# Appendix B

# User Guide

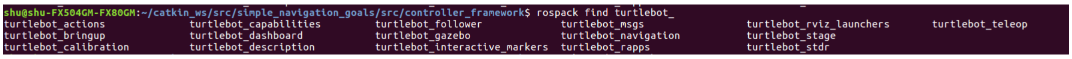
1. Please ensure you are using a linux machine with Ubuntu 16.04 installed and sudo access on this machine.

2. Please install ROS Kinetic from these installation instructions:
   `http://wiki.ros.org/kinetic/Installation/Ubuntu`

3. Please ensure that the catkin workspace is properly configured.

4. Please ensure that RViz installed by typing
   "rospack find rviz" in the terminal. If it is not please follow the instructions on `http://wiki.ros.org/rviz/UserGuide` be advised to replace the version of ROS with kinetic.

5. Please ensure that the Turtlebot packages are installed by typing
   "rospack find turtlebot_"
   and then subsequently hitting the tab button twice. The terminal should display the following if the Turtlebot packages are installed.



```
shu@shu-FX504GM-FX80GM:~/catkin_ws/src/simple_navigation_goals/src/controller_framework$ rospack find turtlebot_
turtlebot_actions           turtlebot_capabilities      turtlebot_follower          turtlebot_msgs              turtlebot_rviz_launchers    turtlebot_teleop
turtlebot_bringup           turtlebot_dashboard         turtlebot_gazebo            turtlebot_navigation        turtlebot_stage
turtlebot_calibration       turtlebot_description       turtlebot_interactive_markers turtlebot_rapps           turtlebot_stdr
```

Figure B.1: The terminal should display the following if the Turtlebot packages are installed.

   If no Turtlebot packages are not installed installation instructions are available here:
   `http://wiki.ros.org/action/show/turtlebot/Tutorials/indigo/Turtlebot%20Installation?action=show&redirect=turtlebot%2FTutorials%2Findigo%2FInstallation`
   please be aware that the only available Turtlebot 2 instructions are in Indigo, and the

actual installation line should be the following:
"sudo apt-get install ros-kinetic-turtlebot ros-kinetic-turtlebot-apps ros-kinetic-turtlebot-interactions ros-kinetic-turtlebot-simulator ros-kinetic-kobuki-ftdi ros-kinetic-ar-track-alvar-msgs"

6. Download the source code from here:
   `https://github.kcl.ac.uk/K1630593/navigation-and-some-kind-of-gui`

7. Unzip the downloaded source code into the home directory.

8. From the source code, move move multi_map_server and simple_navigation_goals into the src folder of the catkin workspace.

9. Move the maps folder from the source code the home directory.

10. Please edit the file locations in the .yaml and .world files within the Maps folders to correctly reflect the location of the files

11. Please edit lines 9, 10, and 11 teleport_node.py file in
    directoryToCatkinWorkspace/src/simple_navigation_goals/src/controller_framework/
    to correctly reflect the location of the specified maps within the map folder.

12. Replace the contents of
    /opt/ros/kinetic/share/turtlebot_navigation/launch/amcl_demo.launch
    with the contents of the amcl_demo.launch file within the source code. Be advised you will require sudo access to do this.

13. Open a terminal and navigate to the catkin workspace. Type in the terminal
    "catkin_make install"

14. Type in the terminal
    "sudo apt-get install espeak"

15. The instructions now differ depending on simulation or hardware.

## B.1 Hardware

1. Please ensure the Turtlebot and Netbook are at full power

2. Connect any additional electronic accessories (a usb port or a mouse). Please be advised that if access to one of the USB ports of the Netbook is required, unplugging the right

socket if the Turtlebot is not currently completing a navigational goal is nonfatal, as the Turtlebot will continue searching the disconnected camera until it is reconnected.

