Chapter 4
URDF

Wende Ke
Department of Mechanical and Energy Engineering
Southern University of Science and Technology

Building a Visual Robot Model with URDF from Scratch

```
$ sudo apt-get install ros-noetic-joint-state-publisher
```

```
$ sudo apt-get install ros-noetic-urdf-tutorial
```

One Shape

```
<?xml version="1.0"?>
<robot name="myfirst">
  <link name="base_link">
    <visual>
      <geometry>
        <cylinder length="0.6" radius="0.2"/>
      </geometry>
    </visual>
  </link>
</robot>
```

```
$ roslaunch urdf_tutorial display.launch model:='$(find urdf_tutorial)/urdf/01-myfirst.urdf'
```

Multiple Shapes

```
<?xml version="1.0"?>
<robot name="multipleshapes">
  <link name="base_link">
    <visual>
      <geometry>
        <cylinder length="0.6" radius="0.2"/>
      </geometry>
    </visual>
  </link>
  <link name="right_leg">
    <visual>
      <geometry>
        <box size="0.6 0.1 0.2"/>
      </geometry>
    </visual>
  </link>
  <joint name="base_to_right_leg" type="fixed">
    <parent link="base_link"/>
    <child link="right_leg"/>
  </joint>

</robot>
```

```
$ roslaunch urdf_tutorial display.launch model:='$(find urdf_tutorial)/urdf/02-multipleshapes.urdf'
```

Origins

```
<?xml version="1.0"?>
<robot name="origins">
 <link name="base_link">
  <visual>
   <geometry>
    <cylinder length="0.6" radius="0.2"/>
   </geometry>
  </visual>
 </link>
 <link name="right_leg">
  <visual>
   <geometry>
    <box size="0.6 0.1 0.2"/>
   </geometry>
   <origin rpy="0 1.57075 0" xyz="0 0 -0.3"/>
  </visual>
 </link>
 <joint name="base_to_right_leg" type="fixed">
  <parent link="base_link"/>
  <child link="right_leg"/>
  <origin xyz="0 -0.22 0.25"/>
 </joint>
</robot>
```

```
$ roslaunch urdf_tutorial display.launch model:='$(find urdf_tutorial)/urdf/03-origins.urdf'
```

https://wiki.ros.org/urdf
/Tutorials/Building%20a%20Visual%20Robot%20Model%20with%20URDF%20from%20Scratch

Material Girl

Let’s take a look at the material tag.

```
$ roslaunch urdf_tutorial display.launch model:='$(find urdf_tutorial)/urdf/04-materials.urdf'
```

https://github.com/ros/urdf_tutorial/blob/master/urdf/05-visual.urdf

Finishing the Model

We add a sphere and a some meshes.

```
$ roslaunch urdf_tutorial display.launch model:='$(find urdf_tutorial)/urdf/05-visual.urdf'
```

https://wiki.ros.org/urdf
/Tutorials/Building%20a%20Visual%20Robot%20Model%20with%20URDF%20from%20Scratch

Building a Movable Robot Model with URDF

```
$ roslaunch urdf_tutorial display.launch model:='$(find urdf_tutorial)/urdf/06-flexible.urdf'
```

The Head

```
<joint name="head_swivel" type="continuous">
  <parent link="base_link"/>
  <child link="head"/>
  <axis xyz="0 0 1"/>
  <origin xyz="0 0 0.3"/>
</joint>
```

The Gripper

```
<joint name="left_gripper_joint" type="revolute">
  <axis xyz="0 0 1"/>
  <limit effort="1000.0" lower="0.0" upper="0.548" velocity="0.5"/>
  <origin rpy="0 0 0" xyz="0.2 0.01 0"/>
  <parent link="gripper_pole"/>
  <child link="left_gripper"/>
</joint>
```

The Gripper Arm

```
<joint name="gripper_extension" type="prismatic">
   <parent link="base_link"/>
   <child link="gripper_pole"/>
   <limit effort="1000.0" lower="-0.38" upper="0" velocity="0.5"/>
   <origin rpy="0 0 0" xyz="0.19 0 0.2"/>
  </joint>
```

https://wiki.ros.org/urdf/Tutorials/Building%20a%20Movable%20Robot%20Model%20with%20URDF

