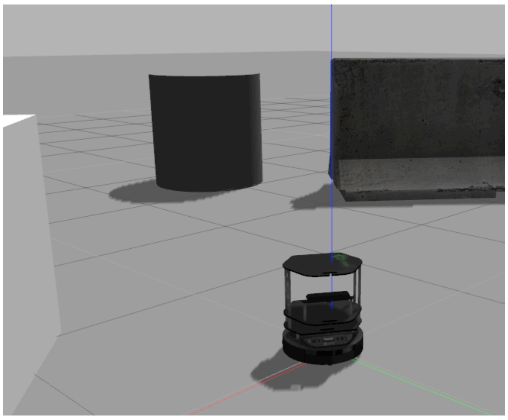
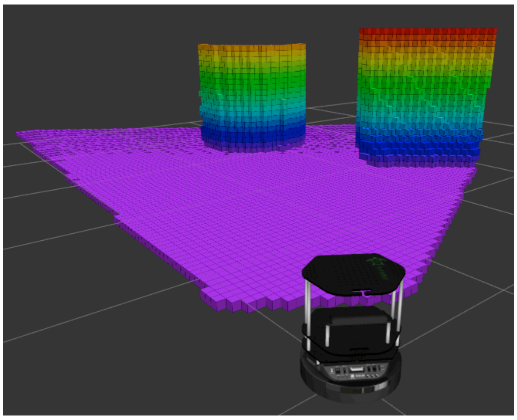
Autonomous Robots

Localization and mapping with Turtlebot

This document will guide you through the practical work related to preparing the Turtlebot for having its localization and mapping ready. It will also propose you some exercises for checking the performance of these systems.





1. ROS dependencies for simulation

Install dependencies:

```
sudo apt-get install ros-kinetic-turtlebot* ros-kinetic-octomap* ros-kinetic-octovis
```

Clone repository:

```
cd ~/catkin_ws/src
git clone https://bitbucket.org/udg_cirs/mapping_planning_tutorial.git
```

Compile:

```
cd ~/catkin_ws/
catkin_make
```

2. Checking vehicle's Odometry in simulation.

In one terminal execute (turtlebot simulator and teleoperation):

```
roslaunch control_turtlebot start_turtlebot_modules.launch
```

In a second terminal check the vehicle's odometry:

```
rostopic echo /odom
```

Teleoperate the turtlebot around the virtual environment:

```
Over the first terminal, and using the keyboard, control the turtlebot to move around:
u      i      o
j      k      l
m      ,      .
```

Verify that the vehicle's odometry has changed (first terminal):

```
rostopic echo /odom
```

3. Building a map using Octomap

Stop and restart the simulator. In the first terminal execute (simulator and teleoperation):

```
roslaunch control_turtlebot start_turtlebot_modules.launch
```

In the second terminal execute (Octomap, mapping framework):

```
roslaunch control_turtlebot start_turtlebot_mapping_control.launch
```

Teleoperate the turtlebot around the virtual environment:

```
Over the first terminal, and using the keyboard, control the turtlebot to move around:

u      i      o

j      k      l

m      ,      .
```

In a third terminal, save the octomap:

```
cd ~/

rosrun octomap_server octomap_saver map.bt
```

In the same terminal, visualize the octomap using octovis:

```
cd ~/

octovis ~/map.bt

What happens when pressing 1 in octovis ?

What happens when pressing f in octovis ?
```

4. ROS dependencies for real Turtlebot (optional)

Do not forget to set up the ROS_MASTER_URI to the turtlebot IP!! From your computer, copy the control_turtlebot package:

```
scp -r ~/catkin_ws/src/mapping_planning_tutorial/control_turtlebot/
turtlebot@10.42.0.1:~/catkin_ws/src/
```

Install `dependencies` **(connect to the turtlebot):**

```
ssh turtlebot@10.42.0.1 #password: ros

sudo apt-get install ros-kinetic-octomap-server ros-kinetic-octovis #password: ros
```

Compile (in the turtlebot):

```
cd ~/catkin_ws/

catkin_make
```

5. Checking vehicle's Odometry (optional)

In one terminal (connected to the Turtlebot), edit the launch file in the turtlebot:

```
rosed control_turtlebot start_turtlebot_modules.launch

#set this flag to false

<arg name="enable_turtlebot_simulation" value="false"/>
```

Then, execute (turtlebot controller and teleoperation):

```
roslaunch control_turtlebot start_turtlebot_modules.launch
```

In a second terminal (also connected to the turtlebot):

```
rostopic echo /odom
```

Teleoperate the turtlebot around:

```
Over the first terminal, and using the keyboard, control the turtlebot to move around:

u       i       o

j       k       l

m       ,       .
```

Verify that the vehicle’s odometry has changed (first terminal):

```
rostopic echo /odom
```

6. Building a map using Octomap with the real Turtlebot (optional)

Stop (ctrl+c) and restart the vehicle nodes (first terminal) In a second terminal (also connected to the turtlebot):

```
rosed control_turtlebot start_turtlebot_mapping_control.launch

#edit the following lines

<!--remap from="/cloud_in" to="/camera/depth/points" /--> <!-- simulator -->

<remap from="/cloud_in" to="/camera/depth_registered/points" /> <!-- real turtlebot -->
```

In the second terminal execute (connected to the turtlebot):

```
roslaunch control_turtlebot start_turtlebot_mapping_control.launch
```

Teleoperate the turtlebot around the environment:

```
Over the first terminal, and using the keyboard, control the turtlebot to move around:

u       i       o

j       k       l

m       ,       .
```

In a third terminal (not connected to the turtlebot), launch rviz:

```
roslaunch control_turtlebot octomap_saver map.bt
```

In a fourth terminal (not connected to the turtlebot), run:

```
cd ~/

rosrun octomap_server octomap_saver map.bt
```

In the same terminal, visualize the octomap using octovis:

```
cd ~/

octovis ~/map.bt
```

7. Submission.

WORK TO DO: After following the previous sections, perform some movements with the Turtlebot for checking the following systems:

a) Map building. Move the robot through the corridors and verify the accuracy of the measurements. Perform several trials in order to have several maps.

b) Repeat the same exercise using the real robot (optional). Perform also movements in order to estimate the lineal and angular errors of the odometry.

c) Report. Write a report explaining all the work done on this laboratory exercise. Submit the report in pdf and also the files corresponding to the maps (.bt).