Tartu University

Faculty of Science and Technology

Institute of Technology

Malcom Radigon

Continuous Collaborative Mapping in Unknown Environments: A Multi-Robot System Approach

Master's thesis (30 EAP)
Robotics and Computer Engineering

Supervisor:

Dr Naveed Muhammad

Tartu 2023

Continuous Collaborative Mapping in Unknown Environments: A Multi-Robot System Approach

Abstract:
Navigating and exploring unknown terrains remains a critical challenge within the field of mobile robotics. Achieving rapid and consistent exploration is crucial for the prompt generation of precise maps. This thesis introduces a cutting-edge distributed exploration system utilizing a multi-robot fleet. This innovative system is crafted to facilitate continuous exploration in unexplored areas by implementing a novel drone-based communication relay method. Additionally, it enables the synthesis of an integrated, comprehensive global map, providing crucial, rapid insights for human operators into the explored areas. The effectiveness of this system has been thoroughly evaluated through a series of simulated experiments encompassing various trials. These evaluations underscore the system's capability in smoothly conducting exploration tasks, notably overcoming delays traditionally linked to communication challenges.

Keywords:
Multi-Robot Systems, Autonomous Exploration,Collaborative Mapping,Simultaneous Localization and Mapping (SLAM),Mobile Robotics,Robotic Fleet Management

CERCS: T125 Automation, robotics, control engineering ; T120 Systems engineering, computer technology

Pidev koostööl põhinev kaardistamine tundmatus keskkonnas: Mitme roboti süsteemi lähenemisviis

Lühikokkuvõte:
Tundmatul maastikul navigeerimine ja selle uurimine on endiselt kriitiline väljakutse mobiilse robootika valdkonnas. Kiire ja järjepidev uurimine on oluline täpsete kaartide kiireks koostamiseks. Käesolevas väitekirjas tutvustatakse tipptasemel hajutatud uuringusüsteemi, mis kasutab mitut robotit hõlmavat laevastikku. See uuenduslik süsteem on loodud selleks, et hõlbustada pidevat uurimistegevust uurimata aladel, rakendades uudset droonidel põhinevat kommunikatsioonivahetuse meetodit. Lisaks võimaldab see luua integreeritud, tervikliku globaalse kaardi, mis annab inimoperaatoritele olulise ja kiire ülevaate uuritud piirkondadest. Selle süsteemi tõhusust on põhjalikult hinnatud mitmete simuleeritud katsete abil, mis hõlmavad erinevaid katseid. Need hindamised rõhutavad süsteemi võimet sujuvalt teostada uurimisülesandeid, eriti ületades viivitusi, mis on tavapäraselt seotud kommunikatsiooniprobleemidega.

Võtmesõnad:
Mitme robotiga süsteemid, autonoomne uurimine, koostööl põhinev kaardistamine, samaaegne lokaliseerimine ja kaardistamine (SLAM), mobiilne robootika, laevastiku hal-

damine

CERCS: T125 Automatiseerimine, robootika, juhtimistehnika ; T120 Süsteemitehnoloogia, arvutitehnoloogia

Contents

List of Figures

Terms and Notations

C-SLAM	Collaborative Simultaneous Localisation and Mapping
CSfM	Collaborative Structure from Motion
DOOR-SLAM	Distributed, Online, and Outlier Resilient Simultaneous Localisation and Mapping
D^2SLAM	Decentralised and Distributed Simultaneous Localisation and Mapping
LIDAR	Light Detection and Ranging
MAV	Micro Aerial Vehicle
ORBSLAM	Oriented FAST and Rotated BRIEF Simultaneous Localisation and Mapping
ORBSLAMM	Oriented FAST and Rotated BRIEF Simultaneous Localisation and Multi Mapping
RGB-D	RGB-Depth
ROS	Robot Operating Software
SLAM	Simultaneous Localisation and Mapping
UAV	Unmanned Aerial Vehicles
VSLAM	Visual Simultaneous Localisation and Mapping
DDS	Data Distributed Service
ROS2	Robot Operating System 2
EKF	Extended Kalman Filter
Pose	Position and Orientation

1 Utilization of ChatGPT in Manuscript Refinement

I am giving this explanation in line with section 4 of the University of Tartu's guidelines regarding the use of ChatGPT [2].
The ChatGPT tool [3], developed by OpenAI, served as an auxiliary aid in fine-tuning the content of this thesis. It proved beneficial in translating complex concepts into more accessible language and in choosing fitting vocabulary. The tool's role was crucial in clarifying and succinctly expressing the ideas and arguments, ensuring they accurately reflected the intended meaning. It is important to emphasize that all the work presented in this manuscript is solely my own. The research ideas, experimental designs, and all informational content within this text originated from my efforts. ChatGPT was employed solely for refining and enhancing the manuscript, which is entirely a product of my research and writing.

2 Introduction

In the modern-day landscape of technology, robotics, particularly within the domain of mobile robots, has gained traction with the potential to redefine human interaction with the environment. Robotics has demonstrated significant promise for reshaping various sectors. Recent years have witnessed the rise of intelligent robotic systems, equipped with advanced sensors, perceptive capabilities, and decision-making prowess. This rise has found application across diverse fields such as agricultural surveying [4], radiation monitoring in nuclear facilities [5], mine exploration [6], pipeline surveillance [7], and search and rescue operations [8].

Recent trends and growing interest in robotics have made robot parts like sensors and microcontrollers cheaper and better. This improvement has made it easier to run complex programs and has drawn more attention from researchers and industries. As a result, robots and self-driving vehicles are increasingly being used for repetitive or dangerous jobs, leading to more innovation and growth in this field.

At the same time, the idea of having multiple robots work together has become more popular. This approach involves coordinating several robots to work as a team, which makes them more efficient by dividing big tasks into smaller, easier ones. They can work on these smaller tasks at the same time, which speeds up the process and improves results. This team approach is especially useful in urgent situations, like in search and rescue missions after disasters. For example, during the earthquake in Turkey [9], teams of drones and mobile robots were sent to search through damaged buildings to find people. This method reduces the risk to human rescuers by using robots in dangerous indoor places.

Additionally, having many robots work together to create a single, detailed map of an environment is important. This combined map helps people understand the area better and facilitates informed decision-making and navigation.

In summary, the capability of multi-robot systems to simultaneously manage various tasks is essential in contemporary robotics. This dissertation examines the functioning of these systems, their advantages, and innovative methods to fully utilize their potential in exploring unfamiliar environments.

2.1 Contribution and Structure

2.2 Contribution

The primary contributions of this dissertation are diverse, showing the broad scope of the research in multi-robot exploration systems. Firstly, the thesis introduces an innovative architecture for a distributed multi-robot fleet, designed to enhance the exploration and mapping of unfamiliar environments. This design includes a new way of using drones as messengers to overcome common issues with how far robots can communicate.

Secondly, the research contributes a scalable exploration system that integrates advanced SLAM techniques within a multi-robot framework. By leveraging the collaborative capabilities of ground robots equipped with LiDAR sensors and aerial drones, the system demonstrates improved efficiency in creating detailed and accurate merged maps. These maps are vital for providing operators with a comprehensive view of the explored environment, significantly aiding decision-making processes.

Furthermore, this work offers a comparative analysis of two distinct exploration trials, operating robots independently versus collaboratively with shared data. The knowledge gained from these experiments provides important information about how multi-robot systems operate, especially how sharing information affects the efficiency of their exploration.

Lastly, the dissertation provides a foundation for future research directions, suggesting enhancements in communication protocols and the development of specialized algorithms for collaborative decision-making. These contributions not only serve to advance the field of mobile robotics but also have the potential to inform real-world applications where multi-robot systems can be deployed to achieve efficient and effective exploration outcomes.

2.3 Structure

Chapter 2, Introduction: This chapter introduces the thesis's central theme and research goals. It gives a concise summary of the study's importance and extent, establishing a base for the chapters that follow.

Chapter 3, Literature Review: This chapter provides a thorough review of the literature related to the thesis's focus, particularly SLAM (Section 3.1) and multi-robot exploration (Section 3.3). It outlines previous studies, pinpoints existing knowledge gaps, and sets the stage for the thesis's research questions.

Chapter 4, Background and Theory: Delving into the theoretical aspects, this chapter explores the foundational concepts crucial to the research. It includes discussions on ROS usage (Section 4.1), selecting the optimal SLAM technique (Section 4.2), Navigation Theory(Section 4.3), and Exploration Theory (Section 4.4).

Chapter 5, System Description and Design: This chapter provides a detailed description of the proposed system's structure and design, focusing on key components such as the drones, ground robots, and the operator base, and their interplay. It outlines how this design aligns with the objectives of the research.

Chapter 6, Implementation: This section addresses the practical implementation of the proposed system, including the Gazebo simulation environment and different trial setups.

Chapter 7, Results and Discussion: Here, the outcomes of the system implementation are showcased. The chapter evaluates the results and discusses their implications.

Chapter 8, Conclusion: Summarizing the research's main discoveries, this final chapter reflects on the study's objectives and achievements. It also proposes directions for future research.

3 Literature review

Simultaneous Localization and Mapping (SLAM) is a fundamental problem in robotics and computer vision that involves creating a map of an unknown environment while simultaneously estimating the location of the robot(s) within that environment [10]. SLAM has numerous real-world applications, including autonomous vehicles, augmented reality, and Unmanned Aerial Vehicles (UAV). In recent years, collaborative UAVs and multi-robot systems have emerged as promising areas of research that combine multiple robots or UAVs to accomplish complex tasks such as environmental monitoring, search and rescue operations, and disaster response [11].

Collaborative UAVs and multi-robot systems require robust and efficient SLAM algorithms to achieve accurate and reliable mapping and localization in dynamic and unknown environments. Employment of several robots rather than one robot can accelerate numerous tasks such as exploration and mapping of unknown environments, or enable a team of robots to accomplish a task where each robot has a unique specialization [12]. However, The use of multiple robots introduces new challenges such as coordination, communication, cooperation, and control among the robots, which must be addressed for successful collaboration among a fleet of robots [13] [14]. The development of effective SLAM algorithms and techniques for collaborative UAVs and multi-robot systems is critical for advancing the field of robotics and enabling these systems to be deployed in various real-world applications.

This literature review aims to provide an overview of SLAM techniques and specifically the background in Visual Simultaneous Localisation and Mapping (VSLAM), Light Detection and Ranging (LIDAR), and Simultaneous Localisation and Mapping (SLAM). Apart from that, Collaborative Simultaneous Localisation and Mapping (C-SLAM) in the context of multi-robot systems and multi-robot exploration are discussed. By understanding the current state of the art in SLAM for collaborative UAVs and multi-robot systems, a research gap can be defined to improve and optimise these systems to better equip them with the technology needed to accomplish complex tasks, ultimately improving efficiency and reducing costs in various fields such as agriculture, transportation, and logistics.

3.1 SLAM

Robot mapping and localization is a complex perception problem that uses SLAM techniques to solve the issue. It can be divided into two separate parts which are front-end and back-end. The front-end SLAM algorithm is where the raw sensor data is fed for processing to obtain feature extraction, data association, and loop closing. Whereas, the back-end algorithm produces and optimizes the robot location and map estimation [7]. The SLAM problem was discussed in [15] and [16] using classical solution methods. Since that time, SLAM techniques have been extensively researched for decades and many different SLAM algorithms have been proposed. Visual SLAM and LIDAR-based SLAM are two widely used SLAM techniques, each with its own strengths and weaknesses. Visual SLAM relies on cameras to estimate the robot's pose and environment, while LIDAR-based SLAM uses laser sensors to generate a 2D or 3D point cloud of the surroundings.

