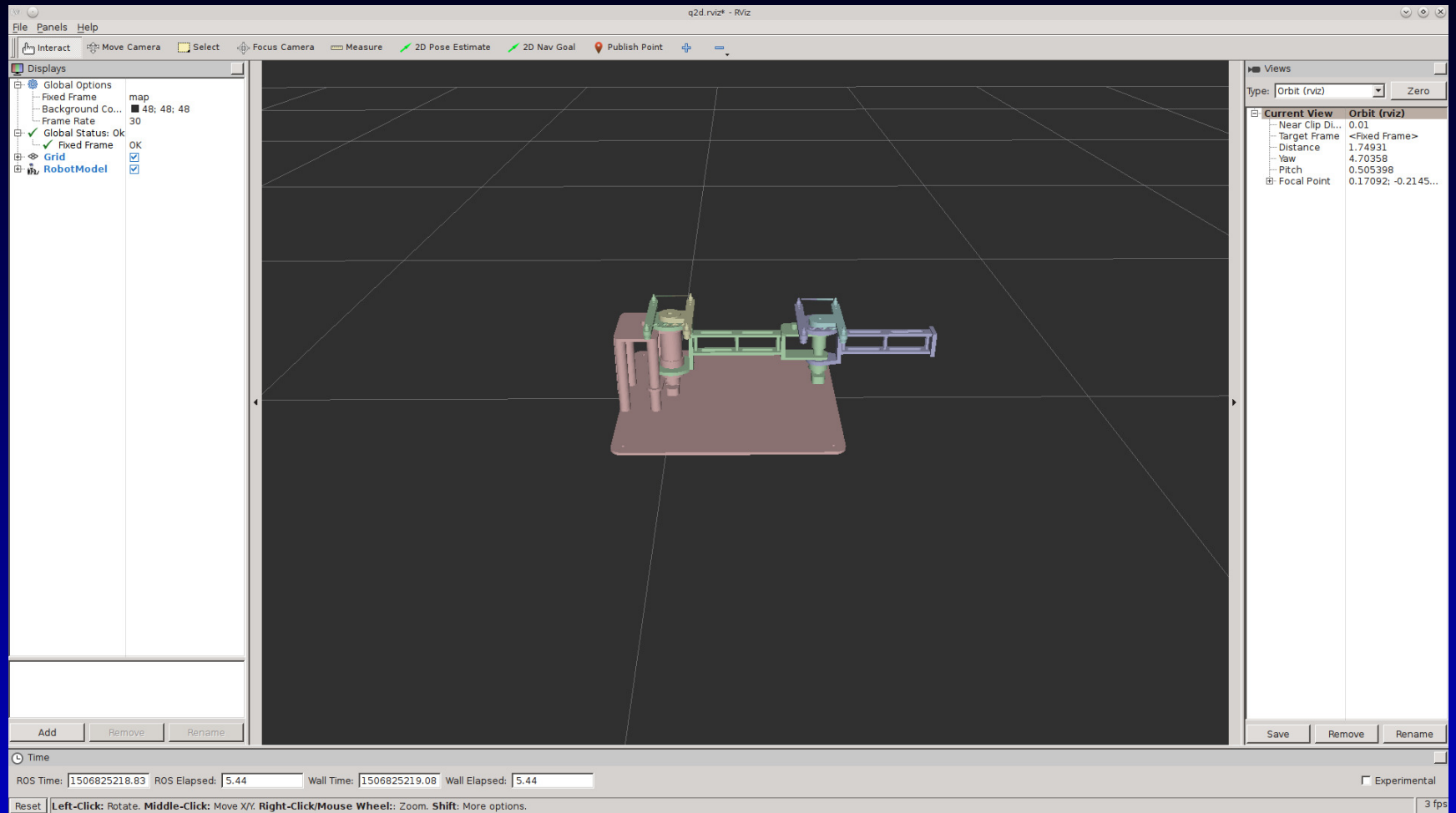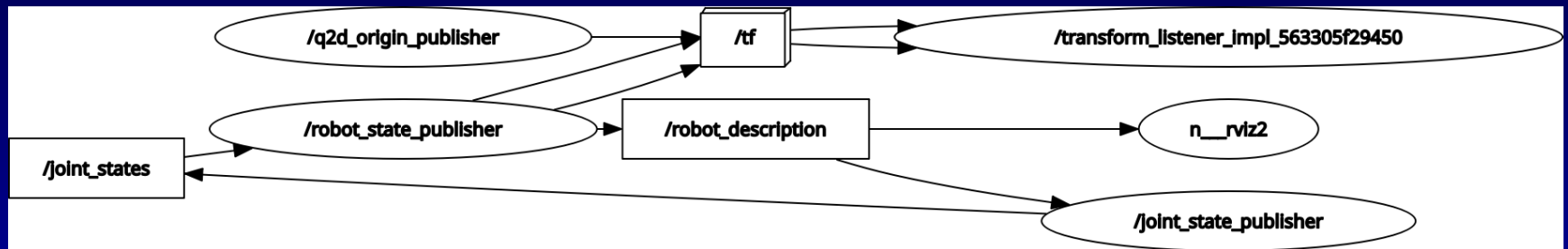
Visualização no Rviz

Gráfico de Computação

- Em outro terminal:

---

source ~/colcon_ws/install/setup.bash

rqt_graph &

---

ros2 topic

- Listar tópicos

  ros2 topic list

- Verificar o tipo da mensagem

  ros2 topic info /joint_states

- Visualizar o que é publicado no tópico

  ros2 topic echo /joint_states

- Obter a taxa com que mensagens são publicadas

  ros2 topic hz /joint_states

ros2 topic

- Publicar mensagem

---

ros2 topic pub /turtle1/cmd_vel geometry_msgs/msg/Twist '{

linear: {x: 1.0}, angular: {z: 1.0} }'

