

第十课 ROS 话题理解

1. 开始

Roscore: 首先确保 roscore 已经运行, 打开一个新的终端:

```
$ roscore
```

Turtlesim: 在本教程中我们也会使用到 turtlesim, 请在一个新的终端中运行:

```
$ rosrun turtlesim turtlesim_node
```

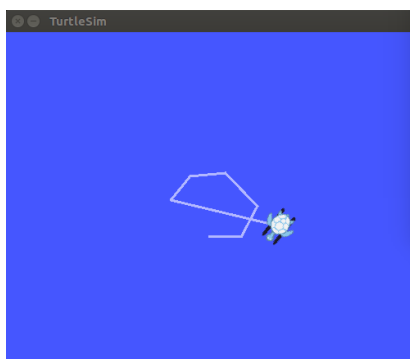
Turtle: 我们也需要通过键盘来控制 turtle 的运动, 请在一个新的终端中运行:

```
$ rosrun turtlesim turtle_teleop_key
```

```
huanyu@ubuntu:~$ rosrun turtlesim turtle_teleop_key
Reading from keyboard
-----
Use arrow keys to move the turtle.
```

现在你

可以使用键盘上的方向键来控制 turtle 运动了。如果不能控制, 请选中 teleop_key 所在的终端窗口以确保你的按键输入能够被捕获。



现在你可以控制 turtle 运动了, 下面我们一起来看看这背后发生的事。

2. ROS Topics

turtlesim_node 节点和 turtle_teleop_key 节点之间是通过一个 ROS 话题来互相通信的。turtle_teleop_key 在一个话题上发布按键输入消息, 而 turtlesim 则订阅该话题以接收该消息。下面让我们使用 rqt_graph 来显示当前运行的节点和话题。

使用 rqt_graph: rqt_graph 能够创建一个显示当前系统运行情况的动态图形。rqt_graph 是 rqt 程序包中的一部分。如果你没有安装, 请通过以下命令来安装:

- `sudo apt-get install ros-kinetic-rqt`
- `$ sudo apt-get install ros-kinetic-rqt-common-plugins`

在一个新终端中运行：你会看到类似下图所示的图形：

```
$ rosrun rqt_graph rqt_graph
```



如果你将鼠标放在 `/turtle1/command_velocity` 上方，相应的 ROS 节点（蓝色和绿色）和话题（红色）就会高亮显示。正如你所看到的，`turtlesim_node` 和 `turtle_teleop_key` 节点正通过一个名为 `/turtle1/command_velocity` 的话题来互相通信。



rostopic 介绍： `rostopic` 命令工具能让你获取有关 ROS 话题的信息。你可以使用帮助选项查看 `rostopic` 的子命令：

```
$ rostopic -h
```

```
Commands:
  rostopic bw      display bandwidth used by topic
  rostopic delay   display delay of topic from timestamp in header
  rostopic echo    print messages to screen
  rostopic find    find topics by type
  rostopic hz      display publishing rate of topic
  rostopic info    print information about active topic
  rostopic list    list active topics
  rostopic pub     publish data to topic
  rostopic type    print topic or field type
```

接下来

我们将使用其中的一些子命令来查看 `turtlesim`。

rostopic echo： `rostopic echo` 可以显示在某个话题上发布的数据。

```
用法: rostopic echo [topic]
```

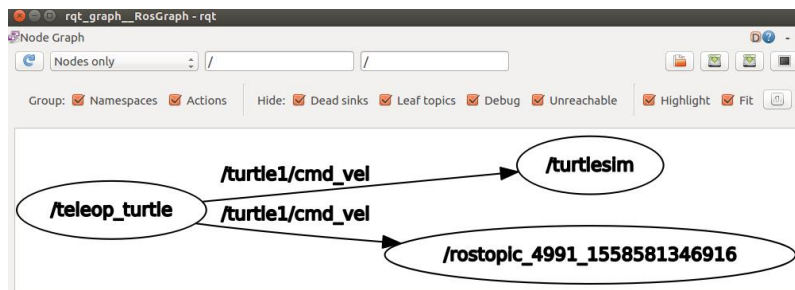
让我们在一个新终端中看一下 `turtle_teleop_key` 节点在 `/turtle1/cmd_vel` 话题上发布的数据。

```
$ rostopic echo /turtle1/cmd_vel
```

你可能看不到任何东西因为现在还没有数据发布到该话题上。接下来我们通过按下方向键使 turtle_teleop_key 节点发布数据。记住如果 turtle 没有动起来的话就需要你重新选中 turtle_teleop_key 节点运行时所在的终端窗口。现在当你按下向上方向键时应该会看到下面的信息：

```
huanyu@ubuntu:~$ rostopic echo /turtle1/cmd_vel
linear:
  x: 2.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
---
```

现在再看一下 rqt_graph（需要刷新一下 ROS graph）。正如你所看到的，rostopic echo(红色显示部分) 现在也订阅了 turtle1/cmd_vel 话题。



rostopic list: 能够列出所有当前订阅和发布的话题，查看一下 list 子命令需要的参数，在一个新终端中运行：

```
$ rostopic list -h
```

```
Options:
  -h, --help            show this help message and exit
  -b BAGFILE, --bag=BAGFILE
                        list topics in .bag file
  -v, --verbose          list full details about each topic
  -p                    list only publishers
  -s                    list only subscribers
  --host                group by host name
```