3.1.1 VSLAM

VSLAM has seen significant advances in recent years, due to its low cost, low power consumption, lightweight sensors, and easy integration with deep learning-based methods [12] [17]. These benefits make it a viable option to incorporate Visual SLAM systems onto micro-ground or aerial mobile robots. Due to these reasons, Hayyan Afeef Daoud [18] developed a VSLAM algorithm, Oriented FAST and Rotated BRIEF Simultaneous Localisation and Multi Mapping (ORBSLAMM) that is based on the popular monocular Oriented FAST and Rotated BRIEF Simultaneous Localisation and Mapping (ORB-SLAM) [19] algorithm and integrated a monocular camera using a Parrot Bebop Drone [20]. Parrot Bebop Drone is a Micro Aerial Vehicle (MAV) for indoor mapping and exploration. Although ORBSLAM is a VSLAM solution that can be performed using only monocular cameras, depth measurement is not observable from just one camera [19]. Therefore, in order to solve or mitigate this issues, the creator of ORBSLAM developed a modified version of ORBSLAM and proposed ORBSLAM2 which utilizes a setup of stereo or an RGB-Depth (RGB-D) camera allowing for a more robust VSLAM solution [17].

3.1.2 LIDAR SLAM

LIDAR-based SLAM has been widely used in various applications, such as autonomous driving and robotics. LIDAR sensors enable more accurate and robust mapping of the environment, as it is less affected by lighting and environmental conditions than visual SLAM. Apart from that, LIDARs are valued for their high measurement range and the ability to estimate the depth of observed objects with highly detailed data [21]. The 2D LIDAR systems are used for indoor navigation and complex 3D LIDAR systems are used for outdoor navigation [22]. Advances in recent research have also focused on multi-sensor fusion and semantic mapping to improve the accuracy and robustness of the maps by combining the LIDAR sensors with inertial measurement unit (IMU) or vision sensors such as cameras [23]. However, LIDAR sensors are in general heavy and increase the size of the robot, and the cost of such sensor remains high compared to a visual-based SLAM system [24].

3.1.3 Comparison of Visual SLAM and LIDAR-based SLAM

Visual SLAM and LIDAR-based SLAM have their respective strengths and weaknesses. Visual SLAM is faster, cheaper, and more versatile, while LIDAR-based SLAM is more accurate, robust, and less affected by lighting and environmental conditions. The trade-offs between these two techniques depend on the specific application, and researchers have explored the integration of both techniques to improve overall SLAM performance. The use of cameras and LIDAR sensors together can improve mapping accuracy and robustness, especially in dynamic and complex environments. For example, Petráček [25] discusses that for large-scale exploration of caves, an RGB-D-based SLAM approach is sub-optimal compared to a LIDAR-based SLAM due to limited range and field of view (FoV). However, equipping RGB-D cameras that can complement the LIDAR system enables to fill in the blind spots of the LIDAR system. Additionally, Lee, Har and Kum [26] proposed a system integrating a Hokuyo LIDAR with an Intel Realsense Infrared (IR) Depth camera to find victims in search and rescue missions as fusing these sensors together helped to offset their weaknesses.

3.2 Collaborative SLAM (C-Slam)

C-SLAM is an extension of single-robot SLAM that involves multiple robots working together to generate a map of an unknown environment [14]. C-SLAM has numerous advantages over traditional SLAM, including increased efficiency, redundancy, performance, and robustness as each agent can allocate the workload among them reducing the computational power of the individual agent [27]. However, Aitken [7] and Saeedi [14] categorizes the complex multi-robot or C-SLAM problems as:

- Which data will be shared among the agents?
- How is the data shared among the agents?
- Where will the data be processed?

Concerning the processing of the data, C-SLAM processing can be broadly classified into three types: centralised, decentralised, and distributed. Centralised C-SLAM processing involves all robots sending their sensor data to a central server such as a ground station computer or a cloud-based server, fusing the data, and generating a map. On the other hand, decentralised C-SLAM processing involves robots communicating with each other and fusing their sensor data locally to generate a map. Additionally, distributed data fusion processing utilises sensor data that is processed locally, and then they are fused in a centralised processing node [7],[28],[29], [13], [23], [14]. Among these methods, decentralised C-SLAM is more robust and fault-tolerant, as it does not rely on a central server, but can be more challenging to implement due to the need for communication between robots [12].

However, the choice of a processing method for C-SLAM depends on various factors, including the specific application, the number and distribution of robots, the communication and computation resources available, and other constraints.

Centralised processing can provide a globally consistent map and can be effective in scenarios where there is a single central node with high computational power. For example, Forster[27] proposed a method where the task of executing a centralised SLAM system is delegated to a ground station. The MAVs handle low-level tasks, while the ground station performs higher-level tasks such as mapping and loop-closure detection. Each MAV estimates its motion using onboard visual odometry, and the relevant data is sent to the ground station. The ground station runs a Collaborative Structure from Motion (CSfM) system to create individual maps for each MAV and merge them together. The design enables the system to save processing power and require much less transmission bandwidth among the MAVs. However Cieslewski, Choudhary, and Scaramuzza [12] mention that having a centralised SLAM system must always rely on the central node to be reachable, always be in stable network connection and scale sufficiently in computational power and bandwidth.

Distributed processing can provide the benefits of both centralised and decentralised processing, allowing for global consistency and fault tolerance while leveraging the scalability of decentralised processing. However, it may require more sophisticated algorithms and communication protocols to ensure effective collaboration among the agents. For example, Lajoie [30] developed a C-SLAM algorithm, Distributed, Online, and Outlier Resilient Simultaneous Localisation and Mapping (DOOR-SLAM) to be implemented in a distributed SLAM system. It relies on peer-to-peer communication between each robot to perform either single-robot SLAM when no teammates are nearby or distributed SLAM protocol during a rendezvous procedure using Robot Operating Software (ROS) as the middleware and BUZZ [31] which is a scripting language specialised for multi-robot programming. The author ran an experiment using two quad-copters equipped with stereo cameras flying over a football field and found that the back-end transmission power requirements were improved by roughly 50% compared to a centralised system setup.

Decentralised processing can provide better fault tolerance and can be more scalable, as the processing load is distributed among multiple robots or nodes. For example, Hao Xu[23] proposed a novel Decentralised and Distributed Simultaneous Localisation and Mapping (D^2SLAM) that performs the front-end and back-end totally among each UAV without a central node. The map-merge process in D^2SLAM merges only the coordinate system data between the drones while the robots keep their own sparse map. The result of this SLAM system shows a better state estimation accuracy compared to DOOR-SLAM [30] which is a distributed SLAM system. Despite the great potential of decentralised SLAM systems, the author mentions a few limitations with the system such as the scale growth being limited by the front-end computing capabilities of the drone.

Therefore, the best processing method for C-SLAM depends on the specific requirements and constraints of the application, and a careful evaluation of the pros and cons of each approach is necessary to make an informed decision.

3.3 Multi-Robot Exploration

Schack [32] describes a concept related to robotic exploration in uncertain and hazardous environments, where multiple robots work together to map an unknown area. The principal emphasis of this research resides in the optimization of exploration through the utilization of sub-groups within a swarm of autonomous robots, designated for the task of exploration and mapping. This strategic approach introduces an inherent adaptability, where malfunctioning robots pose minimal disruption, given that other robots within the sub-group swarm seamlessly assume their role. In essence, the individual significance of each robot diminishes, as they remain readily replaceable by their counterparts within the swarm group.

The proposed methodology in [32] operates on the premise of sub-teams advancing to the exploration frontiers, where recursive planning is based on the newfound spatial information. As these sub-teams converge at rendezvous points, observations are systematically relayed to the base station, effectively consolidating the collective insights.

In light of the prevailing condition where a substantial portion of information originates from uncharted territories, the reward evaluation is tailored accordingly. Notably, the evaluation focuses on the unknown space's reward rather than encompassing the entirety of the trajectory, a choice that reinforces the scalability of the approach. In an effort to estimate the maximum potential reward stemming from the exploration of unknown spaces, the unknown expanse is methodically segmented into distinct frontier regions. Nonetheless, this approach mandates the utilization of a substantial number of robots, which, in turn, introduces potential challenges to effective robot management and the pragmatic feasibility of the associated costs incurred by deploying such a system.

Cesare[12] aims to enhance the efficiency of exploration and mapping in the context of multi-UAV exploration within indoor environments. The research confronts challenges associated with unreliable communication and limited battery life, focusing primarily on scenarios involving Unmanned Aerial Vehicles (UAVs). The central innovation of this work lies in the integration of four distinct operational states: ”explore,” ”meet,” "sacrifice," and "relay." This integration aims to optimize the processes of exploration, information exchange, and collaborative behaviours among the UAVs.

The experimental phase of this research involved the utilization of two cost-effective autonomous UAVs. These UAVs were tasked with exploring the environment using a frontier exploration algorithm. Communication between the UAVs played a pivotal role in determining exploration paths, leveraging the capabilities of the frontier exploration algorithm. As the UAVs' battery levels reduce, they autonomously returned to their home base. During this phase, they shared their respective maps, which were subsequently merged to create a comprehensive representation of the environment.

The research's core contribution lies in its novel approach to optimizing exploration strategies, enabling effective communication and coordination among UAVs, and addressing the challenges posed by resource constraints. The integration of exploration, communication, and collaboration states exemplifies the study's innovative approach to enhancing the performance of multi-UAV exploration in indoor environments.

3.4 Summary

This literature review serves as the foundational framework for the master's dissertation, focusing on the development of a multi-robot exploration system. The review initiates by emphasizing the significance of Simultaneous Localization and Mapping (SLAM) within the realm of robotics, particularly in the context of multi-robot systems and Collaborative UAVs. The review acknowledges the need for robust SLAM algorithms in these systems due to their potential applications across various domains, including environmental monitoring and disaster response

The literature review delves into the specifics of SLAM techniques, with a specific focus on Visual SLAM (VSLAM) and LIDAR-based SLAM, discussing their respective strengths and limitations. By comparing these techniques, the review emphasizes the trade-offs involved and highlights the potential advantages of integrating them to enhance mapping accuracy and robustness. The analysis then extends to Collaborative SLAM (C-SLAM), shedding light on its inherent advantages, complexities, and processing methodologies. The exploration encompasses centralized, decentralized, and distributed processing methods, along with their implications for multi-robot collaboration and fault tolerance.

Moreover, the review explores the concept of multi-robot exploration, centring on optimization-based exploration through the utilization of sub-teams within a swarm of robots. The discussion takes awareness of the challenges linked with communication, resource limitations, and cost-effectiveness in deploying such systems.

An additional study highlights the efficiency of multi-UAV exploration within indoor environments, underscoring the integration of exploration, communication, and collaboration phases.

In essence, this comprehensive literature review constructs a profound understanding of SLAM techniques and their relevance within collaborative UAVs and multi-robot systems. It addresses the intricate nuances of multi-robot exploration and highlights existing gaps and challenges in the field. Through this critical analysis, the review paves the way for the dissertation's central aim to design and implement a multi-robot exploration system in an uninterrupted and time-efficient manner.

4 Background and Theory

This chapter explains the robotic concept and theory used as a prerequisite to meet the dissertation objective needs, building on a system depending on these robotic blocks to develop a multi-robotic exploration fleet management system. The particular packages and algorithms explained and used throughout the dissertation are due to the knowledge and understanding gained from the literature review section. There are 4 main robotic concepts which will be applied to facilitate the objective.

- ROS2
- SLAM using the SLAM Toolbox
- Navigation using the NAV2 stack
- Exploration using Frontier Exploration algorithm

4.1 ROS Theory

This section provides an in-depth overview of Robot Operating System 2 (ROS2) and its significance in the context of the master's dissertation. Additionally, it outlines the reasons for utilizing ROS in the research and explores the integration of Gazebo simulation for the TurtleBot3 robot. The combination of ROS and Gazebo simulation offers a powerful and efficient platform for developing, testing and validating autonomous robotics applications.

ROS 2 is an advanced middleware framework designed for building and controlling robotic systems [33]. It offers a collection of tools, libraries, and conventions that make it easier to develop complex robotic applications. Unlike its predecessor, ROS 1, ROS 2 was developed with a focus on addressing the limitations of ROS 1 while introducing new features, improvements, and enhanced modularity to meet the evolving needs of robotics.

