* 1. Rosbag

（1）简介：rosbag是一套用于记录和回放Ros topics的工具，旨在提高性能，并避免消息的反序列化（将在序列化过程中生成的二进制串转换成数据结构或者对象）和重新排序。rosbag package 提供了命令行工具以及C++类和Python的API。而且rosbag命令行工具和代码API是稳定的，始终保持向后的兼容性。

（2）Bags（包）: 用来储存message数据的一种文件格式。bags的命名以.bag作为后缀。所有发布在topic上的数据都储存在.bag文件中，并带有时间戳，可以根据用户的需求回放。包通常由诸如rosbag之类的工具创建，可以用来订阅一个或多个ROS topic 并将序列化的消息数据存储在接收到的文件中。

（3）rosbag命令行的功能：录制，重新发布来自包的信息，获取包的概括信息，检查包的消息类型，使用Python表达式过滤包中信息，压缩解压包，重新索引包。指令表如下：

|  |  |
| --- | --- |
| 命令 | 作用 |
| record | 用指定topics的内容记录一个包（.bag）文件 |
| info | 查看包中的内容 |
| play | 回放包中的内容 |
| check | 确定一个包是否可以在当前系统中使用，或者是否可以迁移 |
| fix | 修复包中的消息，以便在当前系统中播放 |
| filter | 用Python表达式转换包 |
| compress | 压缩一个或多个包 |
| decompress | 解压一个或多个包 |
| reindex | 重新索引一个或多个包 |

在ROS上操作机器人的时候，通过以与ROS兼容的格式保存传感器的数据，messages，指令和基本上所有数据类型，可以模拟机器人遇到的的任何物理条件。

（4）在terminal中输入命令：

记录所有topic的信息(a表示all)：

$ rosbag record -a

之后按Ctrl+C，带有时间戳的包就会被保存在当前文件夹里，比如2015-12-03-31-15.bag

记录特定的topic：

$ rosbag record <topic-names>

当运行一个复杂的系统时，比如PR2软件系统，会有几百个话题被发布，有些话题会发布大量数据（比如包含摄像头图像流的话题）。在这种系统中，要想把所有话题都录制保存到硬盘上的单个bag文件中是不切实际的。**rosbag record**命令支持只录制某些特别指定的话题到单个bag文件中，这样就允许用户只录制他们感兴趣的话题。

改变包的名称：

$ rosbag record -O <filename.bag> <topic-names>

显示包的信息：

$ rosbag info <filename.bag>

包括包的长度，topics和时间戳

回放包文件：

$ rosbag play <filename.bag>

播放时会按照记录时的时间顺序，在指令末尾加上-l会无限循环播放

播放包的特定时间段，例如：

$ rosbag play -u 240 <filename.bag>

即播放包的前240秒

将包的数据保存在文本文件中：

$ rostopic echo -b <filename.bag> -p /<topic-names> > data.txt

这条指令将一个topic的数据保存在叫data.txt的文件里

（5）rosbag package中的可执行文件：

通过添加合适的node元素，record和play这些可执行文件使包文件成为launch文件的一部分。

记录的node：

<node

pkg="rosbag"

name="record"

type="record"

args="-O <filename.bag> topic-names“

/>

播放的node：

<node

pkg="rosbag"

name=“play"

type=“play"

args="-O <filename.bag> topic-names“

/>

（6）使用Python访问bag文件：

像普通文件一样，bag可以通过python API被访问

“””用来读取rosbag数据的脚本“”“

import rosbag #导入rosbag接口

bag = rosbag.Bag(’test.bag’) #获取包中的信息

for (topic, msg, t) in bag.read\_messages(): #遍历所需信息

print topic, msg, t

“”“在bag文件中写入信息“”“

import rosbag #导入rosbag工具

from std\_msgs.msg import Int32, String #导入数据类型

bag = rosbag.Bag('test.bag', 'w') #打开文件

try: #异常检测

str = String()

str.data = 'foo' #赋值

i = Int32()

i.data = 42 #赋值

bag.write('chatter', str)

bag.write('numbers', i) #写入信息

finally:

bag.close() #关闭文件

（7）应用：

运行机器人，用rosbag工具记录下需要的数据，之后可以任意回放这些信息，也可以通过实验来处理这些数据，免去了多次运行机器人的麻烦。