

[← Back \(/tutorials?cat=sensors\)](/tutorials?cat=sensors)

Force/Torque Sensor

Introduction

This tutorial describes how to use a force/torque sensor on a joint. This sensor publishes force and torque readings to a topic.

Quick Start

Part 1: See the sensor in action

Create a world with a force/torque sensor

Save this world as `force_torque_tutorial.world`

(https://github.com/osrf/gazebo_tutorials/raw/master/force_torque_sensor/files/force_torque_tutorial.world).

```

<?xml version="1.0"?>
<sdf version="1.6">
  <world name="default">
    <physics name="default_physics" default="0" type="ode">
      <max_step_size>0.001</max_step_size>
      <real_time_factor>1</real_time_factor>
      <real_time_update_rate>1000</real_time_update_rate>
      <ode>
        <solver>
          <type>quick</type>
          <iters>50</iters>
          <sor>1.0</sor> <!-- Important, see issue #2209 -->
          <use_dynamic_moi_rescaling>false</use_dynamic_moi_rescaling>
        </solver>
      </ode>
    </physics>
    <include>
      <uri>model://ground_plane</uri>
    </include>
    <include>
      <uri>model://sun</uri>
    </include>
    <model name="model_1">
      <link name="link_1">
        <pose>0 0 2.0 0 0 0</pose>
        <inertial>
          <inertia>
            <ixx>0.100000</ixx>
            <ixy>0.000000</ixy>
            <ixz>0.000000</ixz>
            <iyy>0.100000</iyy>
            <iyz>0.000000</iyz>
            <izz>0.100000</izz>
          </inertia>
          <mass>10.000000</mass>
        </inertial>
        <visual name="visual_sphere">
          <geometry>
            <sphere>
              <radius>0.100000</radius>
            </sphere>
          </geometry>
        </visual>
        <visual name="visual_cylinder">
          <pose>0 0 -0.75 0 0 0</pose>
          <geometry>
            <cylinder>
              <radius>0.0100000</radius>
              <length>1.5</length>
            </cylinder>
          </geometry>
        </visual>
        <collision name="collision_sphere">
          <max_contacts>250</max_contacts>
          <geometry>
            <sphere>
              <radius>0.100000</radius>
            </sphere>
          </geometry>
        </collision>
      </link>
      <joint name="joint_01" type="revolute">
        <parent>world</parent>
        <child>link_1</child>
        <pose>0 0 -1.5 0 0 0</pose>
        <axis>
          <limit>
            <lower>-1.57079</lower>

```

```

    <upper>1.57079</upper>
  </limit>
  <dynamics>
    <damping>0.000000</damping>
    <friction>0.000000</friction>
  </dynamics>
  <xyz>1.000000 0.000000 0.000000</xyz>
</axis>
<sensor name="force_torque" type="force_torque">
  <update_rate>30</update_rate>
</sensor>
</joint>
</model>
</world>
</sdf>

```

Launch the world

Open a terminal and run this command:

```
gazebo --verbose force_torque_tutorial.world
```

View the output of the force/torque sensor

In a new terminal open the topic viewer with the following command:

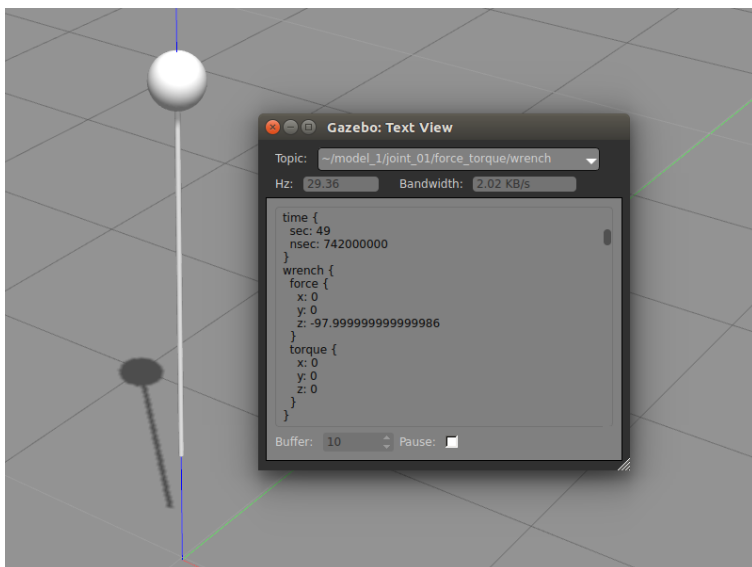
```
gz topic --view /gazebo/default/model_1/joint_01/force_torque/wrench
```

Apply forces and torques in gazebo

Apply a force (tutorials?tut=apply_force_torque) to `link_1` of 500 N in the +Y direction. Observe the output in the topic viewer window.

