Course name	Module name	Coordinator	Date
Intelligent Systems and Robotics, MSc	Intelligent Mobile Robots	Dr Aboozar Taherkhani	06/04/2021
Title	Assignment 1		
Author	Lukasz Pszonak (P2619730)		
Type	Academical report		

ABSTRACT

The report concerns the work related to the first Assignment of Intelligent Mobile Robots led by Dr Aboozar Taherkhani. The objective of the assignment is to write a program which creates a 2D map of the robot's environment. Out of three options that were available to students, the author is choosing an occupancy grid that will be displayed with the usage of RVIZ graphical tool. It is a real-time image library. The report contains 6 important sections, which are: Introduction (Assignment overview), Map Construction Technique (Further explanation upon the choice of the map construction method), Software Implementation (Basic overview of the code), Testing and Results (Figures displaying screenshots of created maps and their brief analysis), Conclusions (summary upon the realized work and future ideas for extending the program) and Bibliography which will contain all references mentioned in the report.

TABLE OF CONTENTS

Introduction

The Intelligent Mobile Robots Assignment 1 requires utilizing already gained knowledge of ROS environment as well as Python programming language in order to create a 2D map of a robot's environment. The robot used in this work is the popular Turtlebot3 Burger model. It is a new generation mobile robot that is highly modular, compact and customizable, however, the latter is not important for the purpose of this work[1].

ROS means Robot Operating System and this name stands for a highly flexible framework that serves the purpose of writing robot software. It simplifies the creation of complex robot behaviors across a variety of platforms by being the collective system of environments, tools, conventions and libraries, all to make a robust general-purpose robot operating system[2].

Tackling the Assignment is possible thanks to the broad spectrum of learning materials that were either prepared or recommended by the Module Leader. The author is referring to them in specific sections of the report and lists them all down in the Bibliography.

Map Construction Technique

The author chooses to proceed with the occupancy grid mapping as the technique for creating a map that resembles the environment inside which the robot is put. It is one of 3 techniques that are available for this assignment, with other being sets of either 2D points or line segments. The motivation behind the author's choice is the fact 4 learning sessions with the module leader have addressed the occupancy grid mapping for the purpose of this assignment and the author feels confident with his understanding of the topic. Below represents the author's point of view of the subject.

---

https://www.robot-advance.com/EN/art-turtlebot3-burger-1997.htm#:~:text=TurtleBot3%20Burger%20is%20a%20new,connected%20objects...)., date and time of access: 05.04.2021 |17.03.

[2] https://www.ros.org/about-ros/, date and time of access: 05.04.2021 | 18:31.

The occupancy grid mapping (2D) is used to represent a workspace (environment), within which a robot operates, as a grid. In order to do that we need information about the environment that would be collected in real-time or pre-loaded prior to the robot's start. The "occupancy" refers to the probability that the detected area/space is being occupied in the time of data-collection. Occupancy grids are especially useful in path-planning because they show where obstacles are. Probability values from 0 to 1 with 0 representing an empty space and 1 representing an occupied space. More advanced occupancy grids use values between 0 and 1 (as opposed to only 0 and 1) because it is especially useful in unknown environments with dynamic elements other than the robot[3].

Software Implementation

The author uses a python file called mapper.py, shared by the module leader for the purpose of this assignment, and modifies it in order to create an occupancy grid and display it in rviz, which is a graphical tool accessible in ROS. Prior to that the author creates a file named scan.py according to laboratory instructions. Scan.py is responsible for collecting the data from all 360 laser sensors (from 0 to 359) of the robot as well as moving it in a manner such as to avoid any obstacles on its way[4,5]. Below overviews the modifications of the mapper.py file that are made in order to create an occupancy grid:

- Adding the necessary libraries such as math, euler_from_quaternion, and quaternion_from_euler in order to operate on mathematical functions and units (radians),
- Defining a function 'get_indix' that fits the occupancy grid to the map,

