

# Using XARCRO/URDF

Defining a robot using the Unified Robot Description Format

---

Jan van Hulzen

September 25, 2025

# Overview



## Assignment 1: Installation

- Check if Rviz2 is installed:

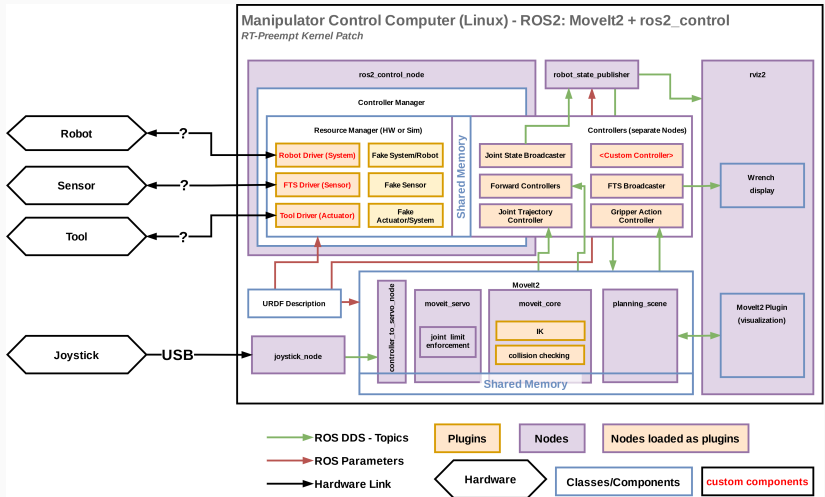
```
ros2 run rviz2 rviz2
```

- Install moveit2:  
<https://moveit.ai/install-moveit2/binary/>
- Check installation:

```
ros2 launch moveit2_tutorials demo.launch.py
```

- Bore-out preventer: [https://moveit.picknik.ai/main/doc/tutorials/quickstart\\_in\\_rviz/quickstart\\_in\\_rviz\\_tutorial.html#getting-started](https://moveit.picknik.ai/main/doc/tutorials/quickstart_in_rviz/quickstart_in_rviz_tutorial.html#getting-started)

# Overview software architecture ROS2 Control



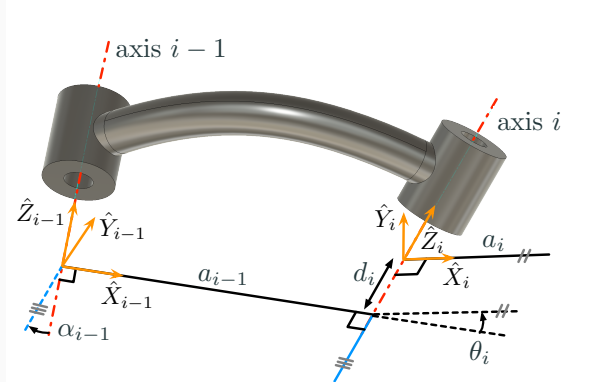
# Overview software architecture ROS2 Control

- Robot control is based on Moveit2+ros2\_control
- Ros2\_control nodes require configuration from .YAML files
  - Controller Manager
  - Resource Manager
  - Controllers (Joint State Broadcaster/Joint Trajectory Controller)
- Moveit planners require configuration
  - URDF
  - Moveit setup assistant to generate SRDF and .YAML files

# Robot modelling using Modified Denavit–Hartenberg parameters

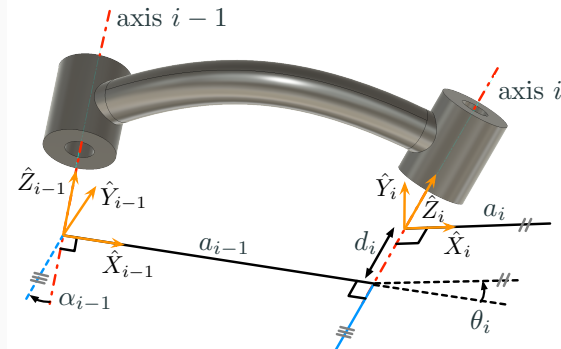
---

# Modified Denavit–Hartenberg parameter conventions



- $a$  (Link Length): The distance along the  $X_i$ -axis from the origin of frame  $i$  to the origin of frame  $i+1$ .
- $\alpha_i$  (Link Twist): The rotation about the  $X_i$ -axis from the  $Z_{i-1}$ -axis to the  $Z_i$ -axis.

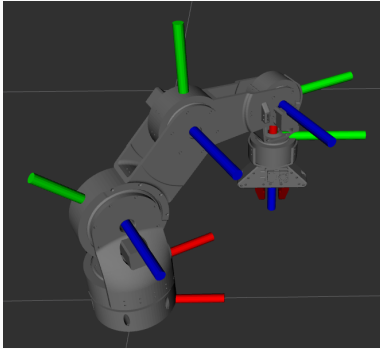
# Modified Denavit–Hartenberg parameter conventions



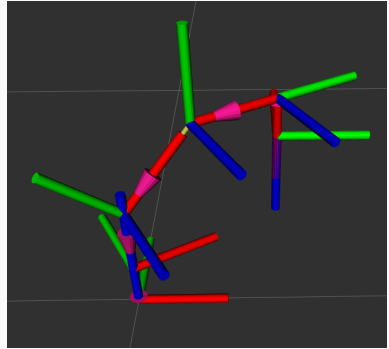
- $d_i$  (Joint Offset): The distance along the  $Z_{i-1}$ -axis from the origin of frame  $i-1$  to the origin of frame  $i$ .
- $\theta_i$  (Joint Angle): The rotation about the  $Z_{i-1}$ -axis from the  $X_{i-1}$ -axis to the  $X_i$ -axis.



