Ros Navigation   10_6_2020 **Practice Ch4 and Navigation**

TEST THE PACKAGES ------------

ROS NavigationGazebo   **https://risc.readthedocs.io/1-ros-navigation.html**

In this tutorial, you will work with a simulated robot called TurtleBot in the Gazebo simulator and Rviz.

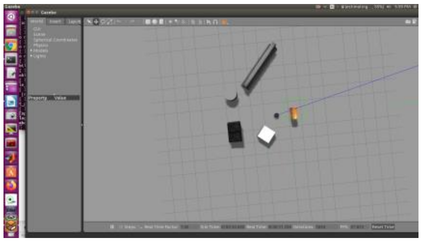
ROS_PACKAGE_PATH=/home/harman/baxter_ws/src:/home/harman/catkin_ws/src:/opt/ros/kinetic/share

I. We add the Laser Scanner with RVIZ.

GAZEBO TELEOP RVIZ Laser Scan MOVE TB (3 TERMINALS)

```
I_1 $ roslaunch turtlebot_gazebo turtlebot_world.launch
```

(Physics and Obstacles – Edit>Reset Model Poses if TB not at 0.0)



```
PARAMETERS
 * /bumper2pointcloud/pointcloud_radius: 0.24
 * /cmd_vel_mux/yaml_cfg_file: /opt/ros/kinetic/...
 * /depthimage_to_laserscan/output_frame_id: /camera_depth_frame
 * /depthimage_to_laserscan/range_min: 0.45
 * /depthimage_to_laserscan/scan_height: 10
 * /robot_description: <?xml version="1....
 * /robot_state_publisher/publish_frequency: 30.0
 * /rosdistro: kinetic
 * /rosversion: 1.12.16
 * /use_sim_time: True

NODES
  /
    bumper2pointcloud (nodelet/nodelet)
    cmd_vel_mux (nodelet/nodelet)
    depthimage_to_laserscan (nodelet/nodelet)
    gazebo (gazebo_ros/gzserver)
    gazebo_gui (gazebo_ros/gzclient)
    laserscan_nodelet_manager (nodelet/nodelet)
    mobile_base_nodelet_manager (nodelet/nodelet)
    robot_state_publisher (robot_state_publisher/robot_state_publisher)
    spawn_turtlebot_model (gazebo_ros/spawn_model)
      .
[spawn_turtlebot_model-4] process has finished cleanly
log file: /home/harman/.ros/log/e1d8666c-0756-11eb-a5a4-
9cb6d00f6f89/spawn_turtlebot_model-4*.log
```

If stuck Processes running **$ top**

Keyboard to drive TB

```
I_2 $ roslaunch turtlebot_teleop keyboard_teleop.launch    (Move TB)
```

