Dual-Robotic System for Autonomous Search and Rescue in Simulated Office Environment

By

Saravanan, Aravind Sairam

2264341

MSc Robotics Dissertation

Department of Engineering Mathematics

UNIVERSITY OF BRISTOL

&

Department of Engineering Design and Mathematics

UNIVERSITY OF THE WEST OF ENGLAND

A MSc dissertation submitted to the University of Bristol and the University of the West of England in accordance with the requirements of the degree of MASTER OF SCIENCE IN ROBOTICS in the Faculty of Engineering.

September 5, 2023

Declaration of own work

I declare that the work in this MSc dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

Name and Date
Aravind Sairam Saravanan
05/09/2023

Acknowledgments

I would like to extend my deepest gratitude to my supervisor, Dr. Hamidreza Nemati, for his invaluable guidance and support throughout the course of this research. His expertise and insights were instrumental in shaping this project, and his availability to address my questions in progress meetings was deeply appreciated.

I am also thankful to the MSc Robotics department for providing a conducive environment for my research endeavors. The faculty members have been supportive and provided me with the resources necessary for carrying out this project successfully.

Last but not least, I wish to thank my friends for their constant motivation and unwavering support, especially during the setbacks and challenges I faced while experimenting with the processes involved in this study. Your encouragement made this journey both possible and rewarding.

Abstract

This study comprehensively evaluated the navigational and task-specific capabilities of two types of robots: a Husky robot and TurtleBots. Conducted in a simulated office environment, the research was guided by a set of predefined hypotheses targeting key performance indicators. For the Husky robot, the focus was on assessing localization accuracy and obstacle avoidance. In contrast, the TurtleBots were evaluated based on their ability to detect and follow human targets and successfully return to home positions. Methodologies employed included probabilistic sensor data integration for the Husky and the use of the OpenCV DNN module for human detection in TurtleBots. The findings indicate that the Husky robot demonstrated a localization error marginally above the hypothesized 1-meter threshold, implicating potential challenges in sensor fusion and algorithmic constraints. The TurtleBots exceeded expectations in human detection accuracy but fell short in achieving a consistent return-to-home rate, thus highlighting the need for further optimization in their navigational algorithms. Overall, the results suggest that while both types of robots possess promising capabilities for tasks such as surveillance and search and rescue missions, specific technical improvements are imperative for more demanding, real-world applications. The study concludes with targeted recommendations, emphasizing the integration of advanced machine learning techniques and more robust sensor fusion methods to overcome identified limitations.

Number of words in the dissertation: 6672 words.

Contents

List of Tables

1 Introduction

In contemporary workplace settings, the risk of fire disasters continues to exist despite the progress that has been made in safety measures. According to the National Fire Protection Association (NFPA), an estimated average of 3,340 fires occurred in office properties per year between 2007-2011, resulting in an annual average of four deaths, 44 injuries, and $112 million in property damage [1]. Additionally, a study by the U.S. Fire Administration (USFA) states that each year, an estimated 86,500 non-residential building fires are reported to fire departments, causing 85 deaths, 1,325 injuries, and $2.6 billion losses [2]. About 20% of these incidents happen in office buildings.In the event of a fire, conventional search and rescue techniques can prove to be ineffective and perilous, thereby endangering both emergency personnel and individuals in need of assistance. There exists a pressing demand for novel solutions capable of effectively addressing these deficiencies.

The research is motivated by an urgent need to improve the effectiveness and security of disaster management systems, specifically in office environments susceptible to fires. The utilisation of robotics presents a distinct opportunity to minimise the inherent hazards and enhance the efficiency of rescue operations. The objective is to utilise sophisticated technologies as a necessary supplement rather than a substitute for human labour, so establishing a synergistic framework that can enhance the effectiveness of life-saving endeavours.

In an era characterised by remarkable technical progress, robotics and artificial intelligence have emerged as prominent instruments for enhancing societal welfare. An area that necessitates significant innovation is disaster management, particularly in office buildings that are susceptible to fires. Although current fire safety procedures provide certain advantages, they frequently demonstrate insufficiency when confronted with unforeseen disasters. Conventional search and rescue operations, while their commendable nature, have notable hazards and inefficiencies that expose both rescuers and victims to precarious circumstances.

The objective of this research is to address these issues by the implementation of an integrated, two-stage robotic system specifically developed for the purpose of disaster management. The system endeavours to enhance safety and efficiency by utilising a Husky robot for initial mapping and a swarm of TurtleBots for search and rescue, thereby offering an innovative alternative to traditional approaches. Different types of robots have specific roles in their operations. The Husky robot is primarily designed for the purpose of mapping unfamiliar area. On the other hand, the TurtleBots are equipped with capabilities to identify humans and are utilised for navigating around the previously mapped environment. Their main objective is to locate individuals and provide guidance to ensure their safety.

1.1 Aims & Objectives

The primary objective of this study is to create, execute, and evaluate a comprehensive, multi-robotic framework that will greatly improve the efficacy, productivity, and security of search and rescue missions in office settings that are vulnerable to fire emergencies. This research aims to do the following:

1. Investigate capabilities and limitations of current mapless navigation algorithms for guiding a Husky robot through unknown, complex terrains.
2. Examine feasibility of using TurtleBots for autonomous human detection and guidance after mapping environment.
3. Examine the cooperative performance of a Husky robot and TurtleBots in a model office setting, concentrating on metrics that have practical applicability.

In order to achieve these aims, the following objectives have been established:

1. Mapless Navigation Development for Husky:
    (a) Design and implement a mapless navigation algorithm for the Husky robot using ROS (Robotic Operating System).
    (b) Evaluate the performance of the Husky robot in traversing unknown environments and mapping them effectively.

2. TurtleBot's Implementation:

    (a) Design and simulate a swarm of 20 TurtleBots equipped with LIDAR and cameras for rescue operation.

    (b) Implement human detection algorithms using OpenCV DNN module.

3. Performance Metrics for TurtleBots:

    (a) Evaluate TurtleBot's effectiveness using metrics like Detection Accuracy, Guidance Accuracy, Return-to-home Accuracy and Stuck & Recovery Rate.

4. Simulate and assess the integrated system's overall efficiency and effectiveness in locating and guiding humans to safety.

5. Incorporate real-time obstacle avoidance in TurtleBots.

6. Develop and test algorithms for path planning and human guidance to exits.

7. Assess the potential impact of this system on existing search and rescue operations and explore possibilities for future improvements.

The Husky robot is primarily designed for the purpose of mapping unfamiliar areas. The accuracy of the created maps and time taken to complete it will serve as a measure for effectiveness in this aspect. Specifically, the Husky robot is expected to navigate to designated way points with an average localization error below 1 meter and to perform better in environments with fewer obstacles. On the other hand, the TurtleBots are equipped with capabilities to identify humans and are utilized for navigating around the previously mapped environment. Their main objective is to locate individuals and provide guidance to ensure their safety. Effectiveness in this context will be evaluated using metrics such as Detection Accuracy, Guidance Accuracy, Return-to-home Accuracy, and Stuck & Recovery Rate.

The expected outcomes for these evaluations include the TurtleBots successfully detecting and following a human target at least 75% of the time and demonstrating a return-to-home success rate of over 80% in controlled, known environments. The success metrics not only contribute to verifying the hypothesis but also to filling the existing research gap in robotic-assisted search and rescue. While traditional methods are less precise and riskier, the introduction of a robotic framework aims to quantifiably improve both the safety and the efficacy of such operations.

By achieving these objectives, this research will provide a foundational study for the use of integrated robotic systems in disaster management. This is particularly crucial given the deficiencies of current strategies in mitigating risks and optimizing rescue during fire emergencies in office settings.

2 Literature Review

The rapidly evolving landscape of search and rescue robotics presents an intricate mosaic of technological innovations, problem-solving algorithms, and human-centric designs. This expanding field aims to provide versatile solutions to an array of challenges, from navigating treacherous terrains to enhancing real-time decision-making capabilities, all with the ultimate goal of saving human lives and minimizing risks [3]–[5]

The initial adoption of search and rescue robotics was made prominent by their deployment in the aftermath of the 9/11 attacks. Conventional human-led efforts faced monumental challenges due to unstable wreckage and risk of additional collapses. To mitigate these risks, robots were introduced to the rescue operations. These robots, equipped with cameras and chemical detectors, traversed the hazardous and confined rubble to provide real-time video feeds and data, aiding human operators in assessing structural conditions and locating survivors. This real-world example, one of the first large-scale deployments of search and rescue robots[6].

Firstly, the physical settings in which these robots operate are incredibly diverse. They can be deployed in complex urban infrastructures where there are multiple layers of constructed environments[7]. In these settings, the robots can navigate through collapsed buildings, tight spaces, and confined areas that are otherwise inaccessible or too dangerous for human rescuers. Robots are also utilized in wilderness settings for tasks like wildlife monitoring, offering yet another application of their capabilities[8].

The variety of terrains and contexts necessitates a highly adaptive sensory apparatus. LiDAR sensors have emerged as a reliable technology for precise distance measurement, providing robots with a depth perception that closely mimics human vision[9]. In addition to LiDAR, ultrasonic sensors offer a more cost-effective means of object detection, allowing for the broad deployment of robotic systems even in resource-constrained settings[10]. Passive Infrared (PIR) sensors give these robots the ability to detect life forms by sensing heat, a crucial feature for locating survivors in search and rescue missions[3]. Depth sensors like Kinect extend these capabilities, especially useful for indoor navigation where conventional GPS systems are ineffective[11].

However, gathering raw sensor data is just the initial step; making sense of this data to facilitate real-time decision-making is where machine learning algorithms come into play [9], [12], [13]. These algorithms process sensor inputs to recognize complex patterns, classify objects, and even predict future events based on historical data. For example, machine learning has been integrated into the robot's systems to identify whether an object is a simple obstacle or a life-threatening danger, enabling the robot to decide whether to avoid it or approach for further investigation[13].

