Journal of Robotics and Control (JRC)
Volume 4, Issue 6, 2023
ISSN: 2715-5072, DOI: 10.18196/jrc.v4i6.18392

Design and Development of swarm AGV's alliance for Search and Rescue operations

Ratan Pyla [1], Vikranth Pandalaneni [2], P Jeevan Narayana Raju [3], Guga Priya G [4*]
[1,2,3,4] School Of Electroics Enginnering, Vellore Institute of Technology, Chennai, India
Email: [1] pyla.ratan2020@vitstudent.ac.in, [2] pandalaneni.vikranth2019@vitstudent.ac.in, [3] jeevan.narayanaraju2019@vitstudent.ac.in, [4] gugapriya.g@vit.ac.in
[*]Corresponding Author

***Abstract*—Rapid response is essential for saving lives in search and rescue operations since the amount of time is critical. In this project, a swarm of autonomous ground vehicles (AGVs) equipped with ROS-based software architecture will be designed and built for rapid search and rescue missions. The swarm of AGVs will function autonomously to navigate through challenging areas and be outfitted with a variety of sensors, including cameras, and LIDAR. The proposed system will be capable of performing 2D mapping, live video surveillance, autonomous navigation, victim/object detection, and two-way audio communication. The goal of the project is to reduce the risk to human life in dangerous areas by providing a quick and efficient response system for search and rescue operations. A centralized management system with a de-centralized module will be created as part of the project to keep an eye on and manage the AGV horde. However, there will be a functionality to take control of a specific robot in the swarm network when needed. In difficult areas where it might not be safe for humans to operate, the suggested method will enable quick and efficient search and rescue operations.**

***Keywords*—Search and Rescue, De-centralized Control, Swarm Robotics, Robot Operating System(ROS), Mobile Robot.**

I. Introduction

Swarm robotics has emerged as a promising field that draws inspiration from the collective behaviour observed in social insects, such as ants, bees, and termites. It aims to design and control a large group of relatively simple robots, known as a swarm, to perform complex tasks collectively [1]. This research delves into focusing on its application in search and rescue operations.

Swarm robotics is rooted in the concept of swarm intelligence, which refers to the ability of a group of simple agents to exhibit complex behaviours through local interactions and self-organization. This concept draws inspiration from the natural world, where social insects achieve remarkable feats through collective decision-making and collaboration. [2] By applying these principles to robotics, researchers aim to develop systems that can exhibit similar emergent behaviours, adaptability, and robustness.

Identify applicable funding agency here. If none, delete this.

The field of swarm robotics has gained significant momentum in recent years, with numerous studies and advancements contributing to its growth. One of the primary motivations behind swarm robotics is the idea that a group of simple and relatively inexpensive robots can outperform a single complex robot in terms of efficiency, fault tolerance, and scalability [3]. By distributing tasks among swarm members, swarm robotics offers the potential to tackle complex tasks in a parallel and cooperative manner.

Swarm robotics has garnered significant attention due to its potential applications in various domains, including search and rescue, environmental monitoring, exploration, and industrial automation [4]. The ability of a swarm to distribute tasks, work in parallel, and provide fault tolerance makes it an attractive solution for complex and dynamic scenarios [5]. By leveraging the collective intelligence of the swarm, these systems can achieve superior performance compared to individual robots.

In this research project, a swarm of autonomous ground vehicles (AGVs) equipped with a ROS-based software architecture is designed for rapid search and rescue missions. The swarm operates autonomously, navigating through challenging environments, and is equipped with a range of sensors, including cameras and LIDAR, to gather relevant information. [6] The proposed system encompasses a detailed and unique architectural design that includes existing algorithms, hardware, and software to make a dedicated application of swarm robotics make possible.

The success of an architecture of swarm robotics relies heavily on the design and implementation of effective algorithms. These algorithms play a crucial role in achieving efficient coordination, decision-making, and task allocation within the swarm [7]. Various algorithms can be employed to facilitate swarm behaviour, such as consensus algorithms, behaviour-based algorithms, or optimization algorithms. For example, path planning algorithms, such as particle swarm optimization (PSO), ant colony optimization, or artificial potential field methods, are employed to ensure optimal navigation while avoiding obstacles and collisions. The algorithms leverage swarm intelligence principles to enable adaptive and robust behaviour within the swarm.

The hardware components of the proposed system include autonomous ground vehicles (AGVs) equipped with sensors and communication modules. The AGVs are designed to withstand challenging terrains and adverse conditions typically encountered in search and rescue operations. They are equipped with cameras and LIDAR sensors to capture visual and depth information, enabling mapping, object detection, and victim identification [8]. Additionally, the AGVs may incorporate other sensors, such as inertial measurement units (IMUs) for localization and navigation, and communication modules for inter-robot communication. The hardware is carefully selected and configured to ensure reliable and accurate data acquisition and processing within the swarm.

The software framework employed in the architecture design is based on the Robot Operating System (ROS), a popular open-source platform for robotic development. ROS provides a flexible and modular architecture that enables seamless integration of various components, including perception, control, communication, and decision-making modules. The software framework leverages ROS capabilities to facilitate real-time data processing, communication between swarm members, and centralized management of the swarm. The swarm can achieve distributed coordination, collaborative decision-making, and efficient communication through ROS .

The proposed system aims to accomplish several key functionalities, including 2D mapping, live video surveillance, autonomous navigation, victim/object detection, and two-way audio communication. These functionalities are crucial in search and rescue operations, where rapid response and accurate information are vital for saving lives [9]. The integration of these capabilities enables the swarm to operate effectively in complex and hazardous environments, providing critical information to rescue teams and reducing the risk to human life.

Therefore, swarm robotics represents a fascinating field that draws inspiration from nature to develop autonomous systems capable of collective behaviour and adaptive responses [10]. The proposed system for search and rescue operations utilizes a swarm of autonomous ground vehicles equipped with advanced algorithms, carefully selected hardware components, and a software framework based on ROS. This system enables efficient coordination, robust navigation, real-time data acquisition, and communication, ultimately enhancing the effectiveness of search and rescue missions. The integration of swarm robotics principles holds tremendous potential in revolutionizing the field of search and rescue, providing rapid and efficient response systems for saving lives in dangerous and challenging situations.

The research objectives of this study are twofold. While it aims to design and build a swarm of autonomous ground vehicles (AGVs) equipped with a ROS-based software architecture for search and rescue missions, the focus is on developing a system that can navigate through challenging areas, perform 2D mapping, provide live video surveillance, autonomously detect victims/objects, and establish two-way video and audio communication. Additionally, the objective is to evaluate the effectiveness and efficiency of the proposed swarm system in search and rescue scenarios, specifically assessing its performance in terms of response time, mapping accuracy, victim/object detection capabilities, and overall operational robustness.

To achieve these research objectives, several research questions will be addressed:

- How can swarm robotics principles and algorithms be effectively applied to search and rescue operations?
- What are the key hardware components required to build an autonomous ground vehicle (AGV) swarm for search and rescue missions?
- How can the Robot Operating System (ROS) be leveraged to create a software framework that enables distributed coordination, communication, and centralized swarm management?
- What are the optimal algorithms for path planning, mapping, victim/object detection, and two-way audio communication within the swarm?
- How does the proposed swarm system perform in terms of response time, mapping accuracy, victim/object detection capabilities, and overall operational robustness in search and rescue scenarios?

Addressing these research questions will contribute to the existing body of knowledge in swarm robotics and its application in search and rescue operations. The outcomes of this research will provide insights into the feasibility, effectiveness, and potential limitations of using swarm robotics for rapid response and efficient search and rescue missions.