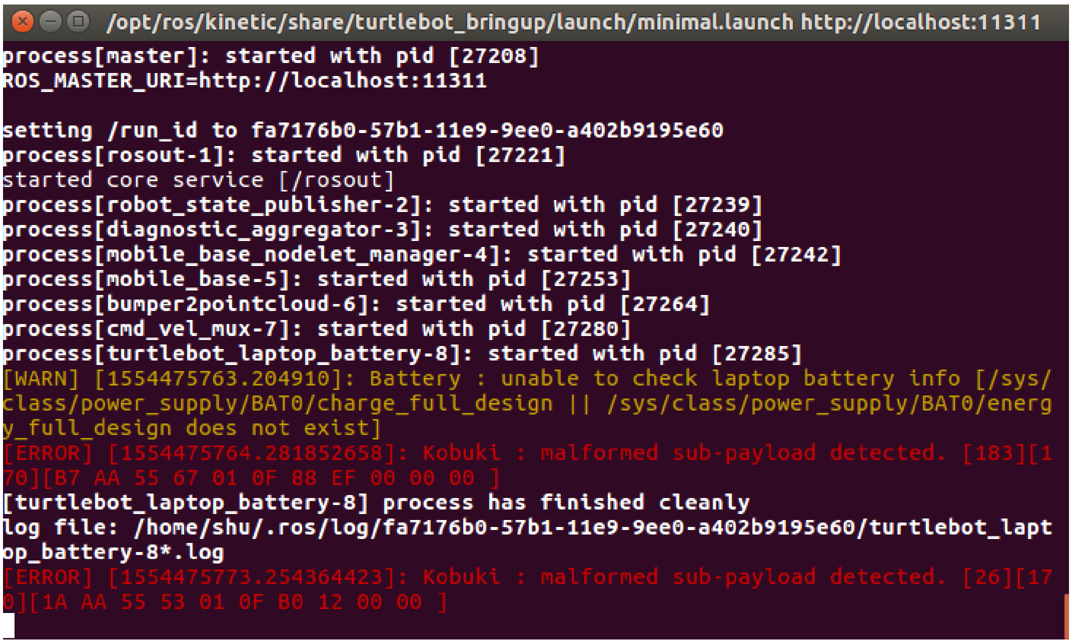
3. Open a terminal and enter the following command:

   “roslaunch turtlebot_bringup minimal.launch”

4. Wait until the terminal displays the

   [turtlebot_laptop_battery-8] process has finished cleanly

   message has appeared.



/opt/ros/kinetic/share/turtlebot_bringup/launch/minimal.launch http://localhost:11311

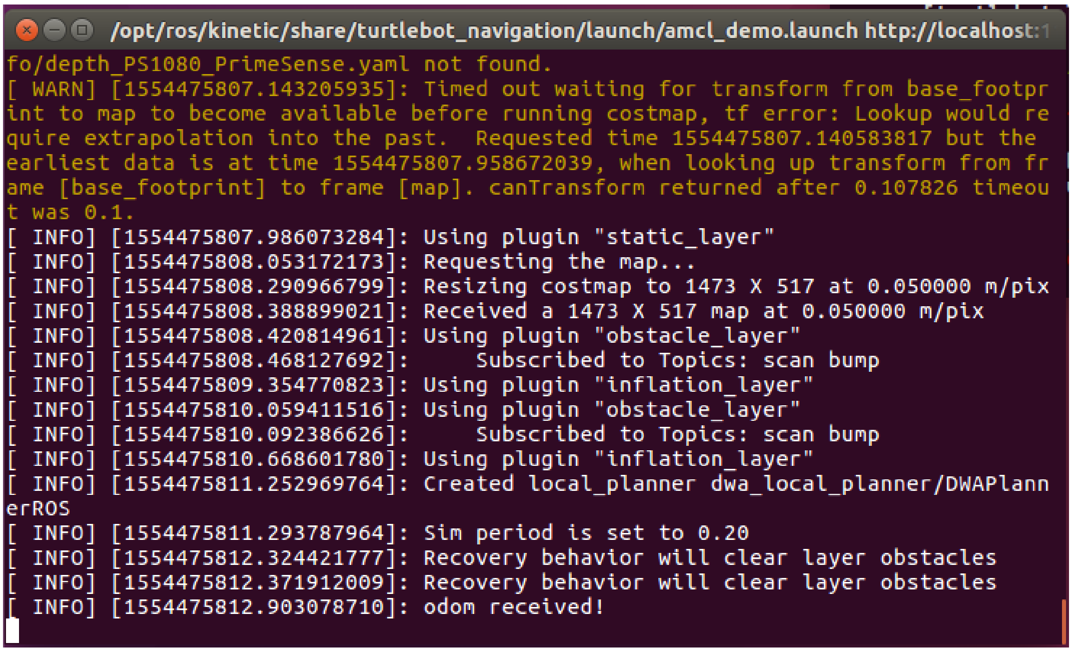
```
process[master]: started with pid [27208]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to fa7176b0-57b1-11e9-9ee0-a402b9195e60
process[rosout-1]: started with pid [27221]
started core service [/rosout]
process[robot_state_publisher-2]: started with pid [27239]
process[diagnostic_aggregator-3]: started with pid [27240]
process[mobile_base_nodelet_manager-4]: started with pid [27242]
process[mobile_base-5]: started with pid [27253]
process[bumper2pointcloud-6]: started with pid [27264]
process[cmd_vel_mux-7]: started with pid [27280]
process[turtlebot_laptop_battery-8]: started with pid [27285]
[WARN] [1554475763.204910]: Battery : unable to check laptop battery info [/sys/
class/power_supply/BAT0/charge_full_design || /sys/class/power_supply/BAT0/energ
y_full_design does not exist]
[ERROR] [1554475764.281852658]: Kobuki : malformed sub-payload detected. [183][1
70][B7 AA 55 67 01 0F 88 EF 00 00 00 ]
[turtlebot_laptop_battery-8] process has finished cleanly
log file: /home/shu/.ros/log/fa7176b0-57b1-11e9-9ee0-a402b9195e60/turtlebot_lapt
op_battery-8*.log
[ERROR] [1554475773.254364423]: Kobuki : malformed sub-payload detected. [26][17
0][1A AA 55 53 01 0F B0 12 00 00 ]
```

