Turtlebot2i simulation environment setup instructions

Author: Zuchen Zhang

If you don't want to go through this tedious work log file, you can download the Turtlebot2i simulation environment from my Github repository, and follow the instructions in README to use it. My Github repository address:

https://github.com/Roboticist999/turtlebot2i_simulaition_environment

1. If you are using VMware and Gazebo crashes on start, try to uncheck Acceletation 3D Graphics in the setting of virtual machine display

http://answers.gazebosim.org/question/13214/virtual-machine-not-launching-gazebo/

https://kb.vmware.com/s/article/59146

I'm using VMware workstation, and it works on my virtual machine. But by unchecking this, the gazebo simulation graphics could be not fluent.

For Fusion:

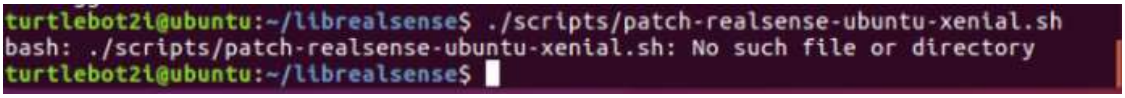
1. Shutdown the Virtual Machine.
2. From the VMware Fusion menu bar, select **Window** > **Virtual Machine Library**.
3. Select a virtual machine and click **Settings**.
4. In the **Settings** Window, in the **System Settings** section, select **Display**.
5. Uncheck **Accelerate 3D graphics**.

For Workstation:

1. Shutdown the virtual machine.
2. Select the virtual machine and select **VM** > **Settings**.
3. On the **Hardware** tab, select **Display**.
4. Uncheck **Accelerate 3D graphics**.
5. Click **OK**.

2. Do commands I marked green and yellow in Full Build Instructions · Interbotix_turtlebot2i Wiki.pdf

When doing RealSense package installation, I meet this problem:



And I find a explanation that says we can omit it
https://github.com/IntelRealSense/librealsense/issues/2712

3. Now if you run `roslaunch turtlebot2i_gazebo turtlebot2i_world.launch`, you will meet this problem:

```
[WARN] [1583684534.209884, 0.000000]: Controller Spawner couldn't find the expec
ted controller_manager ROS interface.
[turtlebot/controller_spawner-5] process has finished cleanly
log file: /home/turtlebot2i/.ros/log/e5150fa0-6158-11ea-8593-000c29704efd/turtle
bot-controller_spawner-5*.log
[ INFO] [1583684597.169451054, 0.024000000]: waitForService: Service [/gazebo/se
t_physics_properties] is now available.
[ INFO] [1583684597.170404371, 0.026000000]: waitForService: Service [/gazebo/se
t_physics_properties] is now available.
[ INFO] [1583684597.326663890, 0.188000000]: Physics dynamic reconfigure ready.
[ INFO] [1583684597.339738038, 0.200000000]: Physics dynamic reconfigure ready.
```

I found solution in the following website:https://github.com/ros-simulation/gazebo_ros_demos/issues/30

So I did 2 steps, from http://gazebosim.org/tutorials?tut=ros_installing

- ROS Kinetic:

```
sudo apt-get install ros-kinetic-gazebo-ros-pkgs ros-kinetic-gazebo-ros-control
```

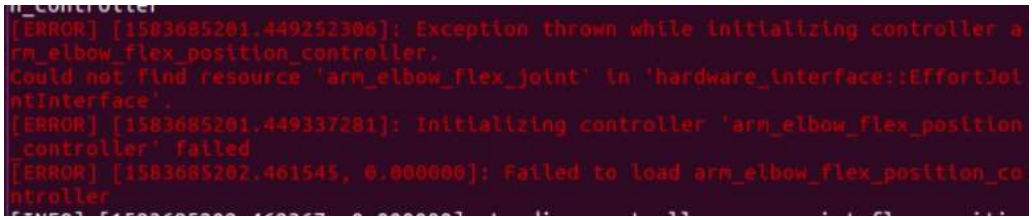
From https://github.com/ros-simulation/gazebo_ros_demos/issues/30

cpuwanan commented on 5 Apr 2019

This could be helpful
$ sudo apt-get install ros-kinetic-joint-state-controller
$ sudo apt-get install ros-kinetic-effort-controllers
$ sudo apt-get install ros-kinetic-position-controllers

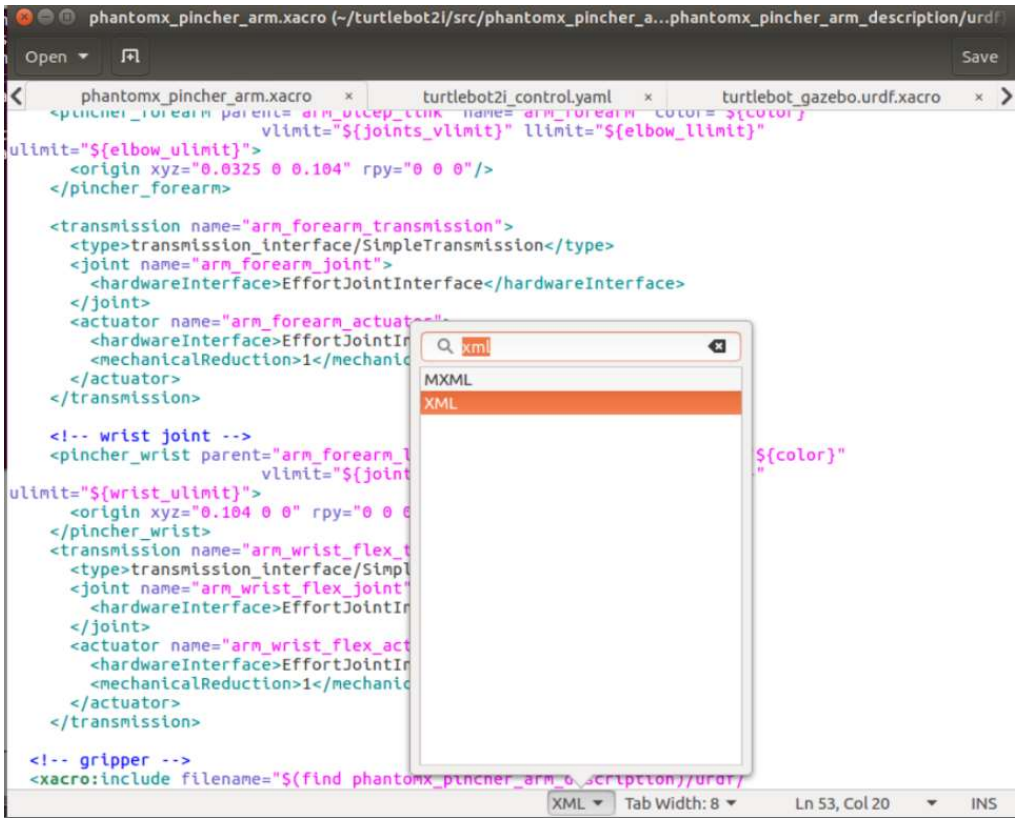
5 2

4. If you run `roslaunch turtlebot2i_gazebo turtlebot2i_world.launch again`, you will meet several error of this kind:

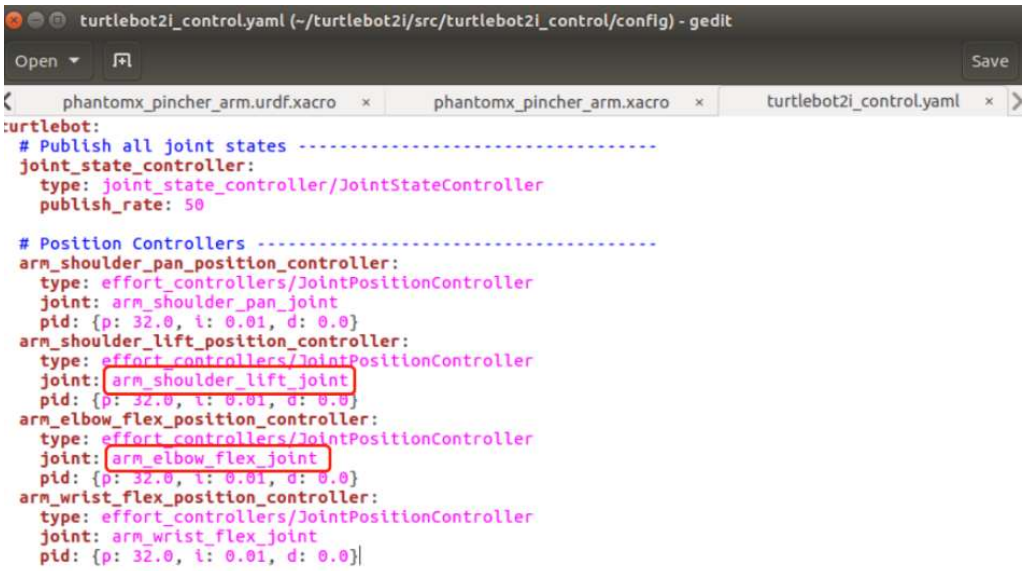


This is because some controller names don't match in ROS files provided in Turtlebot2i packages.

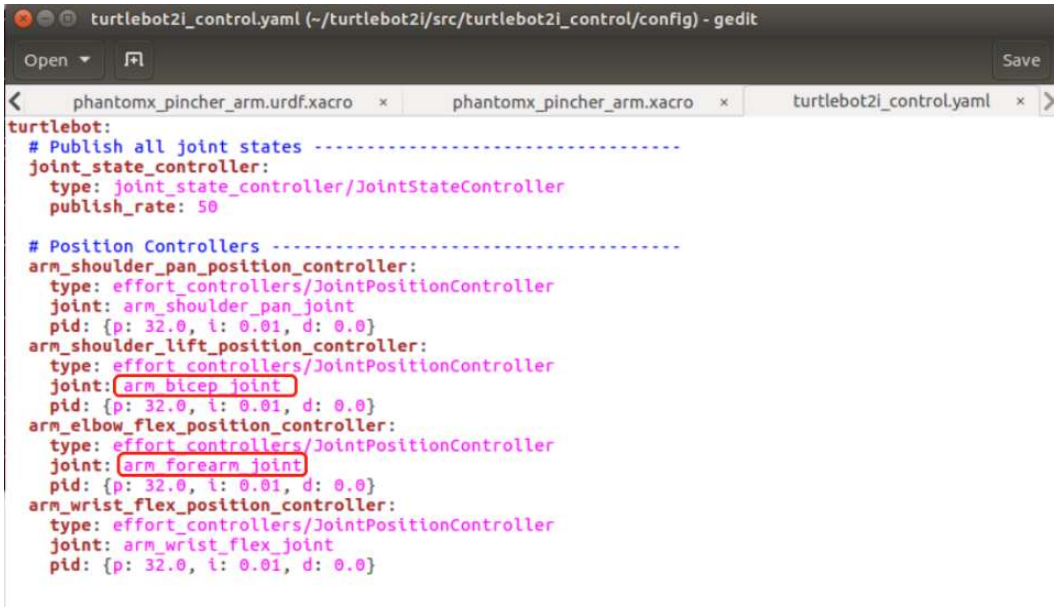
Tpye `gedit $(rospack find phantomx_pincher_arm_description)/urdf/phantomx_pincher_arm.xacro` in a terminal, you can find the file defining joints of the phantomx_pincher_arm, set the viewing format to XML will help you view the file:



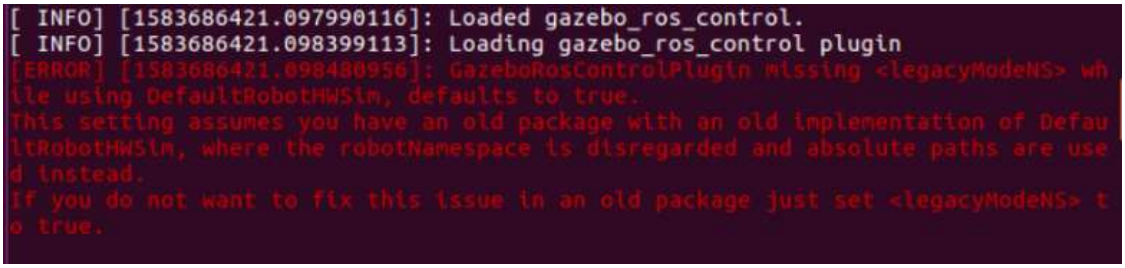
Open another terminal, type `gedit $(rospack find turtlebot2i_control)/config/turtlebot2i_control.yaml` you'll find how the arm controllers are defined, and you will find the joint names don't match…..



So change the 2 joint names according to the definition in phantomx_pincher_arm.xacro, and don't forget to save it.
Then it should look like this:

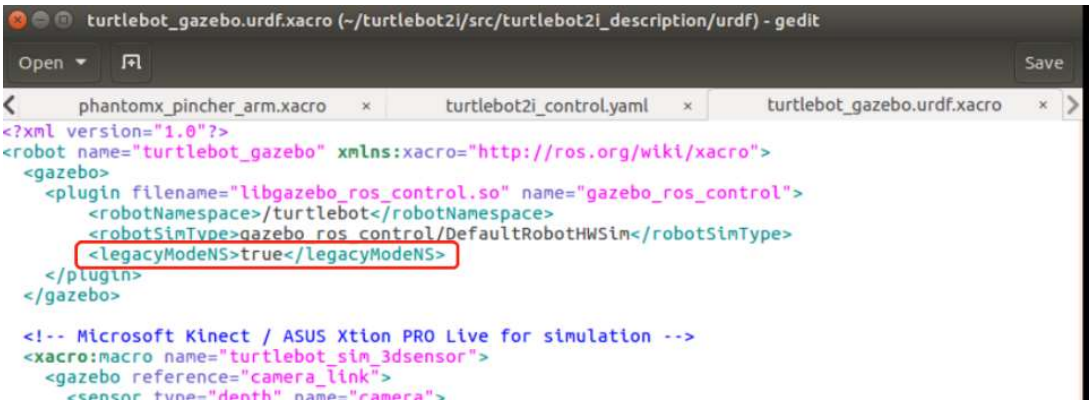


5. If you run `roslaunch turtlebot2i_gazebo turtlebot2i_world.launch` again, you will find an error like this:

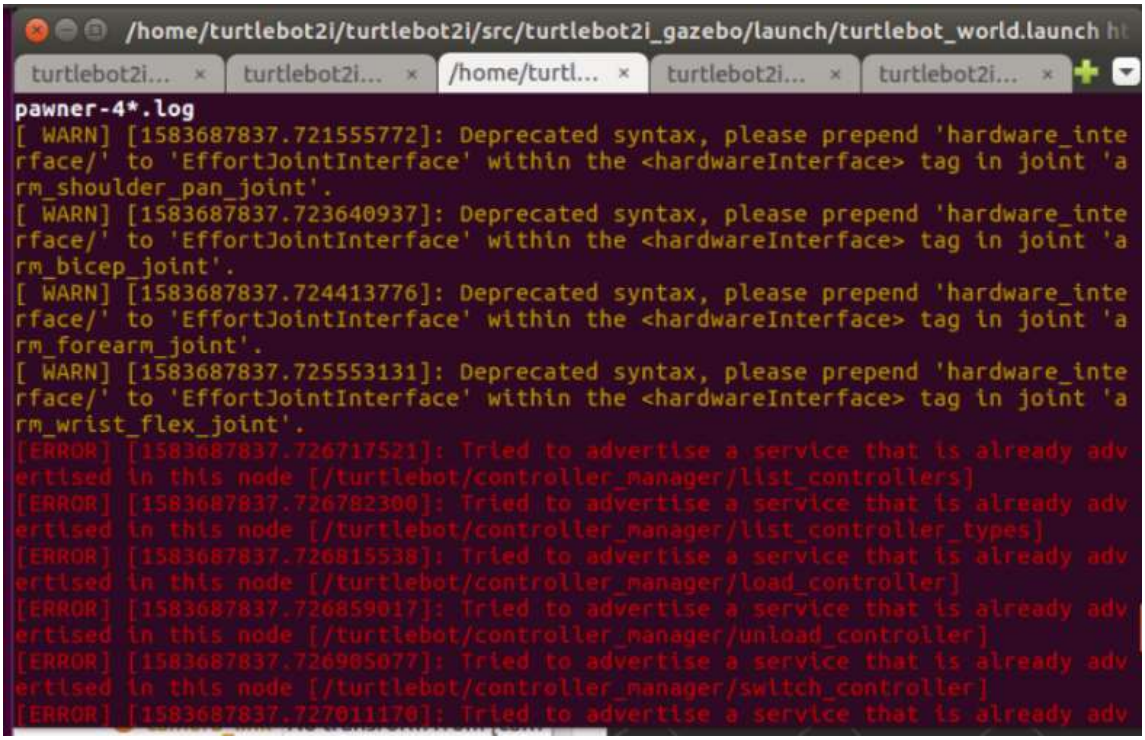


I found solution of this error on this website:https://answers.ros.org/question/292444/gazebo_ros_control-plugin-gazeboroscontrolplugin-missing-legacymodens-defaultrobothwsim/

Type `gedit ~/turtlebot2i/src/turtlebot2i_description/urdf/turtlebot_gazebo.urdf.xacro` in a terminal, and add 1 line of text as suggested by the answer above, and don't forget to save it.



6. Now if you run again, you will still see these error



For the orange error, I found explanation here. http://wiki.ros.org/urdf/XML/Transmission

The transmission has the following elements:

<type> (one occurrence)

Specifies the transmission type.

<joint> (one or more occurrences)

A joint the transmission is connected to. The joint is specified by its **name** attribute, and the following sub-elements:

<hardwareInterface> (one or more occurrences)

Specifies a supported joint-space hardware interface. Note that the value of this tag should be **EffortJointInterface** when this transmission is loaded in Gazebo and **hardware_interface/EffortJointInterface** when this transmission is loaded in RobotHW.

<actuator> (one or more occurrences)

An actuator the transmission is connected to. The actuator is specified by its **name** attribute, and the following sub-elements:

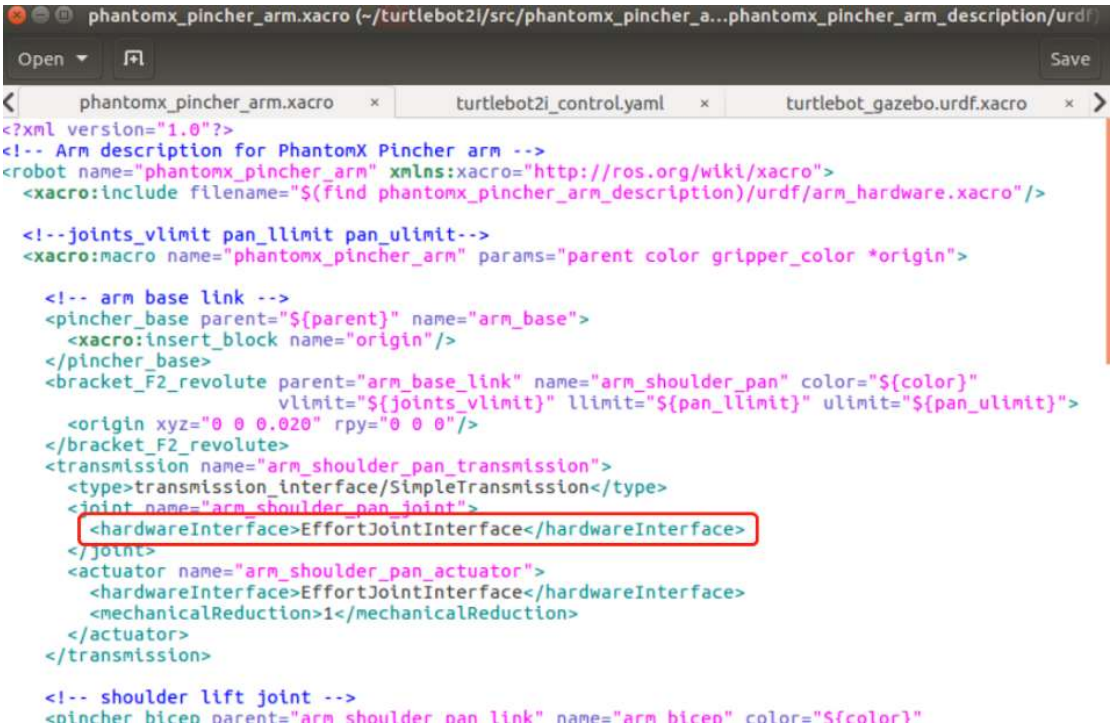
<mechanicalReduction> (optional)

Specifies a mechanical reduction at the joint/actuator transmission. This tag may not be needed for all transmissions.

<hardwareInterface> (optional) (one or more occurrences)

Specifies a supported joint-space hardware interface. Note that **<hardwareInterface>** tag should only be specified here for ROS releases prior to Indigo. The correct place to specify this tag is in **<joint>** tag. More details about this can be found here.

So it seems that it should be EffortJointInterface when we do simulation in Gazebo. And it is EffortJointInterface in phantomx_pincher_arm.xacro
So I just leave it here.



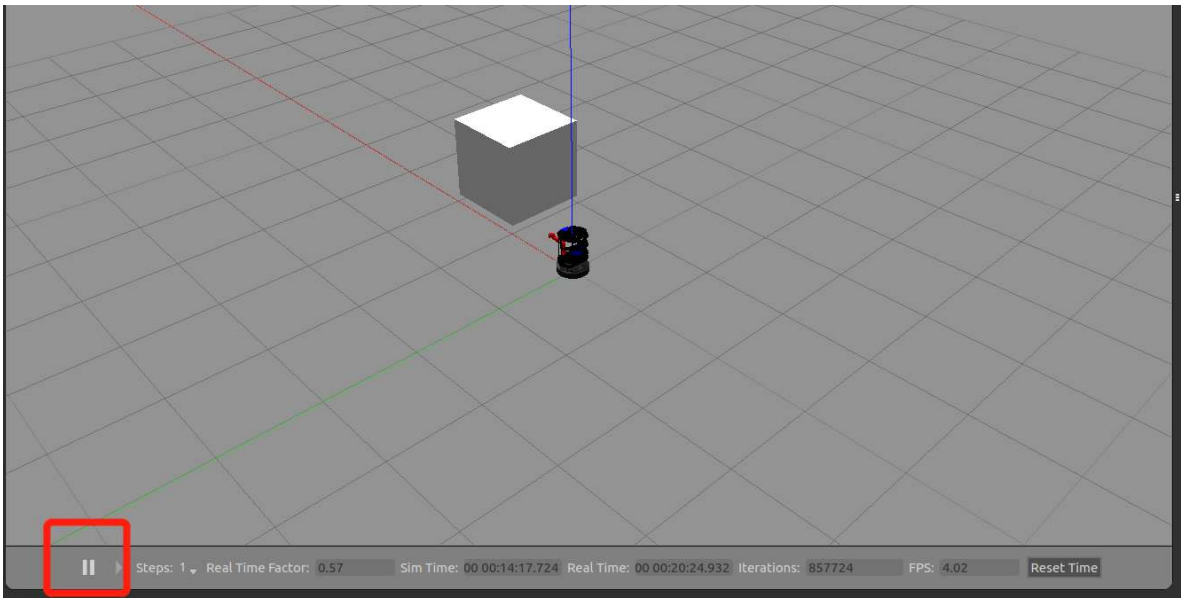
For the red error, I have figured out neither whether it will affect simulation nor its solution.