As we delve into machine learning, it's evident that advances in computer vision technologies are accelerating the field's progress. Techniques like YOLOv4 and OpenCV's Deep Neural Network (DNN) modules are no longer confined to academic papers but are being actively integrated into operational robotic systems[14], [15]. Specifically, these technologies have been adapted for high-accuracy detection tasks. For instance, YOLOv4 and its variants have been demonstrated to accurately detect vehicles and motorbike riders without helmets in real-time traffic, adding another layer of safety and responsiveness to search and rescue operations[15].

When it comes to multi-robot coordination and control, communication systems act as the neural networks connecting the various agents. Protocols like Zigbee have been integrated into robotic systems to enable real-time data sharing, essential for coordinating efforts in obstructed or densely populated areas [16]. RFID technology serves as an augmentative feature, offering additional tracking and localization capabilities which are vital in dynamically changing environments where conventional methods may fall short[11], [17].

Algorithms and computational models play a foundational role in adding layers of autonomy and adaptability. The traditional PID controllers have evolved to more complex mathematical models like mixed-integer linear programming (MIP) for effective multi-agent coordination[5], [18]. These algorithms are designed to not only allow for dynamic path planning but also make on-the-fly adjustments based on real-time data[8], [15].

The effort to universalize performance standards is also gaining traction. These would encompass everything from hardware and sensor capabilities to machine intelligence metrics, aiding in the development of innovative designs like mobile-snake robot configurations[19], [20]. The concept of mapless navigation—where the robot doesn't rely on pre-existing maps but uses its sensors and machine learning algorithms to navigate—is becoming increasingly relevant. Here, biological inspirations like place cells provide an internal measure of localization, allowing robots to function in large outdoor environments without the need for Cartesian or topological maps[21], [22].

Moreover, the domain is starting to integrate traditional and modern search algorithms to guide robots more effectively in locating survivors. These algorithms are scalable, meaning that they can be adapted for use by multiple robots to expedite the search process[23]. The algorithms, grounded in nonlinear mathematical models, allow quick and repeated testing without practical installation, which is crucial for rapid deployment in disaster situations.

Lastly, as these individual components continue to mature, integrating them into a unified, user-friendly system remains a significant challenge and an area of active research. From developing intuitive user interfaces to rigorous usability studies, the human factor is an integral part of this technological ecosystem[24], [25]. Specialized test arenas are being developed to mimic real-world conditions, offering objective ways to evaluate the capabilities of these robots[26].

In summary, search and rescue robotics is converging towards an integrated, highly functional, and adaptive system that can operate in dynamically challenging environments. This integration covers a vast range of disciplines—from hardware and sensor technology to software algorithms and human -machine interfaces. The ongoing research spans a broad spectrum, but it is collectively moving the field closer to developing robots that are not just functional tools but also reliable partners in life-saving missions.

3 Research Methodology

The research makes use of Robotic Operating System (ROS) for the purpose of search and rescue operation using Husky robot and TurtleBots. The chosen environment is a ROS package designed in Gazebo, containing the layout of an office and models of humans spread across various locations. The environment is static and does not include real-time obstacles. The layout of the environment and the position of human models is unknown initially as it would be the case in real-life scenarios. The Husky robots are made to explore using mapless navigation algorithm to generate a fixed map of the environment which is then fed to the swarm of TurtleBots to search and rescue the human beings by guiding them to the safe exit.

The Husky robot was chosen for initial mapping primarily for its robustness, adaptability, and high sensor fidelity. Its durable build makes it ideal for navigating complex terrains, ensuring reliable data collection. The robot's seamless integration with ROS streamlines operational flow, while its ability to perform mapless navigation provides flexibility in dynamic environments[27]. Together, these attributes make Husky a natural choice for the crucial task of creating an initial map for further navigation by the TurtleBots.

Both the Husky and TurtleBot platforms utilise the move_base controller provided by ROS for their navigational methods. The computational model undertakes a range of complex tasks, involving path planning and real-time obstacle avoidance[28]. The Husky robot traverses the area by utilising a mapless navigation algorithm to generate an environmental map of the office space. The mapping systems is subjected to a detailed performance evaluation, which involves assessing many metrics such as localization accuracy, collision avoidance, and path length. This evaluation process enhances the robustness of the study.

TurtleBots are selected for use in rescue scenarios due to their compact size, maneuverability, and cost-effectiveness. Their small footprint allows them to navigate through narrow spaces and challenging terrains, which are common in disaster-stricken areas. Furthermore, their open-source software stack and modular design make them highly customisable, enabling the integration of specialized sensors and software required for rescue missions.

In the context of rescue operations, a swarm of TurtleBots offers distinct advantages over a single robot. The swarm approach increases the system's robustness and fault tolerance, as the failure of a single unit does not halt the entire operation. It also enhances the efficiency of search and rescue tasks by enabling parallelism; multiple robots can explore different areas simultaneously, thereby speeding up the identification of victims and hazards.

Table 3.1: Comparison of Husky Robot with Pioneer 3-DX and Jackal UGV, Focusing on Aspects Essential for Efficient Mapping in Unknown Environments

Criteria	**Husky**	**Pioneer 3-DX**	**Jackal UGV**
Sensory Capabilities	High-end sensors	Limited sensors	Moderate sensors
Payload Capacity (kg)	75	23	20
Robustness	High	Low	Moderate
Software Integration	ROS-compatible	Limited compatibility	ROS-compatible

Table 3.2: Comparative Analysis of TurtleBots with e-puck and Khepera IV, Evaluating Suitability for Search and Rescue Operations in Terms of Cost, Agility, and Software Compatibility

Criteria	**TurtleBots**	**e-puck**	**Khepera IV**
Cost-Effectiveness	High	Low	Moderate
Agility and Size	Compact	Compact	Bulky
Sensor Compatibility	High	Low	Moderate
Software Integration	ROS-compatible	Limited compatibility	ROS-compatible

The navigational system of TurtleBots incorporates the A* algorithm and Adaptive Monte Carlo localization[29], which was developed for the purpose of path planning spanning the shortest route. Every TurtleBot is equipped with an onboard LIDAR sensor system, which plays a crucial role in supplying essential data inputs to the move_base algorithm for the purpose of navigating obstacles in real-time[30].

The process of detecting humans is carried out by integrating OpenCV DNN module through the RGB camera, which employs the YOLOv4-tiny (You Only Look Once) algorithm[31]. The

Table 3.3: Comparison of Relevant Hardware Specifications for Rescue Scenario

Features	**Husky**	**TurtleBot**
Mobility	4-wheel drive	2-wheel drive with caster
Sensors	3D LIDAR, Thermal Camera	2D LIDAR, RGB-D Camera
Power Source	24V Li-ion Battery (8 hours)	12V Li-ion Battery (3 hours)
Payload Capacity	75kg (Equipment and supplies)	5kg (First aid kit)
Navigation	GPS, IMU	Wheel Encoders, IMU
Communication	Wi-Fi, LTE	Wi-Fi
Form Factor	Larger, Rugged	Smaller, Agile
Obstacle Clearance	High	Low
Software Support	ROS-based Swarm Coordination	ROS-based Swarm Coordination

complex assemblage serves the purpose of facilitating swift and accurate recognition of isolated persons inside the simulated setting[32]. Furthermore, the performance of the system was measured using metrics like detection accuracy, the success rate of rescues, and time-to-rescue, among others. The metrics provided serve as a comprehensive evaluation pertaining to the effectiveness of the algorithmic performance and operational efficiency of the robotic swarm in situations involving the rescue of human individuals.

3.1 Simulation Environment

For an indoor search and rescue mission, the chosen environment is an office space with human models placed meticulously in the corners3.1. The virtual office environment has minimal obstacles and lacks real-time dynamic elements or simulated fire scenarios[33]. It primarily focuses to serve as a controlled platform for testing the Husky robot and a swarm of 20 TurtleBots on tasks like mapping, navigation, and human detection in a disaster management context.

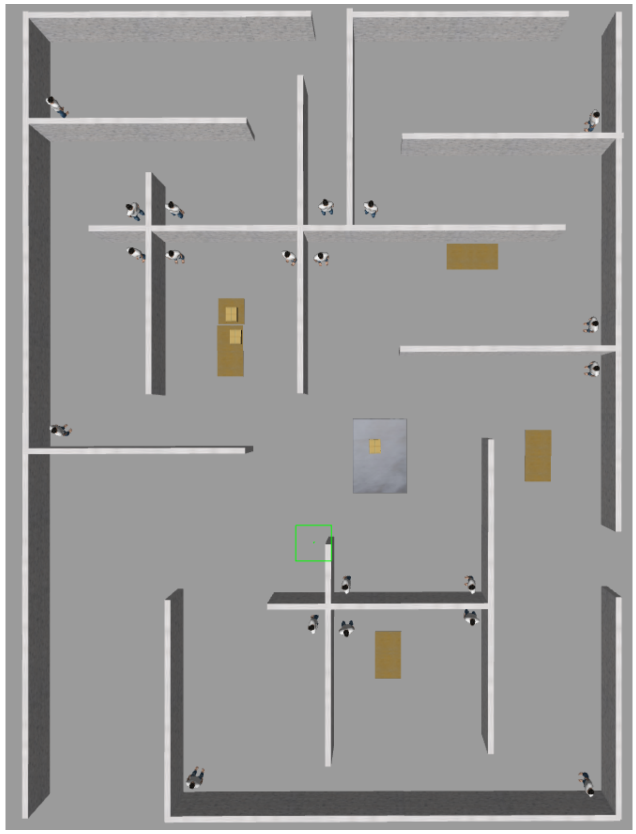


Figure 3.1: Gazebo Office Environment

3.2 Robot Platform & Functionalities

This research leverages two distinct types of robots to perform a complete search and rescue operation: the Husky robot and a swarm of 20 TurtleBots. Each of these platforms is equipped with specialised functionalities and algorithms to execute specific tasks.

3.2.1 Husky Robot

The Husky robot3.2 serves as the primary unit responsible for initial navigation and mapping of the simulated office environment3.1. Utilizing the move_base algorithm within ROS, the Husky conducts mapless navigation, meaning it navigates without the need for a pre-built environmental map3.4. This navigation mode is initiated allowing Husky to use its sensors and algorithms to autonomously navigate the environment.

