《ROS机器人开发技术》

课程讲稿

|  |  |  |
| --- | --- | --- |
| 课程名称 | ： | ROS机器人开发技术 |
| 教师姓名 | ： |  |
| 提交时间 | ： | 2018年9月9日 |

中国大学MOOC制

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程  名称 | 《ROS机器人开发技术》 | | 章 | | 第三章 | | | 课程类型 |
| 节 | | 第一节 | | | 授课(√ )  实训( ) |
| 名称 | 通信架构（一） | | | | | | |
| 教师 |  | 课时 | | | | 1课时 | | |
| 参考  资料 | 1. ROS wiki <http://wiki.ros.org/> 2. 《ROS机器人编程实践》 3. 《ROS机器人开发实践》 | | | | | | | |
| 教学  目的  要求 | 掌握： topic实践 | | | | | | | |
| 教学  重点  难点 |  | | | PPT页面 | | | 时间分配 | |
| 教学重点 | | | | | | | |
|  | | |  | | |  | |
|  | | |  | | |  | |
|  | | |  | | |  | |
| 教学难点 | | | | | | | |
| 1. | | |  | | |  | |
|  | | |  | | |  | |
|  |  | | |  | | |  | |

|  |  |  |
| --- | --- | --- |
| 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：**  本章我们介绍ROS的topic通信的具体内容和命令，并结合Gazebo仿真讲解具体应用。  本章是学习topic的一个重要部分，所以大家一定要认真听讲，认真对待这部分内容。  在之前的ROS下运行输出hello world的课程中我们已经运行了node，这里我们将学习node的一些具体介绍。 |  |  |
| **PPT第2页:**  这是一个提纲。分别总结了本章将要介绍到的一些概念以及要讲解的一些命令。  主要分为三个步骤。  现在先给大家看一下，让大家有一个大概的印象。 |  |  |
| **PPT第3页:**  首先，我们要进入第一步，topic&msg简介  **PPT第4页:**  节点之间最重要的通信机制就是基于发布（publish）/订阅（subscrib）模型的消息（message）通信。每一个消息都是一种严格的数据结构，支持标准数据类型（整型、浮点型、布尔型等），也支持嵌套结构和数组，还可以根据需求由开发者自主定义。 |  |  |
| **PPT第5页:**  topic在ROS中使用最为频繁。在ROS中有两个节点，一个是发布者（publisher），一个是订阅者（Subscriber），两个节点分别发布、订阅同一个topic，启动顺序没有强制要求，此处假设publisher先启动，可大致分成以下七个步骤来分析建立通信的详细过程。  ……  接下来我们通过图解的方式一步步让大家理解这七个步骤 |  |  |
| **PPT第6页:**  1.publisher注册  publisher启动，向ROS Master注册发布者的信息，包含所发布消息的话题名，ROS Master会将节点的注册信息加入到注册列表中。  2.subscriber注册  subscriber启动，同样向ROS Master注册订阅者的消息，包含需要订阅的话题名。 |  |  |
| **PPT第7页：**  3.ROS Master进行信息匹配  Master根据subscriber的订阅信息从注册列表中查找，如果没有找到匹配的发布者，则等待发布者的加入；如果找到匹配的发布者的信息，则向subscriber发送publisher的地址信息。  4.subscriber发送连接请求  subscriber接收到Master发回的subscriber的地址信息，尝试向publisher发送连接请求，传输订阅的话题名、消息类型以及通信协议。 |  |  |
| **PPT第8页：**  5.publisher确认连接请求  publisher接收到连接请求后，继续向subscriber确认连接信息，其中包含了自身的TCP地址信息。  6.subscriber尝试与publisher建立  publisher接收到确认信息后，使用TCP尝试与subscriber建立网络连接。  **PPT第9页：**  7.publisher向subscriber发送数据  成功建立连接后，publisher开始向subscriber发送话题消息数据。  subscriber接收到消息会进行处理，一般这个过程叫做回调（callback）。就是提前定义好了一个处理函数(写在代码中),当有消息来就会触发这个处理函数,函数会对消息进行处理。 |  |  |
| **PPT第10页:**  ppt上是一个topic通讯的示意图。  我们根据这个示意图再回忆一遍到底有哪七个步骤：  1.publisher注册  2.subscriber注册  3.ROS Master进行信息匹配  4.subscriber发送连接请求  5.publisher确认连接请求  6.subscriber尝试与publisher建立  7.publisher向subscriber发送数据 |  |  |
