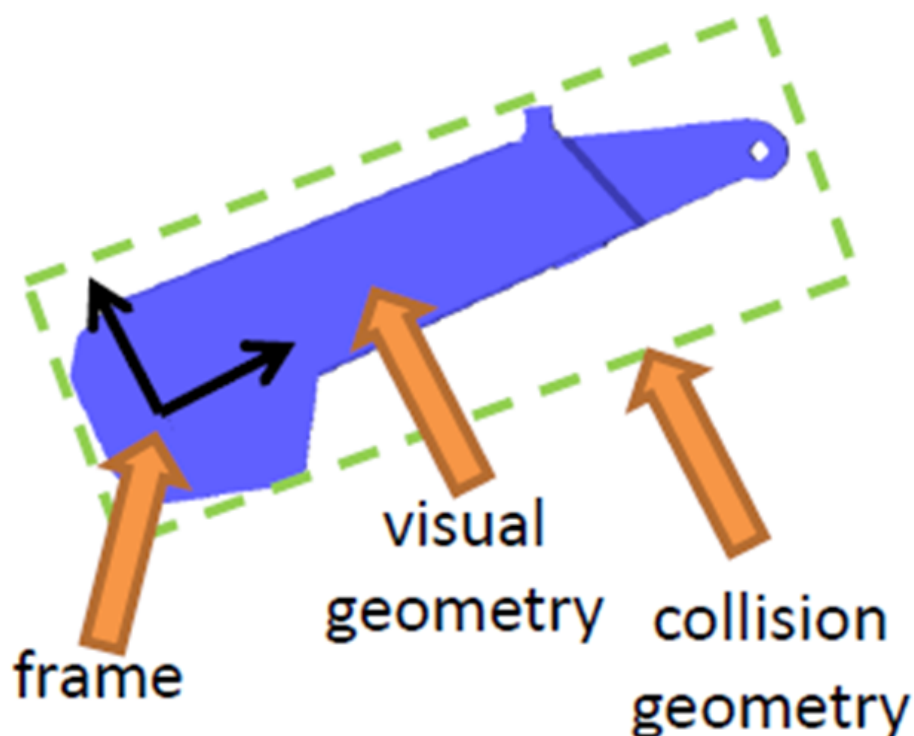


3.1、连杆Link的描述

标签用来描述机器人某个刚体部分的外观和物理属性，外观包括尺寸、颜色、形状，物理属性包括质量、惯性矩阵、碰撞参数等。



以这个机械臂连杆为例，它的link描述如下：

```
<link name="link_4">
  <visual>
    <geometry>
      <mesh filename="link_4.stl"/>
    </geometry>
    <origin xyz="0 0 0" rpy="0 0 0" />
  </visual>
  <collision>
    <geometry>
      <cylinder length="0.5" radius="0.1"/>
    </geometry>
    <origin xyz="0 0 -0.05" rpy="0 0 0" />
  </collision>
</link>
```

