ASSIGNMENT 4

1a) Determine the values of xL, yL, and θ, corresponding to TurtleBot3 Burger. One way of doing so, is to use the TF display in Rviz. TurtleBot3 calls its world, robot, and Lidar frames, as odom, base_link, and base_scan, respectively. Another way is to use the terminal command rosrun tf tf_echo. For instance, typing the following command at the command line, prints the relative position and orientation of /base_link with respect to /odom: rosrun tf tf_echo /odom /base_link.

ANSWER:

Step 1: We use turtlebot3 of model type burger for this assignment.

To export and launch this, we use the following command by opening a new terminal.

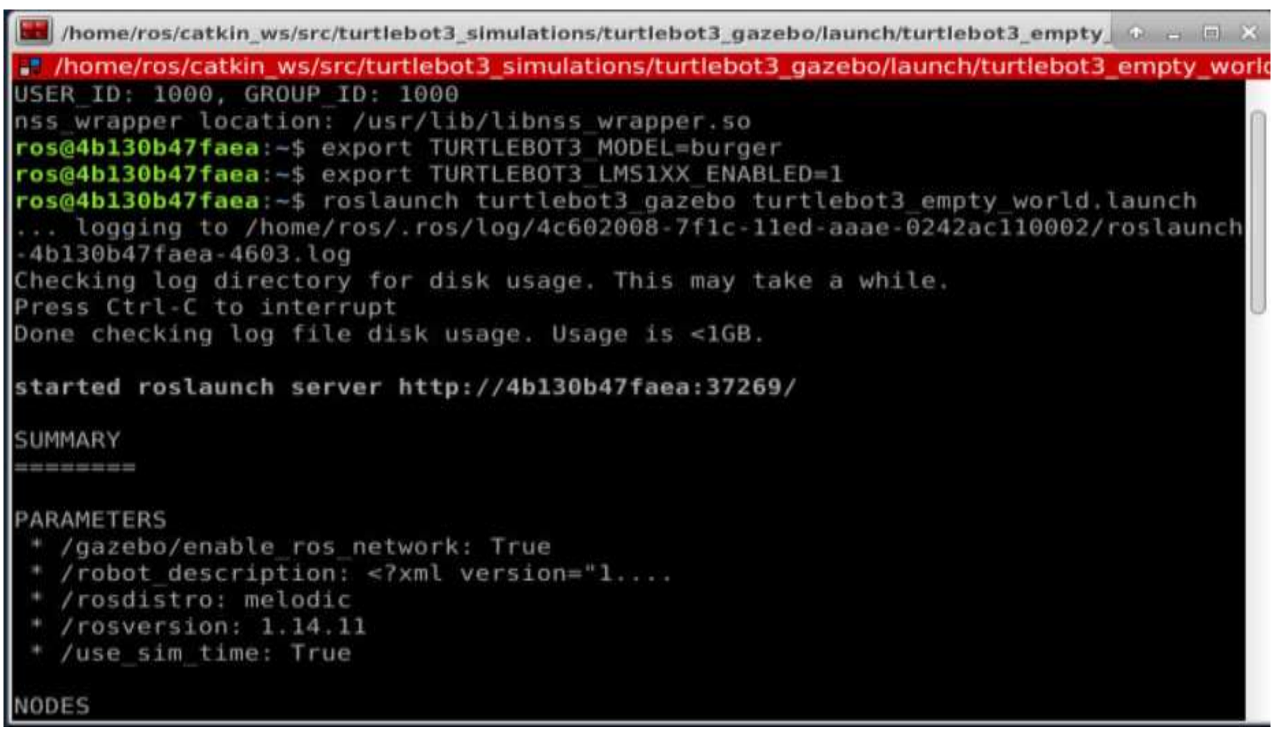
$ export TURTLEBOT3_MODEL=burger

Step 2: The next step is to enable the LiDAR sensor in this turtlebot. The command we use for this is given below