Part 2: Explanation of the above steps

At the start of the world



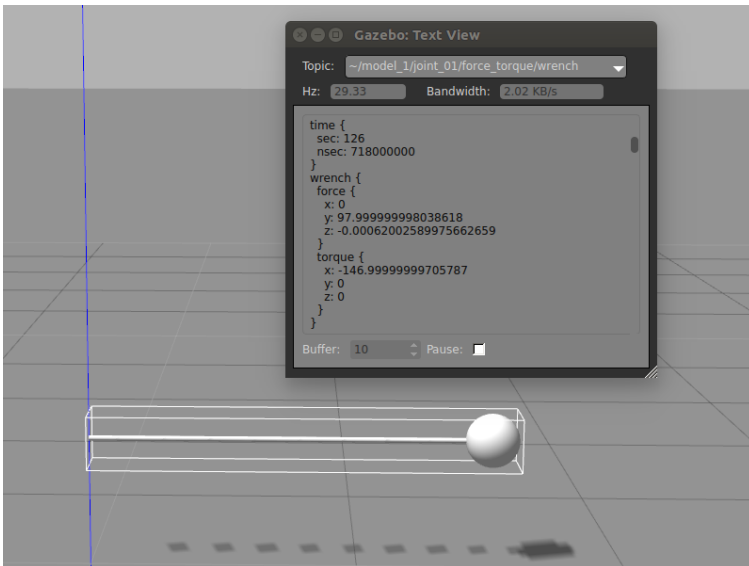
This world has one link and one joint. The link is a sphere with a mass of 10 kg offset 1.5 m from the joint. The joint connects the sphere to the world, allowing rotation on its X axis. At the start the sphere is balanced above the joint. There is no torque on the joint. The force on the joint comes from gravity.

```

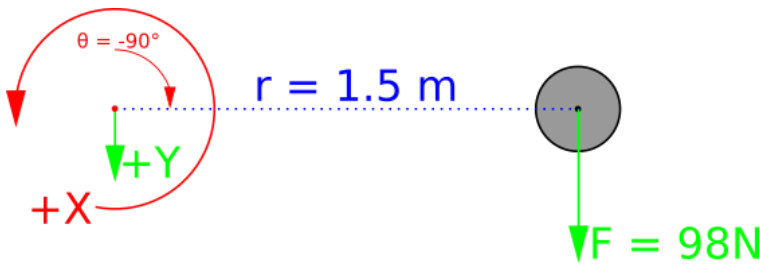
forceJointZ = mass * g
              = 10 kg * -9.8 m/s^2
              = -98 N

```

After applying a force to the link



Applying a force to `link_1` causes it to topple over and rest at a 90 degree angle, the limit of the joint. The limit is what keeps the sphere suspended above the ground plane. The joint's +Y axis points towards the ground plane. Gravity acting on the sphere applies a torque about the X axis.



The mass supported by the joint remains the same, so the magnitude of the force is the same. The direction of the force changes to be +98 N on the Y axis. Because the sphere is positioned 1.5 m away on the joint's Z axis the torque about the X axis is:

```
torqueJoint01_x = r X R
                = ||r|| * ||F|| * sin(theta)
                = distanceZ * (massLink * g) * sin(theta)
                = 1.5 m * (10 kg * 9.8 m/s^2) * sin(-90)
                = -147 Nm
```

Note: Measurements near joint limits may jump depending on physics engine parameters. See issue #2209 (<https://github.com/osrf/gazebo/issues/2209>)

Understanding the Force/Torque Sensor

SDF parameters

Generic parameters

All sensors have a common set of parameters in the SDF sensor schema (<http://sdformat.org/spec?ver=1.6&elem=sensor>).