Therefore this research paper introduces the field of swarm robotics and its application in search and rescue operations. The proposed system, comprising a swarm of autonomous ground vehicles equipped with a ROS-based software architecture, aims to enhance the effectiveness and efficiency of search and rescue missions. The paper has provided an overview of the context and background of swarm robotics, described the proposed system's algorithms, hardware components, and software framework, and explicitly addressed the state of research objectives and research questions. The subsequent sections will delve further into the methodology, experimental setup, results, and discussion to comprehensively analyze the performance and capabilities of the proposed swarm system in search and rescue scenarios.

II. Literature Survey

Swarm robotics is an interdisciplinary field that focuses on the study of large groups of relatively simple robots, known as swarms, working together to accomplish tasks. Inspired by the collective behavior observed in social insects, such as ants and bees, swarm robotics aims to understand and harness the principles underlying self-organization and emergent intelligence in biological systems. By designing decentralized algorithms and control strategies along with dedicated communications, understanding swarm intelligence, and exploring search and resume, swarm robotics seeks to create artificial systems that exhibit robustness, adaptability, and scalability.

To discuss the understanding, The review [11] presents the evolution of swarm robotics beyond optimization algorithms and highlights the challenges in hardware, autonomy, explainability, and trust. It emphasizes the potential for swarm robotics to transition from the lab to real-world applications with advancements in materials, AI, and user acceptance, while [12] explores swarm robotics as an engineering discipline, proposing systematic procedures for modeling, designing, verifying, and operating swarm robotics systems. It discusses the limitations of swarm robotics and suggests future research directions.

Further exploring learning, behavior, and social aspects,focus on understanding the social and behavioral aspects of swarm robotics [13]. This paper explores the implementation of social learning algorithms in swarm robotics and highlights the potential for complex social behaviors and addressing real-world challenges. It discusses the application of distributed online reinforcement learning in a collective of robots and the potential of dense robot swarms as an active matter. Additionally, [14] investigates user perceptions of robot swarms in real-world applications and provides design principles for their deployment. It emphasizes the importance of mutual shaping between users and technology developers to ensure trust and successful adoption of swarm robotics technology.

Focusing on the development of algorithms [15] explores the use of artificial pheromone systems in swarm robotics, drawing inspiration from the effective utilization of pheromones in social insects. It demonstrates the feasibility of the proposed pheromone system for swarm robotic applications and highlights the potential for realistic emulation of environmental effects on pheromone distribution. Understanding control system planning, [16] investigates the self-assembly process of a distributed robotic system called swarm-bot, where autonomous mobile robots can self-assemble into a larger structure. The paper provides a comprehensive study of the swarm-bot's self-assembling capabilities and highlights its effective connection mechanism and advanced sensing and communication devices.

Additionally, [17] presents a procedure for designing and verifying the local behavior of robots with limited cognition, allowing them to self-organize into desired global patterns. The paper introduces a formal proof procedure to verify the emergence of the desired pattern from the local actions of the robots. It demonstrates the feasibility of implementing the behavior in real robots and identifies challenges for future research. Moreover, [18] presents a consensus algorithm for artificial swarms of primitive agents, enabling them to collectively make decisions in an uncertain environment. The paper introduces a probabilistic model that incorporates individual observations, local interactions, and stochasticity, demonstrating its effectiveness in achieving reliable decision-making in swarm robotics applications.

In general, we gain a more comprehensive understanding of swarm robotics. This highlights the conceptual frameworks, behavioral aspects, and algorithmic foundations of swarm robotics. They provide insights into the challenges and potential applications of swarm robotics, explore the dynamics of social interactions among robotic agents, and delve into the technical advancements in algorithm development, control strategies, and self-organization. The interdisciplinary nature of swarm robotics makes it a fascinating field with significant potential for real-world applications.

In a swarm, communication plays a vital role in facilitating cooperation, information sharing, and synchronization among the individuals. Key aspects include formation and tracking [19]–[21]. [19] presents the Local Charged Particle Swarm Optimization (LCPSO) algorithm for tracking a moving target with scalar information. [20] proposes a leader-follower model for exploration tasks in robot swarms with limited communication. [21] introduces a decentralized controller for swarm aggregation without communication or global positioning. These papers address the challenges of maintaining stable formations, optimizing group partitioning, and achieving swarm aggregation while considering communication limitations. Further, it is important to understand swarm interaction and behavior. The work focuses on analyzing swarm interaction and behavior. [22] proposes a swarm interaction network framework for understanding and comparing swarm-based algorithms. It examines the structure of social interaction in different algorithms, offering a comprehensive understanding of swarm behavior. [23] investigates the impact of communication topologies on Particle Swarm Optimization (PSO) performance, emphasizing the importance of considering local topologies for better optimization results. In general, information sharing in communication is another vital aspect. [24] presents a method for swarm self-monitoring and aggregate information display in a distributed manner, compensating for limited communication. Paper [25] proposes a multiple-access energy transfer solution using code division for multiple-access wireless power transfer for energy exchange among heterogeneous robot swarms. These papers focus on improving information sharing and energy transfer within swarms while considering communication limitations.

Based on swarm control and optimization techniques, [26] presents a cooperative control technology for robot formation using an Internet of Things (IoT) platform, employing a particle swarm optimization deep learning algorithm for improved accuracy and efficiency. The paper [27] addresses the challenges of spectrum sharing in unmanned swarm communication systems (USCS) and proposes an intelligent spectrum management technique empowered by machine learning for effective spectrum sharing. These focus on control, optimization, and efficient resource management in swarm communication systems.

In summary, the papers cover a wide range of topics related to swarm communication, formation, tracking, behavior analysis, optimization, and resource management. They provide insights into overcoming communication constraints, improving swarm coordination, understanding swarm dynamics, and enhancing swarm performance through intelligent algorithms and frameworks. The research presented in these papers contributes to the advancement of swarm robotics, optimization algorithms, and communication systems for both theoretical understanding and real-world applications.

Swarm applications have numerous potential uses across various fields, including robotics, military operations, agriculture, disaster response, and more. Among them, search and rescue refers to the coordinated efforts and techniques used to locate and provide aid to individuals who are in distress or missing. SAR operations are typically conducted in emergency situations, such as natural disasters, wilderness incidents, maritime accidents, or urban disasters. Based on Robot Design and Implementation for search and rescue (SAR), The discussion in the papers focuses on the design, development, and implementation of mobile robots specifically tailored for search and rescue operations. It [28] discusses the design process of a mobile robot that aims to detect and recover victims efficiently while facilitating communication with emergency management centers. Research [29] presents the design and implementation of Karo, a mobile rescue robot with high mobility and dexterity for urban search and rescue missions; another [30] focuses on the development of ResQbot 2.0, a robot designed for casualty extraction tasks employing a loco-manipulation approach synchronized with a conveyor belt and a mobile base. These papers contribute to SAR by providing practical solutions for enhancing the capabilities and performance of robots in rescue operations.

Additional focus on sensing, mapping, and reconstruction techniques for SAR environments brings Paper [31], a mobile robot application that constructs semantic and metric maps using point-based deep learning and RTAB-Map. Paper [32] explores the use of mobile robots with gas-sensing capabilities for real-time gas discrimination and mapping in emergency response scenarios. Paper [33] discusses the reconstruction of 3D objects using visual SLAM techniques for search and rescue disaster scenarios. These contribute to SAR by providing methods for accurate mapping, gas detection, and environmental understanding, which are crucial for effective rescue operations in challenging environments.

