**实验项目三 ROS的机器人建模**

1. **实验目的**

1、 掌握URDF文件格式及基本概念；

2、 掌握ROS机器人建模方法。

1. **实验设备**

1、 雷达模块一台；

2、 无线键鼠一套；

3、 显示器一套。

1. **实验步骤和内容**

URDF是Unified Robot Description Format的首字母缩写，直译为统一（标准化）机器人描述格式，可以以一种XML的方式描述机器人的部分结构，比如底盘、摄像头、激光雷达、机械臂以及不同关节的自由度,该文件可以被C++内置的解释器转换成可视化的机器人模型，是ROS中实现机器人仿真的重要组件。

URDF文件是一个标准的XML文件，在ROS中预定义了一系列的标签用于描述机器人模型，机器人模型可能较为复杂，但是ROS的URDF中机器人的组成却是较为简单，可以主要简化为两部分：连杆（link标签）与 关节（joint标签）：

* robot 根标签，类似于launch文件中的launch标签
* link 连杆标签
* joint 关节标签
* gazebo 集成gazebo需要使用的标签

关于gazebo标签，后期在使用gazebo仿真时，才需要使用到，用于配置仿真环境所需参数，比如：机器人材料属性、gazebo插件等，但是该标签不是机器人模型必须的，只有在仿真时才需设置。

**另请参考：**<https://wiki.ros.org/urdf/XML>

1. **UDRF的建模实现**

**需求描述：**

创建一个四轮圆柱状机器人模型，机器人参数如下，底盘为圆柱状，半径10cm，高8cm，四轮由两个驱动轮和两个万向支撑轮组成，两个驱动轮半径为3.25cm，轮胎宽度1.5cm，两个万向轮为球状，半径0.75cm，底盘离地间距为1.5cm（与万向轮直径一致）。

**实现流程：**

创建机器人模型可以分步骤实现：

**（1）新建urdf以及launch文件**

urdf文件：基本实现参考test\_02中urdf下的robot.urdf文件。

<robot name="mycar">

<!-- 设置 base\_footprint -->

<link name="base\_footprint">

<visual>

<geometry>

<sphere radius="0.001" />

</geometry>

</visual>

</link>

<!-- 添加底盘 -->

<!-- 添加驱动轮 -->

<!-- 添加万向轮(支撑轮) -->

</robot>

launch文件（test\_02下launch文件夹里的robot.launch文件）：

<launch>

<!-- 将 urdf 文件内容设置进参数服务器 -->

<param name="robot\_description" textfile="$(find test\_02)/urdf/robot.urdf" />

<!-- 启动 rivz -->

<node pkg="rviz" type="rviz" name="rviz" args="-d $(find test\_02)/config/robot.rviz" />

<!-- 启动机器人状态和关节状态发布节点 -->

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

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

<!-- 启动图形化的控制关节运动节点 -->

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

</launch>

**（2）底盘搭建**

<link name="base\_link">

<visual>

<geometry>

<cylinder radius="0.1" length="0.08" />

</geometry>

<origin xyz="0 0 0" rpy="0 0 0" />

<material name="yellow">

<color rgba="0.8 0.3 0.1 1.0" />

</material>

</visual>

</link>

<joint name="base\_link\_to\_base\_footprint" type="fixed">

<parent link="base\_footprint" />

<child link="base\_link"/>

<origin xyz="0 0 0.055" />

</joint>

**（3）添加驱动轮**

<link name="left\_wheel">

<visual>

<geometry>

<cylinder radius="0.0325" length="0.015" />

</geometry>

<origin xyz="0 0 0" rpy="1.57 0 0" />

<material name="black">

<color rgba="0.0 0.0 0.0 1.0" />

</material>

</visual>

</link>

<joint name="left\_wheel\_to\_base\_link" type="continuous">

<parent link="base\_link" />

<child link="left\_wheel" />

<origin xyz="0 0.1 -0.0225" />

<axis xyz="0 1 0" />

</joint>

<link name="right\_wheel">

<visual>

<geometry>

<cylinder radius="0.0325" length="0.015" />

</geometry>

<origin xyz="0 0 0" rpy="1.57 0 0" />

<material name="black">

<color rgba="0.0 0.0 0.0 1.0" />

</material>

</visual>

</link>

<joint name="right\_wheel\_to\_base\_link" type="continuous">

<parent link="base\_link" />

<child link="right\_wheel" />

<origin xyz="0 -0.1 -0.0225" />

<axis xyz="0 1 0" />

</joint>

**（4）添加万向轮**

<link name="front\_wheel">

<visual>

<geometry>

<sphere radius="0.0075" />

</geometry>

<origin xyz="0 0 0" rpy="0 0 0" />

<material name="black">

<color rgba="0.0 0.0 0.0 1.0" />

</material>

</visual>

</link>

<joint name="front\_wheel\_to\_base\_link" type="continuous">

<parent link="base\_link" />

<child link="front\_wheel" />

<origin xyz="0.0925 0 -0.0475" />

<axis xyz="1 1 1" />

</joint>

<link name="back\_wheel">

<visual>

<geometry>

<sphere radius="0.0075" />

</geometry>

<origin xyz="0 0 0" rpy="0 0 0" />

<material name="black">

<color rgba="0.0 0.0 0.0 1.0" />

</material>

</visual>

</link>

<joint name="back\_wheel\_to\_base\_link" type="continuous">

<parent link="base\_link" />

<child link="back\_wheel" />

<origin xyz="-0.0925 0 -0.0475" />

<axis xyz="1 1 1" />

</joint>

1. **URDF优化Xacro**

**需求描述：**

使用xacro优化上一节案例中驱动轮实现，需要使用变量封装底盘的半径、高度，使用数学公式动态计算底盘的关节点坐标，使用Xacro宏封装轮子重复的代码并调用宏创建两个轮子（注意：在此，演示Xacro的基本使用，不必要生成合法的URDF）。

