ENGR 4421:Robotics II

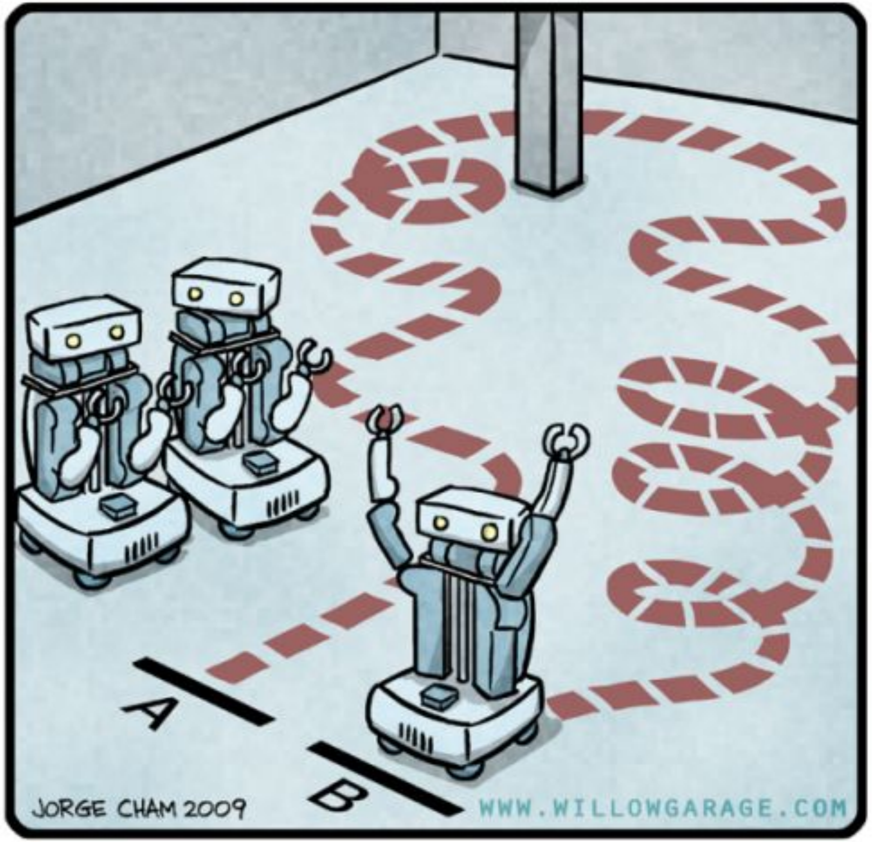
ROS Tutorial: URDF (Unified Robot Description Format)