ROS 2 applications are structured as modular units known as nodes. Each node is a separate process responsible for performing specific tasks. Nodes communicate with one another by publishing and subscribing to topics, forming a publish-subscribe architecture. The concept of a node lifecycle management system adds an extra layer of control over node initialization, termination, and transitions between different states
In ROS 2, communication between nodes occurs via topics. Topics are named channels through which nodes publish messages (data) or subscribe to receive messages. This approach of decoupling publishers from subscribers allows for flexible and efficient communication between various components of a robotic system.

ROS 2 employs a request-response communication model through services. A service consists of a client and a server. The client sends a request to the server, which processes the request and responds accordingly. This mechanism is particularly useful for scenarios requiring synchronous communication, such as requesting sensor data from another robot.

The action system provides a mechanism for managing long-running tasks with asynchronous feedback. It combines the request-response nature of services with progress feedback akin to topics. Actions are initiated by action clients and executed by action servers, allowing for multi-step operations and real-time feedback during complex exploration tasks.

ROS 2 incorporates a parameter server that acts as a centralized configuration manager for nodes. Parameters are key-value pairs that nodes can access and modify during runtime. This feature is invaluable for configuring robot behaviours dynamically, a necessity in multi-robot exploration where each robot might need individualized settings based on its roles and capabilities.

ROS 2's enhanced time management system, compared to ROS 1, provides more accurate synchronization in real-time and distributed systems. This is particularly important in multi-robot scenarios where synchronized actions and coordination are crucial for successful exploration.

The ROS 2 launch system simplifies the initiation of multiple interconnected nodes. It replaces the ROS 1 roslaunch tool and simplifies the management of complex robotic applications. This becomes invaluable in multi-robot exploration projects where orchestrating the launch and coordination of multiple robots and their respective nodes is essential.

ROS 2 provides a robust and versatile platform for experimenting and structuring complex robotic projects, especially for multi-robot exploration missions as the likes of this master's dissertation. The modular nature of nodes, the flexible communication infrastructure of topics, services, and actions, and the centralized parameter management allow researchers to develop and test intricate exploration strategies involving multiple robots. ROS 2's launch system simplifies deployment and orchestration, facilitating the coordination of robotic teams exploring unknown environments. Overall, ROS 2 equips the master's dissertation with the tools needed to delve into the complexities of multi-robot exploration, enabling the development of innovative solutions that push the boundaries of robotic cooperation and navigation.

Gazebo/RVIZ

Gazebo [34] and RViz [35] constitute two widely employed simulation and visualization utilities within the ROS2 framework. These tools assume critical roles in the evaluation and verification of robotic algorithms and functionalities, including those implemented on the TurtleBot3 platform.

Gazebo serves as a versatile and potent robotic simulation environment, facilitating the creation and simulation of intricate robotic systems within a three-dimensional virtual realm. Its capabilities encompass physics-based simulations of robots, sensors, and environments, rendering it a valuable instrument for the trial and development of robotic applications before real hardware implementation. Gazebo meticulously emulates physics, dynamics, and inter-object interactions, thereby yielding realistic robotic movements and behaviors. This utility further extends to the simulation of diverse sensors, including LIDAR and IMU sensors, which are particularly pertinent in the context of this dissertation.

An additional facet of Gazebo is its capacity to formulate custom environments or import real-world maps, thereby facilitating simulation across a spectrum of scenarios. It's worth noting that Gazebo's seamless integration with ROS2, enables interaction and control of simulated robots via ROS nodes and topics. Notably, one can instantiate a TurtleBot3 model within Gazebo, thereby accessing simulated sensors, actuators, and kinematic attributes. This environment affords the capacity to assess navigation, SLAM algorithms, and other behaviors across diverse settings, all without subjecting physical robots to potential risks.

Conversely, RViz serves as a visualization tool that furnishes a three-dimensional graphical interface to depict an array of sensor data, robot statuses, and maps. This tool assumes paramount significance in the visualization and debugging of robotic applications, particularly those reliant on sensor data for navigation or mapping tasks. RViz effectively displays various types of data, such as point clouds, laser scans, maps, and trajectories, within an intuitive interface. Users can seamlessly configure and personalize visualizations to meet specific data requisites. Furthermore, RViz affords real-time representation of the robot's present state, encompassing its position, orientation, and joint angles. Deeply integrated with ROS, RViz seamlessly interfaces with ROS topics, facilitating the visualization of data disseminated through these channels. Consequently, users can effectively visualize sensor data, including LIDAR scans, camera images, and IMU measurements, within an emulated real-world context. This includes real-time displays of the robot's trajectory, position, and pertinent real-time data.

These tools constitute pivotal assets for the simulation and visualization of robotic systems. They provide an efficacious and secure avenue for the development, testing, and validation of diverse algorithms and behaviours, all preceding physical hardware deployment.

4.2 SLAM Theory

Based on existing research and literature, current findings suggest that Pose Graph SLAM represents the most advanced SLAM algorithm, demonstrating superior performance compared to alternative algorithms like Extended Kalman Filter (EKF) and Particle Filter. This is particularly noticeable in the context of filter-based systems versus smoothing-based systems. In order to facilitate the implementation of Pose Graph SLAM, the dissertation will employ a widely adopted SLAM package known as the Slam Toolbox [36]. This package, deeply rooted in graph-based SLAM principles, is extensively used within the ROS2 framework, as elaborated upon in subsequent sections. The Slam Toolbox seamlessly aligns with the objectives of this dissertation due to its inherent features and tools, ensuring a cohesive integration with the research objectives. In the following sections, the dissertation will delve into a comprehensive background on the Slam Toolbox, explaining its core functionalities, and its essential role in the context of Pose Graph SLAM.

4.2.1 Pose Graph SLAM

Pose Graph SLAM is a variant of Simultaneous Localization and Mapping (SLAM) that focuses on solving the problem of estimating both the robot's trajectory (pose) and the map of the environment simultaneously. In traditional SLAM, the robot's trajectory and the map are estimated using filtering techniques like Extended Kalman Filter (EKF) or particle filters. However, in Pose Graph SLAM, a graph-based optimization approach is used to jointly optimize the robot's poses and the map.

The key idea behind Pose Graph SLAM is to represent the problem as a graph as depicted in Figure 1, where nodes represent the robot's poses (positions and orientations) at different time steps, and edges represent measurements or constraints between these poses. These constraints can come from various sources, such as odometry, loop closures, or observations of landmarks in the environment.

The workflow of Pose Graph SLAM typically involves many steps. First, the robot collects sensor data, such as odometry, laser scans, or camera images, to estimate its poses and perceive the environment. Data association involves matching sensor measurements with the robot's poses to create the initial graph structure. Then, the initial pose graph is constructed based on the robot's odometry and sensor measurements. The graph consists of nodes representing robot poses and edges representing constraints between connected poses.

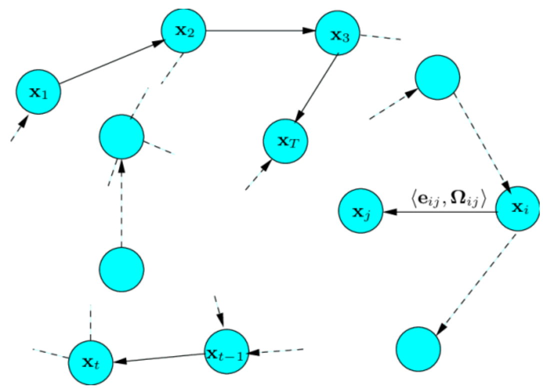


Figure 1. A pose-graph representation of a SLAM process. Every node in the graph corresponds to a robot pose. Nearby poses are connected by edges that model spatial constraints between robot poses arising from measurements [1]

Loop closures occur when the robot revisits a previously visited location. Loop closure detection identifies these revisits by comparing features or signatures from the sensor data collected at different time steps. When a loop closure is detected, a new edge is added to the graph, connecting the relevant nodes and constraining their relative positions, which helps correct drift errors and improve map consistency.

The core of Pose Graph SLAM is the graph optimization process, where the goal is to find the most likely poses and maps that satisfy all the constraints in the graph. The optimization process adjusts the poses and map to minimize the errors in the constraints while considering the uncertainty in the sensor measurements.

Once the graph optimization is performed, the optimized poses and map are used to update the robot's trajectory and the map of the environment. The updated map represents the robot's belief about the surroundings.

Finally, Pose Graph SLAM is an iterative process. The loop closure detection, graph optimization, and map update steps are repeated as the robot explores more of the environment, and new loop closures are detected. This iterative refinement process improves the accuracy and consistency of the estimated trajectory and map over time.

4.2.2 SLAM TOOLBOX

The ”Slam Toolbox” [36] a collection of specialized tools and functionalities designed for 2D Simultaneous Localization and Mapping (SLAM), has been developed by Steve Macenski, and adapted specifically for integration within the ROS2 ecosystem [37].The inclusion of the Slam Toolbox is of considerable significance within the context of this master’s dissertation due to its highly advantageous features that align with the research objectives. This relevance is explained from the compatibility of the Slam Toolbox with Turtlebot3, a robotic platform equipped with an integrated 2D LiDAR sensor, encouraging seamless synergy between the toolkit’s 2D SLAM capabilities and the robot’s hardware.

Furthermore, the utilization of the Pose Graph SLAM algorithm, as elaborated upon in the preceding section, further reinforces the merit of incorporating the Slam Toolbox into this dissertation. A notable attribute of the Slam Toolbox is its competence in serializing and deserializing the generated maps resulting from the SLAM process for individual robots. This capability assumes particular importance as it facilitates the storage of map data in a serialized format, enabling subsequent retrieval, deserialization, and application to either the same or different robotic platforms. This characteristic distinctly sets the Slam Toolbox apart, as other available SLAM packages or algorithms lack this trait. Unlike alternative solutions that exclusively permit map preservation for the sole objective of localization use cases, and restrict from updating the map, the Slam Toolbox empowers the retention and subsequent transfer of maps, aligning seamlessly with one of the core objectives of this dissertation.

An experiment was conducted to build a map of a virtual environment using ROS2 and the SLAM Toolbox to understand how to implement Pose Graph SLAM and SLAM Toolbox within ROS2. Initially, This environment was entirely unexplored. Figure 2 depicts the robot’s starting pose and the initial map. As the robot explores, new nodes and edges emerge, as shown in Figure 3. In this figure, the red dots represent nodes, and the blue lines link them as edges. Moving through the area updates the map, gathering more information. Figure 4 presents the complete map, with all nodes and edges after the entire environment has been explored.