When the Husky is deployed, the `move_base` algorithm initializes with the specified global and local planners, which in this case are `navfn/NavfnROS` and `dwa_local_planner/DWAPlannerROS` respectively. These planners are fed configurations from external YAML files, allowing them to operate under predefined settings tailored for mapless navigation. The local costmap's dimensions are explicitly set to 10×10 meters, while the global costmap dimensions are set to 100×100 meters if `no_static_map` is `true`, ensuring that the robot is not bound by any prior map constraints.

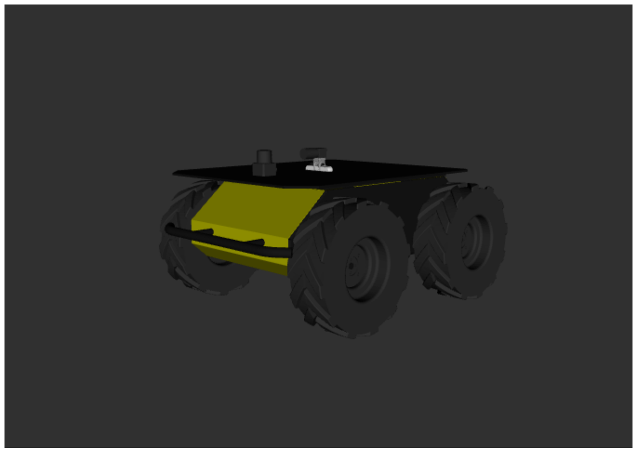


Figure 3.2: Husky Robot

In this mode, the Husky uses its laser scanners to continually update its local and global costmaps. As the robot moves, `move_base` employs its global planner to determine a safe and efficient path based on the real-time costmap data, while the local planner refines this path considering more immediate spatial constraints[30]. This allows the Husky to successfully navigate through unknown terrain, mapping the environment dynamically as it goes. The system is configured to be adaptive, adjusting its costmaps and plans in real-time based on sensor input, making it apt for navigation in unknown or changing environments.

3.2.2 TurtleBots

A swarm of 20 TurtleBots is deployed once the initial mapping is completed by the Husky robot. These TurtleBots are equipped with LIDAR sensors and onboard cameras, which feed into the move_base algorithm for real-time navigation and obstacle avoidance[30]. For path planning the bots use A* algorithm, guiding the stranded humans to evacuate in the shortest path to safe exit[29]. Each TurtleBot integrates a RGB camera using OpenCV DNN library with YOLOv4-tiny[31], ensuring rapid and precise identification of individuals within the simulated environment. Adaptive Monte Carlo localization is used to maintain the accuracy of each robot's position within the map.

The TurtleBots operate based on a predefined algorithmic sequence. They start by planning a path from their spawn locations to specific way points3.3a. Upon reaching these way points, they initiate human detection procedures3.3b. If a human is detected, the robots plan the most efficient path to guide the individual to the nearest safe point, the bot's home location3.3c. Subsequently, all the bots return to their spawn locations, marking the completion of their rescue mission.

(a) Start of the simulation

(b) Process of the simulation

(c) End of the simulation

Figure 3.3: Simulation in Rviz for rescuing operation using a swarm of TurtleBots.

3.3 Navigation Algorithms

The navigational capabilities of the robots serve as a critical aspect of this research, enabling the robots to move effectively within the complex boundary of the simulated environment. The robots must be intelligent to plan its path through the environment as well as avoid any obstacles in order to achieve the chosen objectives, namely mapping the unknown environment and conducting human rescue operations by guiding them to a safe exit.

3.3.1 A* Algorithm

The A* (A-Star) algorithm is a pivotal component of the navigation stack for the TurtleBots, serving as the primary path-planning algorithm in this research. A* is renowned for its efficiency and accuracy in finding the shortest path in a graph between an initial node and a goal node while considering various constraints[29]. The algorithm employs a cost function $f(x) = g(x) + h(x)$, where $g(x)$ represents the exact cost of the path from the starting point to any vertex x, and $h(x)$ is the heuristic estimate of the cost from vertex x to the goal. These cost calculations incorporate real-time data from the TurtleBot's onboard LIDAR sensor, allowing for dynamic obstacle avoidance.

The algorithm continues to explore and expand nodes, constantly updating the costs, until it reaches the goal node. Once the goal is reached, the A* algorithm backtracks from the destination node to the initial node, creating an optimal path for the TurtleBot to follow. This path is then sent to the `move_base` controller, which handles the robot's movements and any real-time obstacle avoidance needed as it navigates along this path.

3.3.2 move_base Controller

The `move_base` package in the Robot Operating System (ROS) serves as a pivotal navigation controller, capable of interfacing with both the Husky and TurtleBot platforms. It acts as a nexus between higher-level navigational tasks, such as path planning and localization, and lower-level actuation functions that directly command the robots' movements. The `move_base` package operates by taking in a goal pose, typically specified in the form of a 2D coordinate and orientation within the map frame, and proceeds to generate an optimal path to achieve this goal.

Internally, `move_base` utilises a layered costmap approach, which enables the robots to understand their surrounding environment and navigate around obstacles dynamically. The costmap is generally populated with sensor data, thus allowing `move_base` to integrate various sensory inputs, such as LIDAR scans, into a coherent map of the environment[30]. Subsequently, `move_base` interacts with both local and global planners to compute the most effective path from the current robot pose to the desired goal pose.

The `move_base` controller serves a critical role in the navigation of both the Husky and the swarm of TurtleBots. In case of Husky robot, `move_base` is deployed in tandem with mapless nav-

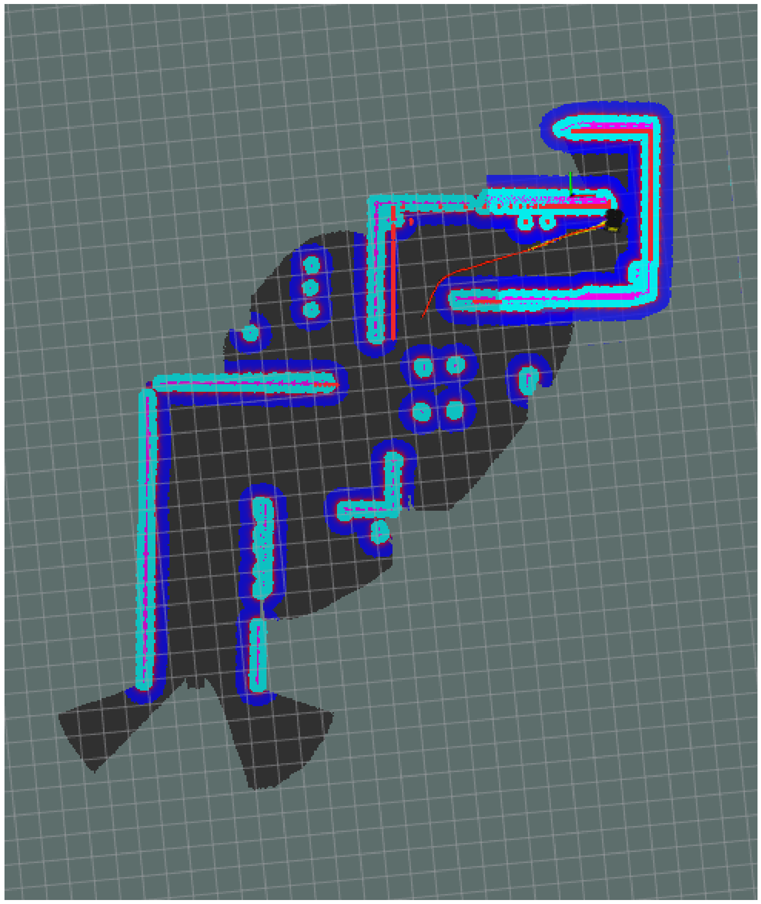


Figure 3.4: Rviz interface displaying the Husky robot's autonomous navigation through an unknown office environment, using LIDAR and the move_base controller for path planning and navigation.

igation, whereby the robot traverses the environment without relying on a pre-generated map. An argument is passed to the `move_base` node, setting the `no_static_map` parameter to `true`, which instructs the Husky to operate in a mapless fashion. This facilitates the Husky's primary objective of initial navigation and mapping, as it scours the simulated office environment to dynamically create a map, which will later serve as a reference for the TurtleBots3.4.

For the TurtleBots, the role of `move_base` is somewhat different but equally vital. Here, `move_base` is configured to operate in conjunction with the A* path planning algorithm, Adaptive Monte Carlo Localization (AMCL) for localization, and the OpenCV DNN module for human detection[29]–[31]. Upon receiving a goal or a waypoint, each TurtleBot utilizes `move_base` to initiate a series of tasks that culminate in the robot navigating to the specified location while avoiding obstacles in real-time. The layered costmap within `move_base` is continually updated with LIDAR data and used by the A* planner to compute the safest and most efficient path.

The controller interfaces with the robots' actuators to execute these planned movements, while also employing a local planner to avoid any unforeseen obstacles. After reaching a waypoint, the TurtleBot performs human detection; if a human is located, `move_base` recalculates the path to guide the individual to the nearest exit point efficiently**??**.

3.3.3 AMCL Localisation

The TurtleBot employs Adaptive Monte Carlo Localization (AMCL) for real-time pose estimation within a known environment. The AMCL node subscribes to the robot's laser scan data and odometry information to continuously update its pose estimate[30]. This is achieved using a particle filter, where each particle represents a probable robot pose. As the robot navigates, the particles are resampled based on how well they align with the incoming sensor data, thereby improving the pose estimate over time.

This real-time localization is critical for high-level navigation tasks such as path planning and obstacle avoidance. It allows the TurtleBot to know its position in both local and global frames of reference, facilitating accurate movement and interaction with its environment.

3.4 Map Generation

In the map generation phase, the Husky robot employs a mapless navigation approach, guided by the `move_base` node in ROS. The robot navigates through the simulated office environment without the use of a pre-existing map. This is achieved by passing a specific argument, `no_static_map`, set to `true` during the launch of the `move_base` node. This instructs the system to rely solely on real-time sensor data for navigation, thereby constructing a dynamic map as the robot traverses the area3.5. Upon successful navigation and mapping, the dynamically generated map is saved using the command:

```
rosrun map_server map_saver -f <filename>
```

facilitating easy retrieval and sharing for subsequent tasks involving the TurtleBot swarm.