Another set focuses on swarm robotics and collaborative approaches for SAR. Research [34] introduces a separable robot system consisting of a mobile robot and a snake robot, highlighting their combined capabilities as a mobile manipulator and a multi-agent system. A study [35] presents a collaborative approach between a ground robot and an aerial robot for mapping and localization in search and rescue missions. [36] proposed a systematic approach for swarm robots to search for multiple targets simultaneously using particle swarm optimization and artificial potential field techniques. The work contributes to SAR by exploring the potential of swarm robotics and collaboration between different robot platforms to enhance search and rescue capabilities.

Research contributes to SAR by also addressing key challenges in sensor fusion and decision-making, enabling robots to operate effectively in dynamic and uncertain environments. The information in [37] provides a comprehensive review of challenges and solutions related to sensor fusion methods in autonomous mobile robots. Research from [38] proposes a hybrid strategy for target search using both static and mobile sensors, optimizing deployment planning and motion planning. A study [39] introduces the LEACH-R protocol with a cache strategy, addressing communication challenges in mobile robot swarms. Paper [40] presents a modeling framework using interval-valued neutrosophic analysis for decision-making in autonomous search and rescue missions. In general, in search and rescue scenarios, it ultimately aids in saving lives and mitigating the impact of disasters.

Another aspect of swarm robotics is swarm intelligence. That refers to the collective behavior of decentralized, self-organized systems in which individuals, known as agents, interact locally with their environment and with each other to achieve common goals. It is inspired by the behavior of social insect colonies, such as ant colonies and bee swarms, where individual agents exhibit simple behaviors but collectively accomplish complex tasks. Both works [41] and [42] collectively contribute to the this advancement of swarm intelligence by shedding light on parallelization strategies and showcasing their potential in solving complex optimization problems and improving system performance. The comprehensive review and analysis provided serve as valuable references for researchers interested in implementing parallel swarm intelligence metaheuristics. Meanwhile, it is also demonstrating the practical benefits of swarm intelligence in addressing challenges related to load balancing, secure data management, and resource optimization in emerging technologies like the Internet of Everything and edge computing.

Robotic path planning is another essential research area in robotics, with the goal of Some works encompass various aspects, such as enhanced ant colony algorithms, self-assembly in swarm robotics, multi-robot systems, target searching, adaptive optimization algorithms, intelligent navigation strategies, and specialized programming languages for swarm robotics.

We can classify and understand the work done so far in the following categories

1) Swarm Robotics Platforms and Hardware: Several research's (e.g., [43]–[45]) introduce innovative swarm robot platforms and hardware solutions. These platforms combine sensing, motion, computing, communication, and power management capabilities, providing a comprehensive solution for swarm robotics. The use of low-cost commercial off-the-shelf components makes these

platforms cost-effective. Integration with the Robot Operating System (ROS) enhances compatibility and facilitates easy integration with the ROS software ecosystem. The SwarmUS platform ([45]) specifically focuses on standardizing swarm robotics platforms.

2) Optimization Algorithms for Path Planning and Coordination: Researches such as [44], [46]–[49] propose various optimization algorithms for path planning, motion coordination, and task allocation in swarm robotics. These algorithms aim to generate optimal trajectories, minimize path length, ensure collision avoidance, and improve energy utilization. Examples include improved versions of particle swarm optimization (IPSO) with evolutionary operators (EOPs), teaching-learning-based optimization (TLBO), and machine learning-based algorithms. These algorithms demonstrate superior performance in terms of arrival time, safety, energy efficiency, and adaptability in complex environments.
3) Multi-Robot Systems and Coordination: Researches like [50]–[52] delve into the coordination and planning aspects of multi-robot systems (MRS). They address challenges related to mission planning, task allocation, motion planning, and coordination among multiple robots. Centralized and decentralized approaches are explored to optimize mission time, reduce power loss in communication, and improve overall efficiency. The coordination algorithms discussed in [52] and [53] integrate techniques such as convolutional neural networks (CNN), fuzzy logic, and hybrid optimization algorithms.
4) Simultaneous Localization and Mapping (SLAM): The topic of SLAM in swarm robotics is covered in [54]. The Swarm-SLAM system presented in this paper is designed to facilitate mapping and localization within swarm robotics. It employs decentralized and sparse techniques to achieve scalability and flexibility. Additionally, novel inter-robot loop closure prioritization techniques are introduced to reduce communication requirements and accelerate convergence.
5) Specific Applications: Some research focuses on specific applications of swarm robotics. For example, [55] discusses self-organized flocking mechanisms for applications like agri-robotics, [53] addresses swarm protection systems for migrants in dangerous territories, and [56] introduces a mobile robot designed for inspecting confined environments. These studies highlight the potential of swarm robotics in addressing specific challenges and offer insights into the design and implementation of specialized systems.

Overall, the discussed research contributes to the advancement of swarm robotics by introducing innovative platforms, optimization algorithms, coordination frameworks, and planning methodologies. They showcase the capabilities, efficiency, and adaptability of swarm robots across various applications.

III. METHODOLOGY

The proposed solution is to design and develop a swarm of AGVs for search and rescue operations. Each robot was equipped with a Raspberry Pi as the central processor and an Arduino to control the motors [57]. A 3D LiDAR was also added to the robot to enable obstacle avoidance. So, The AGV swarm will travel through challenging environments on its own and also help in mapping the environment. Additionally, camera, microphone, and speaker modules were added to facilitate video surveillance and two-way audio communication. [58] In order to keep an eye on and manage the horde of AGVs, A mobile app (Flutter) will be developed which acts as the user interface for the swarm management software that is developed, from which the operator can take control of the de-centralized control system when and where required.

A. Block Diagram

The proposed block diagram as shown in Fig. 1 for the swarm of AGVs for search and rescue operations consists of four main components:

1) Autonomous Guided Vehicles (AGVs): The AGVs include a chassis, motors, a microcontroller and sensors. The AGVs are configured to navigate in an unknown environment and are intended to move autonomously. The AGVs use Lidar, that help them avoid collisions and detect impediments. A camera that sends live video to the Control Center, a microphone, and a speaker that allows two-way audio communication is also included in the AGVs.

2) Control Center (Mobile Application): The Control Center serves as the user interface for the swarm management software and the operators has the ability to take over control of the decentralized swarm network when necessary from the mobile application. The operators can interact with the AGVs and also get real-time data from the AGV's. The Control Center gives the operators access to a 2D map of the area, a live video stream, and two-way audio communication with the AGVs.

3) Data Management (Cloud): It utilizes wireless communication protocols such as HTTP for live video, WebRTC for audio, and MQTT protocol for sending commands to the AGVs from the cloud. The software coordinates the operations of the AGVs, controls their movements, and carries out the search and rescue operation.

4) Central Control Node (ROS): Here the communication is based on the ROS2 message-passing paradigm.The control node, which serves as the system's hub and is in charge of organising each robot node's activities, The control node is also responsible for generating a global map of the area using the local maps generated by individual map.