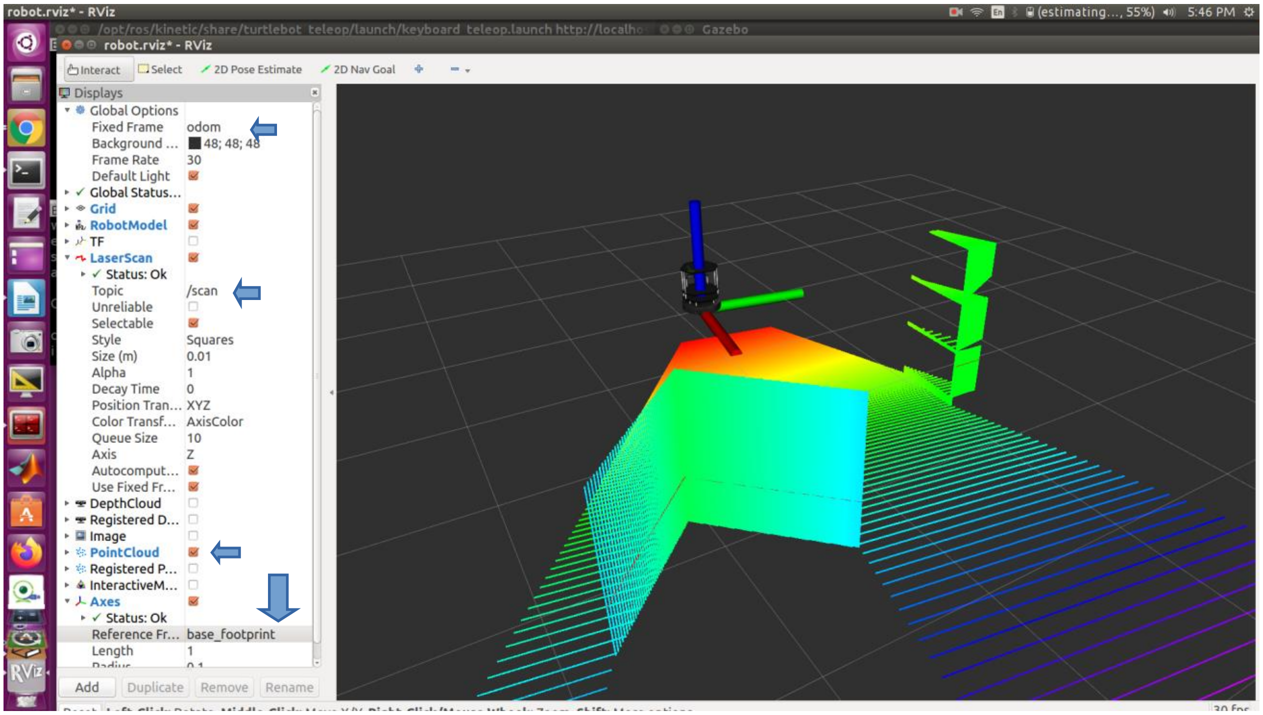
S

RVIZ

```
I_3 $ roslaunch turtlebot_rviz_launchers view_robot.launch   Pg 171
      a. GlobalOptions> Change Fixed Frame >  ODOM
            Laser Scan Topic (Open Dropdown = /scan)
      b. Display ADD Robot Model (If not selected),
            Axes (Change reference frame = base_footprint),
            Point Cloud > Camera/depth/points.
```

I.4 Move the TurtleBot with keys (I , l j and q/z w/x e/c)

To Save Rviz -   File save config as <name>   (Home .rviz)



```
Displays
▾ Global Options
    Fixed Frame        odom
    Background ...     ■ 48; 48; 48
    Frame Rate         30
    Default Light      ☑
▸ ✓ Global Status...
▸ Grid                 ☑
▸ RobotModel           ☑
▸ TF                   □
▾ LaserScan            ☑
  ▸ ✓ Status: Ok
    Topic              /scan
    Unreliable         □
    Selectable         ☑
    Style              Squares
    Size (m)           0.01
    Alpha              1
    Decay Time         0
    Position Tran...   XYZ
    Color Transf...    AxisColor
    Queue Size         10
    Axis               Z
    Autocomput...      ☑
    Use Fixed Fr...    ☑
▸ DepthCloud           □
▸ Registered D...      □
▸ Image                □
▸ PointCloud           ☑
▸ Registered P...      □
▸ InteractiveM...      □
▾ Axes                 ☑
  ▸ ✓ Status: Ok
    Reference Fr...    base_footprint
    Length             1
```

WHEN DONE – CLOSE EVERTHING

S

MAPPING ---------------------Start Afresh - New Terminals

Have Gazebo running

```
II_1 $ roslaunch turtlebot_gazebo turtlebot_world.launch   Pg 170
```