$ export TURTLEBOT3_LMS1XX_ENABLED=1

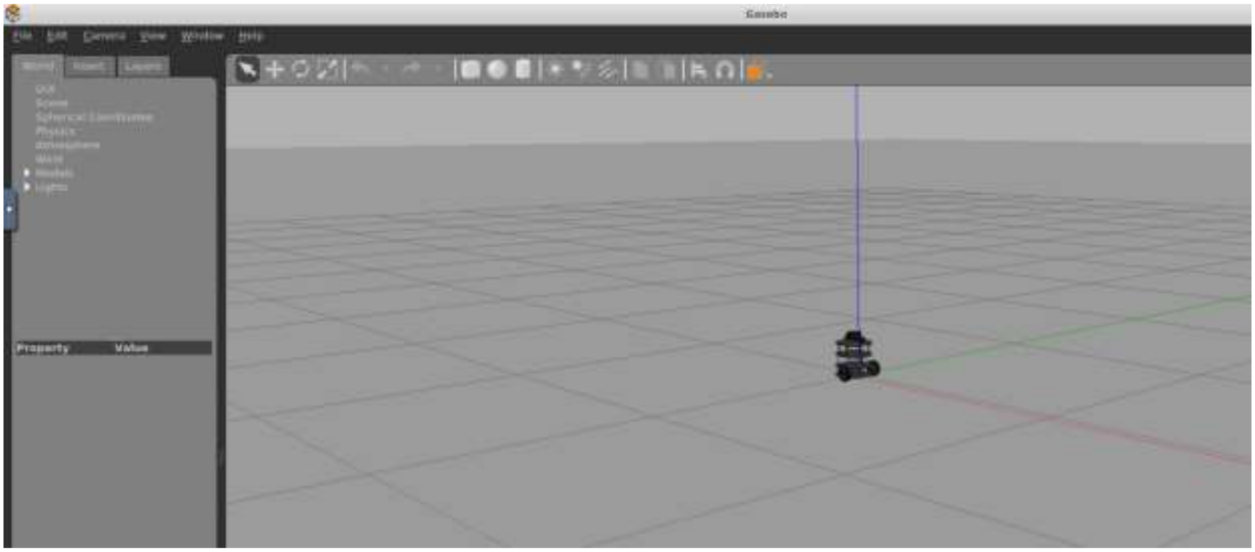
Step 3: Now, we need to launch the gazebo along with the spawning of turtlebot3 in an empty_world environment. To do this we use the following command.

$ roslaunch turtlebot3_gazebo turtlebot3_empty_world.launch



```
/home/ros/catkin_ws/src/turtlebot3_simulations/turtlebot3_gazebo/launch/turtlebot3_empty_
/home/ros/catkin_ws/src/turtlebot3_simulations/turtlebot3_gazebo/launch/turtlebot3_empty_worl
USER_ID: 1000, GROUP_ID: 1000
nss_wrapper location: /usr/lib/libnss_wrapper.so
ros@4b130b47faea:~$ export TURTLEBOT3_MODEL=burger
ros@4b130b47faea:~$ export TURTLEBOT3_LMS1XX_ENABLED=1
ros@4b130b47faea:~$ roslaunch turtlebot3_gazebo turtlebot3_empty_world.launch
... logging to /home/ros/.ros/log/4c602008-7f1c-11ed-aaae-0242ac110002/roslaunch
-4b130b47faea-4603.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://4b130b47faea:37269/

SUMMARY
========

PARAMETERS
 * /gazebo/enable_ros_network: True
 * /robot_description: <?xml version="1....
 * /rosdistro: melodic
 * /rosversion: 1.14.11
 * /use_sim_time: True

NODES
```

The first three commands are used in the above snippet.



