《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 | ： | XXX |
| 提交时间 | ： | 2018年7月X日 |

中国大学MOOC制

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程  名称 | 《ROS机器人开发技术》 | | 章 | | 第一章 | | | 课程类型 |
| 节 | | 第一节 | | | 授课(√ )  实训( ) |
| 名称 | Rospy相关知识介绍 | | | | | | |
| 教师 |  | 课时 | | | | 1课时 | | |
| 参考  资料 | 1. Roswiki，柴老师课件 | | | | | | | |
| 教学  目的  要求 | 掌握：了解clientlibrary，了解rospy相关接口 | | | | | | | |
| 教学  重点  难点 |  | | | PPT页面 | | | 时间分配 | |
| 教学重点 | | | | | | | |
| 1. 介绍clientlibrary相关知识 | | | 4~6页 | | | 7分钟 | |
| 1. 简单介绍一下用python有哪些优势 | | | 7~9页 | | | 3分钟 | |
| 1. rospy和roscpp的对比介绍 | | | 9~11页 | | | 5分钟 | |
| 4. python代码的组织方式 | | | 12~13页 | | | 5分钟 | |
| 5. rospy-node相关的介绍 | | | 14~17页 | | | 5分钟 | |
| 6. rospy-topic相关的介绍 | | | 17~21页 | | | 5分钟 | |
| 7. rospy-service相关介绍 | | | 22~25页 | | | 5分钟 | |
| 8. rospy-param相关介绍 | | | 26~27页 | | | 5分钟 | |
| 9. rospy-time相关介绍 | | | 28~31页 | | | 5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：**  本章我们讲rospy的相关课程。   * 这个专栏是介绍利用Python来编写ROS程序 * 了解ROS参数，ROS服务，ROS发布器，ROS订阅器 * 运行完整的ROS节点 * 分析Python例程 * 分析rospy类API |  |  |
| **PPT第2页:**  本节课主要是对rospy的相关接口以及clientlibrary的介绍 |  |  |
| **PPT第4页:**  ROS为机器人开发者们提供了不同语言的编程接口，比如C++接口叫做roscpp，Python接口 叫做rospy，Java接口叫做rosjava。 |  |  |
| **PPT第5页:**  尽管语言不通，但这些接口都可以用来创建topic、 service、param，实现ROS的通信功能 |  |  |
| **PPT第6页:**  Clinet Lirary有点类似开发中的Helper Class，把一 些常用的基本功能做了封装。 目前ROS支持的Clinet Library包括：    目前最常用的只有roscpp和rospy，而其余的语言版本基本都还是测试版 |  |  |
| **PPT第7页：**  Rospy的讲解开始展开 |  |  |
| **PPT第8页：**  Python语言的简单介绍 |  |  |
| **PPT第9页:**  拥有简单脚本语言和解释型程序语言的易用性  拥有简单脚本语言和解释型程序语言的易用性  Python是一种解释型的，面向对象的，带有动态语义的高级程序设计语言 |  |  |
| **PPT第10页:**  **rospy vs roscpp** |  |  |
| **PPT第11页:**  rospy是Python版本的ROS客户端库，提供了Python编程需要的接口，你可以认为rospy就一个Python的模块(Module)。这个模块位于 /opt/ros/kineetic/lib/python2.7/distpackages/rospy 之中。  rospy包含的功能与roscpp相似，都有关于node、topic、service、param、time相关的操作。 但同时rospy和roscpp也有一些区别：   1. rospy没有一个NodeHandle，像创建publisher、subscriber等操作都被直接封装成了 rospy中的函数或类，调用起来简单直观。 2. rospy一些接口的命名和roscpp不一致，有些地方需要开发者注意，避免调用错误。   相比于C++的开发，用Python来写ROS程序开发效率大大提高，诸如显示、类型转换等细节 不再需要我们注意，节省时间。但Python的执行效率较低，同样一个功能用Python运行的耗 时会高于C++。因此我们开发SLAM、路径规划、机器视觉等方面的算法时，往往优先选择 C++。  ROS中绝大多数基本指令，例如 rostopic , roslaunch 都是用python开发的，简单轻巧。 |  |  |