Check Model pose = 0 (about); `harman@D104-45931:~$ rostopic echo /odom`

Run Mapping DEMO

```
II_2 $ roslaunch turtlebot_gazebo gmapping_demo.launch      Pg 171
```

```
SUMMARY
========

PARAMETERS
 * /rosdistro: kinetic
 * /rosversion: 1.12.16
 * /slam_gmapping/angularUpdate: 0.436
 * /slam_gmapping/astep: 0.05
 * /slam_gmapping/base_frame: base_footprint
 * /slam_gmapping/delta: 0.05
 * /slam_gmapping/iterations: 5
 * /slam_gmapping/kernelSize: 1
 * /slam_gmapping/lasamplerange: 0.005
 * /slam_gmapping/lasamplestep: 0.005
 * /slam_gmapping/linearUpdate: 0.5
 * /slam_gmapping/llsamplerange: 0.01
 * /slam_gmapping/llsamplestep: 0.01
 * /slam_gmapping/lsigma: 0.075
 * /slam_gmapping/lskip: 0
 * /slam_gmapping/lstep: 0.05
 * /slam_gmapping/map_update_interval: 5.0
 * /slam_gmapping/maxRange: 8.0
 * /slam_gmapping/maxUrange: 6.0
 * /slam_gmapping/minimumScore: 200
 * /slam_gmapping/odom_frame: odom
 * /slam_gmapping/ogain: 3.0
 * /slam_gmapping/particles: 80
 * /slam_gmapping/resampleThreshold: 0.5
 * /slam_gmapping/sigma: 0.05
 * /slam_gmapping/srr: 0.01
 * /slam_gmapping/srt: 0.02
 * /slam_gmapping/str: 0.01
 * /slam_gmapping/stt: 0.02
 * /slam_gmapping/temporalUpdate: -1.0
 * /slam_gmapping/xmax: 1.0
 * /slam_gmapping/xmin: -1.0
 * /slam_gmapping/ymax: 1.0
 * /slam_gmapping/ymin: -1.0
```

S

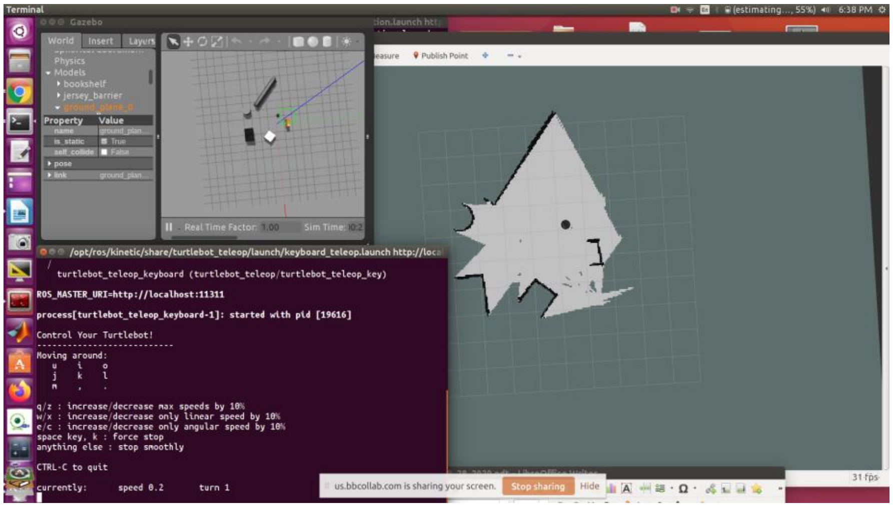
```
NODES
        /
          slam_gmapping (gmapping/slam_gmapping)

      ROS_MASTER_URI=http://localhost:11311

      process[slam_gmapping-1]: started with pid [21923]
      [ INFO] [1601936405.159782957, 58.180000000]: Laser is mounted
      upwards.
       -maxUrange 6 -maxUrange 8 -sigma     0.05 -kernelSize 1 -lstep 0.05 -
      lobsGain 3 -astep 0.05
       -srr 0.01 -srt 0.02 -str 0.01 -stt 0.02
       -linearUpdate 0.5 -angularUpdate 0.436 -resampleThreshold 0.5
       -xmin -1 -xmax 1 -ymin -1 -ymax 1 -delta 0.05 -particles 80
      [ INFO] [1601936405.161233628, 58.180000000]: Initialization complete
      update frame 0
      update ld=0 ad=0
      Laser Pose= -0.0857422 0.0497355 -0.0245193
      m_count 0
      Registering First Scan
```

```
II_3 $ roslaunch turtlebot_rviz_launchers view_navigation.launch
```

