UNIVERSITY OF LIVERPOOL

Department of Electrical Engineering and Electronics

ELEC230
Robotic Systems - Assignment 3
Roomba Vacuum Cleaner Gazebo Simulation in C++

Module	ELEC230
Coursework name	Assignment 3
Component weight	25%
Semester	2
HE Level	5
Lab location	• EEE Lab 402 & 406
Work	Individual
Timetabled time	30 hours - including lab time
Suggested private study	8 hours including report-writing
Assessment method	Individual, formal, word-processed reports in the format instructed in the Marking Criteria, along with other files listed in "What to hand in".
Submission format	Online via CANVAS.
Plagiarism / collusion	Standard University penalties and procedures apply for plagiarism and collusion.
Submission deadline	**Friday 18th March 2022, 23:59**

Late submission	Standard University penalties apply
Resit opportunity	August resit period (if total module failed)
Marking policy	Marked and moderated independently
Anonymous marking	Yes
Feedback	Via comments on your CANVAS submission, online
Learning outcomes	▪ (BH1) Understanding Linux and the mechanisms provided for multi-tasking ▪ (BH2) Understanding the features of an Object Orientated Programming language and the ability to code in C++ ▪ (BH4) Understanding the ROS system ▪ More specifically: understanding the package management system in ROS; understanding and implementing robot simulation using ROS, Turtlesim, Gazebo

Continues on next page

Marking Criteria

Section	Marks	Indicative characteristics	
		Adequate / pass (40%)	**Very good / Excellent**
Presentation, and structure	10%	▪ Contains document with cover page (title, background, academic integrity declaration), a discussion page, and screenshots (see below) ▪ Contains original source code, manifest file, CMakeLists.txt, launch files ▪ Comprehensible language; punctuation, grammar and spelling accurate ▪ Equations legible, numbered and presented correctly	▪ Appropriate use of technical, mathematic and academic terminology and conventions if relevant ▪ Word processed with consistent formatting ▪ Pages are numbered; figures and tables are captioned ▪ All sections are clearly signposted ▪ Correct cross-referencing (of figures, tables, equations) and citations where relevant
Introduction, Method, Design	50%	▪ Problem background is introduced clearly ▪ Clear, original code and comments ▪ Code is clearly laid out and appropriately commented ▪ Design of each code segment follows a logical sequence ▪ Code is tidy, efficient and easy to follow / understand ▪ Some discussion of the procedures undertaken	▪ Excellent understanding of the problem background is displayed ▪ Comments show excellent understanding of syntax and semantics ▪ Coding is elegant and, where relevant, sophisticated for each task, while remaining easy to follow ▪ Principle of "DRY[1]" is followed ▪ Discussion of what worked and what did not ▪ Discussion of all the testing carried out
Results	40%	▪ Screenshots of program output are presented for each task, including the full desktop with relevant windows and the taskbar with date and time, as evidence of original work ▪ Screenshots and code demonstrate at least partially correct operation	▪ Screenshots and code demonstrate successful, correct output for every task

[1] Don't Repeat Yourself

ELEC 230 C++ Assignment 3 (2021-2022) – 30% weighting

Aim of this assignment

The aim of this assignment is to give you a way in to writing software for controlling a (simulated) robot using the Robot Operating System (ROS).

In this assignment you will create a ROS node to drive the robot around with a simple wanderer algorithm, very like a 'Roomba' robot vacuum cleaner. The robot should move forward until it reaches an obstacle, then rotate in the same position until the way ahead is clear, then move forward again and repeat.

Continues on next page

Rules and Requirements of your Assignment

You code must be executable in a Linux environment without any modification. For you to work and be able to execute the assignment, make sure you have a ROS Distribution (i.e., noetic) wit Gazebo installed. For more information please have a look at the lecture notes and lab sessions in CANVAS.

Before you start this assignment, you should make sure that you understand the concepts in ROS tutorials 1-6, 8, 11 or 12, and 13 (These were covered in lab sheets 1-4). In addition, go over all the code samples from class and make sure you understand them thoroughly. Finally, make sure that the turtlebot_gazebo packages are installed on your machine.

The Assignment

1. First make a new ROS package called wander_bot, with the appropriate dependencies. If you need a refresher, please consult either the lecture material or ROS.org tutorials.
2. Create a launch File. This launch file needs to run the Gazebo simulator and the wander_bot node that you are going to write without the user needing to invoke them separately.
3. Your task is now to write the wander_bot node. The node should implement a simple algorithm:
    - If the robot is moving sufficiently close to an obstacle in front of it, then rotate it in the direction that is freer from obstacles (i.e., if there is an obstacle on the robot's right, it should turn left) until the way ahead is clear;
    - If there is no obstacle blocking its path, move forward as a default.
        - You can use the Stopper node we used in Week 4 as a template for this assignment.
4. Verify that the wander_bot node works, by running it and watching the robot in the simulator.

Continues on next page

The Rules

1. Make sure that your code is tidy and well-commented. (You can make your code tidier in terms of indentations by using the Reformat code option inside the Clion IDE).
2. This goes without saying: you should do this lab work on your own. All the work you turn in should be yours, and not done in collaboration with anyone else. If you use any external sources of inspiration, other than ros.org, then let us know in a README file.

What to Hand In

You should hand in everything that someone else needs to run your code.

For this assignment, that means:

- Your source code. This should be adequately commented, so that each distinct part of the code is clearly explained.
- manifest file,
- CMakeLists.txt,
- launch files.

You should also include:

- A Word document including a 1-page cover sheet with title, background and academic integrity declaration, followed by no more than 1 page of discussion of: procedures followed; what worked and what did not; the testing carried out. *The rest of your Word document should include screenshots as described in the Marking Criteria. These should be presented as numbered figures, referred to (where relevant) in your discussion.*

You should *not* hand in executable files, or any other files that can be regenerated.

Your code should be able to run after writing these two commands in a terminal:

```
catkin-make --pkg wander_bot
roslaunch wander_bot wander_bot.launch
```

Continues on next page

Helpful insights into how ROS works and details of the Gazebo simulation software can be found at ROS.org.