**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：

**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。

**第二步操作：**熟悉R2-D2的外观造型。我们结合下图，看看一个R2-D2的造型是由哪些模块结合构造出来的。



**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。我们创建完pkg之后，可以用tree命令来查看最终的目录结构树。

**第四步操作：**

编写urdf代码，从最简单的部分开始，先创建R2D2的主躯干就是一个圆柱体，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。

**第五步操作：**

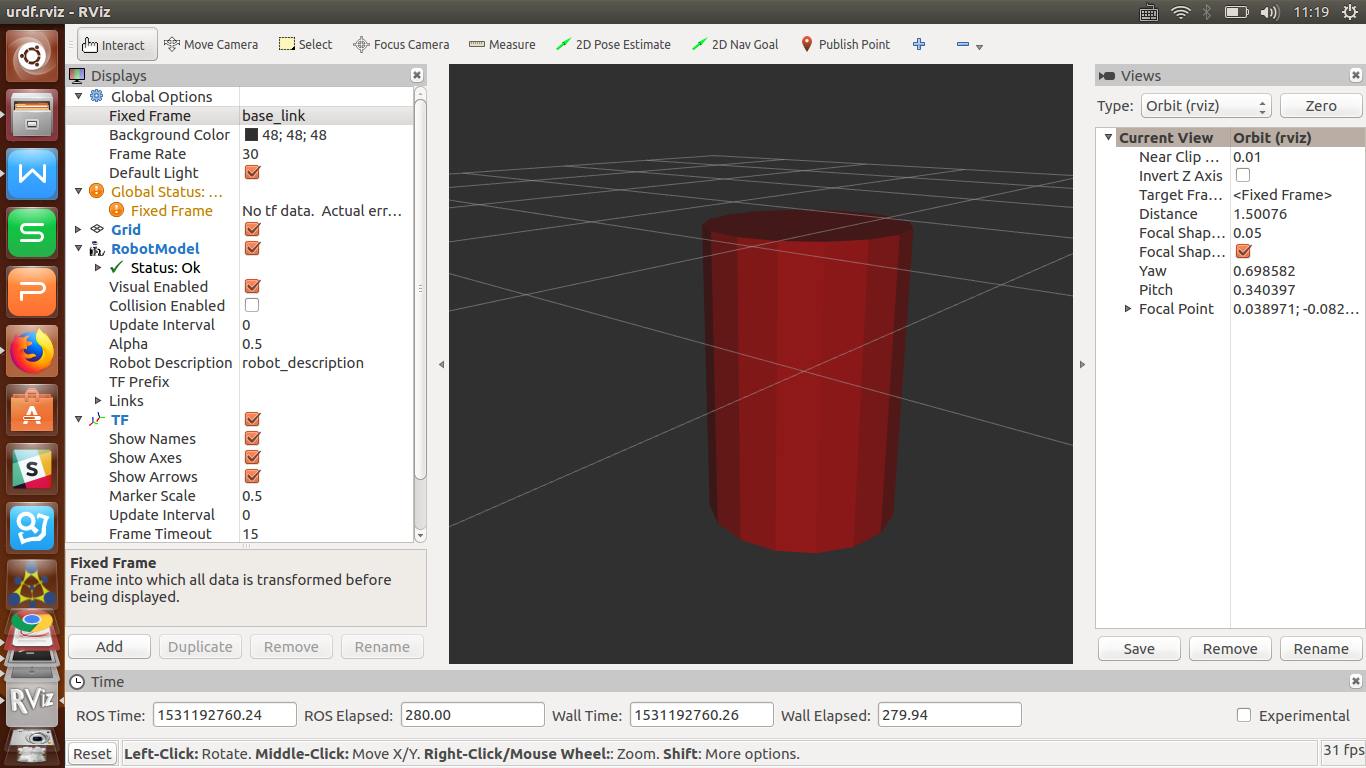
总结思考urdf\_tutorial软件包提供的display.launch是如何将我们的urdf源码文件解析显示出来的。我们来模仿urdf\_tutorial提供的display.launch文件来编写属于我们自己的launch文件，以后方便我们进行DIY。