[3] https://www.mathworks.com/help/robotics/ug/occupancy-grids.html, date and time of access: 05.04.2021 | 21:21.
[4] https://vle.dmu.ac.uk/bbcswebdav/pid-5047451-dt-content-rid-10229601_1/courses/IMAT5233_2021_502/week6%20-%20lab%20sheet.pdf, date and time of access: 06.04.2021 | 15:30.
[5] https://vle.dmu.ac.uk/bbcswebdav/pid-5047454-dt-content-rid-10288697_1/courses/IMAT5233_2021_502/week7-%20lab%20sheet-%20Updated-Creating%20a%202D%20Map%20of%20an%20Unknown%20Environment%281%29.pdf, date and time of access: 06.04.2021 | 23:01.

- Writing a for loop inside ‘scan_callback’ function (responsible for updating the map on every scan callback) that finds the x and y positions of the obstacle, rotates them into the global coordinate system and finally translates them into the global coordinate system x and y coordinates. It is recreated step-by-step with necessary modifications, (for example reducing the robot radius – further explanation provided in the code) following the example shown in the Week 4 presentation, which is provided by the Module Leader[6]. At the end, the map needs to be inverted in order to avoid misplacing the occupied cells.

The code (with comments) is submitted separately as a zip file.

Testing and Results

In this section of the report the author analysis the results and provides insight upon his testing process. Thanks to good obstacle avoidance program and the fact that laser sensors have relatively good range, the approximate time required to build an occupancy grid is around 3 minutes. During the testing process the author comes upon multiple errors that require adjusting (map size, centering the grid, robot’s radius that affects the output of the map). The crucial factor affecting the map building process proves to be the radius of the robot. Figures 1 – 3 display the built map for various radiuses of the Turtlebot3. Figure 4 presents the map created with the radius of the robot equal to that from Figure 3, however with an addition of a third occupancy level (0.1). By occupancy levels the author refers to 3 distinct levels of probability (0, 0.1 and 0.9). The author is confident that the submitted code could be expanded upon either in the next assignment or with his project in order to map dynamic objects (occupancy of cells would vary in time). A use of a recurrent neural network could be necessary to achieve it.

6 https://vle.dmu.ac.uk/webapps/blackboard/content/listContent.jsp?course_id=_559254_1&content_id=_5047438_1, date and time of access: 06.04.2021, 19:43.

