**中国大学慕课**

**实训文档**

**ROS机器人开发技术**

**实训6.1 使用URDF创建一个可视化机器人模型**

|  |  |
| --- | --- |
| **姓名:** | **组号:** |
| **实训负责人:** | **日期:** |

# 背景知识

在本次实训中，我们将要创建一个看起来像星球大战里的R2D2的机器人可视化模型，在后面的实训课中，我们将要学到如何表达模型，如何增加一些物理属性，如何使用xacro生成更为简洁的代码和如何使模型可以在Gazebo中呈现，本节课的实训内容打算从基础开始，让学生从简单的圆柱体入手，去构造一些自己想要的形状。

# 子任务1：创建一个类似与R2D2的机器人的可视化模型：创建机器人主躯干的圆柱体。 (20min)

**第一步操作：**确认安装joint\_state\_publisher软件包，否则的话，可视化模型无法在rviz中显示。另外也要确认我们已经安装了urdf\_tutorial软件包，如果没有安装，最好把这个软件包也安装上，因为urdf\_tutorial 中有本次实训课中提到的所有机器人模型代码，我们可以参考下面的操作：

安装joint\_state\_publisher软件包，在终端输入：

sudo apt install ros-kinetic-joint-state-publisher

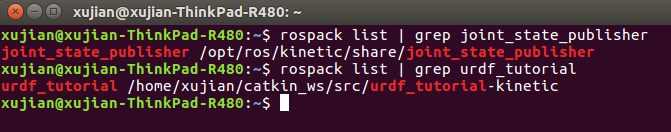
安装urdf\_tutorial软件包

sudo apt install ros-kinetic-urdf-tutorial

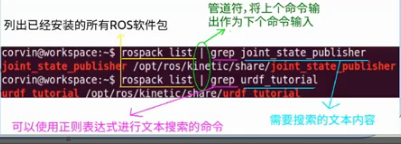
检查是否安装某ROS软件包，命令如下

rospack list | grep \*\*\*

如下图：



相关注释可以参考下图：

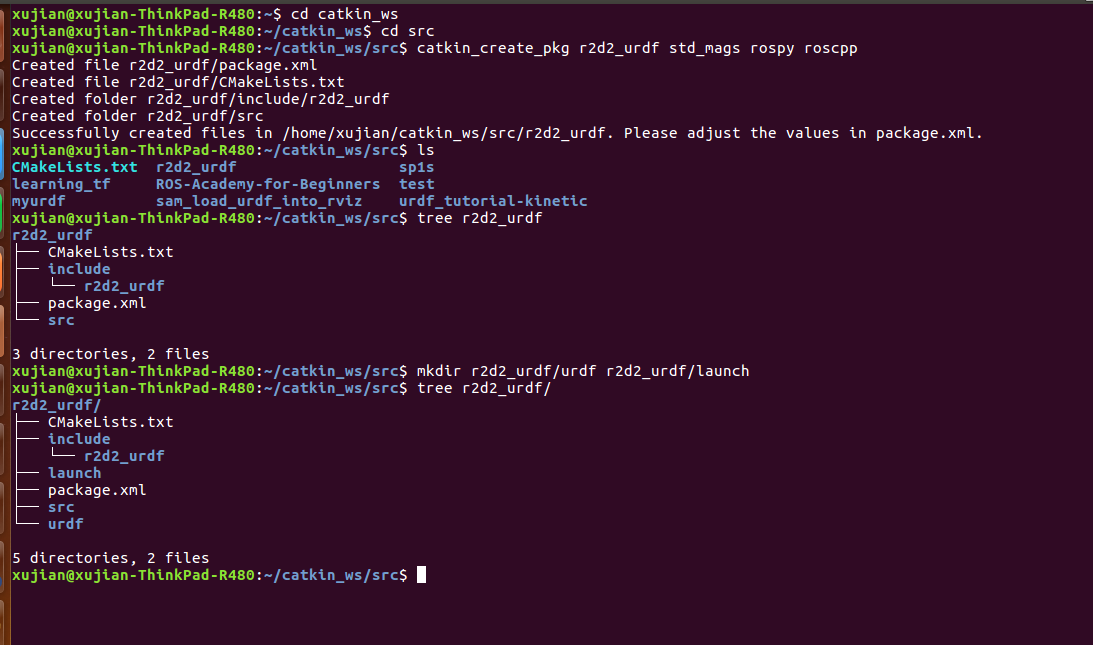


**第二步操作：**熟悉R2-D2的外观造型。参考下图，R2-D2的主要构成模型：头部是一个半球体，凸出的圆柱体是眼睛；主躯干是一个圆柱体；左右两边差不多是两条长方体的支撑腿；主躯干下面有一个类似于金字塔造型的支撑腿，有了以上的几个大模块，一个R2-D2的造型基本能被构造出来。



**第三步操作：**

我们自己创建一个代码目录，创建一个pkg，为r2d2\_urdf，创建launch目录存放启动rviz用于加载模型显示的launch文件。我们可以用tree命令来查看最终的目录结构树，如下图：（下图也可以在<其他素材>文件夹下图片1找到）



**第四步操作：**

编写urdf代码，我们从最简单的部分开始，先创建R2D2的主躯干就是一个圆柱体，我们假设主躯干的半径0.2米，长度0.6米，ROS里面把每个可视化子模块称为link，子模块与子模块之间通过关节joint连接，作为机器人最基础的link统一称为base\_link。在官方的所有代码里面到处可以看到base\_link，所以最好也延续该名称，其他的link都要依附到base\_link上。