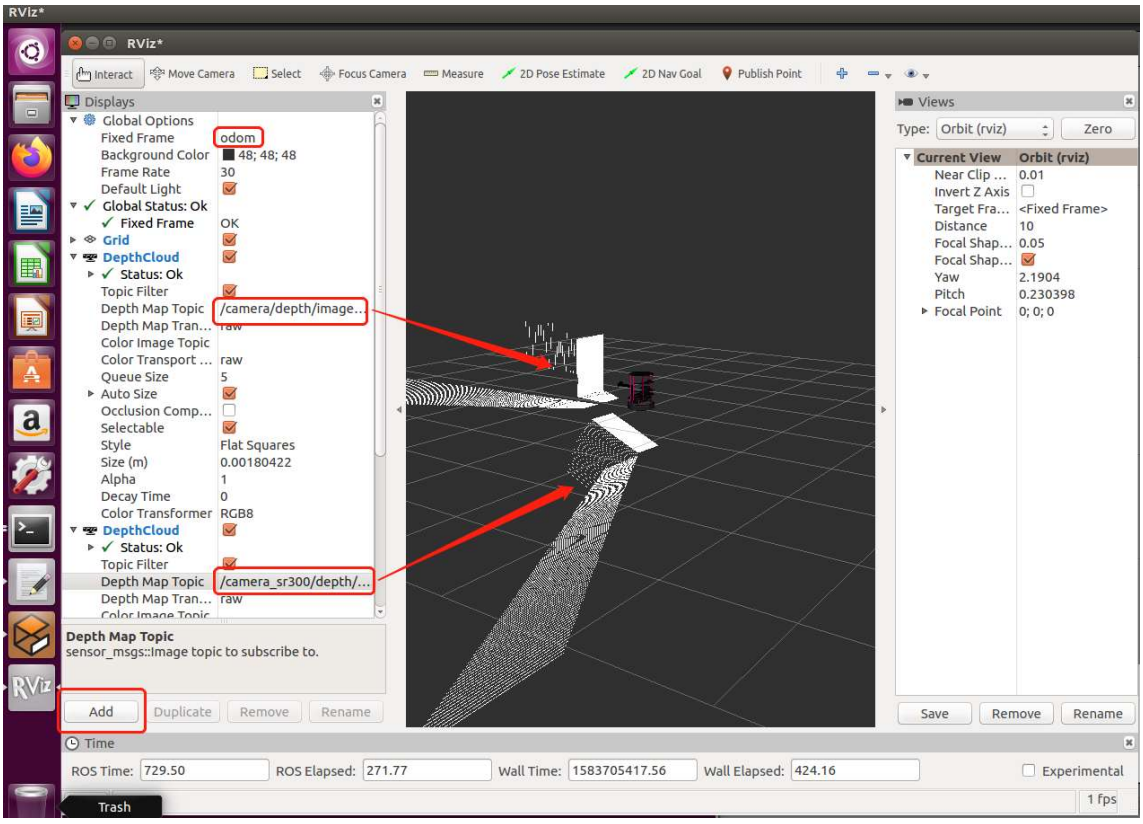
7. Now open Rviz, and subscribe to /camera/rgb/image_raw topic. You will find Gazebo die, code 134. Now you need to update Gazebo by typing following commands in terminal.

```
sudo sh -c 'echo "deb http://packages.osrfoundation.org/gazebo/ubuntu-stable `lsb_release -cs` main" > /etc/apt/sources.list.d/gazebo-stable.list'

wget http://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key add -

sudo apt-get update

sudo apt-get install gazebo7
```

Refertence https://stackoverflow.com/questions/55181205/gazebo-crashing-when-subscribing-to-scan>

8. Open Rviz using rosrun rviz rviz, and click add, then add the components as shown in the following figure, plus RobotModel which is not shown in the figure to visualize the robot. Then you can see 1 depth cloud is on the wrong direction. Make sure your Gazebo is not paused, or there will be no message published to topics. You can also change the “paused” argument to false in turtlebot_world.launch.

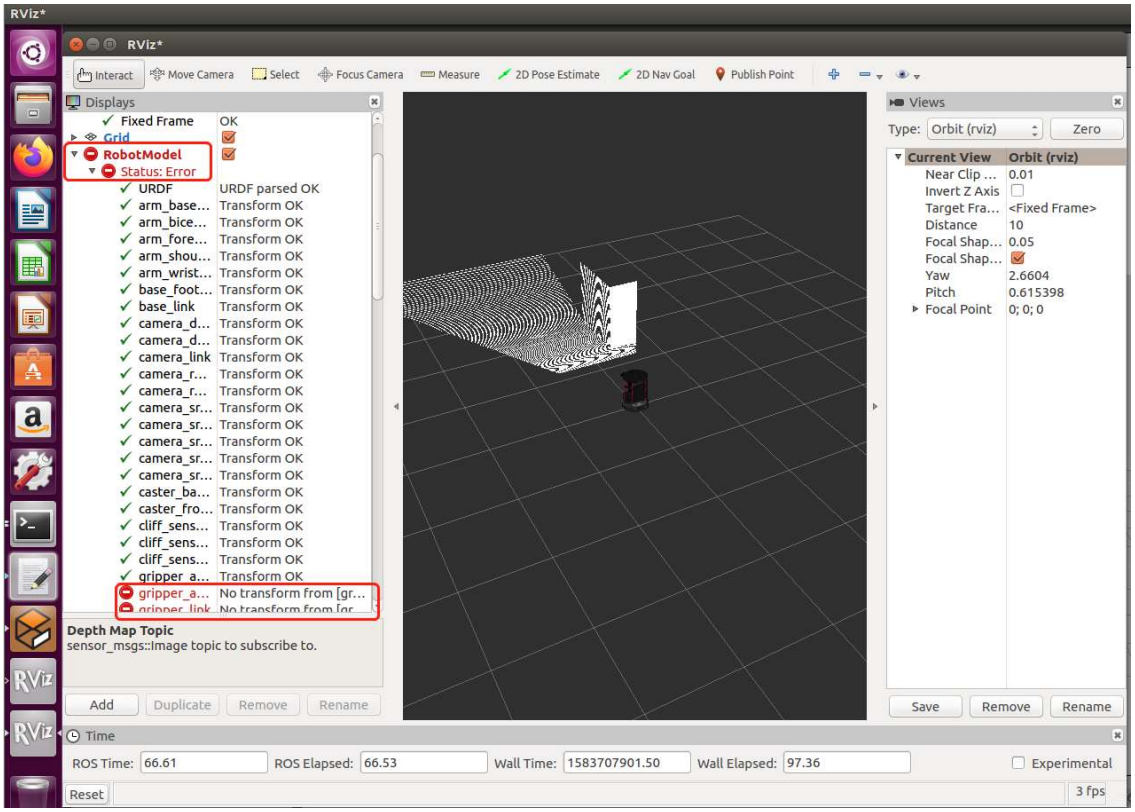




To fix the wrong orientation problem, type
gedit $(rospack find turtlebot2i_description)/urdf/sensors/sr300.urdf.xacro
in your termimal. Change the number "0" to "-1.16", make sure you are making modification for the right camera, then restart Gazebo and Rviz.

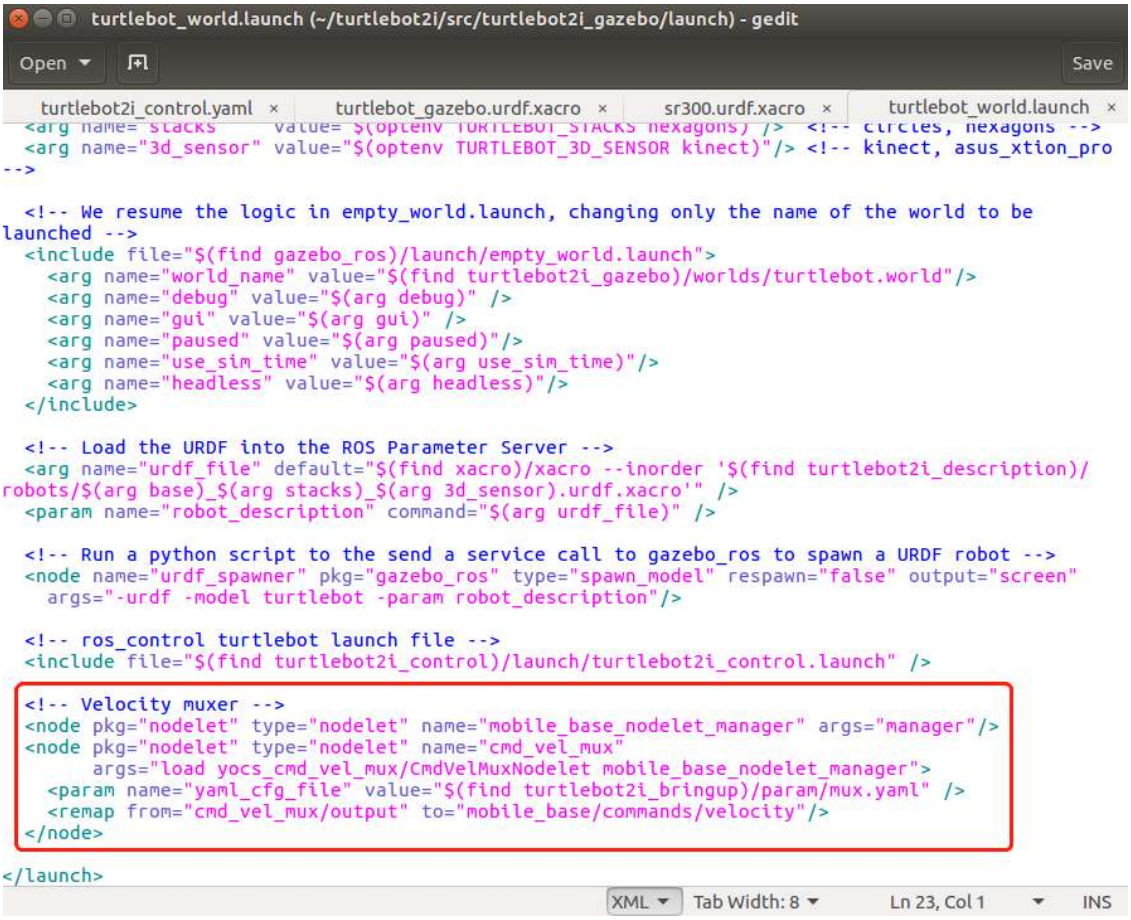
```
<!--
==  Depth joints & links ==
-->

<joint name="camera_sr300_depth_joint" type="fixed">
    <origin xyz="0 ${sr300_cam_depth_offset} 0" rpy="0 0 0" />
    <parent link="camera_sr300_rgb_frame" />
    <child link="camera_sr300_depth_frame" />
</joint>
<link name="camera_sr300_depth_frame"/>

<joint name="camera_sr300_depth_optical_joint" type="fixed">
    <origin xyz="0 0 0" rpy="${-M_PI/2} 0 ${-M_PI/2}" />
    <parent link="camera_sr300_depth_frame" />
    <child link="camera_sr300_depth_optical_frame" />
</joint>
<link name="camera_sr300_depth_optical_frame"/>
```

Now you can still see some error in RobotModel, which is that some frames doesn't have transformation. I have't figured out the solution yet, and I think it might won't affect my simulation at current stage.



9. Now you may want to generate a map using teleop by roslaunch turtlebot_teleop kayboard.launch, make sure you Gazebo is not paused, but you will find the robot doesn't move in Gazebo. To solve this, add the following test in your turtlebot_world.launch, before </launch> at the end. To reach this file, use gedit $(rospack find turtlebot2i_gazebo)/launch/turtlebot_world.launch

```
<!-- Velocity
muxer -->
        <node pkg="nodelet" type="nodelet"
        name="mobile_base_nodelet_manager" args="manager"/>
        <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
        turtlebot2i_bringup)/param/mux.yaml" />
        <remap from="cmd_vel_mux/output"
        to="mobile_base/commands/velocity"/>
        </node>
```

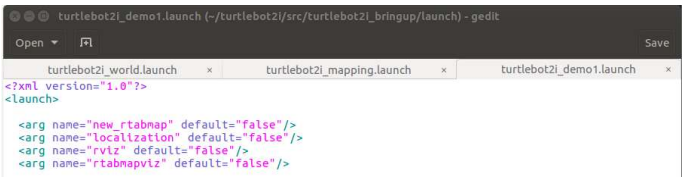


Then restart Gazebo, you will find the you can control the robot using keyboard.

10. Then I tried roslaunch turtlebot2i_bringup turtlebot_mapping.launch, and I've made some modifications but it still can't build map yet.

Tried to run the demo first

Changed parameters



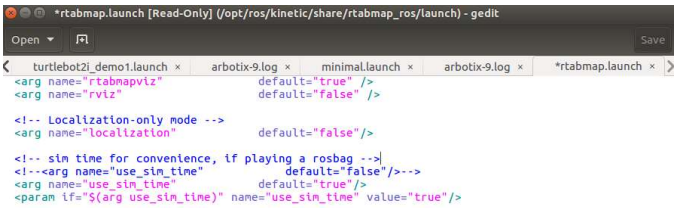
Run roslaunch turtlebot2i_bringup turtlebot2i_demo1.launch, and the program died, showing arbotix-9 died, so check the log file

After checking minimal.launch

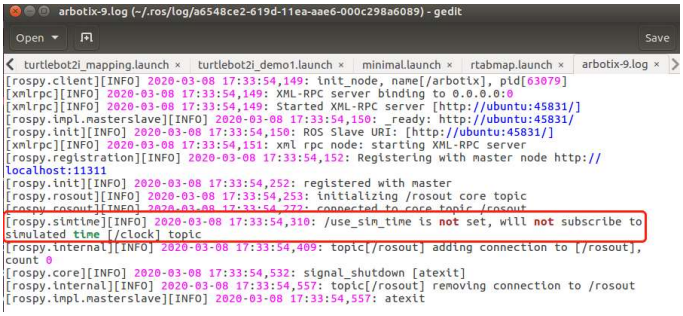
Run again, same error prompt, check the log file

grep -r "use_sim_time" "/opt/ros/kinetic/share/"

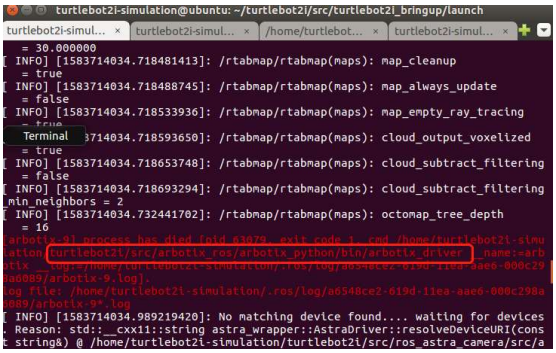
Check it, and comment the original one and set the default value to true



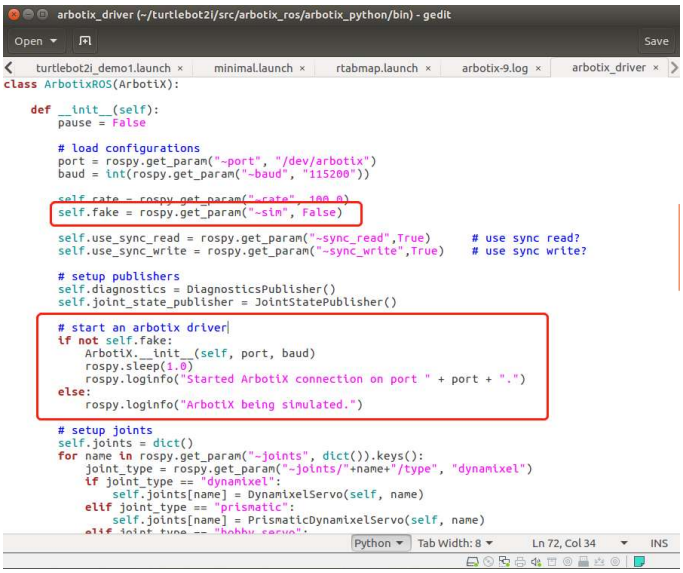
Run again still doesn't work, same log content again



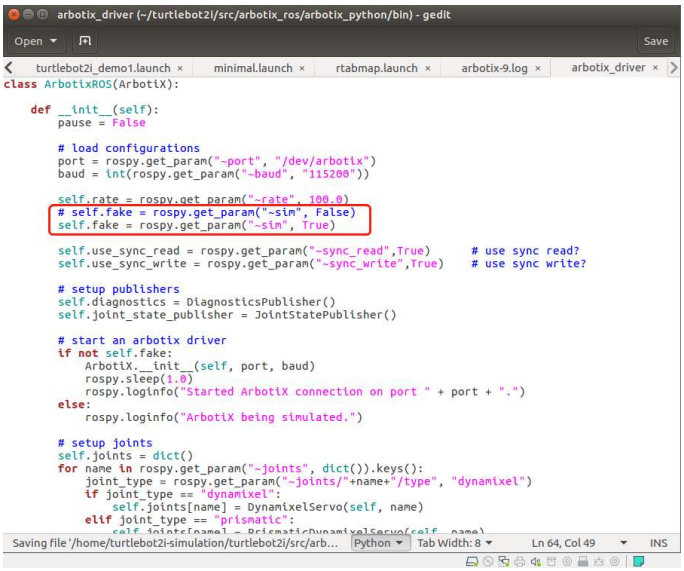
Noticed



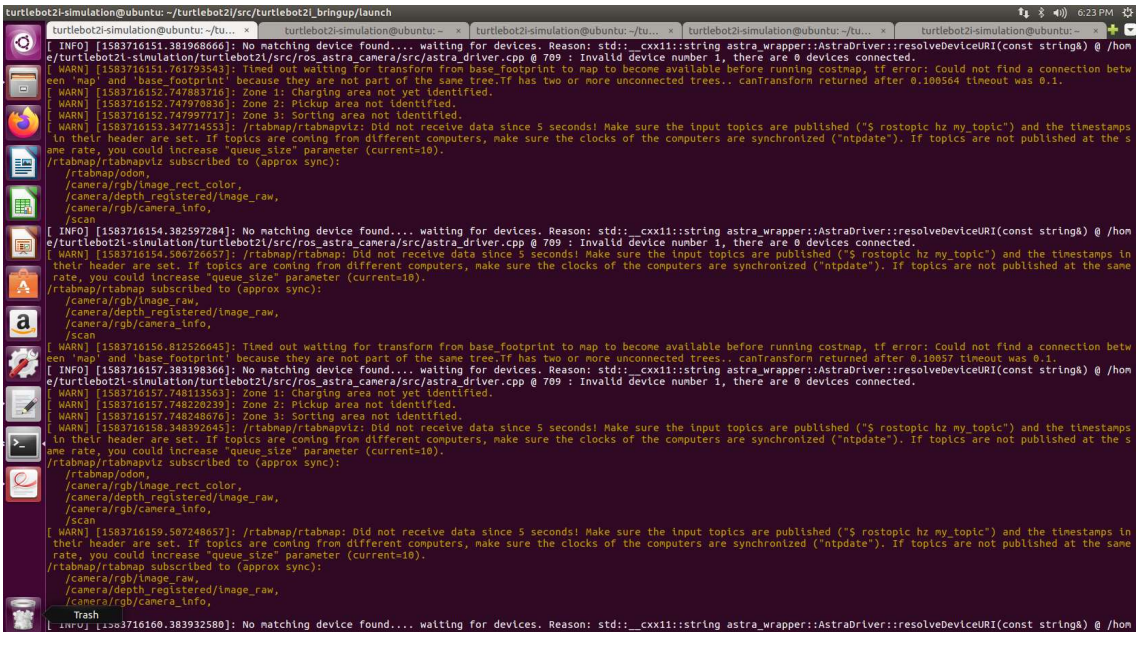
Check that file



Change the parameter to true



I decide to leave it here first, I remember there are redirectioning issues, I want to solve them then come back to this.



Searched for "no matching device found"
http://wiki.ros.org/astra_camera, it shows udev rule, so I want to try udev rules part on Full build instructions https://github.com/Interbotix/turtlebot2i/wiki/Full-Build-Instructions

Get stuck here

udev rules

```
cd ~/turtlebot2i/
```

```
goros
```

```
rosrun kobuki_ftdi create_udev_rules
```

```
rosrun astra_camera create_udev_rules
```

```
cd ~/turtlebot2i/src/turtlebot2i_misc
```

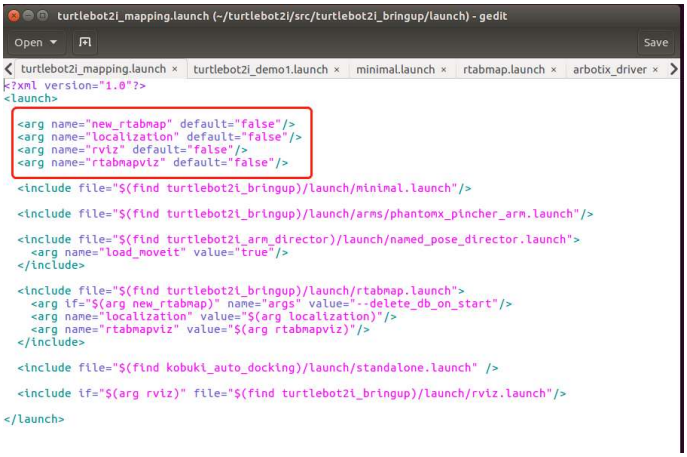
```
gedit 99-turtlebot2i.rules
```

- At this point we need to execute a couple commands that will return the serial number for the ArbotiX and the Kobuki. Remove and/or Re-plug both the Arbotix and the Kobuki USB cables and ensure they are powered on before running these commands:
- `udevadm info -a -n /dev/ttyUSB0 | grep '{serial}' | head -n1`
- `udevadm info -a -n /dev/ttyUSB1 | grep '{serial}' | head -n1`
- The output from these commands are the serial numbers used to fill in the blanks of lines 6 and 7 of the 99-turtlebot2i.rules file.