# Skyentific robot in Rviz



**Figure 1:** Skyentific robot in Rviz



**Figure 2:** Axis definition

- Robot design <https://skyentific.com/>
- Five axis with 3D printed planetary gearboxes driven by standard stepper motors & dynamixel gripper.

# Modified Denavit–Hartenberg parameters of Skyentific robot

**Table 1:** Modified DH parameters robot

joint	$\alpha_{i-1}$	$a_{i-1}$	$d_i$	$\theta_i$	explanation
0	0	0	96.5	fixed	ground - base
1	0	0	150	motor_1	base-link_1
2	$\frac{1}{2}\pi$	0	0	motor_2	link_1-link_2
3	0	225	0	motor_3	link_2-link_3
4	0	200	0	motor_4	link_3-link_4
5	$-\frac{1}{2}\pi$	0	0	motor_5	link_4-link_5

- Gripper is omitted in this overview.

What is XAXRO/URDF and how  
to apply it to a Robot?

---

## XACRO/URDF description of robot link

- XACRO or XML Macro is a macro language based on XML used in combination with URDF files
- It is possible to define macro's, variables and properties
- Math and expressions can be used
- Key featue is the ability to reuse your code for different robot's

# XACRO/URDF description of robot link

```
<link name="base_link">
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <geometry>...</geometry>
    <material name="grey" />
  </visual>

  <collision>
    <origin xyz="0 0 0.075" rpy="0 0 0"/>
    <geometry>...</geometry>
  </collision>

  <inertial>
    <mass value="1.0" />
    <origin xyz="0 0 0.075" rpy="0 0 0"/>
    <inertia ixx="1.0" ... izz="1.0"/>
  </inertial>
</link>
```

## XACRO/URDF description of link geometry

```
<link name="base_link">
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <geometry>
      <mesh filename="file://$(find
        robot_description)/meshes/base_z_ob.stl"
        scale="0.001 0.001 0.001"/>
    </geometry>
    <material name="grey" />
  </visual>
  ...
</link>
```

- Visual is used for visualization only...
- Set origin and orientation according to origin of generated STL.
- Preferably adjust CAD files to avoid confusion (z up etc) .

## XACRO/URDF description of link geometry

```
<link name="base_link">
  ...
  <collision> <!-- cylinder equivalent -->
    <origin xyz="0 0 0.075" rpy="0 0 0"/>
    <geometry>
      <cylinder radius="0.1" length="0.15"/>
    </geometry>
  </collision>
  ...
</link>
```

- For path planning Moveit needs to know when a robot may collide with itself or with the environment.
- Use simple shapes to avoid computational burden (no STL's!).
- Derive shape dimensions from CAD file.

# XACRO/URDF description of link geometry

```
<link name="base_link">
  ...
  <inertial>
    <mass value="1.0" />
    <origin xyz="0 0 0.075" rpy="0 0 0"/>
    <inertia ixx="1.0" ixy="0.0" ixz="0.0" iyy="1.0"
      iyz="0.0" izz="1.0"/>
  </inertial>
  ...
</link>
```


- For simulations where torque and forces play a role we need to set inertia and mass.
- Since we will rely on stepper motors we use placeholder values.
- If needed estimates can be eobtained from CAD file.

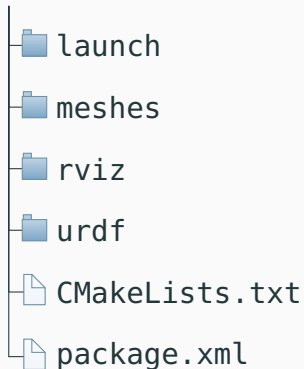


# Building the Robot Description package

---

## Package structure: robot\_description

 robot\_description



## Assignment 2: Create package

- Create directory structure:

```
cd ~  
mkdir -p minor_ws/src  
cd minor_ws/src  
ros2 pkg create robot_description  
cd robot_description  
rm -r include/ src/  
mkdir launch meshes rviz urdf  
cd ..  
code .
```

- Edit CMakeLists.txt

```
code .
```

## Assignment 3: edit CMakeLists.txt

```
cmake_minimum_required(VERSION 3.8)
project(robot_description)

find_package(ament_cmake REQUIRED)

# Install directories
install(
  DIRECTORY launch meshes urdf rviz
  DESTINATION share/${PROJECT_NAME}
)

ament_package()
```

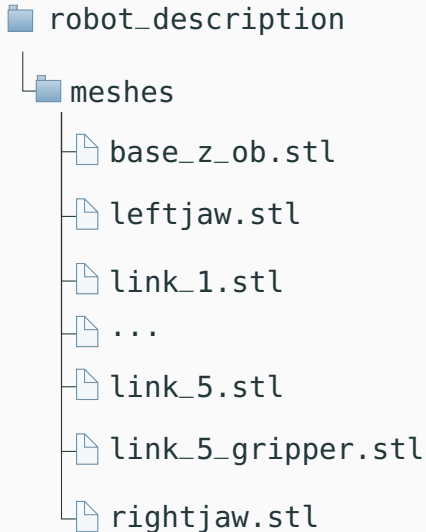
- Remove testing and add instruction to copy directories.
- Do this in package.xml as well.