<always_on>

If `true` the sensor will always measure force and torque. If `false` the sensor will only update if there is a subscriber connected to the sensor's topic. This setting is important when accessing the sensor through code. Calls to `ForceTorqueSensor::Torque()` (http://osrf-distributions.s3.amazonaws.com/gazebo/api/7.1.0/classgazebo_1_1sensors_1_1ForceTorqueSensor.html#a050a369346fb2d338ae956b2fc01b7a0) or `ForceTorqueSensor::Force()` (http://osrf-distributions.s3.amazonaws.com/gazebo/api/7.1.0/classgazebo_1_1sensors_1_1ForceTorqueSensor.html#ad578fc59083b6ceca7924d2335402d55) will return stale data if there are no subscribers. This can be detected by checking if `IsActive()` (<http://osrf->

distributions.s3.amazonaws.com/gazebo/api/7.1.0/classgazebo_1_1sensors_1_1ForceTorqueSensor.html#aa58bb55f74716e3435ae09ed212df26f) returns `false` . Code can make a sensor update with no subscribers by calling `SetActive(true)` ([http://osrf-distributions.s3.amazonaws.com/gazebo/api/7.1.0/classgazebo_1_1Sensor.html#a9a22d0e822e1d2bcd03c8594a0f5a40b](http://osrf-distributions.s3.amazonaws.com/gazebo/api/7.1.0/classgazebo_1_1sensors_1_1Sensor.html#a9a22d0e822e1d2bcd03c8594a0f5a40b)).

<update_rate>

This is the rate in Hz that the sensor should update itself. It is the number of messages that will be published by this sensor per simulated second.

<visualize>

If `true` the gazebo client will show a visualization of the force and torque at the joint.

<topic>

The force/torque sensor does not currently support this parameter.

<frame>

The force/torque sensor does not currently support this parameter.

<pose>

Floating point numbers separated by spaces with this order `x y z roll pitch yaw` . It describes the location of the sensor frame with respect to the parent joint.

Force/Torque specific parameters

A force/torque sensor is created by adding `<sensor>` tag with the attribute `type` set to `force_torque` . There are two additional parameters that can be set.

```
<sensor name="my_cool_sensor" type="force_torque">
  <force_torque>
    <frame>child</frame>
    <measure_direction>child_to_parent</measure_direction>
  </force_torque>
</sensor>
```

<frame>

The value of this element may be one of: `child` , `parent` , or `sensor` . It is the frame in which the forces and torques should be expressed. The values `parent` and `child` refer to the parent or child links of the joint. The value `sensor` means the measurement is rotated by the rotation component of the `<pose>` of this sensor. The translation component of the pose has no effect on the measurement.

Regardless of this setting, the torque component is always expressed about the origin of the joint frame.

<measure_direction>

This is the direction of the measurement. Try changing the example above to `parent_to_child` . After being toppled the sensor reports a force of -98 N on the Y axis and a torque of +147 Nm about the X axis. This is the same magnitude as before but opposite direction.

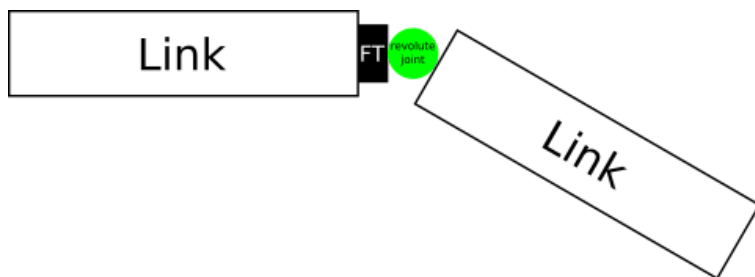
Adding a force/torque sensor to a link

While the SDF schema allows a `<sensor>` tag to be placed on either a link or a joint, the force/torque sensor only works on joints. If the sensor is added to a link, running gazebo with `--verbose` shows the following error:

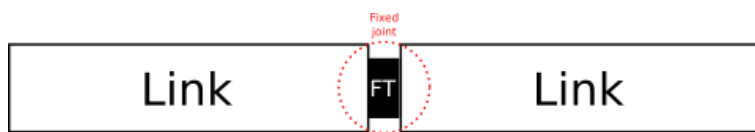
```
[Err] [Link.cc:114] A link cannot load a [force_torque] sensor.
```

Modeling a Real Force/Torque Sensor

The above example places a force/torque sensor on a revolute joint. However, real force/torque sensors are typically rigidly mounted to another rigid body. A real sensor could not measure the force and torque exactly at the revolute joint origin. Modeling this way is reasonable if the real sensor is close enough to the joint that the error from the offset is negligible.





If this error is not negligible, the rigid body can be split into two links with a fixed joint at the location of the real sensor.



©2014 Open Source Robotics Foundation

Gazebo is open-source licensed under
Apache 2.0
(<http://www.apache.org/licenses/LICENSE-2.0.html>)


(<https://plus.google.com/u/0/1159814362965718003>
prsrc=3) 
(<https://www.youtube.com/channel/UCJyqf9XJp>)