

# ENGR 4421: Robotics II

URDF

02/08/2022

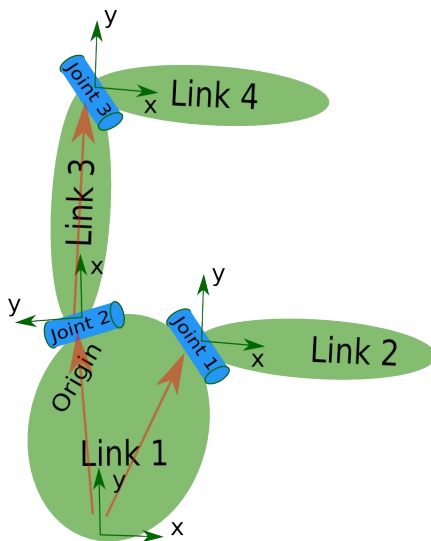


# Outline

- Introduction
- ROS Tutorials
  - Link
  - rviz
  - Joint
  - Properties
  - xacro

# What is URDF

- Unified Robot Description Format (URDF) specifies the geometry and organization of robots in ROS.
- URDF sets up transformations, extremely useful when robots get more complicated.
- URDF is useful for visualizing (via rviz), too.
- The robot model can be set up by describing its links and joints in XML language.



# **URDF Get Started**

<https://docs.ros.org/en/galactic/Tutorials/URDF/URDF-Main.html>

# Preparation

# Note: you can choose your own names in "< >". Remember to get rid of the brackets: "<>"

```
cd ~/<ros workspace>/src # go to `src/` in your ros workspace
ros2 pkg create --build-type ament_python <urdf package dir> # create a package
cd <urdf package dir> # go to your package
mkdir launch rviz urdf # create useful directories
code . # bring up VSCode and open the <urdf_package> directory \
```

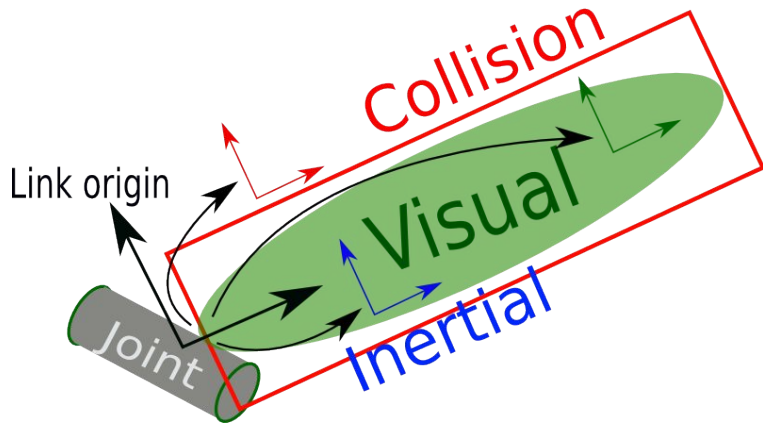
# Now, you can create a new file, e.g. `urdf/robot.urdf`, to start modeling.

# Create a Link

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

  <link name="base_link">
    <visual>
      <geometry>
        <box size="0.4 0.2 0.1" />
      </geometry>
    </visual>
  </link>

</robot>
```



# Visualize in RVIZ

- Download rviz configuration file

```
cd <urdf package dir>/rviz  
wget https://raw.githubusercontent.com/ros/urdf_tutorial/ros2/rviz/urdf.rviz
```

- Create launch file

```
cd <urdf package dir>/launch  
wget https://raw.githubusercontent.com/ros/urdf_tutorial/ros2/launch/display.launch.py  
# make sure line 12 ~ line 14 match the contents in your package.
```

# Visualize in RVIZ

- Update package dependencies in ``<urdf package dir>/package.xml``

```
<package format="3">
  ...
  <license>TODO: License declaration</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>
  <exec_depend>launch</exec_depend>
  <exec_depend>launch_ros</exec_depend>