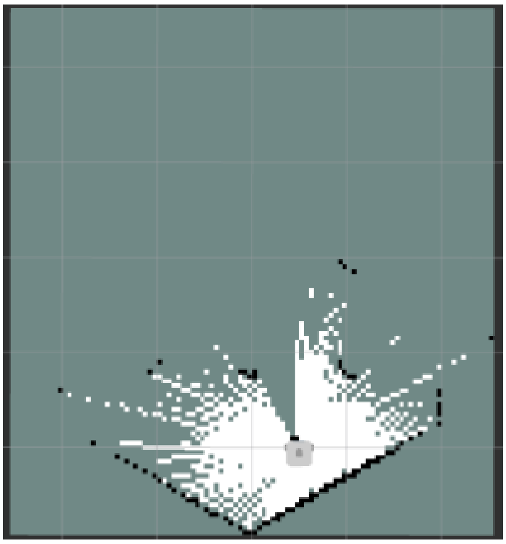


Figure 2. Pose Graph SLAM Initially Mapped

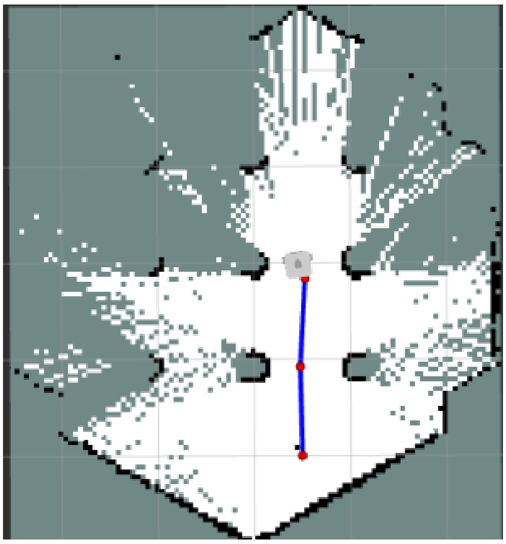


Figure 3. Pose Graph SLAM Partially Mapped

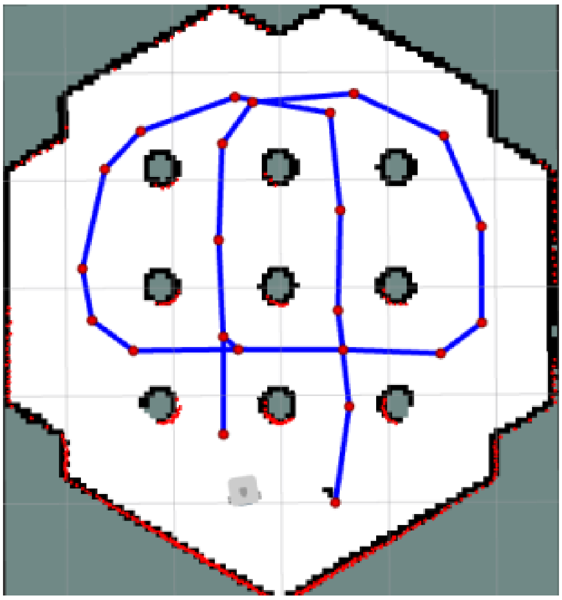


Figure 4. Pose Graph SLAM Fully Mapped

4.3 Navigation Theory

The Navigation2 (nav2) stack [38] in ROS 2 uses a combination of algorithms to navigate from the robot's current position to a goal position along a path. The navigation process involves both global and local path planning, as well as obstacle avoidance and localization. The primary algorithms used in the navigation process are as follows:

The global path planning algorithm is responsible for finding an initial path from the robot's current position to the goal position. One of the commonly used global path planning algorithms in the Navigation2 stack is A* (A-star). A* is a popular graph search algorithm that guarantees an optimal path in terms of the least cost. The algorithm uses a heuristic to estimate the cost from each node in the search space to the goal, enabling it to find an efficient path. A vital addition is the boundary layer, which marks areas around obstacles and influences the path planning process as depicted in Figure 6.

After establishing the global path, the responsibility shifts to the local path planning algorithm, which guarantees a seamless and obstacle-free progression along the route. The local planner uses different algorithms, such as DWA (Dynamic Window Approach) and TEB (Timed Elastic Band), the local planner computes velocity commands that guide the robot along the predefined global path while adeptly avoiding obstacles in real-time. As depicted in Figure 7, the local planner interfaces with the local costmap, which provides a detailed representation of the robot's immediate surroundings. Considering the boundary layer, nearby obstacles, and moving barriers, the local planner changes the path as needed, making sure the navigation is both safe and effective.
Figure 8 encapsulates the fusion of global and local planning layers. The waypoints from the global plan are handed over to the local planner, which incorporates information from the local costmap to guide the robot's trajectory in real-time. This cooperative approach facilitates effective navigation by navigating around obstacles while adhering to the desired path.

To ensure accurate navigation, the utilization of the Adaptive Monte Carlo Localization (AMCL) algorithm becomes imperative. Employing particle filtering, a sophisticated mathematical technique, AMCL undertakes the task of estimating the precise position of the robot within a given map. This process can be visually seen in Figure 9, wherein a collection of distinct green dots signifies potential robot positions. With each iteration of navigation, the gradual convergence of these individual green dots emerges to AMCL's refining precision as depicted in Figure 10. This convergence demonstrates the algorithm's increasing confidence in confirming the actual pose of the robot. This iterative refinement process adds to AMCL's seamless integration of data collected from laser scans and odometry readings, allowing the algorithm to consistently recalibrate the robot's orientation and location.

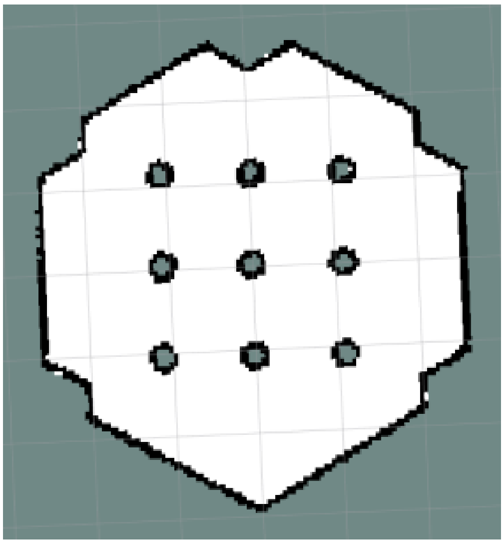


Figure 5. 2D Occupancy Grid Map of the Environment

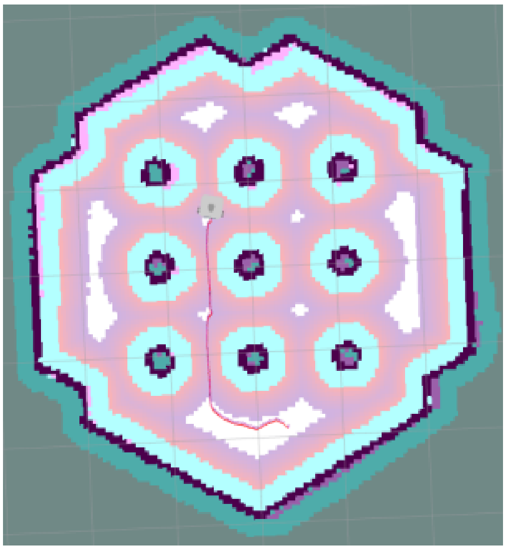


Figure 6. Global Planner

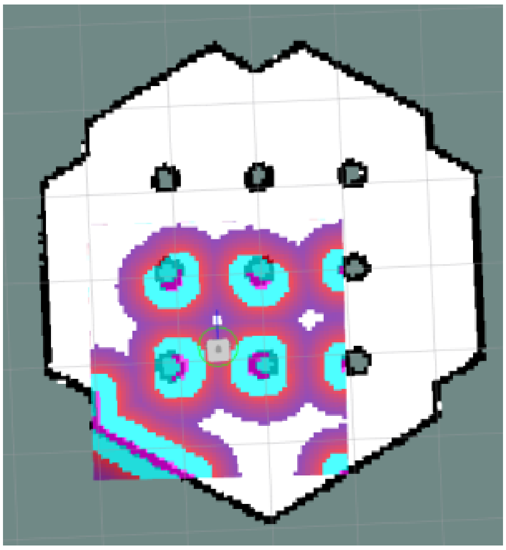


Figure 7. Local Planner of the Robot

Figure 8. Local and Global Planner

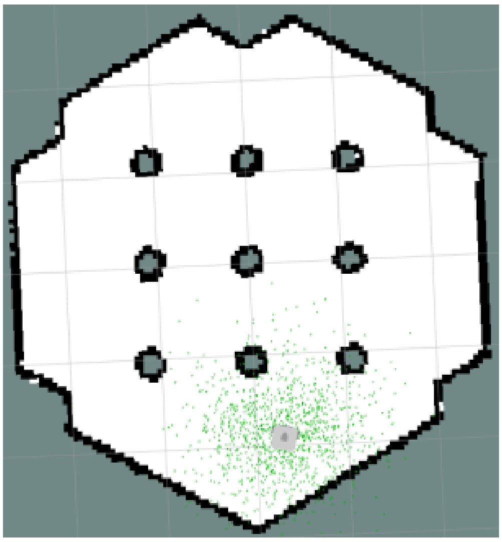


Figure 9. AMCL at the Initial Pose the Robot

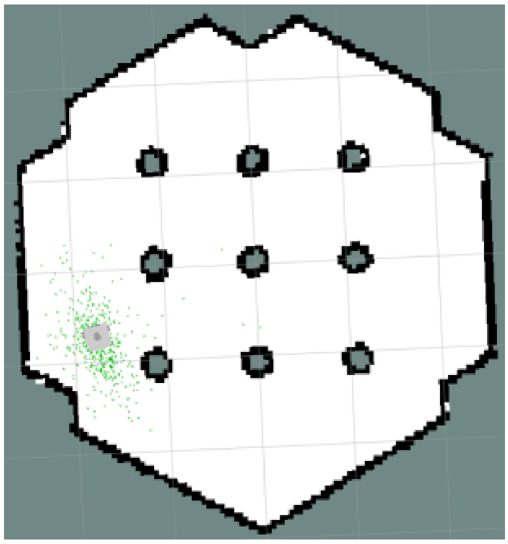


Figure 10. AMCL Algorithm after navigating for few seconds

4.4 Exploration Theory

Frontier exploration is a robotic exploration strategy that aims to efficiently explore and map unknown environments by identifying and visiting frontier points. Frontier points are locations on the boundary between explored and unexplored areas, representing the frontier of the known environment. The concept of frontier exploration is commonly used in robotic exploration scenarios, such as mapping an unknown environment or searching for specific targets.

The exploration process begins with the robot's sensors, such as cameras or laser scanners, collecting data about the surrounding environment. The robot creates a map of the known areas based on the sensor data, using techniques like SLAM (Simultaneous Localization and Mapping) to simultaneously localize itself and build a map of the environment. Once the robot has a map of the explored areas, it uses this information to identify frontier points. Frontier points are locations on the boundary between the known and unknown areas (Figure 11). These are areas that the robot has not yet explored, and visiting them could potentially reveal new information about the environment.

The robot evaluates the potential frontier points based on certain criteria, such as distance, accessibility, and information gain. Frontier selection methods may consider factors like the proximity of the frontier to the robot's current position, the amount of unexplored space beyond the frontier, and the likelihood of finding interesting or valuable information. Once the frontier points are identified and ranked, the robot uses path planning algorithms to find the most efficient paths to reach these frontiers while avoiding obstacles. The robot plans its trajectory to visit one or multiple frontier points, maximizing its coverage of the unexplored areas.

The robot executes the planned path and reaches the selected frontier points. Upon arrival at a frontier, the robot may further assess the area with its sensors and update the map as needed. It may also revisit certain frontiers if there is a need for additional information.

Frontier exploration is typically an iterative process. As the robot explores and updates its map, new frontier points may become visible due to changes in the environment. The robot continually detects and selects frontiers to explore until it has sufficiently covered the target area or achieved its exploration goals (Figure 12).

Frontier exploration strategies help robots explore unknown environments in a systematic and efficient manner, ensuring that they focus on the most promising and informative areas. This approach is widely used in various robotics applications, such as search and rescue missions, environmental monitoring, and mapping uncharted territories.

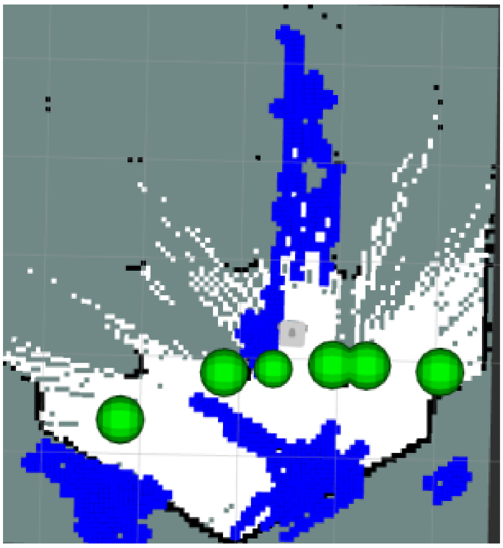


Figure 11. Initialization of Frontier Explore

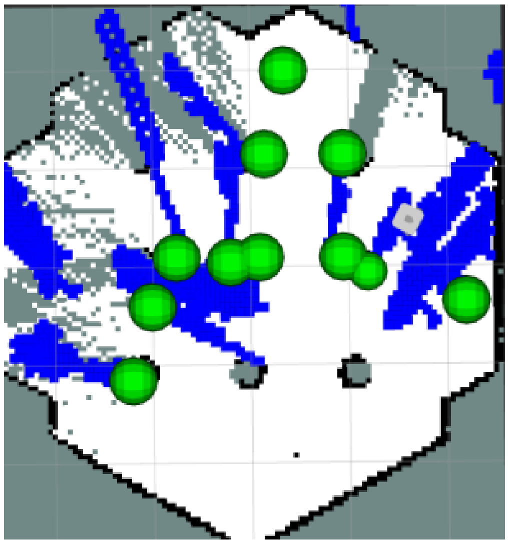


Figure 12. Map Coverage after Frontier Exploration

5 System Description and Design

Building upon the theoretical underpinnings introduced earlier, This chapter will elaborate on the architecture crafted for the implementation of a cooperative multi-robot mapping system. This design, demonstrated using a scenario with two ground robots, is inspired by the principles of collaborative exploration and efficient data sharing. It underpins the primary objective of this dissertation: the development of a system for continuous, coordinated mapping by multiple robots. While the current simulation scenario involves two ground robots, the architecture is scalable and can potentially be expanded to accommodate 'n' number of robots for larger-scale operations.