Figure B.2: The malform sub-payload detected error is a kobuki base error and can be ignored.

5. Open a new terminal and enter the following command:

   “roslaunch turtlebot_navigation amcl_demo.launch map_file:=directoryToMaps/7thedited.yaml initial_pose_x:=36.5 initial_pose_y:=17 initial_pose_a:=0.0”

   This will initialize the Turtlebot to the area in the lobby on the 7the floor between the North and South elevators. The Turtlebot should be facing the wall between the two South elevators.


/opt/ros/kinetic/share/turtlebot_navigation/launch/amcl_demo.launch http://localhost:1

```
fo/depth_PS1080_PrimeSense.yaml not found.
[ WARN] [1554475807.143205935]: Timed out waiting for transform from base_footpr
int to map to become available before running costmap, tf error: Lookup would re
quire extrapolation into the past.  Requested time 1554475807.140583817 but the
earliest data is at time 1554475807.958672039, when looking up transform from fr
ame [base_footprint] to frame [map]. canTransform returned after 0.107826 timeou
t was 0.1.
[ INFO] [1554475807.986073284]: Using plugin "static_layer"
[ INFO] [1554475808.053172173]: Requesting the map...
[ INFO] [1554475808.290966799]: Resizing costmap to 1473 X 517 at 0.050000 m/pix
[ INFO] [1554475808.388899021]: Received a 1473 X 517 map at 0.050000 m/pix
[ INFO] [1554475808.420814961]: Using plugin "obstacle_layer"
[ INFO] [1554475808.468127692]:     Subscribed to Topics: scan bump
[ INFO] [1554475809.354770823]: Using plugin "inflation_layer"
[ INFO] [1554475810.059411516]: Using plugin "obstacle_layer"
[ INFO] [1554475810.092386626]:     Subscribed to Topics: scan bump
[ INFO] [1554475810.668601780]: Using plugin "inflation_layer"
[ INFO] [1554475811.252969764]: Created local_planner dwa_local_planner/DWAPlann
erROS
[ INFO] [1554475811.293787964]: Sim period is set to 0.20
[ INFO] [1554475812.324421777]: Recovery behavior will clear layer obstacles
[ INFO] [1554475812.371912009]: Recovery behavior will clear layer obstacles
[ INFO] [1554475812.903078710]: odom received!
```