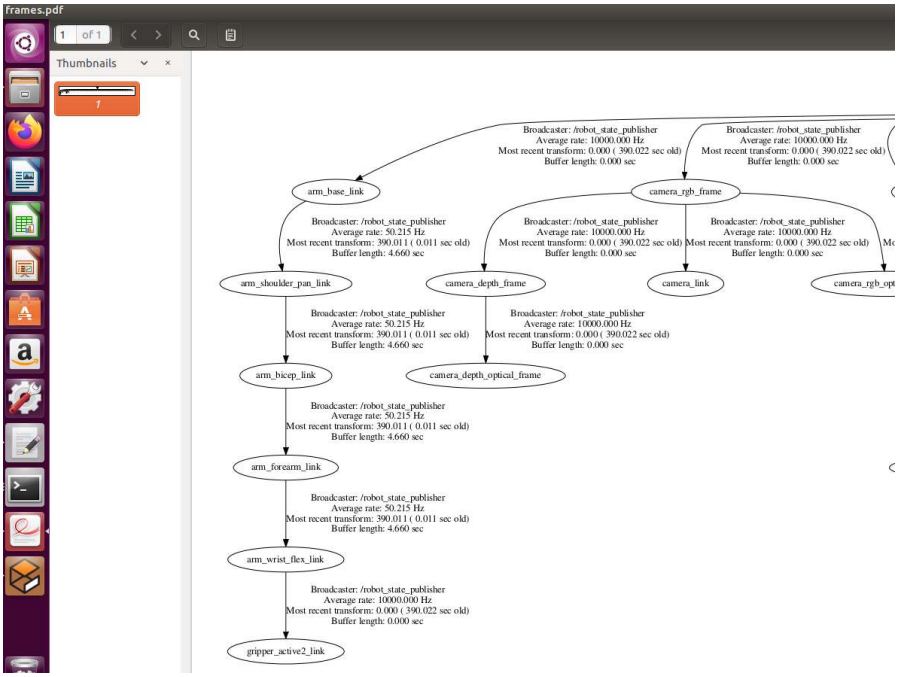
It seems that the demo1.launch and its dependenies are too complecated, so run mapping.launch

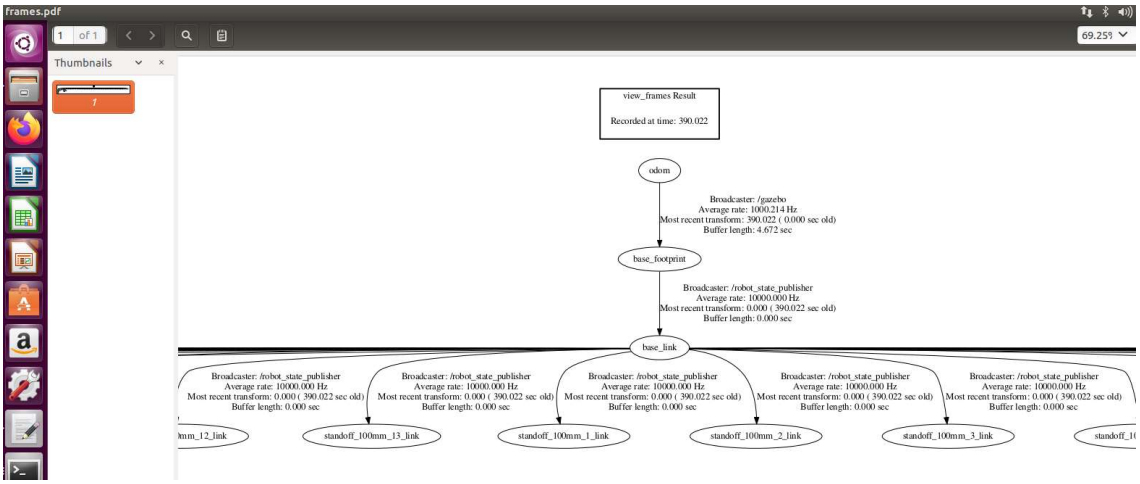
It seems that the demo1.launch and its dependenies are too complecated, so run mapping.launch

Modified these parameters


turtlebot2i_mapping.launch (~/turtlebot2i/src/turtlebot2i_bringup/launch) - gedit
Open
Save
turtlebot2i_mapping.launch
turtlebot2i_demo1.launch
minimal.launch
rtabmap.launch
arbotix_driver
<?xml version="1.0"?>
<launch>
<arg name="new_rtabmap" default="false"/>
<arg name="localization" default="false"/>
<arg name="rviz" default="false"/>
<arg name="rtabmapviz" default="false"/>
<include file="$(find turtlebot2i_bringup)/launch/minimal.launch"/>
<include file="$(find turtlebot2i_bringup)/launch/arms/phantomx_pincher_arm.launch"/>
<include file="$(find turtlebot2i_arm_director)/launch/named_pose_director.launch">
<arg name="load_moveit" value="true"/>
</include>
<include file="$(find turtlebot2i_bringup)/launch/rtabmap.launch">
<arg if="$(arg new_rtabmap)" name="args" value="--delete_db_on_start"/>
<arg name="localization" value="$(arg localization)"/>
<arg name="rtabmapviz" value="$(arg rtabmapviz)"/>
</include>
<include file="$(find kobuki_auto_docking)/launch/standalone.launch" />
<include if="$(arg rviz)" file="$(find turtlebot2i_bringup)/launch/rviz.launch"/>
</launch>
turtlebot2i_mapping.launch (~/turtlebot2i/src/turtlebot2i_bringup/launch) - gedit
Open
Save
turtlebot2i_mapping.launch
turtlebot2i_demo1.launch
minimal.launch
rtabmap.launch
arbotix_driver
<?xml version="1.0"?>
<launch>
<arg name="new_rtabmap" default="true"/>
<arg name="localization" default="false"/>
<arg name="rviz" default="true"/>
<arg name="rtabmapviz" default="true"/>
<include file="$(find turtlebot2i_bringup)/launch/minimal.launch"/>
<include file="$(find turtlebot2i_bringup)/launch/arms/phantomx_pincher_arm.launch"/>
<include file="$(find turtlebot2i_arm_director)/launch/named_pose_director.launch">
<arg name="load_moveit" value="true"/>
</include>
<include file="$(find turtlebot2i_bringup)/launch/rtabmap.launch">
<arg if="$(arg new_rtabmap)" name="args" value="--delete_db_on_start"/>
<arg name="localization" value="$(arg localization)"/>
<arg name="rtabmapviz" value="$(arg rtabmapviz)"/>
</include>
<include file="$(find kobuki_auto_docking)/launch/standalone.launch" />
<include if="$(arg rviz)" file="$(find turtlebot2i_bringup)/launch/rviz.launch"/>
</launch>

I think mapping should need Gazebo, so I launched Gazebo first, and I remember there are TF issues between map->odom and odom->base_footprint, so I checked TF after launching Gazebo

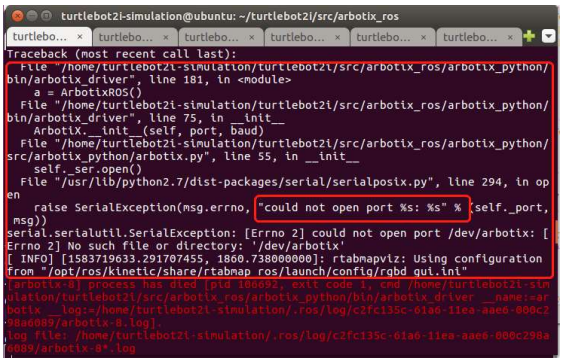




So we can see that the TF tree should be map->odom->base_footprint->base_link and there should be frames on arm_base_link branch

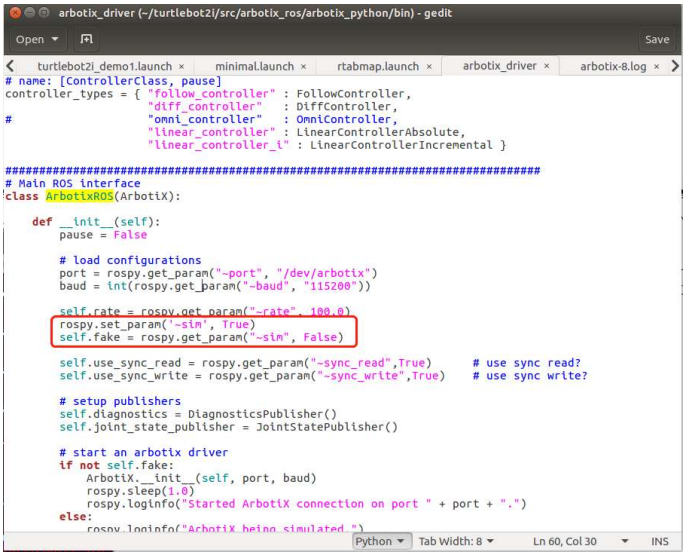
Now run turtlebot2i_mapping.launch and wait for error….

Arbotix died again, but I found something new

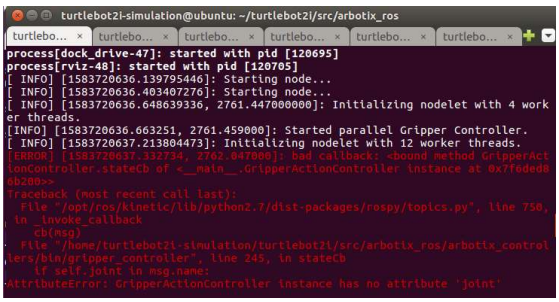


It seems that the program is looking for real port, so there maybe simulation flag, I want to check it.

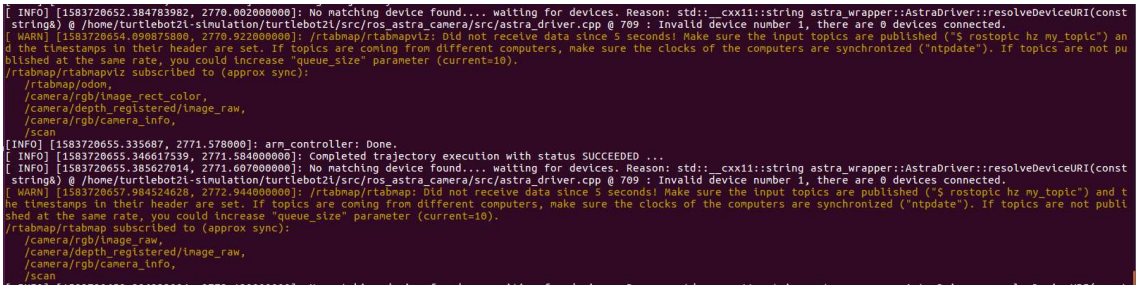
I added 1 line rospy.set_param('~sim', True)



New error, never met before



I think these errors are because of remapping of topics in some files



Wait, maybe the red error is because of controller conflict? Which means controller is loaded twice?

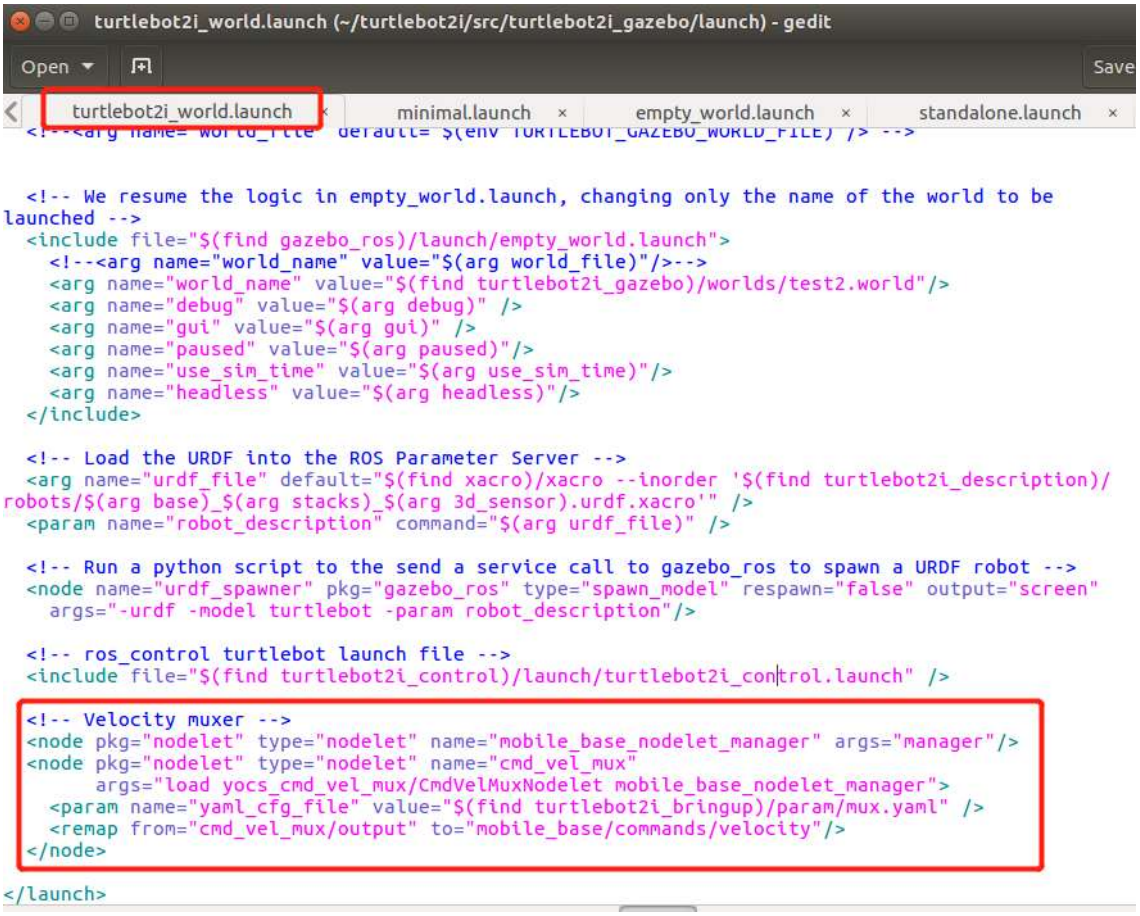
I think that's right, I shut down gazebo and red error is gone, only yellow warning remains, now I think I should comment the controller part in turtlebot2i_world.launch

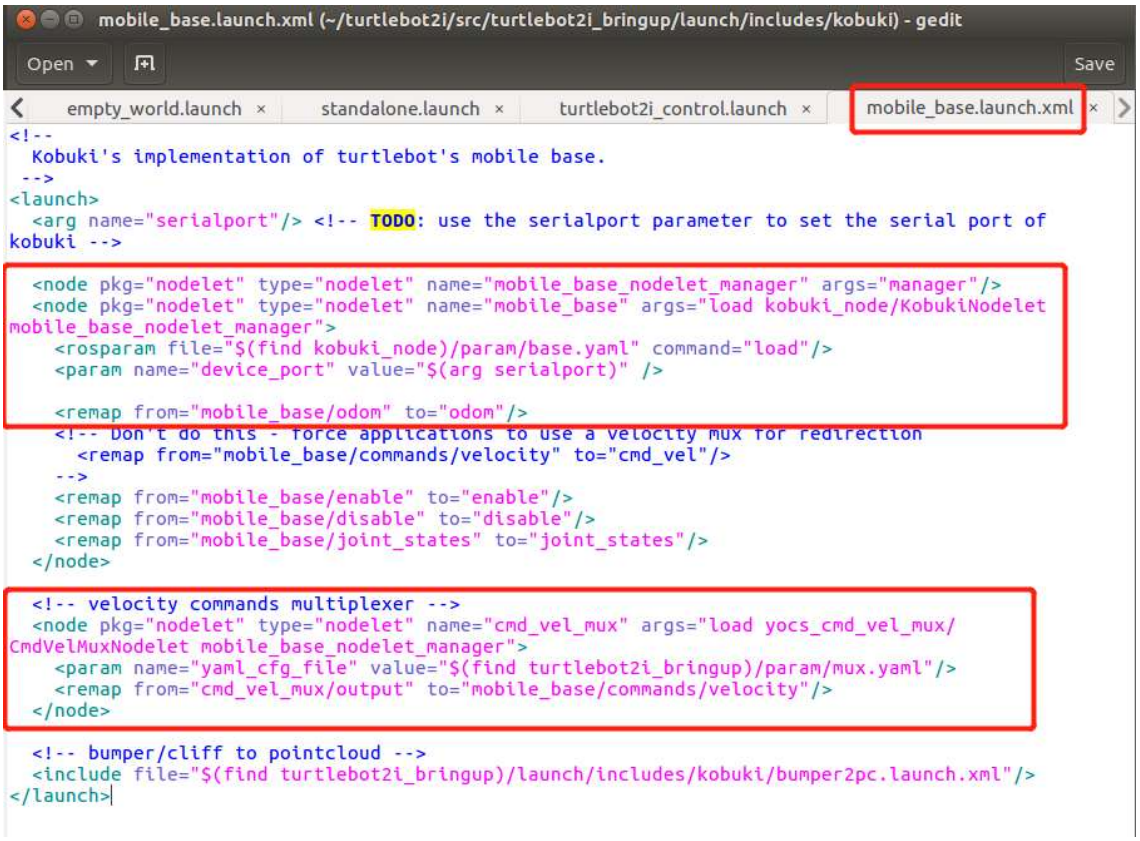
03/09/2020

Checking file contents, which will be screenshot in the next document.

Wait, I think it now may not be the controller issue, as when we operate real robot, the controller should be imbedded?

I find some part repeated in Simulation files and Physical robot operating files


turtlebot2i_world.launch (~/turtlebot2i/src/turtlebot2i_gazebo/launch) - gedit
Open
Save
turtlebot2i_world.launch
minimal.launch
empty_world.launch
standalone.launch
<!--<arg name="world_file" default="$(env TURTLEBOT_GAZEBO_WORLD_FILE)" /> -->
<!-- We resume the logic in empty_world.launch, changing only the name of the world to be launched -->
<include file="$(find gazebo_ros)/launch/empty_world.launch">
<!--<arg name="world_name" value="$(arg world_file)"/>-->
<arg name="world_name" value="$(find turtlebot2i_gazebo)/worlds/test2.world"/>
<arg name="debug" value="$(arg debug)" />
<arg name="gui" value="$(arg gui)" />
<arg name="paused" value="$(arg paused)"/>
<arg name="use_sim_time" value="$(arg use_sim_time)"/>
<arg name="headless" value="$(arg headless)"/>
</include>
<!-- Load the URDF into the ROS Parameter Server -->
<arg name="urdf_file" default="$(find xacro)/xacro --inorder '$(find turtlebot2i_description)/robots/$(arg base)_$(arg stacks)_$(arg 3d_sensor).urdf.xacro'" />
<param name="robot_description" command="$(arg urdf_file)" />
<!-- Run a python script to the send a service call to gazebo_ros to spawn a URDF robot -->
<node name="urdf_spawner" pkg="gazebo_ros" type="spawn_model" respawn="false" output="screen"
args="-urdf -model turtlebot -param robot_description"/>
<!-- ros_control turtlebot launch file -->
<include file="$(find turtlebot2i_control)/launch/turtlebot2i_control.launch" />
<!-- Velocity muxer -->
<node pkg="nodelet" type="nodelet" name="mobile_base_nodelet_manager" args="manager"/>
<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 turtlebot2i_bringup)/param/mux.yaml" />
<remap from="cmd_vel_mux/output" to="mobile_base/commands/velocity"/>
</node>
</launch>


mobile_base.launch.xml (~/turtlebot2i/src/turtlebot2i_bringup/launch/includes/kobuki) - gedit
Open
Save
empty_world.launch
standalone.launch
turtlebot2i_control.launch
mobile_base.launch.xml
<!--
Kobuki's implementation of turtlebot's mobile base.
-->
<launch>
<arg name="serialport"/> <!-- TODO: use the serialport parameter to set the serial port of kobuki -->
<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="$(find kobuki_node)/param/base.yaml" command="load"/>
<param name="device_port" value="$(arg serialport)" />
<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="$(find turtlebot2i_bringup)/param/mux.yaml"/>
<remap from="cmd_vel_mux/output" to="mobile_base/commands/velocity"/>
</node>
<!-- bumper/cliff to pointcloud -->
<include file="$(find turtlebot2i_bringup)/launch/includes/kobuki/bumper2pc.launch.xml"/>
</launch>

minimal.launch (~/turtlebot2i/src/turtlebot2i_bringup/launch) - gedit

Open Save

minimal.launch × empty_world.launch × standalone.launch × turtlebot2i_control.launch ×

```
<launch>
  <!-- Turtlebot2i -->
  <arg name="base"              default="$(env TURTLEBOT_BASE)"         doc="mobile base type
[kobuki]"/>
  <arg name="stacks"            default="$(env TURTLEBOT_STACKS)"       doc="stack type displayed
in visualisation/simulation [interbotix]"/>
  <arg name="3d_sensor"         default="$(env TURTLEBOT_3D_SENSOR)"    doc="3d sensor types
[zr300, astra]"/>
  <arg name="simulation"        default="$(env TURTLEBOT_SIMULATION)"   doc="set flags to indicate
this turtle is run in simulation mode."/>
  <arg name="serialport"        default="$(env TURTLEBOT_SERIAL_PORT)"  doc="used by create to
configure the port it is connected on [/dev/ttyUSB0, /dev/ttyS0]"/>

  <!--Robot-->
  <!--Description-->
  <arg name="urdf_file" default="$(find xacro)/xacro --inorder '$(find turtlebot2i_description)/
robots/$(arg base)_$(arg stacks)_$(arg 3d_sensor).urdf.xacro'" />
  <param name="robot_description" command="$(arg urdf_file)" />

  <!-- important generally, but specifically utilised by the current app manager -->
  <param name="robot/name" value="$(optenv ROBOT turtlebot)"/>
  <param name="robot/type" value="turtlebot"/>

  <node pkg="robot_state_publisher" type="robot_state_publisher" name="robot_state_publisher">
    <param name="publish_frequency" type="double" value="5.0" />
  </node>
  <node pkg="diagnostic_aggregator" type="aggregator_node" name="diagnostic_aggregator" >
    <rosparam command="load" file="$(find turtlebot2i_bringup)/param/$(arg base)/
diagnostics.yaml" />
  </node>

  <!--Mobile Base-->
  <include file="$(find turtlebot2i_bringup)/launch/includes/$(arg base)/mobile_base.launch.xml">
    <arg name="serialport" value="$(arg serialport)" />
  </include>
</launch>
```

turtlebot2i_control.launch (~/turtlebot2i/src/turtlebot2i_control/launch) - gedit