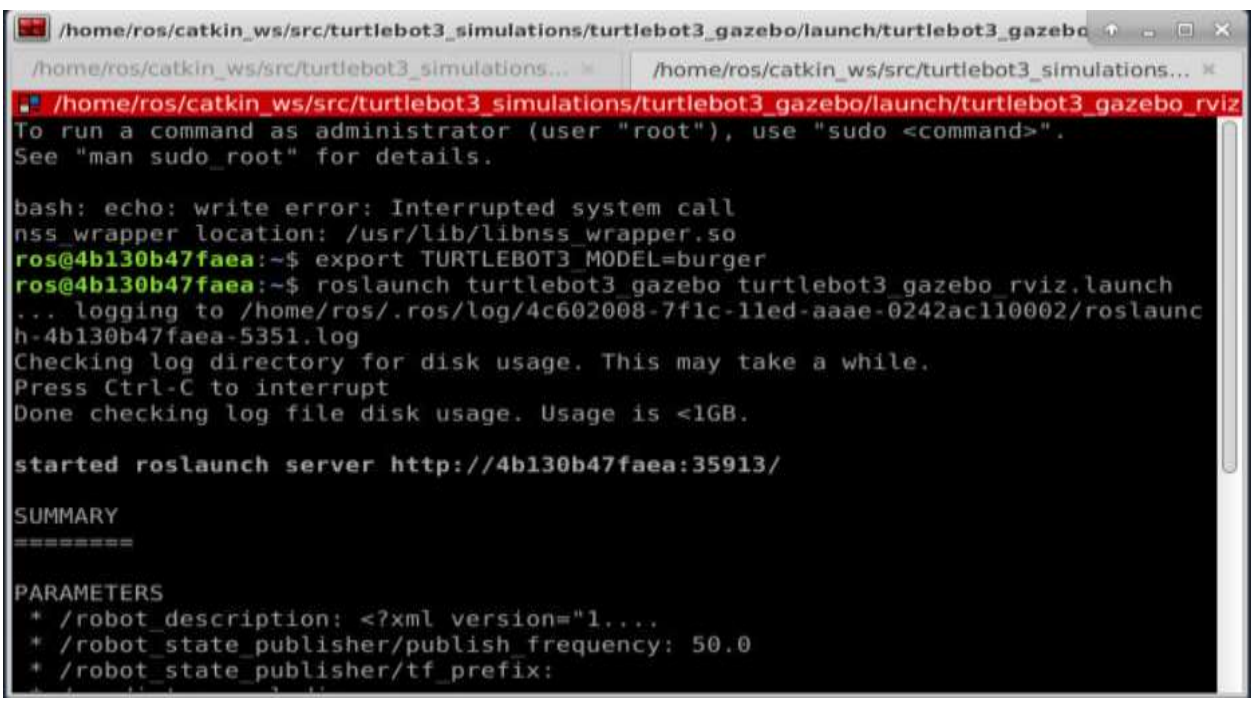
Gazebo is launched along with turtlebot3 of model burger spawned.

Step 4: Now open a new terminal and enter the export model command again

$ export TURTLEBOT3_MODEL=burger

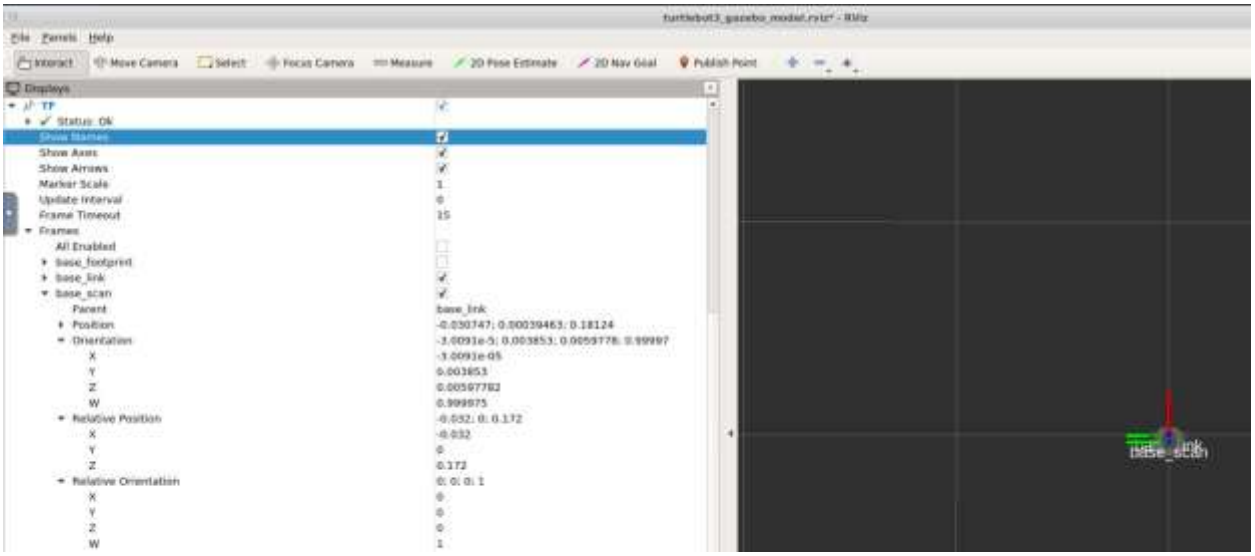
Step 5: To launch the Rviz and spawn the turtlebot3, we use the following command, which spawns the turtlebot3 of model type ‘burger’ in the Rviz environment.