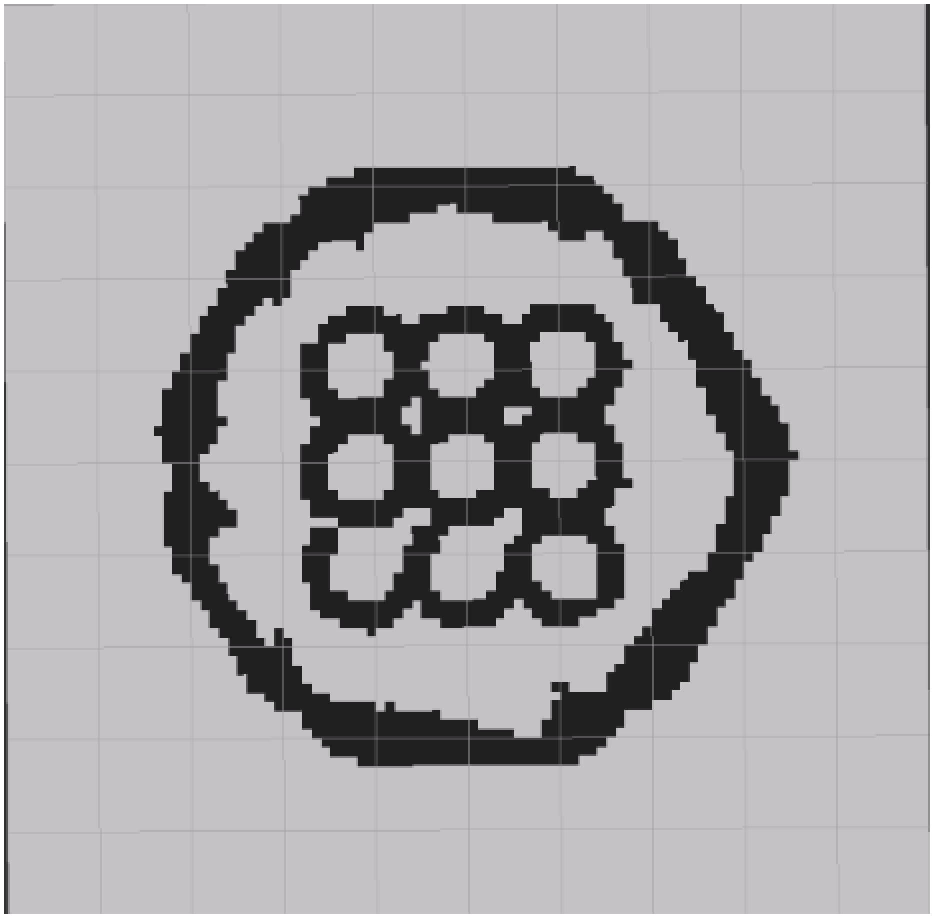

Figure 1. The occupancy grid map for the radius of the robot equal to 0.6.

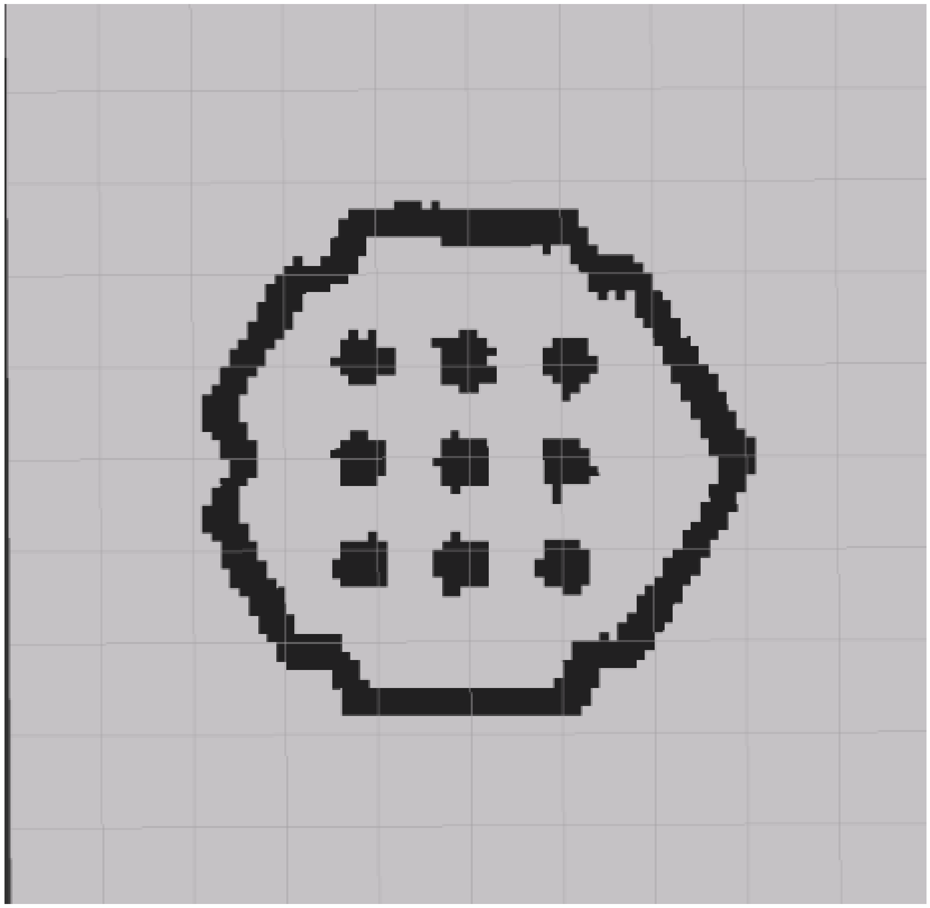

Figure 2. The occupancy grid map for the radius of the robot equal to 0.1.