几何结构除了<cylinder>圆柱体、<box>立方体、 <cylinder>圆柱体、<sphere>球体，后面会都介绍到。

代码内容如下：

1 <?xml version="1.0"?>

2 <robot name="myfirst">

3 <link name="base\_link">

4 <visual>

5 <geometry>

6 <cylinder length="0.6" radius="0.2"/>

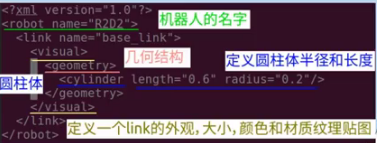
7 </geometry>

8 </visual>

9 </link>

10 </robot>

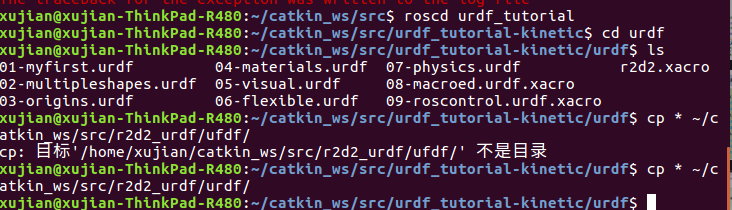
代码注解如下图：



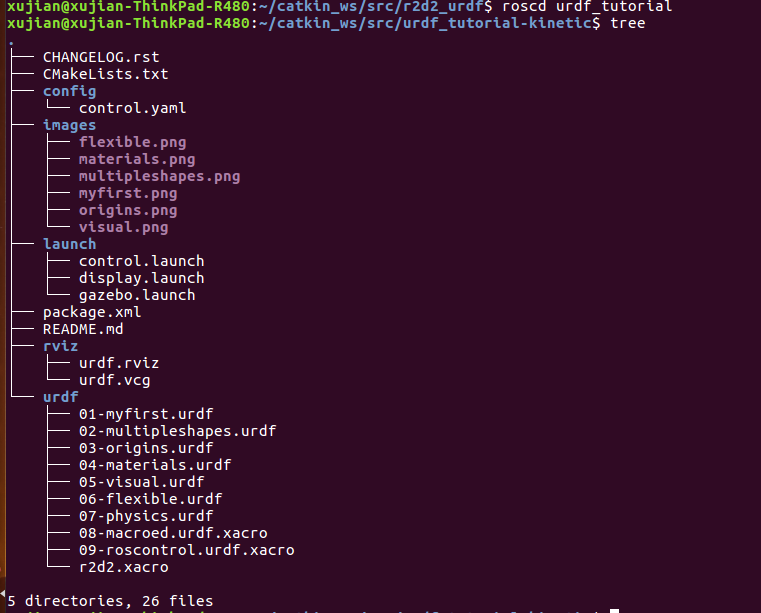
**第五步操作：**

总结思考urdf\_tutorial软件包提供的display.launch是如何将我们的urdf源码文件解析显示出来的。我们来模仿urdf\_tutorial提供的display.launch文件来编写属于我们自己的launch文件，以后方便我们进行DIY。

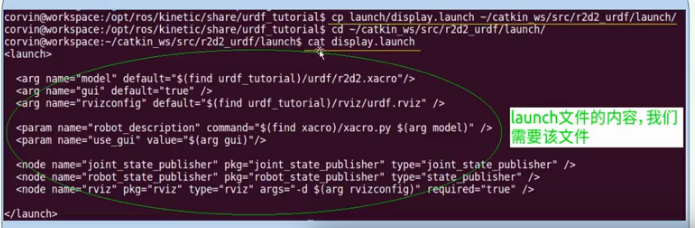
这里，我们把之前下载的urdf\_tutorial下面的urdf文件全部拷贝到我们刚刚建立的r2d2工作空间下即可，命令如下图：



目录树如下：



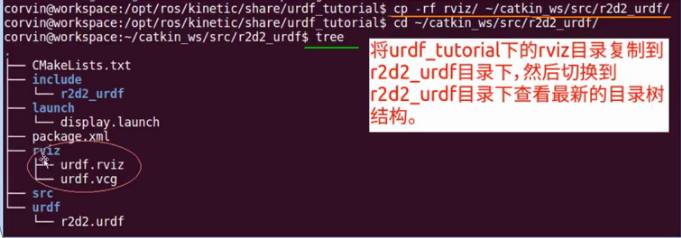
这里，rviz文件夹用于存放rviz显示时的配置文件，以后也会用到。首先将urdf\_tutorial提供的display.launch文件，复制到我们自己工作目录下，如下图的操作：



由于在launch文件里面，有：

<argname="rvizconfig"default="$(find urdf\_tutorial)/rviz/urdf.rviz" />

也就是在launch文件里面，有：rviz文件夹下面的文件，因此，我们也要把urdf\_tutorial 下面的 rviz 文件夹拷贝到我们的r2d2\_urdf文件夹下面，操作如下图：



接着，我们修改代码如下，变成我们自己可以执行的launch文件，并且可以在rviz显示我们的r2d2模型。这里其实就是换了几行代码。具体代码如下，代码在文件夹<urdf实训源码>里的launch文件夹下面：