03/02/2023

Outline

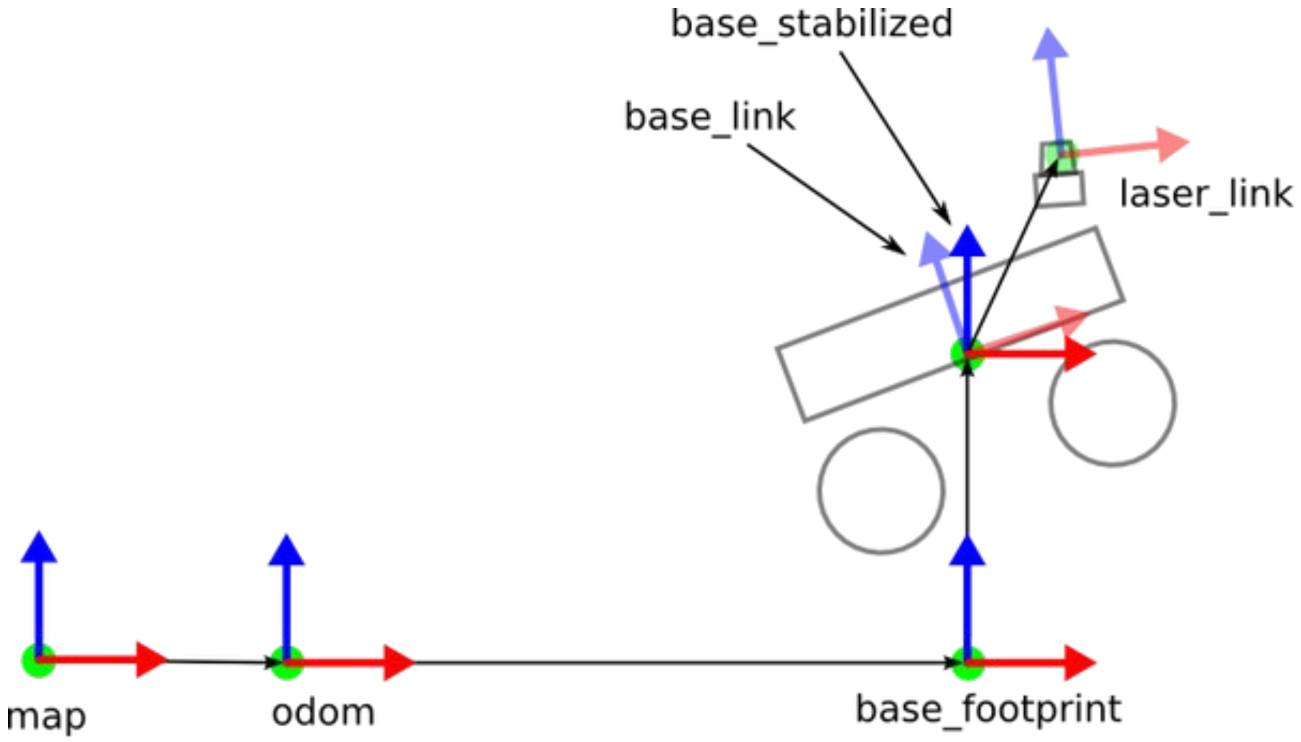
- Concepts
- Link
- Joint
- Example

R.O.B.O.T. Comics



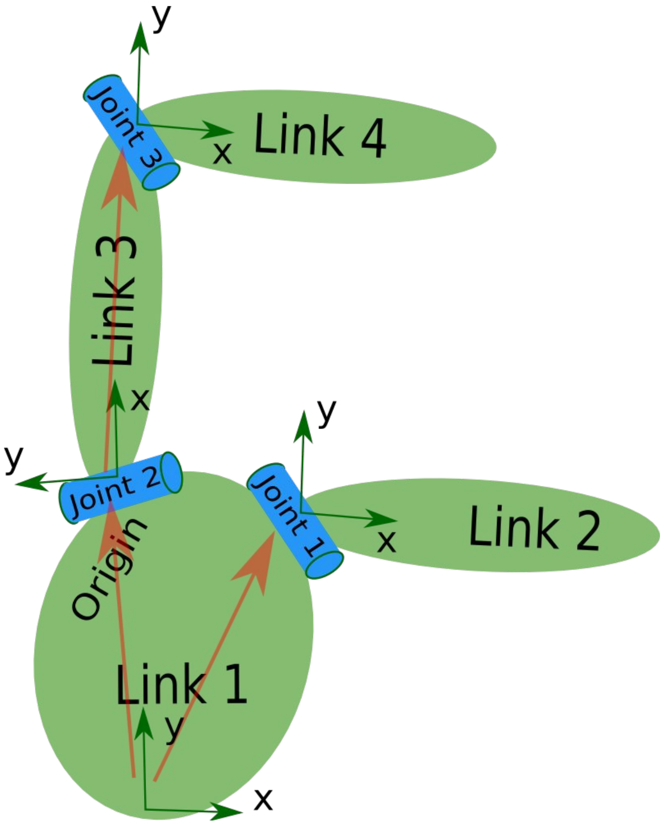
"HIS PATH-PLANNING MAY BE SUB-OPTIMAL, BUT IT'S GOT FLAIR."

Transforms



We can use reference frames to represent spatial relationships between robotic components. In a navigation task, such relationships will be used to localize the robot in a map.

URDF Concepts



- The Unified Robot Description Format (URDF) is an XML specification to describe a robot, which covers
  - Kinematic and dynamic description of the robot
  - Visual representation of the robot
  - Collision model of the robot

- The description of a robot consists of a set of link elements, and a set of joint elements connecting the links together.

- Limitations: cannot describe parallel robots; cannot deal with flexible/transformable links (rigid body only).