## Assignment 3: edit package.xml

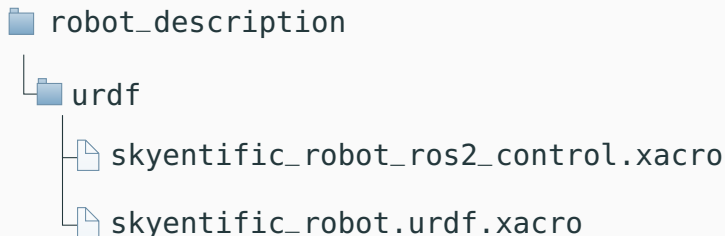
```
<?xml version="1.0"?>
<?xml-model href="http:..."?>
<package format="3">
  <name>robot_description</name>
  <version>0.0.0</version>
  <description>Skyentific Robot Description
    package</description>
  <maintainer email="aap@noot.nl">Mies</maintainer>
  <license>Apache 2.0</license>
  <buildtool_depend>ament_cmake</buildtool_depend>
  <export>
    <build_type>ament_cmake</build_type>
  </export>
</package>
```

- Test build process with colcon build...

## Assignment 4: Copy meshes from moodle



## Assignment 4: create files in urdf



- create files using vscode or terminal

```
cd ~/minor_ws/src/robot_description/urdf  
touch skyentific_robot_ros2_control.xacro  
touch skyentific_robot.urdf.xacro
```

## Assignment 5: Setup skyentific\_robot.urdf.xacro

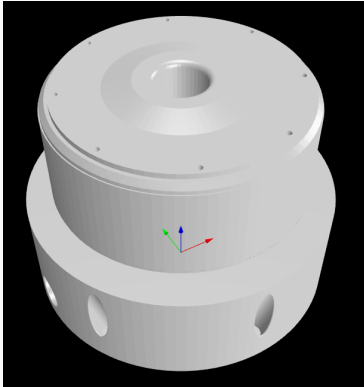
```
<?xml version="1.0"?>
<robot name="skyentific_robot"
  xmlns:xacro="http://www.ros.org/wiki/xacro">

  <!-- Include ros2_control Plugins -->
  <xacro:include filename="$(find
    robot_description)/urdf/
    skyentific_robot_ros2_control.xacro" />

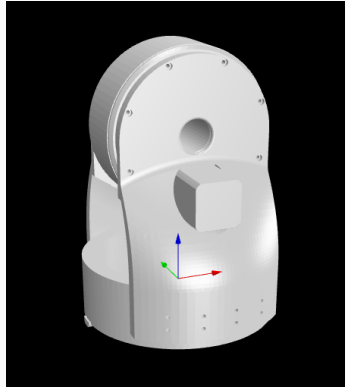
  <!-- Useful XACRO Variables (Properties) -->
  <xacro:property name="PI" value="3.14159265359" />
  <material name="grey">
    <color rgba="0.5 0.5 0.5 1.0"/></material>
  <material name="red">
    <color rgba="0.5 0.0 0.0 1.0"/></material>
</robot>
```



## Add Base and first link to URDF



**Figure 3:** The file `base_z_ob.stl`



**Figure 4:** The file `link_1.stl`

- The distance from the base to the first link is 96.5 mm.
- There are no rotations.
- Add *inside* `<robot name ...> </robot>`.

## Assignment 6: Add base\_link to URDF

```
<link name="base_link">
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <geometry>
      <mesh filename="file://$(find
        robot_description)/meshes/base_z_ob.stl"
        scale="0.001 0.001 0.001"/>
    </geometry>
    <material name="grey" />
  </visual>
  <collision> <!-- cylinder equivalent -->
    <origin xyz="0 0 0.075" rpy="0 0 0"/>
    <geometry>
      <cylinder radius="0.1" length="0.15"/>
    </geometry>
  </collision>
</link>
```

## Assignment 7: Add link\_1 to URDF

```
<link name="link_1">
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <geometry>
      <mesh filename="file://$(find
        robot_description)/meshes/link_1.stl"
        scale="0.001 0.001 0.001"/>
    </geometry>
    <material name="grey" />
  </visual>
  <collision>
    <origin xyz="0 0 0.150" rpy="1.57 0 0"/>
    <geometry>
      <cylinder radius="0.095" length="0.1"/>
    </geometry>
  </collision>
</link>
```

## Assignment 8: Add joint\_1 to URDF

```
<joint name="joint_1" type="revolute">
  <parent link="base_link"/>
  <child link="link_1"/>
  <origin xyz="0 0 0.0965" rpy="0 0 0"/>
  <axis xyz="0 0 1"/>
  <limit effort="1000.0" velocity="100.0"
    lower="-3.14" upper="3.14" />
</joint>
```

- The joint is actuated so we need to include a controller.
- We have no hardware yet so we use a mock component.
- Configure the command\_interface and state\_interface as position.
- set the initial\_value to 0.0.