In this scenario, two ground robots and an associated pair of drones are deployed for exploration. The ground robots, equipped with advanced LiDAR sensors, are responsible for mapping the environment. Each ground robot is linked to a dedicated drone that plays a pivotal role in data transmission. At predetermined intervals, these drones rendezvous with a third drone connected to the operator base. This meeting point serves as a hub for exchanging map data, ensuring continuous information flow and map updating.

A critical aspect of this system is managing the exploration territory of each ground robot to avoid redundant mapping. To this end, the drones not only transmit their respective robot's map data to the base but also receive the map data of the other robot. Upon returning to their ground robots, they share this new information, enabling each robot to update its exploration strategy based on the combined knowledge of the terrain.

When the third drone reaches the operator base, it transmits the collected map data, enabling the base to merge the maps received from the drones. This merged map offers a comprehensive view of the explored area, significantly enhancing the understanding of the environment. The base station's role is crucial in synthesizing individual robot maps into a singular, unified representation of the mapped area.

This system's design guarantees an uninterrupted mapping operation by both ground robots at all times. With the drones dedicated to data transmission, they enable the ground robots to persist in their exploration tasks seamlessly. This ongoing cycle of mapping, coupled with the constant exchange and updating of data, significantly boosts the overall efficiency of the mapping. The Implementation section will delve into the specifics of implementing this system.

5.1 System Design of Exploring Robot

Designing an autonomous exploring robot entails the integration of several critical components to enable effective navigation and environmental exploration. The robot's core functionalities hinge on Simultaneous Localization and Mapping (SLAM), to develop a comprehensive map of the surroundings and continuously refine the robot's position estimation.

This map serves as the foundation for the subsequent components. Frontier Exploration plays a pivotal role in identifying unexplored regions, determining potential points of interest, and calculating the viability of reaching these frontiers. This involves a careful analysis of factors like distance, terrain, and obstacles. The subsequent Motion planning phase generates a collision-free trajectory from the robot's current location to the chosen frontier point. This task employs advanced algorithms such as A* or RRT, which factor in the robot's kinematics and environmental constraints. This fusion of SLAM, frontier exploration, and motion planning orchestrates the robot's Exploration Strategy, periodically updating its map, identifying frontiers, and selecting goals for further exploration based on predefined criteria. The robot's Obstacle Avoidance mechanisms play a pivotal role during path execution, dynamically recalculating paths to circumvent obstacles while staying the course toward the goal. Figure 13 shows a diagram of the different robotic components to achieve an autonomous exploring robot.

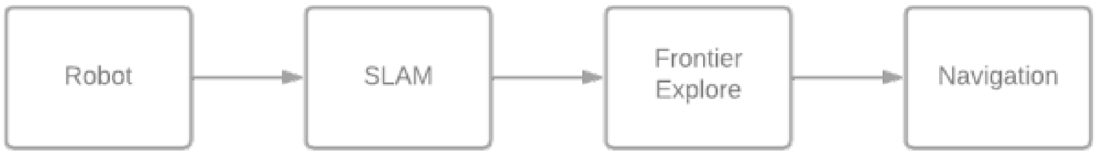


Figure 13. Block Diagram Representation Of Robot Exploration

5.2 ROS Multi-Robot Architecture

The operational workflow is meticulously orchestrated, as represented in the series of flowcharts Figure 14, Figure 15 and Figure 16. The figures provided delineate the systematic workflow of a continuous robotic mapping operation.

In Figure 14, ground Robots A and B, paired with their respective drones, perpetually engage in mapping activities, punctuated by routine checks for action triggers. At every designated interval T, each robot communicates its collected map data to its affiliated drone, which then conveys this information to a predetermined rendezvous point for data exchange.

As depicted in Figure 15, the drones utilize their downtime efficiently by recharging on the ground robots, waiting for their next operational command. Upon receiving the instruction to assemble, they navigate to the rendezvous point, where they await the arrival of their counterparts. It is only when all drones have assembled within a certain communication range that they start the data-sharing process. After this exchange, the drones return to their links: Drones A and B to Ground Robots A and B for information dissemination, and Drone C to the Operator Base.

In Figure 16, the Operator Base functions cyclically, echoing the periodic actions of the drones. With each passing period T, the base dispatches Drone C to the rendezvous point. Upon its return, the base undertakes the crucial task of merging the maps gathered from the other robots, thus maintaining a current and comprehensive understanding of the mapped environment. This cyclical process, carefully illustrated in the figures, ensures a seamless and efficient collaborative mapping effort, with each component of the system ground robots, drones, and the operator base performing synchronized tasks vital to the overarching goal of continuous exploration and data integration.

5.3 Considered Alternative Not Incorporated in the Final Design

In contemplating potential modifications to the current multi-robot mapping architecture, one plausible alteration is the transition from a multi-drone framework to a singular drone system. This unique drone would cyclically traverse between each ground robot and the operator base, serving as the sole conduit for the essential exchange of mapping data.

However, this approach introduces several challenges that necessitate rigorous consideration and strategic planning. Transitioning to a singular drone framework for mapping operations could inadvertently decelerate the overall process due to several logistical complexities. The primary challenge lies in the coordination of one drone with multiple ground robots, a significant shift from the current multi-drone setup. In the existing system, each drone is linked to its respective ground robot, efficiently tracking and returning to it after data exchanges at the central rendezvous point. This efficiency is primarily due to each drone's familiarity with its robot's last reported location.

However, this dynamic changes considerably in a single drone scenario. The solitary drone, tasked with servicing multiple ground robots, faces challenges in quickly pinpointing and reaching each robot. This challenge is exacerbated when these robots are actively exploring and changing their positions. To address this, ground robots in both scenarios are programmed to stay within a communicable range of their last known positions, a critical feature that ensures the drone can successfully locate each robot upon its return.

Yet, the impact of this programming is more pronounced in the single drone model. Due to its longer travel route, encompassing visits to each ground robot in sequence, the single drone's round trip is considerably lengthened. This results in increased waiting periods for the ground robots, as they are required to remain within the range of their last known position until the drone's arrival. Such prolonged periods of inactivity for the ground robots could lead to less efficient mapping operations compared to the multi-drone system. In the case of multi-drones, the relatively quicker return of each dedicated drone significantly reduces the idle time for its corresponding ground robot, thereby maintaining a more continuous and efficient mapping workflow. This contrast highlights the need for careful consideration of the drone's travel time and its consequent effect on the operational efficiency of the ground robots in a single drone mapping system.

Another pivotal aspect to address is the mapping efficiency. In the multi-drone system, each robot receives updated mapping information from its dedicated drone, including data collected by other robots, thereby ensuring a cohesive and comprehensive mapping process. However, with a single drone, there arises a sequential delay in information dissemination. The ground robot first visited by the drone would not immediately benefit from the data collected by subsequent robots. This sequential gap could result in less efficient mapping, as each robot may not have access to the most current collective mapping data during its exploration phase.

Furthermore, the issue of drone battery life emerges as a significant constraint, particularly in scenarios involving expansive mapping areas. The solitary drone, tasked with covering extensive distances to reach each ground robot and returning to the operator base, might face battery depletion, potentially incapacitating the system's ability to maintain continuous mapping operations. This challenge underscores the necessity for advanced battery technology or alternative strategies like mid-mission charging stations or swappable battery systems to ensure uninterrupted operation.

In summary, while the concept of a single drone system presents a streamlined and potentially cost-effective approach, it is imperative to meticulously evaluate and address these logistical and technical challenges to ensure the system's efficacy and efficiency in large-scale mapping operations. Further research and development in drone technology, particularly in areas of battery longevity and autonomous navigation algorithms, could play a pivotal role in overcoming these challenges and making the single drone model a viable alternative in future iterations of the mapping system

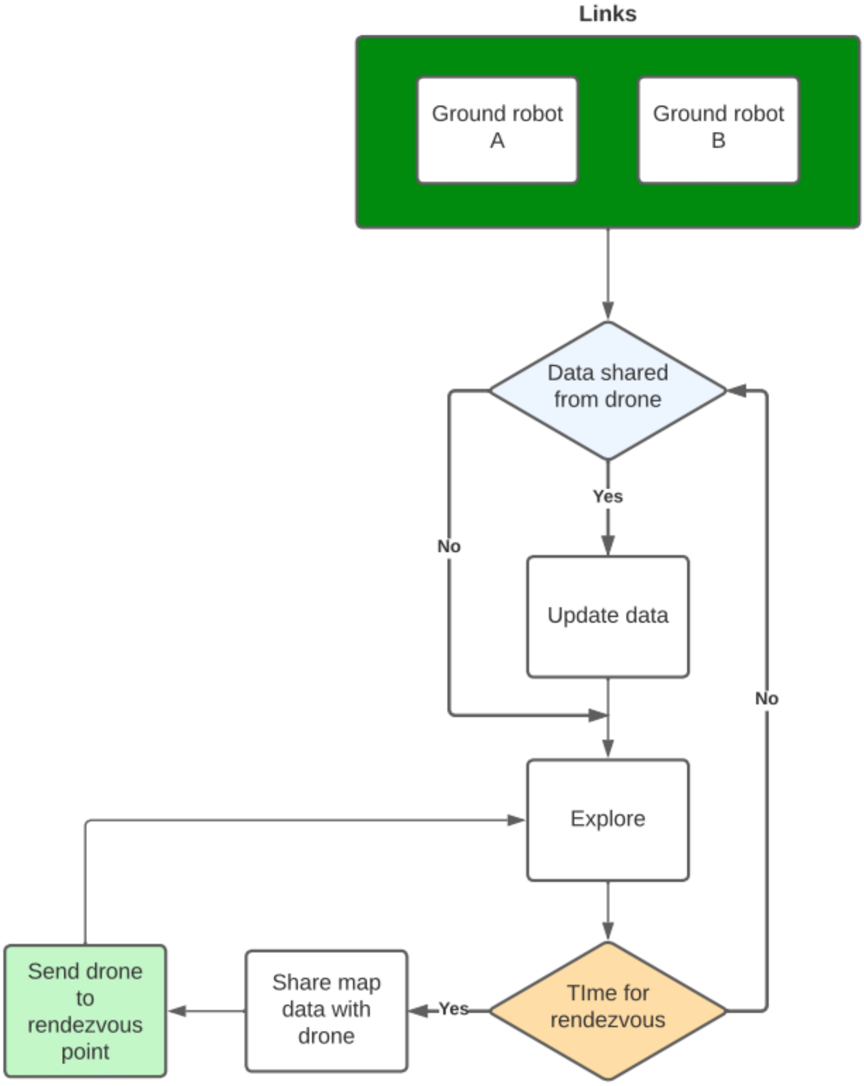


Figure 14. Block Diagram Representation of Each Robot in a Multi-Robot Fleet Exploration System

