2 Expanding RazBot Description for Simulation

Previous: 1 Creating RazBot description
Next: 3 Creating the RazBot Control Configuration

This tutorial will show how to flesh out the robot description from the last tutorial, so that it can be used for simulation. You can find the output for this stage of the tutorials in https://github.com/clearpathrobotics/razbot_tutorials/tree/expanded-description

1. Add collision properties to your links in razbot.urdf.xacro and wheel.urdf.xacro. The relevant links will be *_wheel_link and base_link, since base_footprint is purely for convenience purposes.

```
...
  <!-- Base link, represents center of mass for the robot -->
  <link name="base_link">
    ...
     <!-- <collision> tags describe the collision boundaries of a link. You can
use
    the same contents as the <visual> tag. Even if you use a complex shape for
the
    <visual> tag, it's more efficient to use a basic bounding shape for
collisions -->
    <collision>
      <origin xyz="0 0 0" rpy="0 0 0" />
      <geometry>
        <box size="${base_length} ${base_width} ${base_height}"/>
      </geometry>
    </collision>
  </link>

  ...

  <!-- Wheel link -->
  <link name="${wheel_prefix}_wheel_link">
    ...
    <collision>
      <origin xyz="0 0 0" rpy="${M_PI/2} 0 0" />
      <geometry>
        <cylinder length="${wheel_width}" radius="${wheel_radius}" />
      </geometry>
    </collision>
  </link>
  ...
```

2. Add intertial parameters to your links in razbot.urdf.xacro and wheel.urdf.xacro. Intertial information can be estimated, or calculated using a CAD package.

```
...
  <!-- Base link, represents center of mass for the robot -->
  <link name="base_link">
    ...
    <!-- <interial> tags describe intertial properties of the link. These are 
relative
    to the origin of the link, so make sure the data you obtain is consistent -->
    <inertial>
      <origin xyz="0 0 0" />
      <mass value="1.5" /> <!-- mass in kg -->
      <inertia  ixx="0.01" ixy="0" ixz="0"
                iyy="0.01" iyz="0"
                izz="0.01" /> <!-- inertia around each axis pair -->
    </inertial>
  </link>

  ...

  <!-- Wheel link -->
  <link name="${wheel_prefix}_wheel_link">
    ...
      <inertial>
        <origin xyz="0 0.1 0"/>
        <mass value="0.15" />
        <inertia  ixx="0.0001" ixy="0" ixz="0"
                  iyy="0.0001" iyz="0"
                  izz="0.0001" />
      </inertial>
  </link>
  ...
```

3. The wheels require some special treatment for proper simulation in Gazebo, see the two additional blocks below. For an explanation of the physical constants (mu1, mu2, kp, kd), please see http://gazebosim.org/tutorials/?tut=ros_urdf.

```
...
  <xacro:macro name="razbot_wheel" params="wheel_prefix *joint_pose">
    ....
    <!-- This block provides the simulator (Gazebo) with information on a few
additional
    physical properties. See http://gazebosim.org/tutorials/?tut=ros_urdf for
more-->
    <gazebo reference="${wheel_prefix}_wheel_link">
      <mu1 value="200.0"/>
      <mu2 value="100.0"/>
      <kp value="10000000.0" />
      <kd value="1.0" />
    </gazebo>

    <!-- This block connects the wheel joint to an actuator (motor), which
informs both
    simulation and visualization of the robot -->
    <transmission name="${wheel_prefix}_wheel_trans" type="SimpleTransmission">
      <type>transmission_interface/SimpleTransmission</type>
      <actuator name="${wheel_prefix}_wheel_motor">
        <mechanicalReduction>1</mechanicalReduction>
      </actuator>
      <joint name="${wheel_prefix}_wheel">
        <hardwareInterface>VelocityJointInterface</hardwareInterface>
      </joint>
    </transmission>
    ...
  </xacro:macro>
  ...
```