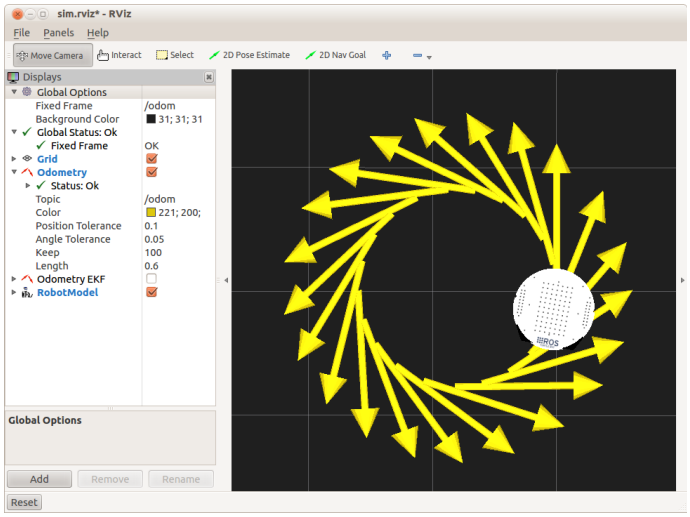
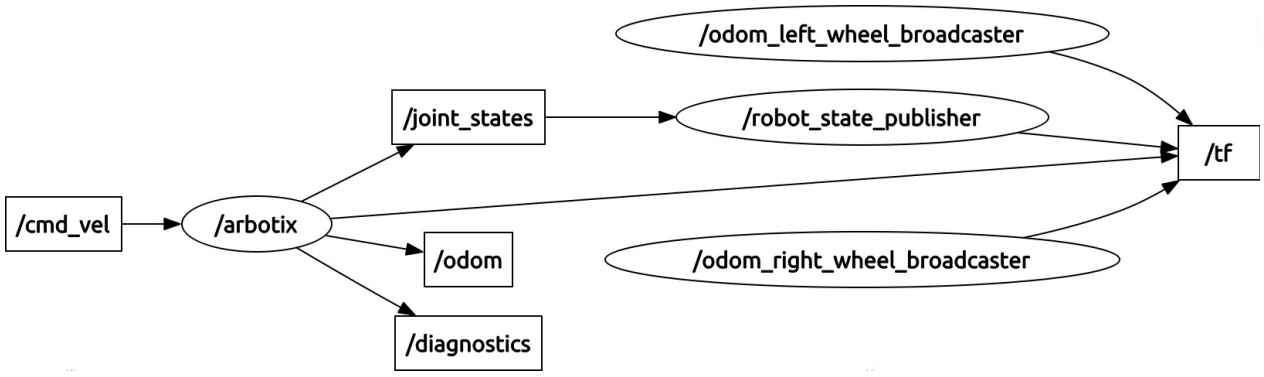




Tutorial 3

Packages, Simulation & Basic Control

Ralph SEULIN

Le2i UMR CNRS 6306 – http://le2i.cnrs.fr
ralph.seulin@u-bourgogne.fr

v1.3 - January 2016

1- Introduction

This tutorial is mainly inspired from the ROS by Example Book – v1.04 for ROS Hydro.

The goal of this tutorial is to cover the following topics :

- Learn how to install some packages
- Install and test a robot simulator
- Control the *simulated* mobile base by sending velocity commands
- Control the *real* mobile base by sending velocity commands

1.1- References

- ROS by Example eBook -

  http://www.lulu.com/shop/r-patrick-goebel/ros-by-example-hydro-volume-1/paperback/product-21663223.html
- Ros By Example git repository – https://github.com/pirobot/rbx1/tree/hydro-devel
- Ros By Example Forum - https://groups.google.com/forum/#!forum/ros-by-example

2- Packages installation

In this chapter we will install some necessary packages to simulate the TurtleBot robot. By doing this we will cover the installation package topic and provide some guidelines.