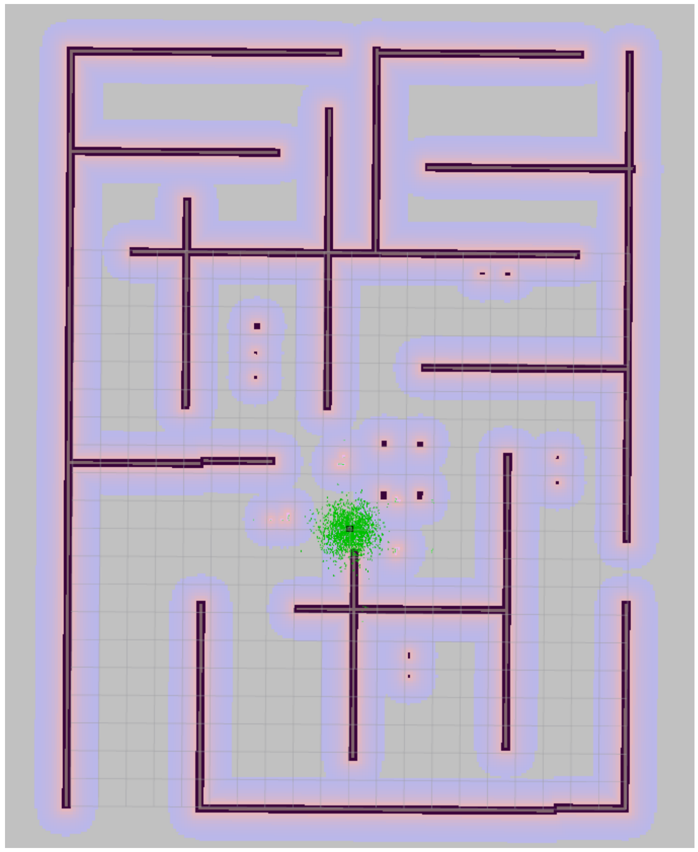


Figure 3.5: Husky generated map to facilitate TurtleBot's navigation

3.5 Human Detection

OpenCV Deep Neural Network (DNN) module is utilized for human detection, leveraging the power of the pre-trained YOLOv4-tiny model running on the Darknet framework[31]. The DNN module offers a seamless way to import the YOLO model into the ROS-based simulation environment. Each TurtleBot and the Husky robot are equipped with RGB cameras, serving as their primary sensors for visual perception. These cameras feed real-time image data to the OpenCV DNN module, which then processes the images to detect the presence of humans[34]. Upon detecting a human, the robot recalculates its path, guiding the individual towards the nearest exit. The use of the OpenCV DNN module in conjunction with Darknet and the robots' onboard RGB cameras creates an effective and efficient system for human detection, integral for the success of search and rescue missions[32].

4 Results

The simulation setting was designed to replicate a search-and-rescue scenario in an office environment, and the following results section explains the key findings obtained through this simulation. The dual-robotic framework—consisting of a Husky robot for preliminary mapping and a swarm of 20 TurtleBots for human identification and rescue—achieved its set objectives with notable efficacy.

The Husky robot, leveraging the `move_base` controller and LIDAR sensor, successfully navigated and mapped the office environment. This generated map provided a base of operation for the subsequent deployment of the TurtleBots. utilising Adaptive Monte Carlo Localization (AMCL) and A* path planning algorithms, the TurtleBots searched the area for human entities. Human detection was executed via an onboard RGB camera equipped with OpenCV's DNN module. Upon successful identification, a terminal prompt denoting "robot 'n' rescued human" was generated, where 'n' represents each robot's unique identification number ranging from 1 to 20.

Each TurtleBot was tasked with guiding the identified humans to the nearest of four designated safe exits. Upon successful completion of this rescue mission, a terminal prompt confirmed that the human had been guided to a safe exit. Finally, each TurtleBot returned to its initial home location, thereby completing its rescue assignment.

These outcomes, as elucidated through terminal prompts and operational performance, affirm the capability of the deployed robotic system for effective autonomous mapping and human rescue. The forthcoming discussion elaborates on the quantitative performance metrics, thereby offering an academically rigorous validation of the system's functionality and its potential applicability in real-world scenarios.

4.1 Performance Metrics for Husky Robot

4.1.1 Localization Accuracy

To evaluate the localization accuracy of the Husky robot, we performed three separate runs where the robot was commanded to navigate to a specific known location $(x_1 = 14, y_1 = -8)$ within the simulated office environment. The actual estimated positions (x_2, y_2) obtained by the robot were then recorded and compared to the known coordinates. The localization error for each run was calculated using the Euclidean distance formula as follows:

$$\text{Error} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

First Run: The recorded estimated location for the first run was $(x_2 = 12.943, y_2 = -9.674)$. The localization error was calculated as:

$$\text{Error}_1 = \sqrt{(12.943 - 14)^2 + (-9.674 - (-8))^2} \approx \sqrt{1.116249 + 2.801476} \approx \sqrt{3.917725} \approx 1.9804 \text{ m}$$

Second Run: For the second run, the estimated location was $(x_2 = 13.234, y_2 = -8.783)$, yielding an error of:

$$\text{Error}_2 = \sqrt{(13.234 - 14)^2 + (-8.783 - (-8))^2} \approx \sqrt{0.587236 + 0.612289} \approx \sqrt{1.199525} \approx 1.0952 \text{ m}$$

Third Run: The estimated location for the third run was $(x_2 = 14.896, y_2 = -7.988)$. The resulting error was:

$$\text{Error}_3 = \sqrt{(14.896 - 14)^2 + (-7.988 - (-8))^2} \approx \sqrt{0.802816 + 0.000144} \approx \sqrt{0.802960} \approx 0.8960 \text{ m}$$

Finally, the average localization error across the three runs was calculated as:

$$\text{Average Error} = \frac{\text{Error}_1 + \text{Error}_2 + \text{Error}_3}{3} = \frac{1.9804 + 1.0952 + 0.8960}{3} \approx \frac{3.9716}{3} \approx 1.3239 \text{ m}$$

The average localization error for the Husky robot was approximately 1.324 m, providing a quantifiable measure of its localization capabilities in the given simulated environment.

Figure 4.1: Bar chart illustrating the localization errors for three separate runs and the calculated average error. The y-axis represents the localization error in meters, providing a quantifiable measure of the Husky robot's localization capabilities in the simulated environment. Run 1 yielded an error of approximately 1.9804 meters, Run 2 an error of approximately 1.0952 meters, and Run 3 an error of approximately 0.8960 meters. The average localization error across the three runs was approximately 1.324 meters.

4.1.2 Collision Avoidance

In the simulated office environment, a total of 22 obstacles were manually identified, primarily consisting of table legs. During the autonomous navigation of the Husky robot, it was observed that the robot got stuck under two tables, accounting for a collision with 6 of the obstacles. To evaluate the performance of the Husky's collision avoidance capability, the Collision Avoidance Ratio was calculated as follows:

$$\text{Collision Avoidance Ratio} = \frac{\text{Number of Successfully Avoided Collisions}}{\text{Total Number of Obstacles}} = \frac{22-6}{22} = \frac{16}{22} \approx 0.727 \tag{4.1}$$

The Collision Avoidance Ratio of approximately 0.727 suggests that the Husky was moderately successful in avoiding obstacles, but there is room for improvement in its collision avoidance algorithms.

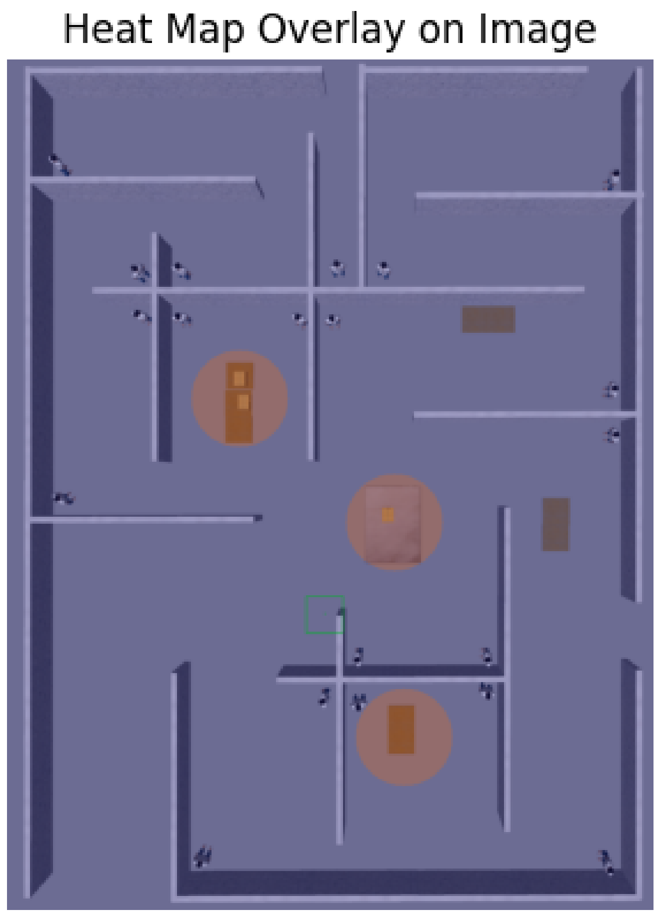


Figure 4.2: Heat map showing locations in the simulated office environment where the Husky robot had collisions.

4.1.3 Path Length

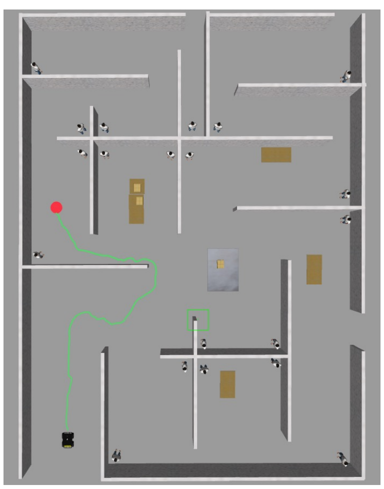
The path length is an essential metric to gauge the efficiency of the robot's navigation strategy. It signifies the total distance traversed by the Husky robot during its mission. The measurement was obtained using odometry data, which was extracted and analyzed to calculate the path length.