Adding Physical and Collision Properties to a URDF Model

Collision

```
<link name="base_link">
  <visual>
   <geometry>
    <cylinder length="0.6" radius="0.2"/>
   </geometry>
   <material name="blue">
    <color rgba="0 0 .8 1"/>
   </material>
  </visual>
  <collision>
   <geometry>
    <cylinder length="0.6" radius="0.2"/>
   </geometry>
  </collision>
 </link>
```

https://wiki.ros.org/urdf/Tutorials/Adding%20Physical%20and%20Collision%20Properties%20to%20a%20URDF%20Model

```
<link name="base_link">
  <visual>
   <geometry>
    <cylinder length="0.6" radius="0.2"/>
   </geometry>
   <material name="blue">
    <color rgba="0 0 .8 1"/>
   </material>
  </visual>
  <collision>
   <geometry>
    <cylinder length="0.6" radius="0.2"/>
   </geometry>
  </collision>
  <inertial>
   <mass value="10"/>
   <inertia ixx="0.4" ixy="0.0" ixz="0.0" iyy="0.4" iyz="0.0" izz="0.2"/>
  </inertial>
 </link>
```

Inertial

mu - Friction coefficient
kp - Stiffness coefficient
kd - Dampening coefficient

Using Xacro to Clean Up a URDF File

```
<param name="robot_description"
  command="xacro --inorder '$(find pr2_description)/robots/pr2.urdf.xacro'" />
```

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

https://wiki.ros.org/urdf/Tutorials/Using%20Xacro%20to%20Clean%20Up%20a%20URDF%20File

Constants

```
<xacro:property name="width" value="0.2" />
<xacro:property name="bodylen" value="0.6" />
<link name="base_link">
  <visual>
    <geometry>
      <cylinder radius="${width}" length="${bodylen}"/>
    </geometry>
    <material name="blue"/>
  </visual>
  <collision>
    <geometry>
      <cylinder radius="${width}" length="${bodylen}"/>
    </geometry>
  </collision>
</link>
```

Constants

```
<xacro:property name=” robotname” value=” marvin” />
<link name=” ${robotname}s_leg” />
```

```
<link name=” marvins_leg” />
```

Math

```
<cylinder radius="${wheeldiam/2}" length="0.1"/>
<origin xyz="${reflect*(width+.02)} 0 0.25" />
```

```
<link name="0.833333333333"/>
```

Simple Macros

```
<xacro:macro name="default_origin">
   <origin xyz="0 0 0" rpy="0 0 0"/>
</xacro:macro>
<xacro:default_origin />
```

Parameterized Macros

```
<xacro:macro name="default_inertial" params="mass">
      <inertial>
            <mass value="${mass}" />
            <inertia ixx="1.0" ixy="0.0" ixz="0.0"
                iyy="1.0" iyz="0.0"
                izz="1.0" />
      </inertial>
   </xacro:macro>
```

```
<xacro:default_inertial mass="10"/>
```

Use entire blocks as parameters

```
<xacro:macro name="blue_shape" params="name *shape">
  <link name="${name}">
    <visual>
      <geometry>
        <xacro:insert_block name="shape" />
      </geometry>
      <material name="blue"/>
    </visual>
    <collision>
      <geometry>
        <xacro:insert_block name="shape" />
      </geometry>
    </collision>
  </link>
</xacro:macro>

<xacro:blue_shape name="base_link">
  <cylinder radius=".42" length=".01" />
</xacro:blue_shape>
```

Leg macro

```
<xacro:macro name="leg" params="prefix reflect">
  <link name="${prefix}_leg">
    <visual>
      <geometry>
        <box size="${leglen} 0.1 0.2"/>
      </geometry>
      <origin xyz="0 0 -${leglen/2}" rpy="0 ${pi/2} 0"/>
      <material name="white"/>
    </visual>
    <collision>
      <geometry>
        <box size="${leglen} 0.1 0.2"/>
      </geometry>
      <origin xyz="0 0 -${leglen/2}" rpy="0 ${pi/2} 0"/>
    </collision>
    <xacro:default_inertial mass="10"/>
  </link>
```

Leg macro

```
  <joint name="base_to_${prefix}_leg" type="fixed">
    <parent link="base_link"/>
    <child link="${prefix}_leg"/>
    <origin xyz="0 ${reflect*(width+.02)} 0.25" />
  </joint>
  <!-- A bunch of stuff cut -->
</xacro:macro>
<xacro:leg prefix="right" reflect="1" />
<xacro:leg prefix="left" reflect="-1" />
```

Using a URDF in Gazebo

```
$ roslaunch urdf_sim_tutorial gazebo.launch
```

To link Gazebo and ROS, we specify the plugin in the URDF, right before the closing </robot> tag:

```
<gazebo>
  <plugin name="gazebo_ros_control" filename="libgazebo_ros_control.so">
   <robotNamespace>/</robotNamespace>
  </plugin>
 </gazebo>
```

```
$ roslaunch urdf_sim_tutorial 09-joints.launch
```

https://wiki.ros.org/urdf/Tutorials/Using%20a%20URDF%20in%20Gazebo
More links:
https://github.com/ros/urdf_sim_tutorial/blob/master/config/joints.yaml

