

《ROS机器人开发技术》

**课程名称:ROS机器人开发技术**

**教师姓名:徐健**

**提交时间:2018年7月x日**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程信息 | 课程名称 | | 名称 | | 章数 | 节数 | | 课程类型 |
| 《ROS机器人开发技术》 | | URDF知识学习 | | 六 | 一 | | 授课(√ )  实训( ) |
| 教师 |  | | | 时长 |  | |
| 参考文献 | 1. ROS机器人程序设计（原书第二版） [西班牙]恩里克.费尔南德斯等著 ，刘锦涛 等译 | | | | | | | |
| 教学  目的  要求 | 掌握：使用统一机器人描述格式 | | | | | | | |
| 教学  重点  难点 | 重难点 | | | PPT页面 | | | 时间分配 | |
| 重点 | 1. URDF 介绍 | | 8~17页 | | | 15分钟 | |
| 1. URDF 文件 | | 18~21页 | | | 10分钟 | |
| 1. 制作URDF模型 | | 22~31页 | | | 20分钟 | |
| 难点 | 1. URDF 文件 | | 18~21页 | | | 10分钟 | |
| 1. 制作URDF模型 | | 22~31页 | | | 20分钟 | |
| 教学方法 | 本授课以课堂讲授为主，与课堂演示方式相结合 | | | | | | | |

