Note: This tutorial assumes that you have completed the previous tutorials: Navigating the Wiki (/ROS/Tutorials/NavigatingTheWiki).

💡 Please ask about problems and questions regarding this tutorial on 🌐 answers.ros.org (http://answers.ros.org). Don't forget to include in your question the link to this page, the versions of your OS & ROS, and also add appropriate tags.

Where Next?

Description: This tutorial discusses options for getting to know more about using ROS on real or simulated robots.

Tutorial Level: BEGINNER

At this point in the beginner's tutorials you should have an understanding of the core concepts of ROS.

Given a robot that runs ROS, you could use this understanding to list topics published and subscribed by the robot, to identify the messages consumed by these topics and then write your own nodes that process sensor data and act in the world.

The real attraction of ROS is not the publish/subscribe middle-ware itself but that ROS provides a standard mechanism for developers around the world to share their code. The best "feature" of ROS is its enormous community.

The number of packages available can be overwhelming. This tutorial attempts to give you an idea of what to explore next.

1. Launching a Simulator

Even if you have a real robot, it is good to get started using a simulator so that if something goes wrong you don't injure yourself or damage an expensive robot.

You can get started with the PR2 Simulator (/pr2_simulator/Tutorials) or the Turtlebot Simulator (/turtlebot_simulator/Tutorials). Alternately, you might search for your robot (/Robots) and check whether it has a simulator of its own.

At this point, you might try to control the simulated robot using a 'teleop' package (e.g., turtlebot_teleop (/turtlebot_teleop/Tutorials/Teleoperation)) or use your understanding of ROS to find a topic and write code that sends an appropriate message to drive your robot.

2. Exploring RViz

RViz (/rviz) is a powerful visualization tool that allows you to view the robot's sensors and internal state. The user guide (/rviz/UserGuide) will help you get started.

3. Understanding TF

The TF (/tf) package transforms between different coordinate frames used by your robot and keeps track of these transforms over time. A good understanding of TF is essential when working with any real robot. It is worthwhile to work through the tutorials.

If you're building your own robot, you might at this point consider constructing a URDF model (/urdf /Tutorials) for your robot. If you're using a "standard" robot then one has probably already being built for you. Nevertheless, it may be worthwhile to briefly familiarize yourself with the URDF (/urdf) package.

4. Going Deeper

At this point, you're probably ready to start getting your robot to perform more sophisticated tasks. The following pages may help you:

1. actionlib (/actionlib) - The actionlib package provides a standardized interface for interfacing with preemptible tasks. This is widely used by "higher-level" packages in ROS.

2. navigation (/navigation) - 2D navigation: map-building and path planning.

3. 🌐 MoveIt (http://moveit.ros.org/) - To control the arms of your robot.

Wiki: ROS/Tutorials/WhereNext (dernière édition le 2016-09-08 00:59:36 par DanielHeater (/DanielHeater))

Brought to you by: Open Source Robotics Foundation

(http://www.osrfoundation.org)