link标签中的name表示该连杆的名称，我们可以自定义，未来joint连接link的时候，会使用到这个名称。

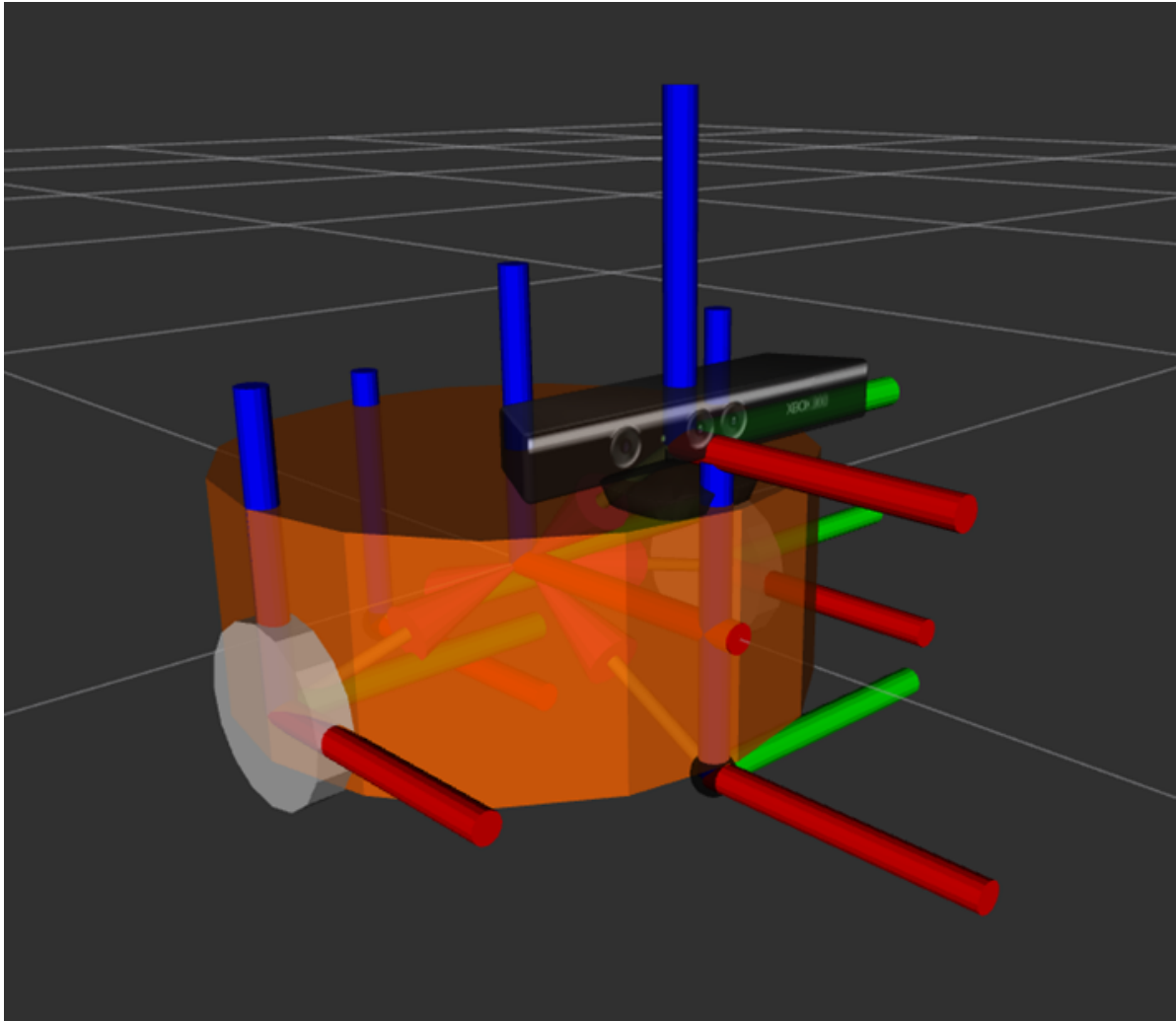
link里边的部分用来描述机器人的外观，比如：

- 表示几何形状，里边使用调用了一个在三维软件中提前设计好的蓝色外观，就是这个stl文件，看上去和真实机器人是一致的
- 表示坐标系相对初始位置的偏移，分别是x、y、z方向上的平移，和roll、pitch、yaw旋转，不需要偏移的话，就全为0。

第二个部分，描述碰撞参数，里边的内容似乎和一样，也有和，看似相同，其实区别还是比较大的。

- 部分重在描述机器人看上去的状态，也就是视觉效果；
- 部分则是描述机器人运动过程中的状态，比如机器人与外界如何接触算作碰撞。

在这个机器人模型中，蓝色部分是通过来描述的，在实际控制过程中，这样复杂的外观在计算碰撞检测时，要求的算力较高，为了简化计算，我们将碰撞检测用的模型简化为了绿色框的圆柱体，也就是里边描述的形状。坐标系偏移也是类似，可以描述刚体质心的偏移。