## Assignment 9: Edit skyentific\_robot\_ros2\_control.xacro

```
<?xml version="1.0"?>
<robot xmlns:xacro="http://www.ros.org/wiki/xacro">
  <ros2_control name="Arm" type="system">
    <hardware>
      <plugin>mock_components/GenericSystem</plugin>
    </hardware>
    <joint name="joint_1">
      <command_interface name="position"/>
      <state_interface name="position">
        <param name="initial_value">0.0</param>
      </state_interface>
    </joint>
  </ros2_control>
</robot>
```

## Package structure: launch



- Launch files contain instructions to start each node in the correct sequence and contain all necessary parameters.
- Launch files are used in combination with

```
colcon build
source install/setup.bash
ros2 launch robot_description display.launch.xml
```

## Assignment 10: edit launch.xml

```
<launch>
  <let name="urdf_path"
    value="$(find-pkg-share robot_description)/urdf/
    skyentific_robot.urdf.xacro"/>
  <let name="rviz_config_path" value="$(find-pkg-share
    robot_description)/rviz/urdf_config.rviz"/>
  <node pkg="robot_state_publisher"
    exec="robot_state_publisher">
    <param name="robot_description"
      value="$(command 'xacro $(var urdf_path)')"/>
  </node>
  <node pkg="joint_state_publisher_gui"
    exec="joint_state_publisher_gui"/>
  <node pkg="rviz2" exec="rviz2" output="screen"
    args="-d $(var rviz_config_path)"/>
</launch>
```

## Assignment 11: Copy display.rviz from moodle

### robot\_description

#### launch

#### meshes

#### rviz

 display.rviz

#### urdf

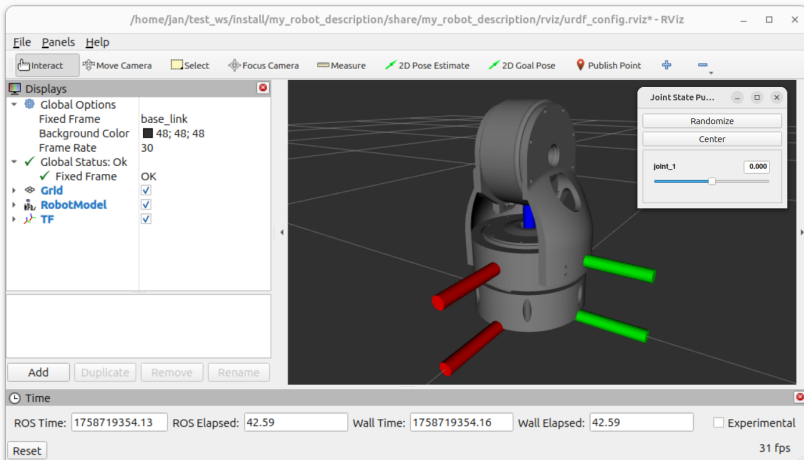
 CMakeLists.txt

 package.xml

- Rviz files contain configuration parameters of rviz.
- Store values from rviz in this file after first startup.
- When changing settings use Save Config (CTRL+S) to retain them.



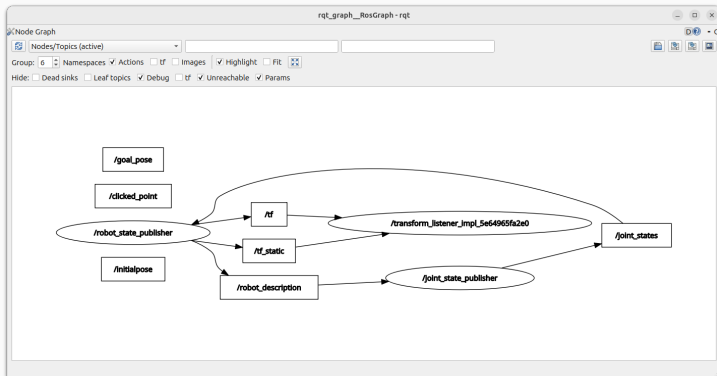
# Assignment 12: Test Functionality



- Test package in Rviz2

```
ros2 launch robot_description display.launch.xml
```

# Assignment13: Display package structure



- Open new terminal, source workspace and type:

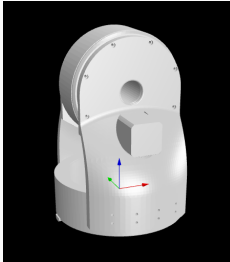
```
rqt_graph
```

## Assignment14: Test Package

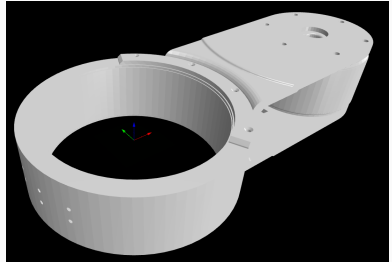
- Open new terminal, source workspace and type:

```
cd minor_ws
source install/setup.bash
ros2 node list
ros2 param list /robot_state_publisher
ros2 param get /robot_state_publisher
    robot_description
ros2 param get /robot_state_publisher
    publish_frequency
ros2 param list /joint_state_publisher
ros2 param get /joint_state_publisher
    publish_default_positions
```

## Add link\_2 and joint\_2 to URDF



**Figure 5:** The file link\_1.stl



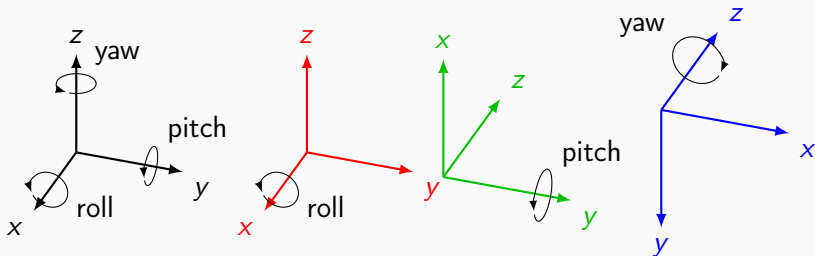
**Figure 6:** The file link\_2.stl

- Link\_1 and link\_2 are connected by joint\_2.
- We need to pay close attention to this joint since the z-axis is the axis of rotation.

