Autonomous Systems & Introduction to Robotics

ROS practical session

João Quintas & Rui Bettencourt

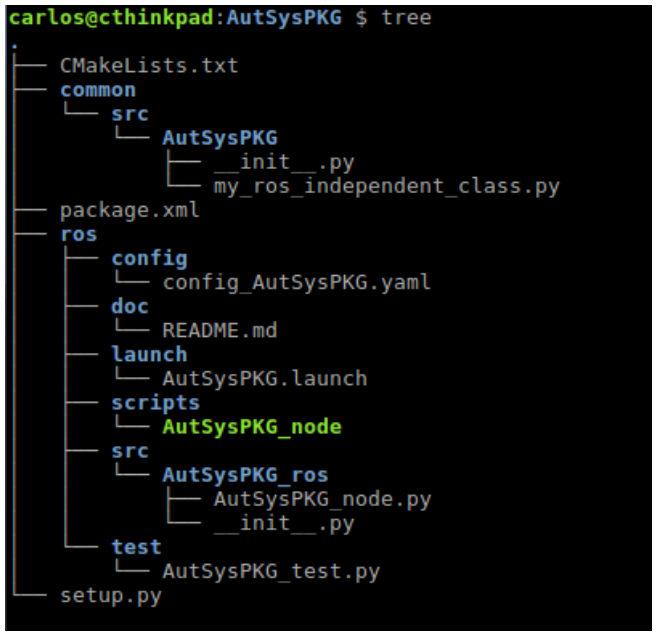
(Oscar Lima & Carlos Azevedo & Rute Luz)

ISR: Institute for Systems and Robotics
LARSyS: Laboratory for Robotics and Engineering Systems
IST: Instituto Superior Técnico, Lisboa Portugal

September 24th 2020

ROS pkg structure[1]

- ROS nodes, a ROS-independent library, a dataset, configuration files, a third-party piece of software, etc
- ROS packages tend to follow a common structure
- For python code it will look like this:
- To create a package you can use the command:
  catkin_create_pkg {*name of package*} {*dependencies*}



[1] http://wiki.ros.org/Packages

rosbash[2]

- Offers a set of shell commands for using ros with bash (linux terminal)
- Most popular include:
  - roscd pkg_name (cd to pkg_name easily)
  - rosed pkg_name filename (quickly edit a file)
  - roscat pkg_name filename (quickly visualize a file in terminal)
  - rosrun pkg_name executable (run executable from anywhere without having to give its full path)
- enables tab completion on: roslaunch, rosparam, rosnode, rostopic, rosservice, rosmsg, rossrv, rosbag.

[2] `http://wiki.ros.org/rosbash`

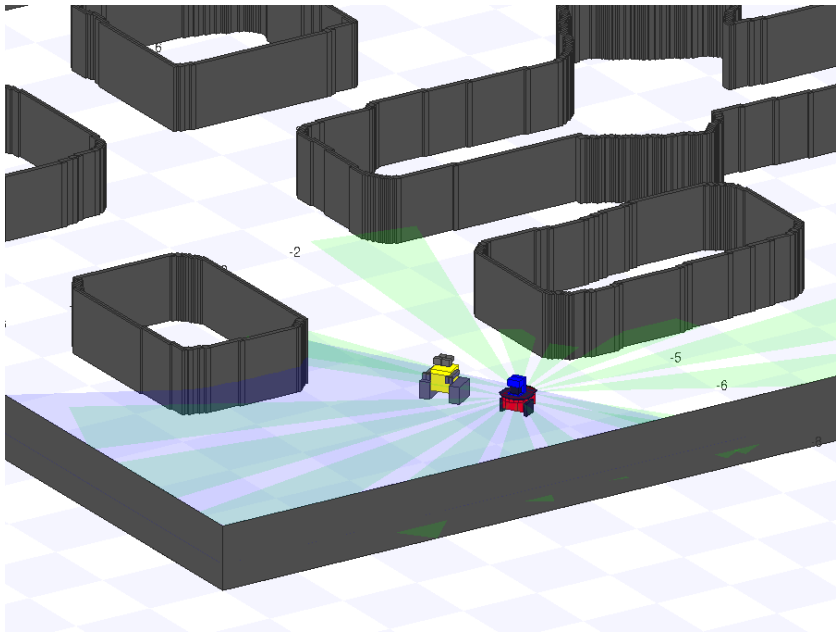
rosrun[3]

- Part of rosbash suite
- Usage: rosrun pkg_name executable_name
- It will run ONLY executable files
- About files being executable (important!)
  - make sure your python nodes (i.e. my_python_node.py) are executable
  - check by doing: ls -l , if it has an x is executable (i.e. -rwxr–r–)
  - alternatively, if your terminal has colors, the file shows green when doing ls
  - rosrun will also look for your compiled c++ executables (under devel/lib/pkg_name)

[3] `http://wiki.ros.org/rosbash#rosrun`

Stage simulator[4]

- Simulates a population of mobile robots, sensors and objects in a two-dimensional bitmapped environment
- Stage was designed with multi-agent systems in mind, so it provides fairly simple, computationally cheap models of lots of devices rather than attempting to emulate any device with great fidelity.



[4] http://playerstage.sourceforge.net/index.php?src=stage

rostopic[5]

- Displays information about ROS topics
- Most useful:
- rostopic list (get a list of active topics)
- rostopic info topic_name (get topic type, publishers and subscribers)
- rostopic echo topic_name
- rostopic pub topic_name topic_type msg_press_tab! (publish a topic from console), options:
  - no args (latched)
  - -r float_number (at a certain rate)
  - - -once (latch for 3 secs, then dies)
- rostopic hz topic_name (get the publish frequency rate)

[5] `http://wiki.ros.org/rostopic`

roslaunch[6]

- A tool for easily launching multiple ROS nodes
- Implemented with XML syntax (<launch>... </launch>)
- Allows to load parameters to param server
- A launch file can call other launch files
- Launch a node <node pkg="..." type="..." name="..." respawn=true ns="..." />
- Run syntax: roslaunch pkg_name my_file.launch

[6] http://wiki.ros.org/roslaunch

parameter server[7]

- Is a shared, multi-variate dictionary that is accessible via network API
- Nodes can use this server to store or retrieve parameters during runtime
- Is not high performance
- Globally viewable
- Usage from terminal: rosparam set param_name param_value, rosparam get param_name
- Usage from python api: rospy.set_param(param_name, param_value), rospy.get_param("param_name")
- Suitable for for static, non-binary data such as configuration parameters

[7] http://wiki.ros.org/Parameter%20Server

Rviz[8]

- 3D visualization tool
- Powerful for topic visualization (useful in debugging)
- Sensoring state information (laser scans, pointclouds, coordinate frames, cameras)
- Can publish some topics (2D pose estimate, 2D nav goal)
- Is recommended to comply with ROS standard topics to enable topic visualization
- Launch using : rosrun rviz rviz (a roscore must be running)
- Not a simulator

`https://www.youtube.com/watch?v=i--Sd4xH9ZE`

[8] `http://wiki.ros.org/rviz`

Reading Material

- For nice tutorials you can read the book "Programming Robots with ROS: A Practical Introduction to the Robot Operating System"
- For a tutorial on turtlebot3 simulation environment you can check

`https://emanual.robotis.com/docs/en/platform/turtlebot3/simulation/#ros-1-simulation`

Thank you!
Questions? :)

If you have a question please create a Github issue so that we can all benefit from the posted answers under:
`https://github.com/socrob/autonomous_systems/issues`