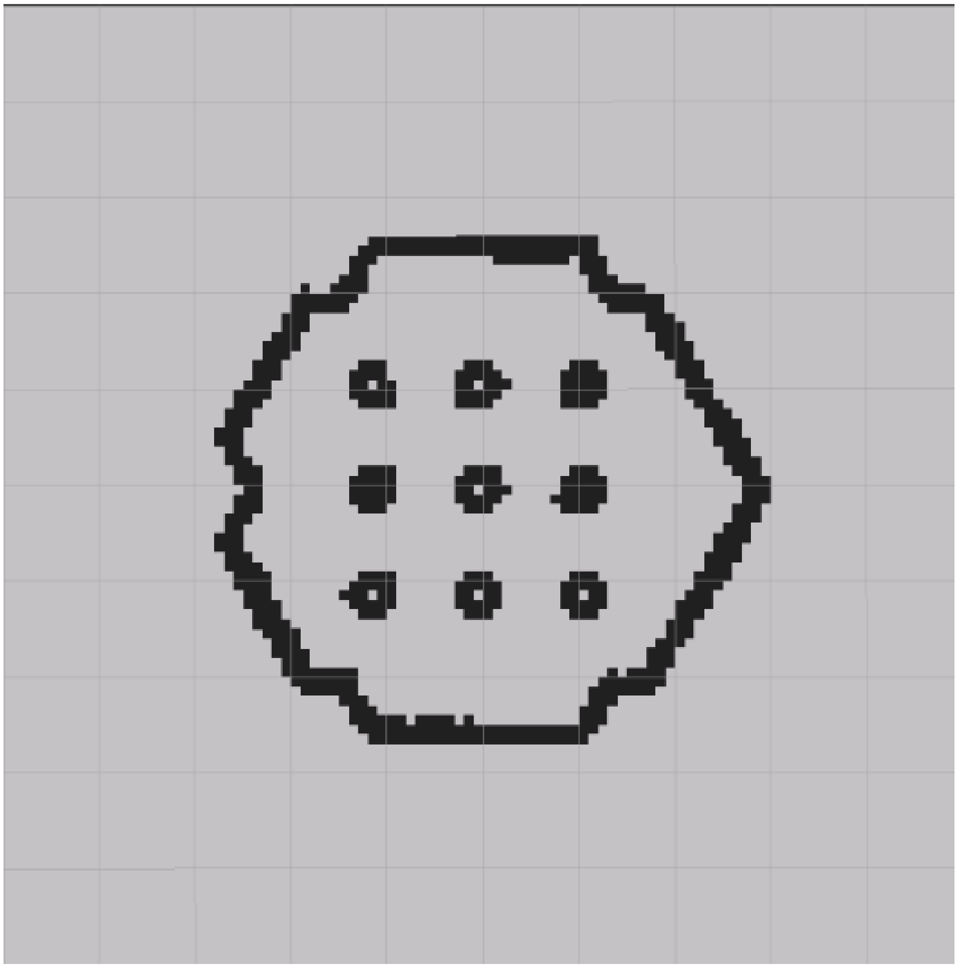

Figure 3. The occupancy grid map for the radius of the robot equal to 0.00005.