|  |  |  |
| --- | --- | --- |
| 教学内容 | 操作演示 | 知识点 |
| **PPT第1页:**  大家好，我是徐健，今天由我来带领大家学习统一机器人描述格式的内容。 |  |  |
| **PPT第２页:**  URDF指的是统一机器人描述格式，里面会定义具体的一些机器人的模型，而URDF会反应出机器人的关节和链接关系。今天，我们将带领大家定义具体的一些机器人的模型，直观的了解机器人的关节和链接关系。 |  |  |
| **PPT第３页:**  我们的教学内容主要分成3个部分：  6.2.1 URDF 介绍  6.2.2 URDF 文件  6.2.3 制作URDF模型 |  |  |
| **PPT第４页:**  首先是6.2.1部分的介绍，让同学们来了解这一部分的内容。 |  |  |
| **PPT第５页:**  图片是pr2机器人插插座的例子，那他和URDF有什么联系呢？  首先：URDF(Unified Robot Description Format), 是一种特殊的xml文件格式, 作为机器人的一种描述文件, 在ROS里面大量使用. 接触ROS比较久的同学, 会经常见到一种类似命名的包(package) – xxx\_description。  这个包里面就是包含某个机器人的描述文件。比如：pr2\_decription, baxter\_descrition, 以及ur\_description等，这里举例的机器人描述包都是可以通过apt-get的方式进行安装。使用命令格式如下:  sudo apt-get install ros-indigo-pr2-descrition  其中indigo是所安装的ROS版本名, 后面就是所需要安装的包名, 下划线用中画线代替。  《其他素材》文件夹里面的图片1和图片2就是自己在电脑上安装好的过程图。 | 操作演示： 打开终端，在终端编译代码。  $ sudo apt-get install ros-kinetic-pr2-descrition  图片1  可以在我们的文件下找到对应的文件：  **图片2** |  |
| **PPT第６页：**  右图是xbot机器人的路径规划图， 如果想使用开源库moveit对机器人进行路径规划, 在moveit setup assistant教程中, 第一步就是关于如何将机器人模型导入进来, 导入的机器人模型就是urdf(导入xacro格式时也是先将其解析为urdf)  Xacro（XML Macros）Xacro是一种XML宏语言。 使用xacro，您可以通过使用宏命令构建更精悍短小但又具有更高可读性的XML文件，这种宏命令可以扩展达到更大的XML表达范围。 xacro文件是urdf文件的进阶版，可以通过宏定义，文件包含来精简模型文件。还可以通过定义常量、变量等来反复调用，相当有用。  xacro文件, 在使用时, 均先会调用rosrun xacro xacro.py xxx.urdf.xacro > xxx.urdf, 将其解析成对应的urdf文件. 然后再使用。  **PPT第７页：**  机器人URDF模型主要由两个文件组成：  .xacro 是主文件，包含URDF项，包括关节，连杆，是机器人的外形构造；  .gazebo包含gazebo的具体信息以便在gazebo中仿真，包括机器人的运动属性和物理属性等。  PPT中图片是在gazebo仿真的机器人，它具有外形和物理运动等属性。 |  |  |
| **PPT第８页：**  在ppt中，上图是gazebo官网。  对机器人使用gazebo进行仿真时, 需要加载的机器人模型就是urdf模型, 当然, 单纯的urdf是不能精确描述机器人以及所需要仿真的世界的。  gazebo对其进行了扩展, 感兴趣的同学可以查看gazebo官网的一些教程. 其中会提供一些标签, 对系统动态, 重心等的设定。 |  |  |
| **PPT第９页:**  事实上，ROS wiki是一个学习的地方，大家可以浏览相关网站。http://wiki.ros.org/  URDF语法规范，参考链接:http://wiki.ros.org/urdf/XML。 |  |  |
| **PPT第10页:**  那同学们问URDF组件具体哪些呢？这里的图片给出了答案，它是由不同的功能包和组件组成: 其中urdf\_parser和urder\_interface已经在hydro之后的版本中去除了。urdf\_paser\_plugin是URDF基础的插件,衍生出了urdfdom(面向URDF文件)和collar\_parser(面向相互文件)。这里，大家简单了解一下即可。 |  |  |
| **PPT第11页:**  左边是Xbot机器人的模型，它并不是一个固定的完整的模型，它主要由两部分组成，  一部分是link，一部分是joint。所谓link就是部件、零件，每个link都有xyz三轴，就是这个rgb三个轴，joint是连接两个link的关节  大家应该还记得我们之前讲的tf，tf是一棵树型的数据结构  tf就反应了它运行时的link和joint的关系，tf反映的是它当前的状态，  而link的属性和joint的属性都定义在了urdf文件里，  Urdf文件和我们之前见到的launch文件一样，也是遵循着xml标签语言语法。 |  |  |
| **PPT第12页:**  在urdf文件里有两个最重要的标签就是link和joint，在《其他素材》的文件夹下面，机器人的树形结构，就是通过这个文件定义的，在我们这里例子里面，首先定义了这些机构，base link ， laser link ， imu link，还有连接这些机构的关节joint，从base footprint到base link，base\_link到左右轮。这里的代码大家简单了解即可，后面有详细的介绍。 | **相关操作：**  在ppt图中放大代码的图片，让学生看的更清楚：  图片包含 文字  已生成高可信度的说明 |  |
| PPT第13页: Link和joint标签还有自己的一套属性，大家可以结合ppt看一下：其中的一些形状等外观属性可以在rviz中体现，物理碰撞等属性在gazebo中能体现，后面的实训课会有相关介绍。 |  |  |
| PPT第14页: 我们接着开始6.2.2节的学习： URDF 文件 |  |  |
| PPT第15页: 提问：大家可以想想小时候，这种玩具小车，是如何通过组件拼装的？  在URDF当中,当你想要描述一个机器人的时候，例如小车的base\_link和右轮right，两个link之间需要joint来连接。 PPT第16页: link 和 joint：一般, 机器人都是由link和joint进行描述. 都会呈现为树状(想象数据结构里面的树), 如下图, 由一个根link(link1)向上, 分别出现了两个分支–link2和-link3, 分别由joint连接link. link4就可以类似的理解。  这里的三维坐标系使用的是右手坐标系，右上角的图是tf树：xyz：5 3 0，表示x轴平移5个单位，y轴平移3个单位，z轴平移0个单位。rpy表示旋转的欧拉角对应的三个分量。 |  |  |
| PPT第17页: 问题：大家根据上一页的PPT，想象的出来机器人大概是一个什么样子? 每个link长什么样子? 他们又是一个什么样子的位置关系?  答：每一个不同的同学, 所联想到的样子都是不一样的, 一个仙人掌? 一个两指的夹持器? 当然, urdf文件是不能够被正确解析的, 也是不能够可视化出来的. 但上面内容就类似于整个机器人的骨架, 机器人就由这些东西组成. 一共拥有4个link, 和3个joint, 像图中所示的样子连接起来。 |  |  |