**第六步操作：**

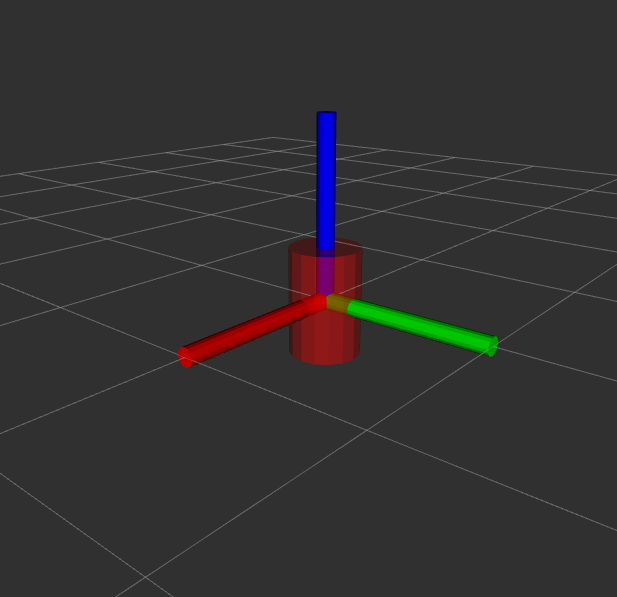
要想用修改后的launch文件，首先需要切换到catkin\_ws工作空间下，catkin\_make编译完成后，

