# Autonomous Systems

## ROS practical session

Carlos Azevedo & Oscar Lima

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

September 18, 2018



# ROS pkg structure[1]

- ROS packages tend to follow a common structure
- For python code it will look like this:

```
oscar@oscar-lenovo:test $ source ~/scripts/create_ros_pkg.sh my_ros_pkg

oscar@oscar-lenovo:my_ros_pkg $ tree
.
├── CMakeLists.txt
├── common
│   └── src
│       └── my_ros_pkg
│           ├── __init__.py
│           └── my_ros_independant_class.py
├── package.xml
├── ros
│   ├── config
│   │   └── config_my_ros_pkg.yaml
│   ├── doc
│   │   └── README.md
│   ├── launch
│   │   └── my_ros_pkg.launch
│   ├── scripts
│   │   └── my_ros_pkg_node
│   ├── src
│   │   └── my_ros_pkg_ros
│   │       ├── __init__.py
│   │       └── my_ros_pkg.py
│   └── test
│       └── my_ros_pkg_test.py
└── setup.py
```

[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

# rostopic[4]

- Displays information about ROS topics
- Most useful:
- rostopic list (get a list of active topics)
- rostopic echo topic_name
- rostopic info topic_name (get topic type, publishers and subscribers)
- 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)

[4] http://wiki.ros.org/rostopic

# parameter server[5]

- 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

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

# roslaunch[6]

- A tool for easily launching multiple ROS nodes
- Implemented with XML syntax
- Allows to set parameters in param server
- A launch file can call other launch files
- Run syntax: roslaunch pkg_name my_file.launch

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

# Rviz[7]

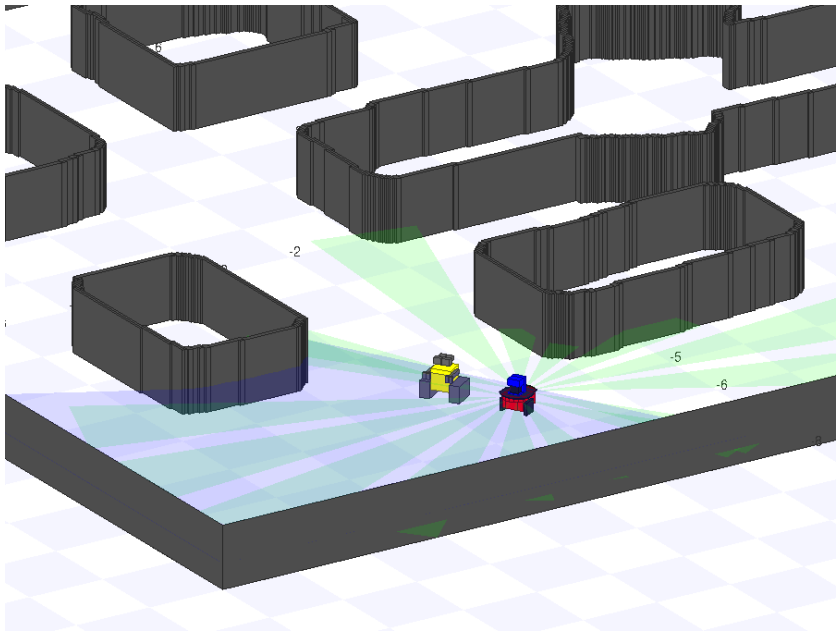
- Powerful tool for topic visualization
- Useful for debugging
- launch using : rosrun rviz rviz (a roscore must be running)
- Can publish some topics (2D pose estimate, 2D nav goal, Point)
- Is recommended to comply with ROS standard topics to enable topic visualization

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

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

# Stage simulator[8]

- Stage is a robot simulator from Richard Vaughan and contributors (1998-2009)
- Simulates a population of mobile robots, sensors and objects in a two-dimensional bitmapped environment



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

# 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`