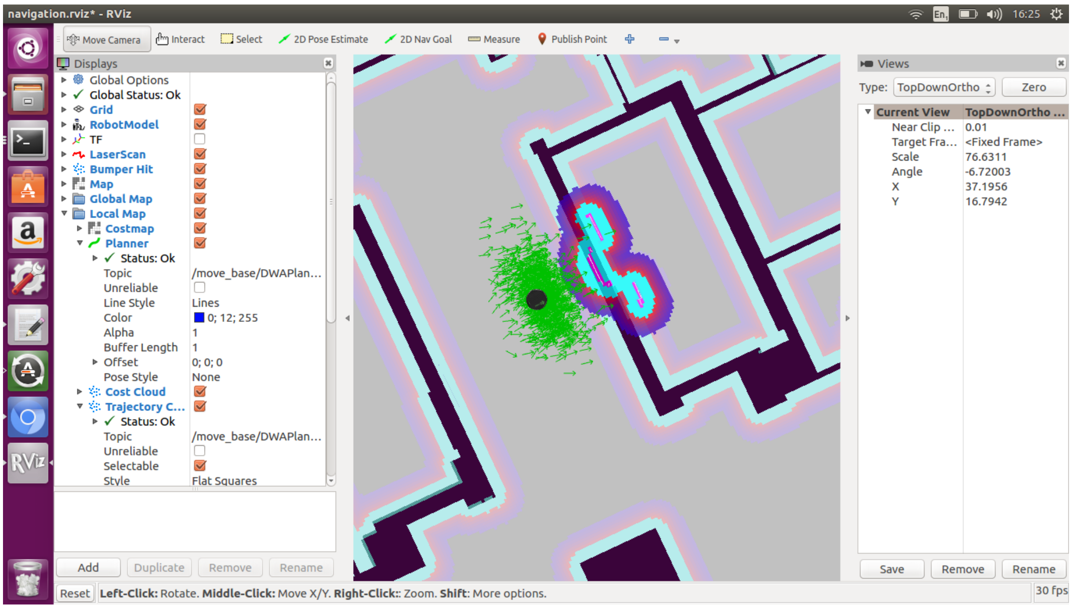
Figure B.3: Wait for the Odom Received! message before continuing.

6. Open RViz in a new terminal with the following command:

   "roslaunch turtlebot_rviz_launchers view_navigation.launch"

7. Re-adjust the Turtlebot's coordinates by picking up the Turtlebot to move it if necessary.

   Do not re-adjust the coordinates by moving the Turtlebot wheels.



| Displays | |
|---|---|
| Global Options | |
| Global Status: Ok | |
| Grid | ☑ |
| RobotModel | ☑ |
| TF | ☐ |
| LaserScan | ☑ |
| Bumper Hit | ☑ |
| Map | ☑ |
| Global Map | ☑ |
| Local Map | ☑ |
| Costmap | ☑ |
| Planner | ☑ |
| Status: Ok | |
| Topic | /move_base/DWAPlan... |
| Unreliable | ☐ |
| Line Style | Lines |
| Color | 0; 12; 255 |
| Alpha | 1 |
| Buffer Length | 1 |
| Offset | 0; 0; 0 |
| Pose Style | None |
| Cost Cloud | ☑ |
| Trajectory C... | ☑ |
| Status: Ok | |
| Topic | /move_base/DWAPlan... |
| Unreliable | ☐ |
| Selectable | ☑ |
| Style | Flat Squares |

Type: TopDownOrtho

| Current View | TopDownOrtho ... |
|---|---|
| Near Clip ... | 0.01 |
| Target Fra... | <Fixed Frame> |
| Scale | 76.6311 |
| Angle | -6.72003 |
| X | 37.1956 |
| Y | 16.7942 |

