Simulating Mowito Rosbot Documentation

Release 0.0.1

Mowito

Jul 03, 2020

Contents

Mowito's Navigation Stack

CHAPTER 1

Setting up mowito

1.1 User Registration

Register yourself on this website https://mowito.in/navigation_stack.html

We need your email to mail you the password, and to count how many people are using Mowito.

We won't spam. :)

1.2 Installing Mowito on Computer

1.2.1 Ubuntu 18 - ROS Melodic

1. Clone the repo in the home directory, using

```
git clone -b melodic https://github.com/mowito/mowito_amd64.git ~/mowito_amd64
```

2. Remove any previous installation of Mowito stack

   ```
   cd ~/mowito_amd64
   ./remove_mowito.sh melodic
   ```

3. Install the new Mowito stack

   ```
   ./setup_mowito.sh melodic
   ```

1.2.2 Ubuntu 16 - ROS Kinetic

1. Clone the repo in the home directory, using

```
git clone -b kinetic https://github.com/mowito/mowito_amd64.git ~/mowito_amd64
```

2. Remove any previous installation of Mowito stack

   ```
   cd ~/mowito_amd64
   ./remove_mowito.sh kinetic
   ```

3. Install the new Mowito stack

   ```
   ./setup_mowito.sh kinetic
   ```

CHAPTER 2

How To Use

On all the terminals you open, source the ROS:

For ROS Melodic

```
source /opt/ros/melodic/setup.bash
```

For ROS Kinetic

```
source /opt/ros/kinetic/setup.bash
```

```
TIP add the above line to your bashrc, so that it is automatically sourced
```

2.1 For Simulation

The package comes with Husarion's Rosbot Simulator. Here are the steps to use it.

2.1.1 A. Running Navigation with no Map / Navigation to create Map

1. Create a map using either of these three methods:

    1.1. **Manual navigation**:

    ```
    roslaunch ~/mowito/launch/simulation/sim_mw_mapping.launch
    ```

    in another terminal, start the remote control:

    ```
    rosrun teleop_twist_keyboard teleop_twist_keyboard
    ```

    and use it move the robot around

    1.2. **Navigation, by giving goals through the rviz**:

    ```
    roslaunch ~/mowito/launch/simulation/sim_navigation_with_no_map.
    launch
    ```

on rviz, give goals on the map, and the robot will move autnomously while creating the map

1.3. **Autonomous goal selection ,throuh Exploration**:

```
roslaunch~/mowito/launch/simulation/sim_mw_mapping_with_explore.
launch
```

on rviz you can see the robot automatically moving and exploring the area

2. Once you are done creating the map on rviz, save the map on a new terminal exeute the following:

```
cd && rosrun map_server map_saver -f mymap
```

the map (pgm and yaml) is saved in the home directory with the name mymap.pgm and mymap.yaml

2.1.2 B. Running Navigation with a pre-exitsting Map

1. Place the robot at the origin of map (the place where you started mapping)
2. Now, for running the entire system with mowito's controller run

```
roslaunch ~/mowito/launch/simulation/sim_mw_navigation.
launch
```

If you want to use the map created in the previous section use this

```
roslaunch ~/mowito/launch/simulation/sim_mw_navigation.launch
map_name:=mymap
```

3. In the rviz, click on the second top panel, click on the nav goal option, and click on the displayed map to give goal to the robot
4. look at the output on the rviz, the path planned and the motion of the robot.

2.2 On Real Robot

2.2.1 A. Running Navigation with no Map / Navigation to create Map

1. create a map using either of these three methods:

1.1. **manual navigation**

```
roslaunch ~/mowito/launch/run_mw_mapping.launch
```

in another terminal, start the remote control

```
rosrun teleop_twist_keyboard teleop_twist_keyboard
```

and use it move the robot around

1.2. **Navigation, by giving goals through the rviz**

```
roslaunch ~/mowito/launch/run_navigation_with_no_map.
launch
```

in another terminal start rviz

```
roslaunch ~/mowito/launch/start_rviz.launch
```

on rviz, give goals on the map, and the robot will move autnomously while creating the map

1.3. **Autonomous goal selection ,throuh Exploration**

```
roslaunch ~/mowito/launch/run_mw_mapping_with_explore.
launch
```

in another terminal start rviz

```
roslaunch ~/mowito/launch/start_rviz.launch
```

on rviz you can see the robot automatically moving and exploring the area

2. Once you are done creating the map on rviz, save the map on a new terminal exeute the following

```
cd && rosrun map_server map_saver -f mymap
```

the map (pgm and yaml) is saved in the home directory with the name mymap.pgm andmymap.yaml

2.2.2 B. Running Navigation with a pre-exitsting Map

1. Place the robot at the origin of map (the place where you started mapping)
2. Now, for running the entire system with mowito's controller run

```
roslaunch ~/mowito/launch/run_mw_navigation.launch map_name:=mymap
```

3. in another terminal start rviz

```
roslaunch ~/mowito/launch/start_rviz.launch
```

4. in the rviz, click on the second top panel, click on the nav goal option, and click on the displayed map to give goal to the robot
5. look at the output on the rviz, the path planned and the motion of the robot.

CHAPTER 3

Tips

If you have any problems with laser scan it probably means that you don't have a dedicated graphic card (or lack appropriate drivers). If that's the case then you'll have to change couple of things in /rosbot_description/urdf/rosbot_gazebo file:

Find:

```
<!-- If you cant't use your GPU comment RpLidar using GPU and uncomment␣
↪RpLidar using CPU gazebo plugin. -->
```

next coment RpLidar using GPU using <!– –> from <gazebo> to </gazebo> like below:

```
<!-- gazebo reference="rplidar">
<sensor type="gpu_ray" name="head_rplidar_sensor">
<pose>0 0 0 0 0 0</pose>
<visualize>false</visualize>
<update_rate>40</update_rate>
<ray>
      <scan>
      <horizontal>
      <samples>720</samples>
      <resolution>1</resolution>
      <min_angle>-3.14159265</min_angle>
      <max_angle>3.14159265</max_angle>
      </horizontal>
      </scan>
      <range>
      <min>0.2</min>
      <max>30.0</max>
      <resolution>0.01</resolution>
      </range>
      <noise>
      <type>gaussian</type>
      <mean>0.0</mean>
      <stddev>0.01</stddev>
```

(continues on next page)

(continued from previous page)

```
      </noise>
</ray>
<plugin name="gazebo_ros_head_rplidar_controller" filename="libgazebo_ros_
↪gpu_laser.so">
      <topicName>/rosbot/laser/scan</topicName>
      <frameName>rplidar</frameName>
</plugin>
</sensor>
</gazebo -->
```

Now uncomment RpLidar using CPU plugin removing <!– –>.

If you want to make your laser scan visible just change:

```
<visualize>false</visualize>
```

to:

```
<visualize>true</visualize>
```

in the same plug in.