TECHNICAL UNIVERSITY OF CRETE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
TELECOMMUNICATIONS DIVISION

Integration of RFID Localization, Obstacle Avoidance and Visualization in a Moving Robot

By

Antonis Nikolaidis

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DIPLOMA OF

ELECTRICAL AND COMPUTER ENGINEERING

THESIS COMMITTEE

Professor Aggelos Bletsas, Thesis Supervisor

Professor Aikaterini Mania

Professor Michail G. Lagoudakis

Chania, October 2023

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΠΡΟΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

Ενσωμάτωση Εντοπισμού RFID, Αποφυγής Εμποδίων και Οπτικοποίησης σε Κινούμενο Ρομπότ

από τον Αντώνη Νικολαΐδη

ΜΙΑ ΔΙΑΤΡΙΒΗ ΠΟΥ ΥΠΟΒΑΛΛΕΤΑΙ ΣΕ ΜΕΡΙΚΗ ΕΚΠΛΗΡΩΣΗ ΤΩΝ ΑΠΑΙΤΗΣΕΩΝ ΓΙΑ ΤΟ ΔΙΠΛΩΜΑ

ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Εξεταστική Επιτροπή

Καθηγητής Άγγελος Μπλέτσας (επιβλέπων)

Καθηγήτρια Αικατερίνη Μανιά

Καθηγητής Μιχαήλ Λαγουδάκης

Χανιά, Οκτώβριος 2023

Abstract

This thesis is pertinent to robot navigation across the same or different floors of the same building, while avoiding obstacles in extensive indoor areas. The robot is mobile across multiple floors exploiting camera and laser sensors to ensure elevator entry and navigation through different levels. The designed robot system is autonomous and capitalizes on precise localization algorithms for ultra-low cost radio frequency identification (RFID) tags. However, the autonomy is temporarily halted, when the robot requests an individual to press buttons through voice commands at the elevator. It is exhibited that the robot can enter and exit the elevator, when moving between different floors, and navigate safely, avoiding standing or moving humans in the building. The visualization of the robot's path and the accessed RFID tags is performed with tools provided by the robot operating system (ROS). The user has the opportunity to select different routes, where the robot can either move on both floors, or in one of them exclusively. The selected route established in this thesis, was used to gather the traveling measurements and was also set to visit three specific locations and localize RFID tags on each location. Two of these locations were placed on the first floor, while the remaining one was positioned on the second one. When the robot navigates to these locations, while avoiding obstacles, it has to enter the elevator twice, which results in a real-world travel time of 19 minutes and 11 seconds, covering approximately a total distance of 194 meters, with a constant robot velocity at 0.25 meters/second. In free-space, without unpredictable obstacles, such as moving people, it would require 12 minutes and 56 seconds. However, by taking into consideration the time during the elevator (3 minutes) and the RFID tag localization procedures (2 minutes and 45 seconds), an additional time interval of 5 minutes and 45 seconds is added, giving a total time of 18 minutes and 41 seconds.

Thesis Supervisor: Professor Aggelos Bletsas

Περίληψη

Αυτή η διατριβή αφορά την πλοήγηση με ρομπότ στους ίδιους ή διαφορετικούς ορόφους του ίδιου κτιρίου, αποφεύγοντας ταυτόχρονα εμπόδια σε εκτεταμένους εσωτερικούς χώρους. Το ρομπότ είναι κινητό σε πολλούς ορόφους και εκμεταλλεύεται αισθητήρες κάμερας και λέιζερ για να εξασφαλίσει την είσοδο του στον ανελκυστήρα και την πλοήγηση σε διαφορετικά επίπεδα. Το σχεδιασμένο σύστημα ρομπότ είναι αυτόνομο και αξιοποιεί αλγόριθμους ακριβούς εντοπισμού για ετικέτες εξαιρετικά χαμηλού κόστους αναγνώρισης ραδιοσυχνοτήτων (RFID). Ωστόσο, η αυτονομία διακόπτεται προσωρινά όταν το ρομπότ ζητά από ένα άτομο να πατήσει κουμπιά μέσω φωνητικών εντολών στο ασανσέρ. Αποδεικνύεται ότι το ρομπότ μπορεί να μπαίνει και να βγαίνει από τον ανελκυστήρα όταν κινείται μεταξύ διαφορετικών ορόφων και να πλοηγείται με ασφάλεια, αποφεύγοντας ανθρώπους που μπορούν να στέκονται ή να μετακινούνται στο κτίριο. Η απεικόνιση της διαδρομής του ρομπότ και των προσπελάσιμων ετικετών RFID εκτελείται με εργαλεία που παρέχονται από την οπτική απεικόνιση του λειτουργικού συστήματος ρομπότ (ROS) που ονομάζεται RVIZ. Ο χρήστης έχει τη δυνατότητα να επιλέξει διαφορετικές διαδρομές, όπου το ρομπότ μπορεί είτε να κινηθεί και στους δύο ορόφους, είτε σε έναν από αυτούς αποκλειστικά. Η επιλεγμένη διαδρομή που καθιερώθηκε σε αυτή τη διατριβή, χρησιμοποιήθηκε για τη συγκέντρωση των ταξιδιωτικών μετρήσεων και ορίστηκε επίσης για επίσκεψη σε τρεις συγκεκριμένες τοποθεσίες και εντοπισμό ετικετών RFID σε κάθε τοποθεσία. Δύο από αυτές τις θέσεις τοποθετήθηκαν στον πρώτο όροφο, ενώ η άλλη στον δεύτερο. Όταν το ρομπότ πλοηγείται σε αυτές τις τοποθεσίες, αποφεύγοντας τα εμπόδια, πρέπει να εισέλθει στον ανελκυστήρα δύο φορές, πράγμα που οδηγεί σε πραγματικό χρόνο ταξιδιού 19 λεπτών και 11 δευτερολέπτων, καλύπτοντας περίπου συνολική απόσταση 194 μέτρων, με σταθερή ταχύτητα ρομπότ στα 0,25 μέτρα/δευτερόλεπτο. Σε ελεύθερο χώρο, χωρίς απρόβλεπτα εμπόδια, όπως η μετακίνηση ανθρώπων, θα χρειαζόταν 12 λεπτά και 56 δευτερόλεπτα. Ωστόσο, λαμβάνοντας υπόψη τον χρόνο κατά τη διάρκεια χρήσης του ανελκυστήρα (3 λεπτά) και τις διαδικασίες εντοπισμού των ετικετών RFID (2 λεπτά και 45 δευτερόλεπτα), προστίθενται επιπλέον 5 λεπτά και 45 δευτερόλεπτα, δίνοντας συνολικό χρόνο 18 λεπτών και 41 δευτερολέπτων.

Επιβλέπων Διπλωματικής Εργασίας : Καθηγητής Άγγελος Μπλέτσας

Acknowledgments

I would like to express my sincere gratitude to all those who have contributed to the successful completion of this thesis.

First and foremost, I am thankful to my supervisor Prof. Aggelos Bletsas, whose guidance, support, and invaluable insights have been instrumental throughout this journey. Your mentorship and unwavering commitment to my academic growth have been truly inspiring.

I extend my appreciation to the Telecommunication Lab for providing all the necessary equipment for this thesis.

My friends and colleagues from the Telecommunication Lab, especially E. Giannelos, G. Apostolakis and E. Andrianakis for their knowledge, support, and guidance during this thesis.

Table of Contents

List of figures

Chapter 1: Introduction

In the recent years, both individuals and organizations encounter significant challenges on a daily basis for navigation in large spaces. A very common example is experienced in a multileveled large hospitals crowded by people and obstacles. In such scenario, imagine a surgeon doctor who needs a tool or equipment from the other side of the hospital instantly. Either the doctor or another individual has to stop what they are doing and seek to obtain that item. It is not practical to leave behind their current task and walk to other ends of the hospital for a certain item. An additional barrier in this scenario occurs when the person responsible to deliver the item does not hold the knowledge of its exact location. Humans also tend to get distracted and have traits of procrastination, leading to more waste of time and inefficiencies. Another scenario could take place at a hotel where the receptionist is servicing a lot of customers. The receptionist is not able to leave the reception area at most times, where his/her responsibilities lie. Meanwhile, the receptionist needs to check if any guests have left their room in order to inform a housekeeper to clean and decontaminate the room so new guests can enter. Challenges occur when the room is at the far end of the building and at instances where the receptionist cannot reach any housekeepers.

In situations like this, where moving between large areas, crowded by people is not easily accessible by an individual, robots using sensors can be a possible solution. Robots can move and navigate autonomously through large areas, avoid humans and obstacles, and smoothly transition between floors to reach their destination. The primary goal of this thesis is to explore whether the aforementioned scenario can be addressed in an automated way.

The movement of robots across floors in a large area and the autonomous navigation toward defined destinations could prove to be a valuable strategy for achieving this goal. This approach involves the robot's movement on the floor and obstacle avoidance using a LiDAR sensor [1], entering elevators using voice commands and the Astra Camera [2], and also the localization of RFID tags in predefined locations using antennas for asset localization. To address the challenges encountered in the large space areas that are presented in this thesis, the Science Building on TUC Campus was chosen for experimentation. The robot's movement and RFID tag localization were conducted on the ground floor and first floor. The Science Building was selected to simulate the initial problem, bearing resemblance to a hospital or a hotel that is often crowded. Navigation through such extensive spaces, while avoiding humans, is crucial, mirroring real-world scenarios.

In this thesis, the implemented approach for RFID localization was found in [3] and [4]. Chapter 2 provides a comprehensive overview of the utilized hardware and software. Notably, the robot's hardware had already been set up by colleagues in the Telecommunication Lab. Chapter 3 details the algorithm for robot movement, building localization, autonomous navigation, obstacle avoidance, and the utilization of an elevator for floor transition. Chapter 4 explores the RFID tag localization procedure (found in [3, 4]), the integration of autonomous robot movement, and RFID tag localization. It also presents the experimental results of this thesis. Chapter 5 encapsulates the conclusion and outlines future work for improvement.

Chapter 2: Hardware Architecture

Figure 1 : Robot fully Assembled

2.1 Hardware components and Computing Infrastructure

The integration of various hardware components forms the foundation on which the functionalities and capabilities of the robot are being built. This chapter presents an overview of the essential hardware components that compose the core of the robotic system. Those components collectively contribute to the robot's ability of mobility, perception, interaction, communication, and power distribution. The subsequent sections have details for each of those hardware components.

2.1.1 Robot Base

Figure 2 : Front view of the Kobuki base

Figure 3 : Side view of the Kobuki Base

Figure 4 : Kobuki Base Docking Station

An important part of the robot's mobility is the robot base. The design and construction of the robot base have an important role in determining the robot's stability, maneuverability, and adaptability to diverse environments. The robot base being used is the Kobuki Mobile Base [5], which is categorized as a ground robot base. Kobuki Mobile Base is made by Yujin Robot. The base is powered by a 4S2P Lithium-Ion battery with a voltage of 14.8V and a capacity of 4400mAh. According to manufacturer specifications, the robot's battery can function up to 7 hours and the battery needs 1.5 – 2.6 hours to be fully charged according to manufacturer specifications. The maximum payload can be up to 5 kg in hard ground.

2.1.2 Sensors

Sensors detect the objects that surround them. Each sensor captures specific types of data. These include cameras for visual input, LIDAR [1] for precise distance measurement, localization, map creation, and obstacle avoidance. The second camera being used is the Astra Camera [2], for obstacle avoidance in specific tasks (doors opening or closing).

2.2.1 Hokuyo UST-20LX LiDAR

Figure 5 : Hokuyo UST-20LX LiDAR

This sensor device is based on Light Detection and Ranging (LiDAR) technology [1] (see Figure 5). It is a compact, lightweight 2D laser scanner used for obstacle detection, localization on autonomous robots, and automated guided vehicles. It is equipped with an Ethernet Interface, and it can obtain high-speed and accurate measurement data in a 270 degrees field of view, for a distance detection range from 0.06 meters up to 20 meters.