- Não é muito prático, em geral se encapsula em um *script*

Simulação no Gazebo

- Instalação do Gazebo para uso com o ROS 2

---

sudo apt install gazebo11

sudo apt install ros−$ROS_DISTRO−gazebo−ros−pkgs

Simulação no Gazebo

- `launch/gazebo.launch.xml`

```xml
<launch>
  <arg name="pause" default="true"/>
  <arg name="gui" default="true"/>
  <arg name="use_sim_time" default="true"/>

  <include file="$(find-pkg-share gazebo_ros)/launch/gazebo.launch.py">
    <arg name="pause" value="$(var pause)"/>
    <arg name="gui" value="$(var gui)"/>
    <arg name="use_sim_time" value="$(var use_sim_time)"/>
    <arg name="world" value="worlds/empty_sky.world" />
    <arg name="extra_gazebo_args" value="--ros-args --params-file
      $(find-pkg-share q2d_description)/config/gazebo.yaml"/>
  </include>
```

Simulação no Gazebo

```xml
<include file="$(find-pkg-share q2d_description)/launch/q2d.launch.xml
   ">
  <arg name="use_sim_time" value="$(var use_sim_time)"/>
</include>

<node name="q2d_spawner" pkg="gazebo_ros" exec="spawn_entity.py"
   args="-topic robot_description -entity q2d" />
</launch>
```

- Executar

  ros2 launch q2d_description gazebo.launch.xml

gazebo.yaml

gazebo:

    ros__parameters:
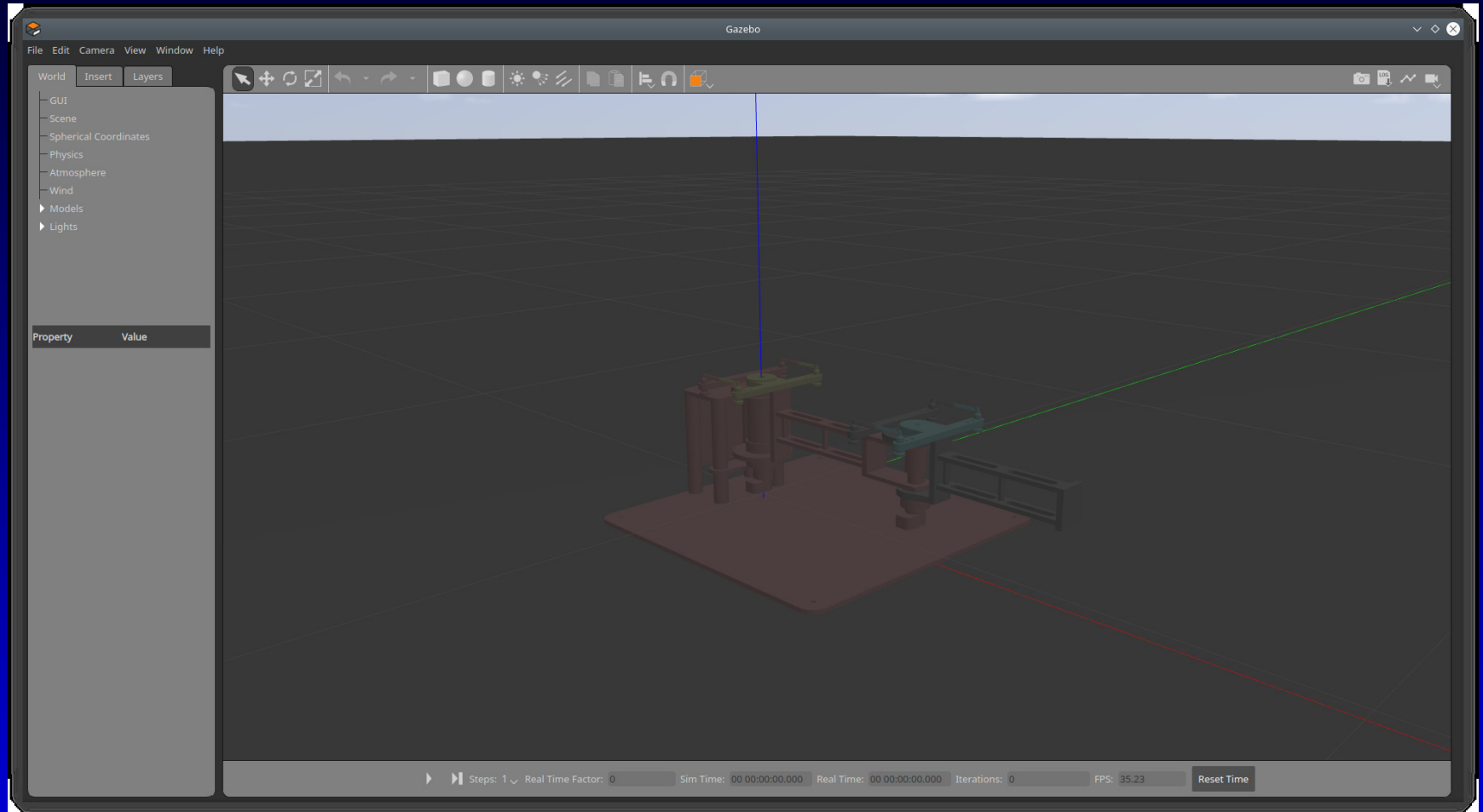
        # This is the /clock update rate

        publish_rate: 1000.0

Simulação no Gazebo

Gráfico de Computação

- `rqt_graph`