| **PPT第11页:**  消息（msg）文件就是一个描述ROS中所使用消息类型的简单文本。它们会被用来生成不同语言的源代码。msg文件存放在package的msg目录下，msg文件其实就是每行声明一个数据类型和变量名，可以使用的数据类型如下：  bool、int8、int16、int32、int64(以及uint)、float、float64、string、time、duration、header、可变长数组array[]、固定长度数组array[C]等等。ROS中还有一个特殊的数据类型：Header，含有时间戳和坐标系信息。在msg文件的第一行经常可以看到Header header的声明。  下面的图片是一个简单的msg文件的案例： |  |  |
| **PPT第12页:**  经过上面的介绍我们对topic和msg有了一定的了解，下面我们进入本章第二部分：topic&msg命令 |  |  |
| **PPT第13页:**  msg也有相对应的ros命令：  rosmsg  这个命令比较简单，只有两个子命令：  ……  我们可以通过rosmsg list命令 查看系统上所有的消息，  然后从当中随便选择一个msg运行rosmsg show msg\_name命令 |  |  |
| PPT第14页: 下面我们进入最后的一个十分重要的部分：  关于topic的命令：rostopic  这个命令的子命令比较多，需要同学们在后面的实训环节当中跟着老师多练习，才能记住每个命令的作用。 |  |  |
| PPT第15页: topic命令我们介绍的差不多了，现在我们进入最重要的一个步骤：Gazebo仿真中的topic介绍  **PPT第16页:**  第一个命令：rostopic list命令  首先我们要启动Gazebo仿真，在终端输入如下命令：  $ roslaunch robot\_sim\_demo robot\_spawn.launch  我们的仿真环境就启动了。  新建终端，输入如下命令：  $ rostopic list  这个命令让我们可以看到启动的topic PPT第17页: 如图所示是很多的topic，我们只截取了一部分，大家可以在自己电脑上看一看。 |  |  |
| PPT第18页: rostopic info命令  在终端输入如下命令：  $ rostopic info /cmd\_vel  我们可以看到这个topic的具体内容：type ，publishers，subscriber |  |  |
| **PPT第19页:**  这里我们查看另一个topic信息，它是机器人控制的一个topic，  在终端输入如下命令：  $ rostopic info /cmd\_vel/input/teleop  这里它的 publishers是：none。后面我们将启动机器人键盘移动控制，它的publishers将会改变。 |  |  |
| PPT第20页: 启动键盘控制机器人移动，  新建终端，输入如下命令：  $ rosrun robot\_sim\_demo robot\_keyboard\_teleop.py  这样我们就启动了机器人键盘控制脚本 |  |  |
| PPT第21页: 如图所示，我们可以在这个终端对机器人移动进行控制。（可具体介绍一下如何控制） |  |  |
| **PPT第22页：**  现在我们再来查看一下/cmd\_vel/input/teleop，这里它的 publishers已经改变了：/robot\_teleop |  |  |
| **PPT第23页:**  rostopic echo命令  在终端输入如下命令：  $ rostopic echo /cmd\_vel/input/teleop  我们可以看到机器人运动的角速度和线速度，现在没有移动，所以变化都为0  **PPT第24页:**  我们现在控制机器人移动，再查看速度变化  在键盘控制机器人的终端下控制机器人移动  回到当前终端重新输入如下命令：  $ rostopic echo /cmd\_vel/input/teleop  **PPT第25页:**  如图所示，机器人的角速度和线速度产生了变化，因为这里速度移动较慢，所以变化不大。大家可以自己调节速度试试看其变化。 |  |  |
| **PPT第26页:**  rosmsg list命令  在终端输入如下命令：  $ rostmsg list  我们可以看到所有的msg  **PPT第27页:**  这里我们只截取了部分内容，大家可以在自己电脑上查看其余部分。  **PPT第28页:**  rosmsg show命令  在终端输入如下命令：  $ rostmsg show  如图所示，我们可以看到 geometry/Twist这个msg的具体内容。  **PPT第29页:**  本节部分就到此结束了，谢谢各位同学的收听，希望大家多回顾这部分的讲义和ppt，起到记忆的效果。 |  |  |
| 知识点框图 | | |
| 通讯架构之话题  概念  命令  rostopic  roslaunch  roscore  rosnode  rosrun  rosmsg  launch文件  msg  topic  Master  Node | | |