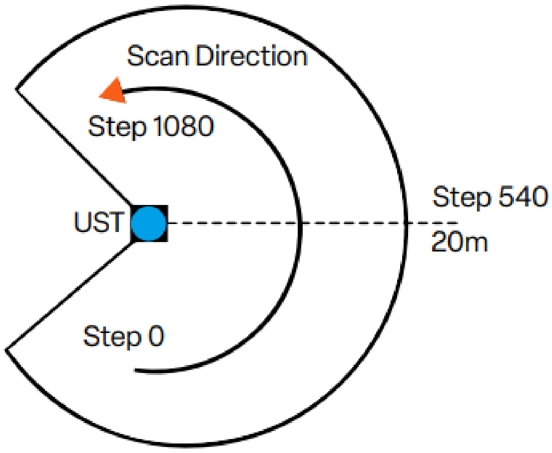

Figure 6 : LiDAR Scanning Image

Supply voltage operation range is from 10V to 30V within 10% ripple. It needs 450mA on start-up and 150mA during operations. Due to its decreased power consumption, this sensor is suitable for battery-operated platforms. So, it can be safe to be powered from the Kobuki Mobile Base [5]. It is connected to the computer via the Ethernet port. The default Ethernet settings IP number is: 192.168.0.10 and the port number: 10940

Figure 6 (see [6]) shows the Laser scanning image including 1081 measuring steps, 270 degrees detection angle for its field of view, and 0.25 degrees angular resolution.

2.2.2 Astra Camera

Figure 7 : Astra Camera

This device is a powerful and reliable 3D camera which includes the proprietary Orbbec 3D microchip and VGA color. Astra 3D cameras are excellent for a wide range of scenarios, including gesture control, interactive systems, robotics, retail, 3D scanning, and point cloud development.

It contains an RGB Image Sensor and Depth Image cameras. Depth cameras can represent obstacles that are closer to the camera with darker representation while the objects that are further are projected lighter accordingly. Both RGB and Depth camera have a resolution of 640 x 480 pixels with 30 FPS, a horizontal Field of View of 60 degrees, and a vertical Field of View of 49.5 degrees (also 73 degrees diagonal). The distance range of the camera ranges 0.6 up to 8.0 meters (with 0.6 – 5.0 meters optimal distance). Lastly, the camera is connected to the computer via USB port.

2.1.3 Actuators

Actuators provide the robot's physical movement and interaction with the environment. The Kobuki's mechanisms have motors that drive the four wheels and allow precise rotation movements. These actuators are fundamental in comprehending the robot's desired actions based on the information gathered from its sensors. The two wheels that are located on each side are for movement and the remaining two wheels that are located on the front and back are for the rotation of the robot. The maximum velocity of this robot is 70 cm/s and the maximum rotation is 180 deg/s [7]. The velocity used for this experiment is set to 25 cm/s.

2.1.4 Communication Interfaces

Figure 8 : Network Switch

Communication between hardware components has one of the most important key roles for communication interfaces. Various hardware components such as USB, Ethernet, network switch (see Figure 8), are interconnected with each other so that data exchange, and coordination among the components can be established. Voice commands [8] were integrated into the system to instruct individuals on calling the elevator and subsequently choosing the desired floor. These interfaces ensure that the different parts of the robot work in harmony to achieve the requested tasks.

2.1.5 Power Management

Power management components are responsible for efficiently distributing power to the various hardware elements. This also includes regulating voltage levels, managing battery charging, and ensuring consistent power supply to critical components. Battery is used to power up NUC computer and other hardware components connected to NUC.

2.1.6 Computing Infrastructure: NUC Computer

Figure 9 : NUC front view

Figure 10 : NUC back view

The integration of the functioning of the robotic system is the utilization of a Next Unit of Computing (NUC) computer. The NUC computer serves as the brain of the robot, providing computational power for sensor data processing, execution of control algorithms, and facilitating communication of the various hardware components. NUC's capabilities and small dimensions make it an ideal choice for real-time decision-making and controlling tasks within the robotic system. Through this thesis, the role of NUC is its interaction with other hardware components.

2.1.6.1 Remote Control Connection

While operating the robot, it was essential to visualize the on-screen results. To facilitate this, a Remote Desktop Application (Any Desk) was employed to connect the NUC (2.1.6) to a computer.

2.1.7 Antennas, RFID Tag reader and RFID Tags

Apart from the core components, the robotic system includes antennas and RFID (Radio Frequency Identification) tags as integral elements.

2.1.7.1 RFID Tag Reader

Figure 11 : RFID Reader Speedway R420

An RFID tag reader was mandatory to use in the experiments for connecting and use multiple antennas. This reader is small factor UHF Gen2 RFID tag reader with a maximum transmit power of 30dbm and supports up to 4 antennas. The reader (see Figure 11) was deployed. In this reader two antennas were implemented and used for the tag localization procedure.

2.1.7.2 Antenna FlexiRay SF-2110

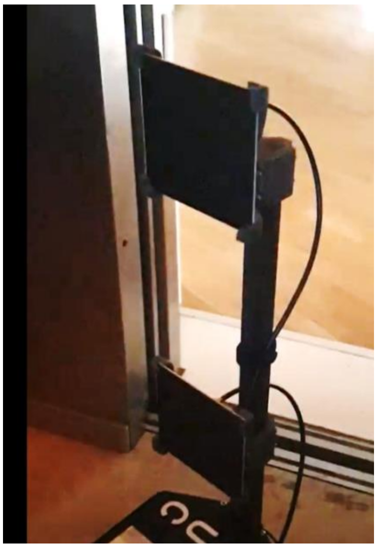

Figure 13 : Antennas on the robot

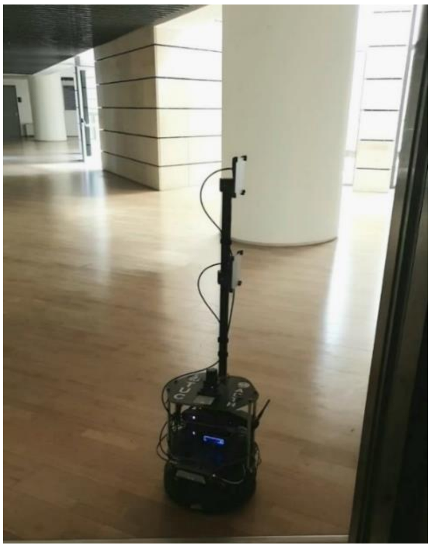

Figure 12 : Antennas on the robot 2

Antennas are responsible for the wireless communication, facilitating and exchanging data between the robot and the external devices or systems, such as RFID tags. Two FlexiRay antennas 5dBi each (see Figure 13 and Figure 12), are mounted on the top of the robot in order for the RFID tag reader to work. Their height is 1.05 meters, and 1.27 meters. The two antennas are facing 90 degrees from the robot's facade.

2.1.7.3 RFID Tags

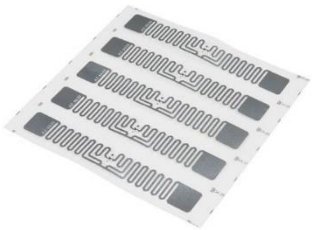

Figure 14 : RFID tags

Tags are small devices that can store and transmit information using radio waves to a reader. RFID tags are frequently used for various telecommunication devices which are used to track vehicles, pets, storage items, objects, and even patients with disease. There are three main RFID bands: Low Frequency (120kHz-150kHz), High Frequency (13.56 MHz), and Ultra High Frequency (865 MHz to 868 MHz in Europe). Our main emphasis is to focus across Ultra High Frequency RFID tags. Tags are inactive devices and are only powered by electromagnetic waves that are transmitted from the Reader of an antenna. These devices do not emit any signal, they just absorb the incoming signal from the antenna so that they can encode the necessary data. Every tag has an integrated unique code, called Electronic Product Code (EPC) which assists the robot through the antennas to identify their location.

2.1.8 Other Hardware

The experiment has been enhanced with the inclusion of additional devices to ensure security and ease of use. Devices such as a Wireless Adapter for improved connectivity and a Wireless Keyboard for safety and connectivity reasons are incorporated.

2.1.8.1 TP-Link Wireless USB Adapter

Figure 15 : TP-Link Wireless USB Adapter

An additional wireless adapter was employed as a precaution in the event of a disconnection of the NUC computer during the transition between floors.

2.1.8.2 Savio Wireless Keyboard

Figure 16 : Savio Wireless Keyboard

This is a KW-03 wireless keyboard. It is a compact, multifunctional device that combines the features of a full-size keyboard and a touchpad. It is mainly used during the map creation phase. The robot moves manually. By using the up/down arrow buttons the robot increases and decreases its speed, and by pressing the left/right arrows it rotates. The building has two wing-shaped areas, and a lot of walls in the center of it. There is a possibility that the connection might get interrupted. In the worst scenario, if a command to increase the velocity is executed, and the signal is lost, the user will lose control of the robot, and the collision will cause severe damage. This is why the Wireless keyboard is connected straight to the NUC computer to prevent such incidents from occurring.

2.1.9 Robot Overview

Figure 17 : The final robot structure used for the experiments

After installing all the necessary devices, managing cables correctly, and connecting the antennas with RFID reader, the final robot structure is presented in the Figure 17.

2.2 Visualization tool RVIZ (ROS Visualization)

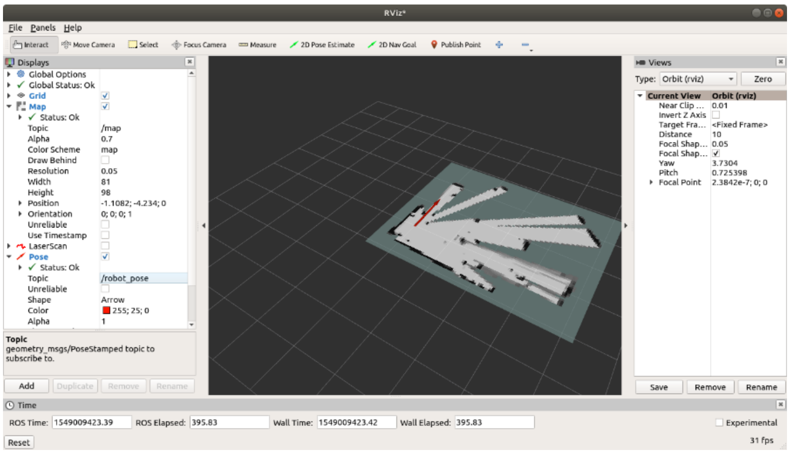

Figure 18 : RVIZ Visual while creating the map

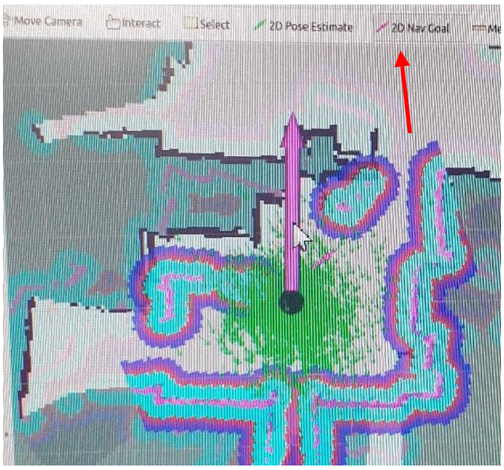

Figure 19 : 2D Nav Goal: Set the robot's navigation goal and orientation

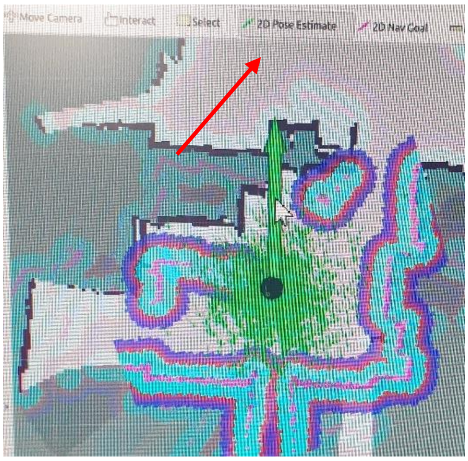

Figure 20 : 2D Pose Estimate – Set the robot's initial position and orientation

RVIZ [9] translates complex map data into an intelligible visual format. RVIZ is serving as a 3D visualization software tool for robots, sensors, and algorithms. RVIZ bestows the capability to render the robot's perceptual understanding of its surroundings and its world. The main purpose of RVIZ is to provide a means to visually apprehend the robot's configuration, obstacles, localization, and related aspects. This visualization unfolds in real-time, impacting immediate insights into the robot's operational status. In this experiment, RVIZ is mainly used for visualizing the robot's movement in the building. To complete that, the initial pose and final destination have to be set on the RVIZ interface for the robot's manual traversal.