The x and y coordinates of the robot's position were captured at various timestamps. Subsequently, the Euclidean distance between each consecutive pair of points was calculated using the formula:

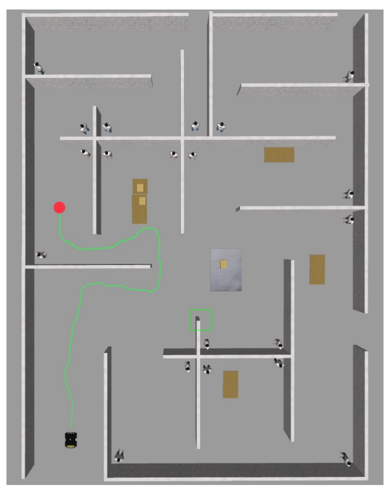
$$\text{Distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

The distances between all adjacent points were summed to compute the total path length for each run. The results for the three runs are as follows:

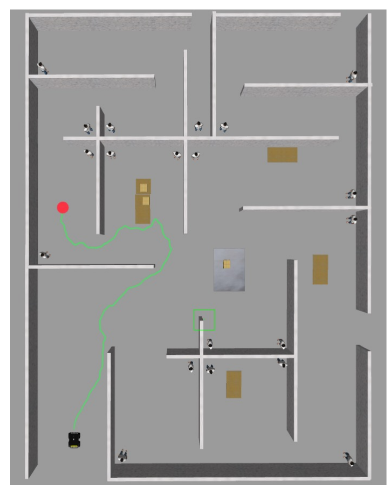
- First run: 10.049 m
- Second run: 11.689 m
- Third run: 14.547 m



(a) Run 1: Path Length = 10.049 m



(b) Run 2: Path Length = 11.689 m



(c) Run 3: Path Length = 14.547 m

Figure 4.3: Comparison of Husky Robot's Path Length for Three Different Runs. The green line represents the path taken, and the red circle indicates the destination.

To acquire a more robust measure of the robot's performance, the average path length across the three runs was calculated:

$$\text{Average Path Length} = \frac{10.049\,\text{m} + 11.689\,\text{m} + 14.547\,\text{m}}{3} \approx 12.095\,\text{m}$$

This average value provides a reliable metric for the path length typically covered by the Husky robot in the given environment.

4.1.4 Navigation Efficiency

In evaluating the efficiency of the Husky robot's navigation capabilities, two key parameters were considered: the time taken to complete the mapping and the total area covered during this mapping process.

Area Covered

To estimate the area covered by the Husky robot, a grid overlay was applied to the mapped environment. Each grid cell was assumed to have an area of $1m^2$. The number of occupied grids was counted to be 243, implying that the Husky covered an area of $243m^2$.

Time Taken

The time taken for the Husky to complete the mapping was recorded to be 38 minutes.

Efficiency Calculation

The navigation efficiency can be calculated by normalizing the total time taken by the total area covered. Mathematically, this can be represented as:

$$\text{Navigation Efficiency} = \frac{\text{Total Time Taken}}{\text{Total Area Covered}} = \frac{38\text{ minutes}}{243m^2} \approx 0.156\text{ minutes/m}^2$$

Analysis

The efficiency metric offers a quantitative measure of the Husky robot's capability to navigate and map an area within a given time-frame. A lower value for the navigation efficiency indicates a more efficient mapping process, and vice versa. In our case, the Husky robot exhibited an efficiency of approximately 0.156 minutes/m^2, which can serve as a benchmark for future optimizations or comparisons with other robotic platforms.

4.2 Performance Metrics for TurtleBots

In order to evaluate the efficiency and effectiveness of the TurtleBots in the simulated environment, various performance metrics were devised and calculated. These metrics capture different aspects of the TurtleBots' abilities to perform their designated tasks. During the experiment, the performance of each TurtleBot was carefully monitored and recorded. The metrics include whether each robot was able to find a human, guide them to a safe exit, return to its home position, or if it got stuck during any of these tasks. The data was manually extracted from the terminal window for each robot 4.4, providing a real-time assessment of their operational efficiency and capabilities. Table 4.1 summarizes these observations.

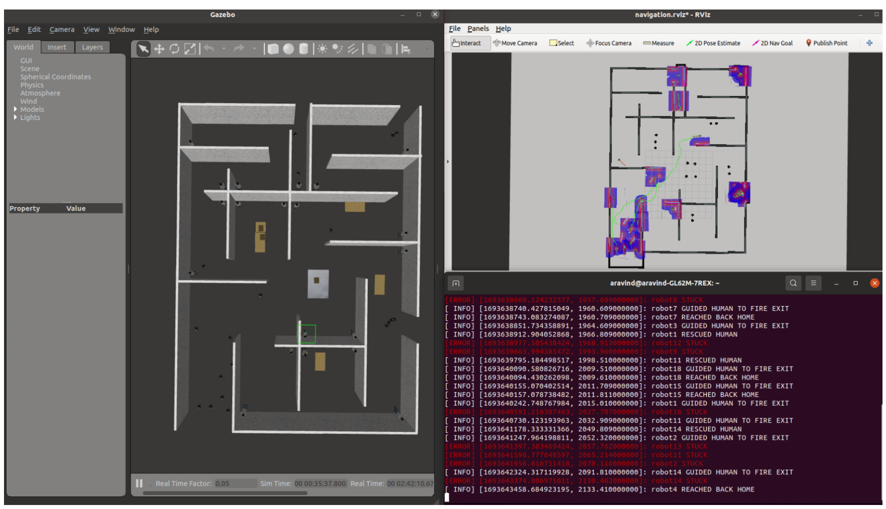


Figure 4.4: An integrated view of the rescue operations simulation featuring the Gazebo environment, Rviz visualization, and terminal window. This captures the real-time decision-making and movements of TurtleBots during their tasks of human detection, guidance to safe exits, and navigation back to home positions.

4.2.1 Detection Accuracy

The first metric examined is Detection Accuracy, which measures how accurately the robots are able to detect human models in the environment.

$$\text{Detection Accuracy} = \left(\frac{\text{Number of Robots that Detected Human}}{\text{Total Number of Robots}}\right) \times 100$$
$$= \left(\frac{19}{20}\right) \times 100 = 95\%$$

4.2.2 Guidance Accuracy

Next, we consider the Guidance Accuracy, which gauges the ability of the robots to guide the human models to safety.

$$\text{Guidance Accuracy} = \left(\frac{\text{Number of Robots that Guided Human}}{\text{Number of Robots that Detected Human}}\right) \times 100$$
$$= \left(\frac{13}{19}\right) \times 100 \approx 68.42\%$$

4.2.3 Return-to-Home Accuracy

This metric looks at how many robots were able to successfully return to their home position after completing the guidance task.

$$\text{Return-to-Home Accuracy} = \left(\frac{\text{Number of Robots that Returned Home}}{\text{Number of Robots that Guided Human}}\right) \times 100$$
$$= \left(\frac{8}{13}\right) \times 100 \approx 61.54\%$$

4.2.4 Stuck and Recovery Rate

Finally, the Stuck and Recovery Rate evaluates how well the robots can recover from operational challenges.

$$\text{Stuck Rate} = \left(\frac{\text{Number of Robots that Got Stuck}}{\text{Total Number of Robots}}\right) \times 100$$
$$= \left(\frac{10}{20}\right) \times 100 = 50\%$$

$$\text{Recovery Rate} = \left(\frac{\text{Number of Robots that Recovered}}{\text{Number of Robots that Got Stuck}}\right) \times 100$$
$$= \left(\frac{9}{10}\right) \times 100 = 90\%$$

Table 4.1: Data of TurtleBots in Human Detection and Guiding Tasks.

Robot Number	**Found Human**	**Guided to Exit**	**Returned Home**	**Stuck**
1	Yes	Yes	No	No
2	Yes	Yes	No	Yes
3	Yes	Yes	Yes	No
4	Yes	Yes	Yes	No
5	Yes	Yes	Yes	No
6	Yes	No	No	Yes
7	Yes	Yes	Yes	No
8	Yes	No	No	Yes
9	Yes	No	No	Yes
10	Yes	No	No	Yes
11	Yes	Yes	No	Yes
12	Yes	No	No	Yes
13	No	No	No	Yes
14	Yes	Yes	No	Yes
15	Yes	Yes	Yes	No
16	Yes	No	No	Yes
17	Yes	Yes	Yes	No
18	Yes	Yes	Yes	No
19	Yes	Yes	Yes	No
20	Yes	Yes	No	No

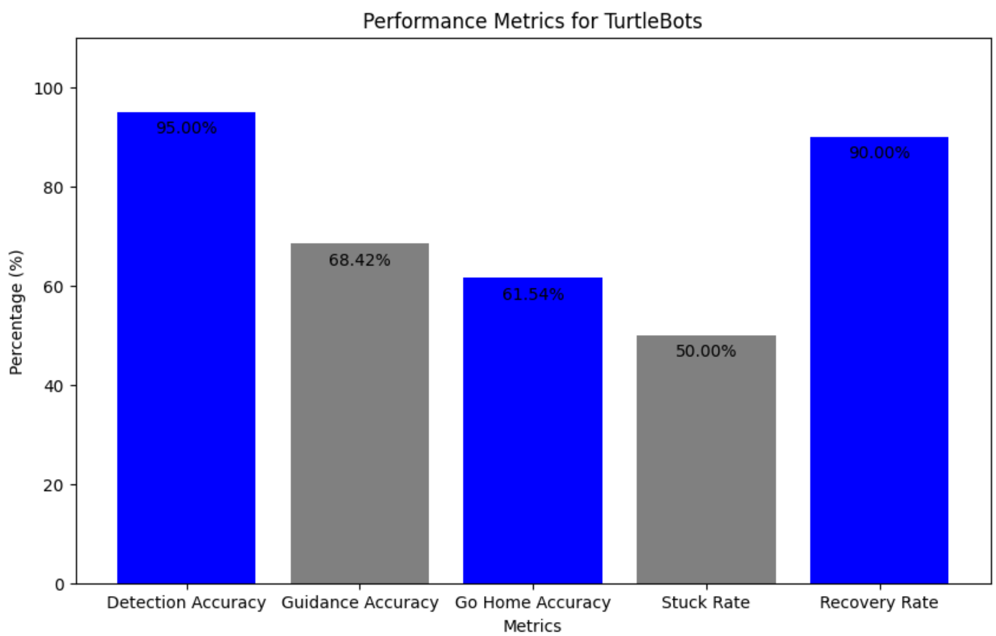


Figure 4.5: Performance Metrics for TurtleBots in the simulated environment. The graph showcases five key metrics. 'Detection Accuracy' represents the percentage of robots that successfully detected human beings. 'Guidance Accuracy' illustrates the percentage of robots that, after detecting humans, successfully guided them to the exit. 'Return-to-Home Accuracy' quantifies the percentage of robots that returned to their home positions after guiding humans. 'Stuck Rate' shows the percentage of robots that got stuck at some point during the simulation. Lastly, 'Recovery Rate' indicates the percentage of stuck robots that managed to recover and continue their tasks. Each metric is represented as a percentage of the total number of robots involved in the simulation.