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。

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：



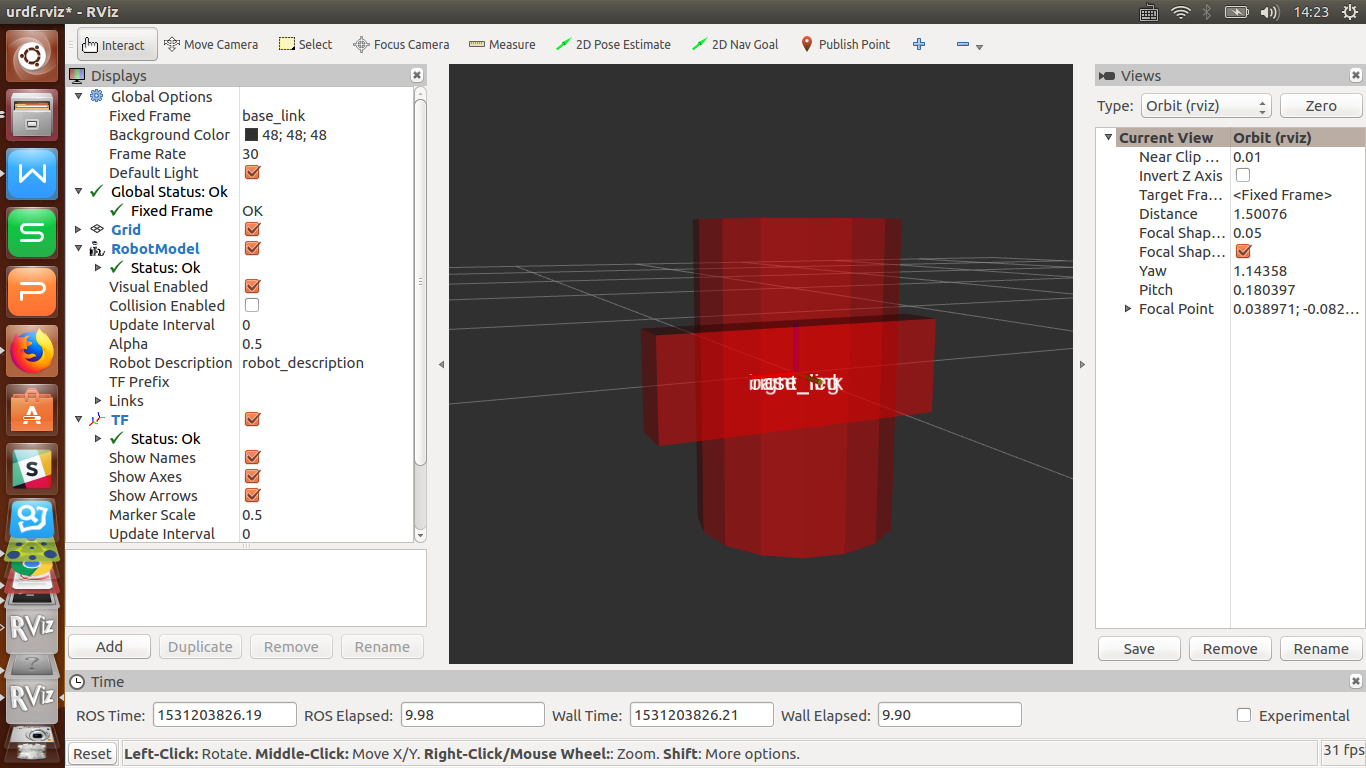
**背景知识**

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。

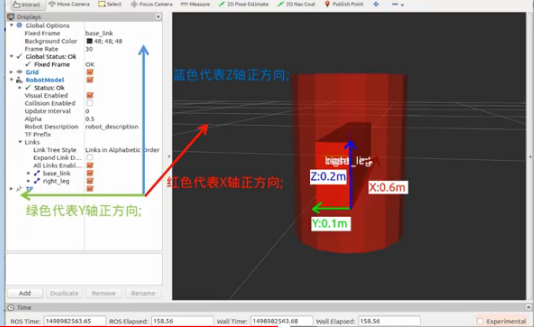
# 子任务2：

**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，仿真图如下图，同学们自己尝试给圆柱体加一个四棱柱。



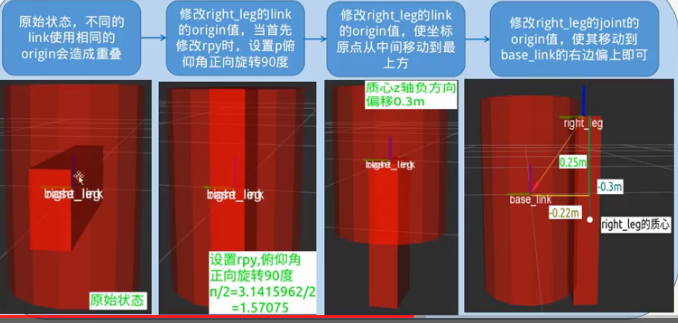
相关示意图如下：



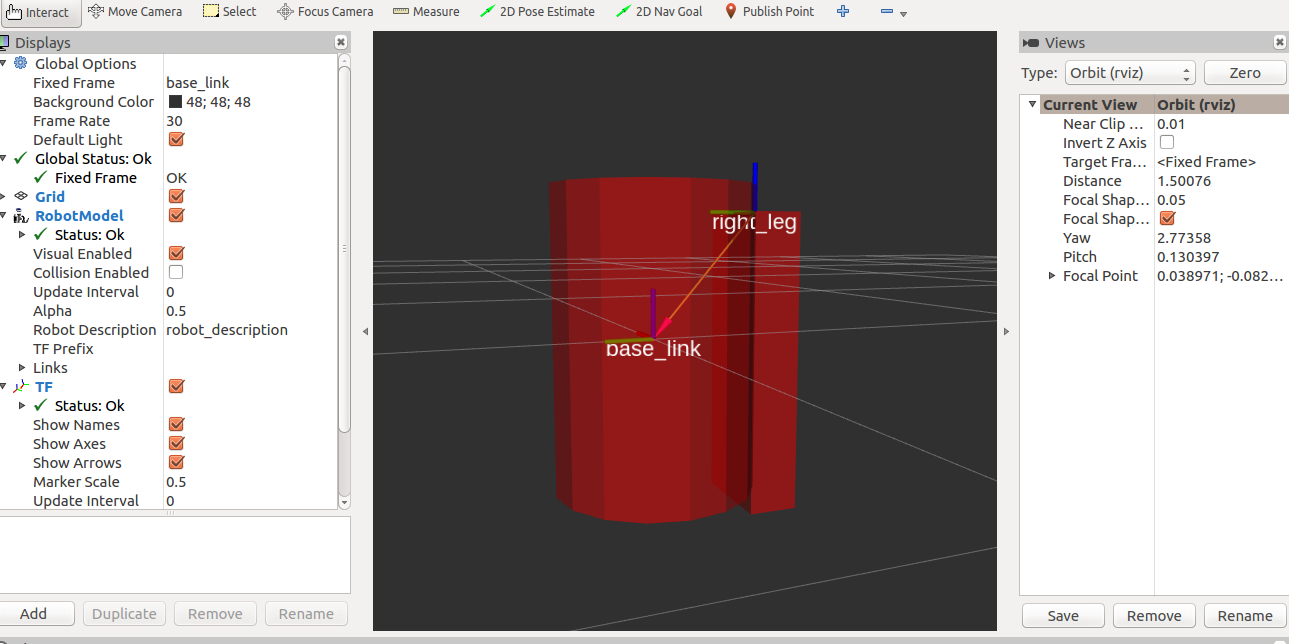