| **PPT第12页:**  **ROS中Python代码的组织方式** |  |  |
| PPT第13页: 要介绍rospy，就不得不提Python代码在ROS中的组织方式。通常来说，Python代码有两种组 织方式，一种是单独的一个Python脚本，适用于简单的程序，另一种是Python模块，适合体 量较大的程序。  对于一些小体量的ROS程序，一般就是一个Python文件，放在script/路径下，非常简单。  your\_package  |- script/  |- your\_scrip  **Python模块**  当程序的功能比较复杂，放在一个脚本里搞不定时，就需要把一些功能放到Python Module 里，以便其他的脚本来调用。ROS建议我们按照以下规范来建立一个Python的模块：  your\_package  |- src/  |-your\_package/  |- \_init\_.py  |- modulefiles.py  |- scripts/  |- your\_script.py  |- setup.py  在src下建立一个与你的package同名的路径，其中存放 \_init\_.py 以及你的模块文件。这样 就建立好了ROS规范的Python模块，你可以在你的脚本中调用。 ROS中的这种 Python模块组织规范与标准的Python模块规范并不完全一致，你当然可以按照Python的标准去建立一个模块，然后在你的脚本中调用，但是我们还是建议按照ROS推荐的标准来写，这样方便别人去阅读。 |  |  |
| PPT第14页: **常用rospy的API**  Rospy就为python开发者提供了一套接口，让你可以用python来和主题服务参数来交互。  相比roscpp来说，rospy的开发更加简单快捷，一般我们写一些轻量级的ros程序首选用rospy来写。 因为开发起来比C++更快捷。所以rospy最适合一些需要快速实现，同时对运行速度没有那么大要求的场合。  其实我们常用的ros命令，roslaunch rostopic都是用python开发，也就是基于rospy开发的。 |  |  |
| PPT第15页: 和ros-node相关的API |  |  |
| PPT第16页: 来看一下rospy的官方文档，你可以看到rospy里面有很多submodule子模块  很多的类以及函数，你可能感觉rospy文档写的不像roscpp那样层次分明，  实现的代码有点乱，不过没关系，我把常用的主要函数都给放在ppt上。  这一页主要是Node和流程控制相关的rospy函数,怎么用呢？  在ros程序里，先import rospy，然后就可以rospy.init\_node。。。了  这一页主要是node相关的函数功能 |  |  |
| PPT第17页 rospy-Topic相关 |  |  |
| **PPT第18~20页：**  Topic通信相关的函数和类  主要是这个Publisher和Subscriber,  要理解publisher和subsceiber的工作机制  这种通信方式的主要特点也要强调一下 |  |  |
| PPT第21页： Rostopic的几种快捷操作的指令  在终端直接输入rostopic –h可以查看相关帮助  Ros的很多命令都有这种类似用法 |  |  |
| PPT第22页: Rospy-Service相关 |  |  |
| PPT第23~25页: Service相关函数和类  Call与\_\_call\_\_的区别 作用一样，但是用法不同 |  |  |
| PPT第26页: rospy-Param相关 |  |  |
| PPT第27页: |  |  |
| PPT第28页: rospy-Time相关 |  |  |
| **PPT第29~31页:**  时间控制相关的类  Time的now是静态方法，rospy.TIme.now()  Duration理解为一段时间  Time理解为一个时刻  所以这两者可以相加，Time+Duratoin=Time  Duration+Duration=Duration  Time-Time=Duration |  |  |
| 知识点框图 | | |
| Rospy相关介绍  Clientlibrary介绍  提供ROS编程的库P4-P6  rospy-Node相关P15-P16  P32  rospy-service相关P22-P25  rospy  Rospy vs Roscpp P10-P11  代码组织方式P12-P13  Rospy的API P14  rospy-topic相关P17-P21  rospy-param相关P26-P27  rospy-time相关P28-P31 | | |