ROS Installation and Turtle BOT Demo Steps for ubuntu linux...

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Need a Ubuntu Linux installed on a VM or in dual boot mode.

VM is not preferable

PS1="R:"

To know your Linux version

$ uname -r

Confirm it is running Ubuntu Ubuntu 20.04 LTS

https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/

Install ROS on Remote PC

Open the terminal with Ctrl+Alt+T and enter below commands one at a time.

$ sudo apt update

$ sudo apt upgrade

$ wget https://raw.githubusercontent.com/ROBOTIS-GIT/robotis\_tools/master/install\_ros\_noetic.sh

$ chmod 755 ./install\_ros\_noetic.sh

$ bash ./install\_ros\_noetic.sh

Now Install - Install Dependent ROS Packages

$ sudo apt-get install ros-noetic-joy ros-noetic-teleop-twist-joy \

ros-noetic-teleop-twist-keyboard ros-noetic-laser-proc \

ros-noetic-rgbd-launch ros-noetic-rosserial-arduino \

ros-noetic-rosserial-python ros-noetic-rosserial-client \

ros-noetic-rosserial-msgs ros-noetic-amcl ros-noetic-map-server \

ros-noetic-move-base ros-noetic-urdf ros-noetic-xacro \

ros-noetic-compressed-image-transport ros-noetic-rqt\* ros-noetic-rviz \

ros-noetic-gmapping ros-noetic-navigation ros-noetic-interactive-markers

Now Install TurtleBot3 Packages

Install TurtleBot3 via Debian Packages.

$ sudo apt install ros-noetic-dynamixel-sdk

$ sudo apt install ros-noetic-turtlebot3-msgs

$ sudo apt install ros-noetic-turtlebot3

$ env | grep ROS

$ source ~/.bashrc

$ env | grep ROS

Should Contain all the ROS environment variables

Do this every time you open a new terminal

$ source ~/.bashrc

The below should be there by default

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Connect PC to a WiFi device and find the assigned IP address with the command below.

$ ifconfig

Open the file and update the ROS IP settings with the command below.

$ nano ~/.bashrc

Press Ctrl+END or Alt+/ to move the cursor to the end of line.

Modify the address of localhost in the ROS\_MASTER\_URI and ROS\_HOSTNAME with the IP address acquired from the above terminal window.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Step 2:

https://emanual.robotis.com/docs/en/platform/turtlebot3/simulation/

This Gazebo Simulation uses ROS Gazebo package, therefore, proper Gazebo version for ROS1 Noetic has to be installed before running this instruction.

Install Simulation Package:

The TurtleBot3 Simulation Package requires turtlebot3 and turtlebot3\_msgs packages as prerequisite.

Without these prerequisite packages, the Simulation cannot be launched.

Please follow the PC Setup instructions if you did not install required packages and dependent packages.

$ cd ~/catkin\_ws/src/

$ git clone -b noetic-devel https://github.com/ROBOTIS-GIT/turtlebot3\_simulations.git

$ cd ~/catkin\_ws && catkin\_make

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Step 3:

Launch the Simulation World:

1) Empty World

$ export TURTLEBOT3\_MODEL=burger

$ roslaunch turtlebot3\_gazebo turtlebot3\_empty\_world.launch

2) TurtleBot3 World

$ export TURTLEBOT3\_MODEL=waffle

$ roslaunch turtlebot3\_gazebo turtlebot3\_world.launch

3) TurtleBot3 House

$ export TURTLEBOT3\_MODEL=waffle\_pi

$ roslaunch turtlebot3\_gazebo turtlebot3\_house.launch

Operate TurtleBot3

Open another terminal

Do this every time you open a new terminal

$ source ~/.bashrc

In order to teleoperate the TurtleBot3 with the keyboard, launch the teleoperation node with below command in a new terminal window.

$ export TURTLEBOT3\_MODEL=burger

$ roslaunch turtlebot3\_teleop turtlebot3\_teleop\_key.launch

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DEMO 2:

ROS Demo: Hello World Example - Start the example by using following command.

The first step in starting any node in ROS is roscore.

$ roscore

Start the talker node by using the following command in another terminal.

$ rosrun roscpp\_tutorials talker

Now you see the messages printing on the terminal screen.