5 Discussion and Conclusion

The dissertation focuses on the design, development, and performance evaluation of a robotic search and rescue system in a simulated office environment. Utilising ROS Noetic on Gazebo and RViz platforms, the study employs a dual-robotic framework featuring a Husky robot for initial mapping and a swarm of 20 TurtleBots for rescue operations. The Husky uses the `move_base` controller for navigation and mapping, and its performance is assessed using metrics such as localization accuracy and collision avoidance. Once the map is generated, it is uploaded to the TurtleBots, which use Adaptive Monte Carlo Localization and A* path planning for navigation. These bots are equipped with LIDAR and RGB cameras and uses OpenCV's DNN modules for human detection. Upon locating humans, the bots guide them to the nearest safe exit. Performance metrics for the TurtleBots include detection accuracy, guidance accuracy, return-to-home accuracy, and stuck & recovery rate. The simulations are conducted thrice under identical conditions to obtain average performance metrics. This comprehensive approach allows the study to address a critical real-world issue by combining mapping, localisation, and human detection in a unified framework.

In the virtual office environment used for testing the Husky and TurtleBot swarm, it's crucial to acknowledge several limitations that may not entirely replicate real-world conditions. Firstly, the simulation lacks real-time dynamic elements such as moving obstacles, fire, or smoke, which are common in actual disaster scenarios. The absence of these elements means that sensor data won't have to adapt to rapidly changing conditions, potentially affecting the robots' performance in an actual emergency. Moreover, the human models are statically placed in corners, which might not reflect the random and sometimes inaccessible locations where individuals could be trapped in a real disaster.

Another constraint is the simplified office environment itself, which doesn't account for the varied and often cluttered spaces in real offices, such as intricate cubicle arrangements, elevators, or stairwells. These complex elements in an actual office could pose navigation challenges for the robots, potentially hindering their efficiency in search and rescue operations. Lastly, simulations can't fully replicate the unpredictable nature of human behavior during emergencies, such as panic or uncooperative actions, which could significantly influence the robots' performance in real-world applications. Therefore, while the simulated environment offers a controlled setting for initial testing, it may not fully prepare the robots for the complexities and unpredictability of real-world disaster situations.

5.1 Evaluation of Hypotheses and Performance Metrics

The primary objective of this research was to scrutinize the capabilities and shortcomings of a Husky robot and TurtleBots operating in a simulated office environment. Specific hypotheses guided the evaluation: for the Husky robot, the expected average localization error was below 1 meter, and better performance was hypothesized in environments with fewer obstacles. For the TurtleBots, successful human detection was expected at least 75% of the time, and a return-to-home success rate above 80% was hypothesized in controlled settings.

5.1.1 Husky Robot

The Husky robot displayed an average localization error of approximately 1.324 meters, marginally exceeding the hypothesized 1-meter threshold. Despite its generally effective navigation capabilities in the simulated environment, these results reveal areas where localization accuracy can be enhanced. Moreover, the Collision Avoidance Ratio of 0.727 indicates a degree of success in evading obstacles but shows room for further improvement, partially confirming the hypothesis concerning performance in less cluttered environments.

5.1.2 TurtleBots

In contrast to the hypothesis, the TurtleBots excelled in human detection, achieving a 95% success rate, well beyond the hypothesized 75%. However, they were less successful in return-to-home tasks, with a 61.54% success rate, falling short of the expected 80%. These results indicate that although the robots excel in human detection, their navigation and home-return mechanisms need refinement.

5.2 Reasons for Unmet Hypotheses

5.2.1 Husky Robot

1. **Localization Error**: The Robot Operating System (ROS) framework used employed probabilistic methods for sensor data integration, which, while robust, are still susceptible to cumulative error in a complex environment. Moreover, limitations in the SLAM (Simultaneous Localization and Mapping) algorithm might have contributed to the increased error.

2. **Collision Avoidance**: The moderate Collision Avoidance Ratio suggests that while the robot is designed to navigate around larger obstacles effectively, finer structures like table legs pose a challenge. This indicates the need for algorithms with better sensor fusion or machine learning capabilities for navigating intricate spaces.

5.2.2 TurtleBots

1. **Guidance Accuracy and Return-to-Home Rate**: The discrepancy between the hypothesized and actual return-to-home rate suggests that the path-planning algorithms employed are less reliable. Limitations could stem from inaccuracies in real-time obstacle detection and avoidance.

2. **Methodological Constraints**: The study relied solely on a simulated environment, which may not fully capture the complexities of real-world scenarios, affecting the validity of the results.

5.3 Recommendations for Future Research

5.3.1 Husky Robot

1. **Sensor Fusion**: Implement advanced sensor fusion methods to filter out noise and enhance localization accuracy. The implementation of sensor fusion methods can substantially improve a Husky robot's situational awareness. By combining data from multiple sources—such

as GPS, IMU, and LIDAR—we can filter out noise and anomalies that are often present when using a single sensor. Advanced Kalman filtering or particle filtering techniques can be incorporated to improve localization accuracy, particularly in complex or unpredictable terrains.

2. **Machine Learning Algorithms**: Introduce machine learning techniques like Random Forest or Neural Networks to reduce cumulative error. Currently, traditional control algorithms are used for decision-making in Husky robots. Implementing machine learning techniques, such as Random Forest or Neural Networks, can minimize cumulative error and increase the robot's adaptability to new situations. For instance, a Neural Network could be trained to recognize patterns related to specific types of obstacles or terrain and adjust the robot's motion planning accordingly. These algorithms could also be used for predictive maintenance, enabling the robot to foresee mechanical issues before they become critical, thus enhancing the reliability of the system.

5.3.2 TurtleBots

1. **Improved Path-Planning Algorithms**: Utilise D* Lite or other path planning algorithms for more reliable and dynamic navigation. While existing path-planning algorithms offer a reasonable degree of effectiveness, incorporating more advanced algorithms like D* Lite could substantially improve the TurtleBots' navigation. These algorithms take into account both static and dynamic obstacles, providing a more realistic and reactive approach to navigation, particularly in constantly changing environments such as disaster zones.

2. **Multi-Sensor Fusion**: Combine multiple sensor types like LIDAR, cameras, and ultrasonic sensors for improved tracking and guidance. Given that TurtleBots are often deployed in swarms for broader coverage, it becomes crucial for individual units to possess enhanced sensory capabilities. By fusing data from multiple types of sensors—like LIDAR for distance measurement, cameras for visual context, and ultrasonic sensors for close-range object detection—we can significantly improve the TurtleBots' tracking and guidance systems. This multi-sensory approach will enable better object recognition and localization, crucial for tasks such as victim identification and obstacle avoidance.

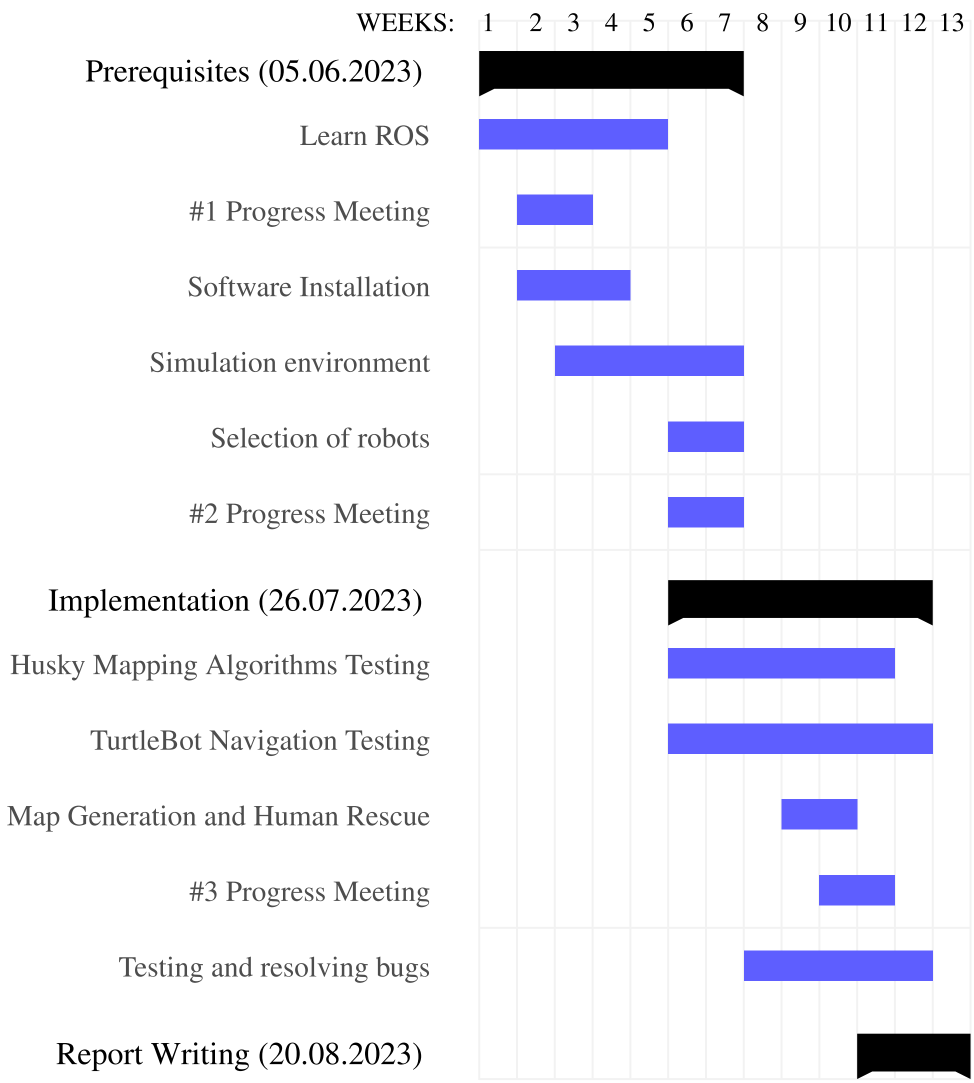
5.4 Methodological Enhancements

1. **Real-World Testing**: Include real-world tests to augment the simulation-based evaluation.