Grid, Robot Model, Laser Scan topic name /scan;
Bumper hit, Map topic name /map, Global&Local Map, AMCL particles,
Full Plan -   Page 176 explains terms

```
II_4 $ roslaunch turtlebot_teleop keyboard_teleop.launch
```

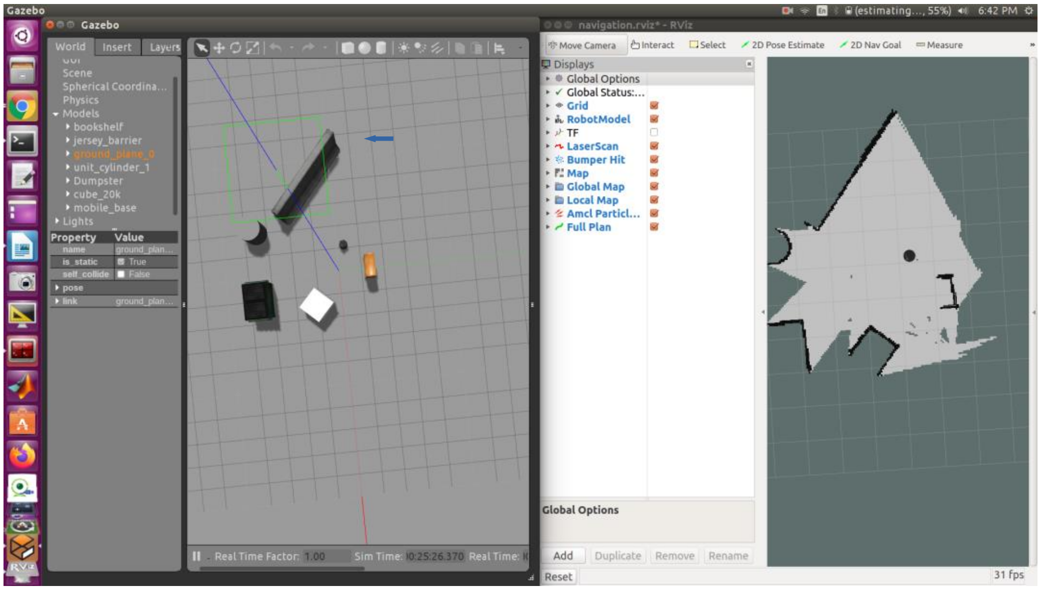
Start driving the robot using keyboard keys
and observe how the map is updated in Rviz
Move TB and rotate TB to draw map.
Line up RVIZ and Gazebo images.



/opt/ros/kinetic/share/turtlebot_teleop/launch/keyboard_teleop.launch http://loca

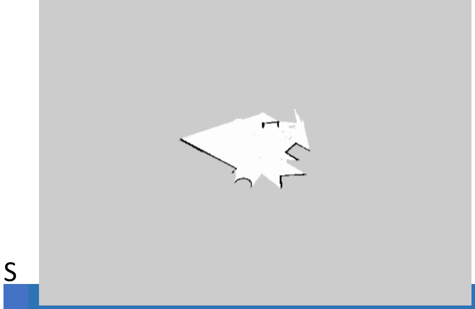
```
/
  turtlebot_teleop_keyboard (turtlebot_teleop/turtlebot_teleop_key)

ROS_MASTER_URI=http://localhost:11311

process[turtlebot_teleop_keyboard-1]: started with pid [19616]

Control Your Turtlebot!
---------------------------
Moving around:
   u    i    o
   j    k    l
   m    ,    .

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%
space key, k : force stop
anything else : stop smoothly

CTRL-C to quit

currently:      speed 0.2       turn 1
```