## Assignment 15: Add link\_2 to URDF

```
<link name="link_2">
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <geometry>
      <mesh filename="file://$(find
        robot_description)/meshes/link_2.stl"
        scale="0.001 0.001 0.001"/>
    </geometry>
    <material name="grey"/>
  </visual>
  <collision>
    <origin xyz="0.225 0 0 " rpy="0 0 0"/>
    <geometry> <cylinder radius="0.07" length="0.1"/>
    </geometry>
  </collision>
</link>
```

## Translation of MDH parameters to REP-103 for joint\_2



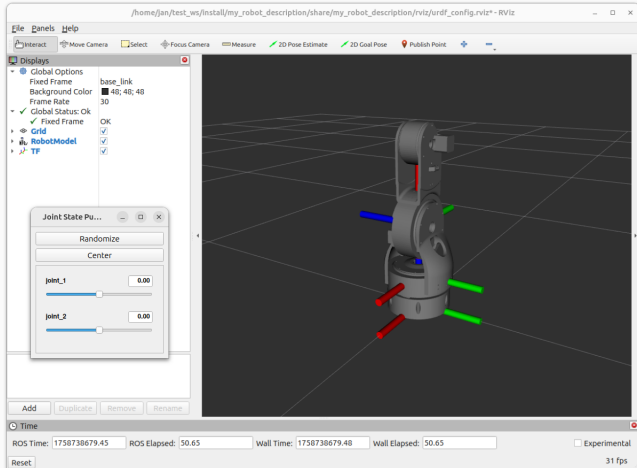
- Joint\_2 has a link twist  $\alpha_1 = \frac{1}{2}\pi$ , a link length and offset of 0 mm but is situated 150 mm above joint\_1
- We deviate slightly from the DH parameter convention here.
- The link twist is realized using the REP-103 Convention for Euler angles used by ROS which is a roll (rotation around X), followed by a pitch (around Y), and then a yaw (around Z).

## URDF: joint\_2

```
<joint name="joint_2" type="revolute">
  <parent link="link_1"/>
  <child link="link_2"/>
  <origin xyz="0 0 0.150" rpy="0 -1.57 1.57 "/>
  <axis xyz="0 0 1"/>
  <limit effort="1000.0" velocity="100.0"
    lower="-2.51" upper="2.51" />
</joint>
```

- We start with a displacement in  $z$  direction in the frame of the previous link (deviating from MDH convention).
- Then have a roll of 0, a pitch of  $-\frac{1}{2}\pi$  and a yaw of  $+\frac{1}{2}\pi$  by the right hand rule.
- The end result is that the  $z$ -axis is the angle of rotation by motor\_2 and the  $x$ -axis points along the  $x$ -axis of joint\_2.

# Assignment 16: Test Functionality

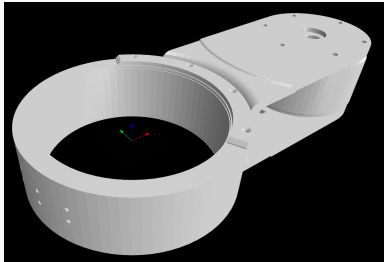


- Test package in Rviz2

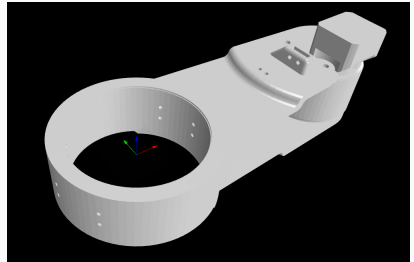
```
ros2 launch robot_description display.launch.xml
```



## Add link\_3 and joint\_2 to URDF



**Figure 7:** The file link\_2.stl



**Figure 8:** The file link\_3.stl

- Link\_2 and link\_3 are connected by joint\_3.
- This has a link length of 225 mm, a link twist  $\alpha_2$ , a joint angle  $\theta_3$  and a joint offset  $d_3$  of 0.

## Assignment 17: Add link\_3 to URDF

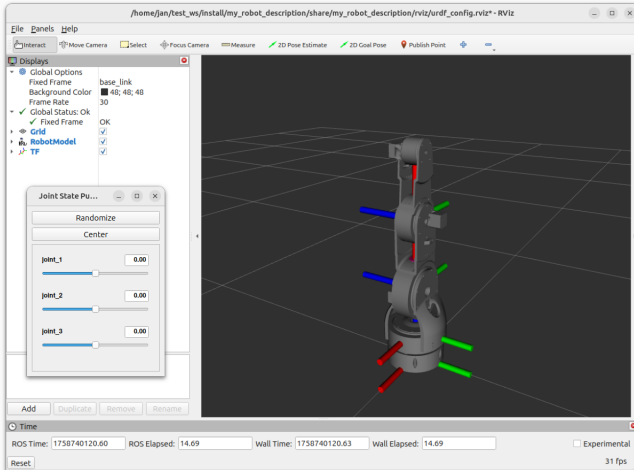
```
<link name="link_3">
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <geometry>
      <mesh filename="file://$(find
        robot_description)/meshes/link_3.stl"
        scale="0.001 0.001 0.001"/>
    </geometry>
    <material name="grey" />
  </visual>
  <collision>
    <origin xyz="0.2 0 0" rpy="0 0 0"/>
    <geometry> <cylinder radius="0.055" length="0.1"/>
    </geometry>
  </collision>
</link>
```

## Assignment 18: Add joint\_3 to URDF

```
<joint name="joint_3" type="revolute">  
  <parent link="link_2"/>  
  <child link="link_3"/>  
  <origin xyz="0.225 0 0" rpy="0 0 0"/>  
  <axis xyz="0 0 1"/>  
  <limit effort="1000.0" velocity="100.0"  
    lower="-2.51" upper="2.51" />  
</joint>
```

- The link length of 225 mm is realized by a displacement along the x-axis of joint\_2.