| PPT第18页: 我们来制作一个URDF的实例来实践这次的授课内容 |  |  |
| **PPT第19页：**  在URDF当中,当你想要描述一个机器人的时候,例如小车的base\_link和右轮right,两个link之间需要joint来连接。  (1)添加基本模型  我们以构建一个小车为例子,为大家讲解这部分内容：(相关的示例代码可以从我们的urdf\_demo中找到)，我们的想法是：首先构建base\_link作为小车的父坐标系,然后在  base\_link基础上,再构建左轮,右轮和雷达link. 最后不同的link之间通过joint来连接。大家来看一下PPT中的参考代码的内容。  相关代码注释：  1行：机器人的名字为 “mycar”  2-5行：设置了4个link，名字分别为：“base\_link”、“right”、“left”、”rplidar”，其中base\_link作为小车的父坐标系。  7-10行：连接名为“右连接”，类型是可连续的，父link是base\_link，子link是right，下面代码的类似。 | **相关操作：**  在RoBoWare里面输入下面图中的代码，并且逐行解释。  55 |  |
| PPT第20页: 小技巧:  sudo apt-get install liburdfdom-tools，安装完毕后,执行检查check\_urdf my\_car.urdf。如果一切正常,就会有PPT图中的显示。 | **相关操作：**   1. 在终端输入命令：   sudo apt-get install liburdfdom-tools   1. 在终端切入到catkin\_ws/src/urdf\_demo/urdf的文件夹 2. 执行下列命令：   $ check\_urdf my\_car.urdf  xu |  |
| PPT第21页: 随后打开新的终端,输入：  roslaunch urdf\_demo display\_urdf\_link\_joint.launch  回车之后，发现所有的link和joint都在一起了，见PPT图。 | **相关操作：**  1、打开终端  2、在终端输入下列指令：  $ roslaunch urdf\_demo display\_urdf\_link\_joint.launch  2018-07-17 20-02-39屏幕截图 |  |
| PPT第22页: (2)添加机器人link之间的相对位置关系  在基础模型之上,我们需要为机器人之间link来设相对位置和朝向的关系,URDF中通过  <origin>来描述这种关系。  相关代码注释：  Origin中的rpy的三个分量，指的是欧拉角的三个分量，xyz指的是相对base\_link的位移情况。  **PPT第23页:**  随后打开新的终端,输入：  roslaunch urdf\_demo display\_urdf\_link\_position.launch  回车之后,发现所有的link和joint已经有在固定的位置上了。 | **相关操作：**  1、打开终端  2、在终端输入下列指令：  $ roslaunch urdf\_demo display\_urdf\_link\_position.launch  **shixi** |  |
| PPT第24页: (3)添加模型的尺寸,形状和颜色等  在已经设置好模型的link基础上,添加模型的形状(例如圆柱或长方体),相对于link的位  置,颜色等。其中形状用<geometry>来描述,颜色用<color>来描述。  相关代码注释：  这里的<cylinder length=” .06” radius=”0.27”></cylinder> 指的是使用圆柱体，半径为0.27,米，高为0.06米  Material name 和 color rgba 一致，rgba代表颜色的四个通道，四个不同通道的值代表不同的颜色。 |  |  |
| PPT第25页: (4)显示URDF模型  想要在rviz中显示出我们制作好的小车的URDF模型,可以写一个launch文件。  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8-10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  档都停止执行、关闭。 | 相关操作： 在RoBoWare输入下面的代码，并且逐行讲解：  111 |  |
| PPT第26页: 问题：我们之前每次运行出现的下图在launch文件哪里能够体现出来？  答：除了launch文件中的前3句,导入我们制作小车的URDF模型外,还需要添加  joint\_state\_publisher和robot\_state\_publisjer这两个节点，在launch的第8行和第9行。 |  |  |
| **PPT第27页:**  效果如ppt图。我们可以将launch文件中的param name="use\_gui"的值由false改成true会弹出一个窗口，移动进度条,可以临时改变joint的朝向。 | **相关操作：**  1、打开终端  2、在终端输入下列指令：  $ roslaunch urdf\_demo display\_urdf\_color\_geometry.launch  3、找到urdf\_demo文件夹下的launch文件夹下的display\_urdf\_color\_geometry.launch 文件，用IDE打开   1. 修改param name="use\_gui"的值由true改成false，观察之前弹出的窗口是否再次出现。   hehe pp |  |
| PPT第28页: 另外，我们可以输另入rosrun rqt\_tf\_tree rqt\_tf\_tree,可以看到tf树。如图。 | **相关操作：**  1、新打开一个终端  2、在终端输入下列指令：  $ rosrun rqt\_tf\_tree rqt\_tf\_tree ii |  |
| PPT第29页: 这里展示的是我们的xbot模型，那这个xbot模型，实际上也是借助这个package能够直接在rviz中显示的 | **相关操作：**   1. 在robot\_sim\_demo查找robot.xacro文件 2. 打开终端输入：   $ roscd urdf\_sim\_demo  $ cd urdf  $ ls  $ cp robot.xacro /home/xujian/catkin\_ws/src/urdf\_demo/urdf  3、修改launch文件，参考<源码>文件夹下的display\_xbot.launch  ４、在终端输入，roslaunch urdf\_demo display\_xbot.launch |  |
| PPT第30页: 这个图，是去掉坐标轴和名称的效果图。 | **相关操作：** 去掉Rviz中TF中的axis和name |  |
| PPT第31页: 我们的第一节授课就讲到这里，谢谢。 |  |  |
| **知识点总结：**   1. URDF介绍：理解URDF，知道URDF可以构造机器人模型 2. URDF文件，掌握link和joint的作用   3、制作URDF模型：搭建小车模型，进一步理解link和joint |  |  |
|  |  |  |
|  | | |