

1. Install Page 80

```
$ sudo apt-get install ros-kinetic-turtlebot ros-kinetic-turtlebot-apps  
ros-kinetic-turtlebot-interactions ros-kinetic-turtlebot-simulator ros-  
kinetic-kobuki-ftdi
```

2. Check it

```
harman@D104-45931:~$ rospack list | grep turtlebot
```

```
    turtlebot_gazebo /opt/ros/kinetic/share/turtlebot_gazebo (Simulator)
```

3. The Base of Real TurtleBot

```
harman@D104-45931:~$ rospack list | grep kobuki
```

4. Gazebo

```
$ roslaunch turtlebot_gazebo turtlebot_world.launch
```

4_1 Click on TurtleBot and View Pose

TurtleBot is about in the center x,y,z, roll, pitch, yaw Notice not 0,0,0,0,0,0

Check /odom

```
harman@D104-45931:~$ rostopic echo /odom -n 1
```

header:

seq: 43618

stamp:

secs: 440

nsecs: 250000000

frame_id: odom

child_frame_id: base_footprint

pose:

pose:

position:

x: 6.76578452429e-07

y: 6.31646141337e-07

z: 0.0

orientation:

x: 0.0

y: 0.0

z: -0.0824463752295

w: 0.996595502303

Take a picture LC on Camera icon /home/harman/.gazebo/pictures

```
harman@D104-45931:~$ rosservice call gazebo/get_model_state '{model_name: mobile_base}'
header:
  seq: 1
  stamp:
    secs: 156
    nsecs: 330000000
  frame_id: "
pose:
  position:
    x: 0.00239835565989      Assume 0
    y: 0.021647088024
    z: -0.00113111570117
  orientation:
    x: -0.000144075725062
    y: -0.00399553056254
    z: -0.0303578644476
    w: 0.999531097587      Assume 1
```

Position about zero. Pointing in his x direction. Sines $(\theta/2) = 0$, cosine $(0) = 1$.

```
harman@D104-45931:~$ rosservice call gazebo/get_model_state '{model_name: mobile_base}'
header:
  seq: 2
  stamp:
    secs: 573
    nsecs: 390000000
  frame_id: "
pose:
  position:
    x: 0.139027771669
    y: 0.968094927903
    z: -0.00113075972974
  orientation:
    x: 0.000823182519719
    y: 0.00391288420493
    z: 0.211738389596
    w: -0.977318201037
twist:
  linear:
    x: 0.000149913218815
    y: 0.000284748365889
    z: 7.30171681197e-06
  angular:
    x: 0.00100320081428
    y: -0.000483476400071
    z: -0.000556257309208
```

```
success: True
status_message: GetModelState: got properties
```

Page 85 Move TurtleBot with Command Line command

```
harman@D104-45931:~$ rostopic list | grep mobile_base
/mobile_base/commands/reset_odometry      Push Tab  Need std_msgs/Empty
/mobile_base/commands/velocity
      Push Tab  Need      geometry_msgs/Twist
      "linear:
        x: 0.0
        y: 0.0
        z: 0.0
      angular:
        x: 0.0
        y: 0.0
        z: 0.0"
```

```
$ rostopic pub /mobile_base/commands/reset_odometry std_msgs/Empty
```

Page 89

```
harman@D104-45931:~$ roslaunch turtlebot_gazebo turtlebot_world.launch
```

```
harman@D104-45931:~$ roslaunch turtlebot_teleop keyboard_teleop.launch
```

Page 96

```
harman@D104-45931:~$ roslaunch turtlebot_dashboard turtlebot_dashboard.launch
```

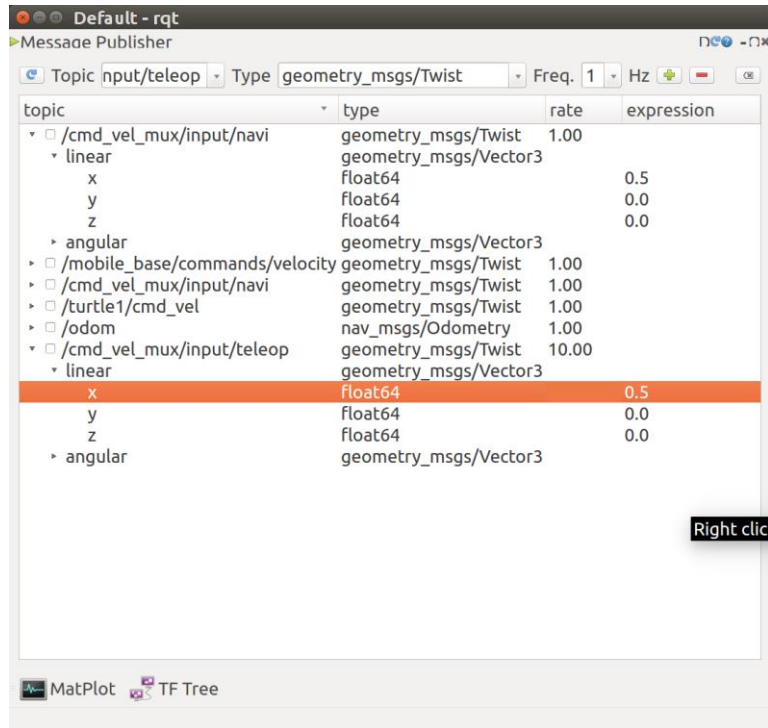
Page 103 Python

```
harman@D104-45931:~/Desktop$ python ControlTurtleBot_circle.py
[INFO] [1517886515.220983, 0.000000]: Press CTRL+c to stop TurtleBot
```

