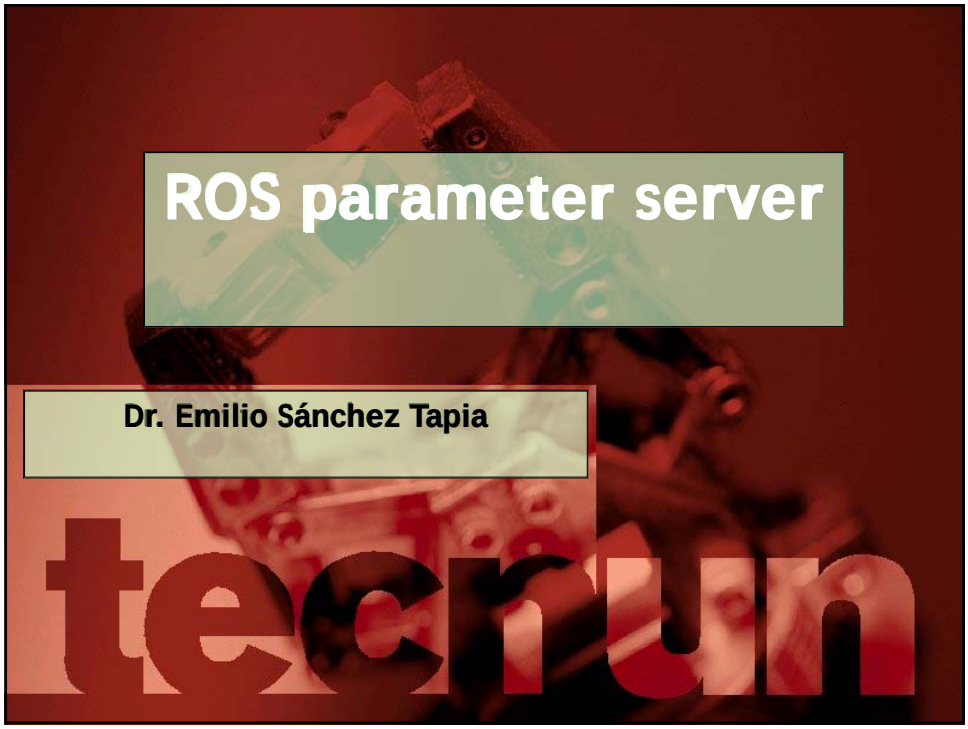
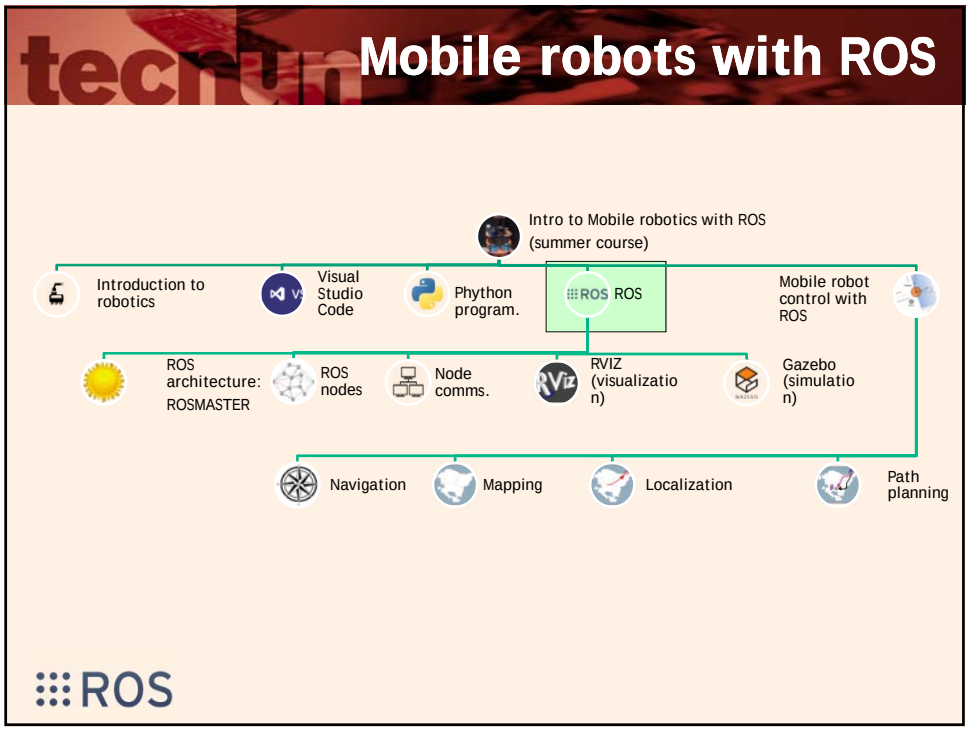
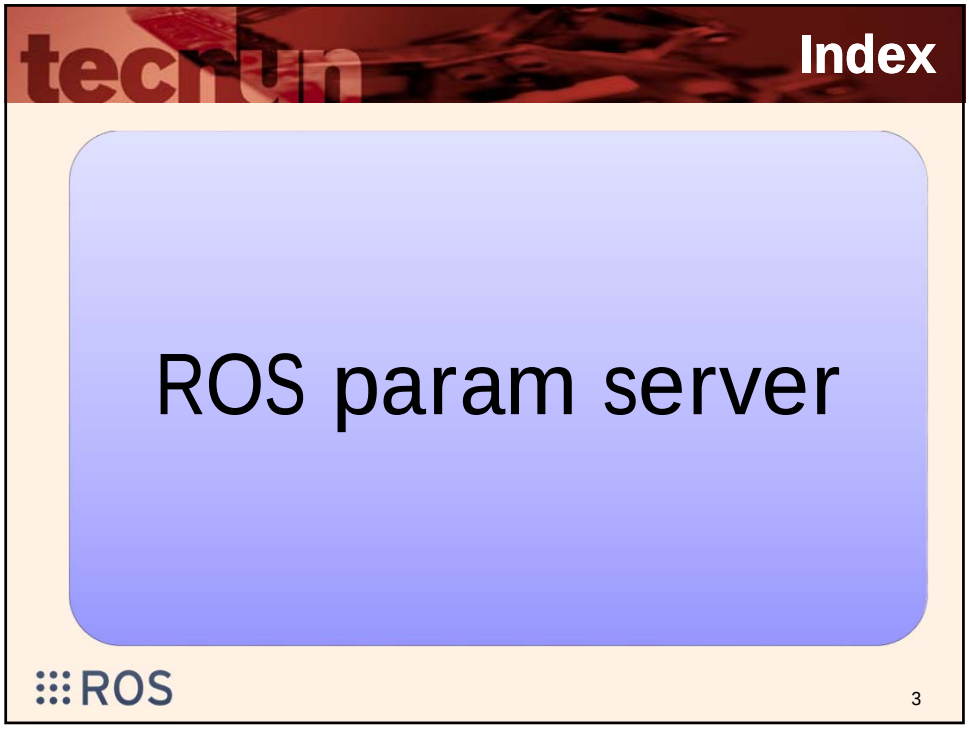