2. **Dynamic Testing Environments**: Develop testing scenarios that closely mimic real-world conditions to ascertain the robots' performance in a broader range of situations.

While the Husky robot and TurtleBots demonstrated laudable, though imperfect, competencies in specific domains, they fell short in others, thereby partially meeting the initial hypotheses. These gaps between the hypothesized outcomes and the empirical findings provide a valuable opportunity for technical advancements. The recommendations and insights drawn from this study lay a solid groundwork for further research and technological improvements, which can enhance the reliability and efficiency of these robotic systems in real-world applications, particularly in search and rescue or emergency response.

A Appendix



WEEKS: 1 2 3 4 5 6 7 8 9 10 11 12 13
Prerequisites (05.06.2023)
Learn ROS
#1 Progress Meeting
Software Installation
Simulation environment
Selection of robots
#2 Progress Meeting
Implementation (26.07.2023)
Husky Mapping Algorithms Testing
TurtleBot Navigation Testing
Map Generation and Human Rescue
#3 Progress Meeting
Testing and resolving bugs
Report Writing (20.08.2023)

MSc project experiment permission form for EMATM0055

Lead supervisor name: Dr. Hamidreza Nemati

Student name(s): Aravind Sairam Saravanan

Student ID(s): 2264341

Project title: Dual-Robotic System for Autonomous Search and Rescue in Simulated Office Environment

Summary of proposed experiment(s):

The research consists of simulation of two robots in an office environment for search and rescue, namely Husky and TurtleBot. The husky is used to map the unknown environment and produce a map whereas a swarm of 20 turtlebots use that given map to search and rescue humans.

For projects to fall within the scope of this ethics process, all answers to the following questions must be "No". If ANY questions are answered with a "Yes", a full ethics application must be submitted and approved before undertaking experimental work. For a fuller description of what is / isn't covered by this process, please check the "Blanket ethics approval process document" (available on the Unit Blackboard page).

Does the proposed experiment gather data from animals? ~~Yes~~/No

Does the proposed experiment gather data from a vulnerable population? ~~Yes~~/No

Does the proposed experiment gather sensitive information? ~~Yes~~/No

Does the proposed experiment take photos or videos of people? ~~Yes~~/No

Does the proposed experiment gather other data which, if lost, would allow participants to be identified? ~~Yes~~/No

Does the proposed experiment gather data from people who have not given full informed consent? ~~Yes~~/No

Does the proposed experiment trick or deceive participants in any way? ~~Yes~~/No

Does the risk assessment for the proposed experiment indicate a greater than low risk of physical or mental harm? ~~Yes~~/No

Does the proposed experiment gather data from more than 100 participants? ~~Yes~~/No

Will participants be recruited via methods other than word of mouth, posters, email lists (if permitted) and online forums (where explicitly allowed by the forum administrators)? ~~Yes~~/No

I certify that the above information is accurate and that I have read the relevant section of the project handbook.

Date: 05/09/2023 **Student signature(s):**

I certify that I have discussed the methodology of the proposed experiment with the student(s) in detail and that I am satisfied the above information is accurate.

Date: **Lead supervisor signature(s):**

Guidance for students and supervisors

Students should include this form in their final submission and email it to grp-dissertation_unit_2022@groups.bristol.ac.uk by 12/09/2023.

Note that this form does not need to cover every experiment in the project – if you later decide to run more experiments, you can submit multiple copies of this form.

Summary of the proposed experiment(s): It is very important that you discuss your plans in detail with your supervisor. However, you do not need to write them in detail on this form. A single sentence should be enough.

If you have any other questions, please contact grp-dissertation_unit_2022@groups.bristol.ac.uk

Ethics reporting form for EMATM0055

Lead supervisor name: Dr. Hamidreza Nemati

Student name(s): Aravind Sairam Saravanan

Student ID(s): 2264341

Project title: Dual-Robotic System for Autonomous Search and Rescue in Simulated Office Environment

Brief summary of project:

The research consists of simulation of two robots in an office environment for search and rescue, namely Husky and TurtleBot. The husky is used to map the unknown environment and produce a map whereas a swarm of 20 turtlebots use that given map to search and rescue humans.

Brief summary of experiments run and data gathered by the student(s):

Has the project been completed and handed in as of today? ~~Yes~~/No

If the project has been completed, please indicate one option with an X.

__X__ This project did not require ethical approval

_____ This project fell within the scope of the ethics pre-approval process and I have attached the student(s)' permission form(s)

_____ This project has been approved by the faculty ethics committee in a separate application.

_____ Other (please give details below)

Guidance for supervisors

Please email this form to grp-dissertation_unit_2022@groups.bristol.ac.uk by the 12/09/2023 (i.e one week after the dissertation deadline) and your student(s) permission form(s) if they have not already been uploaded. If they are not available, please choose the "Other" option and give a brief explanation of why.

Brief summary of the project: This field is intended to contain any background necessary in order for the rest of the form's content to make sense. There is no need to write more than a sentence or two.

Brief summary of experiments run, and data gathered by student(s): This field is intended to give us a quick idea of whether or not the project is likely to fall under ethics pre-approval process. There is no need to go into detail on methodology, as we should be able to get that detail from the project itself. For example, a sentence reading "Showed project to prospective users and asked for feedback" or "Filmed volunteers doing squats to obtain training data" would be fine. For more detail on what projects are covered by ethics pre-approval process please see "Blanket ethics approval process document" (available on the Unit Blackboard page). If you have any other questions, please contact grp-dissertation_unit_2022@groups.bristol.ac.uk.

NEW ETHICAL REVIEW PROCESS FOR Robotics MSc Dissertation Projects

This year, we are introducing a streamlined ethical review process for MSc projects. To summarise, the unit has now obtained blanket ethical approval for "obviously harmless" projects fitting a specific description which we think should cover most use cases. Supervisors will be able to self-certify that their students' projects fit this description by uploading a form. Students will include in their frontmatter a declaration that either their project did not require ethical review at all, or that it fits within the blanket approval, or that they have undergone independent review from the committee.

At the end of the year, a random subset of projects will be checked to make sure they are correctly categorised and review the system for further improvement next year.

WHAT YOU NEED TO DO

All students must include one of the following statements in their compulsory preliminaries:

- "This project did not require ethical review, as determined by my supervisor [fill in name]"
- "This project fits within the scope of ethics pre-approval process, as reviewed by my supervisor [fill in name] and approved by the faculty ethics committee as application 12723"
- "An ethics application for this project was reviewed and approved by the faculty ethics committee as application [fill in number]".

Supervisors using the pre-approval process must fill out the reporting form and email it to grp-dissertation_unit_2022@groups.bristol.ac.uk by 12/09/2023, i.e. one weeks after the project submission deadline. Please also include any participant facing information (i.e blank consent form and PIS) as an appendix to the dissertation

All supervisors who wish to make use of the blanket approval must tell their students to get their permission before running tests or gathering data. Your students will need to fill out a short permission form, included in this document, for you to sign off on. While we have designed the form itself to be short and easy, we do expect that you discuss your students' proposed experiments with them in detail before filling it out. You are just as responsible as them for ensuring that their projects fall within the scope of the application.

Note that asking the student to submit a full ethics application is very far from a "no", as the blanket approval only covers a subset of easy cases. You should not feel under any obligation to sign off on a project if you are not fully confident that it fits within the blanket approval.

DETAILED GUIDANCE FOR SUPERVISORS

Projects not requiring ethical review

A project does not require ethical review only if it does not involve gathering data from human or animal participants, either directly or indirectly. This means, for example, that the author cannot conduct surveys; they cannot ask other people to test software or hardware; and they cannot take pictures or video that contain people.

Two examples of projects that would not require ethical review are as follows:

- Jean is doing a project in swarm robotics. This will involve writing code and running experiments to observe the behaviour of a small robot swarm. The only experiments Jean will run involve testing her robots.

- Adam is doing a project in machine learning, and he is attempting to train a neural network to recognise different art styles and periods. In doing so, he will use a large data set of public domain images downloaded from the Internet, and perhaps one or two photos of his own pieces.

Projects within the scope of ethics pre-approval process

The blanket application covers projects which involve gathering data only from humans (not from animals) and only under the following circumstances:

- The project does not specifically try to gather data from a vulnerable population, such as people affected by illness or economic disadvantage, primary or secondary school students, or people recruited from self-help groups.

- The project does not ask gather data about: racial or ethnic origin; religious or similar beliefs; membership of a trade union; physical or mental health; sexual activities; criminal history; drug use; or other obviously sensitive information.

- All data is anonymous at collection, so that if the data were to be lost there would be no realistic prospect of the participants being identified. In particular, the project does not take photos or videos of people, and it does not ask about: names; addresses; postcodes; phone numbers; email addresses; physical features; or social media handles.

- Before any data is gathered from a participant, they give full informed consent. That is, they understand what data will be collected and what the data will be used for, and they give an explicit verbal or written statement to this effect. If part of the data comes from observation, then participants are told what will be recorded before the start of the test.