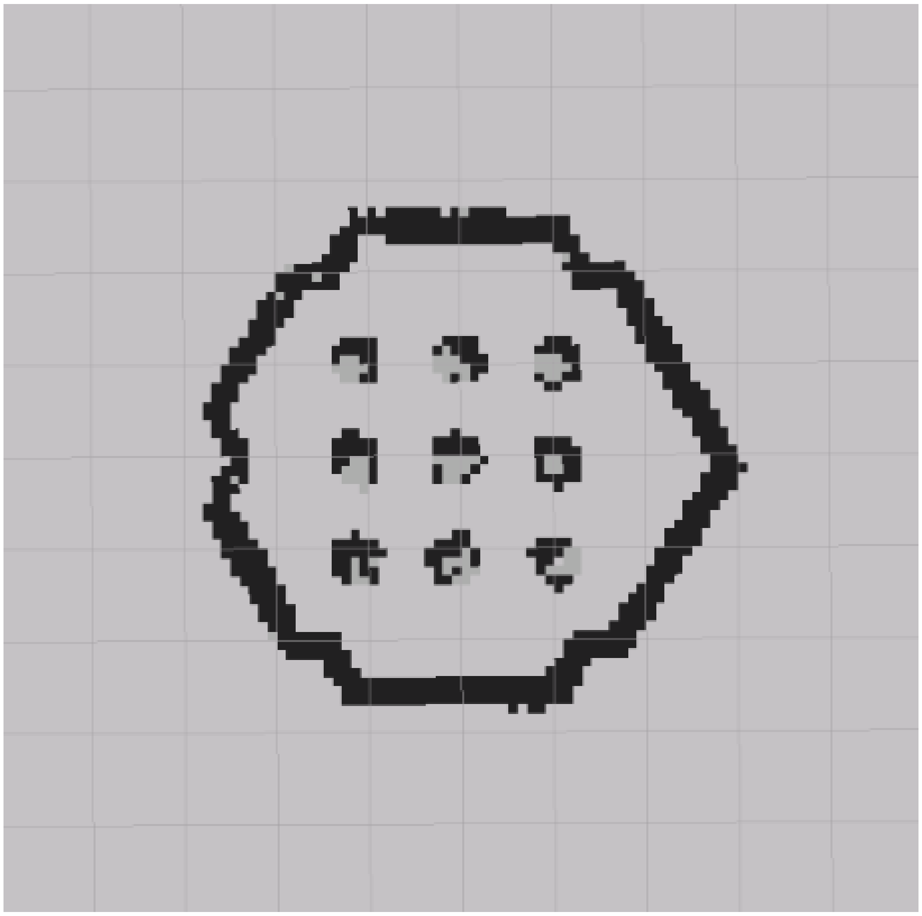

Figure 4. The occupancy grid map for the radius of the robot equal to 0.00005 and 3 occupancy levels (0, 0.1 and 0.9).

The author is content with the map visible on Figure 3, however, a potential to expand the program in order to test it within more complex environments, maybe even with dynamic elements, is discovered thanks to the map displayed on Figure 4. Given a scenario where there are less elements in the middle of the map, but they can move, this approach seems appropriate, however it requires conducting a research and much more testing. The author comes upon articles mentioning the topic of Dynamic Occupancy Grid Mapping with the use of a Recurrent Neural Network, what proves that Occupancy Grid Mapping can be used for dynamic environments[7].

[7] https://arxiv.org/abs/2011.08659, date and time of access: 06.04.2021 | 22:00.

Conclusions

After conducting the tests and getting the results the author concludes as follows:

- The most impactful factor in the accuracy of creating the map is the robot's radius. Reducing it dramatically improves the efficiency and the look of the map of the environment,
- In the best-case scenario, the robot should be treated as a moving point,
- The map displayed on Figure 3 is chosen as the best result, given the author's experience with ROS and his knowledge.
- The map displayed on Figure 4 shows that the program has a great potential (provided having further modifications and most likely with the simultaneous use of a neural network) to be used with more advanced environments.

The comparison between the created map and the environment is displayed on Figure 5.