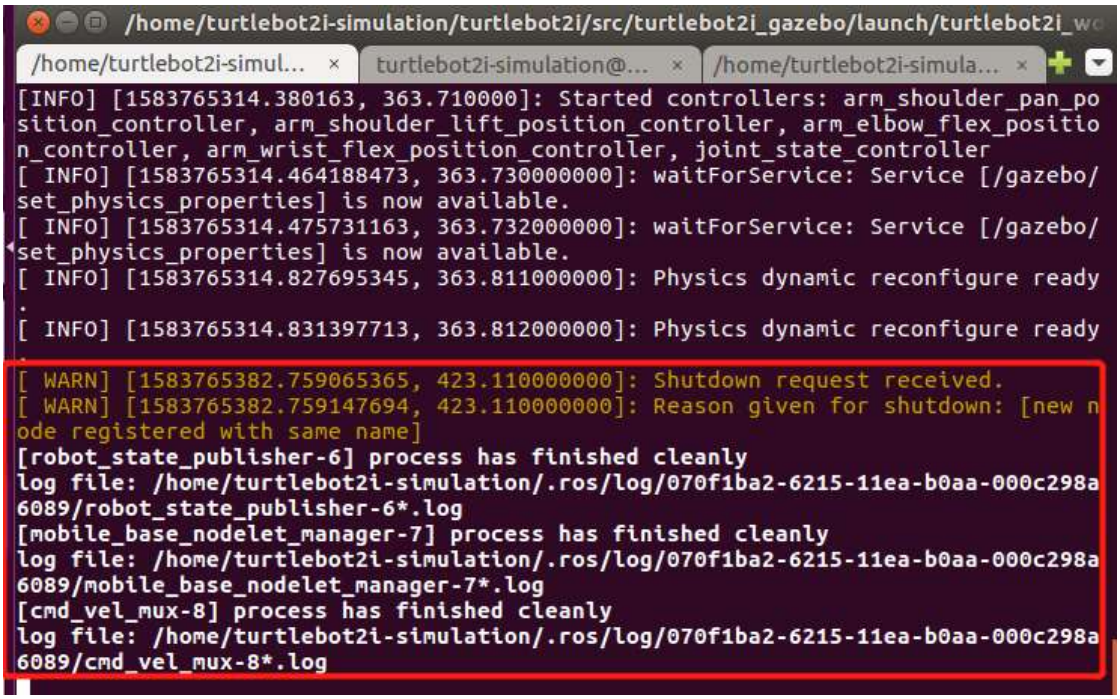
Open Save

empty_world.launch × standalone.launch × turtlebot2i_control.launch × mobile_base.launch.xml ×

```
<launch>

  <!-- Load joint controller configurations from YAML file to parameter server -->
  <rosparam file="$(find turtlebot2i_control)/config/turtlebot2i_control.yaml" command="load"/>

  <!-- load the controllers -->
  <node name="controller_spawner" pkg="controller_manager" type="spawner" respawn="false"
    output="screen" ns="/turtlebot" args="arm_shoulder_pan_position_controller
arm_shoulder_lift_position_controller arm_elbow_flex_position_controller
arm_wrist_flex_position_controller joint_state_controller"/>

  <!-- convert joint states to TF transforms for rviz, etc -->
  <node name="robot_state_publisher" pkg="robot_state_publisher" type="robot_state_publisher"
    respawn="false" output="screen">
    <remap from="/joint_states" to="/turtlebot/joint_states" />
  </node>

</launch>
```

GG, I don't know why, but not modifying any file, and run turtlebot2i_world.launch and turtlebot2i_mapping.launch. Rviz is showing a robot with arm, which doesn't have an arm last night.

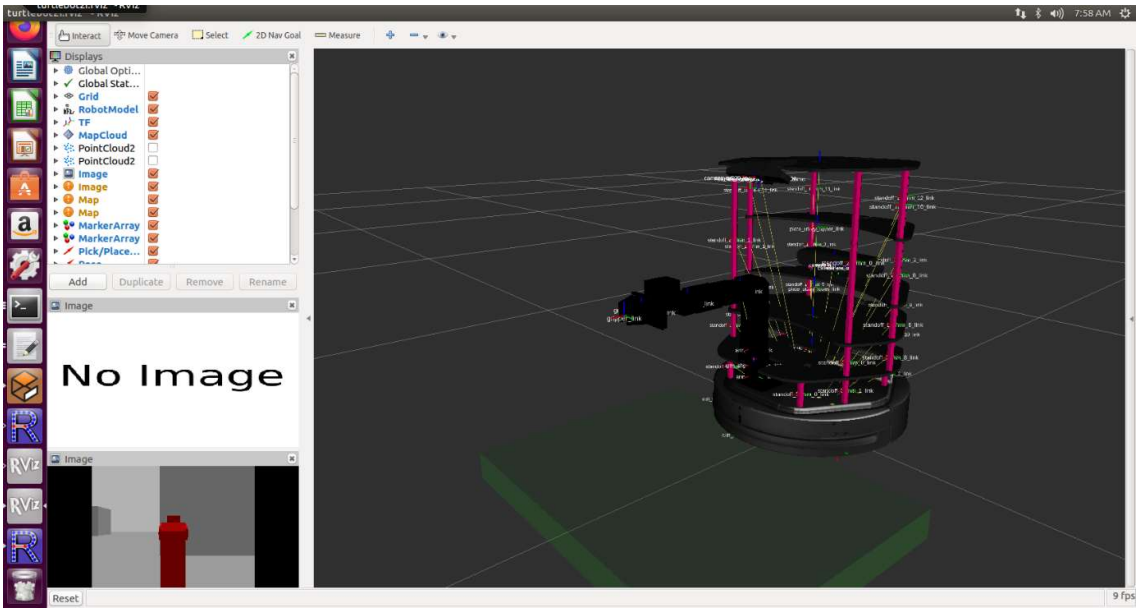
But Gazebo shows error, and I think it matches the my previous conclusion of repeatedly launching nodes. Nothing is in conflict, unbelievable.

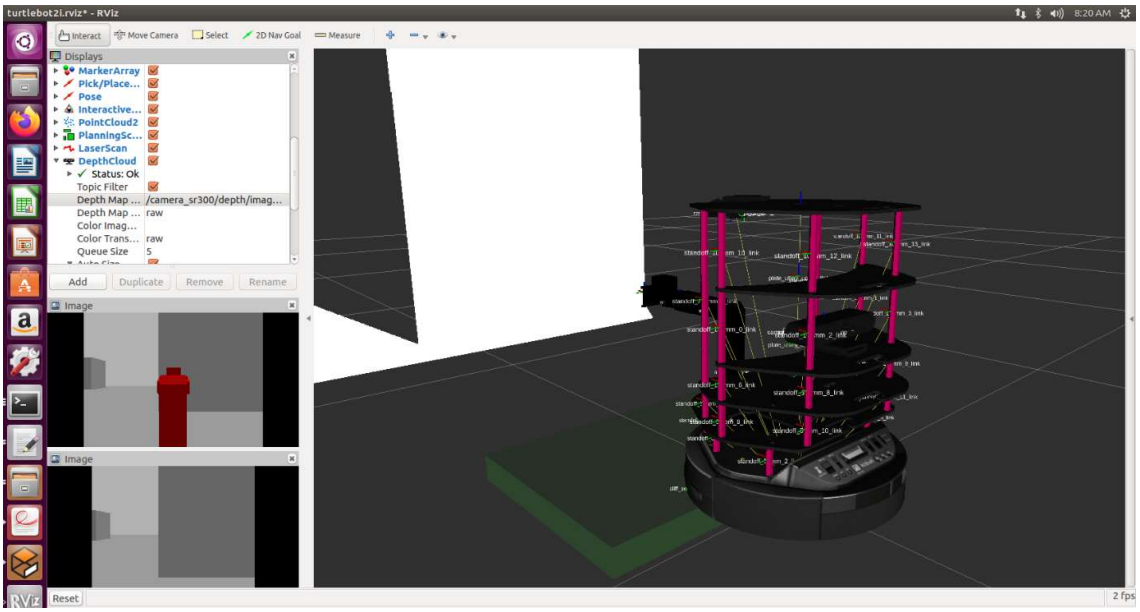


And turtlebot2i_mapping.launch is showing this error as before

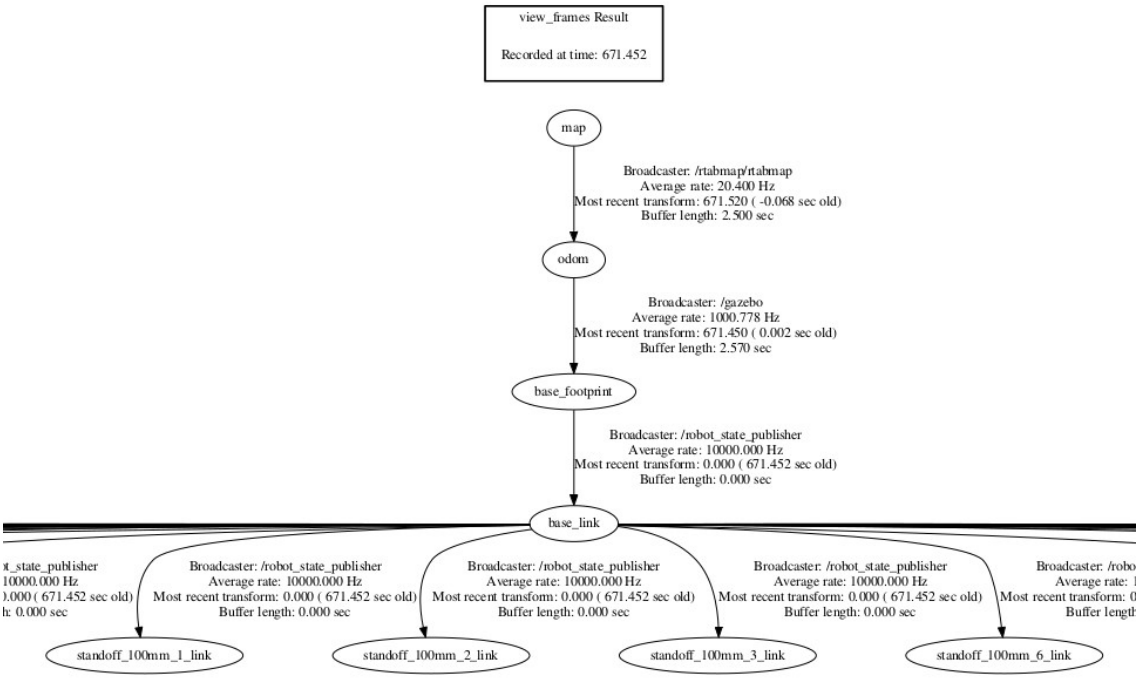
```
[ INFO] [1583765465.013387689, 462.287000000]: No matching device found.... waiting for devices. Reason: std::__cxx11::string astra_wrapper::AstraDriver::resolveDeviceURI(const
string&) @ /home/turtlebot2i-simulation/turtlebot2i/src/ros_astra_camera/src/astra_driver.cpp @ 709 : Invalid device number 1, there are 0 devices connected.
[ INFO] [1583765468.014161024, 463.823000000]: No matching device found.... waiting for devices. Reason: std::__cxx11::string astra_wrapper::AstraDriver::resolveDeviceURI(const
string&) @ /home/turtlebot2i-simulation/turtlebot2i/src/ros_astra_camera/src/astra_driver.cpp @ 709 : Invalid device number 1, there are 0 devices connected.
[ INFO] [1583765471.014923976, 465.434000000]: No matching device found.... waiting for devices. Reason: std::__cxx11::string astra_wrapper::AstraDriver::resolveDeviceURI(const
string&) @ /home/turtlebot2i-simulation/turtlebot2i/src/ros_astra_camera/src/astra_driver.cpp @ 709 : Invalid device number 1, there are 0 devices connected.
[ WARN] [1583765472.114239976, 465.934000000]: /rtabmap/rtabmapviz: Did not receive data since 5 seconds! Make sure the input topics are published ("$ rostopic hz my_topic") and
 the timestamps in their header are set. If topics are coming from different computers, make sure the clocks of the computers are synchronized ("ntpdate"). If topics are not pub
lished at the same rate, you could increase "queue_size" parameter (current=10).
/rtabmap/rtabmapviz subscribed to (approx sync):
   /rtabmap/odom,
   /camera/rgb/image_rect_color,
   /camera/depth_registered/image_raw,
   /camera/rgb/camera_info,
   /scan
[ WARN] [1583765473.337868997, 466.589000000]: /rtabmap/rtabmap: Did not receive data since 5 seconds! Make sure the input topics are published ("$ rostopic hz my_topic") and th
e timestamps in their header are set. If topics are coming from different computers, make sure the clocks of the computers are synchronized ("ntpdate"). If topics are not publis
hed at the same rate, you could increase "queue_size" parameter (current=10).
/rtabmap/rtabmap subscribed to (approx sync):
   /camera/rgb/image_raw,
   /camera/depth_registered/image_raw,
   /camera/rgb/camera_info,
   /scan
[ INFO] [1583765474.015295917, 466.931000000]: No matching device found.... waiting for devices. Reason: std::__cxx11::string astra_wrapper::AstraDriver::resolveDeviceURI(const
string&) @ /home/turtlebot2i-simulation/turtlebot2i/src/ros_astra_camera/src/astra_driver.cpp @ 709 : Invalid device number 1, there are 0 devices connected.
[ INFO] [1583765477.015619766, 468.483000000]: No matching device found.... waiting for devices. Reason: std::__cxx11::string astra_wrapper::AstraDriver::resolveDeviceURI(const
string&) @ /home/turtlebot2i-simulation/turtlebot2i/src/ros_astra_camera/src/astra_driver.cpp @ 709 : Invalid device number 1, there are 0 devices connected.
[ INFO] [1583765480.016143263, 470.051000000]: No matching device found.... waiting for devices. Reason: std::__cxx11::string astra_wrapper::AstraDriver::resolveDeviceURI(const
string&) @ /home/turtlebot2i-simulation/turtlebot2i/src/ros_astra_camera/src/astra_driver.cpp @ 709 : Invalid device number 1, there are 0 devices connected.
[ WARN] [1583765481.787046335, 470.936000000]: /rtabmap/rtabmapviz: Did not receive data since 5 seconds! Make sure the input topics are published ("$ rostopic hz my_topic") and
 the timestamps in their header are set. If topics are coming from different computers, make sure the clocks of the computers are synchronized ("ntpdate"). If topics are not pub
lished at the same rate, you could increase "queue_size" parameter (current=10).
/rtabmap/rtabmapviz subscribed to (approx sync):
   /rtabmap/odom,
   /camera/rgb/image_rect_color,
   /camera/depth_registered/image_raw,
   /camera/rgb/camera_info,
   /scan
[ INFO] [1583765483.016959105, 471.583000000]: No matching device found.... waiting for devices. Reason: std::__cxx11::string astra_wrapper::AstraDriver::resolveDeviceURI(const
string&) @ /home/turtlebot2i-simulation/turtlebot2i/src/ros_astra_camera/src/astra_driver.cpp @ 709 : Invalid device number 1, there are 0 devices connected.
[ WARN] [1583765483.036561746, 471.589000000]: /rtabmap/rtabmap: Did not receive data since 5 seconds! Make sure the input topics are published ("$ rostopic hz my_topic") and th
e timestamps in their header are set. If topics are coming from different computers, make sure the clocks of the computers are synchronized ("ntpdate"). If topics are not publis
hed at the same rate, you could increase "queue_size" parameter (current=10).
```

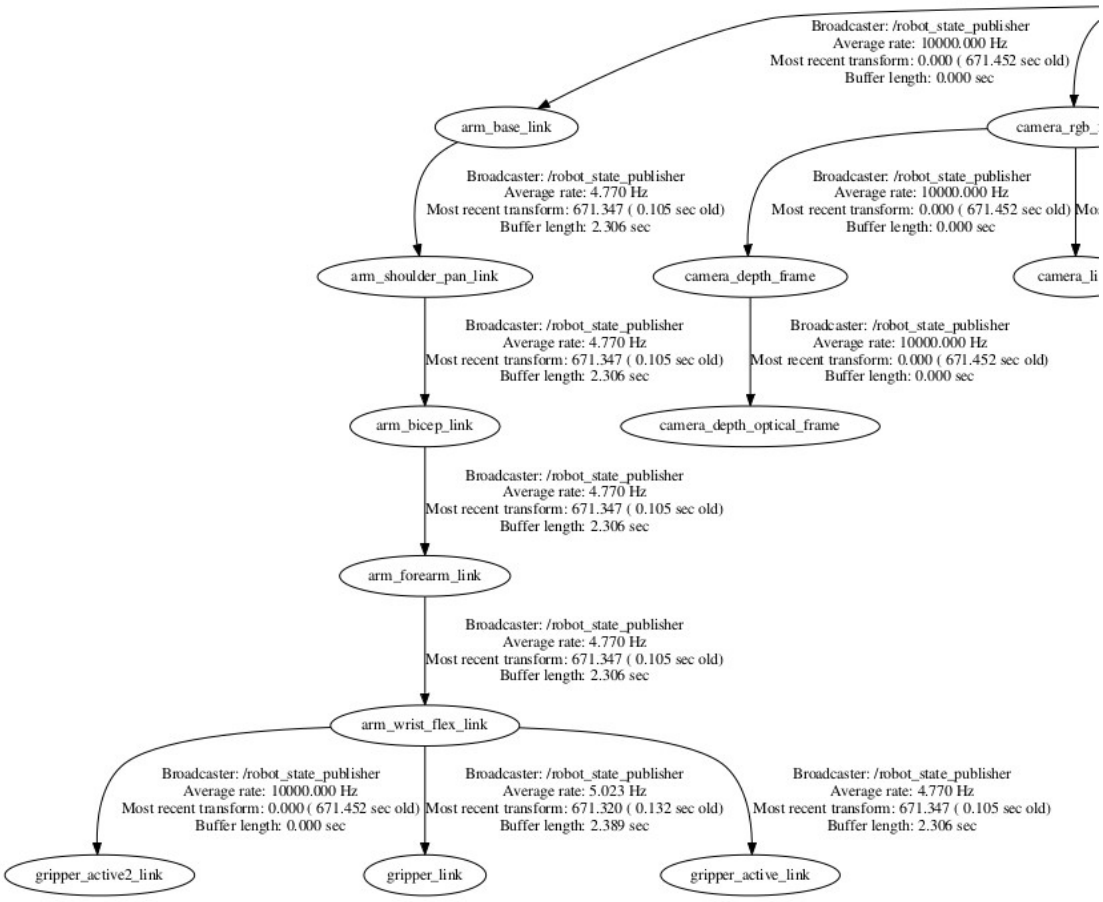
Screenshot of Rviz. There is no transform from any frame missing. rtabmapviz is still not showing image, which is reasonable





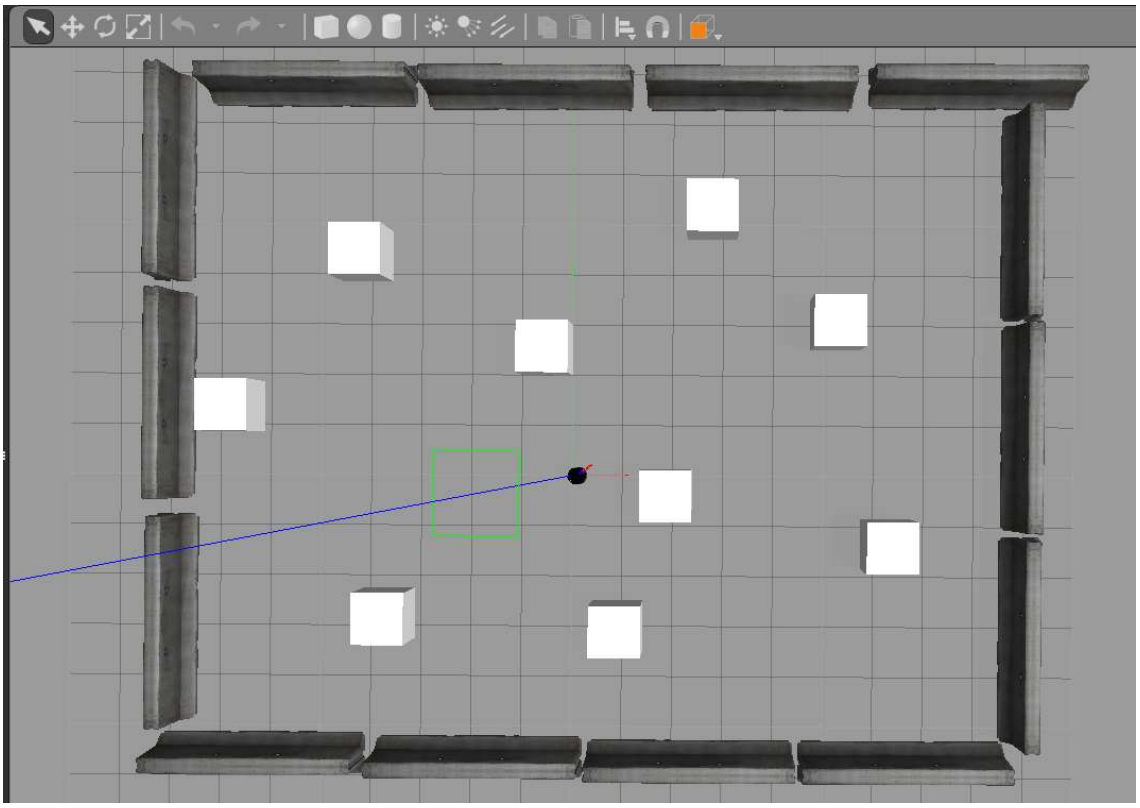
Screenshot of TF frames, every thing is connected






```
turtlebot2i-simulation@ubuntu:~$ rostopic hz /map
subscribed to [/map]
WARNING: may be using simulated time
no new messages
no new messages
no new messages
no new messages
no new messages
no new messages
no new messages
no new messages
no new messages
^Cno new messages
turtlebot2i-simulation@ubuntu:~$
```

