Data logging and processing using rosbag

Ivan Marković Matko Orsag Damjan Miklić

Automation and Robotics
Robot Programming and Simulation

2020

UNIVERSITY OF ZAGREB
Faculty of Electrical Engineering and Computing

Overview

- So far we have covered the basic tools for running ROS software and examining data during run-time:
  - Running nodes (rosrun, roslaunch, rostopic)
  - Coding nodes (in Python)
  - Analyzing node behavior (rosnode, rostopic, rqt_graph, ...)

  It is often more convenient to record the data and analyze it offline:
  - rosbag is the tool for logging data exchanged through ROS
  - How to access the data for processing, plotting etc?

Let's record (bag) a dataset

- Run turtlesim and record some data

```
$ roscore
$ rosrun turtlesim turtlesim_node &
$ rosrun turtlesim turtle_teleop_key
$ rosbag record -a -O test
```

- Drive around, then change the background color

```
$ rosparam set /turtlesim/background_g 255
$ rosservice call /clear
```

- Stop the recording with `Ctrl-C`
- Reset the background

```
$ rosparam set /turtlesim/background_g 86
$ rosservice call /clear
```

Playing back a bagged dataset

- Get info on bag contents

```
$ rosbag info test.bag
```

- Start playback

```
$ rosbag play test.bag
```

- Playback can be paused by pressing `space`
- Note that service calls are not recorded (directly :)

Exercise

Consult the rosbag documentation and find out how to skip the first 10 seconds when playing the bag.

Reading the data

- Let's write a Python script `read_bag.py` to import the data

Don't forget

Configure your editor to insert spaces instead of tabs!

```
#! /usr/bin/env python3
""" A script for reading rosbag data. """

import rosbag

if __name__ == '__main__':
    bag = rosbag.Bag('test.bag')
    for (topic, msg, t) in bag.read_messages():
        print(topic, msg, t)
```

Importing the data into IPython

- We can filter the topics we are interested in

```
bag.read_messages(topics=['/turtle1/cmd_vel'])
```

- Importing into IPython

```
$ ipython --pylab
In[1]: run read_bag.py
```

- All the variables from our code are imported automatically

```
In[2]: %whos
```

- Let's look at the `msg` variable

```
In[3]: msg
```

Obtaining a list of data

- Creating a list by list comprehension

```
vel_x = [msg.linear.x for (topic, msg, t)
                          in bag.read_messages(...)]
vel_t = [t.to_time() for (topic, msg, t)
                          in bag.read_messages(...)]
```

Exercise

Think about the efficiency of the above approach, when reading multiple data points from each message. How could it be improved?

Processing the data

- IPython's pylab extension supports Matlab-like commands

```
plot(vel_t,vel_x,'b-')
```

Assignments

1. Plot the x, y and orientation coordinates of the turtle in three subplots.
2. Compute the total path travelled by the turtle during the experiment.
3. At which second (from the beginning of the recording) did the background color change?

Hint

In IPython, help is accessed by typing ?, followed by the command name, e.g., ?plot.

Summary

- Rosbag provides tools to record and play back messages exchanged within a ROS system
- The python API provides functions for accessing the data from Python programs
- IPython provides a Matlab-like environment for data analysis

Useful links

- Rosbag command line tool reference
- Rosbag code API
- Rosbag cookbook
- IPython tutorial