B. Software Architecture

1) System Overview: The proposed system is a swarm of autonomous ground vehicles (AGVs) designed for search and

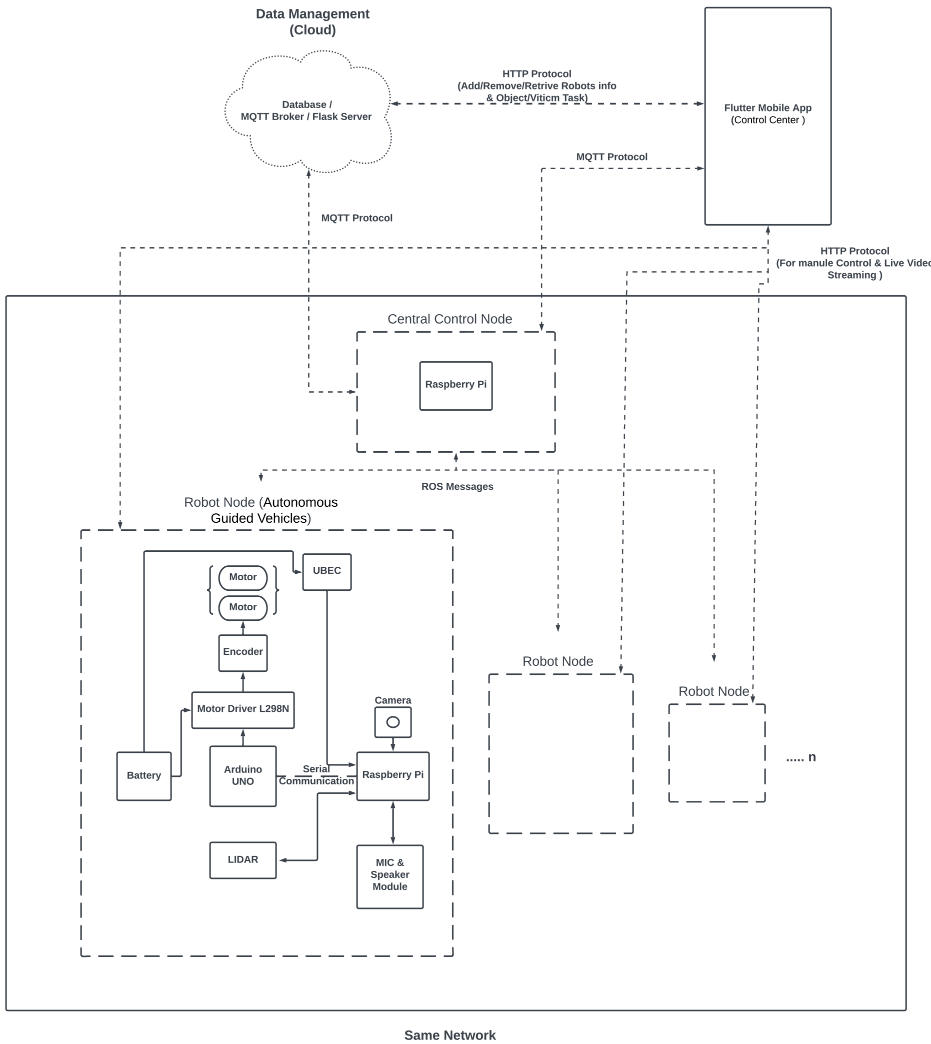

Fig. 1. Block Diagram

rescue operations. Each AGV is equipped with a Raspberry Pi as the central processor, an Arduino for motor control, and a 3D LiDAR for obstacle avoidance and mapping [43]. The system also includes camera, microphone, and speaker modules for live video streaming and two-way audio communication. The AGVs are designed to function autonomously and can explore and map an area on their own. The system is managed by a centralized control node, and the AGVs can be controlled manually through a mobile app when necessary.

2) Working Principle:

- Live video and two way audio streaming: To achieve the objectives of enabling live streaming of video footage and establishing two-way audio communication between a Raspberry Pi and a Flutter mobile app, we implemented an integrated methodology that adheres to IEEE standards. [59] This methodology involved the integration of specific components, including a carefully selected USB camera, ISD1820 modules, and the UV4L streaming server with the two-way audio/video intercom recorder (WebRTC) feature.

Firstly, we integrated a USB camera with the Raspberry Pi to facilitate live video streaming. The USB camera was selected based on compatibility and desired video quality, and we ensured the proper connection and configuration of the camera.

Next, we configured the UV4L streaming server on the Raspberry Pi, leveraging its versatile capabilities for audio and video streaming. [60]This server enabled us to capture video frames from the USB camera, encode them with an appropriate codec, and seamlessly stream them to the

Flutter mobile app.

For two-way audio communication as shown in Fig. 2, we integrated the ISD1820 module with the Raspberry Pi. These module consist of a microphone and a speaker, which are commonly used for audio recording and playback. We meticulously ensured the correct wiring and configuration of the ISD1820 modules.

To achieve secure and real-time two-way audio communication, we leveraged the two-way audio/video intercom recorder feature of the UV4L streaming server, which is based on the WebRTC protocol. [61] WebRTC is an open-source framework that enables peer-to-peer audio and video communication directly within web browsers and mobile apps.

Within the Flutter mobile app development, we integrated the WebRTC package, which provided the necessary APIs and functionalities for establishing and managing the WebRTC connection. We implemented the required logic to establish a secure WebRTC connection with the UV4L streaming server, enabling bidirectional transmission of audio and video data. The user interface components of the Flutter app were thoughtfully designed to display the live video feed and facilitate seamless audio communication.

To address security concerns, we implemented robust encryption measures to safeguard the transmitted audio and video data between the Raspberry Pi and the Flutter app. Secure communication protocols such as HTTPS were employed.

By integrating the USB camera, ISD1820 modules, and the UV4L streaming server with the two-way audio/video intercom recorder (WebRTC) into our methodology, we successfully achieved live streaming of video footage and two-way audio communication between a Raspberry Pi and a Flutter mobile app.

To ensure optimal performance and minimal latency in our live streaming and two-way audio communication system, we implemented several strategies. Firstly, we carefully selected a USB camera that met our specific requirements for video quality and performance. This enabled us to optimize the camera's capturing and encoding capabilities for efficient streaming. [62] Additionally, we leveraged the UV4L streaming server's two-way audio/video intercom recorder (WebRTC) feature, which utilizes advanced codecs and protocols specifically designed for low latency and real-time communication. By incorporating WebRTC, we significantly reduced latency and achieved near-instantaneous audio and video transmission between the Raspberry Pi and the Flutter app. Furthermore, we employed video compression techniques by converting video frames to JPEG format using the MJPEG protocol. This approach effectively reduced bandwidth requirements without sacrificing significant video quality, ensuring a seamless and real-time streaming experience for users while minimizing perceptible delays in video and audio playback.

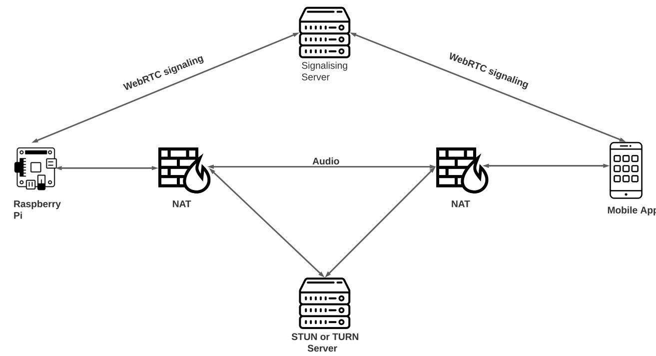

Fig. 2. 2-Way Audio Communication Workflow

- Autonomous Exploration: The autonomous exploration module is implemented using ROS navigation stack. The module consists of a 3D LIDAR sensor, which scans the environment and generates a 2D occupancy grid map using the SLAM algorithm [63]. The map is then used by the ROS navigation stack to plan and execute the robot's autonomous exploration. The frontier exploration algorithm involves several components that work together to enable the robot to navigate and explore unknown environments. These components include:
  1) Frontier detection is the initial phase in the process. This is often accomplished by locating sections of the map where there is a clear transition between explored and unknown territory. Then, on the map, the boundaries are shown as points or groups of points.
  2) Frontier selection: After identifying the frontiers, the robot must choose which one to explore next. This is accomplished by assessing each frontier in terms of its accessibility, proximity to the robot, and other elements including the possibility of discovering intriguing characteristics.
  3) Path planning: Once a frontier is selected, the robot plans a path to reach that frontier while avoiding obstacles and other hazards. This involves generating a sequence of waypoints that the robot can follow to reach the frontier.
  4) Control: Finally, the robot uses feedback control to follow the planned path and reach the selected frontier.
- Local Mapping: The local mapping module is implemented using the slam gmapping package in ROS. The module consists of a 3D LIDAR sensor and a mapping algorithm [64]. The LIDAR sensor scans the environment and generates a 2D occupancy grid map, which is stored