Robot is able to react to keyboard teleop



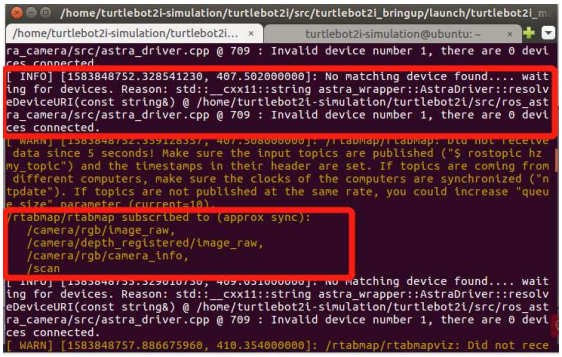
rostopic list

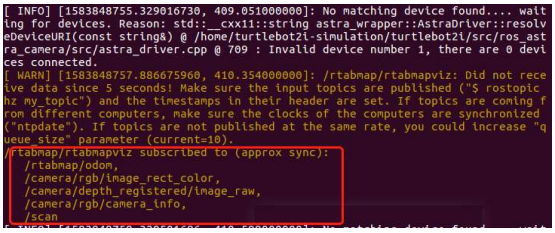
```
turtlebot2i-simulation@ubuntu:~$ rostopic list
/arm_bicep_joint/command
/arm_controller/command
/arm_controller/follow_joint_trajectory/cancel
/arm_controller/follow_joint_trajectory/feedback
/arm_controller/follow_joint_trajectory/goal
/arm_controller/follow_joint_trajectory/result
/arm_controller/follow_joint_trajectory/status
/arm_forearm_joint/command
/arm_shoulder_pan_joint/command
/arm_wrist_flex_joint/command
/attached_collision_object
/block_controls/feedback
/block_controls/update
/block_detection_action_server/block_output
/camera/depth/camera_info
/camera/depth/image
/camera/depth/image/compressed
/camera/depth/image/compressed/parameter_descriptions
/camera/depth/image/compressed/parameter_updates
/camera/depth/image/compressedDepth
/camera/depth/image/compressedDepth/parameter_descriptions
/camera/depth/image/compressedDepth/parameter_updates
/camera/depth/image_raw
/camera/depth/image_rect
/camera/depth/image_rect/compressed
/camera/depth/image_rect/compressed/parameter_descriptions
/camera/depth/image_rect/compressed/parameter_updates
/camera/depth/image_rect/compressedDepth
/camera/depth/image_rect/compressedDepth/parameter_descriptions
/camera/depth/image_rect/compressedDepth/parameter_updates
/camera/depth/points
/camera/depth_registered/image_raw
/camera/ir/image_rect_ir
/camera/ir/image_rect_ir/compressed
/camera/ir/image_rect_ir/compressed/parameter_descriptions
/camera/ir/image_rect_ir/compressed/parameter_updates
/camera/ir/image_rect_ir/compressedDepth
/camera/ir/image_rect_ir/compressedDepth/parameter_descriptions
/camera/ir/image_rect_ir/compressedDepth/parameter_updates
/camera/ir_rectify_ir/parameter_descriptions
/camera/ir_rectify_ir/parameter_updates
/camera/parameter_descriptions
/camera/parameter_updates
```

```
/camera/parameter_descriptions
/camera/parameter_updates
/camera/rgb/camera_info
/camera/rgb/image_raw
/camera/rgb/image_raw/compressed
/camera/rgb/image_raw/compressed/parameter_descriptions
/camera/rgb/image_raw/compressed/parameter_updates
/camera/rgb/image_raw/compressedDepth
/camera/rgb/image_raw/compressedDepth/parameter_descriptions
/camera/rgb/image_raw/compressedDepth/parameter_updates
/camera/rgb/image_rect_color
/camera_sr300/depth/camera_info
/camera_sr300/depth/image_raw
/camera_sr300/depth/points
/camera_sr300/parameter_descriptions
/camera_sr300/parameter_updates
/camera_sr300/rgb/camera_info
/camera_sr300/rgb/image_raw
/camera_sr300/rgb/image_raw/compressed
/camera_sr300/rgb/image_raw/compressed/parameter_descriptions
/camera_sr300/rgb/image_raw/compressed/parameter_updates
/camera_sr300/rgb/image_raw/compressedDepth
/camera_sr300/rgb/image_raw/compressedDepth/parameter_descriptions
/camera_sr300/rgb/image_raw/compressedDepth/parameter_updates
/camera_sr300/rgb/image_rect_color
/clock
/cmd_vel_mux/active
/cmd_vel_mux/input/navi
/cmd_vel_mux/input/safety_controller
/cmd_vel_mux/input/switch
/cmd_vel_mux/input/teleop
/cmd_vel_mux/parameter_descriptions
/cmd_vel_mux/parameter_updates
/collision_object
/diagnostics
/diagnostics_agg
/diagnostics_toplevel_state
/dock_drive/debug/feedback
/dock_drive/debug/mode_shift
/dock_drive_action/cancel
/dock_drive_action/feedback
```

03/10

I found that there is another issue I need to fix, "no matching device found"





I think now I need to disable the calling for physical device and check whether these topics exists and have valid messages published to them.

First find where the physical camera is called and comment it.

Wait a second, now I have the whole simulated robot setted up. What's the point of running turtlebot2i_mapping.launch, which contains some contents for real robot? What I need is just to create a map, why not running rtabmap.launch solely?

It seems that it doesn't work, errors as follow, the same as previous ones

```
[ INFO] [1583850300.961124359, 429.152000000]: No matching device found.... waiting for devices. Reason: std::__cxx11::string astra_wrapper::AstraDriver::resolveDeviceURI(const strin
g&) @ /home/turtlebot2i-simulation/turtlebot2i/src/ros_astra_camera/src/astra_driver.cpp @ 709 : Invalid device number 1, there are 0 devices connected.
[ WARN] [1583850301.911506155, 429.845000000]: /rtabmap/rtabmap: Did not receive data since 5 seconds! Make sure the input topics are published ("$ rostopic hz my_topic") and the tim
estamps in their header are set. If topics are coming from different computers, make sure the clocks of the computers are synchronized ("ntpdate"). If topics are not published at the
 same rate, you could increase "queue_size" parameter (current=10).
/rtabmap/rtabmap subscribed to (approx sync):
   /camera/rgb/image_raw,
   /camera/depth_registered/image_raw,
   /camera/rgb/camera_info,
   /scan
[ INFO] [1583850303.961802783, 431.264000000]: No matching device found.... waiting for devices. Reason: std::__cxx11::string astra_wrapper::AstraDriver::resolveDeviceURI(const strin
g&) @ /home/turtlebot2i-simulation/turtlebot2i/src/ros_astra_camera/src/astra_driver.cpp @ 709 : Invalid device number 1, there are 0 devices connected.
[ INFO] [1583850306.962062421, 433.428000000]: No matching device found.... waiting for devices. Reason: std::__cxx11::string astra_wrapper::AstraDriver::resolveDeviceURI(const strin
g&) @ /home/turtlebot2i-simulation/turtlebot2i/src/ros_astra_camera/src/astra_driver.cpp @ 709 : Invalid device number 1, there are 0 devices connected.
[ WARN] [1583850307.063608191, 433.485000000]: /rtabmap/rtabmapviz: Did not receive data since 5 seconds! Make sure the input topics are published ("$ rostopic hz my_topic") and the
timestamps in their header are set. If topics are coming from different computers, make sure the clocks of the computers are synchronized ("ntpdate"). If topics are not published at
the same rate, you could increase "queue_size" parameter (current=10).
/rtabmap/rtabmapviz subscribed to (approx sync):
   /rtabmap/odom,
   /camera/rgb/image_rect_color,
   /camera/depth_registered/image_raw,
   /camera/rgb/camera_info,
   /scan
```

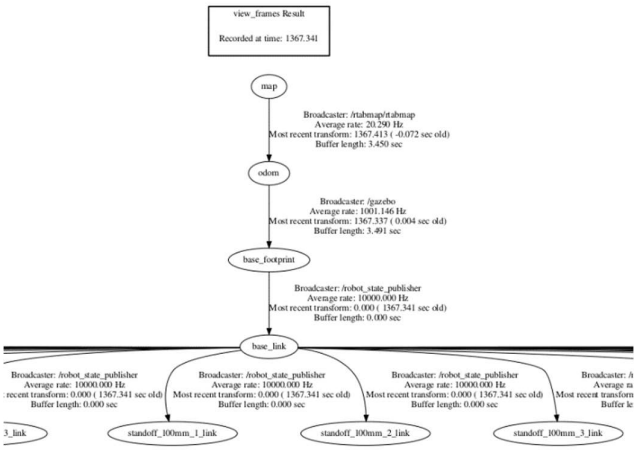
I found something new when referring to turorials for rtabmap on ROS Wiki.
http://wiki.ros.org/rtabmap_ros/Tutorials/SetupOnYourRobot#Kinect_.2B-_Odometry_.2B-_Fake_2D_laser_from_Kinect
http://wiki.ros.org/rtabmap_ros/Tutorials/MappingAndNavigationOnTurtlebot
It seems that I need to run a separate node to generate messages published to /scan topic

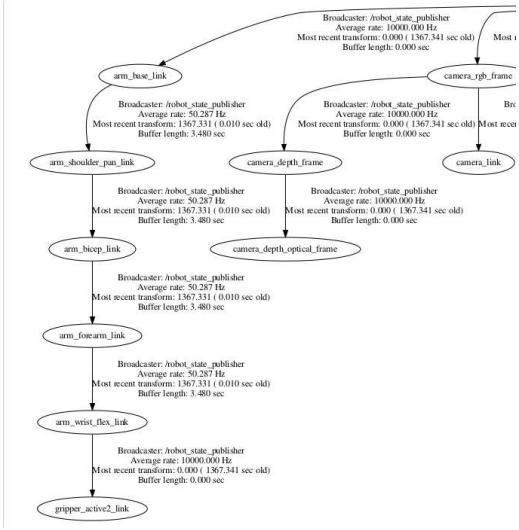
Check whether there are messages published to subscribed topics again,

```
[ INFO] [1583851010.836888918, 965.474000000]: No matching device found.... waiting for devices. Reason: std::__cxx11::string astra_wrapper::AstraDriver::resolveDeviceURI(const strin
g&) @ /home/turtlebot2i-simulation/turtlebot2i/src/ros_astra_camera/src/astra_driver.cpp @ 709 : Invalid device number 1, there are 0 devices connected.
[ INFO] [1583851013.837396950, 967.617000000]: No matching device found.... waiting for devices. Reason: std::__cxx11::string astra_wrapper::AstraDriver::resolveDeviceURI(const strin
g&) @ /home/turtlebot2i-simulation/turtlebot2i/src/ros_astra_camera/src/astra_driver.cpp @ 709 : Invalid device number 1, there are 0 devices connected.
[ WARN] [1583851014.474453084, 968.079000000]: /rtabmap/rtabmapviz: Did not receive data since 5 seconds! Make sure the input topics are published ("$ rostopic hz my_topic") and the
timestamps in their header are set. If topics are coming from different computers, make sure the clocks of the computers are synchronized ("ntpdate"). If topics are not published at
the same rate, you could increase "queue_size" parameter (current=10).
/rtabmap/rtabmapviz subscribed to (approx sync):
   /rtabmap/odom,
   /camera/rgb/image_rect_color,
   /camera/depth_registered/image_raw,
   /camera/rgb/camera_info,
   /scan
[ WARN] [1583851016.668521645, 969.678000000]: /rtabmap/rtabmap: Did not receive data since 5 seconds! Make sure the input topics are published ("$ rostopic hz my_topic") and the tim
estamps in their header are set. If topics are coming from different computers, make sure the clocks of the computers are synchronized ("ntpdate"). If topics are not published at the
 same rate, you could increase "queue_size" parameter (current=10).
/rtabmap/rtabmap subscribed to (approx sync):
   /camera/rgb/image_raw,
   /camera/depth_registered/image_raw,
   /camera/rgb/camera_info,
   /scan
```

Topics in red rectanges don't have messages published to them.
I think /rtanmap/odom could be visual odometery staff, because I didn't enable the following tag

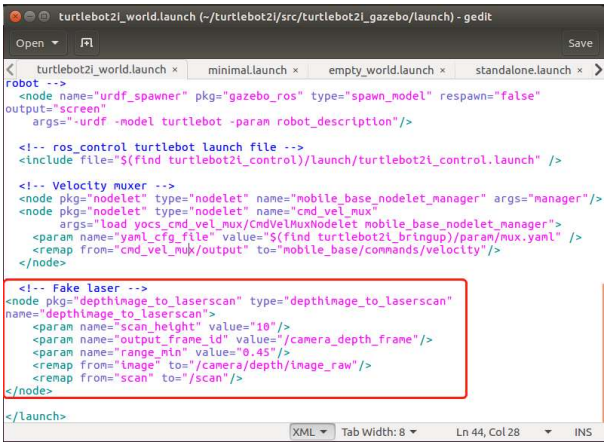
I noticed the TF error didn't show up just now, so I check the TF tree, and TF is in good shape. I think it may because I didn't run turtlebot2i_mapping.launch, so the TF is not redefined.





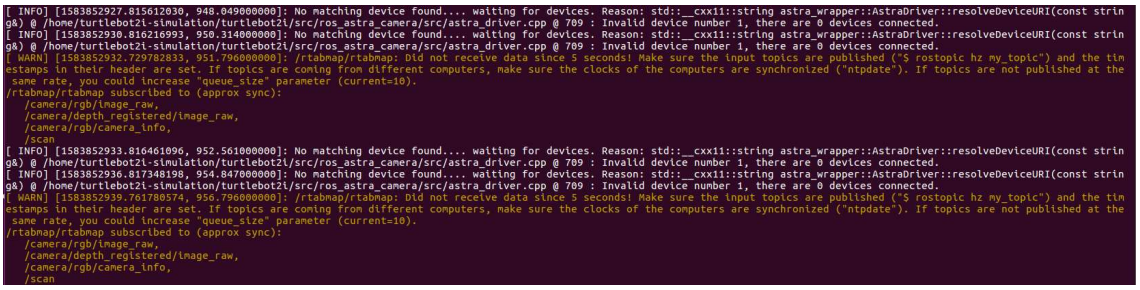
Now I think I'm going to do some modification to rtabmap.launch, adding the /scan part and remap /camera/depth_registered/image_raw

Accidentally see there is Simulation part in this mapping and navigation tutorial for rtabmap, http://wiki.ros.org/rtabmap_ros/Tutorials/MappingAndNavigationOnTurtlebot, so I decided to add the /scan part in turtlebot2i_world.launch

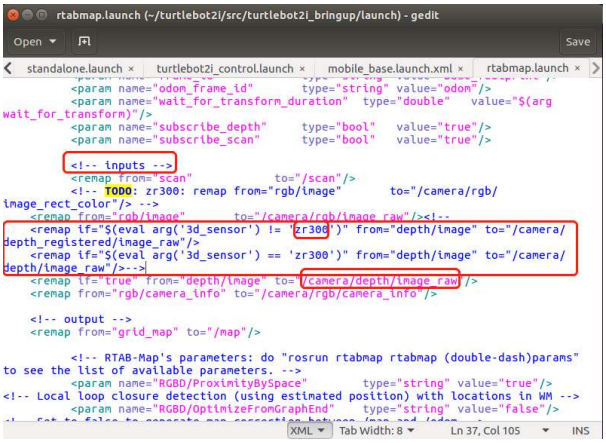


After running turtlebot2i_world.launch(self created), the /scan doesn't have message published to it. When I restart(maybe) and check things for a while, /scan works well.

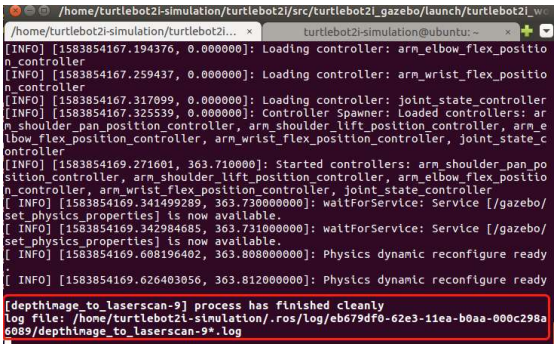
Now I started turtlebot2i_bringup/rtabmap.launch, and it still shows no matching device found and the orange no message published to topic error. I think maybe I run rtabmap_ros/rtabmap.launch will solve this issue? And I check that file, but is doesn't show much difference.



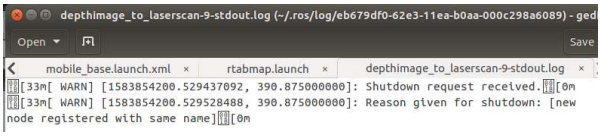
So, now I think I should continue with turtlebot2i_bringup/rtabmap.launch and finish this direction of fixing.
I commented 2 lines and add 1 line. I think the launch file is designed for zr300 camera instead of sr300 camera, and sr300 is publishing the topic in the rectangle, besides these are inputs of rtabmap(black box), so I made modifications as shown in the screenshot.



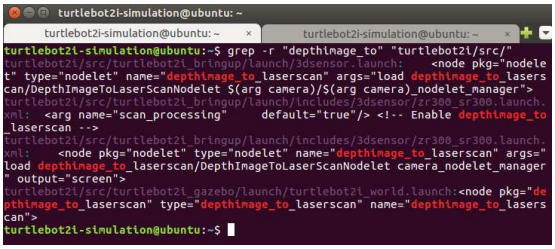
I found depthimage_to_laserscan will die after I ran rtabmap.launch. So I'm going to check the log file.



After checking the log file, it turns out that the node is shut down because there is a node with the same name created.



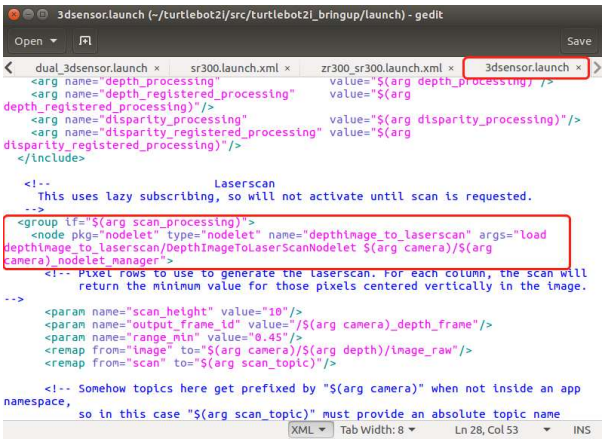
So I checked where the second time this node is created, these are the result(may not complete).



So I think turtlebot2i_bringup/rtabmap.launch may include some of files listed above.

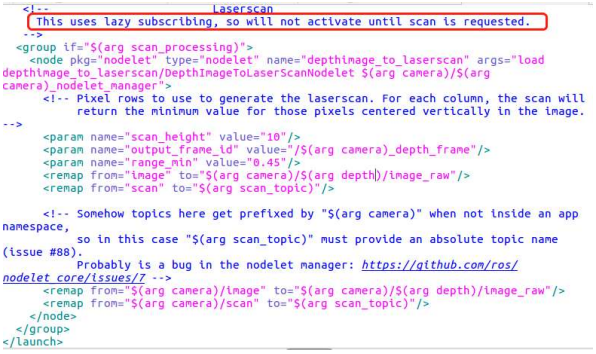
So I checked it, and find it called dual_3dsensor.launch. So I opened it.

After looking for cascaded files for a while, I found the depthimage_to_laserscan node is created here.



So I will comment the one I added in turtlebot2i_world.launch, and try again.

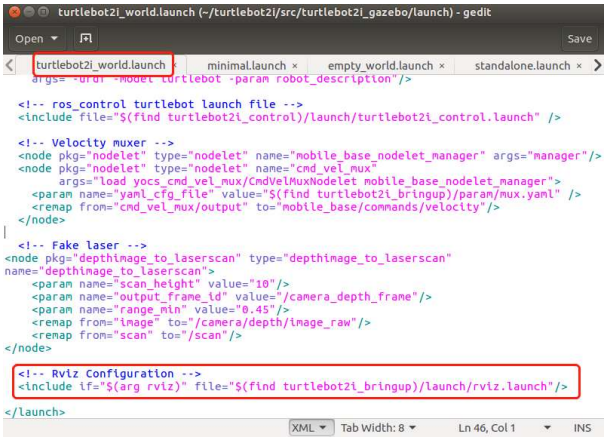
/scan still have no message published to it. I found something novel in the 3dsensor.launch



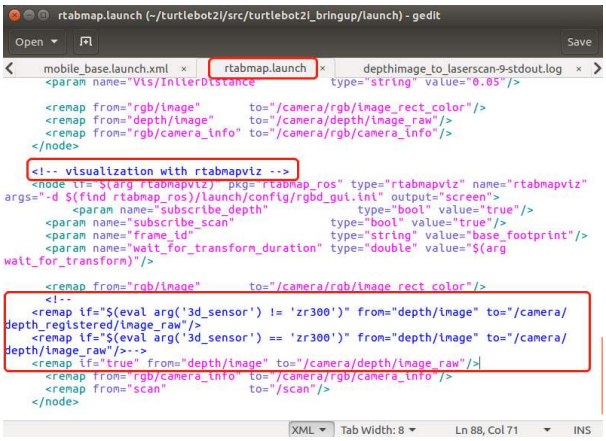
After googling for minutes, I almost find nothing related to this, so I decide to comment this part and uncomment the part in turtlebot2i_world.launch, as it works well before.

Now able to see 2D map, I modified this to see what rtabmapviz can show me.

In order to keep Rviz configuretion, I added 1 line at the end of turtlebot2i_world.launch



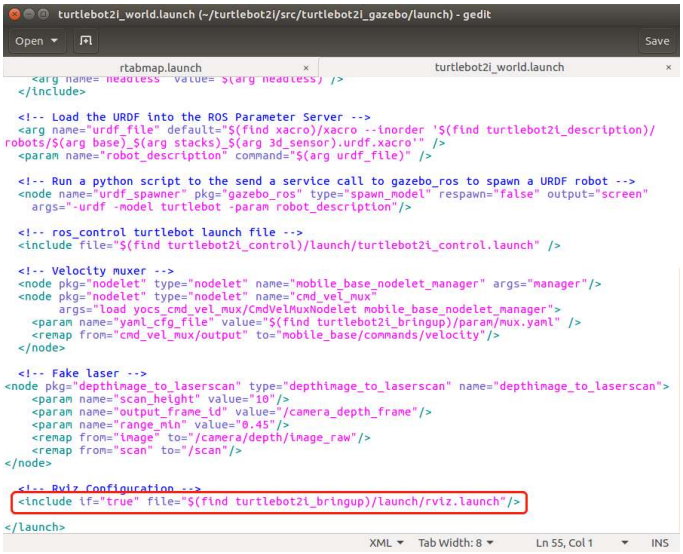
In order to view things correctly in rtabmapviz, I modified rtabmapviz part of rtabmap.launch.



