Robotics
Introduction to ROS

Roscore, catkin, rosnodes, rostopics and rosservices

Jonathan Ferrer Mestres
jonathan.ferrer@gmail.com

What is ROS?

ROS (Robot Operating System) is a framework for writing robot software. It includes a set of libraries and tools that allows robotics community to easily share algorithms and simplify the task of creating robot behavior for a wide variety of robotic platforms.

http://wiki.ros.org/ROS/Tutorials

Rosnode and Roscore

A **ros*node*** is a process that performs a functionality. Nodes are combined together into a graph and communicate with one another using rostopics and rosservices.

Command: *rosnode <option>*

Roscore is a collection of nodes and programs that are pre-requisites of a ROS-based system. Roscore allows rosnodes to communicate between them.

Command*: roscore*

Catkin workspace and ROS packages

A **catkin workspace** is a folder where you can create and build catking packages:

http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment

A **catkin package** can be seen as a program.

http://wiki.ros.org/ROS/Tutorials/CreatingPackage

Running nodes

http://wiki.ros.org/ROS/Tutorials/UnderstandingNodes

Rostopics

Topics: Nodes can *publish* messages to a topic as well as *subscribe* to a topic to receive messages. Rostopics allows nodes to communicate between them.

Messages: ROS data type used when subscribing or publishing to a topic.

Commands: *rostopic <option>, rosmsg <option>*

http://wiki.ros.org/ROS/Tutorials/UnderstandingTopics

Rosservices

Rosservices are another way that nodes can communicate with each other. Services allow nodes to send a **request** and receive a **response**.

Command: *rosservice <option>*

http://wiki.ros.org/ROS/Tutorials/UnderstandingServicesParams

Roslaunch

roslaunch starts nodes as defined in a launch file:

http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch

Creating a ROS msg and srv

msg files are simple text files that describe the fields of a ROS message.

srv files describes a service. It is composed of two parts: a request and a response.

http://wiki.ros.org/ROS/Tutorials/CreatingMsgAndSrv

Writing a Simple Publisher and Subscriber

http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28c%2B%2B%29

http://wiki.ros.org/ROS/Tutorials/ExaminingPublisherSubscriber

Writing a Simple Service and Client

http://wiki.ros.org/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29

http://wiki.ros.org/ROS/Tutorials/ExaminingServiceClient

Exercise

Exercise 1: Create a catkin package *turtlesim_ex1* with a publisher to move the turtle randomly and a subscriber to the current turtle pose.

Exercise 2: Explain the differences between a publisher/subscriber and a client/service