Algorithm 1 ROS Frontier Exploration Algorithm

Input: Robot position, map, and sensor data
Output: Explored map
while *unexplored frontiers exist* **do**
  detect frontiers using sensor data and frontier detection algorithms;
  select the best frontier to explore;
  plan a path to the frontier using path planning algorithm;
  while *robot has not reached the frontier* **do**
    move robot along the planned path while avoiding obstacles;
    update the map as the robot explores the frontier;
  end
  mark the frontier as explored;
end

Algorithm 2 slam_gmapping

Input: Laser scan data, robot odometry data
Output: Occupancy grid map
$map_initialized = False$;
while *SLAM is in progress* **do**
  scan = receive laser scan data;
  odom = receive robot odometry data;

  aligned_scan = align_scan_with_odom(scan, odom) estimated_pose = particle_filter(aligned_scan, odom)
  if *not map_initialized* **then**
    initialize_occupancy_grid_map(estimated_pose) map_initialized = True
  end
  else
    update_occupancy_grid_map(aligned_scan, estimated_pose)
  end
  if *mapping time or distance limit has been reached* **then**
    end mapping process
  end
  if *SLAM optimization cycle has been reached* **then**
    optimize_occupancy_grid_map()
  end
  publish occupancy grid map and other data
  shutdown ROS node and exit algorithm;
end

in the robot's memory. The local map can be accessed by the central control node. The robot's laser scan and odometry topics are subscribed to by the slam gmapping package, which then uses a particle filter algorithm to infer the robot's position and orientation from these sensor values. The estimated pose and the data from the laser scan are then combined to create an occupancy grid map of the surroundings using the Gmapping technique. [65], [66] The occupancy grid map is a representation of the environment in a grid format, where each cell in the grid represents the probability of that cell being occupied by an obstacle. The package also performs loop closure detection and optimization to correct errors in the map and robot pose estimation. The software continues to estimate the robot's position and orientation in real-time while updating the map as it advances. Additionally, it enables the saving and loading of maps, which is advantageous for ongoing mapping and localization work [67]. The slam gmapping package, which is widely used in robotic applications like autonomous navigation, mapping, and exploration, offers an effective and precise solution for SLAM employing a LIDAR sensor.
The slam_gmapping package, which is widely used in robotic applications like autonomous navigation, mapping, and exploration, offers an effective and precise solution for SLAM employing a LIDAR sensor.

- Object Detection: Our project incorporates an object detection module that plays a vital role in the overall system. This module enables users to define specific object detection tasks, allowing for targeted identification and analysis of objects [68]. Object detection is crucial for applications such as surveillance, security, and real-time monitoring. By accurately detecting and tracking objects of interest, our system enhances situational awareness and enables timely decision-making [69].
  - Integration of Cvlib Library: The integration of the Cvlib library was motivated by its reputation for accuracy and real-time performance in object detection tasks. Cvlib provides a comprehensive set of pre-trained deep learning models, including renowned models like YOLO (You Only Look Once). [70] These models have been trained on large-scale datasets and are widely recognized for their capability to detect objects with high accuracy and efficiency. By leveraging the Cvlib library, we tap into its extensive collection of models, ensuring reliable and effective object detection capabilities.
  - User Input and Task Setting:
    Our system offers users a user-friendly interface through the Flutter mobile app, allowing them to precisely define object detection tasks. Users have the flexibility to select specific objects or define custom classes for detection, tailoring the system to their specific needs. This intuitive user input feature empowers users to focus on detecting objects of interest and facilitates seamless interaction with the system.
  - Image Processing and Detection:
    The Raspberry Pi camera serves as a crucial component for capturing images in the object detection process. These images are subsequently processed using the Cvlib library and the selected deep learning

model. Leveraging the advanced functions provided by the Cvlib library, our system performs robust object detection on the captured images. This involves analyzing the images to identify the precise location and class labels of the detected objects. By effectively utilizing the computational capabilities of the Raspberry Pi and the Cvlib library, we achieve accurate and efficient object detection.

– Alert Generation:
Real-time alert generation is a key feature of our system. Upon detecting the specified objects, the system promptly generates alerts to notify the user. These alerts are delivered in the form of push notifications to the Flutter mobile app, ensuring real-time updates on the detection results. By receiving these alerts, users can take immediate and appropriate actions based on the detected objects, enhancing their ability to respond to critical situations swiftly.

– Performance Considerations:
Performance optimization plays a critical role in the effectiveness of object detection systems. To ensure efficient object detection, we employed several strategies:
  1) Model Selection and Optimization: The deep learning model selection process involved evaluating the performance characteristics of various models in the Cvlib library. Factors such as accuracy, processing speed, and resource efficiency were carefully considered. By selecting an optimized model, such as YOLO, we strike a balance between high accuracy and real-time processing speed, maximizing the overall efficiency of our object detection system.
  2) Image Preprocessing: Prior to inputting images into the object detection model, we applied various preprocessing techniques to optimize performance. These techniques include resizing the images to a suitable resolution, normalizing pixel values, and employing specific techniques like image cropping or scaling. By preprocessing the images, we enhance detection accuracy and speed, ensuring effective object detection even in challenging conditions.

By incorporating these enhancements, we provide a comprehensive and coherent explanation of the object detection module. It offers a clear understanding of the significance of the module, the rationale behind integrating the Cvlib library, and the specific steps involved in the object detection process. The performance considerations and image preprocessing techniques further highlight our commitment to achieving accurate and efficient object detection in real-time scenarios.

- Global Mapping: In the implementation of the global mapping as shown in Fig. 3, Multirobot map merger ROS package, it is important to address certain considerations and drawbacks to ensure the accuracy and effectiveness of the approach [71]. The following enhanced methodology highlights the changes and improvements suggested:
  – Publishing Map Topics:
  To facilitate the merging of local maps from each robot [72], all robots should publish their local maps under the topic ¡robot_namespace¿/map. The topic name should be configurable but consistent across all robots. This consistent topic naming convention allows for seamless identification and data exchange within the swarm, ensuring efficient communication during the map merging process.
  – Addressing Unknown Initial Robot Poses:
  One of the primary challenges in this methodology arises when the initial positions of the robots are unknown. This lack of knowledge poses significant difficulties in accurately merging the local maps and creating a globally consistent map. To mitigate this drawback, the Multirobot map merger heavily relies on feature matching algorithms to estimate the transformation between grids. However, it is important to note that feature matching algorithms require a sufficient amount of overlapping space between the grids to achieve a high-probability match. Insufficient overlap can lead to noticeable discrepancies between the merged map and the physical environment.
  – Deployment Strategy and Overlapping Grids:
  To overcome the challenges posed by unknown initial robot poses, a carefully planned deployment strategy is essential. The robots should be deployed in close proximity to each other to ensure a minimum amount of overlapping grids. Sufficient overlapping space enhances the reliability of the feature matching process, significantly improving the probability of obtaining an accurate global map. The deployment strategy should be meticulously designed to position the robots in a manner that optimizes the overlapping grids, taking into consideration the specific environment and constraints of the system.
- Manual Control: The manual control module is an essential component of the robotic system, enabling users to interact with the robot and control its movements remotely [73].
In this research, the manual control module is implemented using a Flask server, which acts as the communication interface between the mobile app and the robot's control system. The methodology for the manual control module is described below, providing detailed insights into each step for research purposes:
  1) Flask Server Setup:

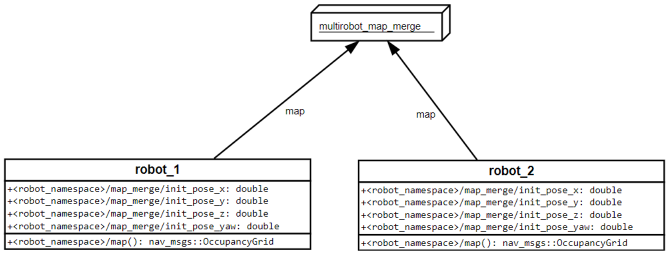

Fig. 3. Global Mapping

A Flask server is set up on the robot, establishing a robust and efficient communication channel between the mobile app and the robot's control system. The server configuration involves installing the necessary Flask dependencies and ensuring compatibility with the robot's operating environment, such as the programming language and libraries used [74]. Considerations are made to enhance security measures, such as implementing encryption protocols or authentication mechanisms, to protect the privacy and integrity of the communication channel. These security measures are crucial to prevent unauthorized access and potential malicious attacks.

2) User Interface Integration:
The mobile app provides a user-friendly joystick interface that enables intuitive control of the robot's movements. The joystick interface is carefully designed to facilitate precise and responsive control, ensuring an enhanced user experience.
The design considerations for the user interface include the layout, responsiveness, and visual feedback mechanisms [75]. These aspects are crucial for users to have a clear understanding of the robot's movement commands and to receive immediate feedback on the impact of their inputs.
The mobile app translates the user's joystick inputs into specific HTTP requests. The mapping of joystick movements to HTTP request parameters is determined to accurately represent the desired direction, speed, and other relevant movement parameters.

3) HTTP Request Handling:
The Flask server receives the HTTP request from the mobile app, which contains information about the desired movement, such as the direction and velocity. The server parses and extracts this information from the request to process it accordingly.
The request handling mechanism ensures the correctness and integrity of the incoming requests. The server performs validations and error checks to prevent unintended behaviors or potential vulnerabilities in the control system.
Error handling strategies are implemented to gracefully handle invalid or unexpected requests. This includes returning appropriate status codes or error messages to the mobile app, ensuring effective communication between the user and the robot.

4) Robot Control Actions:
Based on the received HTTP request, the Flask server triggers the appropriate control actions on the robot. These control actions include moving forward, backward, turning left or right, or stopping the robot's movement [58].
The server communicates with the robot's control system, transmitting the necessary commands to achieve the desired movement. This involves encoding the movement parameters into commands that the robot's control system can interpret and execute.
Advanced control techniques may be employed to optimize the precision and responsiveness of the robot's movements. These techniques could include feedback control mechanisms, such as proportional-integral-derivative (PID) controllers, to regulate the robot's speed and maintain stability during movements.

5) Feedback and Responses:
Once the robot performs the requested movement, the Flask server provides feedback to the mobile app, confirming the successful execution of the command. This feedback ensures effective communication between the user and the robot, providing confidence and assurance in the control process.
The feedback is conveyed through an HTTP response sent back to the mobile app. This response indicates the status or outcome of the requested movement, allowing the user to be aware of any errors, obstacles encountered, or successful completion of the movement.
Additionally, the feedback mechanism can include real-time updates on the robot's position or any other relevant information, enabling the user to have a comprehensive understanding of the robot's current state.
By implementing the manual control module using a Flask server and the HTTP protocol, users can interact with the mobile app's joystick interface to control the robot's movements remotely. The Flask server serves as an intermediary, receiving the user's commands and translating them into corresponding robot movements. This allows for manual control and navigation of the robot in real-time, offering researchers and users a powerful tool to explore the capabilities and potential applications of the robotic system.

3) Node Architecture: The system is divided into three main ROS nodes [76]: the robot node, the central control node, and the cloud server. The robot nodes are responsible for performing autonomous exploration, local and global mapping, object detection, live sensor readings, and live video streaming with two-way audio communication. The central control node manages the horde of AGVs, receives commands from the cloud server and mobile app, and communicates with individual robots to coordinate their actions. The cloud server serves as a bridge between the mobile app and the central control node, allowing the user to take manual control of the AGVs when necessary.

- Central Control Node: The central control node as shown in Fig. 4 is responsible for managing the swarm of robots and processing data generated by the individual robot nodes. [77] It subscribes and publishes to various topics to receive data from the robot nodes, such as map data, and also publishes messages to control the behavior of individual robot nodes, such as starting or stopping exploration or requesting a specific robot to perform a task.

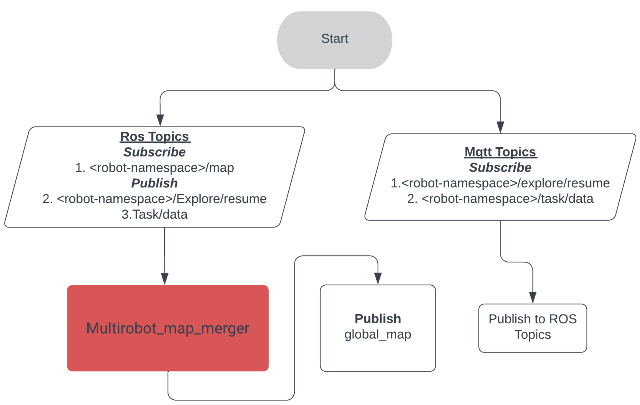

Fig. 4. Central Control Node

- Fig. 5 is Robot Node: Each robot node is responsible for exploring the environment autonomously using the frontier exploration package, then generating a local map using the G-mapping package and communicating with the central control node to share information and receive instructions. The robot node subscribes to various topics to receive messages from the central control node, such as requests to start or stop exploration. If the autonomous exploration is paused then the user can control the robot manually. If a task is set to detect any object or victim then it uses the CVlib object detection algorithm to detect the same and if it is detected then it publishes the notification to the MQTT broker topic. It also live streams the camera and microphone output using HTTP protocol with the help of IP tunneling.

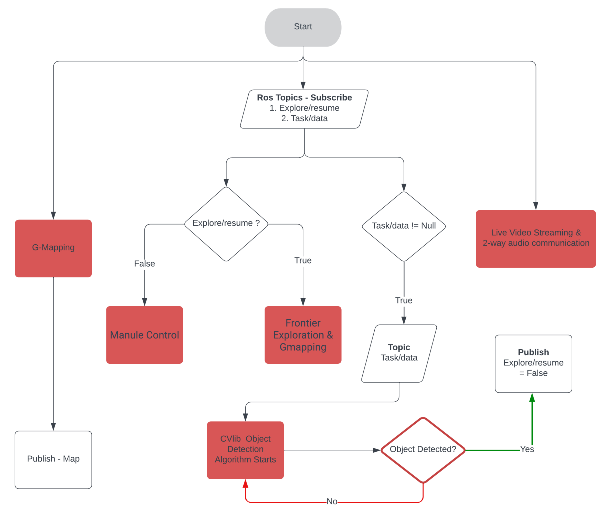

Fig. 5. Robot Node

4) Conclusion: In conclusion, the software architecture proposed for the swarm of AGVs for search and rescue operations consists of multiple ROS nodes communicating with each other through various protocols like MQTT, HTTP, and ROS topics [78]. The central control node is responsible for managing and controlling the swarm of AGVs, including sending commands for autonomous exploration, stopping exploration upon user request, and enabling manual control through a flask server. The robot nodes are responsible for live video streaming, two-way audio communication, local mapping, object detection, and live sensor readings. The cloud server acts as a bridge between the user requests and the central control node using HTTP requests. Overall, the proposed software architecture provides a flexible and efficient solution for search and rescue operations.

