Warning

You're reading the documentation for a version of ROS 2 that has reached its EOL (end-of-life), and is no longer officially supported. If you want up-to-date information, please have a look at Kilted.

Building a movable robot model

Goal: Learn how to define movable joints in URDF.

Tutorial level: Intermediate

Time: 10 minutes

Contents

In this tutorial, we're going to revise the R2D2 model we made in the previous tutorial so that it has movable joints. In the previous model, all of the joints were fixed. Now we'll explore three other important types of joints: continuous, revolute and prismatic.

Make sure you have installed all prerequisites before continuing. See the previous tutorial for information on what is required.

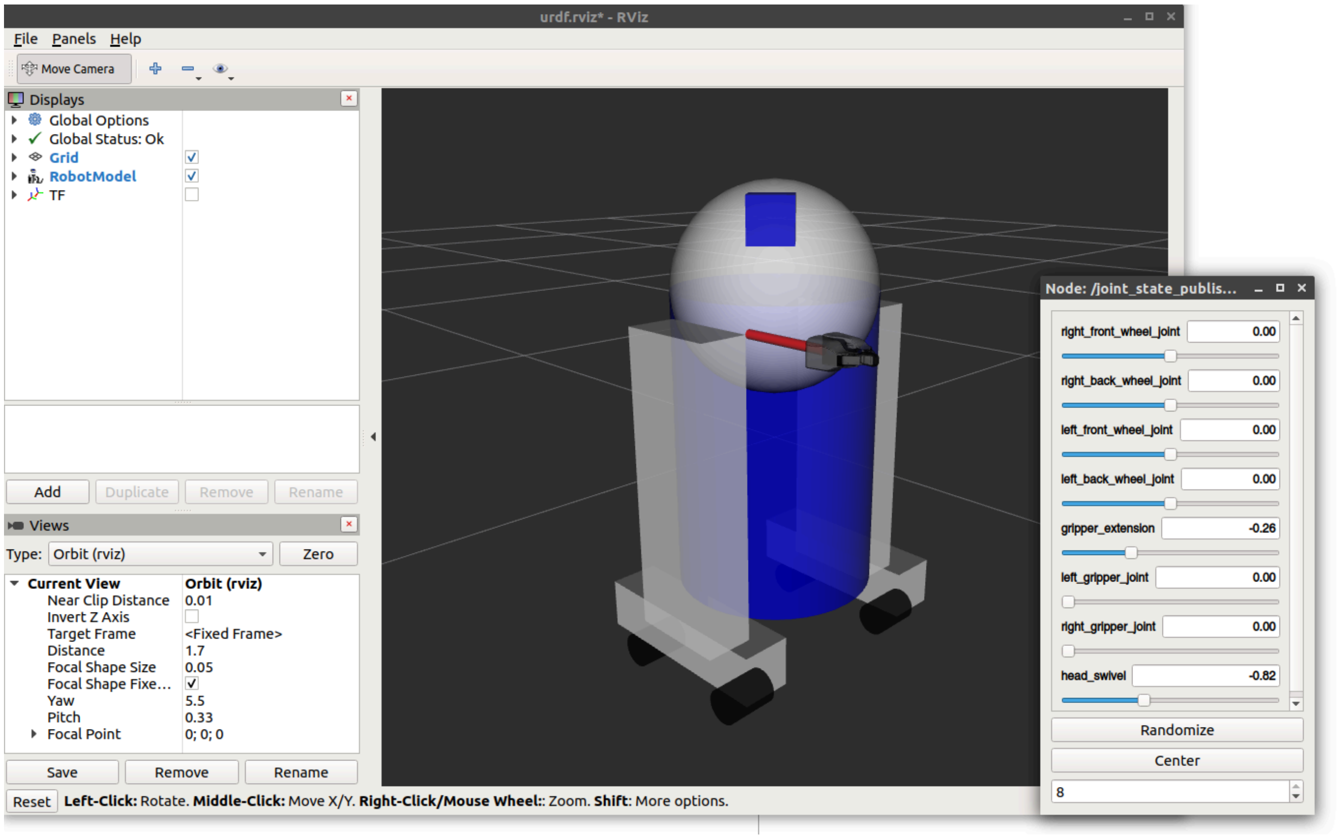
Again, all of the robot models mentioned in this tutorial can be found in the urdf_tutorial package.

Here is the new urdf with flexible joints. You can compare it to the previous version to see everything that has changed, but we're just going to focus on three example joints.

To visualize and control this model, run the same command as the last tutorial:

```
ros2 launch urdf_tutorial display.launch.py model:=urdf/06-flexible.urdf
```

However now this will also pop up a GUI that allows you to control the values of all the non-fixed joints. Play with the model some and see how it moves. Then, we can take a look at how we accomplished this.



Type: Orbit (rviz)

Current View	Orbit (rviz)
Near Clip Distance	0.01
Invert Z Axis	
Target Frame	<Fixed Frame>
Distance	1.7
Focal Shape Size	0.05
Focal Shape Fixe...	✓
Yaw	5.5
Pitch	0.33
Focal Point	0; 0; 0

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

right_front_wheel_joint	0.00
right_back_wheel_joint	0.00
left_front_wheel_joint	0.00
left_back_wheel_joint	0.00
gripper_extension	-0.26
left_gripper_joint	0.00
right_gripper_joint	0.00
head_swivel	-0.82

8

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 connection between the body and the head is a continuous joint, meaning that it can take on any angle from negative infinity to positive infinity. The wheels are also modeled like this, so that they can roll in both directions forever.

The only additional information we have to add is the axis of rotation, here specified by an xyz triplet, which specifies a vector around which the head will rotate. Since we want it to go around the z axis, we specify the vector “0 0 1”.

The Gripper

```xml
<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>
```

Both the right and the left gripper joints are modeled as revolute joints. This means that they rotate in the same way that the continuous joints do, but they have strict limits. Hence, we must include the limit tag specifying the upper and lower limits of the joint (in radians). We also must specify a maximum velocity and effort for this joint but the actual values don't matter for our purposes here.

The Gripper Arm

```xml
<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>
```

The gripper arm is a different kind of joint, namely a prismatic joint. This means that it moves along an axis, not around it. This translational movement is what allows our robot model to extend and retract its gripper arm.

The limits of the prismatic arm are specified in the same way as a revolute joint, except that the units are meters, not radians.

Other Types of Joints

There are two other kinds of joints that move around in space. Whereas the prismatic joint can only move along one dimension, a planar joint can move around in a plane, or two dimensions. Furthermore, a floating joint is unconstrained, and can move around in any of the three dimensions. These joints cannot be specified by just one number, and therefore aren't included in this tutorial.

Specifying the Pose

As you move the sliders around in the GUI, the model moves in Rviz. How is this done? First the GUI parses the URDF and finds all the non-fixed joints and their limits. Then, it uses the values of the sliders to publish sensor_msgs/msg/JointState messages. Those are then used by robot_state_publisher to calculate all of transforms between the different parts. The resulting transform tree is then used to display all of the shapes in Rviz.

Next steps

Now that you have a visibly functional model, you can add in some physical properties, or start using xacro to simplify your code.