

机器人学强化训练 - 移动机器人平台 & ROS

机器人学强化训练 - 移动机器人平台 & ROS

- 0.bash命令 [link](#)
- 1.Installing and Configuring Your ROS Environment
- 2.Navigating the ROS Filesystem
- 3.Creating & Building a ROS Package
- 4.Understanding ROS Nodes & Topic
- 5.了解tf [link](#)
- 6.主要实验流程
- 其他有助于调试的工具

0.bash命令 [link](#)

```
ls          # list files
cd          # change directory
rm          # remove
mv          # move
export      # export variable
echo       # output
whereis/which # check path of binary file
mkdir      # make directory
touch      # make new file if non exist
cat        # get file contents
-----
help/--help
man
```

1.Installing and Configuring Your ROS Environment

ROS Tutorials [link](#)

- 依照[官方文档](#)，我选择的是中科大的源
- 安装[catkin-tools](#): `apt install python-catkin-tools`
 - 更友好的编译输出
 - `catkin build`与 `catkin_make_isolated` 相似，允许并行编译

如果你使用了虚拟机，[这里](#)有一些tips可以尽量的加速你的虚拟机。以及：

- 尽量打开3D加速

- 如果没有打开3D加速的话, gazebo尽量缩小窗口大小, 这样可以减少render带来了cpu占用。

- 可以参考的文件管理结构:

```
└─ 'ROOT_DIR' or '~'
   └─ catkin_ws          # catkin workspace
      └─ build            #
      └─ devel           #
         └─ setup.bash   #
      └─ src              # src -> ../catkin_ws_backup/XXX_ws_src
   └─ catkin_ws_backup   # store packages | multi-workspace backup
```

- 可以简化你的常用命令 `alias rs="source devel/setup.bash"`
- `$ROS_PACKAGE_PATH`

2.Navigating the ROS Filesystem

- package是ROS构建代码或者工程的基本单位(区分node)
- rospack, roscd, rosls
- Tab Completion

3.Creating & Building a ROS Package

- `catkin_create_pkg <package_name>`
- 修改package.xml里面的配置
- CMakeLists (for c++ programming)
- TASK1: 使用rosls运行一个python脚本: 将脚本放在 `<your_package>/scripts` 文件夹

(python文件需要有执行权限)

```
catkin build
source devel/setup.bash
roscore
# open another console
rosls <package_name> <script_name.py>
```

4.Understanding ROS Nodes & Topic

- 进程通信
- Nodes: A node is an executable that uses ROS to communicate with other nodes.
- Messages: ROS data type used when subscribing or publishing to a topic.
- Topics: Nodes can publish messages to a topic as well as subscribe to a topic to receive messages.
- Master: Name service for ROS (i.e. helps nodes find each other)
- rosout: ROS equivalent of stdout/stderr

- roscore: Master + rosout + parameter server

```
rostopic [echo|pub|list|hz|help|...]
roscall [list|info|help|...]
rqt_graph
-----
rqt_plot
rosmmsg [show]
```

5.了解tf [link](#)

- tf的作用
- 在ros中使用tf



6.主要实验流程

- 需要用到的topic & tf

仿真topic/tf	真机topic / tf	类型	作用
/map	/map	nav_msgs/OccupancyGrid	接收地图信息(障碍物)
/course_agv/laser/scan	/scan	sensor_msgs/LaserScan	接收激光雷达信息
/course_agv/velocity	/webService/cmd_vel	geometry_msgs/Twist	发送AGV的速度信息
/course_agv/odom	待定	nav_msgs/Odometry	发送里程计信息
/course_agv/global_path	待定	nav_msgs/Path	发送GlobalPlanner的路径
/move_base_simple/goal	[target name] rviz 自定义	geometry_msgs/PoseStamped	获取rviz上的target信息
/robot_base	TF : /map <-> /base_footprint	tf	MAP坐标系 <-> AGV坐标系

- 熟悉AGV的系统(ubuntu 16.04), 设置AGV为ROS_MASTER, 可以接收topic内容(rostopic echo [topicname]) [link1](#) [link2](#)
- 熟悉rviz, 将消息可视化到本地的rviz

- 书写代码完成相应实验

-
- 实验一 : GlobalPlanner + LocalPlanner
 - 实验二 : Odometry
 - * 可选 实验三 : Odometry + Planner联合

-其他有助于调试的工具

一些比较实用的小工具

- [git](#) # 代码的版本管理软件（可以有效的防止代码丢失）
- [vim/nano](#) # 远程登录只有终端的机器人时可能并没有GUI软件可以用来修改文本文件
- vnc # 远程桌面

明确任务：（尽量在一周之内尝试下，一周后在线答疑）

- 熟悉Ubuntu系统，熟悉bash命令
- 安装,配置ROS环境(参考[ROS Tutorial 1](#))
- 自己尝试编写一对最简单的topic publisher/subscriber成功实现收发消息(c++/python) (参考[ROS Tutorial 11-13](#))
- 学会使用rviz作为可视化工具，编写一个publisher，输出一个path/pose rviz可以显示结果
- 顺利的同学可以考虑在仿真环境中先尝试自己编写算法了
- 两电脑在同一局域网下，一台作为ros master，测试互相通信

库地址

https://github.com/Nlct-RC/2020summer-advanced_robotics_mobile

[ZJUNlct](#) - 小型足球机器人实验室[纳新](#) ([报名表](#))