***Introduction to Robotic Systems Course***

**LAB 7**

**Basic Slam Operation**

Contents

[1 Introduction 1](#_Toc30061848)

[1.1 Lab Overview 1](#_Toc30061849)

[2 Requirements 1](#_Toc30061850)

[3 Simultaneous Localization and Mapping 1](#_Toc30061851)

[4 Task: Install Mapping Packages 1](#_Toc30061852)

[5 Task: Mapping using SLAM 2](#_Toc30061853)

[6 Task: Navigation 3](#_Toc30061854)

[7 Task: Simulate using Gazebo 4](#_Toc30061855)

[7.1 Install Gazebo 4](#_Toc30061856)

[7.2 Launch Gazebo 4](#_Toc30061857)

[7.3 Use SLAM in Gazebo 4](#_Toc30061858)

# Introduction

## Lab Overview

In this lab, we will use the Simultaneous Localization and Mapping (SLAM) and Navigation features of the robot. The robot used in this lab comes with the LDS-01 device that can be used for 360° scanning during SLAM operation.

# Requirements

The following hardware and software are required to complete this lab:

* **Hardware:** TurtleBot 3 Burger.

# Simultaneous Localization and Mapping

SLAM is a technique that uses the LDS data to generate a map of the environment and estimate the location of the robot in the environment.

# Task: Install Mapping Packages

There are different mapping techniques, for example, Gmapping, Cartographer, Hector Mapping, Karto, and Frontier Exploration. In this task, we will install the packages required to run these different mapping techniques for SLAM operation.

Run the following commands on a terminal on the Remote PC to install the packages for the various mapping techniques.

* Gmapping: This was installed during ROS setup.
* Hector Mapping: Run the following command

$ sudo apt-get install ros-kinetic-hector-mapping

* Karto

$ sudo apt-get install ros-kinetic-slam-karto

* Frontier Exploration

$ sudo apt-get install ros-kinetic-frontier-exploration ros-kinetic-navigation-stage

* Cartographer

$ sudo apt-get install ros-kinetic-cartographer ros-kinetic-cartographer-ros ros-kinetic-cartographer-ros-msgs ros-kinetic-cartographer-rviz

* + To use the Cartographer by google on ROS Kinetic, run the commands below.
1. $ sudo apt-get install ninja-build libceres-dev libprotobuf-dev protobuf-compiler libprotoc-dev
2. $ cd ~/catkin\_ws/src
3. $ git clone https://github.com/googlecartographer/cartographer.git
4. $ git clone https://github.com/googlecartographer/cartographer\_ros.git
5. $ cd ~/catkin\_ws
6. $ src/cartographer/scripts/install\_proto3.sh
7. $ rm -rf protobuf/
8. $ rosdep install --from-paths src --ignore-src -r -y --os=ubuntu:xenial
9. $ catkin\_make\_isolated --install --use-ninja

# Task: Mapping using SLAM

In this task, the robot’s environment will be mapped using SLAM.

To map the environment, do the following

* Open a new terminal on the Remote PC and run the following command:

$ roscore

This command allows ROS nodes to communicate in a network.

* On the TurtleBot PC, run the following command to start the basic packages needed for the robot applications.

$ roslaunch turtlebot3\_bringup turtlebot3\_robot.launch

* Open another terminal on the Remote PC and run the following command to launch the SLAM node Gmapping and load the robot on RViz with Gmapping.

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

As the robot scans the room, you will see its environment taking shape on the RViz window.

Before it can navigate the room to better map the environment, we will use teleoperation control the robot.

* Open another terminal window on the Remote PC and run the following commands.

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

After mapping the room, save the map. On a new terminal window, run the following command:

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

The above command saves the map with the name “map.” You can give your map a different name.

# Task: Navigation

The navigation capabilities referred to in this section allow the robot to move from one location to another when a destination is specified. In this operation, the robot uses the information from the saved map to determine its current position to arrive at a direction of movement to get to the specified destination.

To use the navigation features of the TurtleBot, do the following:

* On the Remote PC, run the following command:

$ roscore

* On the TurtleBot PC run the following command:

$ roslaunch turtlebot3\_bringup turtlebot3\_robot.launch

* Run the following command on a new terminal on the Remote PC.

$ roslaunch turtlebot3\_navigation turtlebot3\_navigation.launch map\_file:=$HOME/map.yaml

* + Here, we have specified the name of the map which is “map.yaml.”. If you have saved your map with a different name, then specify it as your\_*map\_name*.yaml.

Next, use teleoperation to move the robot around and 2D Pose Estimate operation in RViz (top tool bar) to aid the robot in estimating its current position on the map.

To do this:

* Unselect Global Map and Amacl Particles on the left panel
* Click the 2D Pose Estimate Button (highlighted in Figure 1)
* Start Teleoperation using the following command on a new terminal on the Remote PC.

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

* When teleoperation is started, move the robot around the room so it can estimate its current and post position around the map.
* When done with teleoperation, close its terminal window.

Next, we will try out the navigation of the robot.

To do this:

* Click 2D Nav Goal button from the top tool bar.
* Choose a place on the map for the robot to navigate to and observe the robot’s movement.

The TurtleBot3 will not get close to the red areas shown on the map as a safety feature.



*Figure 1: RViz window showing highlighted 2D Estimate button and navigation destination.*

# Task: Simulate using Gazebo

Gazebo is a 3D simulation environment used for the development of a virtual robot. With Gazebo, you can use the robot’s features you have learned in this lab.

## Install Gazebo

To download and install Gazebo and the necessary packages, run the following commands.

1. $ cd ~/catkin\_ws/src/
2. $ git clone https://github.com/ROBOTIS-GIT/turtlebot3\_simulations.git
3. $ cd ~/catkin\_ws && catkin\_make\_isolated --install --use-ninja
4. $ source install\_isolated/setup.bash

## Launch Gazebo

Launch Gazebo with the following commands to see the map and robot.

1. $ roslaunch turtlebot3\_gazebo turtlebot3\_world.launch
2. $ roslaunch turtlebot3\_teleop turtlebot3\_teleop\_key.launch

Use teleoperation on another terminal to see if you can move the robot around.

## Use SLAM in Gazebo

To use SLAM in Gazebo

* First, scan and map using gmapping or any other node with teleoperation by running the following commands.
1. $ roslaunch turtlebot3\_slam turtlebot3\_slam.launch slam\_methods:=gmapping
2. $ roslaunch turtlebot3\_teleop turtlebot3\_teleop\_key.launch
* Save the map with a name of your choice, just replace the map at the end with a name.
1. $ rosrun map\_server map\_saver -f ~/map



*Figure 2: RViz window showing simulated environment.*

* Next, close RViz and launch the navigation packages with your map using the following command.

$ roslaunch turtlebot3\_navigation turtlebot3\_navigation.launch map\_file:=$HOME/map.yaml

* Use teleoperation to move the robot around and 2D Pose Estimate operation in RViz to estimate its current position on the map.
* Next, use the 2D Nav Goal to give commands to the robot.
* You can also use another map called the TurtleBot3 House. To do this, run the following command.

$ roslaunch turtlebot3\_gazebo turtlebot3\_house.launch

Figure 3 shows the RViz and the Gazebo simulation windows. The current position of the robot, which corresponds to its position on the simulated environment, is shown on the RViz window. The destination of the robot is also specified in the RViz window.



*Figure 3: RViz and Gazebo simulation window.*