**实现流程：**

**利用xacro优化上述机器人模型可以分步骤实现：**

**（1）Xacro文件编写**

编写Xacro文件，以变量的方式封装属性（常量半径、高度、车轮半径...），以函数的方式封装重复实现（车轮的添加）。

<robot name="mycar" xmlns:xacro="http://wiki.ros.org/xacro">

<!-- 属性封装 -->

<xacro:property name="wheel\_radius" value="0.0325" />

<xacro:property name="wheel\_length" value="0.0015" />

<xacro:property name="PI" value="3.1415927" />

<xacro:property name="base\_link\_length" value="0.08" />

<xacro:property name="lidi\_space" value="0.015" />

<!-- 宏 -->

<xacro:macro name="wheel\_func" params="wheel\_name flag" >

<link name="${wheel\_name}\_wheel">

<visual>

<geometry>

<cylinder radius="${wheel\_radius}" length="${wheel\_length}" />

</geometry>

<origin xyz="0 0 0" rpy="${PI / 2} 0 0" />

<material name="wheel\_color">

<color rgba="0 0 0 0.3" />

</material>

</visual>

</link>

<joint name="${wheel\_name}\_to\_link" type="continuous">

<parent link="base\_link" />

<child link="${wheel\_name}\_wheel" />

<!--

x 无偏移

y 车体半径

z z= 车体高度 / 2 + 离地间距 - 车轮半径

-->

<origin xyz="0 ${0.1 \* flag} ${(base\_link\_length / 2 + lidi\_space - wheel\_radius) \* -1}" rpy="0 0 0" />

<axis xyz="0 1 0" />

</joint>

</xacro:macro>

<!-- 调用语法 -->

<xacro:wheel\_func wheel\_name="left" flag="1" />

<xacro:wheel\_func wheel\_name="right" flag="-1" />

</robot>

**（2）Xacro文件转换成urdf文件**

命令行进入xacro文件所属目录，执行：rosrun xacro xacro xxx.xacro > xxx.urdf，会将xacro文件解析为urdf文件，内容如下：

<?xml version="1.0" encoding="utf-8"?>

<!-- =================================================================================== -->

<!-- | This document was autogenerated by xacro from demo01\_helloworld.urdf.xacro | -->

<!-- | EDITING THIS FILE BY HAND IS NOT RECOMMENDED | -->

<!-- =================================================================================== -->

<robot name="mycar">

<link name="left\_wheel">

<visual>

<geometry>

<cylinder length="0.0015" radius="0.0325"/>

</geometry>

<origin rpy="1.57079635 0 0" xyz="0 0 0"/>

<material name="wheel\_color">

<color rgba="0 0 0 0.3"/>

</material>

</visual>

</link>

<joint name="left\_to\_link" type="continuous">

<parent link="base\_link"/>

<child link="left\_wheel"/>

<!--

x 无偏移

y 车体半径

z z= 车体高度 / 2 + 离地间距 - 车轮半径

-->

<origin rpy="0 0 0" xyz="0 0.1 -0.0225"/>

<axis xyz="0 1 0"/>

</joint>

<link name="right\_wheel">

<visual>

<geometry>

<cylinder length="0.0015" radius="0.0325"/>

</geometry>

<origin rpy="1.57079635 0 0" xyz="0 0 0"/>

<material name="wheel\_color">

<color rgba="0 0 0 0.3"/>

</material>

</visual>

</link>

<joint name="right\_to\_link" type="continuous">

<parent link="base\_link"/>

<child link="right\_wheel"/>

<!--

x 无偏移

y 车体半径

z z= 车体高度 / 2 + 离地间距 - 车轮半径

-->

<origin rpy="0 0 0" xyz="0 -0.1 -0.0225"/>

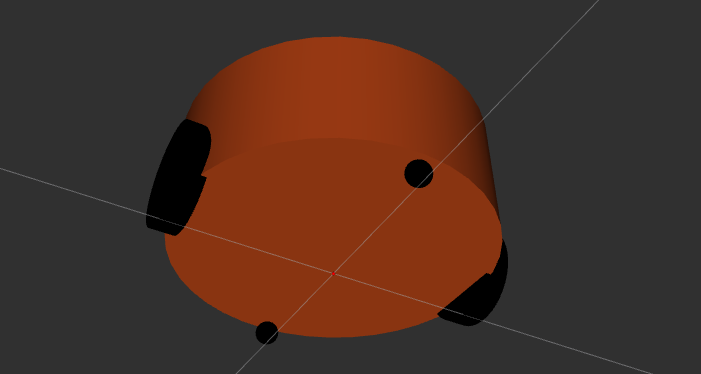
<axis xyz="0 1 0"/>

</joint>

</robot>

注意：该案例编写生成的是非法的URDF文件，目的在于演示Xacro的极简使用以及优点。

1. **实验参考结果**



1. **相关的测试代码**

见代码包test\_02。