S



```
SAVE MAP   - Saved in Home or said directory      Page 173

II_5 $ rosrun map_server map_saver -f <your map name>

eg.  $ rosrun map_server map_saver -f
/home/harman/Desktop/our_map_10_5_2020        (Example)

[ INFO] [1601937424.311886954]: Waiting for the map
[ INFO] [1601937424.519736249]: Received a 480 X 512 map @ 0.050 m/pix
[ INFO] [1601937424.519782437]: Writing map occupancy data to
/home/harman/Desktop/our_map_10_5_2020.pgm
[ INFO] [1601937424.526199115, 1077.150000000]: Writing map occupancy
data to /home/harman/Desktop/our_map_10_5_2020.yaml
[ INFO] [1601937424.526316858, 1077.150000000]: Done

YAML file which contains descriptions about your map setup
Grayscale image that represents your occupancy grid map,
which actually can be edited by an image editor

YAML
```

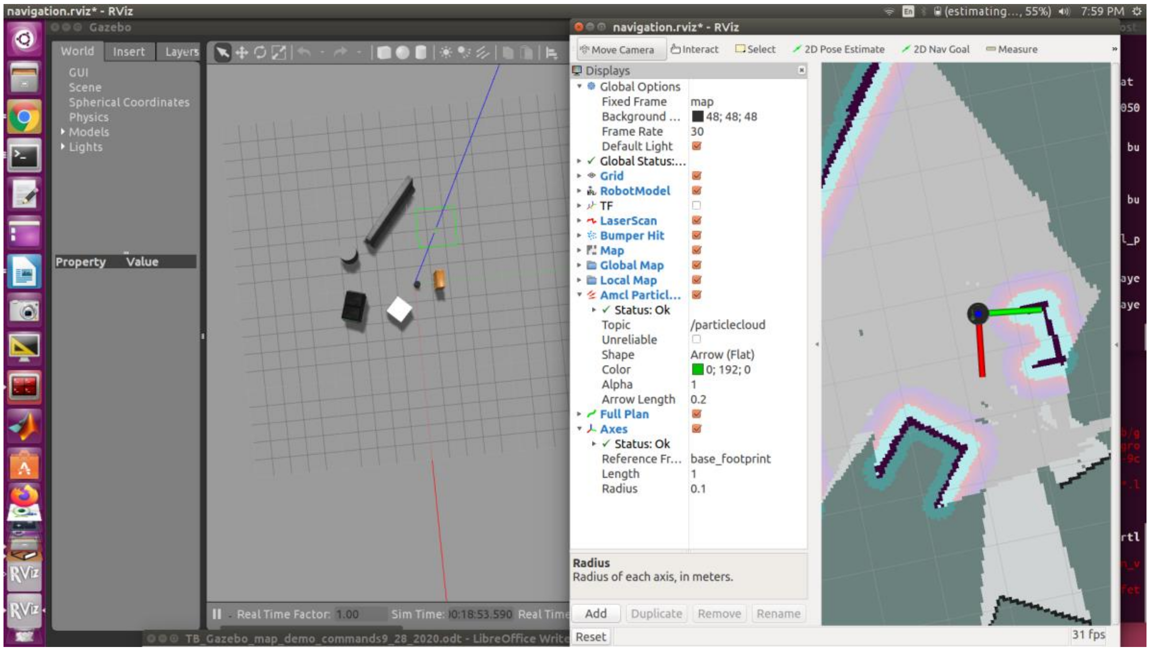
```
image: /home/harman/Desktop/mapdemo9_28.pgm
resolution: 0.050000
origin: [-13.800000, -13.800000, 0.000000]
negate: 0
occupied_thresh: 0.65
free_thresh: 0.196
```