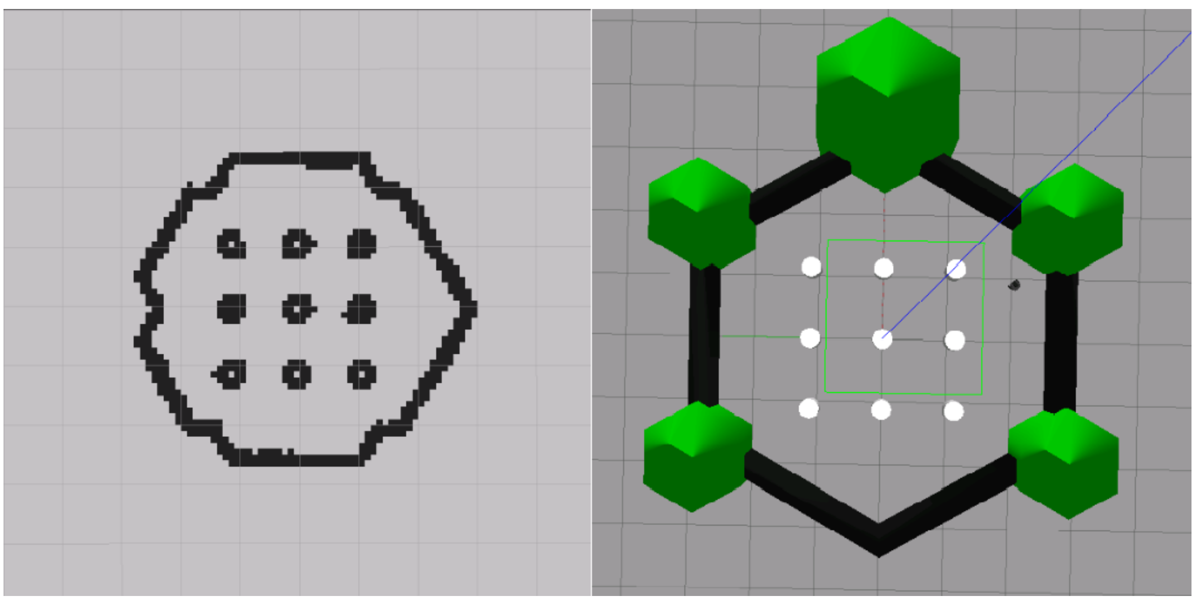

Figure 5. The comparison between the created occupancy grid and the robot's environment.

The Assignment proves demanding and challenging in some aspects. Thanks to it the author feels more confident with python and ROS and looks forward to tackling the next Assignment.

Bibliography

1. https://www.robot-advance.com/EN/art-turtlebot3-burger-1997.htm#:~:text=TurtleBot3%20Burger%20is%20a%20new,connected%20objects...)., date and time of access: 05.04.2021 |17.03.
2. https://www.ros.org/about-ros/, date and time of access: 05.04.2021 | 18:31.
3. https://www.mathworks.com/help/robotics/ug/occupancy-grids.html, date and time of access: 05.04.2021 | 21:21.
4. https://vle.dmu.ac.uk/bbcswebdav/pid-5047451-dt-content-rid-10229601_1/courses/IMAT5233_2021_502/week6%20-%20lab%20sheet.pdf, date and time of access: 06.04.2021 | 15:30.
5. https://vle.dmu.ac.uk/bbcswebdav/pid-5047454-dt-content-rid-10288697_1/courses/IMAT5233_2021_502/week7-%20lab%20sheet-%20Updated-Creating%20a%202D%20Map%20of%20an%20Unknown%20Environment%281%29.pdf, date and time of access: 06.04.2021 | 23:01.
6. https://vle.dmu.ac.uk/webapps/blackboard/content/listContent.jsp?course_id=_559254_1&content_id=_5047438_1, date and time of access: 06.04.2021, 19:43.
7. https://arxiv.org/abs/2011.08659, date and time of access: 06.04.2021 | 22:00.