Figure 19 presents the 2D Pose Estimate button. After the AMCL package installation, this button can manually set the robot's position and orientation on the map. Figure 20 presents the 2D Pose Estimate button. After the AMCL package installation as well, this button can manually set the robot's destination and orientation on the map.

In further use, the robot's movement becomes autonomous. To achieve that, coordinates need to be outputted allowing the robot can be aware of the destination. The "Publish Point" button (see Figure 18) can be very useful for this purpose; by pressing this button and selecting the coordinates on the map, the robot can be aware of the destination to achieve autonomous movement.

Chapter 3: Robot floor movement and obstacle avoidance

As mentioned in the introduction section, the focal objective of this thesis lies in the exploration of robot movement across various floors, the ground and, the first floors, encompassing the unique challenge of floor transition. For navigation across the floors, the robot is designed to use the elevator. This is facilitated by human intervention in the form of pressing elevator buttons, both internally and externally of the elevator, all guided through voice commands [8]. It is important to mention that the robot's movement is predetermined from a registered map that was created through algorithmic techniques, thereby offering a familiar trajectory. More importantly, those maps are distinct for each floor and can adapt to different changes as the circumstances necessitate. For the second part of this chapter, our attention turns to the unique technique of obstacle avoidance. While the robot navigates within these established maps, it may encounter unforeseen obstacles, including humans, waste bins, and more. In this scenario, the concept of obstacle avoidance is being triggered and starts moving around the obstacles. This dictates that the robot recreates its course toward its intended destination, maintaining its functionality and purpose.

3.1 Map creation using Gmapping algorithm

For the advancement of robot movement across floors, it is imperative that the robot possesses an accurate awareness of its navigational path. In achieving this objective, the creation of tailored maps for each floor assumes paramount significance. This necessitates the application of a precise algorithm to facilitate map generation, including Simultaneous Localization and Mapping (SLAM) [10]. SLAM is the computational problem of updating a map of an unknown environment while keeping the location of the robot. There are a lot of popular solutions, of which gmapping and ROS-cartographer algorithms are the most popular. In this context, for the SLAM [10] problem, the gmapping algorithm [11] emerges as an indispensable tool. The gmapping package provides a laser-based SLAM. This establishes a comprehensive understanding of the robot's spatial environment. By using LiDAR technology, the gmapping algorithm improves the creation of the map. This map amalgamates laser-derived data and poses information collected by the robot's movements. Notably, the robot's movement throughout the floor precipitates the map's gradual emergence. In this part, manually moving the robot across the floor serves the map generation. For this part, it is important to ensure the accuracy and quality of the map results in real-time scrutiny for better map results. To fulfill this requirement, the visualization tool known as RVIZ is employed. Once the map generation is completed the map is preserved for subsequent use during the robot's traversal across the floor.

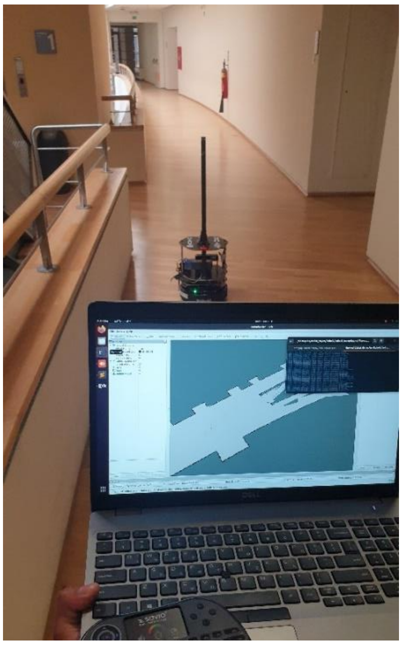

Figure 21 : Map Creation Process for the First Floor of the Science Building

For the map creation the hardware used for this purpose are the wireless keyboard mentioned on 2.1.8.2, a laptop, and Remote Desktop Application using AnyDesk Software [12] connection (see Figure 21).

3.2 Robot Floor Movement

When the map generation is fulfilled, the robot movement becomes feasible. A prerequisite for this stage is to have the integration of an additional software package that is dedicated to the localization of the robot's position on the map. The AMCL [13] package emerges as a probabilistic localization system that can track precisely the robot's position about to a known map. Furthermore, the AMCL package provides the initial position and orientation of the robot on each floor, including the ground and the first floor. As previously mentioned, the RVIZ visualization tool is an important asset in this process. Through RVIZ, the robot's position within the map can be visualized, serving as an initial reference point. As for the instances where the robot's initial position or orientation is different from its default, manual intervention is mandatory. This utilization of the RVIZ 2D Pose Estimate button can initialize the robot's pose manually as mentioned in 2.2. Additionally, it helps fix the robot's position and orientation (see also Figure 20). It is worth noting that even if the initial position of the robot is not fully accurate, the robot's subsequent behavior for localization entails self-adjustment. Upon receiving a command to navigate to a specific destination within the map (see Figure 19), the robot recalibrates its position and orientation, aligning it accurately with the map's coordinates.

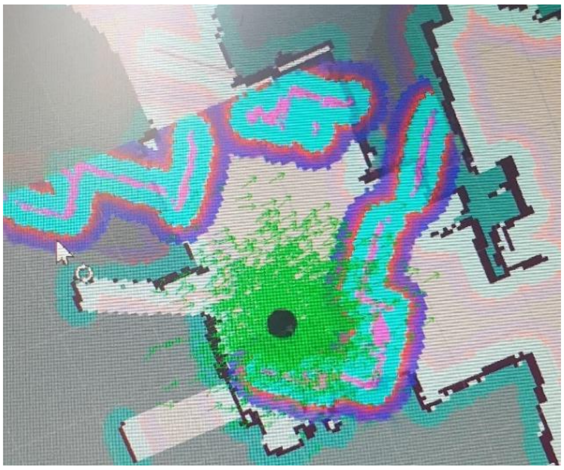

Figure 22 : Robot's wrong initial pose in the Telecommunication Lab area

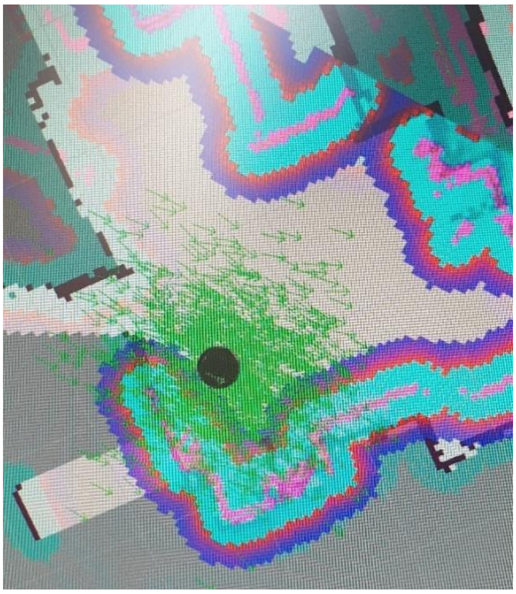

Figure 23 : Robot's self-adjustment behavior

In Figure 22, the location of the robot does not align with its existing position. Without manual intervention, utilizing the initial button from RVIZ (see Figure 20), and with the first command (see Figure 19), the robot rotates 360 degrees and aligns itself with the correct location and orientation on the map. It adapts to the already existing obstacles, such as walls (see Figure 23). With the AMCL package, this behavior can be adjusted in scenarios where the robot loses its alignment while moving and needs re-localization. Additionally, manual intervention using the Pose Estimate button (Figure 20), can solve the initialization problem.

Furthermore, the recalibration procedure transpires through the RVIZ interface, where the 2D Nav Goal button is essential (see Figure 19). Through this button, the robot's destination point and orientation are precisely designated. With this command's initiation, the robot formulates a navigation pathway, recognizing and avoiding obstacles en route to its destination. The package responsible for creating and adjusting navigation pathways is the AMCL. During the phase of the robot movement, it remains feasible to fine-tune the robot's operational parameters. Those parameters can make velocity adjustments, proximity thresholds concerning known obstacles, and preemptive strategies to address any potential collision scenarios through the utilization of the bumper device [14] integrated within the Kobuki robot base [5]. The robot can fail to reach the destination point if many collisions occur and there is an inability to create a new path.

Figure 24 : Robot first move on the Ground Floor of the Science Building

In summary, the orchestration of the robot's movement necessitates the synergy between robust localization software, such as AMCL, and the interactive interface provided by RVIZ. This orchestration culminates in a cohesive and efficient navigational system, which functions through adaptable recalibrations once the robot is launched. In essence, if the initial position of the AMCL package differs from the robot's location, manual intervention is required by using the Pose Estimate button (see Figure 20 :).

The spontaneous final destination can be reached through the Nav Goal button (see Figure 19) on each floor where the robot is located. It should be noted that a new destination to be executed, the prior one must first be fulfilled. The floor transition is analyzed in Chapter 3.6.

3.3 Potential collision

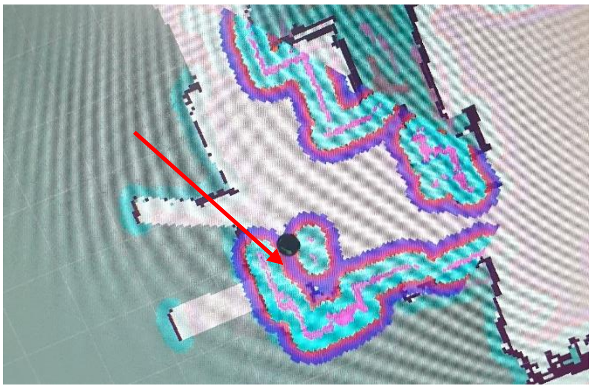

Figure 25 : Collision Visualized on RVIZ

In the event of a collision, the robot engages in a rotate recovery protocol [15], where it attempts to clear space by performing a 360-degree rotation to effectively circumvent the obstacle that has impeded its path. If the encountered obstacle is immovable, the robot seamlessly transitions to an alternative recovery behavior explicitly for obstacle avoidance defined in the AMCL package.

3.4 Obstacle avoidance

This section is dedicated to the implementation of obstacle avoidance mechanisms designed to address entities not accounted for in the pre-existing map. It is important to emphasize that this does not necessarily need the installation of additional software packages but instead entails a configuration process within an already established package. The AMCL package assumes the pivotal role of managing obstacle avoidance procedures.

To enable effective obstacle avoidance, the selection of an appropriate sensor is essential. In this context, the LiDAR [1] sensor has been chosen as the hardware for this purpose. Also, it is worth noting that an Astra Camera [2] can serve as an alternative sensor for the same purpose, but the camera serves for entering or exiting the elevator. The basic concept of this approach revolves around the robot's movement within its environment. When the robot traverses a floor, and a human or any other object obstructs its path and remains stationary, the robot initiates the process of charting a new course of path. However, if the obstruction entity departs from the robot's trajectory, the robot adjusts its velocity based on the proximity of the departing entity. Subsequently, the robot resumes its pre-established path without necessitating the creation of a new trajectory.

In-depth, the "costmap_common_params.yaml" configuration file within the AMCL package regulates the intricacies of the obstacle avoidance algorithm. This file hosts numerous parameters that allow the fine-tuning of the algorithm to align with specific operational

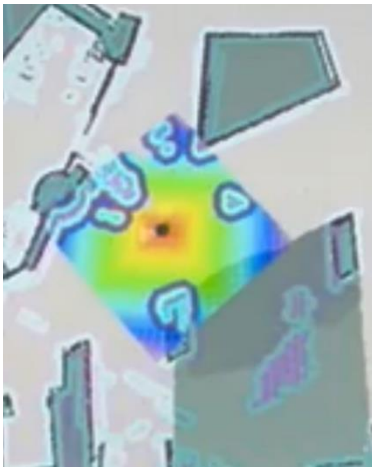

Figure 27 : RVIZ Visualization Part 1

Figure 26 : Third-person Visualization Part 1

Figure 29 : RVIZ Visualization Part 2

Figure 28 : Third-person Visualization Part 2

requirements. Based on the robot's movement within expansive and open environments, it was necessary to give exclusive attention to adjusting the "obstacle range" parameter. This parameter is the range in which the LiDAR detects the obstacles and executes avoidance maneuvers. It was also configured to an extended value beyond the default setting. This value is set to 5.5 meters. It is also essential to underscore the LiDAR sensor's "height" parameter, which serves a critical function of obstacle detection. This parameter has "minimum" and "maximum" values. The minimum threshold is constrained within the range of 55 cm to align with the LiDAR's physical positioning on the robot. Conversely, the maximum height value is adjusted to enable the sensor to accurately detect and respond to human presence while in motion. This value is set to 1.60 meters.