在 rostopic list 中使用 **verbose** 选项：

3. ROS Messages

话题之间的通信是通过在节点之间发送 ROS 消息实现的。对于发布者 (turtle_teleop_key) 和订阅器 (turtlesim_node) 之间的通信，发布器和订阅器之间必须发送和接收相同类型的消息。这意味着话题的类型是由发布在它上面的消息类型决定的。使用 rostopic type 命令可以查看发布在某个话题上的消息类型。

rostopic type: rostopic type 命令用来查看所发布话题的消息类型。

```
用法: rostopic type [topic]
```

```
$ rostopic type /turtle1/cmd_vel
```

```
huanyu@ubuntu:~$ rostopic type /turtle1/cmd_vel
geometry_msgs/Twist
```

```
$ rostopic show geometry_msgs/Twist
```

```
geometry_msgs/Vector3 linear
  float64 x
  float64 y
  float64 z
geometry_msgs/Vector3 angular
  float64 x
  float64 y
  float64 z
```

现在我们已经知道了 turtlesim 节点所期望的消息类型，接下来就可以给 turtle 发布命令了。

rostopic pub：可以把数据发布到当前某个正在广播的话题上。

用法： `rostopic pub [topic] [msg_type] [args]`

```
$ rostopic pub -1 /turtle1/cmd_vel geometry_msgs/Twist -- '[2.0, 0.0, 0.0]' '[0.0, 0.0, 1.8]'
```

以上命令会发送一条消息给 turtlesim，告诉它以 2.0 大小的线速度和 1.8 大小的角速度开始移动。



rostopic hz：rostopic hz 命令可以用来查看数据发布的频率。

用法： `rostopic hz [topic]`

我们看一下 turtlesim_node 发布 /turtle/pose 时有多快：

```
$ rostopic hz /turtle1/pose
```

```
subscribed to [/turtle1/pose]
average rate: 62.486
  min: 0.014s max: 0.018s std dev: 0.00074s window: 61
average rate: 62.489
  min: 0.014s max: 0.018s std dev: 0.00084s window: 124
average rate: 62.489
  min: 0.014s max: 0.018s std dev: 0.00087s window: 187
^Caverage rate: 62.514
  min: 0.014s max: 0.018s std dev: 0.00088s window: 239
```

现在我们可以知道了 turtlesim 正以大约 60Hz 的频率发布数据给 turtle。我们也可以结合 rostopic type 和 rostopic show 命令来获取关于某个话题的更深层次的信息：

```
rostopic type /turtle1/cmd_vel | rosmmsg show
```

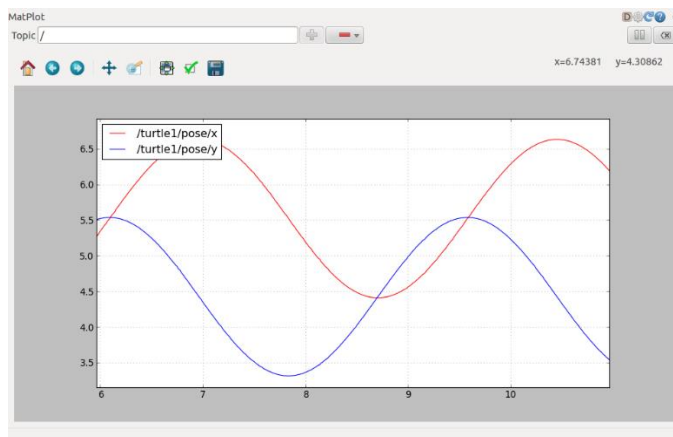
到此我们已经完成了通过 rostopic 来查看话题相关情况的过程，接下来我将使用另一个工具来查看 turtlesim 发布的数据。

4. 使用 rqt_plot

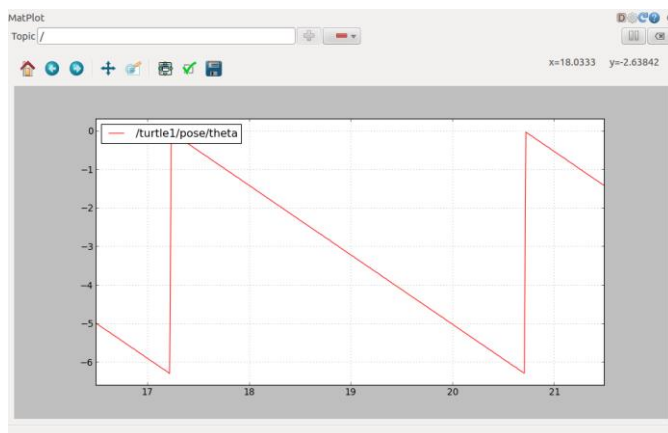
rqt_plot 命令可以实时显示一个发布到某个话题上的数据变化图形。这里我们将使用 rqt_plot 命令来绘制正在发布到 /turtle1/pose 话题上的数据变化图形。首先，在一个新终端中运行 rqt_plot 命令：

```
$ rosrunc rqt_plot rqt_plot
```

这会弹出一个新窗口，在窗口左上角的一个文本框里面你可以添加需要绘制的话题。在里面输入 /turtle1/pose/x 后之前处于禁用状态的加号按钮将会被使能变亮。按一下该按钮，并对 /turtle1/pose/y 重复相同的过程。现在你会在图形中看到 turtle 的 x-y 位置坐标图。



按下减号按钮会显示一组菜单让你隐藏图形中指定的话题。现在隐藏掉你刚才添加的话题并添加 /turtle1/pose/theta，你会看到如下图所示的图形：



5. 总结

本部分教程到此为止，我们已经理解了 ROS 话题是如何工作的。同时学习和使用了话题相关的工具和命令集，本课程在 ROS 开发中占据了很高的使用率。