Notes on TurtleBot Command Velocity Multiplexer

References

- Kobuki Control System from the ROS Wiki:

  http://wiki.ros.org/kobuki/Tutorials/Kobuki%27s%20Control%20System

  !!! Note that the diagram in this wiki page lists incorrect topics !!!

- Chapter 3.2.2 "Velocity Command Multiplexer" from
  - https://cse.sc.edu/~jokane/teaching/574/notes-turtlebot.pdf

Principle

As you control the robot, you may have several different nodes that want to publish cmd_vel messages. Which one should have control of the robot? If everyone publishes directly to cmd_vel, then the robot will always try to execute the command in the most recent message it has received. This is obviously not a good solution if, for example, you'd like to take teleoperative control of the robot to override automatically generated commands sent by your software.

ROS provides a solution to this problem in the form of a multiplexer node. Each node that wants to send movement commands to the robot, instead of publishing directly to cmd_vel, publishes messages on one of three different topics:

- /cmd_vel_mux/input/navi
- /cmd_vel_mux/input/teleop
- /cmd_vel_mux/input/safety_controller

When messages arrive on any of these topics, cmd_vel_mux decides which should take the highest priority, and forwards the corresponding messages to the turtlebot node via cmd vel.

Configuration and launch files

The command velocity multiplexer is launched by the ***minimal.launch*** from the turtlebot_bringup package.

The ***minimal.launch*** file includes the ***mobile_base.launch.xml*** file with “base = kobuki” since we are using the kobuki base:

```
<launch>
  <arg name="base"       default="$(optenv TURTLEBOT_BASE kobuki)"/>  <!-- create,
rhoomba -->
  <arg name="battery"    default="$(optenv TURTLEBOT_BATTERY
/sys/class/power_supply/BAT0)"/>  <!-- /proc/acpi/battery/BAT0 in 2.6 or earlier
kernels-->
  <arg name="stacks"     default="$(optenv TURTLEBOT_STACKS hexagons)"/>  <!--
circles, hexagons -->
  <arg name="3d_sensor"  default="$(optenv TURTLEBOT_3D_SENSOR kinect)"/>  <!--
kinect, asus_xtion_pro -->
  <arg name="simulation" default="$(optenv TURTLEBOT_SIMULATION false)"/>

  <param name="/use_sim_time" value="$(arg simulation)"/>

  <include file="$(find turtlebot_bringup)/launch/includes/robot.launch.xml">
    <arg name="base" value="$(arg base)" />
    <arg name="stacks" value="$(arg stacks)" />
    <arg name="3d_sensor" value="$(arg 3d_sensor)" />
  </include>
  <include file="$(find turtlebot_bringup)/launch/includes/mobile_base.launch.xml">
    <arg name="base" value="$(arg base)" />
  </include>
  <include file="$(find turtlebot_bringup)/launch/includes/netbook.launch.xml">
    <arg name="battery" value="$(arg battery)" />
  </include>
</launch>
```

The ***…/turtlebot_bringup/launch/includes/mobile_base.launch.xml*** file:

```
Selector for the base.
<launch>
  <arg name="base"/>
  <include file="$(find turtlebot_bringup)/launch/includes/$(arg
base)/mobile_base.launch.xml"/>
</launch>
```

includes the ***kobuki/mobile_base.xml*** file:

```
<!--
  Kobuki's implementation of turtlebot's mobile base.
 -->
<launch>
  <node pkg="nodelet" type="nodelet" name="mobile_base_nodelet_manager"
args="manager"/>
  <node pkg="nodelet" type="nodelet" name="mobile_base" args="load
kobuki_node/KobukiNodelet mobile_base_nodelet_manager">
    <rosparam file="$(find kobuki_node)/param/base.yaml" command="load"/>
    <remap from="mobile_base/odom" to="odom"/>
    <!-- Don't do this - force applications to use a velocity mux for redirection
      <remap from="mobile_base/commands/velocity" to="cmd_vel"/>
    -->
    <remap from="mobile_base/enable" to="enable"/>
    <remap from="mobile_base/disable" to="disable"/>
    <remap from="mobile_base/joint_states" to="joint_states"/>
  </node>

  <!-- velocity commands multiplexer -->
  <node pkg="nodelet" type="nodelet" name="cmd_vel_mux" args="load
yocs_cmd_vel_mux/CmdVelMuxNodelet mobile_base_nodelet_manager">
    <param name="yaml_cfg_file" value="$(find turtlebot_bringup)/param/mux.yaml"/>
    <remap from="cmd_vel_mux/output" to="mobile_base/commands/velocity"/>
  </node>

  <!-- bumper/cliff to pointcloud -->
  <include file="$(find
turtlebot_bringup)/launch/includes/kobuki/bumper2pc.launch.xml"/>
</launch>
```