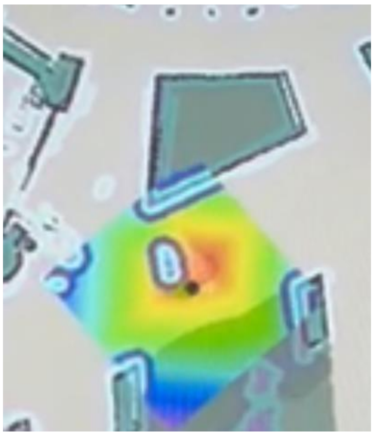

Figure 30 : RVIZ Visualization Part 3

Figure 31 : Third-person Visualization Part 3

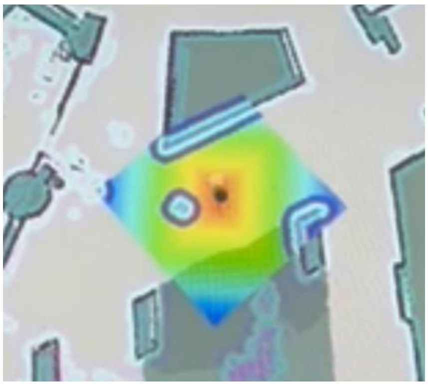

Figure 33 : RVIZ Visualization Part 4

Figure 32 : Third-person Visualization Part 4

Upon real-time operations, the robot's successful obstacle avoidance and LiDAR functionality can be observed through the RVIZ tool, ensuring the system's robustness and effectiveness. This comprehensive approach culminates in an operational paradigm, by parameter adjustments and sensor-specific considerations, thereby enhancing the robot's capacity to navigate unforeseen obstacles adeptly.

The robot visualizes the obstacle that is not in the predefined map. The obstacle obstructs the robot's created path, resulting in the creation of a new path, avoiding the obstacle. The figures (Figure 27, Figure 26) present this procedure. In Figure 29 and Figure 28 the robot starts avoiding the obstacle. It has a new path to arrive at the defined destination avoiding the obstacle and using the fastest path provided. In Figure 30 and Figure 31 the robot has avoided the obstacle and is moving to the already-defined destination. Finally, Figure 33 and Figure 32 show the robot's arrival at its destination. It also corrects its orientation. The obstacle avoidance procedure was executed successfully and stands for further commands.

3.5 Autonomous movement

While the concepts discussed thus far are promising, a true objective lies in automating robot movement. To accomplish this goal, the initial step is defining specific coordinates and orientations for the preferred destinations within the floor space. This process is contingent upon using the RVIZ tool in conjunction with the designated floor map.

The subsequent action involves automatically issuing movement commands to the robot with the coordinates of the destinations. This functionality has been implemented through the use of Python code. This code incorporates a dedicated function responsible for retrieving the coordinates and the orientation information to the robot's final destination on the map. Additionally, the created Python code publishes the coordinates of the destinations to the map, which creates the path that becomes visualized from the RVIZ tool.

As mentioned above, the robot can avoid obstacles during its course. However, if there are multiple or large obstacles there is a possibility that they will prevent it from moving. If this happens, then it will automatically seek to find a new path. In case the new path takes longer than expected, or leads to an endless cycle, then an embedded timer will force it to stop. This timer is coded to start when the robot begins to move for the first time. If the robot moves for a longer time than the one that has been set on the timer then the robot will remain stationary. When this happens, then a pre-arranged message has been set for its re-initialization. This applies to the other floor as well.

As a final point, autonomous movement requires manual intervention for its orienting and positioning stage, which is its initiating phase. Although there is a direct correlation with Chapter 3.2, and there is a critical difference that ought to be mentioned. In this chapter, all the coordinates of the locations are implanted in the Python code and can be used automatically. On the other hand, Chapter 3.2 mentions that every location needs to be added manually.

3.6 Navigation in and out the Elevator

Using the elevator is essential for the robot to move between floors. On each floor, the robot must be positioned precisely outside the elevator, facing the elevator doors. This is necessary because the robot's camera needs a clear view of the elevator's doors and must wait for them to open before entering. To achieve this, the RVIZ tool is used along with predetermined coordinates for the robot's position outside the elevator doors.

The next step involves the use of the Astra Camera [2] which is responsible for the doors of the elevator. By default, the camera covers a wide range which has to be configured so that the robot can enter the elevator. With this being said, the camera is adjusted so that it aims solely at the doors of the elevator. When the Astra Camera recognizes that the distance between the robot and the doors is approximately one meter, then the robot is programmed to remain still. During this time, voice commands are executed from the speakers, requesting for an individual to summon the elevator. After the doors are open, the camera will identify the increase in the distance (between the robot and the elevator's wall), therefore enabling the robot to enter the elevator. After the robot has entered the elevator, it automatically rotates 180 degrees so that it faces the doors of the elevator and requests the destined floor through voice commands. Ultimately, the elevator will reach the selected floor, and as soon as the doors open, the camera will detect the increase in the distance and thus will bring an end to this phase.

Naturally, each floor differs from one to another and the maps were created accordingly and have been placed within the AMCL file package. The switch of the maps happens during the floor transition. This is established through the Python script. The script terminates all processes related to the previous floor and launches all other processes related to the new floor. The coordinates and orientation that are related to the new AMCL file package of the corresponding floor must face the door of the elevator so that it matches the real location of the robot by the maps.

In cases where the elevator remains on the same floor that the robot entered, RFID tags are utilized to verify that it is the correct floor. If the robot detects an incorrect floor, voice commands are executed to request the appropriate button. As explained in Chapter 2.1.7.2, RFID tags have a unique code for identification. In this scenario, two separate RFID tags are placed on the elevator's doors, one for each floor, to ensure floor correctness before the robot continues its journey to the predetermined destination.

Figure 35 and Figure 34 display the RFID tags on the doors of the elevator for the first (white tag) and ground floor (black tag) accordingly. While the robot is facing the doors, the antennas are facing toward the left side of the elevator, aligned with the robot's pose. This 90-degree offset is sufficient to locate the RFID tag, confirm the floor, and proceed to the destination. Otherwise, it will initiate the voice commands so that someone will press the correct button.

Figure 35 : Elevator Door on the Ground Floor with an RFID tag

Figure 34 : Elevator Door on the First Floor with an RFID tag

Figure 36 : Slightly pushing the robot's base after the voice command

It's worth noting that the elevator doors may not always align perfectly with the floor level. During testing, there was observation of some variation in this alignment. As a result, there was an extra voice command as a precaution. Before the robot commences its exit from the elevator, a voice prompt is issued, saying, "Please help me get out of the elevator, with a gentle push". The action depicted in Figure 36 may be unnecessary if the elevator is perfectly leveled with the floor.

Chapter 4: Integration of RFID localization and Results

This chapter discusses the application of RFID localization algorithms to track RFID tags within the building's environment. While the robot moves a straight path, around 5 meters in length, it utilizes its antennas to read and receive responses from the RFID tags. By considering factors like the robot's movement, antenna distance, and antenna height, the algorithm accurately determines the RFID tags' location within a specific area. It's important to note that this process is limited to specific locations within the building. To achieve precise positioning, the RVIZ tool is employed for visualizing the outcomes. The algorithm operated effectively within an area of approximately 2x2x1 meters, which is marked on the map. Within this area, the algorithm can identify RFID tags with high accuracy. The tags that are placed on the wall must be placed within a maximum distance of approximately 2 meters from the robot and at a height ranging from 0.90 meters to 1.90 meters.

4.1 About the Algorithm

Figure 39 : RFID Tags placed on the Ground Floor

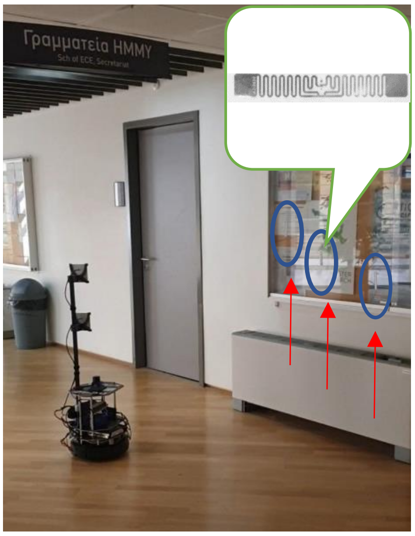

Figure 37 : RFID Tags placed on the First Floor - Office area

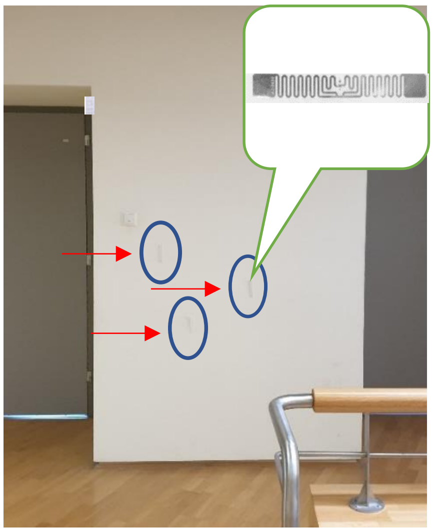

Figure 38 : RFID Tags placed on the First Floor – Secretary's area

The algorithm functions with phase measurements for RFID tag Localization by using particle filters [3], [4]. In this thesis, two FlexiRay SF-2110 5 dBi antennas are utilized. The localization procedure is as follows: the robot is placed approximately 2.5 meters behind the position of the RFID tags. This is the time when the RFID reader (see 2.1.7.1) is launched among the localization algorithm [3], [4]. The robot starts moving in a straight line for about 5 meters (2,5 meters after the RFID tags' position) and by the time it arrives at the destination, the RFID tag reader is terminated while the localization algorithm continues for another 15-20 seconds and then is also terminated.

The figures (Figure 39, Figure 37, and Figure 38) show the location where RFID tags are placed, and the robot can move in that location. Afterward, the initial RFID localization procedure begins and localizes them. To achieve the most accurate results, the robot has to maintain its movement in a straight line. The robot's velocity is 0.25 m/s.

4.2 Integration

In the integration section, various components are specified below. They are also discussed in a demonstrative manner throughout the subsequent chapters. Specifically, this chapter reviews the successful integration of the moving robot, the capability of autonomous floor navigation, obstacle avoidance, and RFID tag localization across three specified positions (see Figure 39, Figure 37, and Figure 38 for the exact tag locations).

The subsequent chapter will be dedicated to presenting visual figures that encapsulate the outcomes and findings from each chapter of this thesis. The following demonstration aims to clarify the core themes and accomplishments of this work.

4.3 Results

The primary focus of this thesis is to achieve the movement of the robot with floor transition, obstacle avoidance while moving, and localization of RFID tags within the robot's environment. This section presents the results obtained through the methodology used.

4.3.1 SLAM Gmapping algorithm

The initial phase of map creation faced challenges related to connectivity issues. However, after overcoming these initial hurdles, the map creation process was successful. The sensor used for this purpose is LiDAR, positioned on the top of the robot, at an approximate height of 55 cm.