C. Workflow

1) Initialize the central control node, cloud server, and all the robot nodes.
2) The robot nodes will start the exploration process autonomously.
3) During exploration, each robot node will generate local maps and send them to the central control node.
4) The central control node will receive the local maps from all the robot nodes and combine them to create a global map of the environment.
5) The live video feed from each robot node will be live streamed and hosted on a web server using HTTP protocol.
6) The two-way audio communication module will allow the user to communicate with each robot node.

7) In case of any object or victim detection, the object detection module will send alert to the mobile app and also stoping the autnomous exploration of that specific robot.
8) If the autonomous exploration needs to be stopped for any specific robot node, the user can request it via the mobile app, which will send an HTTP request to the cloud server.
9) The cloud server will send an MQTT message to the central control node, which will publish it to the respective robot node to stop the exploration process.
10) The manual control module will be activated, and the user can manually control the robot node through a Flask server.

This workflow is an iterative process, and the exploration, mapping, detection, and monitoring will continue until the search and rescue operation is complete.

IV. Design and Implementation

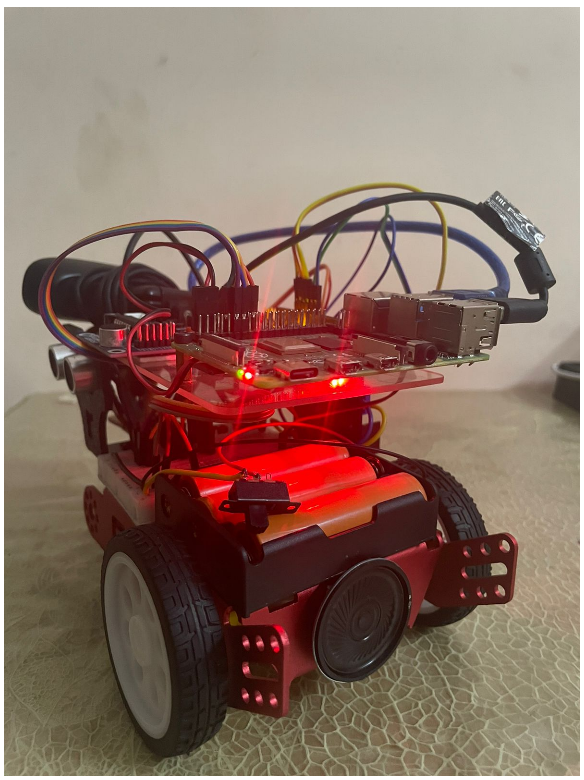

Fig. 6. Hardware

Here the brain of the robot is Raspberry Pi, which is connected to Arduino Uno for bidirectional serial communication via USB cable. The Sensors (Lidar, Camera, Mic and Speaker) are connected to Raspberry pi and actuators (L298N Motor Driver which is connected to both the motors) are connected to the Arduino Uno. And the battery is connected to L298N Motor Driver and UBEC which then connects to Raspberry pi as shown in Fig. 6.

The system's software architecture was developed utilizing a decentralized methodology [79], with each robot node in charge of its own autonomous exploration and environment mapping. The swarm of AGVs was controlled and monitored by the central control node, The user could also take over control of a particular robot in the swarm network if necessary.

The communication architecture was implemented using ROS topics and messages. Each robot node subscribed to the ”explore/resume” topic to receive a true or false message from the central control node to pause or resume its autonomous exploration [80]. The ”local map” topic, which the robot nodes broadcast, featured a map of the immediate area created by the robot's LiDAR sensor.

The HTTP and WebRTC protocols, respectively, were used to create live video streaming and two-way voice communication [81]. The web server streams the live video stream produced by the camera module, and the mobile app receives it via IP tunneling. Using a WebRTC server, the speaker and microphone enable two-way audio communication between the robot node and the mobile app by encoding and transmitting the audio in real time are shown in Fig. 7.

Each robot node had a Flask server that was used for manual control of a particular robot. The mobile app sent an HTTP request to the cloud server to stop the exploration of the desired robot. [82] The MQTT protocol was then used by the cloud server to send a message to the central control node, which subsequently broadcast the message to the particular robot node to halt its exploration. The user may then control the robot using the mobile app after the Flask server on the robot node turned on the manual control.

All things considered, the system was created and put into place to offer a quick and effective response system for search and rescue operations, with the potential to function in difficult areas where it might not be safe for humans to operate. The inclusion of ROS topics and messages made it easier for the system's nodes to communicate with one another, allowing the swarm of AGVs to work independently while simultaneously giving the user access to live video streaming and two-way voice.

V. Results and Discussions

In this section, we will present the results obtained from the implementation of the designed system and discuss their implications.

First, we assessed the autonomous exploration module's performance. The AGV swarm's performance in several gazebo simulation scenarios are shown in Fig. 9 was promising [83]. The 3D LiDAR sensor allowed the robots to avoid obstacles, travel through difficult terrain on their own, and map their surroundings as they moved through it. The exploration module