<launch>

<arg name="model" default="$(find r2d2\_urdf)/urdf/r2d2.xacro"/>

<arg name="gui" default="true" />

<arg name="rvizconfig" default="$(find r2d2\_urdf)/rviz/urdf.rviz" />

<param name="robot\_description" command="$(find xacro)/xacro.py $(arg model)" />

<param name="use\_gui" value="$(arg gui)"/>

<node name="joint\_state\_publisher" pkg="joint\_state\_publisher" type="joint\_state\_publisher" />

<node name="robot\_state\_publisher" pkg="robot\_state\_publisher" type="state\_publisher" />

<node name="rviz" pkg="rviz" type="rviz" args="-d $(arg rvizconfig)" required="true" />

</launch>

**相关代码注释：**

第1行：宣告launch档，格式如：<launch> … </launch>

第2行：引数的语法会像这样：<arg name="…" value="…">

其中name是参数的名称。Value 是参数的值。有时候也用default=”…”来设定预设值。

用$(find <pkg>)这种语法来直接找包裹下的路径，所以不管这个包裹的路径被更改，程式照样能找得到目标。

第5-6行：<param>，用来定义一个设置在parameter server的参数，可以添加到中。参数用法如下：

Name=”namespace/name”参数名字。

Value=”value”（可选）定义参数的值，如果省略这个参数，则应该指定一个文件（binfile/textfile）或命令：command=” (find pkg-name)/exe’ $(find pkg-name)/arg.txt’” ，其中，exe是可执行文件，arg.txt参数文件。

第7-8行：呼叫节点：<node pkg="…" type="…" name="…" respawn=true ns="…" args=”….”/>

pkg：表示该节点所在的包。type：表示这个节点实际的名称，也就是开发的时候取的名字。name：虽然也是指该节点的名称，不过你可以再另外帮这个节点取名字，那么该节点便会把原名给覆盖掉，以这个名称表示。

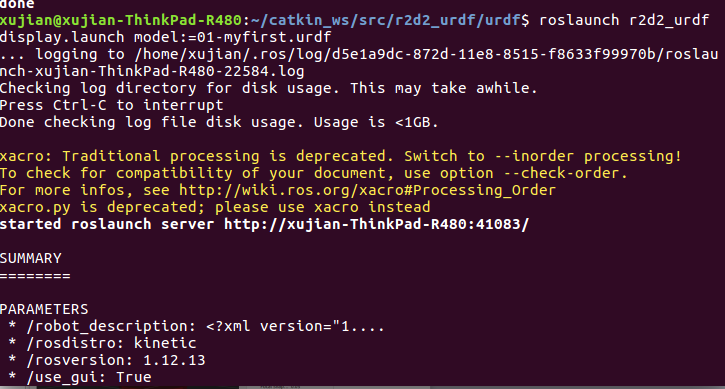
respawn/required：是当该节点由于不明原因停止执行的时候，会自动重新启动。而required比较霸道一点，当该节点停止执行的时候，会让整个launch 文档都停止执行、关闭。

**第六步操作：**

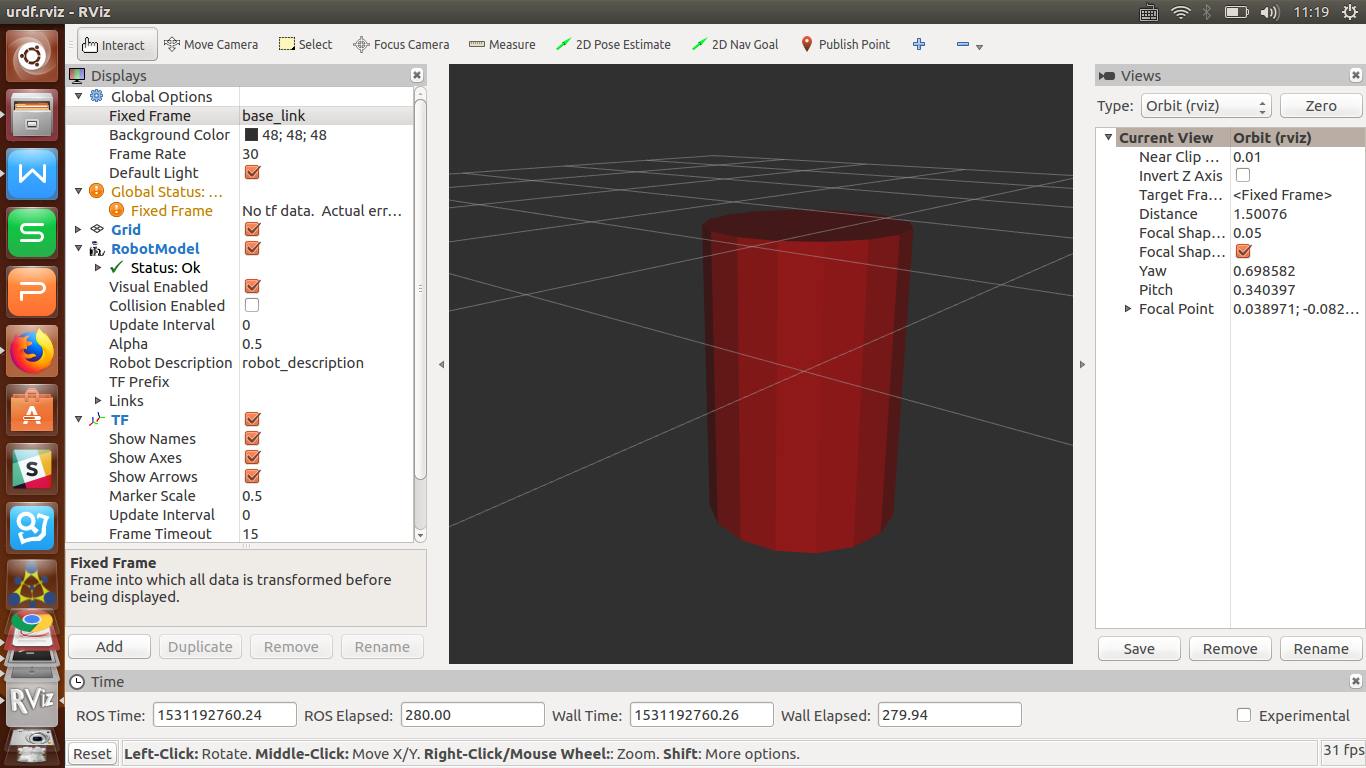
要想用修改后的launch文件，首先需要切换到catkin\_ws工作空间下，catkin\_make编译完成后，

