

AA 274A: Principles of Robotic Autonomy I

(virtual) Section 6: rosbags

Our goals for this section:

1. Finish up the navigator from Section 5.
2. Learn how to use rosbags.

1 Getting the navigator working

Before learning about `rosvbag`, we will need to have a working navigator. If you did not finish the navigator during Section 5, take this opportunity to make sure that it is working. In particular, when you input a 2D Nav Goal in `rviz`, your turtlebot should plan and execute a trajectory to the goal.

2 rosvbag

An important tool for debugging and programming with ROS is `rosvbag`. This tool will allow you to record topic data from a running ROS system for later playback. The topic data will be accumulated in a bag file. In this section, we will use `rosvbag` to record performance of the pose controller under different settings to help choose controller gains.

First, edit `asl_turtlebot/scripts/controllers/P2_pose_stabilization.py` to publish the computed α , δ , and ρ values to the topics `/controller/alpha`, `/controller/delta`, and `/controller/rho` topics respectively.

Problem 1: What message type did you choose for each of these messages? Include your updated code in your submission.

HINT: you'll need to add some imports to this file. Refer to other publishers you've written in the past!

Next, your goal is to use `record` to record the α , δ , and ρ values as your navigator runs on the robot. Record multiple bags for different values of the controller gains (play with the gains passed in to the controllers inside `__init__()` in `navigator.py`).

Take a look at the `rosvbag` tutorials and documentation to find the command to record a `rosvbag` (Hints: flags are capitalization sensitive, `-o` flag can specify a custom prefix, and record only topics you need to keep bag clean!):

- <http://wiki.ros.org/rosvbag/Commandline>
- <http://wiki.ros.org/rosvbag/Tutorials/Recording%20and%20playing%20back%20data>

Problem 2: What command did you use to record the requested topics to a particular file name?

After you are done, in the window running `rosvbag record` exit with a Ctrl-C. Now examine the contents of the directory `/bagfiles`. You should see a file with a name that begins with the year, date, and time and

the suffix `.bag`. This is the bag file that contains all topics published by any node in the time that `rosvim` was running. We can use the following command to examine what is recorded in the bagfile:

```
1 | rosbag info <your bagfile>
```

Problem 3: Include the output of `rosvim info` in your write up.

3 Visualizing results with `rqt`

After recording the data, we can play it back and visualize it using a tool called `rqt_plot`.

Kill the "live" ROS simulation that you launched earlier using `rosvim`. Then, in another terminal open `rqt_plot`:

```
1 | rqt_plot
```

and add the three topics that we logged. Finally, in another terminal, use `rosvim` to playback the data you recorded (take a look at the documentation for the command to playback a `rosvim`).

Problem 4: What happens when you run the command `rosvim play`? Why do we need to start `rosvim` before running `rosvim play`?

Problem 5: Take a screenshot of the resulting plot in `rqt_plot` and include it in your submission.

You may need to play with the x axis limits to get a nice looking plot. Right click on the plot and drag to zoom the axes.

Problem 6: If you have time, record and play back ρ , α , δ for several controller gains. What differences do you see as you change each of the gains? Include the plots in your write up.