- File structure:

```
<robot name="robot_name">
    <link> ... </link>
    <link> ... </link>
    <link> ... </link>

    <joint>  ....  </joint>
    <joint>  ....  </joint>
    <joint>  ....  </joint>
</robot>
```

URDF Helpful Resources

- Tutorial page: https://docs.ros.org/en/humble/Tutorials/Intermediate/URDF/URDF-Main.html
- Tutorial Package: https://github.com/ros/urdf_tutorial/tree/ros2
- Concepts explain/API: http://wiki.ros.org/urdf/XML
- Example Repository: https://github.com/linzhangUCA/homeplater

Build A Robot Description Package from Scratch

- Create workspace:

  ```
  mkdir -p ~/<workspace_name>/src
  ```

- Create a package:

  ```
  cd ~/<workspace_name>/src
  ros2 pkg create --build-type ament_python <package_name>
  ```

- Create data directories:

  ```
  cd <package_name>
  mkdir launch/ urdf/ rviz/
  ```

- Edit package.xml:

  ```
  code package.xml  # make sure vscode is available
  ```

- Edit setup.py:

  ```
  code setup.py
  ```

- Edit launch file:

  ```
  code launch/<launch_filename>.launch.py
  ```

- Edit URDF file:

  ```
  code urdf/<urdf_filename>.urdf
  ```

package.xml

```xml
<?xml version="1.0"?>
<?xml-model href="http://download.ros.org/schema/package_format3.xsd" schematypens="http://www.w3.org/2001/XMLSchema"?>
<package format="3">
  <name>package_name</name>
  <version>0.0.0</version>
  <description>TODO</description>
  <maintainer email="todo@todo">TODO</maintainer>
  <license>TODO</license>

  <exec_depend>joint_state_publisher</exec_depend>
  <exec_depend>joint_state_publisher_gui</exec_depend>
  <exec_depend>robot_state_publisher</exec_depend>
  <exec_depend>rviz2</exec_depend>
  <exec_depend>xacro</exec_depend>

  <test_depend>ament_copyright</test_depend>
  <test_depend>ament_flake8</test_depend>
  <test_depend>ament_pep257</test_depend>
  <test_depend>python3-pytest</test_depend>

  <export>
    <build_type>ament_python</build_type>
  </export>
</package>
```

CHANGE package_name (line 4) TO ACTUAL PACKAGE NAME

setup.py

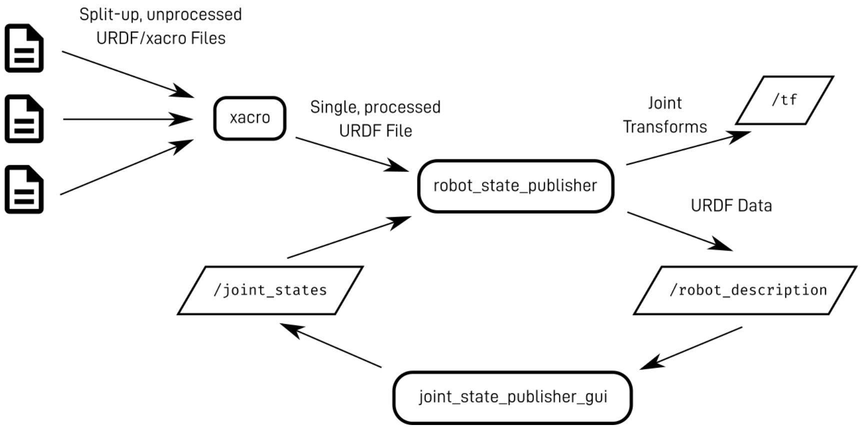
```python
import os
from glob import glob
from setuptools import setup

package_name = '<package_name>'  # CHANGE <package_name> TO ACTUAL PACKAGE NAME

setup(
    name=package_name,
    version='0.0.0',
    packages=[package_name],
    data_files=[
        ('share/ament_index/resource_index/packages',
            ['resource/' + package_name]),
        ('share/' + package_name, ['package.xml']),
        (os.path.join('share', package_name, 'launch'), glob(os.path.join('launch', '*'))),
        (os.path.join('share', package_name, 'urdf'), glob(os.path.join('urdf', '*'))),
        (os.path.join('share', package_name, 'rviz'), glob(os.path.join('rviz', '*'))),
    ],
    install_requires=['setuptools'],
    zip_safe=True,
    maintainer='TODO',
    maintainer_email='todo@todo',
    description='TODO',
    license='TODO',
    tests_require=['pytest'],
    entry_points={
        'console_scripts': [
        ],
    },
)
```

<launch_filename>.launch.py

- Copy the contents in:
  https://raw.githubusercontent.com/linzhangUCA/homeplater/visual/hpr_description/launch/view_homeplater.launch.py to launch/<launch_filename>.launch.py
- Change line 13, 14 according to your own configurations:

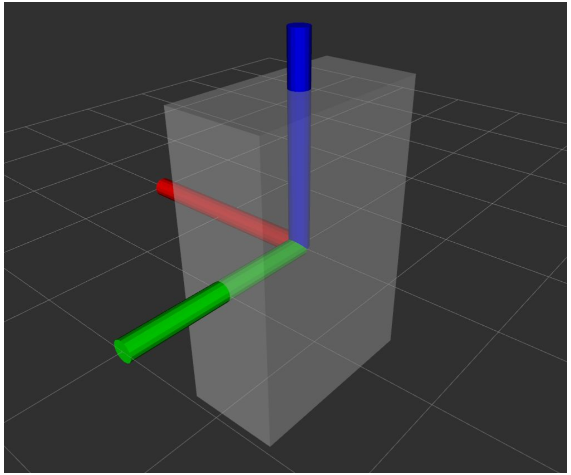


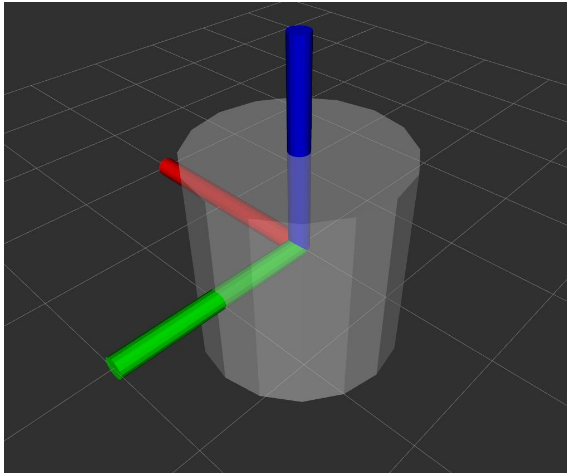
Split-up, unprocessed URDF/xacro Files
xacro
Single, processed URDF File
robot_state_publisher
Joint Transforms
/tf
URDF Data
/robot_description
/joint_states
joint_state_publisher_gui

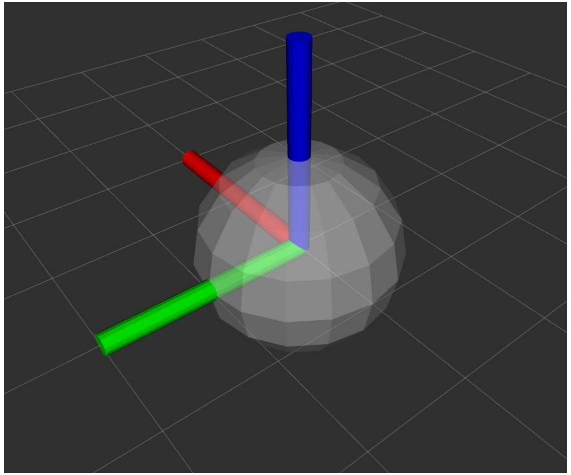
Install Dependencies with rosdep

```
sudo rosdep init
rosdep update
cd ~/<workspace_name>/
rosdep install --from-paths src -y --ignore-src
```

URDF: Link







```
<box size="1 2 3"/>
```

```
<cylinder radius="1" length="2"/>
```

```
<sphere radius="1"/>
```

URDF: Link

```xml
<?xml version="1.0"?>
<robot name="robot_name">
    <link name="base_link">
        <visual>
            <geometry>
                <box size="0.16 0.16 0.005"/>
            </geometry>
        </visual>
    </link>
</robot>
```