Spawning Controllers

```
type: "joint_state_controller/JointStateController"
publish_rate: 50
```

```
$ roslaunch urdf_sim_tutorial 09-joints.launch
```

Note: if errors in RobotModel, please in /urdf_sim_tutorial, follow as:

```
$ sudo chmod 777 09-joints.launch
```

Open the file 09-joints.launch and add the command as:

```
<node pkg="joint_state_publisher" type="joint_state_publisher" name="joint_state_publisher" />
```

```
$ roslaunch urdf_sim_tutorial 09-joints.launch
```

https://github.com/ros/urdf_sim_tutorial/blob/master/config/joints.yaml

Transmissions

```
<transmission name="head_swivel_trans">
  <type>transmission_interface/SimpleTransmission</type>
  <actuator name="$head_swivel_motor">
    <mechanicalReduction>1</mechanicalReduction>
  </actuator>
  <joint name="head_swivel">
    <hardwareInterface>hardware_interface/PositionJointInterface</hardwareInterface>
  </joint>
 </transmission>
```

```
$ roscd urdf_sim_tutorial
$ roslaunch urdf_sim_tutorial 09-joints.launch model:=urdf/10-firsttransmission.urdf.xacro
```

Joint Control

```
type: "position_controllers/JointPositionController"
joint: head_swivel
```

```
$ roslaunch urdf_sim_tutorial 10-head.launch
```

```
$ rostopic pub /r2d2_head_controller/command std_msgs/Float64 "data: -0.707"
```

Note: if errors as “No p gain specified for pid. Namespace: /gazebo_ros_control/pid_gains/head_swivel”, in /urdf_sim_tutorials, follow as:

```
$ sudo chmod 777 ./config/head.yaml
```

Add the contents:

```
pid: [100.0, 0.01, 10.0]
```

https://wiki.ros.org/position_controllers

Move gradually

```
<joint name="head_swivel" type="continuous">
  <parent link="base_link"/>
  <child link="head"/>
  <axis xyz="0 0 1"/>
  <origin xyz="0 0 ${bodylen/2}"/>
  <limit effort="30" velocity="1.0"/>
 </joint>
```

```
$ roslaunch urdf_sim_tutorial 10-head.launch model:=urdf/11-limittransmission.urdf.xacro
```

Another Controller

```
type: "position_controllers/JointGroupPositionController"
joints:
  - gripper_extension
  - left_gripper_joint
  - right_gripper_joint
```

```
$ roslaunch urdf_sim_tutorial 12-gripper.launch
```

If errors as RobotModel status error, remember to add in 12-gripper.launch:

```
  <node pkg="robot_state_publisher" type="robot_state_publisher"
name="robot_state_publisher"></node>
```

```
rostopic pub  /r2d2_gripper_controller/command std_msgs/Float64MultiArray "layout:
  dim:
  - label: ''
    size: 3
    stride: 1
  data_offset: 0
data: [0, 0.5, 0.5]"
```

```
rostopic pub  /r2d2_gripper_controller/command std_msgs/Float64MultiArray "layout:
  dim:
  - label: ''
    size: 3
    stride: 1
  data_offset: 0
data: [-0.4, 0, 0]"
```

In "data: [-0.4, 0.5, 0.5]", first number means the extension length, the second and third numbers mean the open ranges of two fingers.

The Wheels on the Droid Go Round and Round

```
<transmission name="${prefix}_${suffix}_wheel_trans">
    <type>transmission_interface/SimpleTransmission</type>
    <actuator name="${prefix}_${suffix}_wheel_motor">
      <mechanicalReduction>1</mechanicalReduction>
    </actuator>
    <joint name="${prefix}_${suffix}_wheel_joint">
      <hardwareInterface>hardware_interface/VelocityJointInterface</hardwareInterface>
    </joint>
  </transmission>
```

Specify yet another transmission for each of the wheels from within the wheel macro..

```
<transmission name="${prefix}_${suffix}_wheel_trans">
      <type>transmission_interface/SimpleTransmission</type>
      <actuator name="${prefix}_${suffix}_wheel_motor">
        <mechanicalReduction>1</mechanicalReduction>
      </actuator>
      <joint name="${prefix}_${suffix}_wheel_joint">
        <hardwareInterface>hardware_interface/VelocityJointInterface</hardwareInterface>
      </joint>
    </transmission>
```

Specify some additional information about the material of the wheels.

```
<gazebo reference="${prefix}_${suffix}_wheel">
    <mu1 value="200.0"/>
    <mu2 value="100.0"/>
    <kp value="10000000.0" />
    <kd value="1.0" />
    <material>Gazebo/Grey</material>
  </gazebo>
```

```
$ roslaunch urdf_sim_tutorial 13-diffdrive.launch
```