Source devel/setup.bash后，在launch文件中的$（find r2d2\_urdf）才能找到r2d2\_urdf目录，最后通过roslaunch加载该display.launch文件即可。

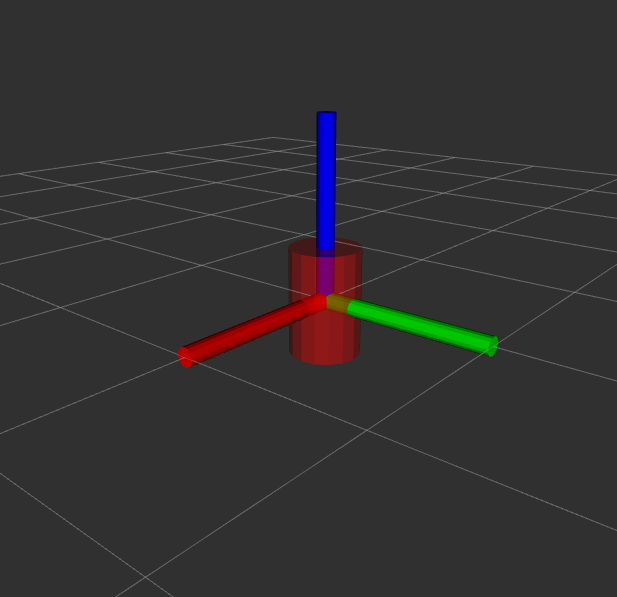
于是，我们可以可视化我们刚刚写的那个圆柱体，命令如下图(在<其他素材>文件夹下的)：



所得到的结果如下图：



我们添加轴以后，会得到下图，可以清楚的看到红绿蓝对应坐标系的xyz轴：



**课堂作业：**

请自己在工作空间新建一个包，包的名字叫做homework，之后建立一个半径为0.3，长度为0.6的圆柱体模型，让它在rviz中可视化。

代码在<源码>文件夹下面的<homework>里，图片在<其他素材>文件夹下。

# 背景知识

在本次实训中，我们将要创建一个看起来像星球大战里的R2D2的机器人可视化模型。我们在上一个子任务中，已经能够完成简单的模型创建。事实上，很多模型都可以看做一系列简单的模型的空间组合，但是如何把这些简单的模型连接起来，如何确定模型的颜色等参数，都是我们需要在设计的过程中考虑的，本次实训的重点在于如何正确的创建出可视化的机器人模型。

# 子任务2：完整创建一个类似与R2D2的机器人可视化模型。 (25min)

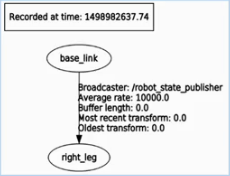
**第一步操作：**

由于现在的R2D2只有一个主躯干的圆柱体，接下来需要增加更多的可视化子模块link，但是如果只是简单的增加link到urdf中，解释器并不知道它们放在哪里，因此需要增加关节joint，这样就可以将link与link连接起来。Joint分为活动的关节和非活动关节，在本次实训中，我们先增加非活动或者叫固定的关节 ，代码在<源码>文件夹下的urdf文件夹下的02-multipleshapes.urdf中