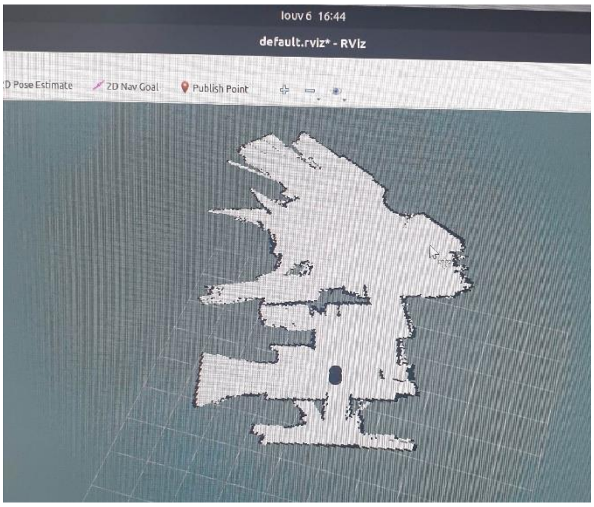

Figure 40 : Created Map in the Telecommunication Lab

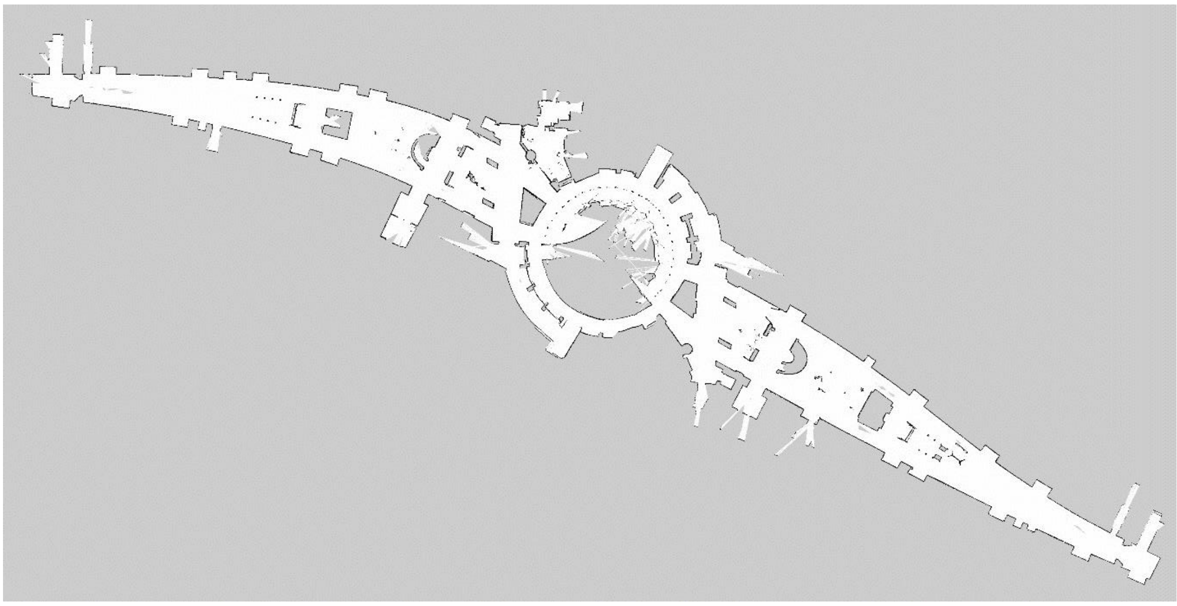

Figure 41 : Ground Floor Map in the Science Building

To establish the movement of the robot in the building, creating a map was necessary. Launching all required nodes for the robot's movement and using the gmapping algorithm for map creation (see 3.1), the first experimental results are shown in the figures.

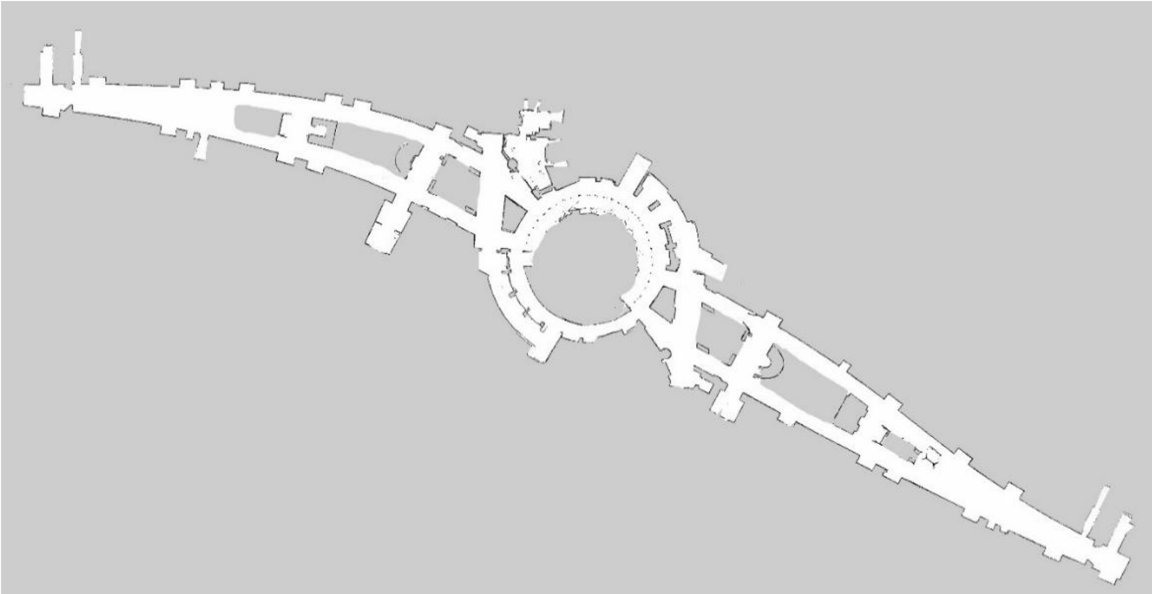

Figure 42 : Ground Floor Final Map in the Science Building – After the photoshop editing

The initial results from the Telecommunication Lab's area, where the map created was accurate and satisfactory (see Figure 40). To ensure the correctness of the map, commands for moving the robot were executed to observe its behavior in the lab area. After several commands, it was confirmed that the map creation was successful. After obtaining these results, the gmapping algorithm was applied to a larger area, specifically the Ground Floor of the Science Building. The preliminary outcomes from the creation of the Ground floor map using gmapping algorithm are presented in Figure 41. The parameters for the gmapping algorithm were kept consistent with those in the smaller-scale area.

An important issue is the sensor's inability to scan and project areas or obstacles beneath its field of vision. Consequently, the robot's perception erroneously categorizes certain areas as passable, whereas in reality, the vision is obstructed by lower obstacles like stairs, occasional elevator doors, and other lower barriers that may cause severe damage to the robot. For the sake of this experiment, all these regions have been named as "forbidden areas". To address this challenge, image processing that includes image Photoshop was undertaken. Essentially, these "forbidden areas" were displayed as "unknown areas" to prevent the robot from attempting navigation through them. The unknown areas represent regions the robot did not happen to explore during the map creation procedure and are depicted in gray on the map (see Figure 41). There was an attempt to prevent the robot from moving in some specific areas. This was completed by photoshopping the map in black to look like there were extra walls that did not exist. The robot identifies the color black as walls, meaning that it cannot go through. However, since the given map does not appear in real-time, it brings disruption to its localization algorithm. A more efficient strategy was adopted, wherein the "forbidden areas" were finally visually marked in gray, signifying areas the robot hadn't visited during the gmapping phase. This adjustment proved to be successful. Figure 42 shows the Ground Floor of Science's Building after the Photoshop editing.

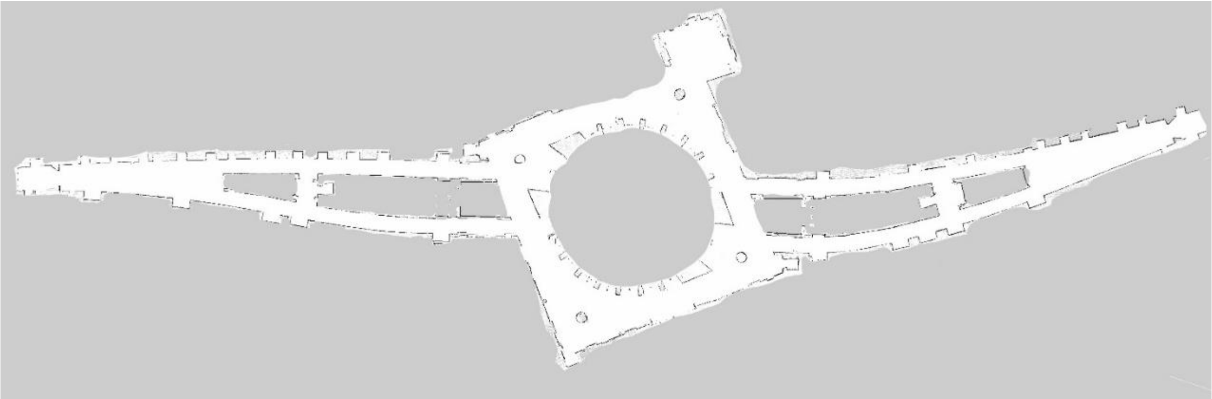

Figure 43 : First Floor Final Map in the Science Building - After editing

After editing the map, it was crucial to assess the robot's ability to move and navigate across the floor. The robot underwent testing in the "unknown areas" by executing commands to ensure the path did not traverse these zones (where some of them had stairs), which could cause serious damage. After successful navigation, this approach proved to be effective. The same methodology was applied for creating the map of the First Floor of the Science Building (see Figure 43). It is worth noting that there was an area where the robot could move, but not with the Kobuki wheels. The nature of the ground determines if the robot will move or not. Again, the area was edited to prevent the robot from getting stuck.

After creating the map for each floor, it was necessary to save the map for future use. To accomplish this, maps were saved using a map server package [16], specifically designed to support gmapping algorithm.

4.3.2 Loop Closure

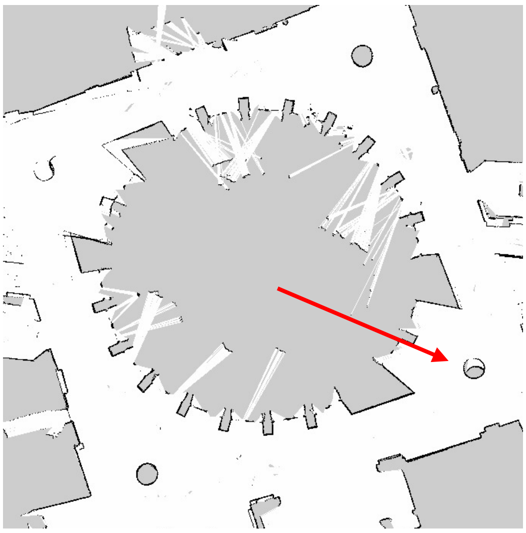

Figure 45 : Loop Closure Example of the Map (Before)

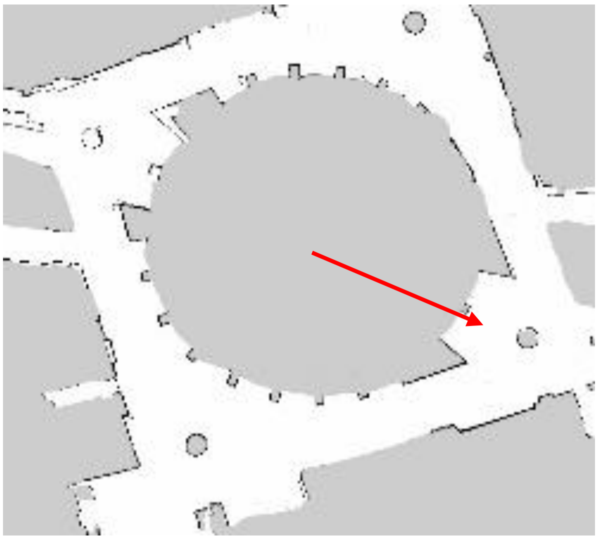

Figure 44 : Loop Closure Example of the Map (After)

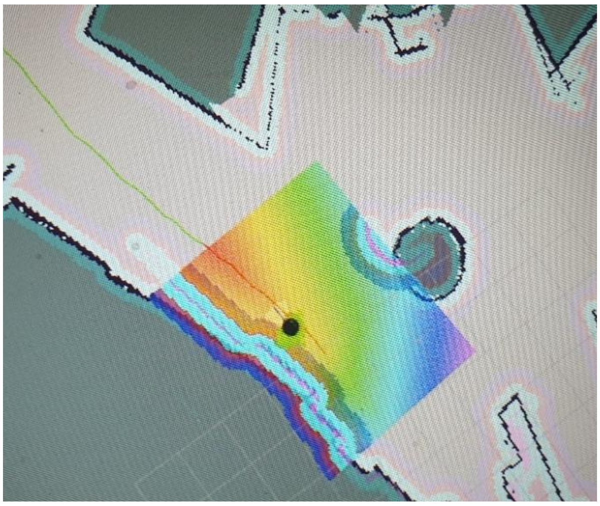

Figure 46 : Loop Closure Error while moving

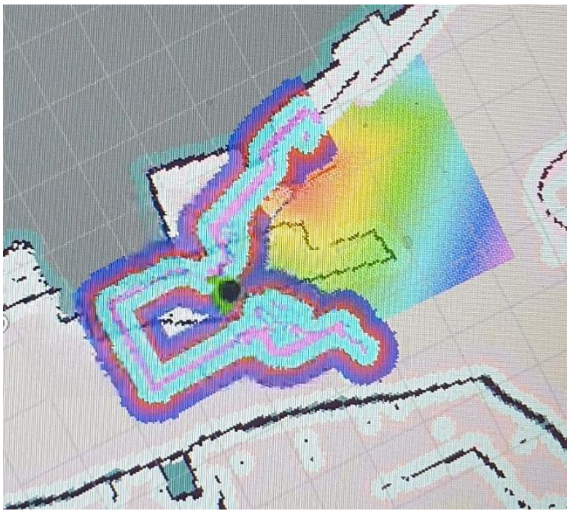

Figure 47 : Loop Closure Error while initializing on the First Floor

After creating the maps, errors became apparent. This was initially observed from the map's visual representation (see Figure 45) and was later confirmed by the robot's localization (see Figure 45, Figure 46). As these figures illustrate, the map was not accurately created, indicating a Loop Closure error. Loop closure is crucial, as it helps the robot recognize previously visited locations during the map creation. It has also allowed it to update its position as well as the map. Loop Closure error, in this case, is the robot's inability to recognize that an object or an area has been visited before. Figure 45, Figure 46, and Figure 47 reveal inaccuracies in the map's visual representation, especially in environments where the robot moves in circular paths in large areas. During map creation, the robot failed to acknowledge that certain areas had already been visited. In Figure 45 the map picture appears to have an error regarding the creation.