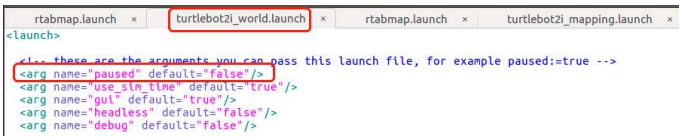
Now make a github copy for current environment, and next step is to change the camera to camera_sr300, as the astra camera is partially block by the arm.

Reboot and run turtlebot2i_gazebo.launch again, error prompts "arg rviz should be provided"

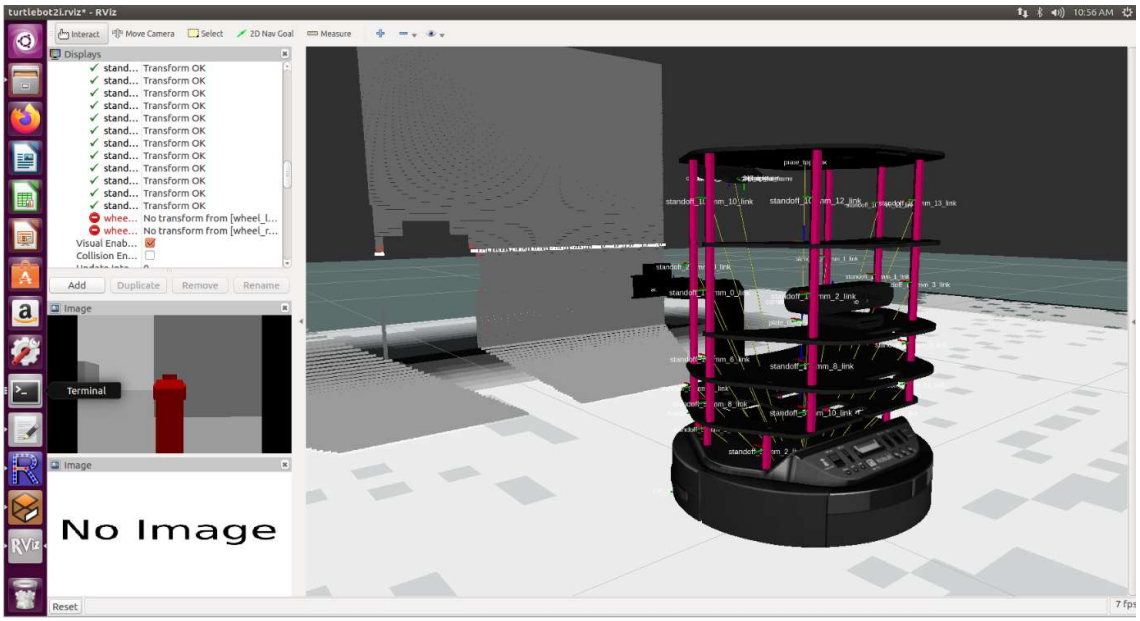
After inspection of that file, I made modification as follows:

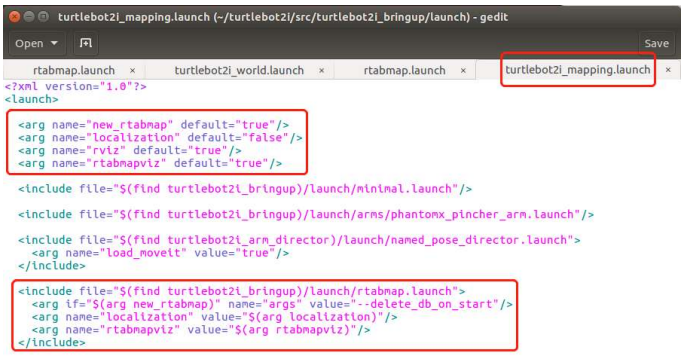


Now restart it again. I found although Rviz subscribes some topics, Rviz received no message from those topics. So I think this may be caused by Gazebo being paused at the beginning. So I made mpdification to turtllebot2i_world.launch



Run the two launch files again, rtabmap.launch shows this error. It seems that the program is trying to find the database, which is the previous map. I remember there is a flag in the launch file that can disable this behavior and start generating a new map. And wheel frames and gripper frames are missing again.

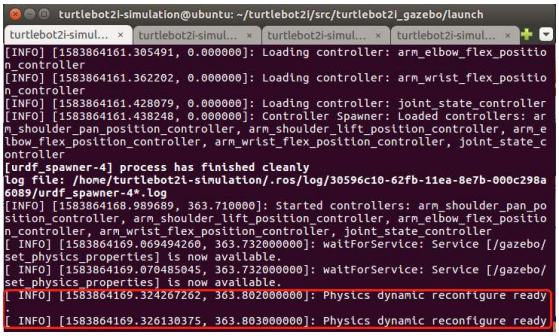




So I think maybe I can do this kind of cascaded calling of launch files and parameter passing as well, so I modified my turtlebot2i_world.launch to launch Rviz in it.

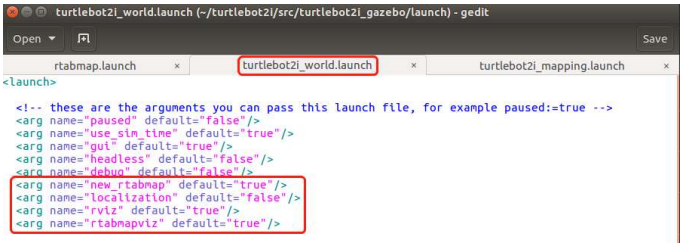
I met some strange error I didn't notice before in turtlebot2i_world.launch, which says robot_semantic_description not found. I have never seen it before, maybe I didn't notice the existence of it at upper part after running roslaunch turtlebot2i_gazebo turtlebot2i_world.launch. But when I run it again, it disappears.

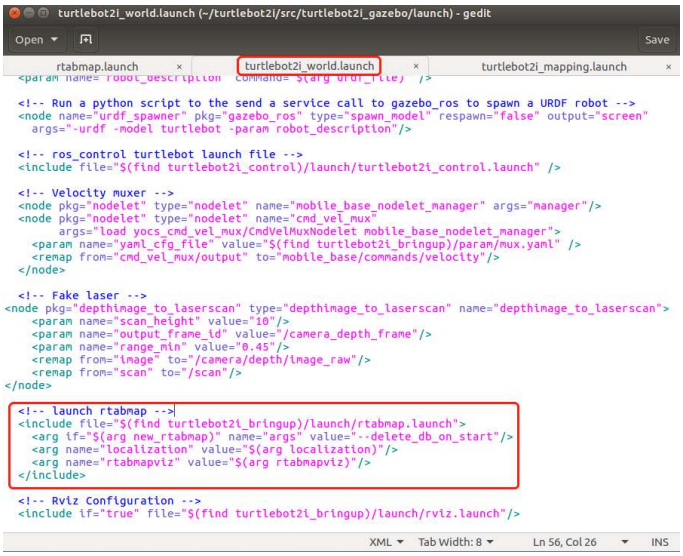
I don't know why I don't see these at the bottom of turtlebot2i_world.launch window, I guess they appear previously after I resume Gazebo after launching. So I set the pause flag to "true" and do it again, and it turns out my inference is right. So now I change it back to "false", and I don't think it will cause problems at this point of time.



I launched turtlbot2i_world.launch solely, TF error still exists. I think they may not be well defined. I will leave it here at this point of time.

Now resume modifying turtlebot2i_world.launch to make it launch rtabmap.launch

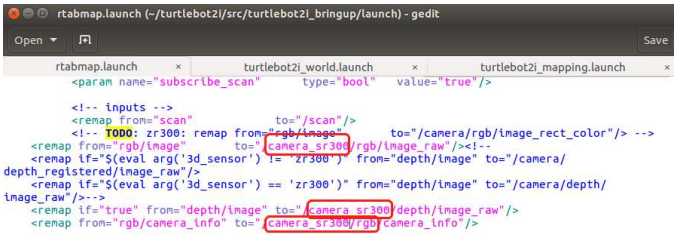




Now run turtlebot2i_world.launch again

TF error still exists, and Rviz will die at the beginning, nut will recover later. I guess this is because of the speed of computation.

I found there is an area of blank in front of robot, which is the same shape as the arm. So I think that's because rtabmap is taking astra camera data, which is partially blocked by the arm, as input. Now I want to remap that to sr300 camera data.
After modification:



rtabmap.launch (~/turtlebot2i/src/turtlebot2i_bringup/launch) - gedit

Open Save

rtabmap.launch | turtlebot2i_world.launch | turtlebot2i_mapping.launch

```
    <!-- Odometry : ONLY for testing without the actual robot! /odom TF should not be already
published. -->
    <node if="$(arg rgbd_odometry)" pkg="rtabmap_ros" type="rgbd_odometry" name="rgbd_odometry"
output="screen">
      <param name="frame_id"                    type="string" value="base_footprint"/>
      <param name="wait_for_transform_duration" type="double" value="$(arg wait_for_transform)"/>
      <param name="Reg/Force3DoF"               type="string" value="true"/>
      <param name="Vis/InlierDistance"          type="string" value="0.05"/>

      <remap from="rgb/image"       to="/camera_sr300/rgb/image_rect_color"/>
      <remap from="depth/image"     to="/camera_sr300/depth/image_raw"/>
      <remap from="rgb/camera_info" to="/camera_sr300/rgb/camera_info"/>
    </node>

    <!-- visualization with rtabmapviz -->
    <node if="$(arg rtabmapviz)" pkg="rtabmap_ros" type="rtabmapviz" name="rtabmapviz" args="-d
$(find rtabmap_ros)/launch/config/rgbd_gui.ini" output="screen">
        <param name="subscribe_depth"             type="bool" value="true"/>
      <param name="subscribe_scan"              type="bool" value="true"/>
      <param name="frame_id"                    type="string" value="base_footprint"/>
      <param name="wait_for_transform_duration" type="double" value="$(arg wait_for_transform)"/>

      <remap from="rgb/image"       to="/camera_sr300/rgb/image_rect_color"/>
      <!--
    <remap if="$(eval arg('3d_sensor') != 'zr300')" from="depth/image" to="/camera/
depth_registered/image_raw"/>
    <remap if="$(eval arg('3d_sensor') == 'zr300')" from="depth/image" to="/camera/depth/
image_raw"/>-->
    <remap if="true" from="depth/image" to="/camera_sr300/depth/image_raw"/>
      <remap from="rgb/camera_info" to="/camera_sr300/rgb/camera_info"/>
      <remap from="scan"            to="/scan"/>
    </node>

  </group>
</launch>
```

XML Tab Width: 8 Ln 81, Col 67 INS

As it always shows this warning, and I think it may be related to the following flag, so I set it to true.

```
[ INFO] [1583866884.906319246, 846.928000000]: rtabmap (440): Rate=1.00s, Limit=
0.700s, RTAB-Map=0.0546s, Maps update=0.0027s pub=0.0000s (local map=1, WM=1)
[ INFO] [1583866886.768318732, 847.941000000]: rtabmap (441): Rate=1.00s, Limit=
0.700s, RTAB-Map=0.0476s, Maps update=0.0032s pub=0.0000s (local map=1, WM=1)
[ INFO] [1583866888.812682556, 849.081000000]: rtabmap (442): Rate=1.00s, Limit=
0.700s, RTAB-Map=0.0435s, Maps update=0.0041s pub=0.0000s (local map=1, WM=1)
[ WARN] [1583866889.066966820, 849.241000000]: /rtabmap/rtabmapviz: Did not rece
ive data since 5 seconds! Make sure the input topics are published ("$ rostopic
hz my_topic") and the timestamps in their header are set. If topics are coming f
rom different computers, make sure the clocks of the computers are synchronized
("ntpdate"). If topics are not published at the same rate, you could increase "q
ueue_size" parameter (current=10).
/rtabmap/rtabmapviz subscribed to (approx sync):
   /rtabmap/odom,
   /camera/rgb/image_rect_color,
   /camera/depth/image_raw,
   /camera/rgb/camera_info,
   /scan
```

rtabmap.launch (~/turtlebot2i/src/turtlebot2i_bringup/launch) - ge

Open

rtabmap.launch | turtlebot2i_world.launch

```
<?xml version="1.0"?>

<launch>
  <arg name="database_path"     default="rtabmap.db"/>
  <arg name="localization"      default="false"/>
  <arg name="rgbd_odometry"     default="false"/>
  <arg name="args"              default=""/>
  <arg name="rtabmapviz"        default="true"/>

  <arg name="wait_for_transform"  default="0.2"/>
```

rtabmap.launch (~/turtlebot2i/src/turtlebot2i_bringup/lau

Open

rtabmap.launch | turtlebot2i_world.launch

```
<?xml version="1.0"?>

<launch>
  <arg name="database_path"     default="rtabmap.db"/>
  <arg name="localization"      default="false"/>
  <arg name="rgbd_odometry"     default="true"/>
  <arg name="args"              default=""/>
  <arg name="rtabmapviz"        default="true"/>

  <arg name="wait_for_transform"  default="0.2"/>
```

Now test it. It shows many errors like this, but things seem to work in the right way. I guess it is because of the speed of computation, and after failure, the program is looping to try is again.
The result shows the following 2 topics doesn't have message published to them

```
[ INFO] [1583867166.363984819, 377.909000000]: rtabmap (11): Rate=1.00s, Limit=0
.700s, RTAB-Map=0.0470s, Maps update=0.0014s pub=0.0000s (local map=1, WM=1)
[ INFO] [1583867168.326267498, 378.947000000]: rtabmap (12): Rate=1.00s, Limit=0
.700s, RTAB-Map=0.0521s, Maps update=0.0027s pub=0.0000s (local map=1, WM=1)
[ WARN] [1583867169.023179438, 379.285000000]: /rtabmap/rtabmapviz: Did not rece
ive data since 5 seconds! Make sure the input topics are published ("$ rostopic
hz my_topic") and the timestamps in their header are set. If topics are coming f
rom different computers, make sure the clocks of the computers are synchronized
("ntpdate"). If topics are not published at the same rate, you could increase "q
ueue_size" parameter (current=10).
/rtabmap/rtabmapviz subscribed to (approx sync):
   /rtabmap/odom
   /camera_sr300/rgb/image_rect_color,
   /camera_sr300/depth/image_raw,
   /camera_sr300/rgb/camera_info,
   /scan
```

I find something new

```
/home/turtlebot2i-simul...  ×  turtlebot2i-simulation@...  ×  turtlebot2i-simulation@u...  ×
[ INFO] [1583867176.920493170, 383.220000000]: rtabmap (16): Rate=1.00s, Limit=0
.700s, RTAB-Map=0.0534s, Maps update=0.0020s pub=0.0000s (local map=1, WM=1)
[ INFO] [1583867178.975265859, 384.245000000]: rtabmap (17): Rate=1.00s, Limit=0
.700s, RTAB-Map=0.0536s, Maps update=0.0009s pub=0.0001s (local map=1, WM=1)
[ WARN] [1583867179.046547153, 384.285000000]: /rtabmap/rtabmapviz: Did not rece
ive data since 5 seconds! Make sure the input topics are published ("$ rostopic
hz my_topic") and the timestamps in their header are set. If topics are coming f
rom different computers, make sure the clocks of the computers are synchronized
("ntpdate"). If topics are not published at the same rate, you could increase "q
ueue_size" parameter (current=10).
/rtabmap/rtabmapviz subscribed to (approx sync):
   /rtabmap/odom,
   /camera_sr300/rgb/image_rect_color,
   /camera_sr300/depth/image_raw,
   /camera_sr300/rgb/camera_info,
   /scan
[ WARN] [1583867179.744564045, 384.647000000]: /rtabmap/rgbd_odometry: Did not r
eceive data since 5 seconds! Make sure the input topics are published ("$ rostop
ic hz my_topic") and the timestamps in their header are set.
/rtabmap/rgbd_odometry subscribed to (approx sync):
   /camera_sr300/rgb/image_rect_color,
   /camera_sr300/depth/image_raw,
   /camera_sr300/rgb/camera_info
[ INFO] [1583867181.225495021, 385.332000000]: rtabmap (18): Rate=1.00s, Limit=0
```

After checking rostopic info, I find rgbd_odometry node subscribes to image_rect_color. So I think I should fix image _rect_color first.

```
turtlebot2i-simulation@ubuntu:~/turtlebot2i/src/turtlebot2i_gazebo/launch$ rosto
pic info /camera_sr300/rgb/image_rect_color
Type: sensor_msgs/Image

Publishers: None

Subscribers:
 * /rtabmap/rtabmapviz (http://ubuntu:42295/)
 * /rtabmap/rgbd_odometry (http://ubuntu:35409/)
```

After checking how the image_rect_color topic is published, I find it seems to be a default name of this package

```
turtlebot2i-simulation@ubuntu:~/turtlebot2i/src/turtlebot2i_gazebo/launch$ grep
-r "/camera/rgb/image_rect_color" "/opt/ros/kinetic/share/"
/opt/ros/kinetic/share/realsense_camera/rviz/realsenseRvizConfiguration3.rviz:
    Image Topic: /camera/rgb/image_rect_color
/opt/ros/kinetic/share/realsense_camera/rviz/realsenseRvizConfiguration2.rviz:
    Image Topic: /camera/rgb/image_rect_color
/opt/ros/kinetic/share/rtabmap_ros/launch/rgbd_mapping.launch:  <arg name="rgb_t
opic"              default="/camera/rgb/image_rect_color" />
/opt/ros/kinetic/share/rtabmap_ros/launch/rtabmap.launch:  <arg name="rgb_topic"
              default="/camera/rgb/image_rect_color" />
/opt/ros/kinetic/share/rtabmap_ros/launch/config/test_odometry.rviz:      Image
Topic: /camera/rgb/image_rect_color
/opt/ros/kinetic/share/rtabmap_ros/launch/config/rgbd.rviz:      Image Topic: /c
amera/rgb/image_rect_color
/opt/ros/kinetic/share/rtabmap_ros/launch/config/rgbdslam_datasets.rviz:      Im
age Topic: /camera/rgb/image_rect_color
/opt/ros/kinetic/share/rtabmap_ros/launch/demo/demo_data_recorder.launch:  <node
 name="republish_rgb"      type="republish" pkg="image_transport" args="compresse
d      in:=/data_throttled_image       raw out:=/camera/rgb/image_rect_color" />
/opt/ros/kinetic/share/rtabmap_ros/launch/demo/demo_turtlebot_mapping.launch:  <
arg unless="$(arg simulation)" name="rgb_topic"   default="/camera/rgb/image_rec
t_color"/>
/opt/ros/kinetic/share/turtlebot_rviz_launchers/rviz/robot.rviz:      Color Imag
e Topic: /camera/rgb/image_rect_color
```

I also find the topic with new camera name has no publisher at all

```
turtlebot2i-simulation@ubuntu:~/turtlebot2i/src/turtlebot2i_gazebo/launch$ rosto
pic info /camera_sr300/rgb/image_rect_color
Type: sensor_msgs/Image

Publishers: None

Subscribers:
 * /rtabmap/rtabmapviz (http://ubuntu:42295/)
 * /rtabmap/rgbd_odometry (http://ubuntu:35409/)
```

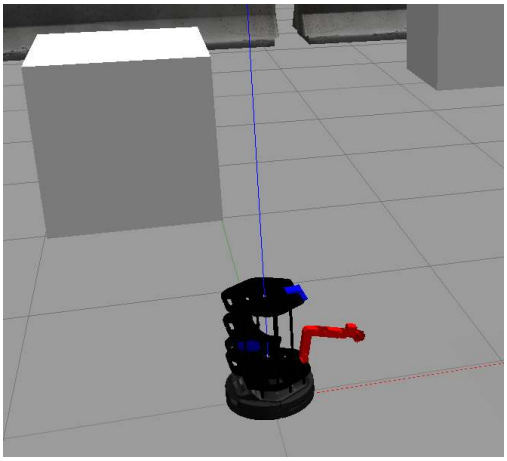
But what is weird is camera/rgb/image_rect_color is not published either.