2.1- Install additional packages

You have first to download the package lists from the repositories and "updates" them to get information on the newest versions of packages and their dependencies:

$ *sudo apt-get update*

Debian packages are installed by the *sudo apt-get install* command - http://linux.die.net/man/8/apt-get

Install the following additional ROS packages we will need later. (Instructions will also be provided for installing individual packages) :

$ *sudo apt-get install ros-hydro-turtlebot-* *
ros-hydro-openni-camera ros-hydro-openni-launch \
ros-hydro-openni-tracker ros-hydro-laser- *
ros-hydro-audio-common ros-hydro-joystick-drivers \
ros-hydro-orocos-kdl ros-hydro-python-orocos-kdl \
ros-hydro-dynamixel-motor- * *ros-hydro-pocketsphinx* \
gstreamer0.10-pocketsphinx python-setuptools python-rosinstall \
ros-hydro-opencv2 ros-hydro-vision-opencv \
ros-hydro-depthimage-to-laserscan \
git subversion mercurial

(The \ character at the end of each line makes the entire block appear like a single line to Linux when doing a copy-and-paste)

Check if some of the packages are already installed.

Additional informations

- Installing packages - ROS by Example Vol.1 – 4.15 Installing Packages with SVN, Git, and Mercurial

2.2- Install ROS by Example Code

To clone the rbx1 repository for Hydro, follow these steps:

Move to the src folder of your catkin workspace

$ cd ~/ros/hydro/catkin_ws/src

Clone the code contained in the repository called rbx1 on Github

$ git clone https://github.com/pirobot/rbx1.git

Move to the rbx1 folder

$ cd rbx1

Select the Hydro branch of the repository. (By default, the clone operation checks out the Groovy branch for the benefit of those still using Groovy.)

$ git checkout hydro-devel

Move back to your catkin workspace folder (do not remain in the src folder!)

$ cd ~/ros/hydro/catkin_ws

Build the new packages

$ catkin_make

Finally run the following command:

$ rospack profile

The *rospack profile* command simply ensures that the new stack and packages will be visible in your ROS package path. It is not usually necessary to run this command but it doesn't hurt either.

Additional informations

- Building a ROS package - http://wiki.ros.org/ROS/Tutorials/BuildingPackages

All of the ROS By Example packages begin with the letters rbx1. To list the packages, move into the parent of the rbx1 meta-package and use the Linux ls command:

$ roscd rbx1

$ cd ..

$ ls -F

which should result in the following listing:

rbx1/

rbx1_apps/

rbx1_bringup/

rbx1_description/

rbx1_dynamixels/

rbx1_experimental/

rbx1_nav/

rbx1_speech/

rbx1_vision/

README.md

Remember that the *roscd* command is used to move from one package to another. For example, to move into the rbx1_speech package, you would use the command:

$ *roscd rbx1_speech*

Note that you can run this command from any directory and ROS will find the package. You do not already have to be in the package directory.

IMPORTANT: If you are using two computers to control or monitor your robot, such as a laptop on the robot together with a second computer on your desktop (TurtleBot configuration), be sure to install the necessary stacks or packages on both machines.

3- Arbotix Simulator

To test our code on a simulated robot, we will use the arbotix_python simulator found in the ArbotiX stack by Michael Ferguson.

3.1- Installing the Simulator

To install the simulator, install the following debian packages:

*ros-hydro-arbotix-**

Finally, run the command:

$ rospack profile

3.2- Testing the Simulator

To make sure everything is working, make sure roscore is running, then launch the simulated TurtleBot as follows:

$ roslaunch rbx1_bringup fake_turtlebot.launch

which should result in the following startup messages:

process[arbotix-1]: started with pid [4896]

process[robot_state_publisher-2]: started with pid [4897]

[INFO] [WallTime: 1338681385.068539] ArbotiX being simulated.

[INFO] [WallTime: 1338681385.111492] Started DiffController

(base_controller). Geometry: 0.26m wide, 4100.0 ticks/m.

Next, bring up RViz so we can observe the simulated robot in action:

$ rosrun rviz rviz -d `rospack find rbx1_nav`/sim.rviz

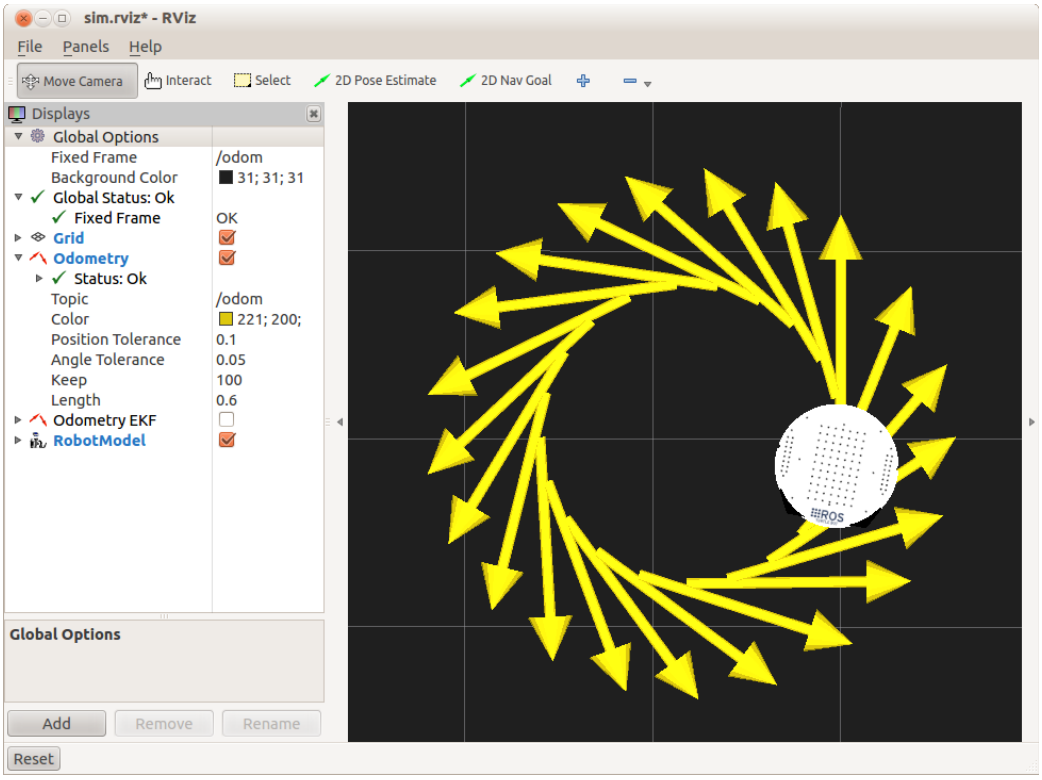
(Note how we use the Linux backtick operator together with the rospack find command to locate the rbx1_nav package without having to type the entire path.)

If everything went right, you should see the TurtleBot in RViz. (The default view is top down. To change views, click on the Panels menu in Rviz.)

To test the simulation, open another terminal window or tab and run the following command which should cause the simulated robot to move in a counter-clockwise circle:

$ rostopic pub -r 10 /cmd_vel geometry_msgs/Twist '{linear: {x: 0.2, y: 0, z: 0}, angular: {x: 0, y: 0, z: 0.5}}'

The view in RViz should look something like this. (The image below was zoomed using the mouse scroll wheel.)



To stop the rotation, type Ctrl-C in the same terminal window, then publish the empty Twist message:

rostopic pub -1 /cmd_vel geometry_msgs/Twist '{}'

Additional informations

Installing Arbotix Simulator - ROS by Example Book – Chapter 6. INSTALLING THE ARBOTIX SIMULATOR

4- Control the *simulated* mobile base

Study carefully and run commands of the following chapters of ROS by Example Book :

7. CONTROLLING A MOBILE BASE
   - 7.1 Units and Coordinate Systems
   - 7.2 Levels of Motion Control
   - 7.3 Twisting and Turning with ROS

You can skip the chapter “7.4 Calibrating Your Robot's Odometry” as Turtlebot2 is already calibrated.

4.1- Simulated robot teleoperation

- Setup the already installed RVIZ teleop panel to teleoperate your simulated robot.
- Arbotix Controller GUI can also be used with the following command:

  $ *arbotix_gui*

Additional informations

http://docs.ros.org/hydro/api/rviz_plugin_tutorials/html/panel_plugin_tutorial.html

5- Control the *real* mobile base

Study carefully the following chapter of ROS by Example Book :

7.5 Sending Twist Messages to a Real Robot

5.1- Todo

- Start the ASUS NetBook and the Kobuki Base.
- Launch the *minimal.launch* file from the *turtlebot_bringup* package.
- Study the active nodes and topics with help from the following additional documentation:
  - ***Notes on TurtleBot Command Velocity Multiplexer***
- Send some velocity commands to your robot by publishing to the appropriate topic.

!!! START WITH LOW VALUES FOR VELOCITY !!!