$ roslaunch turtlebot3_gazebo turtlebot3_gazebo_rviz.launch



```
/home/ros/catkin_ws/src/turtlebot3_simulations/turtlebot3_gazebo/launch/turtlebot3_gazebo
/home/ros/catkin_ws/src/turtlebot3_simulations...    /home/ros/catkin_ws/src/turtlebot3_simulations...
/home/ros/catkin_ws/src/turtlebot3_simulations/turtlebot3_gazebo/launch/turtlebot3_gazebo_rviz
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

bash: echo: write error: Interrupted system call
nss_wrapper location: /usr/lib/libnss_wrapper.so
ros@4b130b47faea:~$ export TURTLEBOT3_MODEL=burger
ros@4b130b47faea:~$ roslaunch turtlebot3_gazebo turtlebot3_gazebo_rviz.launch
... logging to /home/ros/.ros/log/4c602008-7f1c-11ed-aaae-0242ac110002/roslaunc
h-4b130b47faea-5351.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://4b130b47faea:35913/

SUMMARY
========

PARAMETERS
 * /robot_description: <?xml version="1....
 * /robot_state_publisher/publish_frequency: 50.0
 * /robot_state_publisher/tf_prefix:
```

The next two commands are entered in a new terminal as shown in the above snippet

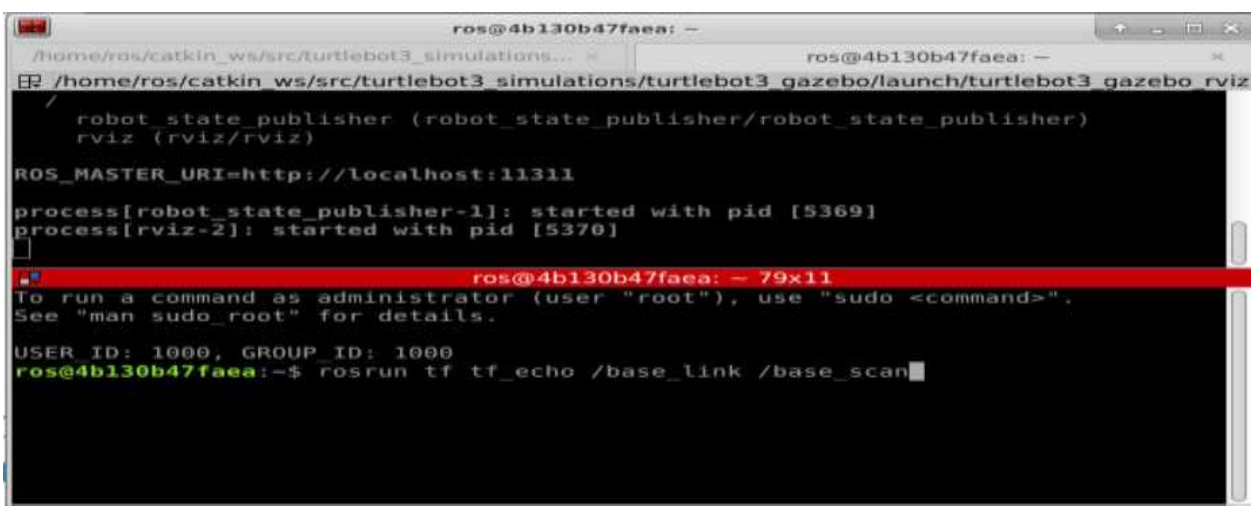


Rviz is launched along with the spawning of turtblebot3 of model burger in the Rviz environment.