If you list the topic by using the following command, you see a new topic called

/chatter

$ rostopic list

Output: /chatter

Now start the listener node by using the following command.

$ rosrun roscpp\_tutorials listener

If you want to run two of the nodes together, use the roslaunch command.

$ roslaunch roscpp\_tutorials talker\_listener.launch

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DEMO 3:

ROS Demo: turtlesim

The turtlesim application is already installed on ROS.

You can start this application by using the following commands.

Starting roscore

$ roscore

Starting Turtlesim application

$ rosrun turtlesim turtlesim\_node

Now you can open a new terminal and list the topics by publishing the turtlesim node.

$ rostopic list

Figure lists the services created by the turtlesim node.

You can list the services by using the following command.

$ rosservice list

List the ROS parameters by using the following command

$ rosparam list

Moving the Turtle: - DEMO 4

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

If you want to move the turtle, start another ROS node by using the following command.

This command has to start in another terminal.

$ rosrun turtlesim turtle\_teleop\_key

You can control the robot using your keyboard’s arrow keys.

When you press an arrow key, it publishes velocity to /turtle1/cmd\_vel, which makes the turtle move

Moving the Turtle in a Square: - DEMO 5

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This section shows how to move the turtle along a square path.

Close all the running nodes by pressing Ctrl+C, and start a new turtlesim session using the following command see Figure.

Starting roscore

$ roscore

Starting turtlesim node

$ rosrun turtlesim turtlesim\_node

Starting the node for drawing square

$ rosrun turtlesim draw\_square

If we want to clear the turtlesim, we can call a service called /reset.

$ rosservice call /reset

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DEMO 6 - SLAM

1) Open a new terminal from Remote PC with Ctrl + Alt + T and launch the SLAM node.

The Gmapping is used as a default SLAM method. Please use the proper keyword among burger, waffle, waffle\_pi for the TURTLEBOT3\_MODEL parameter.

$ export TURTLEBOT3\_MODEL=burger

$ roslaunch turtlebot3\_slam turtlebot3\_slam.launch

2) Run Teleoperation Node

Once SLAM node is successfully up and running, TurtleBot3 will be exploring unknown area of the map using teleoperation.

It is important to avoid vigorous movements such as changing the linear and angular speed too quickly.

When building a map using the TurtleBot3, it is a good practice to scan every corner of the map.

Open a new terminal and run the teleoperation node from the Remote PC

$ export TURTLEBOT3\_MODEL=burger

$ roslaunch turtlebot3\_teleop turtlebot3\_teleop\_key.launch

Another Method:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1) Launch Simulation World

Three Gazebo environments are prepared, but for creating a map with SLAM, it is recommended to use either TurtleBot3 World or TurtleBot3 House.

Use one of the following commands to load the Gazebo environment.

$ export TURTLEBOT3\_MODEL=burger

$ roslaunch turtlebot3\_gazebo turtlebot3\_world.launch

2) Run SLAM Node

Open a new terminal from Remote PC with Ctrl + Alt + T and run the SLAM node. Gmapping SLAM method is used by default.

$ export TURTLEBOT3\_MODEL=burger

$ roslaunch turtlebot3\_slam turtlebot3\_slam.launch slam\_methods:=gmapping

3) Run Teleoperation Node

Open a new terminal from Remote PC with Ctrl + Alt + T and run the teleoperation node from the Remote PC.

$ export TURTLEBOT3\_MODEL=burger

$ roslaunch turtlebot3\_teleop turtlebot3\_teleop\_key.launch

4) Save Map

When the map is created successfully, open a new terminal from Remote PC with Ctrl + Alt + T and save the map.

$ rosrun map\_server map\_saver -f ~/map

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

If needed only ROS Neotic Instatllation

then ROS... NOETIC STEPS...

http://wiki.ros.org/noetic/Installation/Ubuntu

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Trouble Shooting:

This means Gazebo is currently downloading simulation models for you. If that took too long, you can alternatively download all Gazebo models offline and manually place then in the "models" folder (path: ~/.gazebo/models) as follows:

$ wget -l 2 -nc -r "http://models.gazebosim.org/"

$ cd models.gazebosim.org

$ mkdir -p ~/.gazebo/models/

$ cp -r \* ~/.gazebo/models/