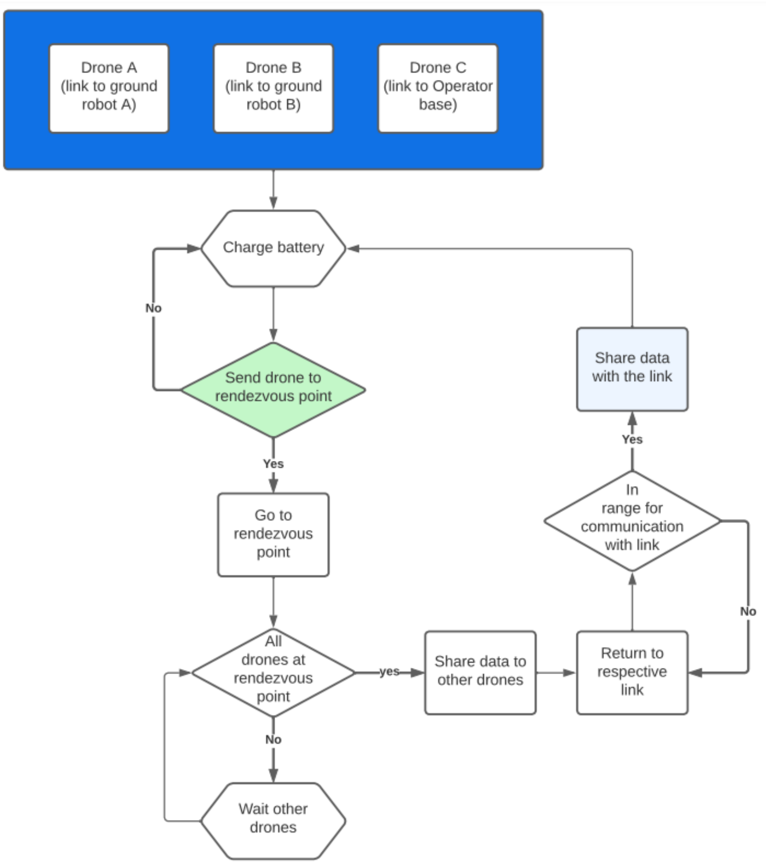


Figure 15. Block Diagram Representation of Each Drone in a Multi-Robot Fleet Exploration System

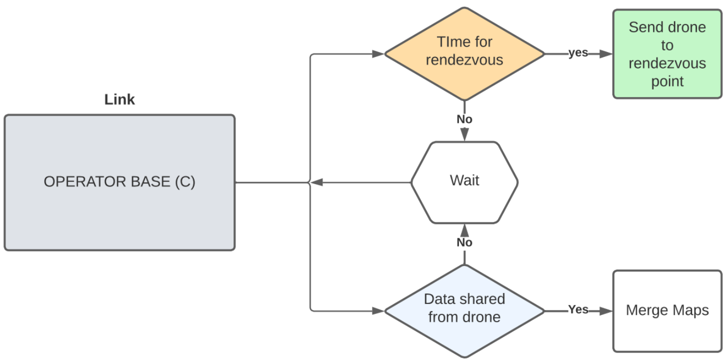


Figure 16. Block Diagram Representation of the Operator Base architecture

6 Implementation

This chapter focuses on implementing the system design discussed in earlier chapters, aiming to validate its effectiveness for managing a multi-robotic fleet. The implementation is carried out through simulation. The simulation phase is a key part of this master's dissertation, intended to confirm the effectiveness of the multi-robot exploration structure designed previously. In this chapter, a detailed methodology for the simulation experiment is presented step by step.

6.1 Gazebo World Environment

To achieve a lifelike simulation environment conducive to robot exploration, an outdoor maze setting has been created into Gazebo (Figure 17), serving as the world within which the TurtleBots3 are tasked to navigate and map.

Here, two ground robots, modelled as Turtlebot3 units, have been initialized at separate locations within a maze environment. For simulation purposes, and owing to model constraints, two drones, modeled as SJTU drones, are configured to hover above their corresponding ground robots, simulating a constant state of readiness to act upon navigational directives (Figure 17). This aerial positioning, while a deviation from real-world operational practices, is necessitated by the current simulation parameters. In a practical real-life application, these drones would ordinarily land on and recharge atop the ground robots, conserving energy for sustained operation.

Additionally, the simulation environment encompasses an operator base, complete with an associated drone (Figure 18). This drone's function is to simulate the critical role of data relay between the mapping robots and the operational oversight at the base. The human element within this virtual setup is represented by an avatar, standing in as the operator who oversees the mapping integration process directly from the base station. This operator avatar symbolizes the human supervision essential for monitoring the progression of the mapping exercise and for the amalgamation of the collected data into a coherent representation of the explored environment.

In the initial phase of the simulation, Two ground robots are placed at different points in a maze. On standby are three drones, ready to take flight. When the operator signals the start, the robots activate and start their journey through the maze, as outlined in Figure 14. As mentioned previously, for the sake of the simulation, the drones follow the ground robots closely, simulating the real-life scenario where they would be mounted on the robots for recharging. At a specified time, referred to as time T, the drones converge at a common meeting point to exchange mapping data (Figure 19). After this exchange, they return to their respective robots and pass on the updated information. Armed with this new data, the ground robots then adjust their maps and set new frontiers to explore, while the operator watches the combined map evolve in real-time. This process repeats until the maze is fully explored and there are no new areas left to discover.

6.2 Independent and Collaborative Mapping Trials

To test our mapping strategy, we ran two different trials. Both follow the scenario we have just described, but they differ in one key aspect: whether or not the ground robots can see each other's maps. In the first trial, named the Independent Mapping Trial, each robot operates on its own, using only its sensors, its known location, and the part of the map it has created. In the second trial, referred to as the Collaborative Mapping Trial, the robots have access to each other's data, thanks to the drones sharing information when they meet. This means that when the drones return to their robots, they bring back extra information. This allows the robots to make smarter choices during exploration, avoiding areas the other has already covered or plans to cover.

However, the second approach presents a challenge. When both robots have access to identical information simultaneously, there is a potential for them to select the same region for their next exploration target. To avoid this, the programming must be adjusted so that one robot is directed to the next nearest unexplored area instead. This ensures that the exploration process remains efficient, with both robots covering different parts of the maze without duplication of effort.

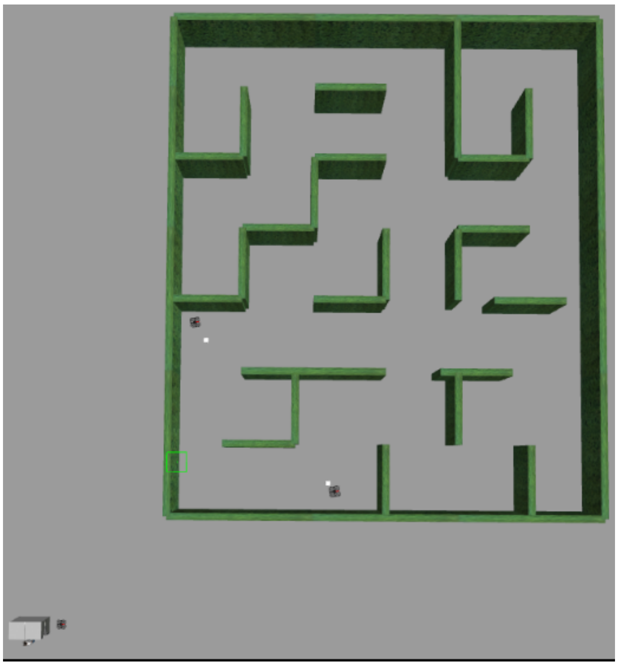


Figure 17. Overhead View of the Gazebo Simulated Environment Featuring the Maze and Initial Positions of Ground Robots(White) and the Drones(Black) on standby

Figure 18. Simulation Control Station in Gazebo, with the Human Operator Avatar Overseeing the Mapping Drones and Ground Robots

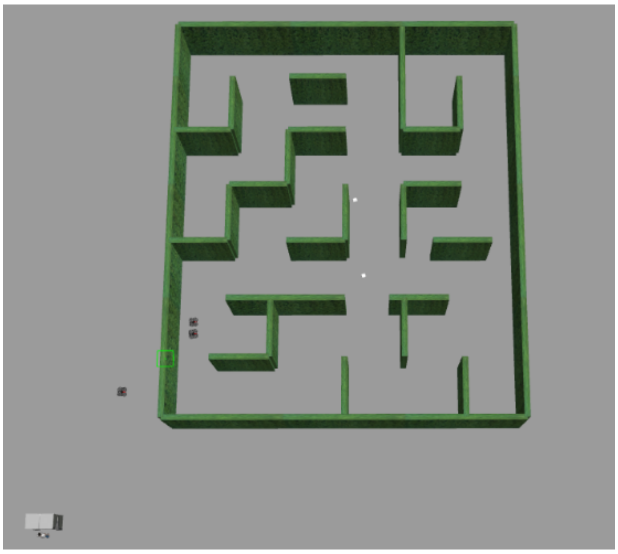


Figure 19. Overhead View of the Gazebo Simulated Environment Featuring the Maze the exploring robots(white) and the Drones(black) at Rendezvous Point

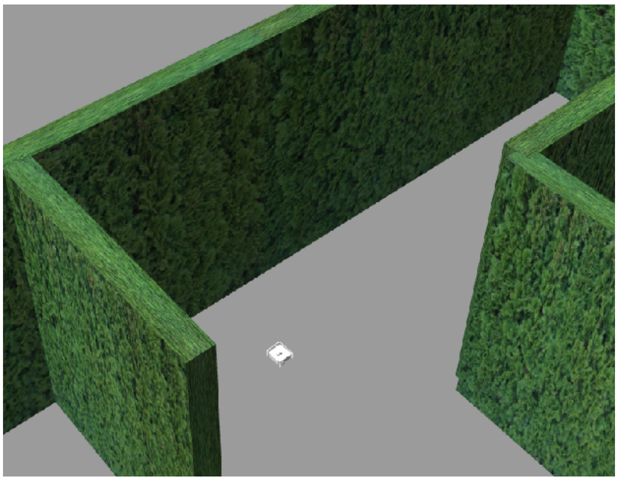


Figure 20. Zoomed View of the Turtlebot3 within the Gazebo Maze, Illustrating the Robot's Scale in Relation to the Surrounding Labyrinth Structure

7 Results and Discussion

This chapter delves into the results obtained from the implementation of the multi-robotic fleet management system described in the previous section. The primary goal of the implementation was to validate the system's functionality and effectiveness in enabling collaborative exploration and mapping by a fleet of robots. The implementation objectives include:

- Using multiple robots for exploration and mapping
- Ensuring continuous, uninterrupted mapping operations.
- Establishing a method for communication relaying.
- Generating a merged map from individual robots' mapping efforts for a comprehensive understanding of the environment

7.1 Assumptions

The implementation and the results operate under several essential assumptions. Initially, it's assumed that the ground robots in a real-life setting are sufficiently large and equipped with batteries of considerable capacity, essentially granting them an almost unlimited operational lifespan for this simulation. Conversely, the drones are not equipped with unlimited battery life. However, they are configured to periodically land on the ground robots for recharging, ensuring they consistently have enough battery power for their operations.

As for communication, it's assumed to have a certain limited range in the simulation, set to an arbitrary value for demonstration purposes. In a practical application, the communication range would vary based on the technology used. Common technologies like Wi-Fi typically offer a range of up to 100 meters outdoors.

7.2 Results

7.2.1 Results of Independent Mapping Trial: Ground Robots Operating Solo

This section examines the outcomes from the first of two distinct trials conducted to assess the mapping strategy. This trial, referred to as the Independent Mapping Trial, involved each ground robot operating in isolation. Relying solely on its own sensors, known location, and the segment of the map it had individually charted, each robot navigated the environment without access to the mapping data of its counterpart. The focus here is to understand the effectiveness and limitations of autonomous mapping when robots work independently, devoid of collaborative data exchange.

Figure 21 and Figure 22 display the starting positions of Robot1 and Robot2, respectively, just before they commence their exploration. As the exploration begins, both robots update their maps simultaneously. After a certain period, having gathered more mapping information, it becomes necessary to dispatch the drones to the rendezvous point. The maps created by each robot before the data-sharing process are depicted in Figures Figure 23 and Figure 24. Following the completion of the data exchange and the return of the operator drone to the base, the merged map is then compiled, as shown in Figure 25.