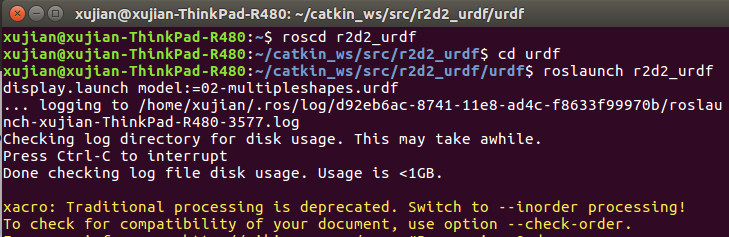
代码解释如下：



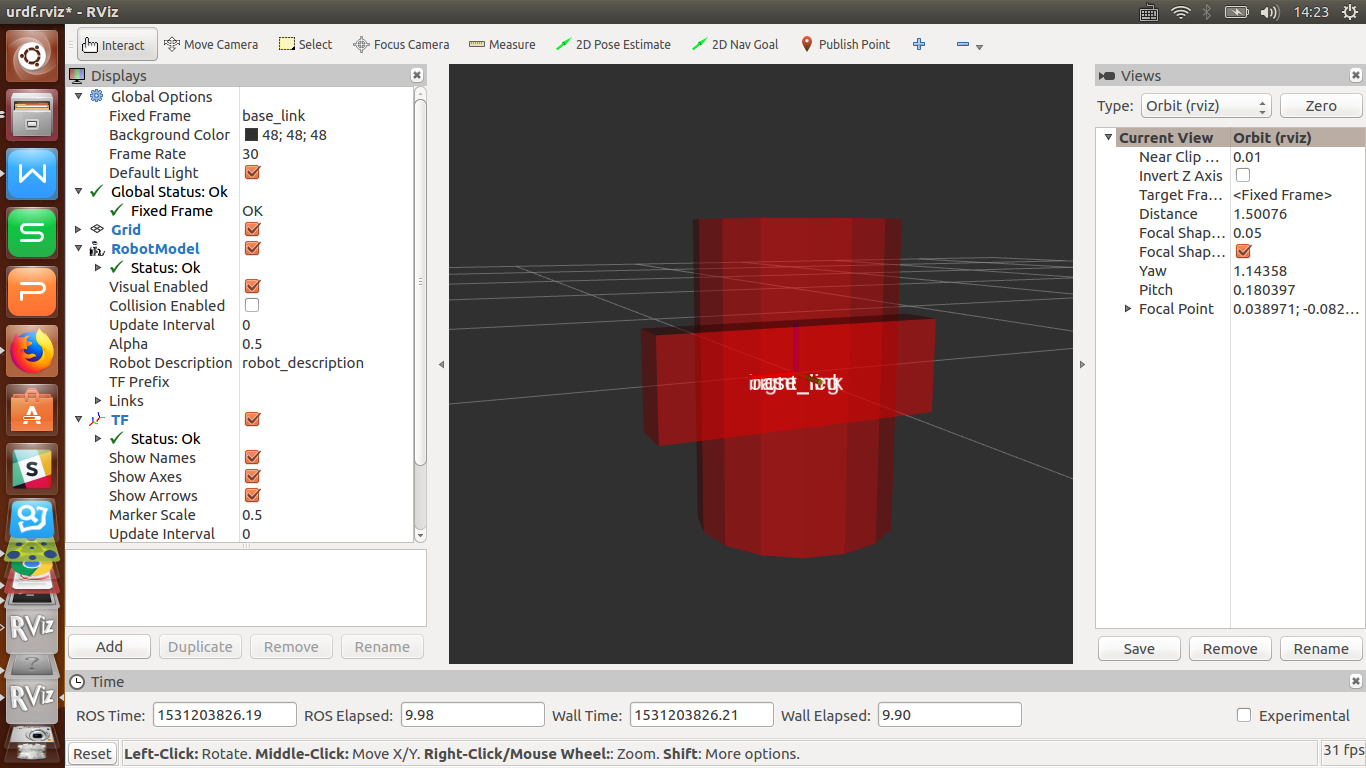
补充说明：URDF其实是只有一个root link的树状结构，这就意味着right\_leg的位姿，其实是依赖于父辈base\_link的姿态的，TF部分示意图如下：



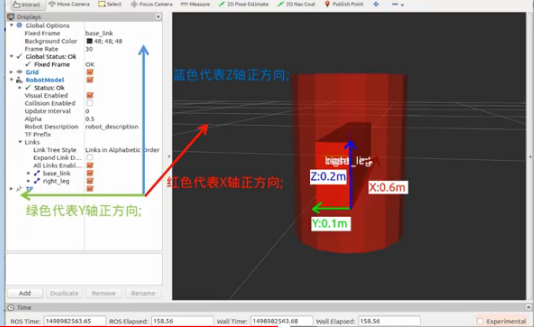
增加一个固定关节的命令如下：



仿真图如下：

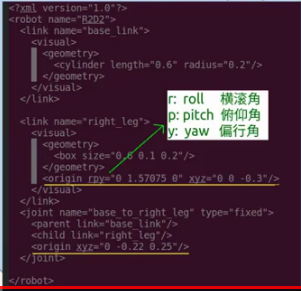


相关示意图如下：

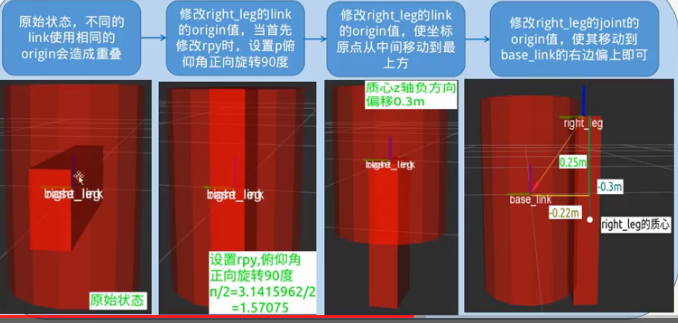


**第二步操作：**

从rviz中可以看出定义的两个形状现在重叠在一起，那是因为这两个link有相同的坐标原点，如果不想从这两个link重叠在一块，就必须为right\_leg定义不同的坐标原点，即设定不同的origin属性值。相关设置见下图代码和解释：