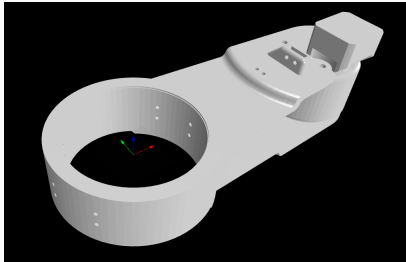
# Assignment 19: Test Functionality



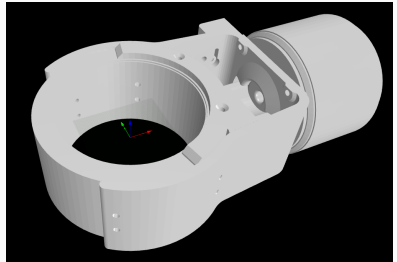
- Test package in Rviz2

```
ros2 launch robot_description display.launch.xml
```

## Add link\_4 and joint\_4 to URDF



**Figure 9:** The file link\_3.stl



**Figure 10:** The file link\_4.stl

- Link\_3 and link\_4 are connected by joint\_3.
- This has a link length of 200 mm, a link twist  $\alpha_3$ , a joint angle  $\theta_4$  and a joint offset  $d_4$  of 0.

## Assignment 20: Add link\_4 to URDF

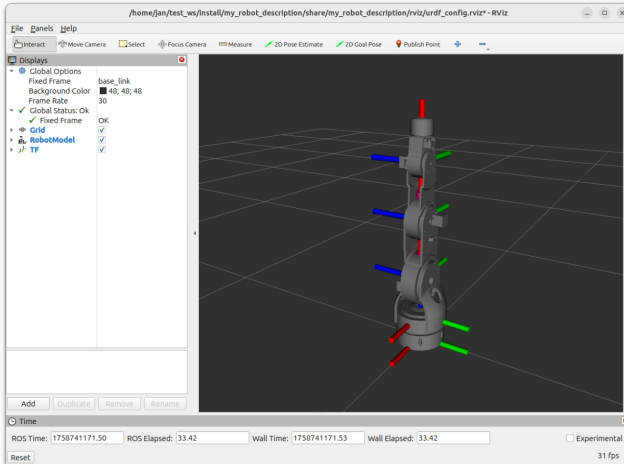
```
<link name="link_4">
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <geometry>
      <mesh filename="file://$(find
        robot_description)/meshes/link_4.stl"
        scale="0.001 0.001 0.001"/>
    </geometry>
    <material name="grey" />
  </visual>
  <collision>
    <origin xyz="0.1 0 0" rpy="0 1.57 0"/>
    <geometry> <cylinder radius="0.055"
      length="0.06"/></geometry>
  </collision>
</link>
```

## Assignment 21: Add joint\_4 to URDF

```
<joint name="joint_4" type="revolute">
  <parent link="link_3"/>
  <child link="link_4"/>
  <origin xyz="0.2 0 0" rpy="0 0 0"/>
  <axis xyz="0 0 1"/>
  <limit effort="1000.0" velocity="100.0"
    lower="-2.51" upper="2.51" />
</joint>
```

- This joint is similar to joint\_3.

# Assignment 22: Test Functionality

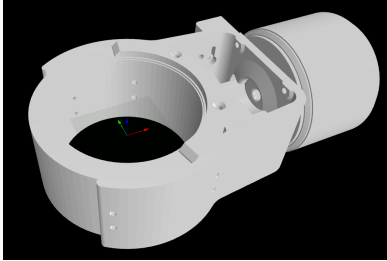


- Test package in Rviz2

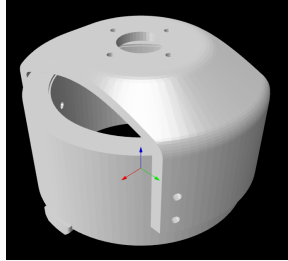
```
ros2 launch robot_description display.launch.xml
```



## Add link\_4 and joint\_5 to URDF



**Figure 11:** The file link\_4.stl



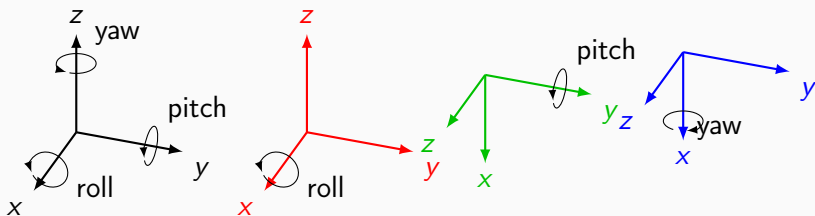
**Figure 12:** The file link\_5.stl

- item

## Assignment 23: Add link\_5 to URDF

```
<link name="link_5">
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <geometry>
      <mesh filename="file://$(find
        robot_description)/meshes/link_5.stl"
        scale="0.001 0.001 0.001"/>
    </geometry>
    <material name="grey" />
  </visual>
  <collision>
    <origin xyz="0 0 0" rpy="1.57 0 0"/>
    <geometry> <cylinder radius="0.055"
      length="0.01"/></geometry>
  </collision>
</link>
```