I decided to change some camera_sr300/rgb/image_rect_color back to camera/rgb/image_rect_color in /turltebot2i_bringup/rtabmap.launch

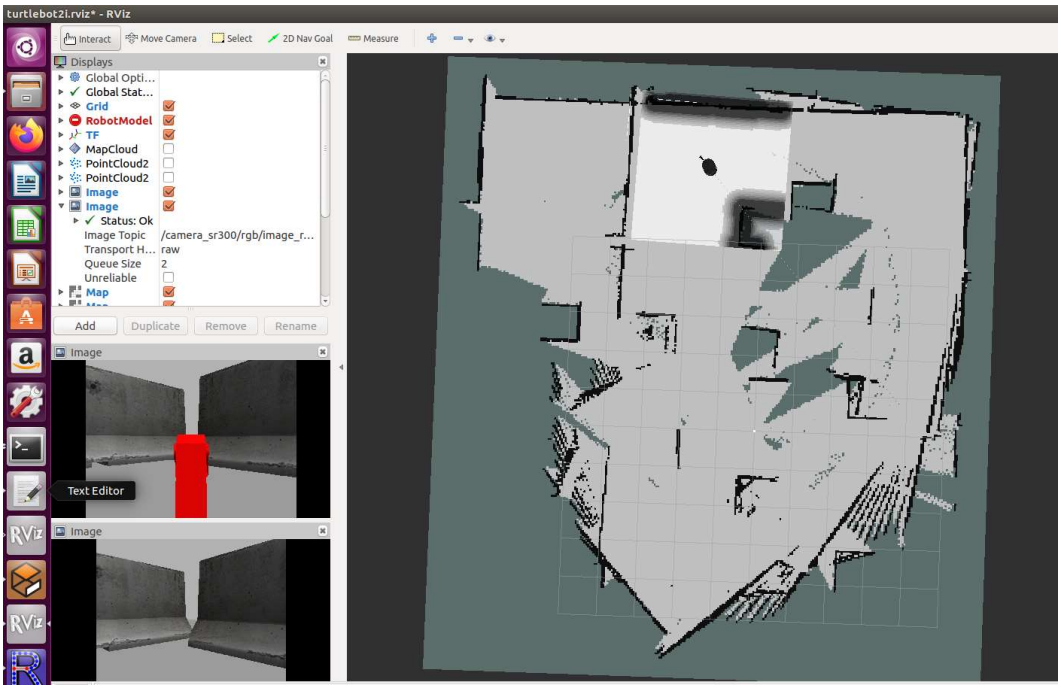
```
    <!-- Odometry : ONLY for testing without the actual robot! /odom TF should not be already
published. -->
    <node if="$(arg rgbd_odometry)" pkg="rtabmap_ros" type="rgbd_odometry" name="rgbd_odometry"
output="screen">
      <param name="frame_id"                    type="string" value="base_footprint"/>
      <param name="wait_for_transform_duration" type="double" value="$(arg wait_for_transform)"/>
      <param name="Reg/Force3DoF"               type="string" value="true"/>
      <param name="Vis/InlierDistance"          type="string" value="0.05"/>

      <remap from="rgb/image"       to="/camera/rgb/image_rect_color"/>
      <remap from="depth/image"     to="/camera_sr300/depth/image_raw"/>
      <remap from="rgb/camera_info" to="/camera_sr300/rgb/camera_info"/>
    </node>

    <!-- visualization with rtabmapviz -->
    <node if="$(arg rtabmapviz)" pkg="rtabmap_ros" type="rtabmapviz" name="rtabmapviz" args="-d
$(find rtabmap_ros)/launch/config/rgbd_gui.ini" output="screen">
        <param name="subscribe_depth"             type="bool" value="true"/>
      <param name="subscribe_scan"              type="bool" value="true"/>
      <param name="frame_id"                    type="string" value="base_footprint"/>
      <param name="wait_for_transform_duration" type="double" value="$(arg wait_for_transform)"/>

      <remap from="rgb/image"       to="/camera/rgb/image_rect_color"/>
```

I don't know whether it will work, maybe the "camera" is only inherited form default="camera"? Anyway, try current modification.

Try to make a map, and I find the arm moves a little. I infer this may be bacause the controller is not ready as soon as Gazebo is ready, namely computaion speed limitation, so it moves a little during that period of time.



Jumping issue again, but this time the parameter set is much more stable than that of Turtlebot. But this jumping is kind of reasonable, as when I turn the robot around, the obstacles matches the previously mapped obstacles

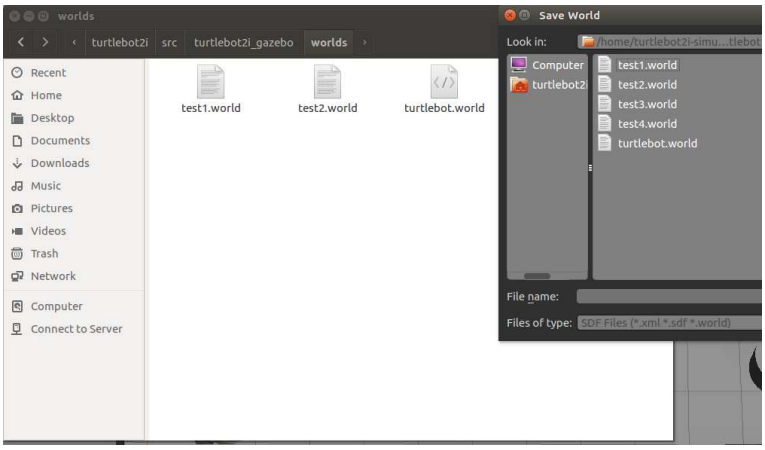


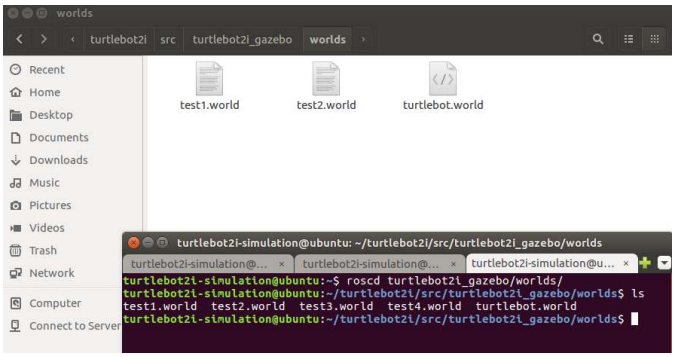
Try again, jump again. I think this is because the environment I created is quite symmetric, so I will modify it a little bit. Maybe use the café model?

For the wheel, when adding café in Gazebo, the robot is lifted while the wheels stay on the same altitude. I launch turtlebot2i_world.launch and turtlebot2i_mapping.launch. The TF tree is complete. Amazing, maybe the gripper and wheel definitions are only called in turtlebot2i_mapping.launch??

Created a new Gazebo world with less similar features

Interesting, I created a new world and use terminal to double check it. It's there. But when I call it, it's not found. I open GUI to find it, and it's not there. Now again.

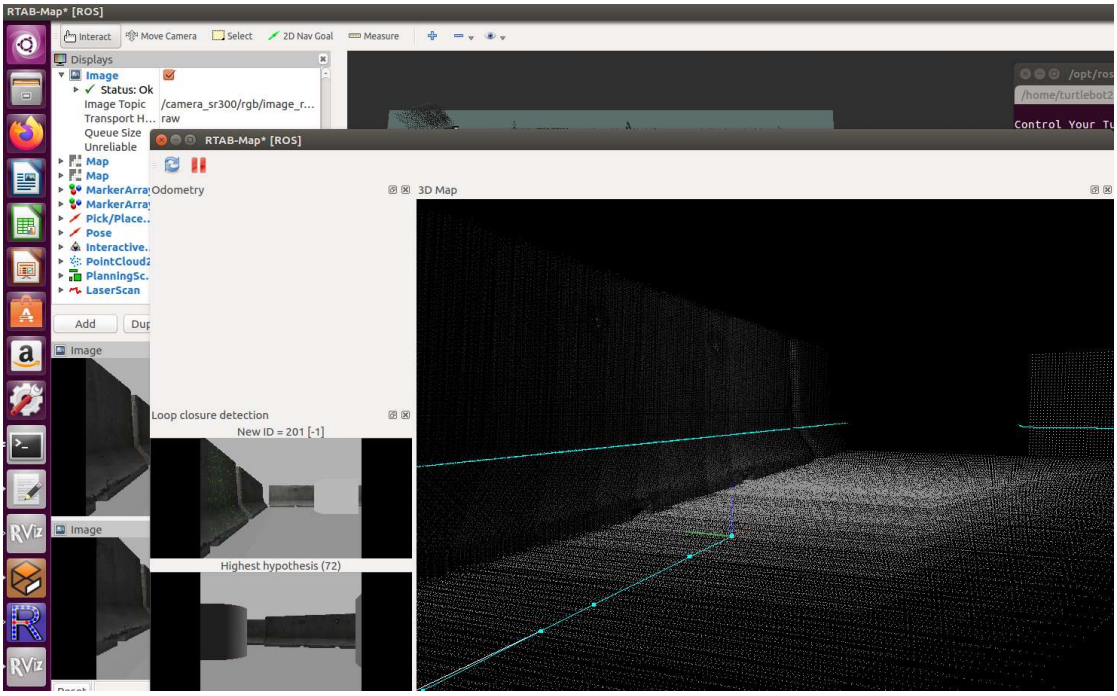


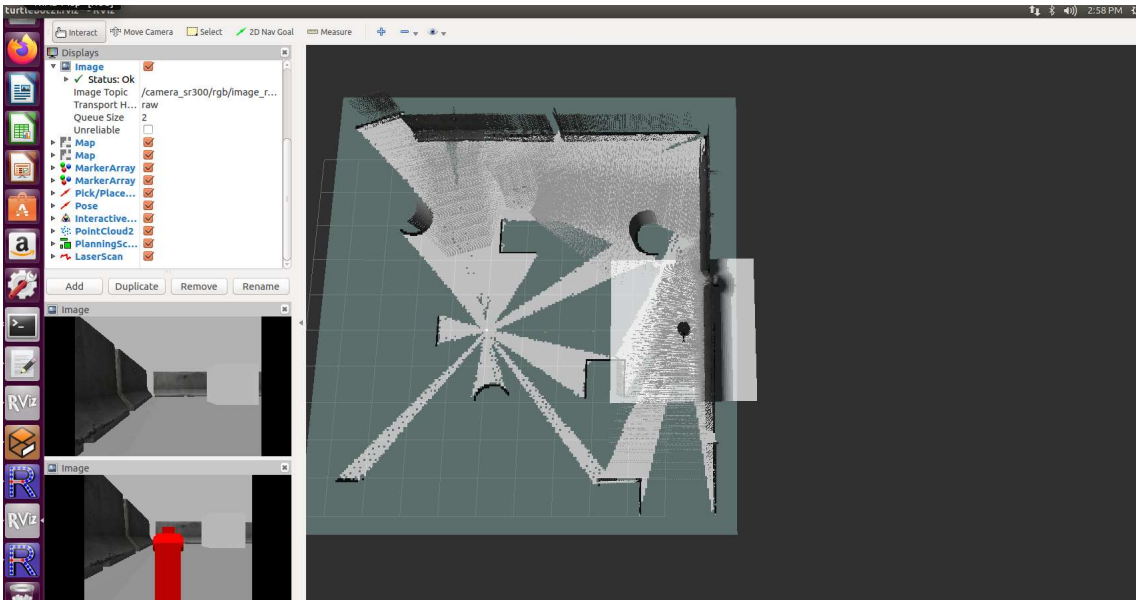


Reboot, and it gets normal. World files appear.

I don't like the slight move of arm, so I set the pause flag to "true", and will resume Gazebo manually.

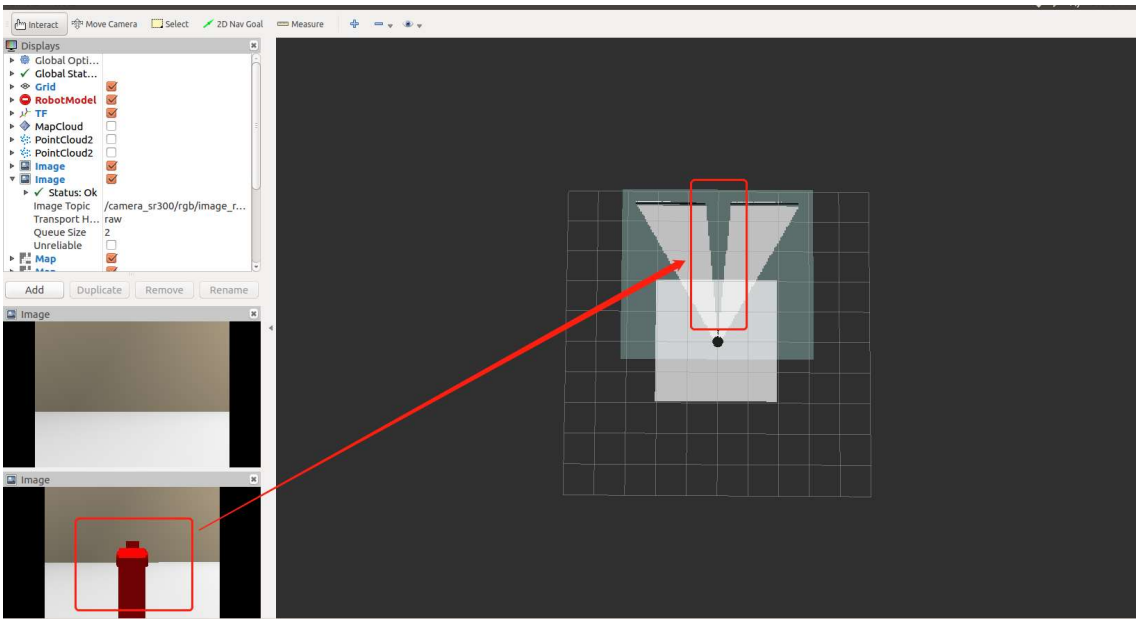
I just find things work kind of so nice and the map in Rviz seems to be a 3D one. Well no, it turns out the 3D illusion in Rviz is just because I failed to disable the display of map cloud.



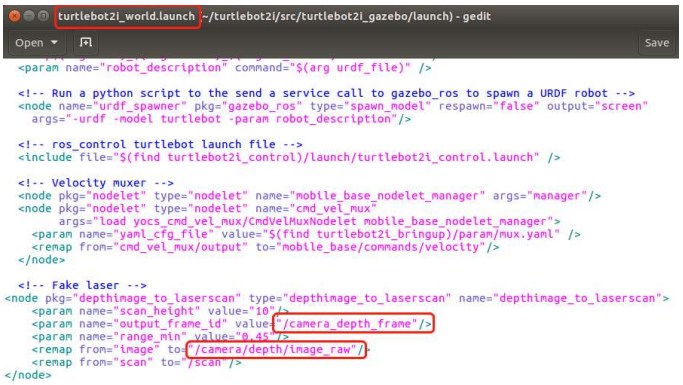


After a while, it jumps again, and the surrounding obstacles still matches previously mapped obstacles well. Now I use the café, but sink it a little to make the robot not lifted by the café floor.

The laserscan data is kind of weird. I infer that because depthimage_to_laserscan is not taking the demanded input image.

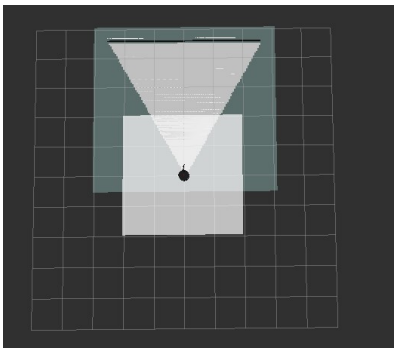


After checking, I changed these to camera_sr300 related topics

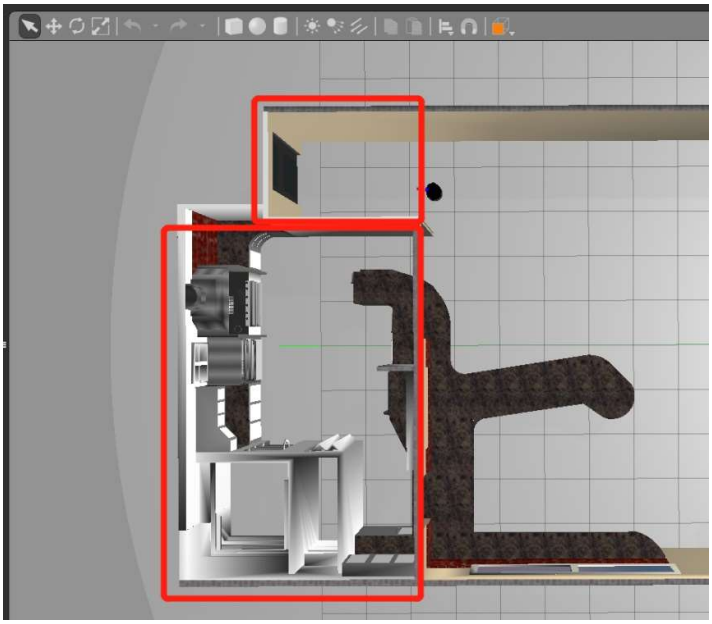


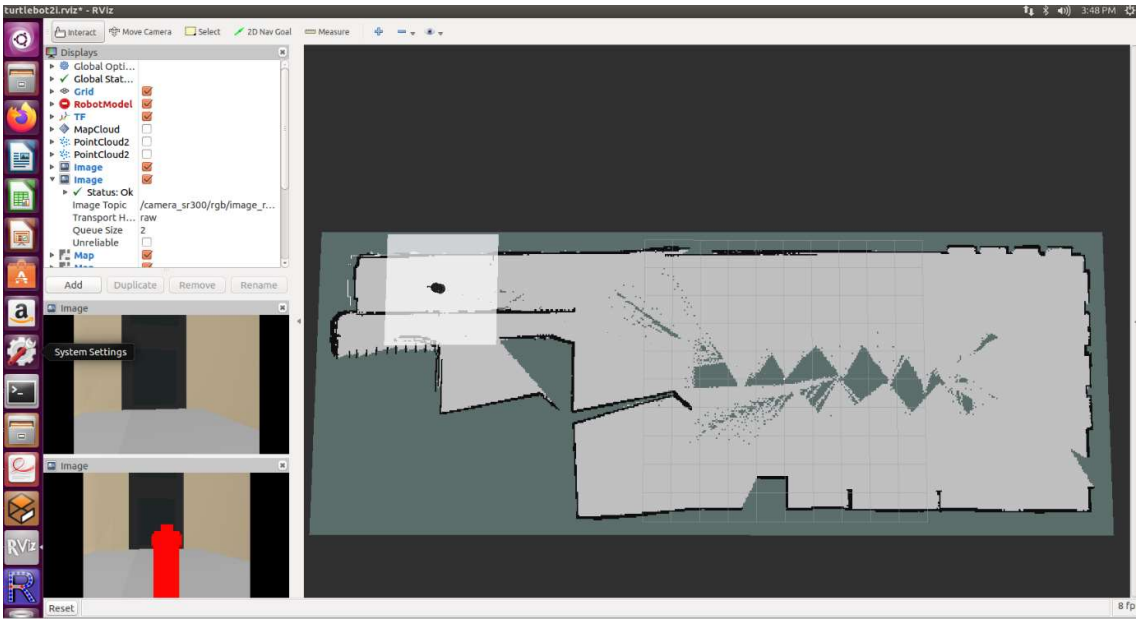
I find another parameter scan_height, as it can be defined arbitrarily, I think it should be defined according to the configuration of robot

Nice, things are working in the right way



Things work almost in the right way. What is interesting is that these 2 parts look accessible but actually not.



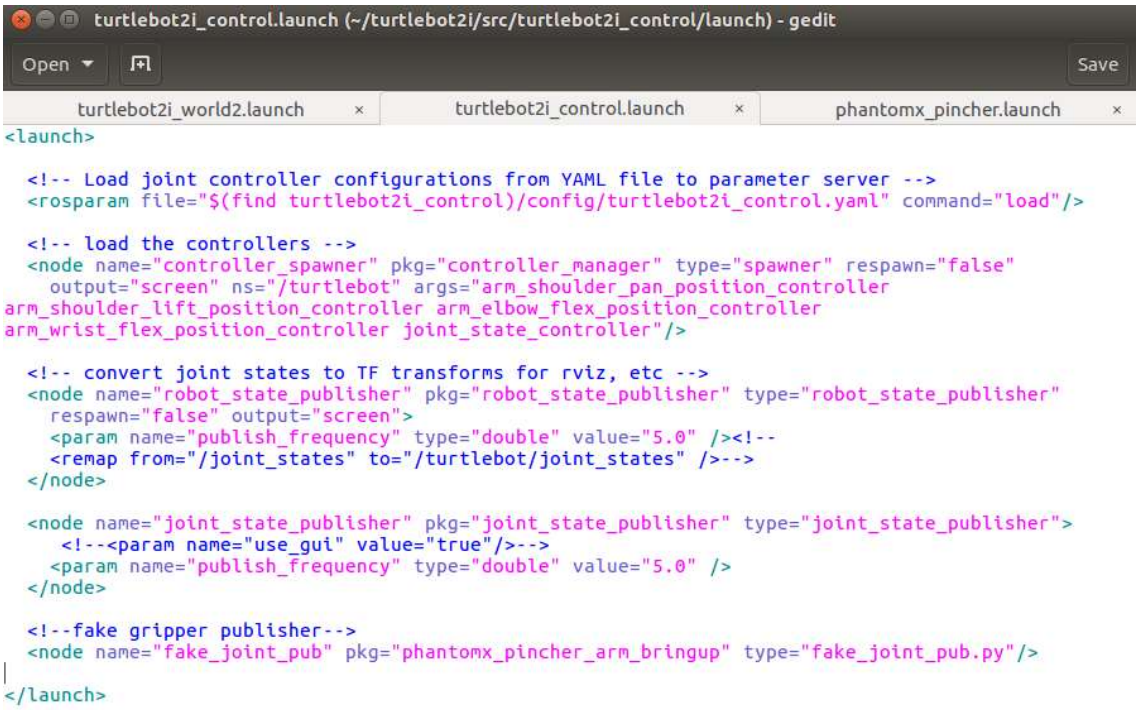


Okay, make a git copy.

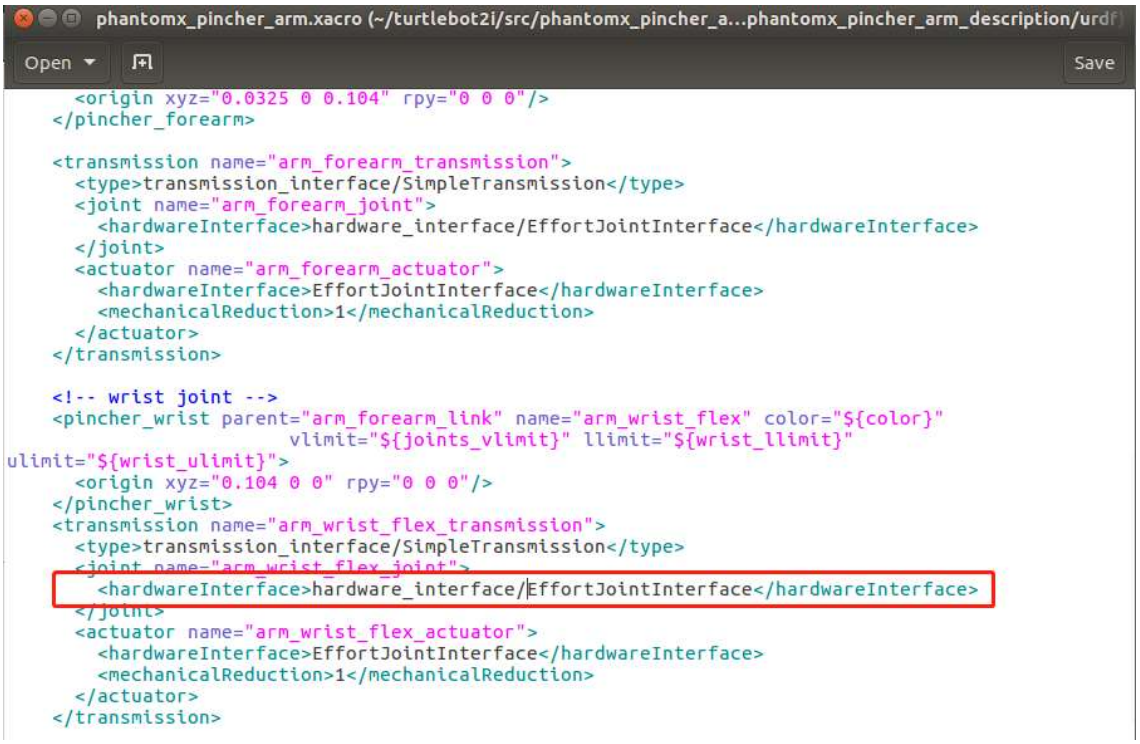
04/03/2020 update

The problem now is transform of movable links, i.e. wheel links and arm links, are published to different topics, i.e. /joint_states and /turtlebot/joint_states.