S

III. **NOW Localization and Navigation**

```
Localization
III_1 $ roslaunch turtlebot_gazebo turtlebot_world.launch

III_2 $ roslaunch turtlebot_gazebo amcl_demo.launch
map_file:=/home/harman/Desktop/mapdemo9_28.yaml
     Page 177        odom received!

III_3 $ roslaunch turtlebot_rviz_launchers view_navigation.launch    P177
    Align Gaxebo grid with RVIZ Map  x-y
```



Global Options	
Fixed Frame	map
Background ...	■ 48; 48; 48
Default Light	☑
Frame Rate	30
✓ Global Status:...	
Grid	☑
RobotModel	☑
TF	☐
LaserScan	☑
Bumper Hit	☑
Map	☑
Global Map	☑
Local Map	☑
Amcl Particl...	☑
✓ Status: Ok	
Topic	/particlecloud
Unreliable	☐
Shape	Arrow (Flat)
Color	🟩 0; 192; 0
Alpha	1
Arrow Length	0.2
Full Plan	☑
Axes	☑
✓ Status: Ok	
Reference Fr...	base_footprint
Length	1
Radius	0.1

Radius
Radius of each axis, in meters.

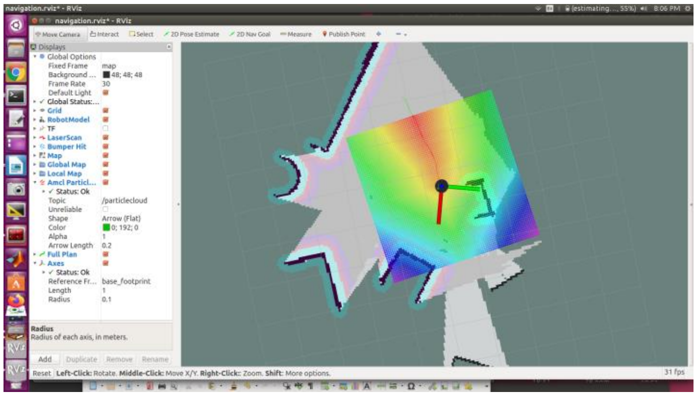
```
2D pose estimate RVIZ Pg 178     Position and Orientation of TB
```

S

2D Nav Goal RVIZ   Pg 179   Select new Position and Orientation

Align Gazebo and RVIZ grid and ADD Axes base_footprint to RVIZ left menu.
2D Pose estimate - Locate TB and orientation Pg 178

Then 2D Nav goal to move TB to new position and orientation.



```
Global Options
  Fixed Frame       map
  Background ...    48; 48; 48
  Frame Rate        30
  Default Light     ☑
✓ Global Status:...
Grid                ☑
RobotModel          ☑
TF                  □
LaserScan           ☑
Bumper Hit          ☑
Map                 ☑
Global Map          ☑
Local Map           ☑
Amcl Particl...     ☑
  ✓ Status: Ok
  Topic             /particlecloud
  Unreliable        □
  Shape             Arrow (Flat)
  Color             🟩 0; 192; 0
  Alpha             1
  Arrow Length      0.2
Full Plan           ☑
Axes                ☑
  ✓ Status: Ok
  Reference Fr...   base_footprint
  Length            1
  Radius            0.1
```

Radius
Radius of each axis, in meters.

Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click: Zoom. Shift: More options.