Despite these map creation issues, the robot was tested to examine if it could move on the floor without encountering errors and to verify the existence of any map creation issues. While moving through the area, the robot did not recognize the error on the map and continued to move without correcting its localization (see Figure 46). In a second attempt, the map creation was improved, in comparison to the previous one (see Figure 45) but was still not perfect. During the testing of the map, the robot's initial pose was entirely incorrect (see Figure 47), and the robot couldn't exit the elevator. To rectify this error, the parameters on the gmapping algorithm were adjusted to accommodate a larger map creation area. The map was then recreated, and this time, the results were accurate (see Figure 44)

There is a possibility of encountering a similar error during the run-time, but using the correct parameters for the map creation, the localization algorithm (AMCL) can adjust the robot's pose on the map and accurately localize it while moving.

4.3.3 Demo Explanation

Utilizing Python code, the primary demonstration involves the robot's movement, obstacle avoidance, and the reading and localizing of RFID tags in specific locations. More precisely, the program features a selection menu, as provided below.

Option menu:

1. Office
2. Cafeteria
3. Random location floor 1
4. Secretary
5. Lab
6. Random location floor 0
7. Route 1 - office - secretary - cafeteria – office
8. Route 2 - office – rand Loc. floor 0 - secretary – office
9. Disable RFID localization
10. Enable RFID localization

With these options, the robot can move to any location from 1 to 6 and also choose whether it will localize the RFID tags or not by enabling RFID localization in option 10. Choices 7 and 8 will always localize RFID tags for the supported locations, which include the office, Secretary, and a random location on the ground floor.

4.3.4 Detailed Demo for Route 2 option: Office – Random location floor 0 – Secretary - Office

Figure 48 : Starting Point - Office area

Figure 49 : Route journey started – Office area

As mentioned earlier, this demonstration involves the robot's movement within the building, transitioning between floors, avoiding obstacles, and localizing RFID tags in predefined locations. There are three locations with RFID tag localization, all included in this demonstration. Additionally, this demo requires only a single click for the menu selection; the rest involves visualization and pressing elevator buttons.

It's important to maintain a low velocity during this process and not increase it. The speed is kept low due to ground obstacles, such as cables on the floor, which could negatively impact the robot's movement and potentially cause damage (see Figure 72).

To begin with Route 2, the robot can be anywhere in the building, and it will move to the office's location as its starting location (see Figure 48). As the robot arrives, the route begins (see Figure 49). The first location is on the ground floor, so the robot starts moving through the elevator (see Figure 50, Figure 51). Figure 48 and Figure 49 show the robot's arrival at the office's location and the immediate initiation of the route journey. Figure 50 and Figure 51 show the robot still moving through the elevator door, as seen on the RVIZ tool. As the robot moves, it can detect obstacles such as walls, which perfectly align with the pre-existed walls (during the map creation). This is a positive result because gmapping algorithm and localization are working correctly. As mentioned before, any obstacle not predefined on the existing maps, observed along the robot's path, will result in new path creation.

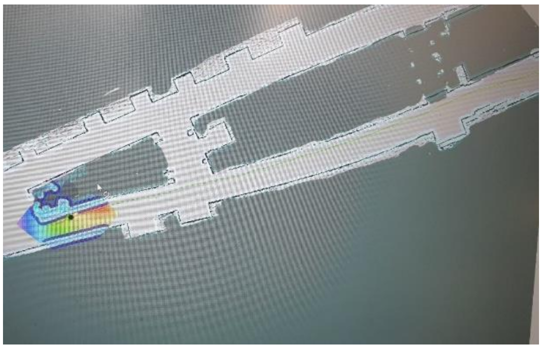

Figure 50 : Robot Movement

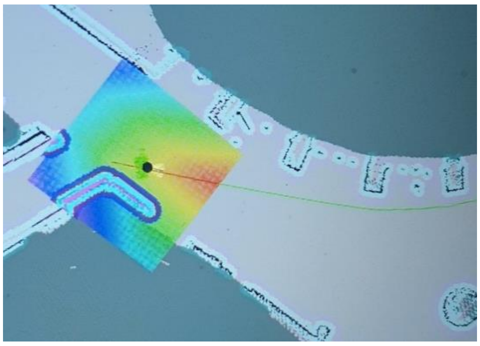

Figure 51 : Robot movement 2

Robot moving from the Office location to the elevator's doors

Figure 52 : Robot enters the elevator

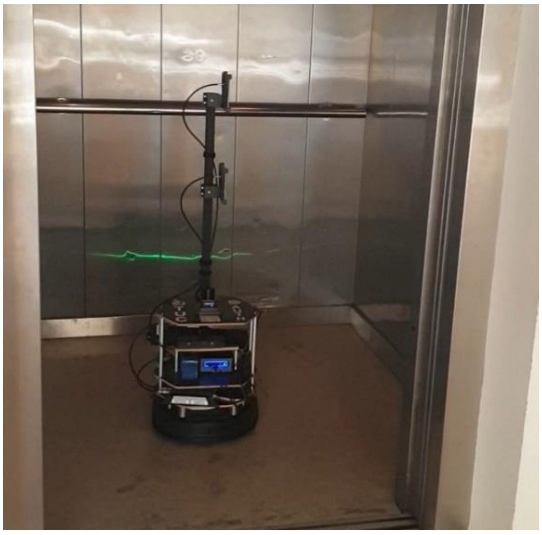

Figure 53 : Robot faces the doors after entering the elevator

When the robot arrives at the elevator doors, it checks if it can enter inside the elevator (see Figure 52). If the doors are open (as Figure 52), the robot will enter the elevator. As soon as it gets inside, its orientation faces the doors again (see Figure 53), and the robot says with voice command, "Please press the button for the ground floor". During the floor transition, the robot has the time to reboot, change maps for the ground floor, and wait until the doors are open again to exit the elevator. Upon the elevator's arrival on the ground floor, the robot verifies if the doors are open and then checks if the floor is correct, using the RFID tag mentioned in Chapter 3.6. If the robot remains on the first floor, it doesn't move, and an additional voice command is issued, saying "Please press the button for the ground floor". When the robot arrives at the correct floor, it can exit the elevator and move to its defined location. This next location is 10 meters from the ground floor elevator.

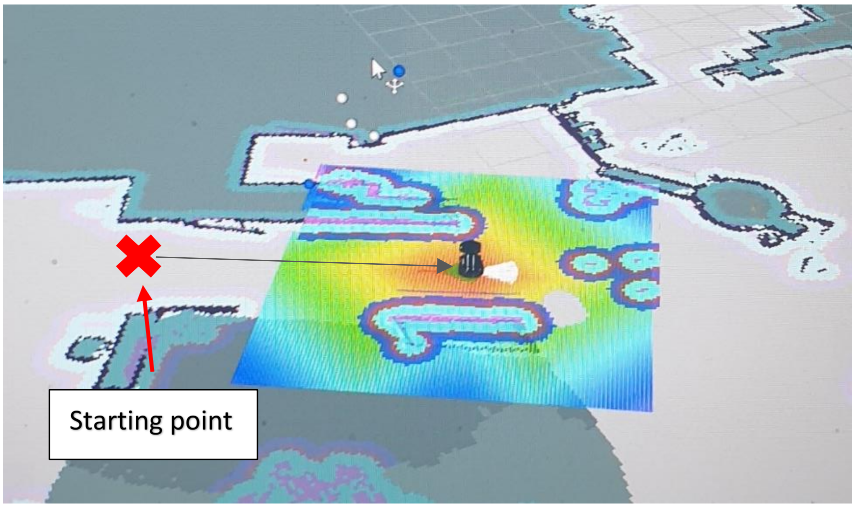

Figure 54 : RFID localization route and results on the Ground Floor area

At this stage, the robot leaves the elevator and proceeds to the specified location to initiate the RFID tag localization process on the ground floor. Figure 54 shows the exact movement of the robot during RFID localization. It initiates from the marked starting point and concludes at the robot's final position in the figure, covering a distance of approximately 5 meters. It is important to mention that in this procedure the RFID reader mentioned in 2.1.4 is enabled from the starting point until the robot's final position in Figure 54. By the robot's arrival at the final position, the RFID reader is terminated, and only the RFID localization algorithm is running for additional 15-20 seconds, as mentioned in 4.1.

The results of the RVIZ tool from Figure 54 compared with Figure 39, the RFID tag position, are remarkably accurate, aligning precisely with the RFID tags positioned in that area (see Figure 39).

Notably, the robot maintains the ability to detect obstacles, like humans directly in its path. Obstacles must not alter the robot's path during RFID reading and localization (see Figure 54).

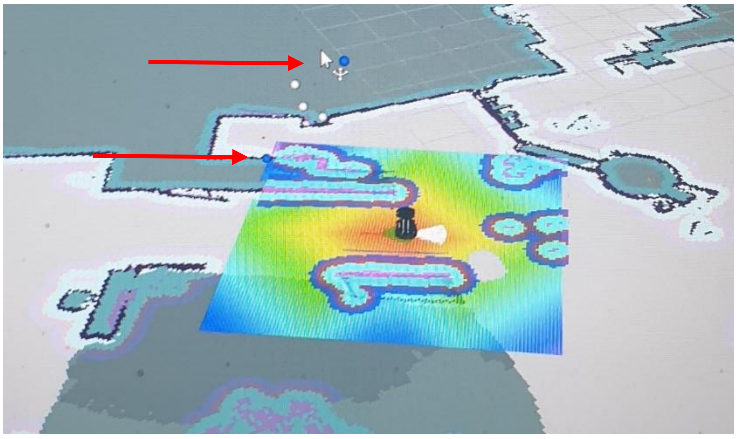

Figure 55 : RFID localization array dimensions

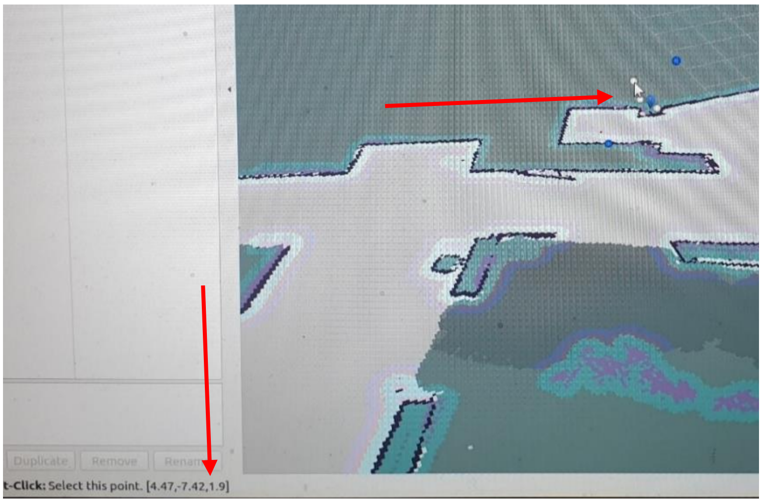

Figure 56 : Ground Floor – RFID localization

The pinpoint locations (indicated by blue color and marked by arrows) represent the dimensional area of approximately 2x2x1 meters designated for the RFID tag localization procedure, while the other pinpoint locations denote the actual positions of the RFID tags (see Figure 56). In Figure 56, following the completion of the localization algorithm, RVIZ can incorporate additional information about the RFID tag localization, including coordinates and height. The displayed height is registered as 1.90 meters, aligning precisely with the RFID tag positioned in that specific location. To verify and confirm these results for the RFID tag localization, details can be extracted using the RVIZ button "Publish Point". Clicking on the tag's position in RVIZ (see Figure 56), details such as the exact positions on the map and the height of each RFID tag are revealed.