In this file a nodelet is started to run the velocity commands multiplexer:

```
<!-- velocity commands multiplexer -->
<node pkg="nodelet" type="nodelet" name="cmd_vel_mux" args="load
yocs_cmd_vel_mux/CmdVelMuxNodelet mobile_base_nodelet_manager">
    <param name="yaml_cfg_file" value="$(find turtlebot_bringup)/param/mux.yaml"/>
    <remap from="cmd_vel_mux/output" to="mobile_base/commands/velocity"/>
</node>
```

The order of priority of the multiplexer is defined in the configuration file ***.../turtlebot_bringup/param/mux.yaml***:

```
# Created on: Oct 29, 2012
#     Author: jorge
# Configuration for subscribers to multiple cmd_vel sources.
#
# Individual subscriber configuration:
#   name:           Source name
#   topic:          The topic that provides cmd_vel messages
#   timeout:        Time in seconds without incoming messages to consider this
topic inactive
#   priority:       Priority: an UNIQUE unsigned integer from 0 (lowest) to MAX_INT
#   short_desc:     Short description (optional)

subscribers:
  - name:        "Safe reactive controller"
    topic:       "input/safety_controller"
    timeout:     0.2
    priority:    10
  - name:        "Teleoperation"
    topic:       "input/teleop"
    timeout:     1.0
    priority:    7
  - name:        "Navigation"
    topic:       "input/navi"
    timeout:     1.0
    priority:    5
```

The default (given) priority order makes sense, teleop over navigation, and safety controller above all.

NB: As such, the topics listed there may be used. Effectively, the command velocity may be remapped to one of the multiplexer input topics for the purposes of the lab as we saw.

Safety controller

The file ***mobile_base.launch.xml*** (path shown above) lists the nodes run by **turtlebot_bringup**.

The safety controller node can be added with the guidelines provided in the following reference: http://answers.ros.org/question/57730/bumpers-with-turtlebot-2/.

This takes care of the remapping arguments for the command velocity input, as well as the events (bumper - goes back, cliff - stops turning wheels).

Testing

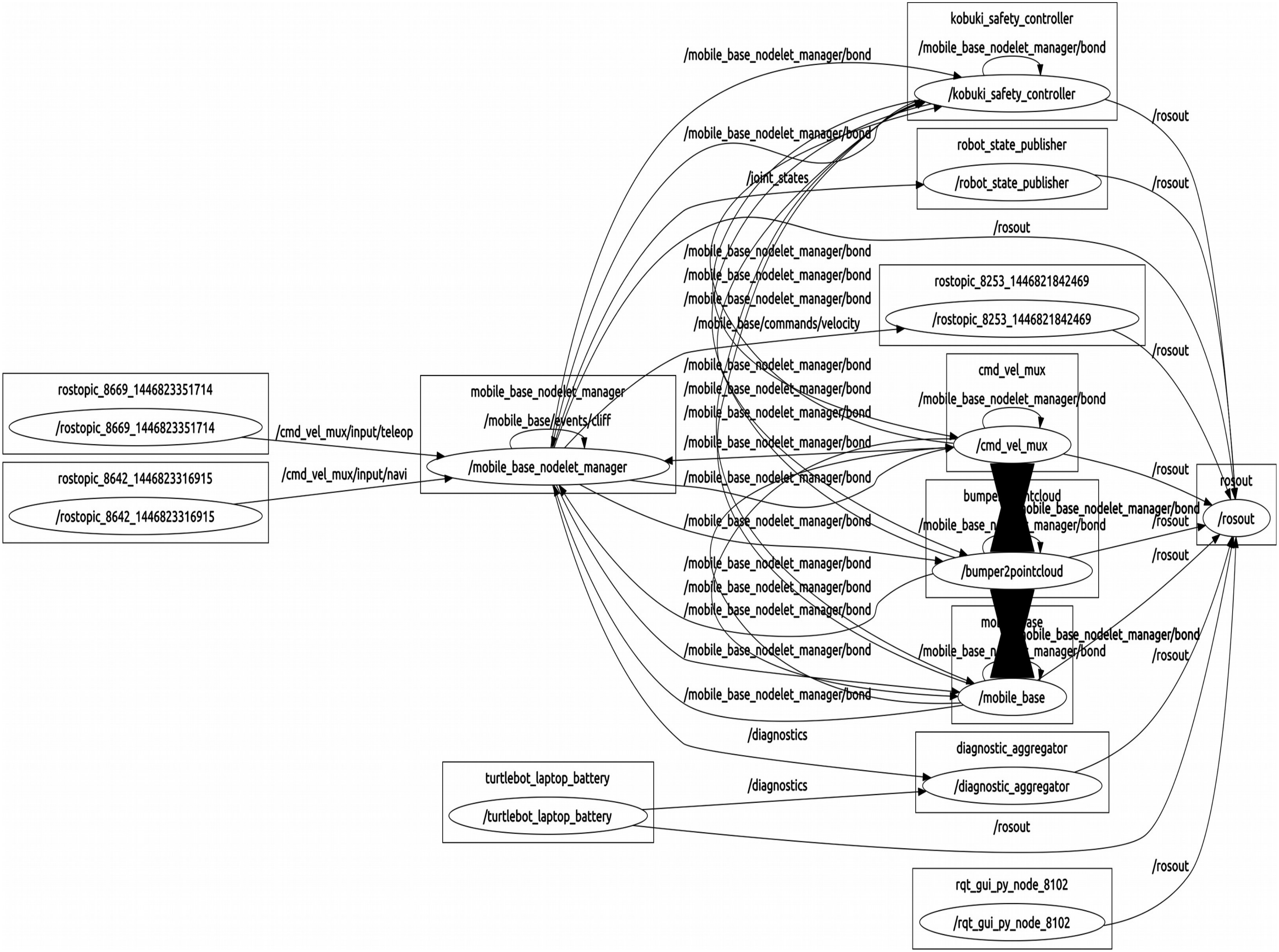
Validation can be performed as follows:

- Publish velocity for reverse movement in **/cmd_vel_mux/input/navi**
  - Turtlebot moves backwards
- Publish forward velocity on **/cmd_vel_mux/input/teleop**, which has higher priority.
  - Turtlebot changes directions to move forwards
- The topic **/mobile_base/commands/velocity** shows the output of the multiplexer, i.e. the velocity finally driving the turtlebot
- Mobile base events (bumper, cliff) behaved as documented in the tests.

The rosgraph of this experiment is presented in the next page.

Note 1: publishing a rotation command velocity on the input/navi topic, there's a noticeable delay till execution, which does not seem to be the case with translation velocities. Maybe there's a reason behind this, or it's just an error in testing, need to be checked and updated later …

Note 2: In fact, a more elegant solution to incorporating the safety controller is to simply include the already existing safety_controller.launch.xml file in the higher mobile_base.launch.xml file, both found on the same path. The one I posted before and this are exactly the same in incorporating the safety controller, only different names are used for this node.


kobuki_safety_controller
/mobile_base_nodelet_manager/bond
/kobuki_safety_controller
/mobile_base_nodelet_manager/bond
/mobile_base_nodelet_manager/bond
/rosout
robot_state_publisher
/joint_states
/robot_state_publisher
/rosout
/rosout
/mobile_base_nodelet_manager/bond
/mobile_base_nodelet_manager/bond
/mobile_base_nodelet_manager/bond
rostopic_8253_1446821842469
/mobile_base/commands/velocity
/rostopic_8253_1446821842469
/rosout
/mobile_base_nodelet_manager/bond
/mobile_base_nodelet_manager/bond
/mobile_base_nodelet_manager/bond
cmd_vel_mux
/mobile_base_nodelet_manager/bond
/cmd_vel_mux
/mobile_base_nodelet_manager/bond
rostopic_8669_1446823351714
/rostopic_8669_1446823351714
/cmd_vel_mux/input/teleop
mobile_base_nodelet_manager
/mobile_base/events/cliff
/mobile_base_nodelet_manager
rostopic_8642_1446823316915
/rostopic_8642_1446823316915
/cmd_vel_mux/input/navi
/mobile_base_nodelet_manager/bond
/rosout
rosout
/rosout
bumper2pointcloud
/mobile_base_nodelet_manager/bond
/mobile_base_nodelet_manager/bond
/rosout
/mobile_base_nodelet_manager/bond
/bumper2pointcloud
/rosout
/mobile_base_nodelet_manager/bond
/mobile_base_nodelet_manager/bond
/mobile_base_nodelet_manager/bond
mobile_base
/mobile_base_nodelet_manager/bond
/mobile_base_nodelet_manager/bond
/mobile_base_nodelet_manager/bond
/rosout
/mobile_base_nodelet_manager/bond
/mobile_base
/diagnostics
diagnostic_aggregator
turtlebot_laptop_battery
/diagnostics
/diagnostic_aggregator
/turtlebot_laptop_battery
/rosout
/rosout
rqt_gui_py_node_8102
/rqt_gui_py_node_8102