Launch joint_state_publisher, robot_state_publisher, rviz

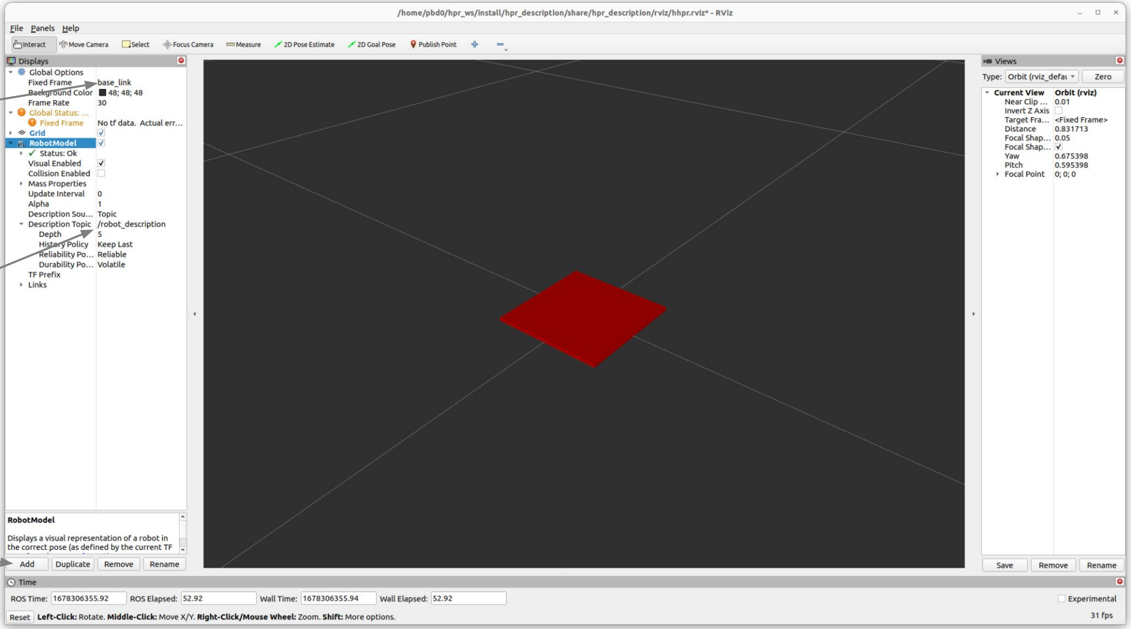
```
cd ~/<workspace_name>/
colcon build
source install/local_setup.bash
ros2 launch <package_name> <launch_filename>.launch.py  # you can execute this anywhere
```

You'll want to perform this step every time you add new stuff in the urdf file.

3.Type base_link in Fixed Frame

2.Select /robot_description in Description Topic

1.Add RobotModel



/home/pbd0/hpr_ws/install/hpr_description/share/hpr_description/rviz/hhpr.rviz* - RViz

Displays

Global Options
Fixed Frame: base_link
Background Color: 48; 48; 48
Frame Rate: 30
Global Status: ...
Fixed Frame: No tf data. Actual err...
Grid ☑
RobotModel ☑
✓ Status: Ok
Visual Enabled ☑
Collision Enabled ☐
Mass Properties
Update Interval: 0
Alpha: 1
Description Sou...: Topic
Description Topic: /robot_description
Depth: 5
History Policy: Keep Last
Reliability Po...: Reliable
Durability Po...: Volatile
TF Prefix
Links

RobotModel

Displays a visual representation of a robot in the correct pose (as defined by the current TF

Views

Type: Orbit (rviz_defa

Current View: Orbit (rviz)
Near Clip ...: 0.01
Invert Z Axis ☐
Target Fra...: <Fixed Frame>
Distance: 0.831713
Focal Shap...: 0.05
Focal Shap... ☑
Yaw: 0.675398
Pitch: 0.595398
Focal Point: 0; 0; 0

Time

ROS Time: 1678306355.92 ROS Elapsed: 52.92 Wall Time: 1678306355.94 Wall Elapsed: 52.92 ☐ Experimental

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:** Zoom. **Shift:** More options.