## Translation of MDH parameters to REP-103 for joint\_5



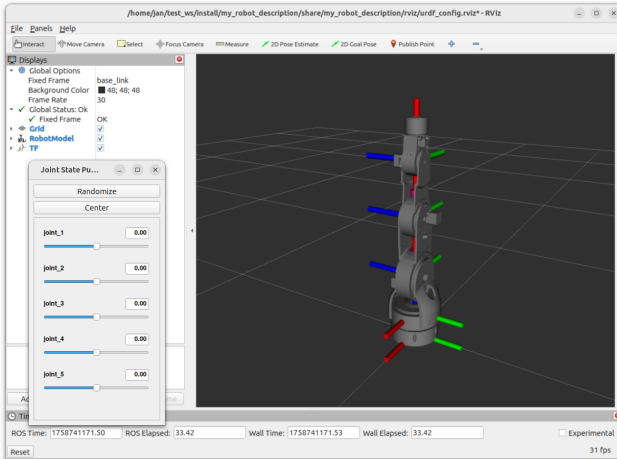
- Joint\_5 has a link twist of  $\alpha_4 = -\frac{1}{2}\pi$ , a link length of 0 and a link offset  $\theta_5$  of 9.15 mm.
- The link twist is realized by a pitch angle of  $\frac{1}{2}\pi$

## Assignment 24: Add joint\_5 to URDF

```
<joint name="joint_5" type="revolute">
  <parent link="link_4"/>
  <child link="link_5"/>
  <origin xyz="0.0915 0 0" rpy="0 1.57 0"/>
  <axis xyz="0 0 1"/>
  <limit effort="1000.0" velocity="100.0"
    lower="-2.51" upper="2.51" />
</joint>
```

- Note that the interface is located at the bottom of joint\_5 rather than at the center of the joint.
- The interface can be changed as needed by adjusting the origin of the visual in z-direction and the joint in x-direction.
- Use XACRO tags like this `<origin xyz="$0.0915+0.075 0 0" rpy="0 1.57 0"/>`.

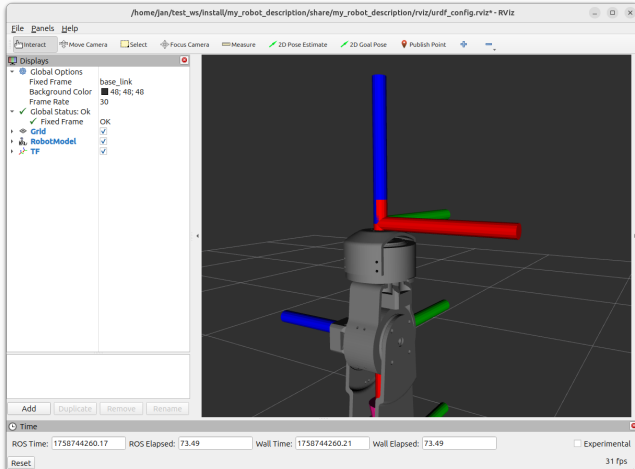
# Assignment 25: Test Functionality



- Test package in Rviz2

```
ros2 launch robot_description display.launch.xml
```

# Assignment 26: Test Functionality



- Test package in Rviz2

```
ros2 launch robot_description display.launch.xml
```

# Modelling the gripper

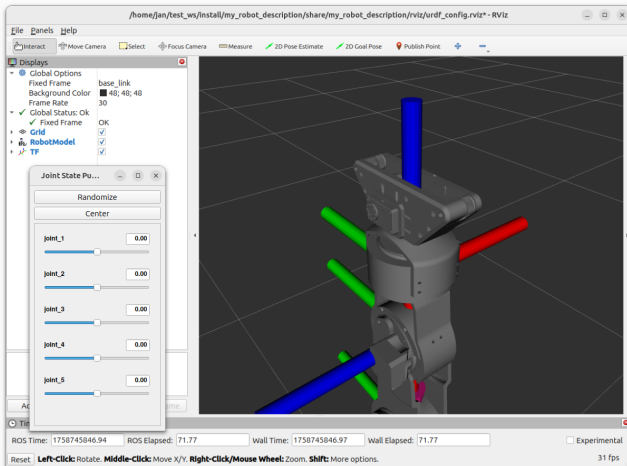
---

## Assignment 27: Edit URDF of link\_5

```
<link name="link_5">
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <geometry>
      <mesh filename="file://$(find
        robot_description)/meshes/link_5_gripper.stl"
        scale="0.001 0.001 0.001"/>
    </geometry>
    <material name="grey" />
  </visual>
  <collision>
    <origin xyz="0 0 0" rpy="1.57 0 0"/>
    <geometry> <cylinder radius="0.055"
      length="0.01"/></geometry>
  </collision>
</link>
```



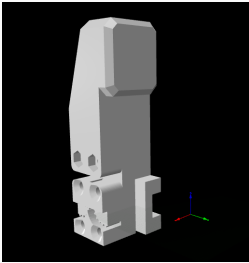
# Assignment 28: Test Functionality



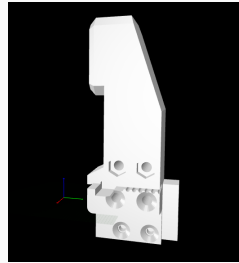
- Test package in Rviz2

```
ros2 launch robot_description display.launch.xml
```

## Add link\_4 and joint\_5 to URDF



**Figure 13:** The file leftjaw.stl



**Figure 14:** The file rightjaw.stl

- Add left and right jaw as link\_6 and link\_7.
- Add prismatic joints joint\_6 and joint\_7.