  <test_depend>ament_copyright</test_depend>
  ...
</package>
```



# Visualize in RVIZ

- Add data files in ``<urdf package dir>/setup.py``

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

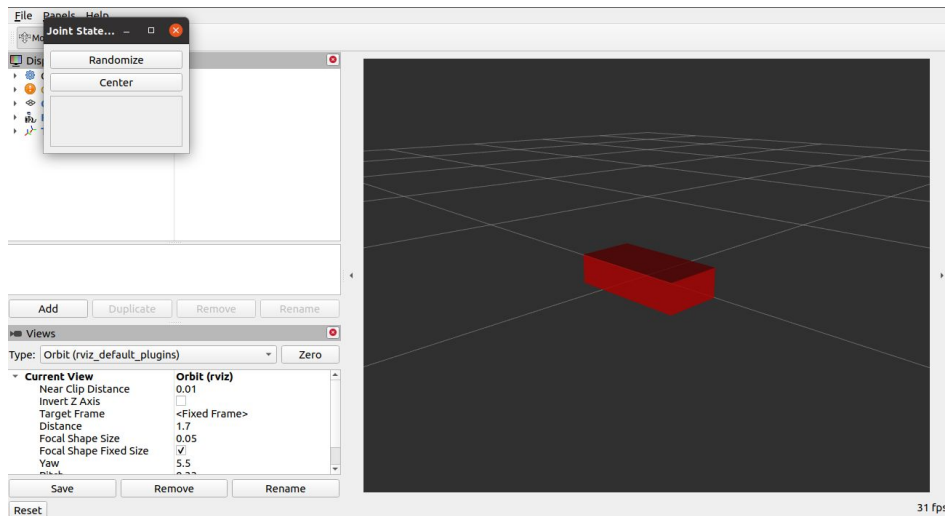
package_name = '<urdf package dir>'

setup(
    ...
    data_files=[
        ...
        ('share/' + package_name, ['package.xml']),
        (os.path.join('share', package_name, 'launch'), glob(os.path.join('launch', '*.launch.py'))),
        (os.path.join('share', package_name, 'urdf'), glob(os.path.join('urdf', '*.urdf'))),
        (os.path.join('share', package_name, 'rviz'), glob(os.path.join('rviz', '*.rviz'))),
    ],
    ...
)
```

# Visualize in RVIZ

- Build package and launch.

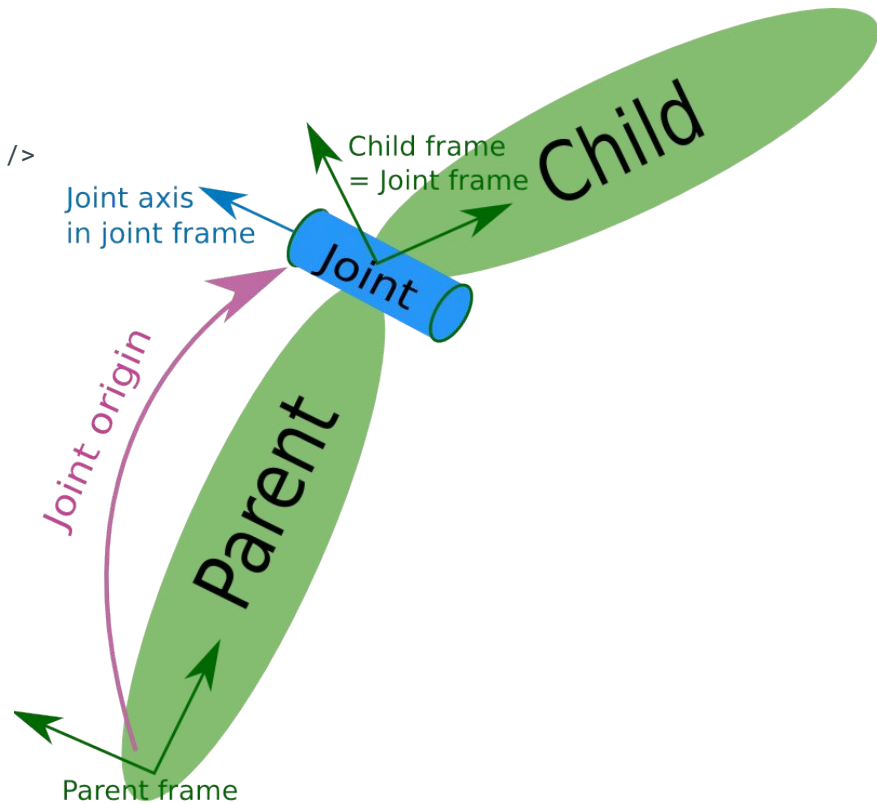
```
cd ~/<ros workspace> # e.g. cd ~/tutorial_ws
rosdep install -i --from-path src --rosdistro galactic -y
colcon build
source install/setup.bash
ros2 launch <urdf package dir> view_bot.launch.py
```