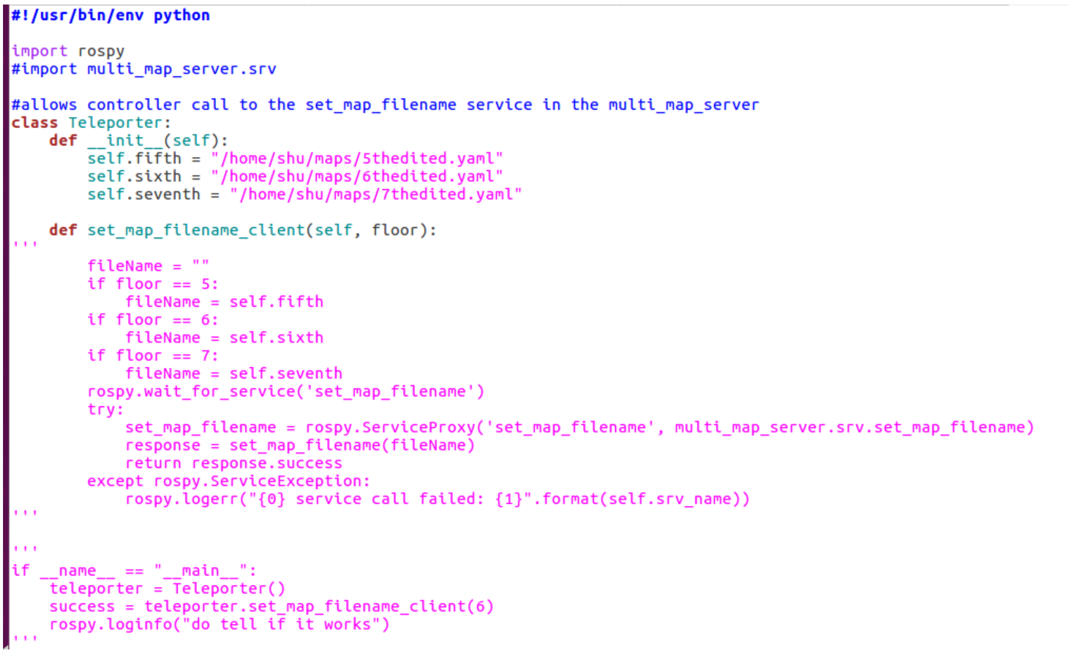
**Left-Click:** Rotate. **Middle-Click:** Move X/Y. **Right-Click::** Zoom. **Shift:** More options.

Figure B.4: The costmaps should align as follows.

8. Open another terminal and source the bash file with the following command: “source directoryToCatkin/devel/setup.bash”
9. In the same terminal, enter the following command: “rosrun simple_navigation_goals controller_framework_node.py”
10. To initialize on a different map, change the yaml file and coordinates as appropriate. Lines 29 to 30 within the controller will also need to be edited to reflect the changes.
11. After the initial localization spin, visitor escort requests may now be entered through the same terminal at the text prompts. The controller can be quit by CTRL + C.

## B.2 Simulation

1. Please be advised that the multi_map_server is not functional in simulation. As a result some changes must be made to the teleport_node file within the controller_framework folder of the simple_navigational_goals/src



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

import rospy
#import multi_map_server.srv

#allows controller call to the set_map_filename service in the multi_map_server
class Teleporter:
    def __init__(self):
        self.fifth = "/home/shu/maps/5thedited.yaml"
        self.sixth = "/home/shu/maps/6thedited.yaml"
        self.seventh = "/home/shu/maps/7thedited.yaml"

    def set_map_filename_client(self, floor):
'''

        fileName = ""
        if floor == 5:
            fileName = self.fifth
        if floor == 6:
            fileName = self.sixth
        if floor == 7:
            fileName = self.seventh
        rospy.wait_for_service('set_map_filename')
        try:
            set_map_filename = rospy.ServiceProxy('set_map_filename', multi_map_server.srv.set_map_filename)
            response = set_map_filename(fileName)
            return response.success
        except rospy.ServiceException:
            rospy.logerr("{0} service call failed: {1}".format(self.srv_name))
'''

'''
if __name__ == "__main__":
    teleporter = Teleporter()
    success = teleporter.set_map_filename_client(6)
    rospy.loginfo("do tell if it works")
'''
```

Figure B.5: This will allow the Controller to run without the ability to change maps

2. Remember to revert the teleport_node.py file if testing in Hardware.
3. Run simulation by launching

   “roslaunch turtlebot_stage turtlebot_in_stage.launch map_file:=”directoryToMaps/7thedited.yaml” world_file:=”directoryToMaps/7thedited.world” initial_pose_x:=36.5 initial_pose_y:=17 initial_pose_a:=0.0”

This will initialize the Turtlebot to the area in the lobby on the 7th floor between the North and South elevators. The Turtlebot should be facing the wall between the two South elevators.

4. Open another terminal and source the bash file with the following command: “source directoryToCatkin/devel/setup.bash”

5. In the same terminal, enter the following command: “rosrun simple_navigation_goals controller_framework_node.py”

6. After the initial localization spin, visitor escort requests may now be entered through the same terminal at the text prompts. Please be advised simulation cannot change maps.

7. To test on a different map, change the world and yaml files in the directory and the information in lines 29 and 30 of the controller.

8. The controller can be quit by CTRL + C.