In Figure 28, the merged map depicted does not precisely match the combination of the maps shown in Figure 26 and Figure 27. This discrepancy is due to the fact that, at the specific moment captured as time T, the operator base's drone had not yet returned to transmit the latest data. Consequently, the integrated map in Figure 28 lacks the most recent updates that would be included once the drone completes its data transmission.

The red lines shown in some of the images indicate the current destination targets determined by the Navigation2 (nav2) system for each robot. These lines act as clear visual cues, indicating the robots' next intended direction within the maze.

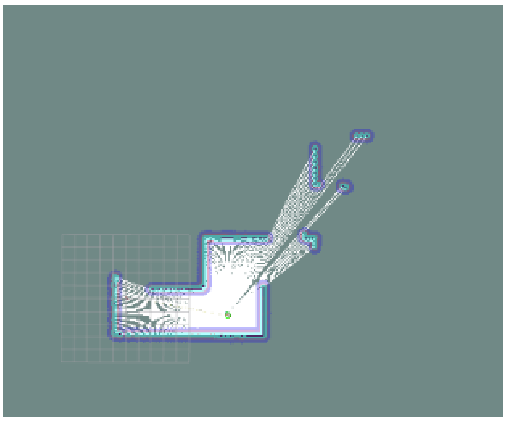


Figure 21. Robot1 Initial Map

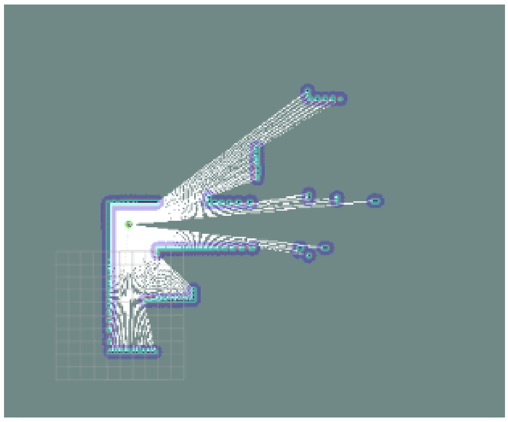


Figure 22. Robot2 Initial Map

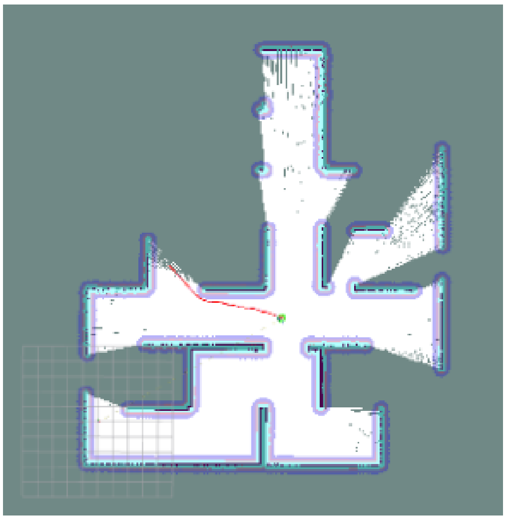


Figure 23. Independent Mapping Trial: Map Generated by Robot1 Prior to the Initial Data Exchange

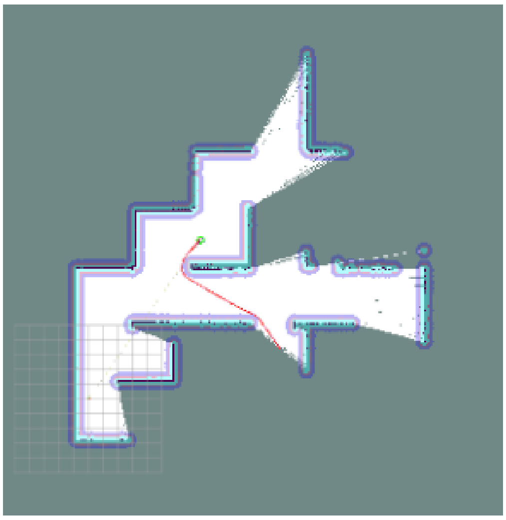


Figure 24. Independent Mapping Trial: Map Generated by Robot2 Prior to the Initial Data Exchange

Figure 25. Independent Mapping Trial: Merged Map after the first rendezvous

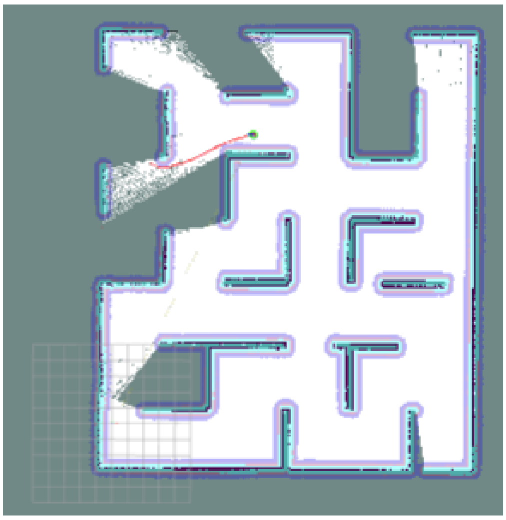


Figure 26. Independent Mapping Trial: Map Generated by Robot1 after a time T of exploration

Figure 27. Independent Mapping Trial: Map Generated by Robot2 after a time T of exploration

Figure 28. Independent Mapping Trial: Merged Map after a time T

7.2.2 Results of Collaborative Mapping Trial: Ground Robots Sharing Data

This part of the study focuses on the results from the second trial, named the Collaborative Mapping Trial. This trial contrasts with the first by enabling ground robots to access each other's mapping data. This data exchange is facilitated through drones that collect and distribute information during their rendezvous points. With the added layer of shared data, the robots are positioned to make more informed decisions in their exploration activities. The primary aim here is to ascertain the enhancements in mapping efficiency and thoroughness when robots utilize collective information to strategically avoid overlapping exploratory efforts.

In the simulation, the approach for utilizing data from another robot during exploration involves integrating the second robot's map with the current robot's map. While this method proves effective within the simulation environment, real-world applications might benefit from alternative strategies that efficiently leverage data from other robots without directly merging the maps.

One such idea involves creating a layered mapping system where each robot's map is maintained separately but referenced against each other. This system would allow each robot to access and review the other's map as an overlay, providing additional context while keeping their primary navigation maps distinct. By doing so, robots can identify areas already covered by their counterparts and focus on unexplored regions, thereby avoiding redundant efforts.

This trial begins with the same initial maps as depicted in Figure 21 and Figure 22. The maps generated by the robots before their first rendezvous are shown in Figure 29 and Figure 30. Additionally, the first merged map, created after this rendezvous, is presented in Figure 31. The key difference in this trial is that upon returning to their respective hosts, the drones of Robot 1 and Robot 2 also disseminate the information gathered from the other robot. The outcomes of this data sharing are illustrated in Figure 32 for Robot 1 and Figure 33 for Robot 2. Notably, despite sharing the same map data at that moment, the robots are observed to be selecting distinct goal frontiers for exploration, as evidenced by the red lines in the images. This highlights the robots' capability to independently identify and navigate toward different exploration points, even when operating with shared mapping information.

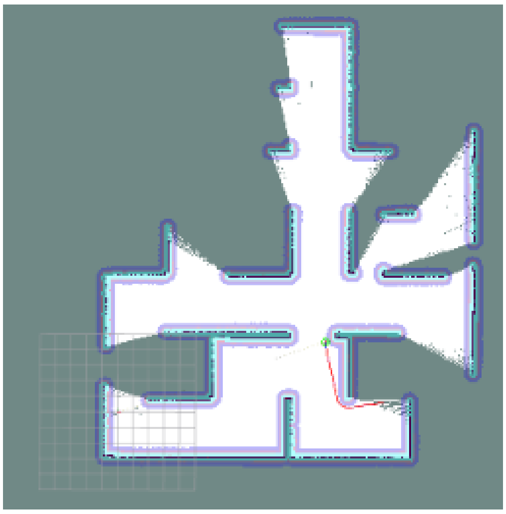


Figure 29. Collaborative Mapping Trial: Map Generated by Robot1 Prior to the to the first rendezvous

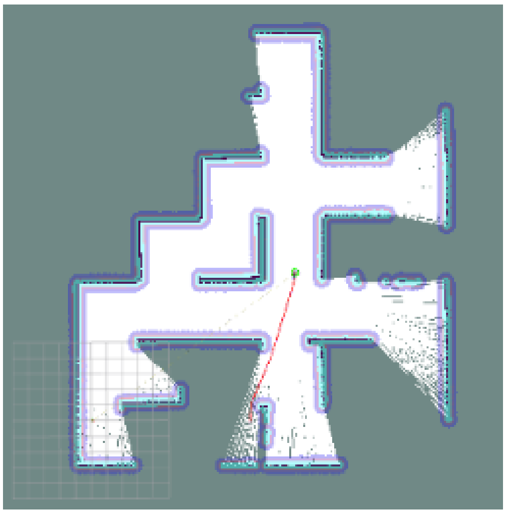


Figure 30. Collaborative Mapping Trial: Map Generated by Robot2 Prior to the first rendezvous

Figure 31. Collaborative Mapping Trial: Merged Map after the first rendezvous

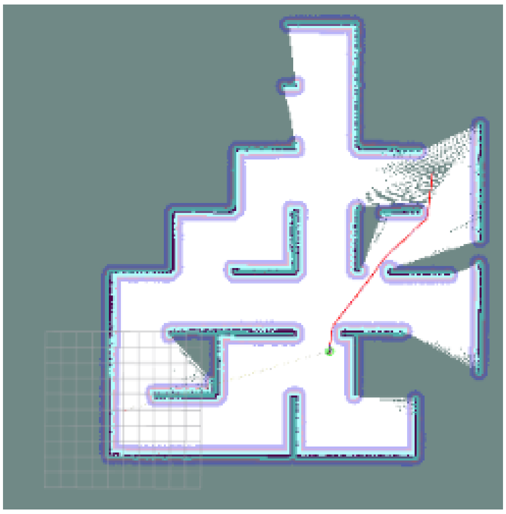


Figure 32. Collaborative Mapping Trial: Robot1 map after receiving the data from Robot2

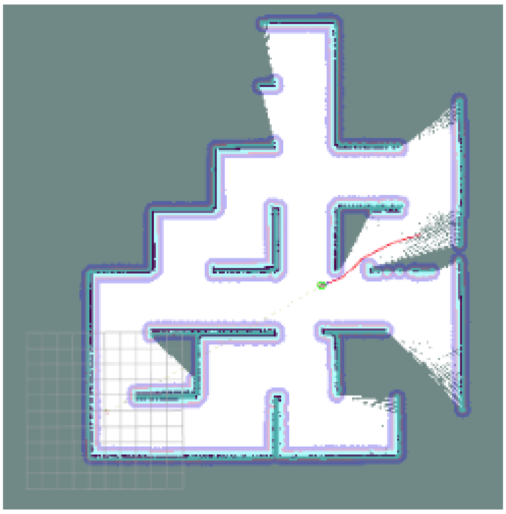


Figure 33. Collaborative Mapping Trial: Robot2 map after receiving the data from Robot1

7.3 Comparative Analysis

In the comparative analysis, each of the two trials was conducted multiple times, over five iterations each, to ensure consistency in the results. A notable difference was observed in the time taken to complete the maze exploration. On average, the trial employing shared data between robots completed the maze exploration in about 5 minutes. In contrast, the trial where robots worked independently, without sharing data, took an average of 8 minutes to complete the same exploration. These average times reflect the differences in efficiency between the two approaches. Despite these differences in completion time, all iterations of both trials were performed under the same conditions, with the robots and drones operating at a predetermined speed and all other variables held constant.

The primary variable between the two trials was the availability of shared data. However, irrespective of this variable, a random factor influenced the exploration process in both scenarios. This randomness stemmed from the computational processing speed of the computer running the simulation. At any given moment, termed as time T, variations in the computer's processing speed could lead to different decisions regarding the choice of the next exploration point or frontier. This phenomenon means that even with two experiments set up identically in terms of parameters and conditions, the exploration paths chosen by the robots might differ. This element of unpredictability underscores the dynamic and complex nature of robotic exploration, highlighting how computational factors can subtly but significantly affect the outcomes of robotic mapping experiments.