这里，把增加手臂的过程分解为下面图中的四个步骤，见图：

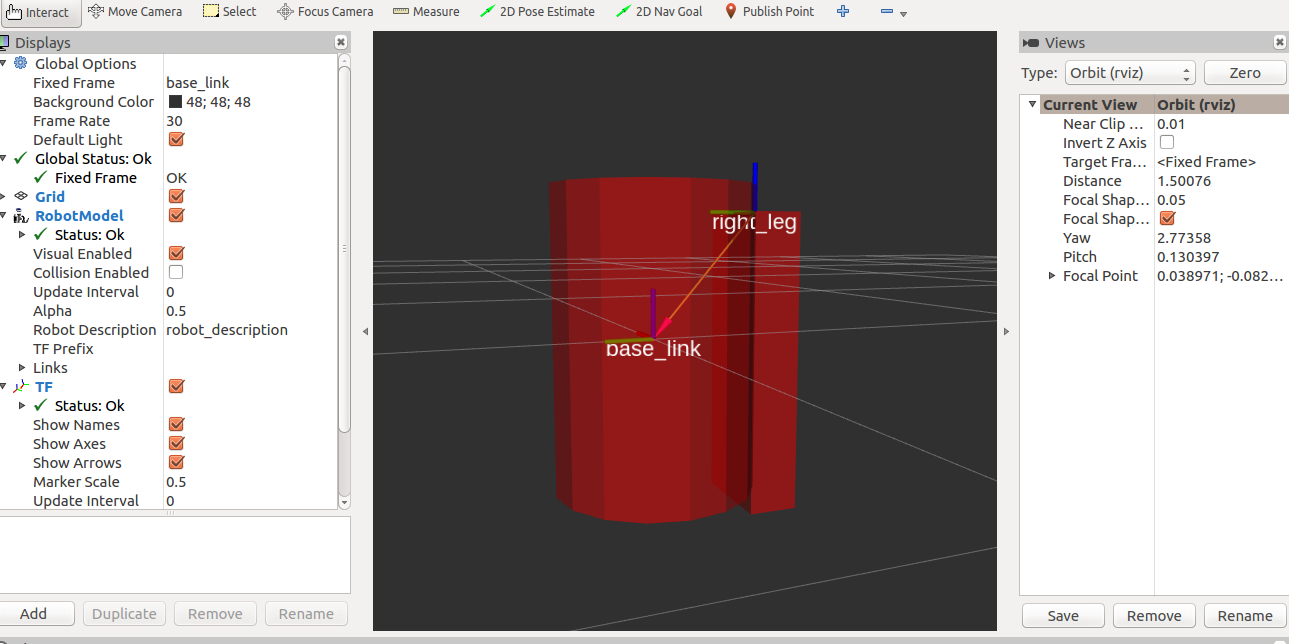


最后一个图，大家可以在终端输入命令：

roslaunch urdf\_tutorial display.launch model:=03-origins.urdf

kk

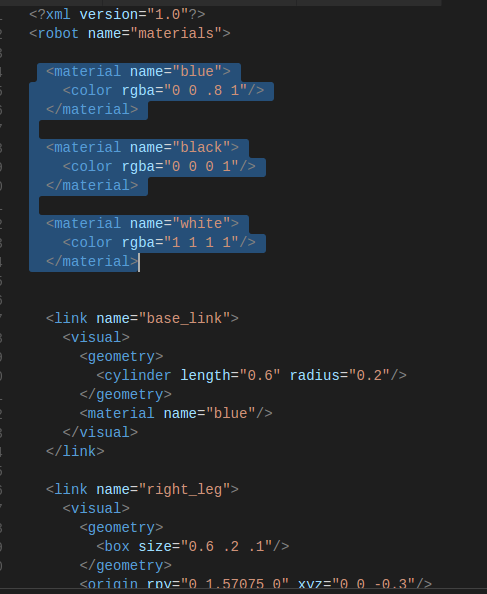
结果如下：



补充说明：launch文件运行后，会根据机器人模型URDF文件中的每一个link创建一个tf坐标，Rviz将根据这些tf坐标信息来判断每一个link该如何显示。

**第三步操作：**

我们接下来继续完善该模型，增加R2D2的左腿而且修改一下颜色，因为默认的颜色是红色。并且在原来代码的基础上增加left\_leg的link和joint。

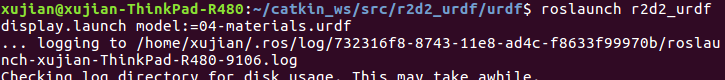