如果是移动机器人的话，link也可以用来描述小车的车体、轮子等部分。

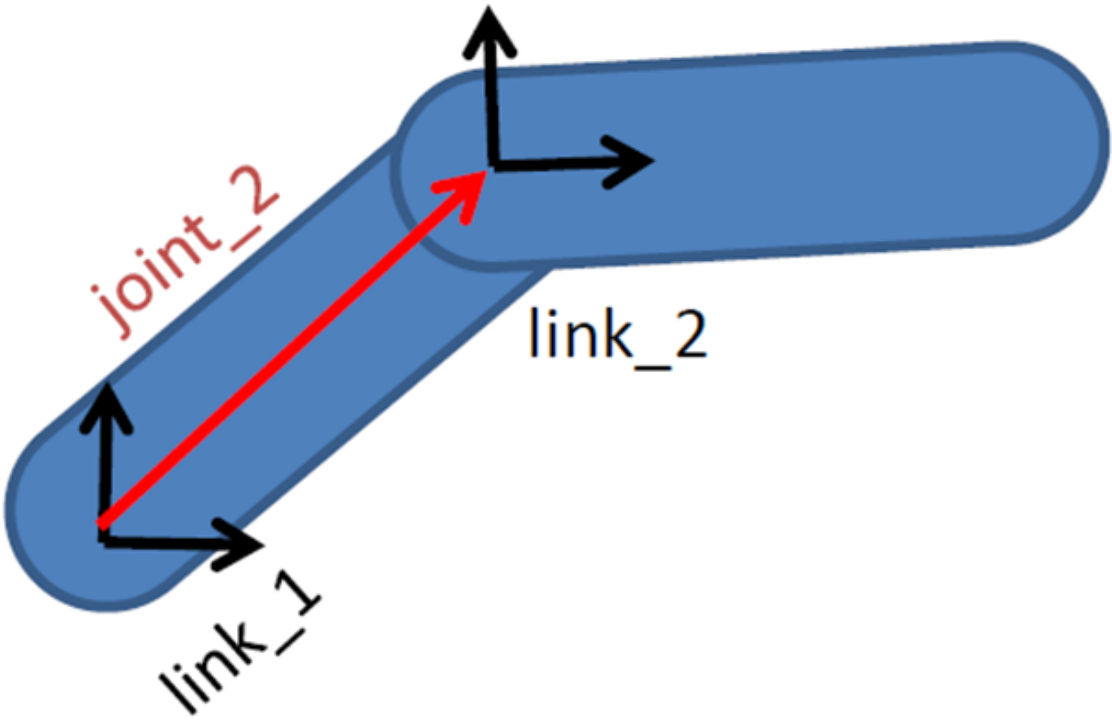
3.2、关节Joint描述

机器人模型中的刚体最终要通过关节joint连接之后，才能产生相对运动。

URDF中的关节有六种运动类型。

关节类型	描述
continuous	旋转关节，可以围绕单轴无限旋转
revolute	旋转关节，类似于continuous，但是有旋转的角度极限
prismatic	滑动关节，沿某一轴线移动关节，带有位置极限
fixed	固定关节，不允许运动的特殊关节
floating	浮动关节，允许进行平移、旋转运动
planar	平面关节，允许在平面正交方向上平移或者旋转

- 1. continuous，描述旋转运动，可以围绕某一个轴无限旋转，比如小车的轮子，就属于这种类型。
- 2. revolute，也是旋转关节，和continuous类型的区别在于不能无限旋转，而是带有角度限制，比如机械臂的两个连杆，就属于这种运动。
- 3. prismatic，是滑动关节，可以沿某一个轴平移，也带有位置的极限，一般直线电机就是这种运动方式。
- 4. fixed，固定关节，是唯一一种不允许运动的关节，不过使用还是比较频繁的，比如相机这个连杆，安装在机器人上，相对位置是不会变化的，此时使用的连接方式就是Fixed。
- 5. Floating是浮动关节，第六种planar是平面关节，这两种使用相对较少。