- No participants are tricked or deceived in any way, for example by being given false feedback about their performance at a task or being misled about the focus of the study.

- If the participants are asked to perform a task, then this task involves only low risks of physical or mental harm and participants are made aware of these risks. Here 'low risk' is in reference to an approved risk assessment, which must be shared with participants prior to the experiment.

Three examples of projects that would fall into this category are:

• Ayodya is working on machine vision algorithm for recognising emotions. As part of refining the core algorithm, she wants to get feedback from potential users. She asks a few of her friends to use the system as she watches and take notes. Afterwards, she asks them a few questions. She is careful to tell them in advance that she'll be observing them.

• Tan is writing a piece of software for research into grumkins for his project. Since Bristol researchers specialise mostly in snarks, he posts to a forum for grumkin researchers explaining his situation, asking what they would like out of a grumkin robot and noting that any replies may be incorporated into his project. He then uses the requirements to decide what features to develop. When writing up his project, he removes all forum names from the feedback before including it.

• Mohammed is testing the effectiveness of a new haptic feedback device as part of an HRI project. To do so, he creates two strategies for controlling the device. He divides participants into two groups, one for each version. He tells each group he is interested in testing the effectiveness of the system and asks them to use it to carry out a task while he watches and takes notes.

Projects requiring ethical review

All other projects must make a formal Stage 1 ethics application to the Faculty Research Ethics Committee here. This is itself a streamlined process – students apply online via a short form, and most applications will be approved within three weeks. If the committee identifies a cause for concern, then students may be asked to fill out a full Stage 2 application which addresses these concerns in more detail.

Four examples of projects that would fall into this category are:

• Chris is doing an HRI project to study factors affecting trust in robot companions. They ask some people to interact with a robot during a task and participants are told the purpose of the study is to evaluate their performance on the task. The tasks are identical, but Chris varies whether the robot users a male, female or androgenous voice during the task to see how this alters perceptions of performance. This would still require a full ethics application, since they are deceiving participants about the purpose of the study.

• Alan is doing a HRI project on people's perception of robot avatars. He puts up a sign in the MVB atrium explaining that an experiment is happening, sets up a screen and speaker for his avatar and then sits nearby observing how participants interact with. This would require a full ethics application, since it involves gathering data from participants who have not given full informed consent.

• Reut suffers from clinical depression, and for her project she is writing a mobile app to help fellow sufferers track their moods. When she has finished coding the app, she posts a link to Big White Wall (an anonymous support forum) asking for feedback. This would require a full ethics application since it involves asking about mental health and gathering data from a vulnerable population.

• Johannes is doing a computer vision project in which he attempts to train a computer to recognise numbers on signs. As part of his test data, he walks through the department and takes photos of all the office doors. This would require a full ethics application, since it involves gathering information (namely office numbers) from participants without their consent.

References

[1] N. F. P. Association, Office property fires 2012, 2012. available from: https://www.nfpa.org/News-and-Research/Data-research-and-tools/Building-and-Life-Safety/Office-Property-Fires.

[2] U. F. Administration, Non-residential building fires 2019, 2019. available from: https://www.usfa.fema.gov/data/statistics/non_residential.html.

[3] J. Jennings, G. Whelan, and W. Evans, "Cooperative search and rescue with a team of mobile robots," in *1997 8th International Conference on Advanced Robotics. Proceedings. ICAR'97*, 1997, pp. 193–200. DOI: 10.1109/ICAR.1997.620182.

[4] S. Burion, "Human detection for robotic urban search and rescue," Tech. Rep., 2004.

[5] J. Berger and N. Lo, An innovative multi-agent search-and-rescue path planning approach, *Computers & Operations Research*, vol. 53 2015, pp. 24–31, 2015.

[6] AzoRobotics. "The role of robotics in natural disasters." (2021), available from: https://www.azorobotics.com/Article.aspx?ArticleID=401.

[7] Z. Uddin and M. Islam, "Search and rescue system for alive human detection by semi-autonomous mobile rescue robot," in *2016 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, 2016, pp. 1–5. DOI: 10.1109/ICISET.2016.7856489.

[8] A. Macwan, J. Vilela, G. Nejat, and B. Benhabib, A multirobot path-planning strategy for autonomous wilderness search and rescue, *IEEE Transactions on Cybernetics* [online], vol. 45, no. 9 2015, pp. 1784–1797, 2015. DOI: 10.1109/TCYB.2014.2360368.

[9] N. Ruangpayoongsak, H. Roth, and J. Chudoba, "Mobile robots for search and rescue," in *IEEE International Safety, Security and Rescue Rototics, Workshop, 2005.*, 2005, pp. 212–217. DOI: 10.1109/SSRR.2005.1501265.

[10] D. Hutabarat, M. Rivai, D. Purwanto, and H. Hutomo, "Lidar-based obstacle avoidance for the autonomous mobile robot," in *2019 12th International Conference on Information Communication Technology and System (ICTS)*, 2019, pp. 197–202. DOI: 10.1109/ICTS.2019.8850952.

[11] O. Hamzeh and A. Elnagar, Localization and navigation of autonomous indoor mobile robots, *International Journal of Computing, Communications & Instrumentation Engineering (IJCCIE)*, vol. 2, no. 2 2015, pp. 228–233, 2015.

[12] J. Borenstein and Y. Koren, Real-time obstacle avoidance for fast mobile robots, *IEEE Transactions on Systems, Man, and Cybernetics* [online], vol. 19, no. 5 1989, pp. 1179–1187, 1989. DOI: 10.1109/21.44033.

[13] N. L. Ferreira, M. S. Couceiro, A. Araújo, and R. P. Rocha, "Multi-sensor fusion and classification with mobile robots for situation awareness in urban search and rescue using ros," in *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2013, pp. 1–6. DOI: 10.1109/SSRR.2013.6719361.

[14] C. Duan and S. Luo, "Design of pedestrian detection system based on opencv," in *2022 4th International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*, 2022, pp. 256–259. DOI: 10.1109/AIAM57466.2022.00055.

[15] F. H. Shubho, F. Iftekhar, E. Hossain, and S. Siddique, "Real-time traffic monitoring and traffic offense detection using yolov4 and opencv dnn," in *TENCON 2021 - 2021 IEEE Region 10 Conference (TENCON)*, 2021, pp. 46–51. DOI: 10.1109/TENCON54134.2021.9707406.

[16] Y. Chang, K. Ebadi, C. E. Denniston, *et al.*, Lamp 2.0: A robust multi-robot slam system for operation in challenging large-scale underground environments, *IEEE Robotics and Automation Letters* [online], vol. 7, no. 4 2022, pp. 9175–9182, 2022. DOI: 10.1109/LRA.2022.3191204.

[17] A. Denker and M. C. İşeri, "Design and implementation of a semi-autonomous mobile search and rescue robot: Salvor," in *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, 2017, pp. 1–6. DOI: 10.1109/IDAP.2017.8090184.

[18] T. S. Abhishek, D. Schilberg, and A. S. A. Doss, “Retracted: Obstacle avoidance algorithms: A review,” in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, vol. 1012, 2021, p. 012 052.

[19] E. Messina and A. Jacoff, “Performance standards for urban search and rescue robots,” in *Unmanned Systems Technology VIII*, SPIE, vol. 6230, 2006, pp. 639–650.

[20] T. Kamegawa, T. Akiyama, S. Sakai, *et al.*, Development of a separable search-and-rescue robot composed of a mobile robot and a snake robot, *Advanced Robotics*, vol. 34, no. 2 2020, pp. 132–139, 2020.

[21] C. Giovannangeli, P. Gaussier, and G. Desilles, “Robust mapless outdoor vision-based navigation,” in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 3293–3300. DOI: 10.1109/IROS.2006.282501.

[22] H. Kanayama, T. Ueda, H. Ito, and K. Yamamoto, “Two-mode mapless visual navigation of indoor autonomous mobile robot using deep convolutional neural network,” in *2020 IEEE/SICE International Symposium on System Integration (SII)*, 2020, pp. 536–541. DOI: 10.1109/SII46433.2020.9025851.

[23] K. J. Worrall, “Guidance and search algorithms for mobile robots: Application and analysis within the context of urban search and rescue,” Ph.D. dissertation, University of Glasgow, 2008.

[24] T. Tomic, K. Schmid, P. Lutz, *et al.*, Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue, *IEEE Robotics Automation Magazine* [online], vol. 19, no. 3 2012, pp. 46–56, 2012. DOI: 10.1109/MRA.2012.2206473.

[25] M. C. Akay, “Active slam with informative path planning for heterogeneous robot teams,” Ph.D. dissertation, Fen Bilimleri Enstitüsü, 2020.

[26] A. Jacoff, E. Messina, B. Weiss, S. Tadokoro, and Y. Nakagawa, “Test arenas and performance metrics for urban search and rescue robots,” in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 4, 2003, 3396–3403 vol.3. DOI: 10.1109/IROS.2003.1249681.

[27] T. Twins, *Husky*, 2023. available from: `https://github.com/Tinker-Twins/Husky.git`.

[28] J.-w. Choi, R. Curry, and G. Elkaim, "Real-time obstacle-avoiding path planning for mobile robots," in *AIAA Guidance, Navigation, and Control Conference*, 2010, p. 8411.

[29] lh9171338, *Astar*, 2021. available from: `https://github.com/lh9171338/Astar.git`.

[30] R. Planning, *Navigation*, No Year Provided. available from: `https://github.com/ros-planning/navigation.git`.

[31] L. Robotics, *Darknet ros*, 2021. available from: `https://github.com/leggedrobotics/darknet_ros.git`.

[32] J. Lee and B. Ahn, Real-time human action recognition with a low-cost rgb camera and mobile robot platform, *Sensors*, vol. 20, no. 10 2020, p. 2886, 2020.

[33] C. Robotics, *Cpr gazebo*. available from: `https://github.com/clearpathrobotics/cpr_gazebo.git`.

[34] M. Ivašić-Kos, M. Krišto, and M. Pobar, "Human detection in thermal imaging using yolo," in *Proceedings of the 2019 5th International Conference on Computer and Technology Applications*, 2019, pp. 20–24.