31 fps

URDF: Joint

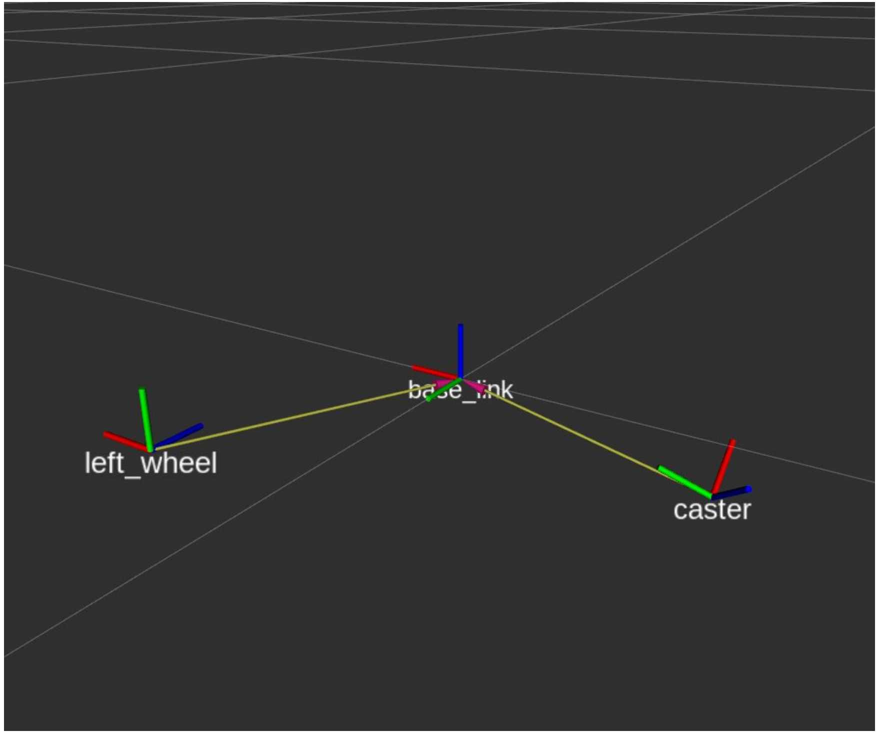
```xml
<?xml version="1.0"?>
<robot name="robot_name">
    ...

    <joint name="caster joint" type="fixed">
        <parent link="base_link"/>
        <child link="caster"/>
        <origin xyz="-0.092 0 -0.0185" rpy="0.707 -0.707 -1.571"/>
    </joint>

    <link name="caster">
    </link>

    <joint name="left_wheel_joint" type="continuous">
        <parent link="base_link"/>
        <child link="left_wheel"/>
        <origin xyz="0.048 0.095 0" rpy="1.571 0 0"/>
        <axis xyz="0 0 1" />
    </joint>

    <link name="left_wheel">
    </link>
</robot>
```



URDF: Color

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

    <material name="purple">
        <color rgba="0.3098 0.1765 0.4980 1" />
    </material>

    <link name="base_link">
        <visual>
            <geometry>
                <box size="0.16 0.16 0.005"/>
            </geometry>
            <material name="purple"/>
        </visual>
    </link>

</robot>
```

Save rviz config

1. Check the <rviz_filename> at line 15, in launch/<launch_filename>.launch.py
2. When everything is ready (`RobotModel` and `TF` added, `Fixed Frame` set to `/base_link`, the location and color of all the links are correct). Click "File -> Save Config As". Then, save the config file to ~/<workspace_name>/src/<package_name>/rviz/<rviz_filename>.rviz
3. Next time, `ros2 launch <package_name> <launch_filename>.launch.py` should bring up rviz with the saved configuration.

xacro:property

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

    <xacro:property name="caster_xoffset" value="-0.044" />
    <xacro:property name="caster_radius" value="0.014" />
    <xacro:property name="wheel_zoffset" value="-0.013" />
    <xacro:property name="wheel_radius" value="0.0325" />

    ...

    <joint name="caster_joint" type="fixed">
        <parent link="base_link" />
        <child link="caster" />
        <origin xyz="${caster_xoffset} 0.0 ${caster_radius - wheel_radius + wheel_zoffset}" />
    </joint>

    <!-- CASTER WHEEL -->
    <link name="caster">
        <visual>
            <geometry>
                <sphere radius="${caster_radius}" />
            </geometry>
            <material name="gray" />
        </visual>
    </link>

    ...

</robot>
```

xacro:macro

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

    <xacro:macro name="wheel" params="prefix y_reflect">
        <joint name="${prefix}_joint" type="continuous">
            <parent link="base_link" />
            <child link="${prefix}_link" />
            <origin xyz="${wheel_xoffset} ${y_reflect*wheel_yoffset} ${wheel_zoffset}" rpy="${-pi/2} 0 0" />
            <axis xyz="0 0 1" />
        </joint>

        <link name="${prefix}_link">
            <visual>
                <geometry>
                    <cylinder radius="${wheel_radius}" length="${wheel_width}" />
                </geometry>
                <material name="black" />
            </visual>
        </link>
    </xacro:macro>

    ...

    <xacro:wheel prefix="left_wheel" y_reflect="1" />
    <xacro:wheel prefix="right_wheel" y_reflect="-1" />

</robot>
```

Full Examples

- Full URDF example:
  https://github.com/linzhangUCA/homeplater/blob/1-visual/hpr_description/urdf/homeplater.urdf
- Full XACRO example:
  https://github.com/linzhangUCA/homeplater/blob/2-xacro/hpr_description/urdf/homeplater.urdf.xacro