Figure 57 : Robot moving back to the elevator with closed doors

After this process is completed, the robot moves through the Secretary's area. Firstly, it has to use the elevator to change floors. In this case, the elevator doors were closed (see Figure 57). When this happens, the robot arrives at the defined location, out of the elevator facing the doors, and it starts repeating the voice command "Please press the button for elevator" every 7 seconds, until the doors are open. After that, the robot enters the elevator with its orientation facing the doors.

Following the robot's exit from the elevator, it proceeds to the Secretary's area. There, it simultaneously runs the RFID tag localization algorithm for the tags positioned in the location, as illustrated in the following figures (see Figure 59, Figure 58). It is important to mention that as the robot moves to the next destination each time, it's crucial to terminate used processes for RFID reading and localization to conserve battery life. The procedure for RFID localization is the same as in the previous location. Additionally, the figures (see Figure 58, Figure 59) show the robot while moving and during the RFID localization process. The velocity stays the same, at 0.25 m/s.

As the RVIZ results show (see Figure 59), and compared with the RFID tags' position in the Secretary's area (see Figure 58), the localization procedure of the RFID tags is very accurate.

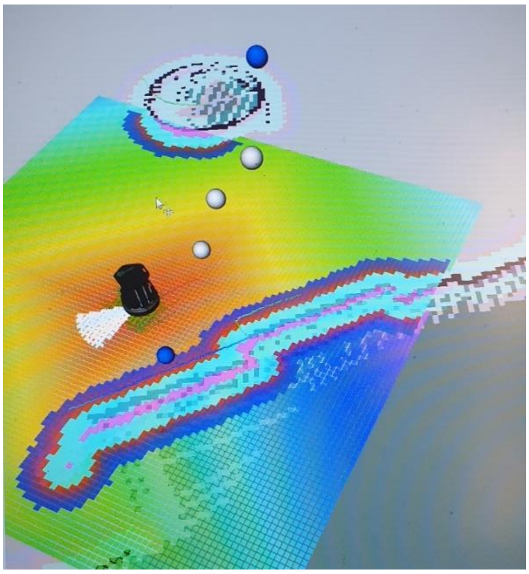

Figure 59 : RVIZ Visualization of RFID Tags in the Secretary – First Floor

Figure 58 : RFID Tags in the Secretary – First Floor

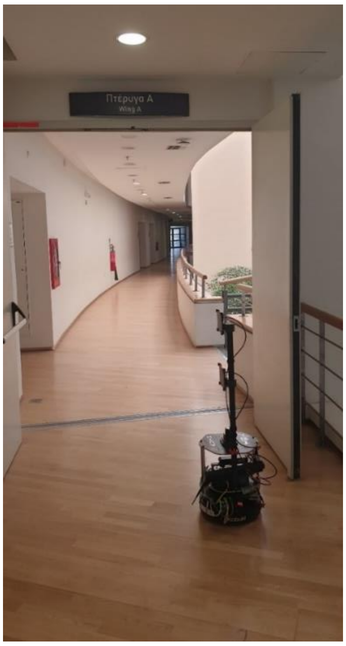

Figure 60 : Moving to the Office

The next and final destination for this route is the robot's return to the office (see **Figure 60**) and localizing RFID tags positioned on the wall. By the robot's arrival at the office destination, it starts its movement in a straight line to localize the RFID tags that are placed on the wall (see **Figure 61**, **Figure 62**, and **Figure 63**).

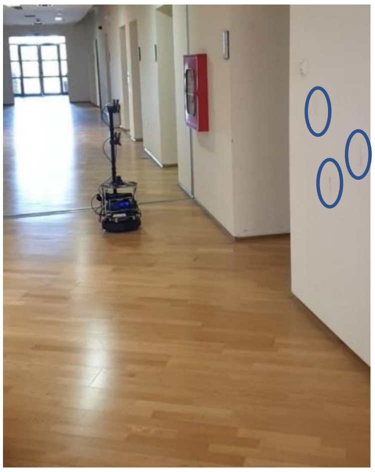

Figure 61 : RFID tag localization Part 1

Figure 62 : RFID tag localization Part 2

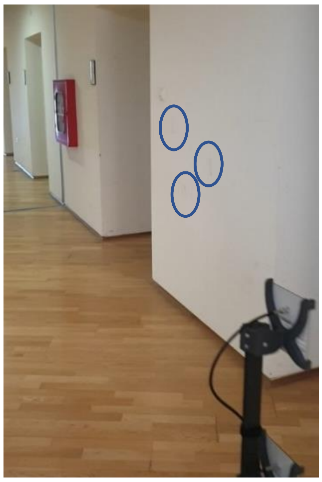

Figure 63 : RFID tag localization Part 3

Robot moving in front of RFID Tags in office location

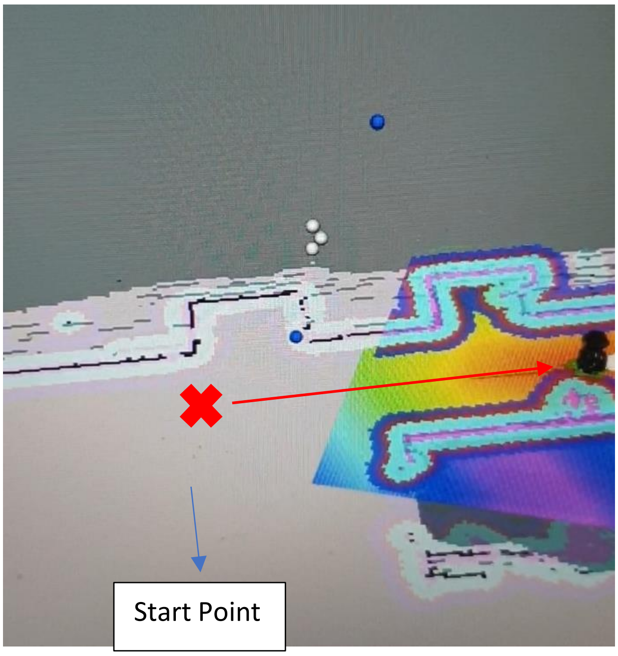

Figure 64: RFID localization route and results in the Office's area on the First Floor

Figure 64 shows the robot's movement in the office area while localizing the RFID tags, using the RVIZ tool. The distance from the starting and ending points is around 5 meters in a straight line. It's important to keep the robot's movements in a straight line during the RFID localization procedure for accurate localization results. The robot's velocity is at 0.25 m/s.

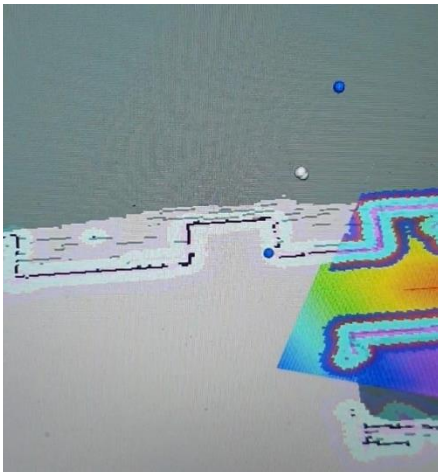

Figure 65 : Localization procedure Part 1

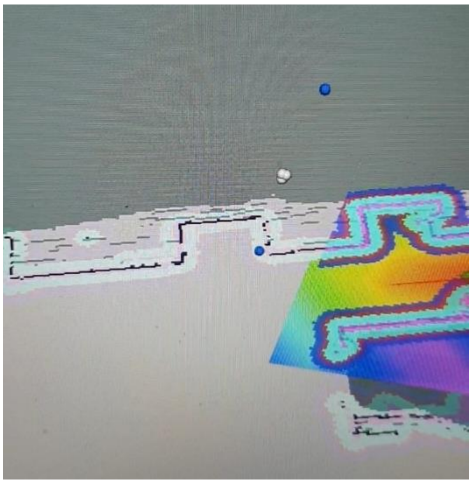

Figure 66 : Localization procedure Part 2

RFID Tag localization process in real-time

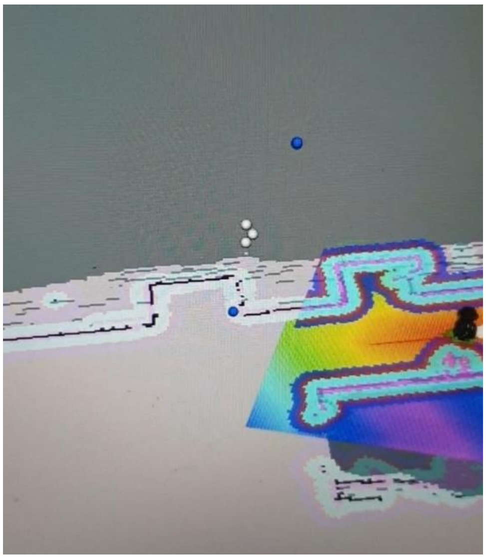

Figure 67 : Final Result Localization RFID Tags in Office Location

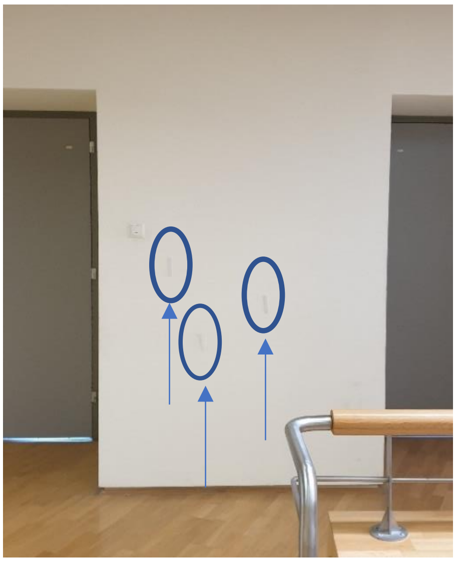

Figure 68 : RFID location on the wall of the Office's area

Figure 65 and Figure 66 show the localization procedure at run-time. The tags are not yet localized perfectly. As the robot moves, the algorithm runs, and the tags explore better localization results. It is important for correct results; the algorithm can run for additional 15-20 seconds after the robot stops moving. If the process is terminated earlier, the results will not be accurate. The final results after the robot's waiting time are shown in Figure 67.

Finally, the accuracy of the results can be verified through a comparative analysis of the RVIZ outcome (see Figure 67) and the RFID tag's position on the wall (Figure 68). The methodical execution of the robot's movement, the strategic waiting intervals for enhanced localization, and the conclusive localization results visualized on the RVIZ tool collectively demonstrates the efficacy of the algorithm in tandem with the robot's movement.

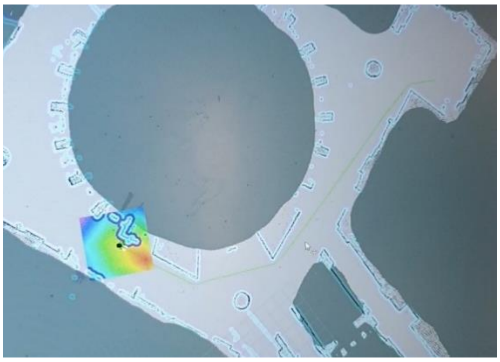

Figure 69 : Robot moving to the cafeteria as seen using the RVIZ tool

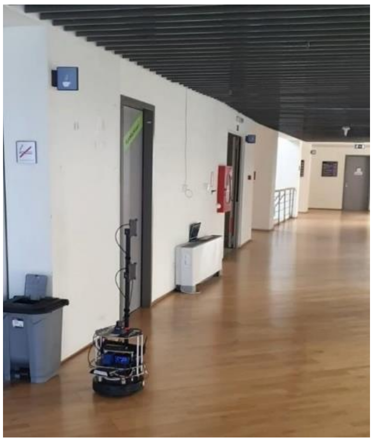

Figure 70 : Robot moving to the cafeteria

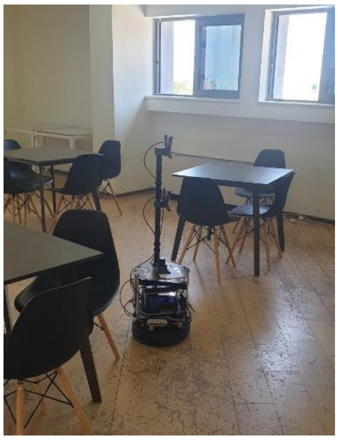

Figure 71 : Robot arrives to the cafeteria

Through the integration of RFID technology with the mobile robot, precise tag localization is achieved with remarkable accuracy. The results affirm the effectiveness of our approach in successfully overcoming challenges such as interference and environmental variations. This methodology is designed to uphold accuracy while accommodating the movement of the robot and adapting to environmental changes.