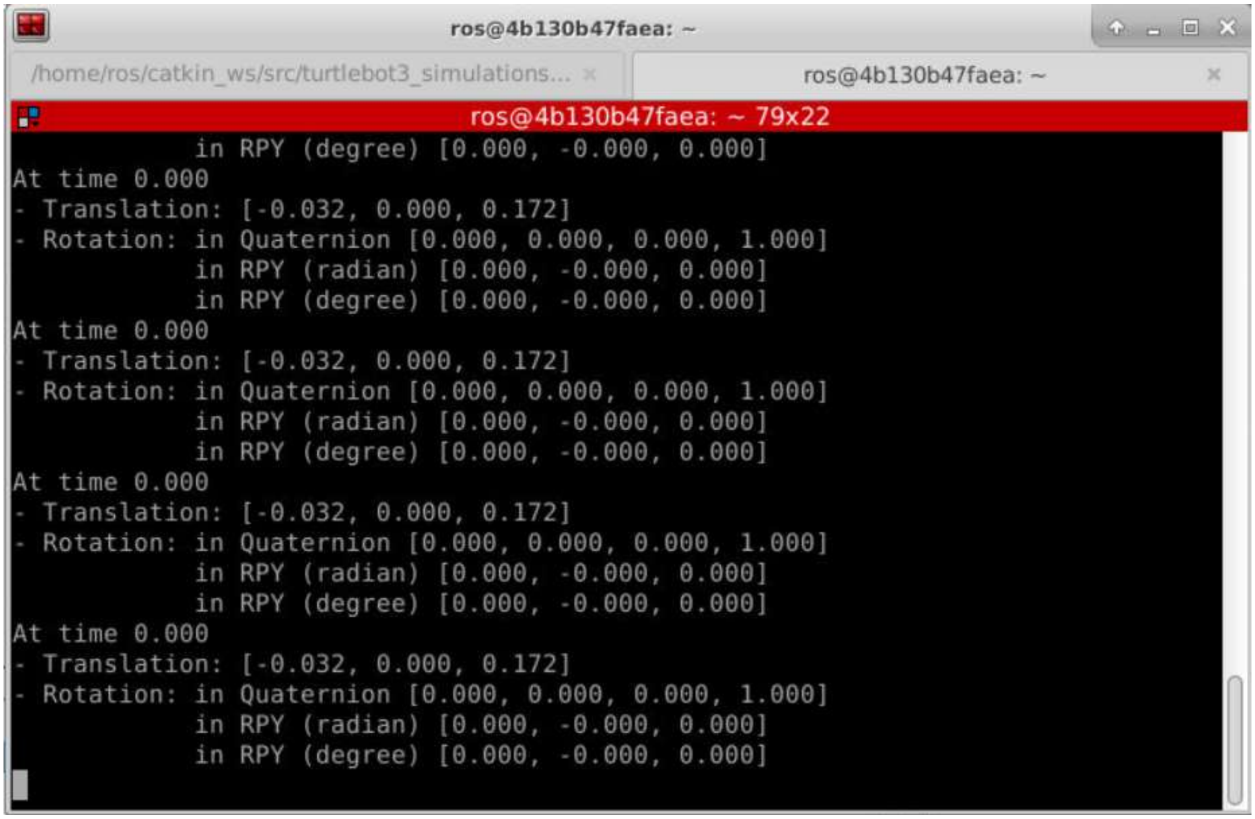
The values of xL, yL, and θ can be determined from the TF display of Rviz as shown in the above snippet. This is done by enabling the TF display, and checking only the base_link and base_scan check boxes under frames. Now, when you expand the base_scan. Here, The relative X position is the value of xL, the relative Y position is the value of yL and orientation of Z value is the value of θ. Hence, xL = -0.032 meters, $yL = 0.00$ meters, $\theta = 0.00$ degrees.

Step 6: To determine the values of xL, yL, and θ, corresponding to TurtleBot3 Burger, we use second way of determining, ie, by using the terminal command rosrun tf tf_echo.

$ rosrun tf tf_echo /base_link /base_scan



The above command displays the translation and orientation components in the terminal, which is displayed in the below snippet. This is another way of determining the values.



Step 7: Now, typing the above command at the command line, prints the relative position and orientation of /base_link in the terminal. We can determine the values of xL, yL, and θ, from the values displayed on the terminal.

We take the value of xL, from the x value and yL, from the y value of the translation components that are displayed. Thus, we get the values of $xL =$ -0.032 meters and $yL = 0.00$ meters.

The orientation can be determined from the Rotation components that are displayed in the terminal. Thus, we get the value of $\theta = 0.00$ degrees.

Hence, we use the values of xL, yL, and θ as follows in my code for the other sub-divisions of the assignment

$xL =$ -0.032 meters

$yL = 0.00$ meters

$\theta = 0.00$ degrees