# Create a Joint

```
<joint name="base_to_left_wheel" type="continuous">  
  <parent link="base_link" />  
  <child link="left_wheel" />  
  <origin rpy="1.5707963267948966 0 0" xyz="0.1 0.13 0" />  
  <axis rpy="0 0 0" xyz="0 0 1" />  
</joint>
```

```
<link name="left_wheel">  
  <visual>  
    <geometry>  
      <cylinder length="0.05" radius="0.1" />  
    </geometry>  
    <material name="black" />  
  </visual>  
</link>
```



# Color

```
<robot name="diffbot">  
  <!-- define colors -->  
  <material name="blue">  
    <color rgba="0 0 0.8 1" />  
  </material>  
  <material name="black">  
    <color rgba="0 0 0 1" />  
  </material>  
  <material name="white">  
    <color rgba="1 1 1 1" />  
  </material>  
  <material name="purple">  
    <color rgba="0.3098 0.1765 0.4980 1" />  
  </material>  
  ...  
</robot>
```

# Collision

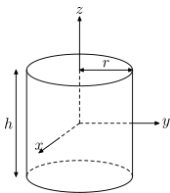
```
<link name="base_link">  
  <visual>  
    ...  
  </visual>  
  <collision>  
    <geometry>  
      <box size="0.4 0.2 0.1" />  
    </geometry>  
  </collision>  
</link>
```

# Inertia

```
<link name="base_link">
  <visual>
    ...
  </visual>
  <collision>
    ...
  </collision>
  <inertial>
    <mass value="1" />
    <inertia ixx="0.0042" ixy="0.0" ixz="0.0" iyy="0.0142" iyz="0.0" izz="0.0167" />
  </inertial>
</link>
```

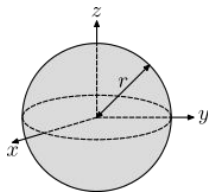
# Moment of Inertia

- Measures the extent to which an object resists rotational acceleration about a particular axis, and is the rotational analogue to mass

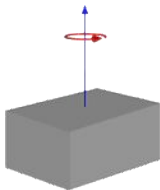


$$I_z = \frac{1}{2}mr^2$$

$$I_x = I_y = \frac{1}{12}m(3r^2 + h^2)$$



$$I = \frac{2}{5}mr^2$$



$$I_h = \frac{1}{12}m(w^2 + d^2)$$

$$I_w = \frac{1}{12}m(d^2 + h^2)$$

$$I_d = \frac{1}{12}m(h^2 + w^2)$$

# XACRO

To clean URDF

- Constants.
- Math.
- Macros.



# Xacro: Constants

```
<?xml version="1.0"?>
<robot xmlns:xacro="http://www.ros.org/wiki/xacro" name="bot">
  <xacro:property name="base_width" value="0.4" />
  <xacro:property name="base_depth" value="0.2" />
  <xacro:property name="base_height" value="0.1" />
  <link name="base_link">
    <visual>
      <geometry>
        <box size="${base_width} ${base_depth} ${base_height}" />
      </geometry>
      <material name="white" />
    </visual>
    <collision>
      <geometry>
        <box size="${base_width} ${base_depth} ${base_height}" />
      </geometry>
    </collision>
  </link>
```

# XACRO: Math

```
<xacro:property name="PI" value="3.141592653589793" />
<xacro:property name="x_offset" value="0.1" />
<xacro:property name="wheel_width" value="0.05" />
<xacro:property name="wheel_gap" value="0.005" />
...
<joint name="base_to_left_wheel" type="continuous">
  <parent link="base_link" />
  <child link="left_wheel" />
  <origin rpy="{PI/2} 0 0" xyz="{x_offset} {base_depth/2+wheel_gap+wheel_width/2} 0" />
  <axis rpy="0 0 0" xyz="0 0 1" />
</joint>
```

# XACRO: Macro

```
<xacro:macro name="box_inertial" params="mass x y z">
  <inertial>
    <mass value="\${mass}" />
    <inertia ixx="\${1/12*mass*(y*y+z*z)}" ixy="0.0" ixz="0.0"
      iyy="\${1/12*mass*(x*x+z*z)}" iyz="0.0"
      izz="\${1/12*mass*(y*y+x*x)}" />
  </inertial>
</xacro:macro>

...

<link name="base_link">
  <visual>
  </visual>
  <collision>
  </collision>
  <xacro:box_inertial mass="1" x="\${base_width}" y="\${base_depth}" z="\${base_height}" />
</link>
```