Taking into account all the locations visited by the robot and the resulting accuracy, the approach presented in this thesis consistently maintains precise RFID tag localization in a dynamically moving robot across expansive areas.

In conclusion, a designated cafeteria location is proposed for the robot's ability to navigate, retrieve, and transport a beverage, contingent upon the addition of a holder to the robot, as part of future work (see Figure 69, Figure 70, and Figure 71 for visual representation).

4.3.5 ChatGPT – Robot Interaction

A component of the robot that was tested but was not integrated is the interaction with ChatGPT [17]. During this testing phase, efforts were made to establish a connection between the chat system and Python code to enhance interaction with the robot. Time was invested in training the chat system to comprehend its role and the specific commands it should execute.

Initially, a set of instructions was conveyed to the chat system, followed by subsequent commands that initiate robot movements. The output generated by the chat system proved to be erratic, rendering it incapable of reliably executing correct commands. The decision was made to disconnect the interaction option. The disconnection happened for several factors. Mainly, the chat system required an extensive amount of time for training. Despite these efforts, the results remained unpredictable.

4.3.6 Timing Results

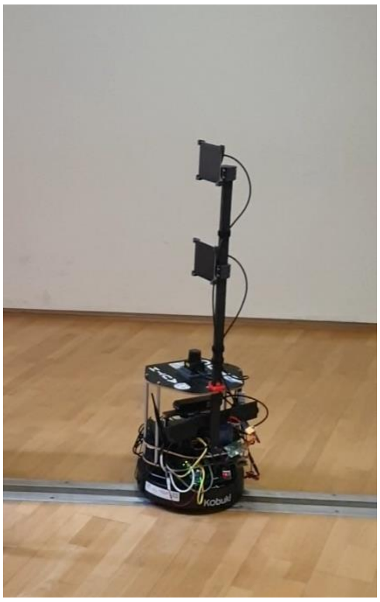

Figure 72 : The robot moves above-ground obstacles

Now that it is coherent how and where the robot navigates in the building, let us take a look into the timing calculations from the experiments. It is important to mention that the exact position of the robot during the run-time, is not known and the robot will find its path to the designated destination. This time has not been considered during time calculations.

The longest distance on the same floor for this thesis starts from the elevator and ends at the office, which is approximately 80-90 meters. This distance is calculated precisely by using the measurement feature of the RVIZ. Given the robot's velocity of 0.25 meters per second, it takes about 5.33 to 6.0 minutes to cover the distance. In practice, the robot completes the journey in approximately 6-6.20 minutes. This time includes avoiding obstacles during the journey.

The robot's velocity is set to a mediocre speed due to the necessity of a few precautions that had to be considered. Firstly, there are cables on the ground of the building which pose a risk if the robot moves too fast, potentially causing it to lose balance and evidently fall (see Figure 72). Secondly, obstacle avoidance detection works in a 5-meter range. If the robot's velocity is increased, it might not detect the obstacles in time which may lead to potential collisions.

Table 1 : Time duration from Point A to point B in the same floor

Location	Destination	Experiment 1 time (minutes)	Experiment 2 time (minutes)	Distance (meters)	Required time (minutes)
Elevator Floor 1	Office	6.20[1]	6.02	83	5.35
Office	Secretary	4.16	4.26	62	4.10
Secretary	Elevator Floor 1	2.00	2.10	27	1.48
Elevator Floor 0	Loc. Floor 0	1.06	0.54	10	0.40
Elevator Floor 0	Lab	1.2	1.30	15	1.00
Secretary	Cafeteria	3.55	4.15	50.5	3.42
Cafeteria	Office	5.2	4.55	70.5	4.40

Table 1 includes the time the robot needs to travel from point A to point B on the same floor. Taking into consideration the distance in meters and the approximately required time of experiment 1 and experiment 2, the results are in close proximity despite the obstacle avoidance procedure during the robot's traversal. Furthermore, when the robot uses the elevator, it takes approximately 1.30 minutes. This accounts for the time the robot waits for the elevator doors to open while entering and exiting.

Table 2 : Time duration from Point A to point B in the different floors

Location:	Destination	Experiment 1 time (minutes)	Experiment 2 time (minutes)	Distance (meters)	Required time (minutes)
Lab	Office	8.5	9.1	98	6.33 + 1.3
Loc. Floor 0	Office	8.55	9.55	95	6.28 + 1.3
Loc. Floor 0	Secretary	4.5	4.6	37	2.28 + 1.3
Lab	Secretary	4.5	5.1	42	2.46 + 1.3

As evident in Table 2, the maximum distance within the building is measured to be 98 meters. The robot's traversal takes up to 8.5 minutes, and the time required is calculated to be 6.33 minutes plus 1.30 for the elevator, totaling 8.03 minutes. It is crucial to note that these experiments simulate real-time scenarios where the robot can navigate obstacles and to wait for the elevator for a longer period of time therefore the additional 50 seconds contributes a positive feedback for the robot's movement for this thesis's results. Having an accurate RFID localization and a robot capable of swift movement across floors is a pivotal aspect of this research. Lastly, to determine the total experimental time, additional 55 seconds must be added for each location of the localization procedure of the RFID tags. The final table, Table 3 shows the times for the routes of multiple locations which are included as options.

[1]This time formatting indicates 6 minutes and 20 seconds.

Table 3 : Multiple Locations Routes

Route	Description	Experiment time (minutes)	Distance (meters)	Required time (minutes)	Total time (minutes)
1	Office-Secr.-Cafeteria-Office	15.4	183	12.2+2*0.55	14.10
2	Office-Loc. Floor 0- Secr. - Office	19.11	194	12.53 + 3.0 + 3*0.55	18.01

The total time for the routes is depicted in table 3. The required and experimental time includes 1.30 minutes for elevator use and 0.55 minutes per RFID localization procedure. The first route involves no elevator use but two RFID localization procedures. In contrast, the second route, which is the longest distance added, includes two elevator uses (going to floor 0 and returning to floor 1) and three RFID localization procedures in total. The total time for Route 2 (see Table 3) is 19.11 minutes, which is almost two minutes longer than the required calculations. These results are accurate due to real-time scenarios in crowded, expansive spaces where avoiding humans while maintaining consistent speed is vital.

4.3.6.2 Reduced Battery Life

After integrating the antennas into the robot, there was an observation of a decreased battery life of approximately 15-20%. This reduction is primarily due to the increased power consumption required to operate the two switches controlling the antennas.

4.3.6.3 Obstacle Avoidance Response Time

Due to the factor mentioned earlier, the response time of our obstacle avoidance system was decreased. If a person moves directly in front of the robot, obstructing its path, the LiDAR sensor, responsible for obstacle detection, will detect the obstacle. However, there might be a slight delay in the data being transmitted. If the person enters the robot's path at a distance of 3 meters or more, the robot can detect the obstacle and promptly reroute to avoid the obstacle.

Chapter 5: Conclusions & Future Work

5.1 Conclusion

The main purpose of this thesis was to create an autonomous robot capable of traversing massive areas and avoiding obstacles. To achieve this laborious task, the movement of the robot had to be programmed, tested, and designed, so it could move freely across any floor of the building with the use of the elevator. After the robot was integrated, the sole addition of speakers served a singular purpose: to assist the robot in transitioning between floors with voice commands. This gives this project a unique interaction between robots and humans that is an important step into the technology used. The Astra Camera also contributed to this task by measuring the distance between the robot and the doors of the elevator, making access to the elevator easier and safer. The robot is capable of autonomous navigation in predefined maps on each floor, can avoid obstacles, particularly humans, while moving in large areas, and successfully reach its destination. Upon reaching their destination, RFID tags can be read and localized autonomously, with the results visualized on the RVIZ tool with high accuracy, down to centimeters. As expected, the unification of all the components into one script posed the most difficult task that ought to be established. After numerous and unceasing attempts, the harmonic synthesis of all the components together has allowed the robot to function entirely autonomously, leading to the success of the thesis.

5.2 Future Work

This project shows the capability of the resulting robot. Hardware and software components were simple to implement, which indicates that there is a big area of improvement with ROS. Firstly, improving the precision of the localization system remains a primary focus. This could involve the development of more accurate RFID tags in larger areas. In addition, combining RFID with other localization sensors, such as LiDAR and cameras, can provide a more comprehensive and reliable localization solution. Furthermore, here is a concept that is based on autonomous decisions regarding RFID data and obstacle information. This includes developing decision-making algorithms or Artificial Intelligence, which can adapt to changing conditions and optimize the robot's trajectories. Also improving the interaction between robots using RFID localization and humans in shared spaces is vital. Further research may explore techniques for robots to safely and effectively collaborate with humans while respecting their presence and intentions. Moreover, as RFID systems become more prevalent, ensuring the security and privacy of transmitted data which is stored in those systems is crucial. Research in this area can focus on encryption methods, authentication protocols, and privacy-preserving techniques. Lastly, leveraging machine learning and artificial intelligence to enhance RFID data analysis and obstacle detection can lead to more adaptable and intelligent robotic systems.

References

[1] "RP LiDAR package," [Online]. Available: http://wiki.ros.org/rplidar.

[2] T. Liu, "ASTRA CAMERA package," [Online]. Available: http://wiki.ros.org/astra_camera.

[3] E. Giannelos, K. Skyvalakis, A. G. Dimitriou, E. Andrianakis and A. Bletsas, "Robust RFID Localization in Multipath with Phase-Based," in IEEE Journal of Radio Frequency Identification, vol.5, no.3,pp.302-310, Sept 2021.

[4] E. Giannelos, "Intelligent Wireless Network and Robots for Low-Cost Battery-less Sensing and Localization," Ph.D dissertation, School of ECE, TUC, expected Oct. 2023.

[5] D. Stonier, Y. Ju, J. S. Santos and M. Liebhardt, "KOBUKI BASE package," [Online]. Available: http://wiki.ros.org/kobuki.

[6] H. A. U. Corporation, "Kokuyo UST-20LX," Hokuyo Automatic USA Corporation [Online]. Available : https://hokuyo-usa.com/products/lidar-obstacle-detection/ust-20lx.

[7] Y. Robot, "Kobuki Documentation," [Online]. Available: https://iclebo-kobuki.readthedocs.io/en/latest/specifications.html.

[8] TTSMP3, "Text to speech mp3," [Online]. Available: https://ttsmp3.com/.

[9] D. Hershberger, D. Gossow, J. F. Faust and W. Woodall , "RVIZ package," [Online]. Available: http://wiki.ros.org/rviz.

[10] W. Hess, D. Kohler, H. Rapp and D. Andor, "Real-time loop closure in 2D LiDAR SLAM, pp. 1271-1278," IEEE Int. Conf. on Robotics and Automation, Stockholm, Sweden, May 2016.

[11] B. Gerkey, "Gmapping package," [Online]. Available: http://wiki.ros.org/gmapping.

[12] AnyDesk, "AnyDesk," [Online]. Available: https://anydesk.com/en.

[13] B. P. Gerkey, "AMCL ROS package," [Online]. Available: http://wiki.ros.org/amcl.

[14] S. J. Santos, "KOBUKI BUMPER package," [Online]. Available: http://wiki.ros.org/kobuki_bumper2pc.

[15] E. Marder-Eppstein, "ROTATE RECOVERY package," [Online]. Available: http://wiki.ros.org/rotate_recovery.

[16] T. P. Brian Gerkey, "MAP SERVER package," [Online]. Available: http://wiki.ros.org/map_server.

[17] OpenAI, "Chat GPT," [Online]. Available: https://chat.openai.com/.