**第二步操作：**

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。

这里，把增加手臂的过程分解为下面图中的四个步骤，见图：



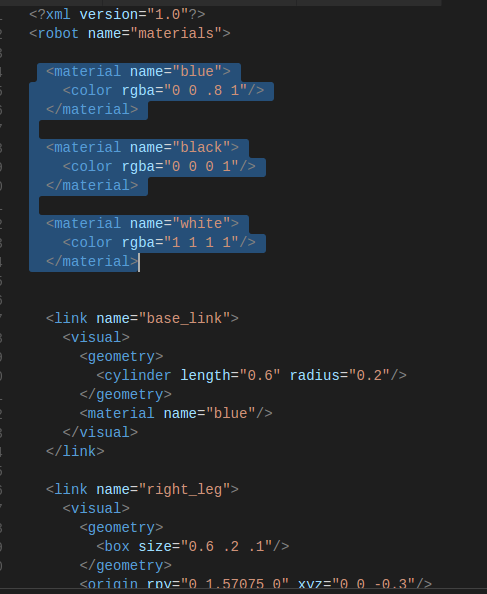
大家尝试改变xyz的值来平移，改变rpy的值来旋转，得到下图的结果：



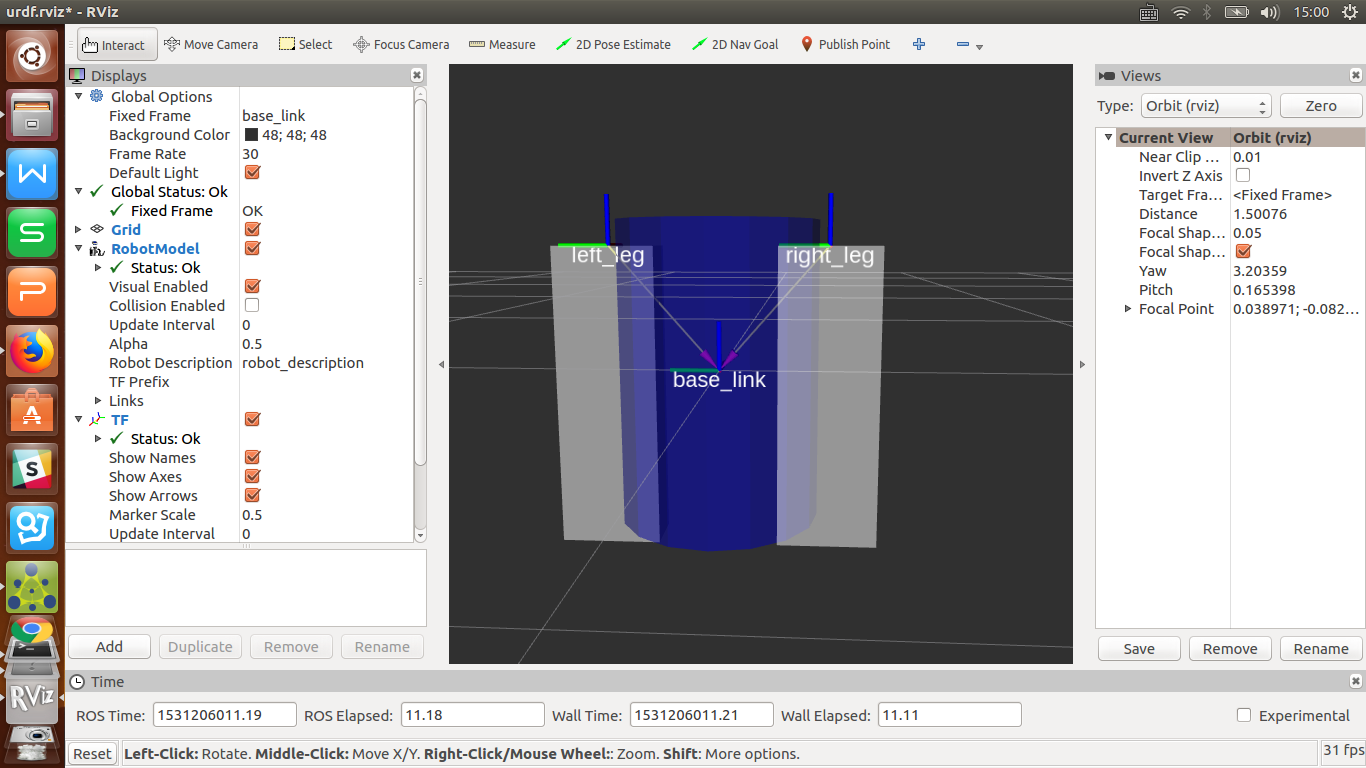
**第三步操作：**

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。在原来代码的基础上增加left\_leg的link和joint。

**提示：**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。



顺利完成以后，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。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：