上图代码中，为link增加颜色的宏定义，方便后面多次调用，减少代码量且方便代码维护。

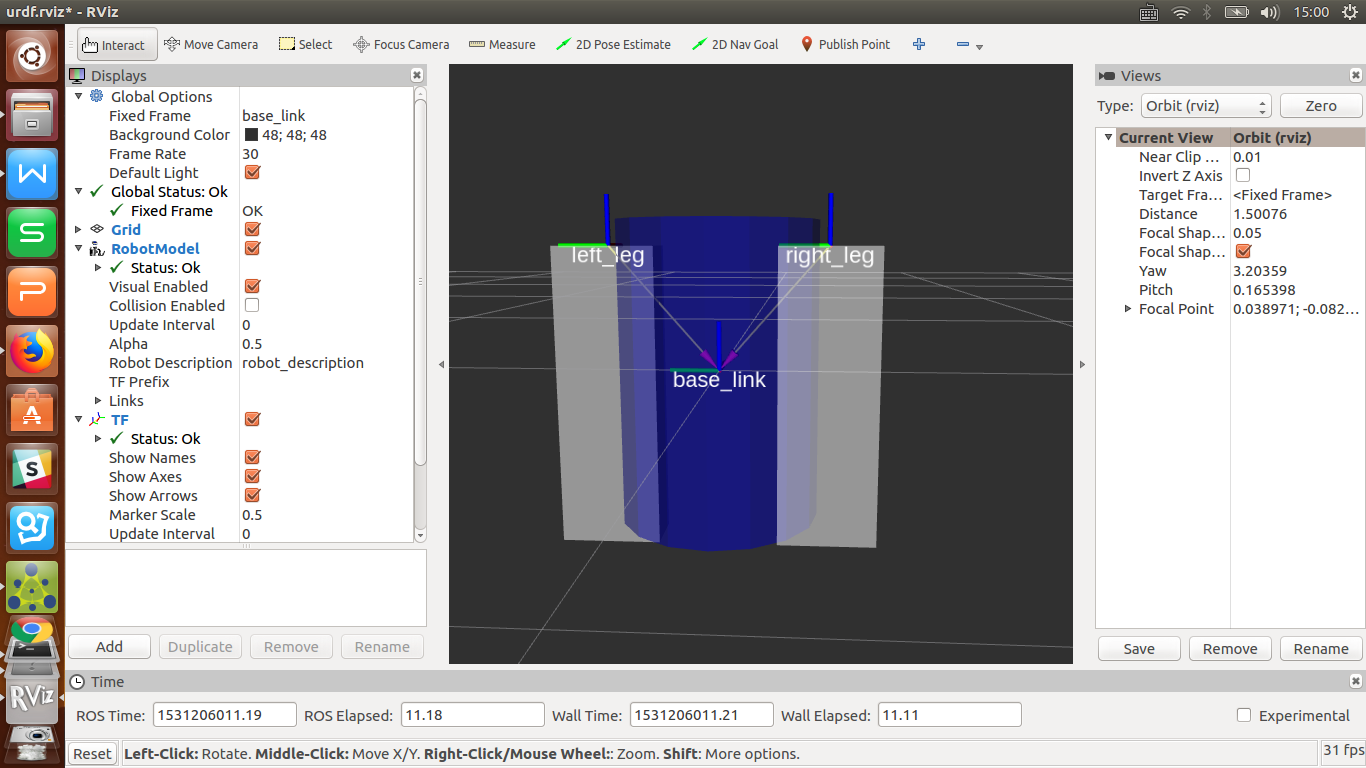
RGB值，最后一个是透明度：1代表黑色。0代表白色。

在终端输入：

roslaunch urdf\_tutorial display.launch model:=04-materials.urdf

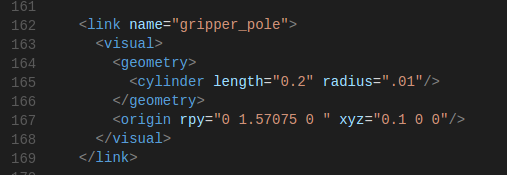


就有下面的图：



**第四步操作：**

加左右底座的代码，加底座下面的轮子，加脑袋，加眼睛，加连接杆。下面是加了一个连接杆的代码，我们可以把连接杆看成一个圆柱体：



代码注释：

该连接杆是半径为0.01米，高度为0.2米，绕着坐标轴旋转的欧拉角为<0, 1.57075, 0>，朝x轴方向平移0.1米的圆柱体。

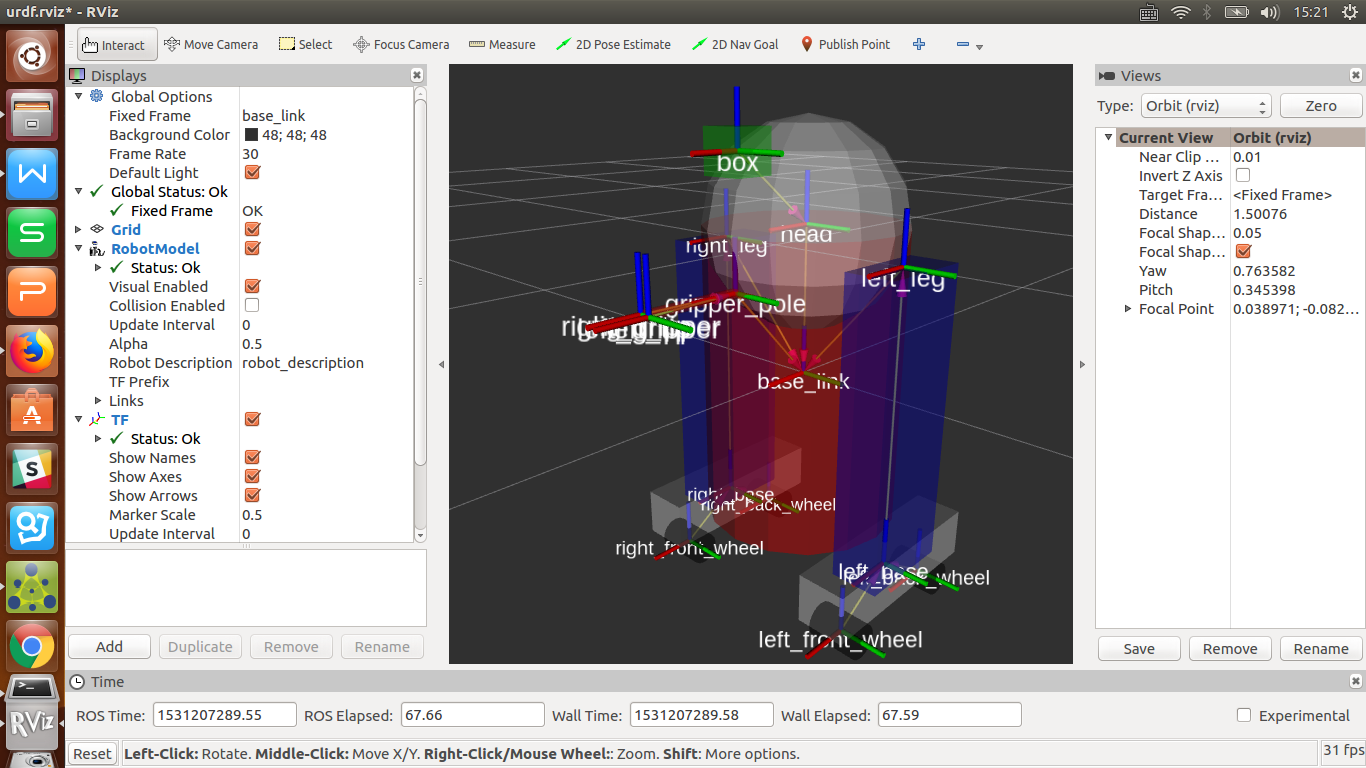