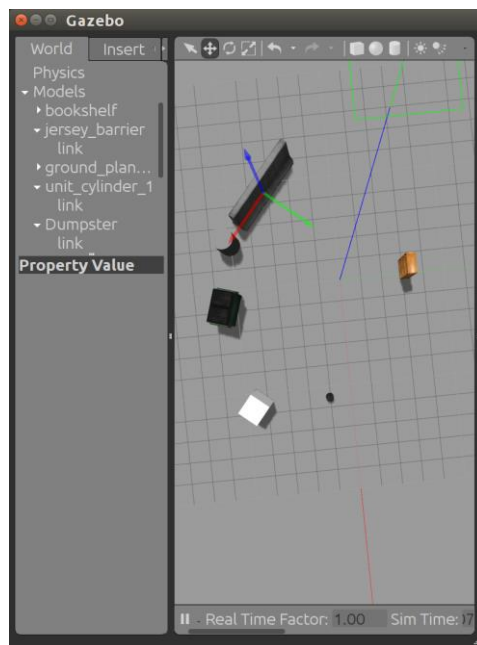
Page 105

```
harman@D104-45931:~$ rqt_graph
```

\$ roslaunch turtlebot_gazebo turtlebot_world.launch
harman@D104-45931:~\$ **rqt**



set velocity x = 0.5



Page 110

```
harman@D104-45931:~$ rostopic type /odom
nav_msgs/Odometry
```

```
harman@D104-45931:~$ rostopic show nav_msgs/Odometry
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
string child_frame_id
geometry_msgs/PoseWithCovariance pose
  geometry_msgs/Pose pose
    geometry_msgs/Point position
      float64 x
      float64 y
      float64 z
    geometry_msgs/Quaternion orientation
      float64 x
      float64 y
      float64 z
      float64 w
  float64[36] covariance
geometry_msgs/TwistWithCovariance twist
  geometry_msgs/Twist twist
    geometry_msgs/Vector3 linear
      float64 x
      float64 y
      float64 z
    geometry_msgs/Vector3 angular
      float64 x
      float64 y
      float64 z
  float64[36] covariance
```

Let's Reset TurtleBot Odometry

Page 113

```
$ rostopic type /mobile_base/commands/reset_odometry std_msgs/Empty
```

Then publish the message to reset the odometry values by typing:

```
$ rostopic pub /mobile_base/commands/reset_odometry std_msgs/Empty
```

P 113 Now echo /odom/pose/pose

position:
x: -5.43375952072e-07
y: 7.32230278349e-09
z: 0.0
orientation:
x: 0.0
y: 0.0
z: -0.0250827962935

harman@D104-45931:~\$ **rostopic echo /mobile_base/sensors/imu_data**

header:
seq: 128375
stamp:
secs: 1284
nsecs: 80000000
frame_id: base_link
orientation:
x: -0.000368510667717
y: -0.00398067523257
z: -0.0982599679862
w: 0.995152750645
orientation_covariance: [1000000.0, 0.0, 0.0, 0.0, 1000000.0, 0.0, 0.0, 0.0, 0.05]
angular_velocity:
x: -0.0003517472502
y: 0.000176141339609
z: -0.000335389775366
angular_velocity_covariance: [1000000.0, 0.0, 0.0, 0.0, 1000000.0, 0.0, 0.0, 0.0, 0.05]
linear_acceleration:
x: 0.00145663949468
y: -0.078498801872
z: 9.91170169504
linear_acceleration_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]

Note the numerical errors?

```
harman@D104-45931:~$ roslaunch turtlebot_rviz_launchers view_robot.launch
```

In rviz, it is necessary to choose several options to show the TurtleBot's odometry arrows on the screen. As shown in the following screenshot, we choose the following:

1. Under Global Options on the left side panel for Fixed Frame, **change base_link or base_footprint to odom** .
2. **Click on Add**, and select the **By topic** tab shown.
3. Choose Odometry and click on OK.
4. On the left side panel, **click on the small arrow** to the left of Odometry to show the various options. The topic is odom and the screen will keep 100 arrows that point to the direction of the simulated TurtleBot as it moves:

In RVIZ, **select fixed frame odom** to see arrows turn – otherwise the world turns!

```
harman@D104-45931:~$ rostopic pub -r 10 /cmd_vel_mux/input/teleop geometry_msgs/Twist "linear:
x: 0.1
y: 0.0
z: 0.0
angular:
x: 0.0
y: 0.0
z: -0.5"
```