I thought that joint_state_publiser will publish all movable joint information to /joint_states. But it was wrong, it is just a GUI to manually change moveable joint angles and publish transformation to /tf. The fake gripper publisher published a joint of gripper, which helps to complete transform tree as well. But transform of all links is complete.



Prepended “hardware_interface/” for 4 joints



The file hierarchy is kind of massed up originally, this shouldn’t be invoked in a camera description file. More specifically, this gazebo_control plugin.

turtlebot_gazebo.urdf.xacro (~/turtlebot2i/src/turtlebot2i_description/urdf) - gedit

Open

```
<?xml version="1.0"?>
<robot name="turtlebot_gazebo" xmlns:xacro="http://ros.org/wiki/xacro">
  <gazebo>
    <plugin filename="libgazebo_ros_control.so" name="gazebo_ros_control">
        <robotNamespace>/turtlebot</robotNamespace>
        <robotSimType>gazebo_ros_control/DefaultRobotHWSim</robotSimType>
        <legacyModeNS>true</legacyModeNS>

    </plugin>
  </gazebo>
```

By the way, <robotNamespace>/turtlebot</robotNamespace> above determines the prefix of topics related to controller

```
/turtlebot/arm_elbow_flex_position_controller/command
/turtlebot/arm_elbow_flex_position_controller/pid/parameter_descriptions
/turtlebot/arm_elbow_flex_position_controller/pid/parameter_updates
/turtlebot/arm_elbow_flex_position_controller/state
/turtlebot/arm_shoulder_lift_position_controller/command
/turtlebot/arm_shoulder_lift_position_controller/pid/parameter_descriptions
/turtlebot/arm_shoulder_lift_position_controller/pid/parameter_updates
/turtlebot/arm_shoulder_lift_position_controller/state
/turtlebot/arm_shoulder_pan_position_controller/command
/turtlebot/arm_shoulder_pan_position_controller/pid/parameter_descriptions
/turtlebot/arm_shoulder_pan_position_controller/pid/parameter_updates
/turtlebot/arm_shoulder_pan_position_controller/state
/turtlebot/arm_wrist_flex_position_controller/command
/turtlebot/arm_wrist_flex_position_controller/pid/parameter_descriptions
/turtlebot/arm_wrist_flex_position_controller/pid/parameter_updates
/turtlebot/arm_wrist_flex_position_controller/state
/turtlebot/joint_states
```

rostopic pub /turtlebot/gripper1_position_controller/command std_msgs/Float64 -- 0.02

Summary of modification:

You can choose to add publish_frequency or not, or change its value.

turtlebot2i_control.launch × phantomx_pincher.launch × kobuki_interbotix_astra.urdf.xacro ×

```
<launch>

  <!-- Load joint controller configurations from YAML file to parameter server -->
  <rosparam file="$(find turtlebot2i_control)/config/turtlebot2i_control.yaml" command="load"/>

  <!-- load the controllers -->
  <node name="controller_spawner" pkg="controller_manager" type="spawner" respawn="false"
    output="screen" ns="/turtlebot" args="arm_shoulder_pan_position_controller
arm_shoulder_lift_position_controller arm_elbow_flex_position_controller
arm_wrist_flex_position_controller joint_state_controller gripper1_position_controller
gripper2_position_controller"/>

  <!-- convert joint states to TF transforms for rviz, etc -->
  <node name="robot_state_publisher" pkg="robot_state_publisher" type="robot_state_publisher"
    respawn="false" output="screen">
    <param name="publish_frequency" type="double" value="5.0" />
    <remap from="/joint_states" to="/turtlebot/joint_states" />
  </node>
```

Change <hardwareInterface> tag of 4 arm joints to “hardware_interface/EffortJointInterface”

phantomx_pincher_arm.xacro (~/turtlebot2i/src/phantomx_pincher_a...phantomx_pincher_arm_description/urdf

Open Save

```
  <!--joints_vlimit pan_llimit pan_ulimit-->
  <xacro:macro name="phantomx_pincher_arm" params="parent color gripper_color *origin">

    <!-- arm base link -->
    <pincher_base parent="${parent}" name="arm_base">
      <xacro:insert_block name="origin"/>
    </pincher_base>
    <bracket_F2_revolute parent="arm_base_link" name="arm_shoulder_pan" color="${color}"
                         vlimit="${joints_vlimit}" llimit="${pan_llimit}" ulimit="${pan_ulimit}">
      <origin xyz="0 0 0.020" rpy="0 0 0"/>
    </bracket_F2_revolute>
    <transmission name="arm_shoulder_pan_transmission">
      <type>transmission_interface/SimpleTransmission</type>
      <joint name="arm_shoulder_pan_joint">
        <hardwareInterface>hardware_interface/EffortJointInterface</hardwareInterface>
      </joint>
      <actuator name="arm_shoulder_pan_actuator">
        <hardwareInterface>EffortJointInterface</hardwareInterface>
        <mechanicalReduction>1</mechanicalReduction>
      </actuator>
    </transmission>
```

```
    <!-- shoulder lift joint -->
    <pincher_bicep parent="arm_shoulder_pan_link" name="arm_bicep" color="${color}"
                           vlimit="${joints_vlimit}" llimit="${shoulder_llimit}"
ulimit="${shoulder_ulimit}">
      <origin xyz="0 0 0.027" rpy="0 0 0"/>
    </pincher_bicep>

    <transmission name="arm_bicep_transmission">
      <type>transmission_interface/SimpleTransmission</type>
      <joint name="arm_bicep_joint">
        <hardwareInterface>hardware_interface/EffortJointInterface</hardwareInterface>
      </joint>
      <actuator name="arm_bicep_actuator">
        <hardwareInterface>EffortJointInterface</hardwareInterface>
        <mechanicalReduction>1</mechanicalReduction>
      </actuator>
    </transmission>
```

```
    <!-- elbow joint -->
    <pincher_forearm parent="arm_bicep_link" name="arm_forearm" color="${color}"
                     vlimit="${joints_vlimit}" llimit="${elbow_llimit}"
ulimit="${elbow_ulimit}">
      <origin xyz="0.0325 0 0.104" rpy="0 0 0"/>
    </pincher_forearm>

    <transmission name="arm_forearm_transmission">
      <type>transmission_interface/SimpleTransmission</type>
      <joint name="arm_forearm_joint">
        <hardwareInterface>hardware_interface/EffortJointInterface</hardwareInterface>
      </joint>
      <actuator name="arm_forearm_actuator">
        <hardwareInterface>EffortJointInterface</hardwareInterface>
        <mechanicalReduction>1</mechanicalReduction>
      </actuator>
    </transmission>
```

```
    <!-- wrist joint -->
    <pincher_wrist parent="arm_forearm_link" name="arm_wrist_flex" color="${color}"
                     vlimit="${joints_vlimit}" llimit="${wrist_llimit}"
ulimit="${wrist_ulimit}">
      <origin xyz="0.104 0 0" rpy="0 0 0"/>
    </pincher_wrist>
    <transmission name="arm_wrist_flex_transmission">
      <type>transmission_interface/SimpleTransmission</type>
      <joint name="arm_wrist_flex_joint">
        <hardwareInterface>hardware_interface/EffortJointInterface</hardwareInterface>
      </joint>
      <actuator name="arm_wrist_flex_actuator">
        <hardwareInterface>EffortJointInterface</hardwareInterface>
        <mechanicalReduction>1</mechanicalReduction>
      </actuator>
    </transmission>
```

Comment all the contents at the end of phantomx_pincher_arm.xacro, apart from those in red rectangles. It seems to me that this link is useless and I met problem when using the gazebo part below that.

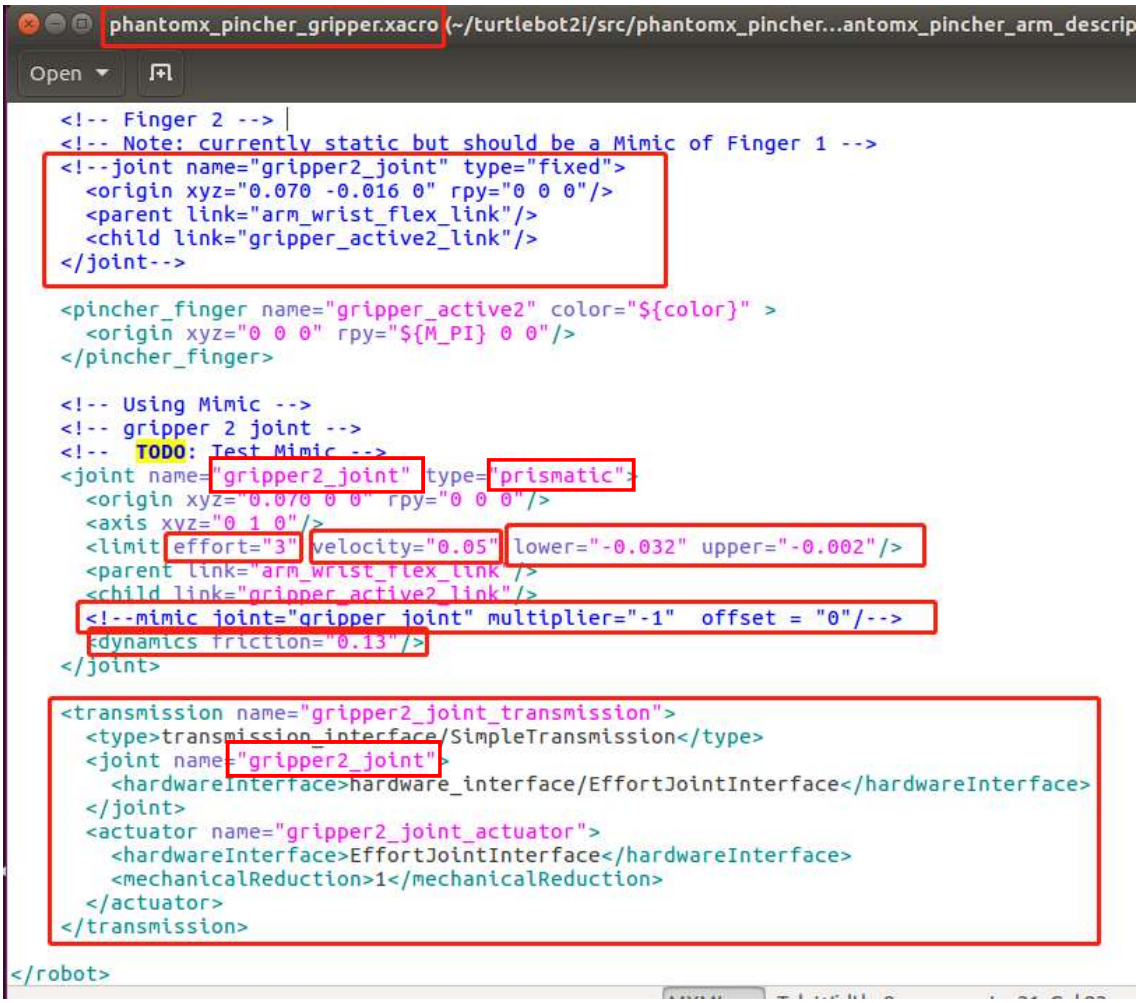
In phantomx_pincher_gripper.xacro, do the following modification.

phantomx_pincher_gripper.xacro (~/turtlebot2i/src/phantomx_pincher...antomx_pincher_arm_descrip

Open

```
<?xml version="1.0"?>
<!-- PhantomX Pincher gripper URDF-->
<robot xmlns:xacro="http://ros.org/wiki/xacro">

    <!-- gripper joint -->
    <!-- Finger 1 -->
    <joint name="gripper_joint" type="prismatic">
      <origin xyz="0.070 0 0" rpy="0 0 0"/>
      <axis xyz="0 1 0"/>
      <limit effort="3" velocity="0.05" lower="0.002" upper="0.032"/>
      <parent link="arm_wrist_flex_link"/>
      <child link="gripper_active_link"/>
      <dynamics friction="0.13"/>
    </joint>

    <pincher_finger name="gripper_active" color="${color}" >
      <origin xyz="0 0 0" rpy="0 0 0"/>
    </pincher_finger>

    <transmission name="gripper_joint_transmission">
      <type>transmission_interface/SimpleTransmission</type>
      <joint name="gripper_joint" >
        <hardwareInterface>hardware_interface/EffortJointInterface</hardwareInterface>
      </joint>
      <actuator name="gripper_joint_actuator">
        <hardwareInterface>EffortJointInterface</hardwareInterface>
        <mechanicalReduction>1</mechanicalReduction>
      </actuator>
    </transmission>
```

I changed the name of joint, because I think it would be more explicit.
effort is decreased to avoid large repulsion when grippers are in conflict with objects. The original effort value 30 will make the robot to fall down in some cases.
velocity is decreased because the original velocity is too large that the robot will be twisted influenced by the movement of grippers.
A friction parameter is added to avoid fluctuation of grippers after moving them.
Transmission is essential for publishing of gripper link in /tf and its actuation.



The lower and upper parameter is modified because they should be identical to those of gripper1_joint

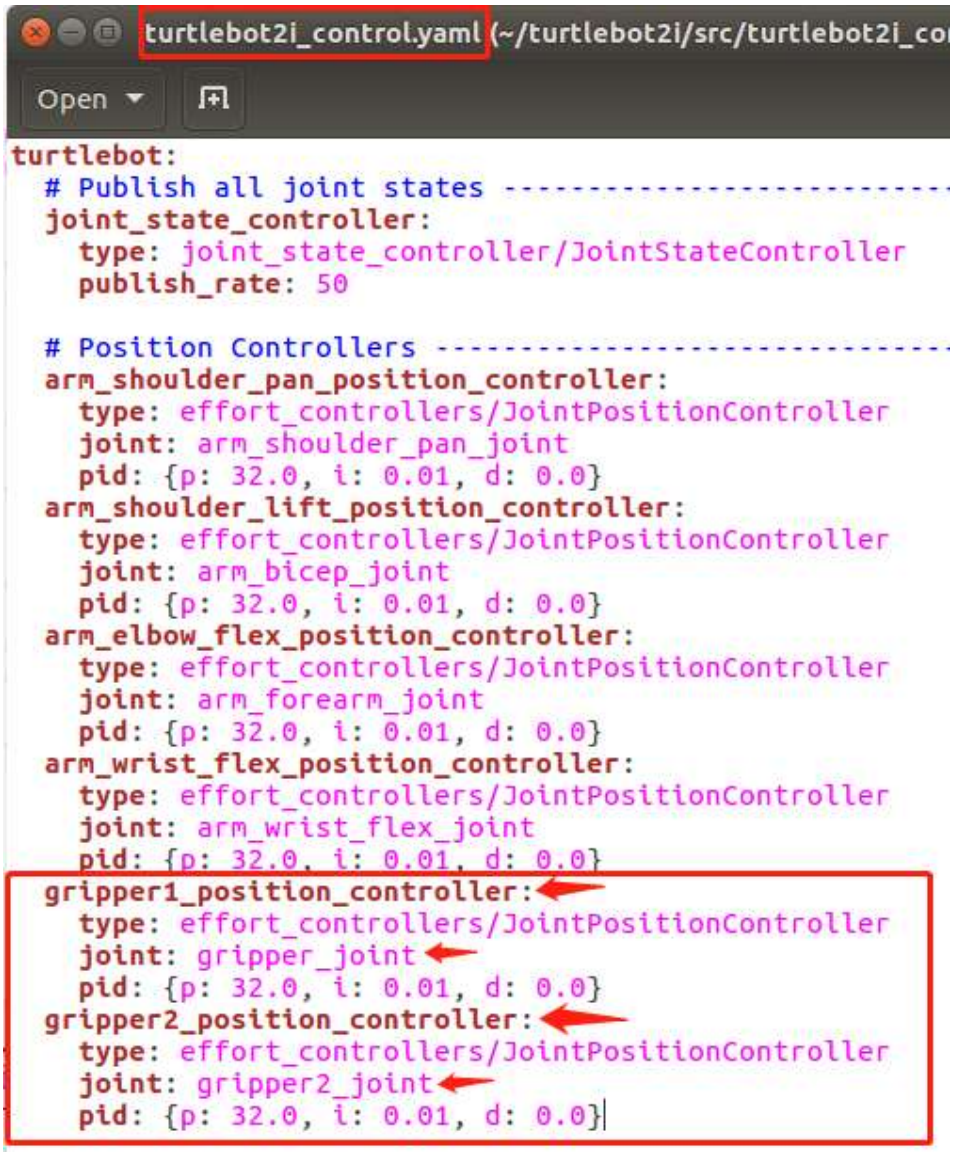
The mimic will be valid when you use joint_state_publisher_gui or visualize your URDF file in Rviz using

```
roslaunch urdf_tutorial display.launch model:=<your urdf file>
```

reference:
https://answers.ros.org/question/10401/how-to-convert-xacro-file-to-urdf-file/
http://wiki.ros.org/urdf/Tutorials/Building%20a%20Visual%20Robot%20Model%20with%20URDF%20from%20Scratch
But mimic tag doesn't work for Gazebo, so I abandoned it. After googling for a while, it seems to me that Gazebo doesn't have official mimic functionality. So I decided to use 2 controllers and 2 prismatic joints to actuate the 2 grippers, which is different from what's happening on real robot that grippers are actuated by the same servo.

Now modify the controller:

Append 2 more controllers for 2 grippers respectively. Here is how the controllers are defined: Make sure names of controllers and joints are consistent as those you defined elsewhere



I uploaded a new launch file “turtlebot2i_world2.launch”, which will start Gazebo while won’t start rtabmap and Rviz.

After running “turtlebot2i_world2.launch” you will be able to control the arm using command similar to the following in terminal:
rostopic pub /turtlebot/gripper2_position_controller/command std_msgs/Float64 -- -0.02

To run Rviz:
roslaunch turtlebot2i_bringup rviz.launch

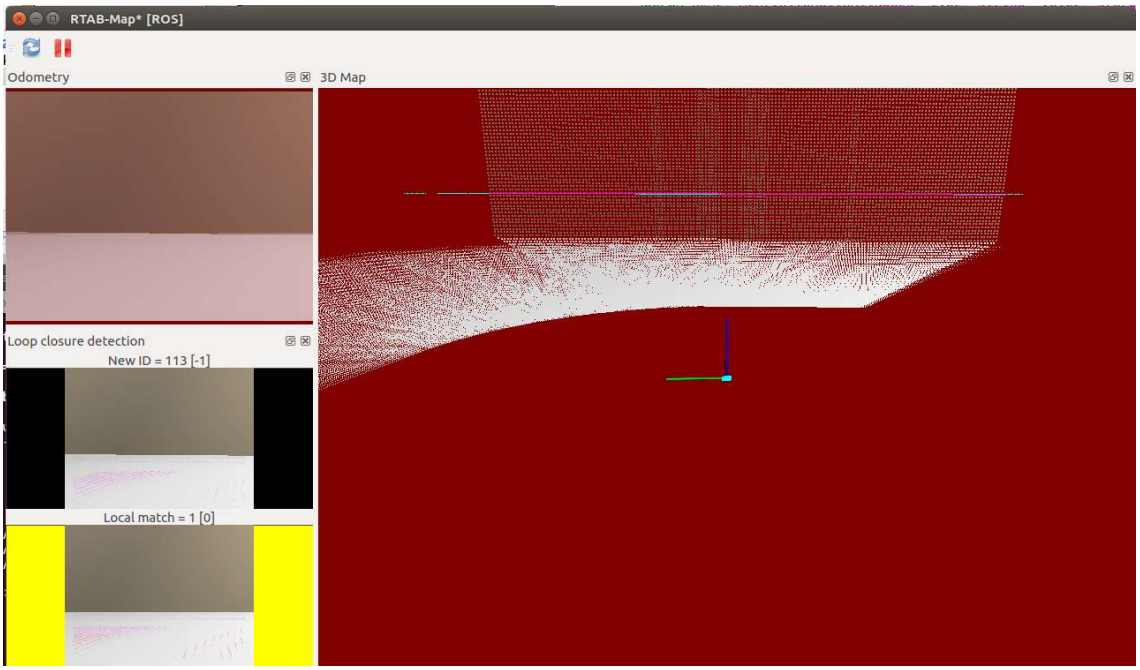
To build a new map using rtabmap:
roslaunch turtlebot2i_bringup rtabmap.launch args:=--delete_db_on_start

I would strongly recommend you look through the launch files to figure out what arguments you should pass in the command.

The remaining problems now are:
1. Gripper doesn’t work accurately, and the process of grasping, i.e. when in collision with other objects, it still doesn’t work like that the real world. I think it can be optimized by changing the gripper speed, using optimal PID parameters and increasing the friction.
2. It seems to me that the gripper doesn’t have force feedback in gazebo. So when it grasps things, it is likely to be too loose(object falls) or too tight(the robot will be shaking…).
3. New problems are likely to show up during the process of developing tasks.
4. The transform of wheels are still missing. It can be fixed if we can remap message in /joint_state to /turtlebot_joint_state, but I haven’t found how to do that. A fundamental way is to write a subscriber of /joint_state and publish the message to /turtlebot_joint state. Anyway, the absence of these 2 links is not a big problem, as far as I can see, it won’t affect the simulation.

Future Tasks:
1. Do path planning or even path planning through way points.
2. rtabmap seems to be an interesting package, and I will learn more about that.
3. Grasping of objects using logical camera.

I modified rtabmap.launch a little bit, basically only use information of sr300 camera, as astra is partially blocked by the arm. But something weird happened. I don’t know why things are turning red:


RTAB-Map* [ROS]
Odometry
3D Map
Loop closure detection
New ID = 113 [-1]
Local match = 1 [0]