在终端输入下图的命令：

xujian

具体的代码内容：打开<源码>文件夹下的<urdf>文件夹下的

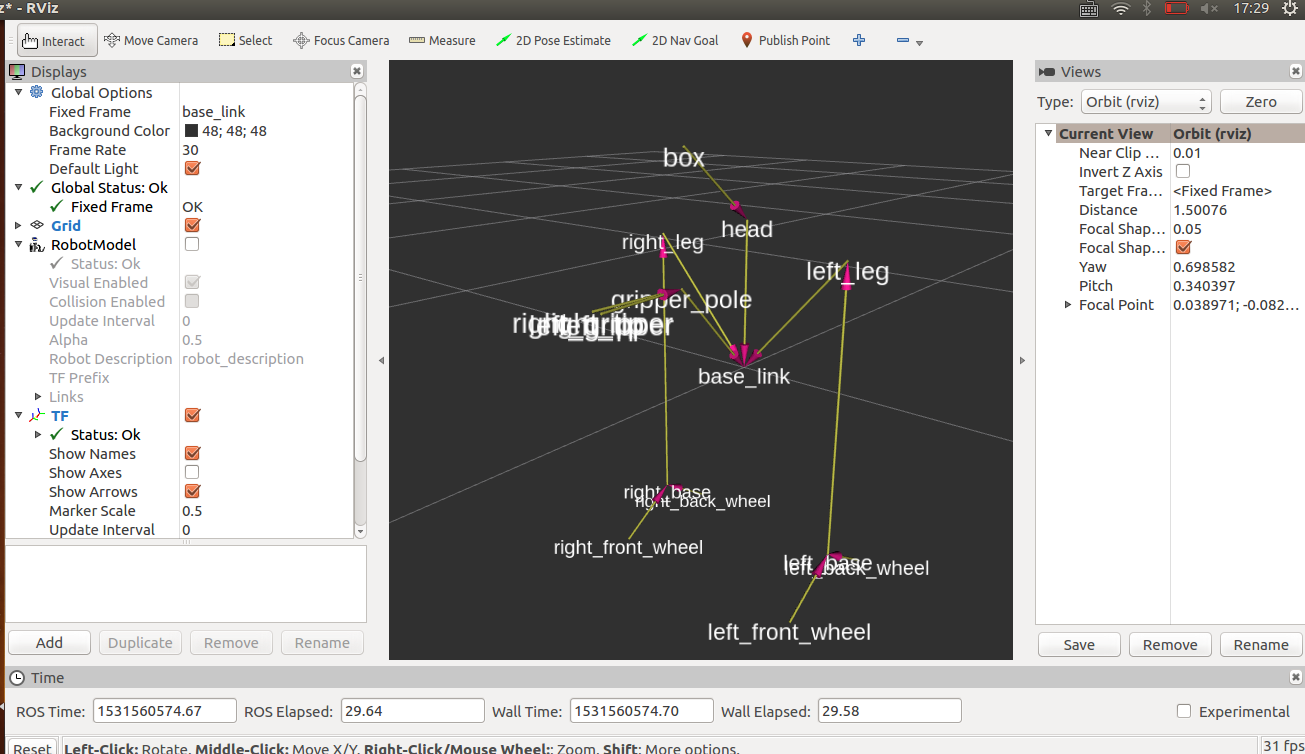
05-visual.urdf 查看

成品图如下：



最终的tf树结构在文件夹<其他素材>的下面。

去掉rviz左边的RobotModel选项，在TF选项下只勾出：Show Names 和Show arrows。通过下图，我们可以看到可视化的模型是什么样子，只不过这个框架都是固定的，不能动，下图在<其他素材>文件下的图片3能看到。后面的6.2节实训课，我们会学习添加一些物理属性和运动属性。



作业：请同学们按照老师的操作步骤，自己独立的搭建这个R2D2机器人模型，并在自己的电脑上呈现出这个机器人模型。

答案：参考上面的操作步骤。参考<源码>文件夹下的<urdf>文件夹下的05-visual.urdf。 要求学生通过实践来加深urdf文件里内容的理解。