## Assignment 28: Add link\_6 to URDF

```
<link name="gripper_right">
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <geometry>
      <mesh filename="file://$(find
        robot_description)/meshes/rightjaw.stl"
        scale="0.001 0.001 0.001"/>
    </geometry>
    <material name="red"/>
  </visual>
  <collision>
    <origin xyz="0.01 0.025 0.045" rpy="0 0 0"/>
    <geometry> <cylinder radius="0.0125"
      length="0.04"/></geometry>
  </collision>
</link>
```

## Assignment 29: Add joint\_6 to URDF

```
<joint name="joint_6" type="prismatic">
  <parent link="link_5"/>
  <child link="gripper_right"/>
  <origin xyz="-0.01 0 0.1015" rpy="0 0 0"/>
  <axis xyz="0 1 0"/>
  <limit effort="1000.0" velocity="100.0"
    lower="-0.018" upper="0.02" />
</joint>
```

- item

## Assignment 30: Add link\_7 to URDF

```
<link name="gripper_left">
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <geometry>
      <mesh filename="file://$(find
        robot_description)/meshes/leftjaw.stl"
        scale="0.001 0.001 0.001"/>
    </geometry>
    <material name="red"/>
  </visual>
  <collision>
    <origin xyz="0.01 -0.025 0.045" rpy="0 0 0"/>
    <geometry> <cylinder radius="0.0125"
      length="0.04"/></geometry>
  </collision>
</link>
```

## Assignment 31: Add joint\_7 to URDF

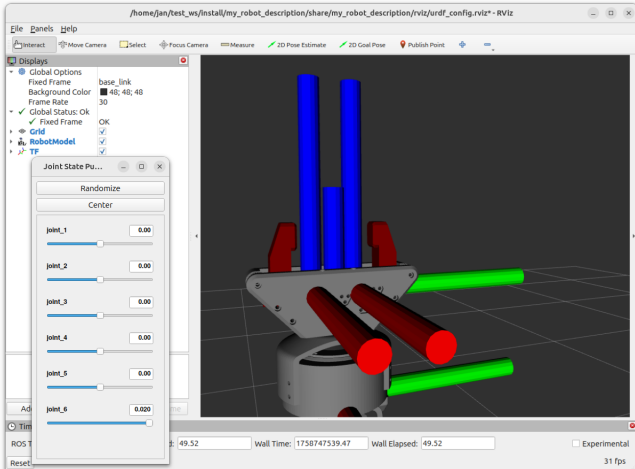
```
<joint name="joint_7" type="prismatic">
  <parent link="link_5"/>
  <child link="gripper_left"/>
  <origin xyz="-0.01 0 0.1015" rpy="0 0 0"/>
  <axis xyz="0 1 0"/>
  <mimic joint="joint_6" multiplier="-1"/>
  <limit effort="1000.0" velocity="100.0"
    lower="-0.018" upper="0.02" />
</joint>
```

- We are using the mimic functionality to copy the movement of joint\_6.

## Assignment 32: Edit skyentific\_robot\_ros2\_control.xacro

```
<?xml version="1.0"?>
<robot xmlns:xacro="http://www.ros.org/wiki/xacro">
  <ros2_control name="Arm" type="system">
    ...
  </ros2_control>
  <ros2_control name="Gripper" type="system">
    <hardware>
      <plugin>mock_components/GenericSystem</plugin>
    </hardware>
    <joint name="joint_6">
      <command_interface name="position"/>
      <state_interface name="position">
        <param name="initial_value">0.0</param>
      </state_interface>
    </joint>
  </ros2_control>
</robot>
```

# Assignment 33: Test Functionality



- Test package in Rviz2

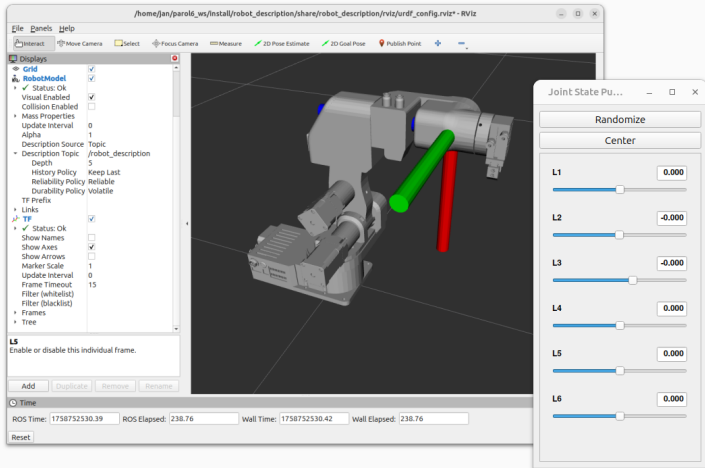
```
ros2 launch robot_description display.launch.xml
```



# Project assignment

---

# Apply your knowledge to the Parol6 robot



- <https://github.com/PCrnnjak/PAROL6-Desktop-robot-arm.git>

**Next week: Movit2 planners**

---