See page 180 and 181 and note "Green" arrow for destination.

```
harman@D104-45931:~$ rostopic echo /odom/pose/pose -n1
```

IV. MOVING TURTLEBOT -WITH VELOCITY
See Page 114 – Drive TB in gazebo

```
$ roslaunch turtlebot_gazebo turtlebot_world.launch
$ rostopic pub -r 10 mobile_base/commands/velocity \geometry_msgs/Twist
'{linear: {x: .2}}'
```

V MOVING TURTLEBOT – POSITION WITH PYTHON

Pg 185

```
$ roslaunch turtlebot_gazebo turtlebot_world.launch

$ roslaunch turtlebot_gazebo amcl_demo.launch
map_file:=/home/harman/Desktop/mapdemo9_28.yaml    (OR our_map2.yaml)

   odom received!

$ roslaunch turtlebot_rviz_launchers view_navigation.launch
```

Align Gazebo and RVIZ screens.

Pg 188
Run rostopic and then click on point in RVIZ to see values.

```
harman@D104-45931:~$ rostopic echo /clicked_point
WARNING: no messages received and simulated time is active.
Is /clock being published?       (After clicking - OK)
header:
  seq: 2
  stamp:
    secs: 330
    nsecs: 550000000
  frame_id: "map"
```

S

```
point:
  x: -2.45488238335
  y: 1.48951196671
  z: -0.00534057617188
---
header:
  seq: 3
  stamp:
    secs: 350
    nsecs: 900000000
  frame_id: "map"
point:
  x: -2.15811944008
  y: 2.65072822571
  z: -0.00143432617188
```

PYTHON SCRIPT TO MOVE TB TO GOAL POINTS – CHOOSE POINTS IN SCRIPT

```
harman@D104-45931:~$ cd Desktop/
harman@D104-45931:~/Desktop$ ls -la | grep MoveTB
-rw-rw-r--  1 harman harman    1624 Sep 25 15:37 MoveTBtoGoalPoints.py
Pg 189-190

harman@D104-45931:~/Desktop$ chmod +x MoveTBtoGoalPoints.py
harman@D104-45931:~/Desktop$ ls -la | grep MoveTB
-rwxrwxr-x  1 harman harman    1624 Sep 25 15:37 MoveTBtoGoalPoints.py
harman@D104-45931:~/Desktop$
```

Set the goal point in the program.

```
$ python MoveTBtoGoalPoints.py
```

```
#!/usr/bin/env python    # MoveTBtoGoalPoints

import rospy
import actionlib       # Use the actionlib package for client and server

from move_base_msgs.msg import MoveBaseAction, MoveBaseGoal

# Define Goal Points and orientations for TurtleBot in a list
      (X,Y,theta)
GoalPoints = [ [(1.0, 0.0, 0.0), (0.0, 0.0, 0.0, 1.0)] ,
[(0.0, 0.0, 0.0), (0.0, 0.0, 0.707, 0.707)]]

# The function assign_goal initializes the goal_pose variable as a
MoveBaseGoal action type.
#
```

```
def assign_goal(pose):

    goal_pose = MoveBaseGoal()
    goal_pose.target_pose.header.frame_id = 'map'
    goal_pose.target_pose.pose.position.x = pose[0][0]
    goal_pose.target_pose.pose.position.y = pose[0][1]
    goal_pose.target_pose.pose.position.z = pose[0][2]
    goal_pose.target_pose.pose.orientation.x = pose[1][0]
    goal_pose.target_pose.pose.orientation.y = pose[1][1]
    goal_pose.target_pose.pose.orientation.z = pose[1][2]
    goal_pose.target_pose.pose.orientation.w = pose[1][3]

    return goal_pose

if __name__ == '__main__':
    rospy.init_node('MoveTBtoGoalPoints')
# Create a SimpleActionClient of a move_base action type and wait for
server.
    client = actionlib.SimpleActionClient('move_base', MoveBaseAction)
    client.wait_for_server()

#
    for TBpose in GoalPoints:
        TBgoal = assign_goal(TBpose)   # For each goal point assign pose
        client.send_goal(TBgoal)
        success = client.wait_for_result()
#        client.wait_for_result()

    if success:
# if (client.get_state() == GoalStatus.SUCCEEDED):
        rospy.loginfo("success")
    else:
        rospy.loginfo("failed")
```