8 Conclusion

In this final chapter, we will summarize the main points and outcomes of the research on multi-robot exploration. The focus is on reviewing the development and testing of a new system for mapping unknown areas using multiple robots. This chapter will look back at what the research achieved and it's also a chance to think about what could be done next in this area of study.

8.1 Assessment of the Multi-Robot Fleet Exploration System

This dissertation has effectively demonstrated the creation and implementation of a multi-robot fleet exploration system, aimed at navigating and mapping unfamiliar environments. Central to this system is the innovative use of drones as communication relays, which has improved the efficiency of data sharing among robots during exploration. A key achievement of this system is the generation of a global merged map, offering a comprehensive view of the explored areas, which is beneficial for human operators in understanding the environment.

Through simulated experiments and various trials, the functionality and effectiveness of this system were rigorously tested. These trials focused on different operational modes of the robots, such as independent and collaborative mapping. The results from these experiments have shown that the system is capable of conducting exploration missions without major delays commonly associated with communication issues in multi-robot settings.

8.2 Future work

This project has successfully devised an operational multi-robot fleet management system. However, several limitations and unexplored challenges remain. Thus, a promising avenue lies in the identification of prospective future enhancements. The following paragraphs delve into the prospective avenues for refinement that can be seamlessly integrated to augment the system developed within this dissertation.

A significant enhancement that could be implemented in future iterations of the robotic system is the development of a communication protocol for the robots to share their respective positions. This feature would be particularly beneficial in scenarios where the environment becomes more constricted or narrow. By continuously exchanging location information, the robots could actively avoid collisions between each other, ensuring smoother and safer navigation. This improvement would not only increase the efficiency of the exploration process but also enhance the overall safety of the operation, especially in environments with unpredictable or tight spaces. The ability to dynamically adjust paths based on the real-time positions of other robots in the proximity would add a layer of intelligence and adaptability to the system, making it more robust and reliable in complex terrain. Such a feature would represent a significant step forward in autonomous robotic exploration and mapping technologies.

Although the frontier exploration algorithm was not the central focus of this dissertation, the current method employed for shared map data between the two ground robots, where one robot selects the second closest point of interest to avoid converging on the same goal as the other, highlights an area ripe for improvement. Significant enhancement in the efficiency of the mapping process could be achieved by developing an algorithm specifically tailored for collaborative decision-making in frontier exploration.

This specialized algorithm would focus on optimizing the decision process when selecting target points for exploration, particularly when multiple robots share mapping data. Instead of simply defaulting to the second closest frontier, the algorithm could analyze a range of factors such as the robots' current positions, their remaining battery life, the terrain's complexity, and areas already covered. By considering these variables, the algorithm would dynamically allocate exploration targets in a manner that maximizes coverage and minimizes overlap.

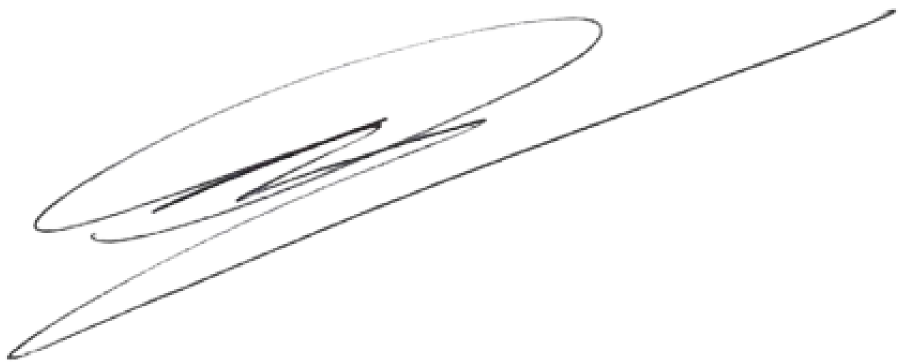
Acknowledgement

I would like to express my heartfelt gratitude and appreciation to the individuals who have played an instrumental role in the successful completion of my Master's dissertation at Tartu University.

First and foremost, I extend my deepest thanks to my supervisor, Dr Naveed Muhammad for his unwavering support, guidance, and mentorship throughout this academic journey. Dr Muhammad's insightful feedback, constructive criticism, and expert advice have been invaluable in shaping the direction of my research and refining the quality of my work. His dedication and enthusiasm have truly been inspiring.

I would also like to express my gratitude to my family and friends for their unwavering support, patience, and understanding throughout this journey. Their encouragement and belief in my abilities have been my constant motivation.

I would like to extend my thanks to the OpenAI team for providing the ChatGPT tool, which assisted me in accurately rephrasing sentences and selecting suitable vocabulary throughout the writing of this thesis.



References

[1] Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010.

[2] University of tartu guidelines for using chatgpt. `https://ut.ee/en/content/university-tartu-guidelines-using-chatgpt-are-now-available`. Accessed: 2023.

[3] OpenAI. Chatgpt: Language model for dialogue. `https://openai.com/chatgpt`, 2023. Accessed: 2023.

[4] Robert Sparrow and Mark Howard. Robots in agriculture: prospects, impacts, ethics, and policy. *Precision Agriculture*, 22, 06 2021.

[5] Robert Bogue. Robots in the nuclear industry: A review of technologies and applications. *Industrial Robot-an International Journal - IND ROBOT*, 38:113–118, 03 2011.

[6] Chris Dinelli, John Racette, Mario Escarcega, Simon Lotero, Jeffrey Gordon, James Montoya, Chase Dunaway, Vasileios Androulakis, Hassan Khaniani, Sihua Shao, Pedram Roghanchi, and Mostafa Hassanalian. Configurations and applications of multi-agent hybrid drone/unmanned ground vehicle for underground environments: A review. *Drones*, 7(2), 2023.

[7] Jonathan M. Aitken, Mathew H. Evans, Rob Worley, Sarah Edwards, Rui Zhang, Tony Dodd, Lyudmila Mihaylova, and Sean R. Anderson. Simultaneous localization and mapping for inspection robots in water and sewer pipe networks: A review. *IEEE Access*, 9:140173–140198, 2021.

[8] Björn Lindqvist, Samuel Karlsson, Anton Koval, Ilias Tevetzidis, Jakub Haluška, Christoforos Kanellakis, Ali akbar Agha-mohammadi, and George Nikolakopoulos. Multimodality robotic systems: Integrated combined legged-aerial mobility for subterranean search-and-rescue. *Robotics and Autonomous Systems*, 154:104134, 2022.

[9] Bingwei Tian, Wenrui Liu, Haozhou Mo, Wang Li, Yuting Wang, and Basanta Raj Adhikari. Detecting the unseen: Understanding the mechanisms and working principles of earthquake sensors. *Sensors*, 23(11), 2023.

[10] Raúl Mur-Artal and Juan D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[11] Seoungjun Lee, Dongsoo Har, and Dongsuk Kum. Drone-assisted disaster management: Finding victims via infrared camera and lidar sensor fusion. In *2016 3rd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE)*, pages 84–89, 2016.

[12] Titus Cieslewski, Siddharth Choudhary, and Davide Scaramuzza. Data-efficient decentralized visual slam. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2466–2473, 2018.

[13] Man Liang and Daniel Delahaye. Drone fleet deployment strategy for large scale agriculture and forestry surveying. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 4495–4500, 2019.

[14] Sajad Saeedi, Michael Trentini, Mae Seto, and Howard Li. Multiple-robot simultaneous localization and mapping: A review. *Journal of Field Robotics*, 33(1):3–46, 2016.

[15] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. *IEEE Robotics Automation Magazine*, 13(2):99–110, 2006.

[16] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (slam): part ii. *IEEE Robotics Automation Magazine*, 13(3):108–117, 2006.

[17] Raúl Mur-Artal and Juan D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[18] Hussein A. Daoud, Ahmad Q. Md. Sabri, Chee K. Loo, and Abdul M. Mansoor. Slamm: Visual monocular slam with continuous mapping using multiple maps. *PLoS ONE*, 13(4), 2018.

[19] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[20] `https://www.parrot.com/en`, 2023. Accessed: May 12, 2023.

[21] Vít Krátký, Pavel Petráček, Tomáš Báča, and Martin Saska. An autonomous unmanned aerial vehicle system for fast exploration of large complex indoor environments. *Journal of Field Robotics*, 38(8):1036–1058, 2021.

[22] Xiaobin Xu, Lei Zhang, Jian Yang, Chenfei Cao, Wen Wang, Yingying Ran, Zhiying Tan, and Minzhou Luo. A review of multi-sensor fusion slam systems based on 3d lidar. *Remote Sensing*, 14(12), 2022.

[23] Hao Xu, Peize Liu, Xinyi Chen, and Shaojie Shen. d^2slam: Decentralized and distributed collaborative visual-inertial slam system for aerial swarm, 2023.

[24] Václav Pritzl, Matouš Vrba, Petr Štěpán, and Martin Saska. Cooperative navigation and guidance of a micro-scale aerial vehicle by an accompanying uav using 3d lidar relative localization. In *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 526–535, 2022.

[25] Pavel Petráček, Vít Krátký, Matěj Petrlík, Tomáš Báča, Radim Kratochvíl, and Martin Saska. Large-scale exploration of cave environments by unmanned aerial vehicles. *IEEE Robotics and Automation Letters*, 6(4):7596–7603, 2021.

[26] Seoungjun Lee, Dongsoo Har, and Dongsuk Kum. Drone-assisted disaster management: Finding victims via infrared camera and lidar sensor fusion. In *2016 3rd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE)*, pages 84–89, 2016.

[27] Christian Forster, Simon Lynen, Laurent Kneip, and Davide Scaramuzza. Collaborative monocular slam with multiple micro aerial vehicles. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3962–3970, 2013.

[28] Joshua J. Morales, Joe J. Khalife, and Zaher M. Kassas. Information fusion strategies for collaborative inertial radio slam. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):12935–12952, 2022.

[29] Youngseok Jang, Changsuk Oh, Yunwoo Lee, and H. Jin Kim. Multirobot collaborative monocular slam utilizing rendezvous. *IEEE Transactions on Robotics*, 37(5):1469–1486, 2021.

[30] Pierre-Yves Lajoie, Benjamin Ramtoula, Yun Chang, Luca Carlone, and Giovanni Beltrame. Door-slam: Distributed, online, and outlier resilient slam for robotic teams. *IEEE Robotics and Automation Letters*, 5(2):1656–1663, 2020.

[31] Carlo Pinciroli and Giovanni Beltrame. Buzz: An extensible programming language for heterogeneous swarm robotics. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3794–3800, 2016.

[32] Matthew A. Schack, John G. Rogers, Qi Han, and Neil T. Dantam. Optimization-based robot team exploration considering attrition and communication constraints. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5864–5871, 2021.

[33] Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. Robot operating system 2: Design, architecture, and uses in the wild. *Science Robotics*, 7(66):eabm6074, 2022.

[34] Open Robotics. Gazebo: A multi-robot simulator for outdoor environments. `http://gazebosim.org`, 2021.

[35] Open Robotics. Rviz: 3d visualization tool for ros. `http://wiki.ros.org/rviz`, 2021.

[36] Steve Macenski. Slam toolbox: Open-source, real-time simultaneous localization and mapping for ros 2 robots. `https://github.com/SteveMacenski/slam_toolbox`, 2021.

[37] Steve Macenski and Ivona Jambrecic. Slam toolbox: Slam for the dynamic world. *Journal of Open Source Software*, 6(61):2783, 2021.

[38] Steve Macenski, Francisco Martin, Ruffin White, and Jonatan Gines Clavero. The marathon 2: A navigation system. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, oct 2020.

II. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Malcom Radigon**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   "Continuous Collaborative Mapping in Unknown Environments: A Multi-Robot System Approach",

   supervised by Dr Naveed Muhammad.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Malcom Radigon
23/12/2023