AA274A Section 7: ROS Parameters

Brian Dobkowski bdobkows@stanford.edu
AA274A, Stanford University

1. Problem 1

The command *rosparam list* shows all the ROS parameters currently on the ROS server. Here is a subset of the active ROS parameters after launching the autonomy stack on the turtlebot:

```
aa274@aa274-taro:~/catkin_ws/src/asl_turtlebot$ rosparam list
/map
/navigator/k1
/navigator/k2
/navigator/k3
/raspicam_node/ISO
/raspicam_node/awb_mode
/raspicam_node/brightness
/raspicam_node/camera_frame_id
/raspicam_node/camera_info_url
/raspicam_node/contrast
/raspicam_node/exposureCompensation
/raspicam_node/exposure_mode
/raspicam_node/framerate
/raspicam_node/hFlip
/raspicam_node/height
/raspicam_node/quality
/raspicam_node/saturation
/raspicam_node/sharpness
/raspicam_node/shutterSpeed
/raspicam_node/shutter_speed
/raspicam_node/tf_prefix
/raspicam_node/vFlip
/raspicam_node/videoStabilisation
/raspicam_node/width
/raspicam_node/zoom
/robot_description
/robot_state_publisher/publish_frequency
/rosdistro
/roslaunch/uris/host_192_168_1_53__35187
/roslaunch/uris/host_missgalaxy_local__33523
/rosversion
/run_id
/rviz/compressed/mode
/sim
/turtlebot3_core/baud
/turtlebot3_core/port
/turtlebot3_core/tf_prefix
/turtlebot3_slam_gmapping/angularUpdate
/turtlebot3_slam_gmapping/astep
/turtlebot3_slam_gmapping/base_frame
/turtlebot3_slam_gmapping/delta
/turtlebot3_slam_gmapping/iterations
```

Figure 1: Active ROS params on the server

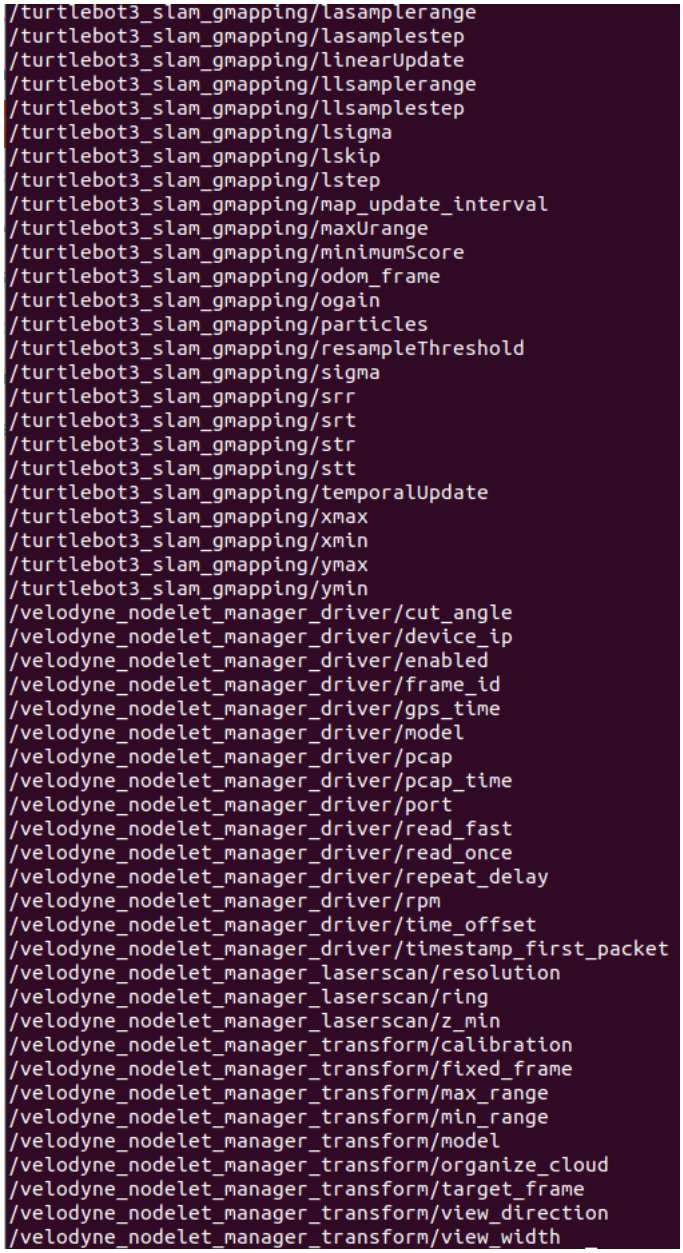

Figure 2: Active ROS params on the server

2. Problem 2

These are some of the launch files run when *section7.launch* is run, and some of the ROS params they set:

1. *root.launch*: sets params 'sim', 'map', 'rviz'
2. *gmapping.launch*: sets params such as 'base_frame', 'odom_frame', 'srr', 'llsamplerange', 'astep', and many more

3. *empty_world.launch*: within the Gazebo ROS package, launches more ROS params

3. Problem 3

Some ROS parameters found by running *rosparam get*:

1. *sim*: false

2. */velodyne_nodelet_manager_transform/model*: VLP16

3. */turtlebot3_slam_gmapping/srt*: 0.02

4. Problem 4

rostb3.sh | section7.launch | root.launch | gmapping_config.launch | section7.launch

```
<launch>
  <arg name="sim" default="true"/>

  <include file="$(find asl_turtlebot)/launch/root.launch">
    <arg name="world" value="project_city" />
    <arg name="x_pos" default="3.15"/>
    <arg name="y_pos" default="1.6"/>
    <arg name="z_pos" default="0.0"/>
    <arg name="rviz" default="section4"/>
    <arg name="model" default="asl_turtlebot"/>
    <arg name="sim" default="$(arg sim)"/>
  </include>

  <node pkg="asl_turtlebot" type="navigator.py" name="navigator" output="screen" >
    <param name="v_max" value="0.2"/>
    <param name="om_max" value="0.4"/>
  </node>
</launch>
```

Figure 3: Changes to section7.launch for params

5. Problem 5

```
# Robot limits
self.v_max = rospy.get_param("~v_max")  # maximum velocity
self.om_max = rospy.get_param("~om_max")  # maximum angular velocity
print(self.v_max)
```

Figure 4: Changes to navigator.py for params

6. Problem 6

rostb3.sh | section7.launch | root.launch | gmapping_config.launch | navigator.py

```
<launch>
  <arg name="sim" default="true"/>
  <arg name="v_max" default="0.2"/>
  <arg name="om_max" default="0.4"/>

  <include file="$(find asl_turtlebot)/launch/root.launch">
    <arg name="world" value="project_city" />
    <arg name="x_pos" default="3.15"/>
    <arg name="y_pos" default="1.6"/>
    <arg name="z_pos" default="0.0"/>
    <arg name="rviz" default="section4"/>
    <arg name="model" default="asl_turtlebot"/>
    <arg name="sim" default="$(arg sim)"/>
  </include>

  <node pkg="asl_turtlebot" type="navigator.py" name="navigator" output="screen" >
    <param name="v_max" value="$(arg v_max)"/>
    <param name="om_max" value="$(arg om_max)"/>
  </node>
</launch>
```

Figure 5: Section7.launch file

root.launch | gmapping_config.launch | section7.launch | navigator.py | supervisor.py | sectio

```
<launch>

  <include file="$(find asl_turtlebot)/launch/sections/section7.launch">
    <arg name="v_max" value="0.0001" />
    <arg name="om_max" value="0.00055"/>
  </include>

</launch>
```

Figure 6: Section7_slow.launch file

7. Problem 7

root.launch | gmapping_config.launch | navigator.py | supervisor.py | section7_slow.launch

```
#!/usr/bin/env python3
PACKAGE = "asl_turtlebot"

from dynamic_reconfigure.parameter_generator_catkin import *

gen = ParameterGenerator()

gen.add("k1",       double_t,    0, "Pose Controller k1", 0.8,  0., 2.0)
gen.add("k2",       double_t,    0, "Pose Controller k2", 0.4,  0., 2.0)
gen.add("k3",       double_t,    0, "Pose Controller k3", 0.4,  0., 2.0)
gen.add("kpx",      double_t,    0, "Trajectory Tracking kpx", 1.5, 0., 4.0)


exit(gen.generate(PACKAGE, "navigator", "Navigator"))

```

Figure 7: Navigator.cfg file with new param kpx

```
def dyn_cfg_callback(self, config, level):
    print(config)
    rospy.loginfo(
        "Reconfigure Request: k1:{k1}, k2:{k2}, k3:{k3}, kpx:{kpx}".format(**config)
    )
    self.pose_controller.k1 = config["k1"]
    self.pose_controller.k2 = config["k2"]
    self.pose_controller.k3 = config["k3"]
    self.kpx = config["kpx"]
    return config
```

Figure 8: navigator.py callback function