ROS parameter server
Dr. Emilio Sánchez Tapia
tecnun


tecnun
Mobile robots with ROS
Intro to Mobile robotics with ROS (summer course)
Introduction to robotics
Visual Studio Code
Phython program.
ROS
Mobile robot control with ROS
ROS architecture: ROSMASTER
ROS nodes
Node comms.
RVIZ (visualizatio n)
Gazebo (simulatio n)
Navigation
Mapping
Localization
Path planning
ROS


Index
ROS param server
ROS
3


ROS parameter server
ROS
tecnun

Param server: shell cmds

- A Parameter Server is a shared dictionary
- ROS uses to store global parameters/constants.
- Can be accesed by nodes
- The terminal cmds to play with them:

```
> rosparam set param value →set parameter
> rosparam get  param → get parameter
> rosparam load file → load parameters from file
> rosparam dump file → dump parameters to file
> rosparam delete param → delete parameter
> rosparam list  list parameter names
```

5

Checking param list

```
~/catkin_ws$ rosparam list
/rosdistro
/roslaunch/uris/host_rosvirtualserver_virtual_machine__32787
/roslaunch/uris/host_rosvirtualserver_virtual_machine__33307
/roslaunch/uris/host_rosvirtualserver_virtual_machine__38671
/roslaunch/uris/host_rosvirtualserver_virtual_machine__41027
/roslaunch/uris/host_rosvirtualserver_virtual_machine__41479
/roslaunch/uris/host_rosvirtualserver_virtual_machine__45735
/roslaunch/uris/host_rosvirtualserver_virtual_machine__45761
/rosversion
/run_id
rosvirtualserver@rosvirtualserver-virtual-
machine:~/catkin_ws$ rosparam get /rosversion
'1.12.7'
```

6

Param server: parameter types

- 32-bit integers
- booleans
- strings
- doubles
- iso8601 dates
- lists
- base64-encoded binary data

ROS

7

Param server: getting from python

```
global_name = rospy.get_param("/global_name")
relative_name = rospy.get_param("relative_name")
private_param = rospy.get_param('~private_name')
default_param = rospy.get_param('default_param', 'default_value')

# fetch a group (dictionary) of parameters
gains = rospy.get_param('gains')
p, i, d = gains['P'], gains['I'], gains['D']
```

25/06/2018 8

Param server: setting from python

```
# Using yaml strings
rospy.set_param('a_string', 'baz')
rospy.set_param('~private_int', '2')
rospy.set_param('list_of_floats', "[1., 2., 3., 4.]")
rospy.set_param('bool_True', "true")rospy.set_param('gains', "{'p': 1, 'i':
2, 'd': 3}")
# Using raw python objects
rospy.set_param_raw('a_string', 'baz')
rospy.set_param_raw('~private_int', 2)
rospy.set_param_raw('list_of_floats', [1., 2., 3., 4.])
rospy.set_param_raw('bool_True', True)
rospy.set_param_raw('gains', {'p': 1, 'i': 2, 'd': 3})
rospy.get_param('gains/P') #should return 1
```

ROS

9

Param server: exercise

- try to set and get parameters using the Python code from previous slides
- try the terminal commands as well

```
> rosparam set param value →set parameter
> rosparam get      param → get parameter
> rosparam load file → load parameters from file
> rosparam dump file → dump parameters to file
> rosparam delete param → delete parameter
> rosparam list     list parameter names
```

ROS

10

Param server: definition in a YAML file

- YAML is a lightweight markup language that supports all parameter types
- Config.yaml

```
camera:
left:
name: left_camera
exposure: 1
right:
name: right_camera
exposure: 1.1
```

- package.launch

```
<launch>
        <node name="name" pkg="package" type="node_type">
                <rosparam command="load"
                        file="$(find package)/config/config.yaml" />
        </node>
</launch>
```

ROS

11

Param server: yaml file→yaml format

- YAML uses "tags" to override types, where the YAML syntax may be ambiguous
- Common tags

```
!!str, !!int, !!float, !!seq, !!map
```

ROS

12