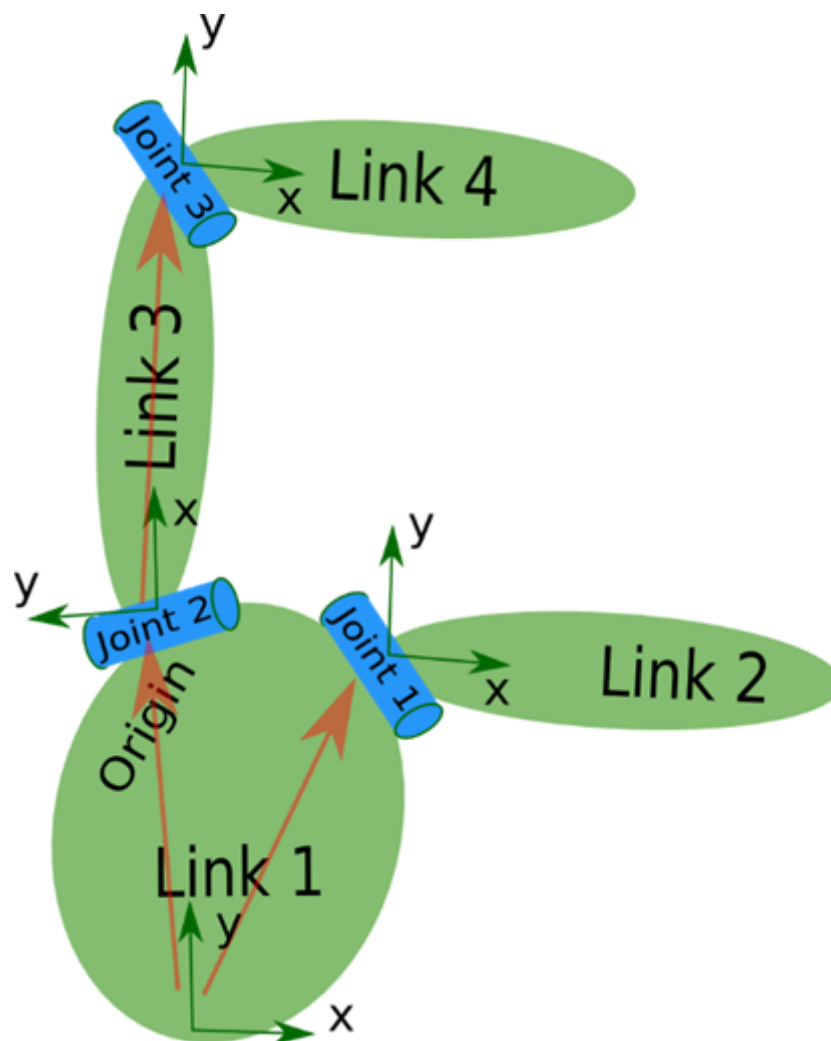


在URDF模型中，每一个link都使用这样一段xml内容描述，比如关节的名字叫什么，运动类型是哪一种。

```
<joint name="joint_2" type="revolute">
  <parent link="link_1"/>
  <child link="link_2"/>
  <origin xyz="0.2 0.2 0" rpy="0 0 0"/>
  <axis xyz="0 0 1"/>
  <limit lower="-3.14" upper="3.14" velocity="1.0"/>
</joint>
```

- parent标签：描述父连杆；
- child标签：描述子连杆，子连杆会相对父连杆发生运动；
- origin：表示两个连杆坐标系之间的关系，也就是图中红色的向量，可以理解为这两个连杆该如何安装到一起；
- axis表示关节运动轴的单位向量，比如z等于1，就表示这个旋转运动是围绕z轴的正方向进行的；
- limit就表示运动的一些限制了，比如最小位置，最大位置，和最大速度等。

4、完整机器人模型



最终所有的link和joint标签完成了对机器人每个部分的描述和组合，全都放在一个robot标签中，就形成了完整的机器人模型。

```
<robot name="<name of the robot>">
  <link> ..... </link>
  <link> ..... </link>

  <joint> ..... </joint>
  <joint> ..... </joint>
</robot>
```

所以大家在看某一个URDF模型时，先不着急看每一块代码的细节，先来找link和joint，看下这个机器人是由哪些部分组成的，了解完全局之后，再看细节。

4.1、创建机器人模型

以muto的模型为例，将本节教程文件夹中的 `yahboomcar_description` 功能包复制到工作空间的src目录下：



- urdf：存放机器人模型的URDF或xacro文件
- meshes：放置URDF中引用的模型渲染文件
- launch：保存相关启动文件
- rviz：保存rviz的配置文件

之后编译功能包

```
colcon build --packages-select yahboomcar_description
```

```
yahboom@yahboom-virtual-machine: ~/workspace
yahboom@yahboom-virtual-machine: ~/workspace 110x24
yahboom@yahboom-virtual-machine:~/workspace$ colcon build --packages-select yahboomcar_description
Starting >>> yahboomcar_description
/usr/lib/python3/dist-packages/setuptools/dist.py:723: UserWarning: Usage of dash-separated 'script-dir' will
not be supported in future versions. Please use the underscore name 'script_dir' instead
  warnings.warn(
/usr/lib/python3/dist-packages/setuptools/dist.py:723: UserWarning: Usage of dash-separated 'install-scripts'
will not be supported in future versions. Please use the underscore name 'install_scripts' instead
  warnings.warn(
--- stderr: yahboomcar_description
/usr/lib/python3/dist-packages/setuptools/dist.py:723: UserWarning: Usage of dash-separated 'script-dir' will
not be supported in future versions. Please use the underscore name 'script_dir' instead
  warnings.warn(
---
Finished <<< yahboomcar_description [0.72s]

Summary: 1 package finished [0.92s]
1 package had stderr output: yahboomcar_description
yahboom@yahboom-virtual-machine:~/workspace$
```

4.2、模型可视化效果

- 刷新环境变量，然后运行启动命令

```
ros2 launch yahboomcar_description display.launch.py
```

之后会自动启动rviz，并显示机器人模型：