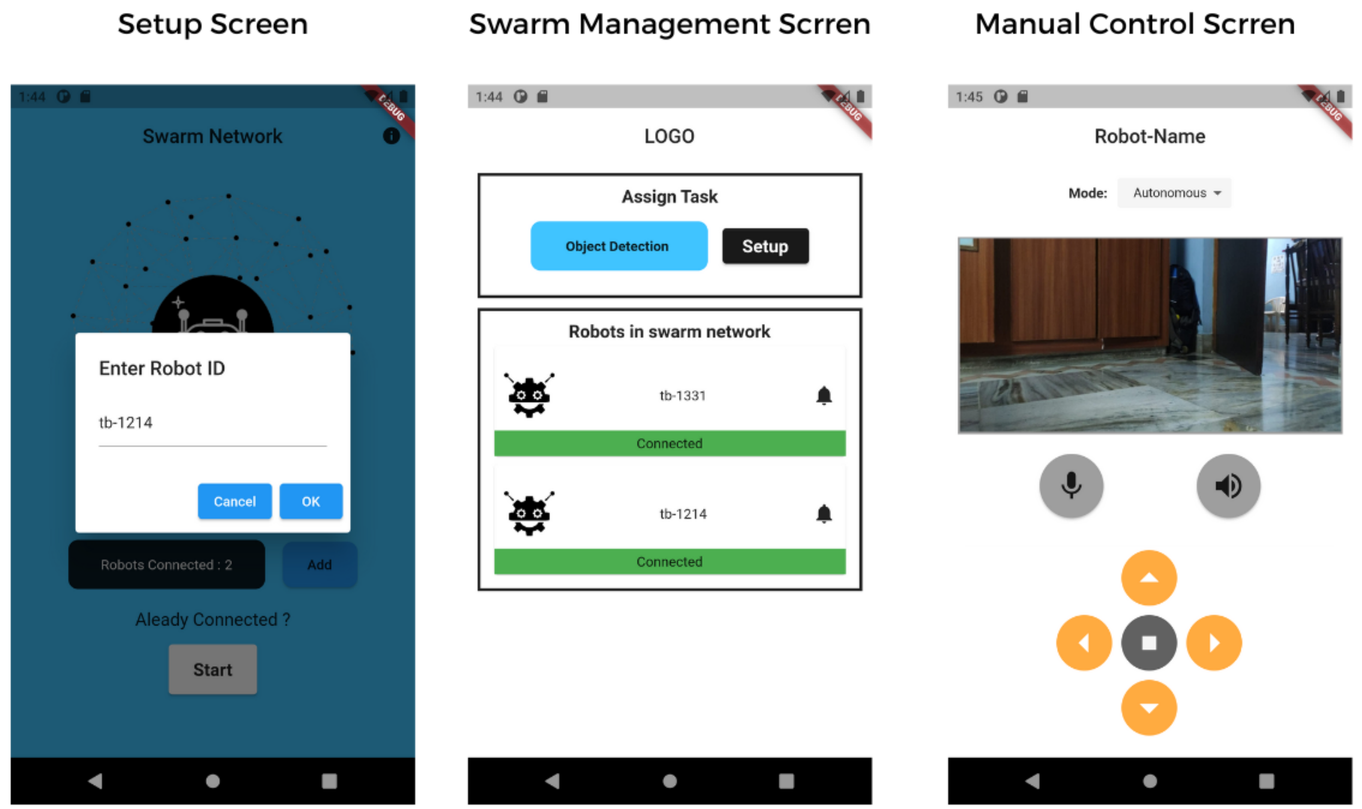

Fig. 7. Mobile Application Screen Shots

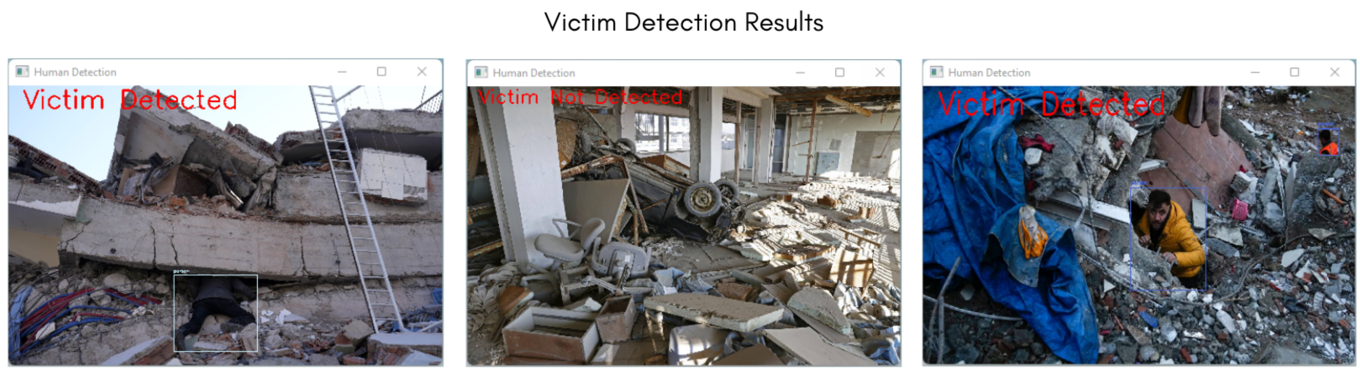

Fig. 8. Victim Detection Results

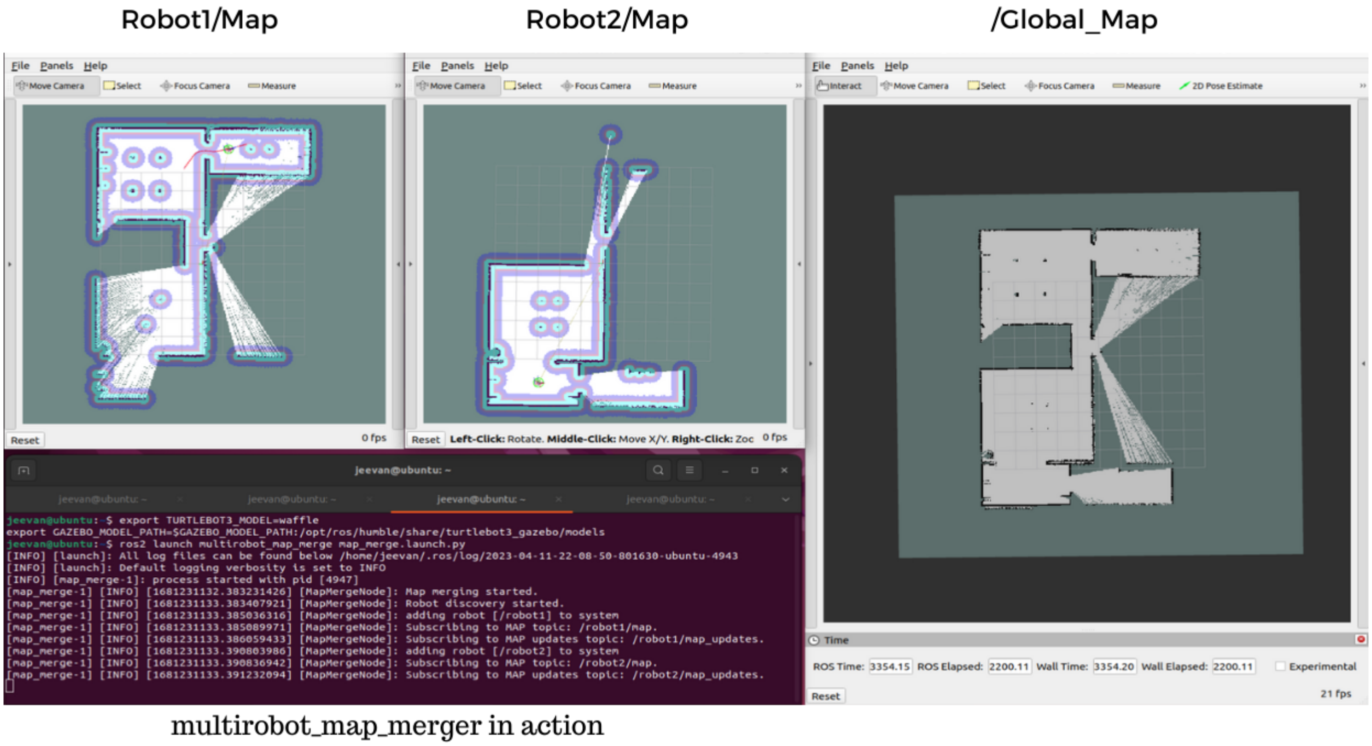

Fig. 9. Gazebo Simulation

was able to provide accurate and detailed maps of the surroundings, which could be used to locate and rescue victims.

Additionally assessed were the two-way audio communication and live video streaming modules. On the mobile app, the robots mere able to deliver a real-time video feed. It was discovered that the two-way audio communication module worked well, which can enable rescuers to speak with victims or other staff who were close to the robots. This capability proved to be very helpful when the robots were placed in hard-to-reach locations.

It was discovered that the local and global mapping modules worked well at producing precise and in-depth maps of the area. The global mapping module permitted the production of a more comprehensive map of the entire area, while the local mapping module allowed the robots to produce a detailed map of their immediate surroundings [84].

Also, the object detection module was especially helpful in identifying and finding victims are shown in Fig. 8. The module successfully found and identified the targeted object, providing precise information on the mobile app. This function proved to be especially helpful in circumstances where victims were either trapped or elusive.

The manual control module was evaluated by allowing users to take control of individual robots through the mobile app. [85] This feature proved to be particularly useful in situations where the robots needed to be controlled manually or where human intervention was necessary.

Overall, the results obtained from the implementation of the designed system were promising. The system was able to autonomously explore challenging environments, provide real-time video feed and two-way audio communication, create accurate and detailed maps of the surroundings, detect and locate victims, and provide real-time sensor readings. The system has the potential to be used in search and rescue operations to reduce the risk to human life and improve the efficiency of rescue operations.

VI. Conclusion and Future Scope

A. Conclusion

The proposed project demonstrated the successful implementation of a swarm of autonomous ground vehicles (AGVs) equipped with sensors and communication modules for search and rescue operations. The central control system, built on ROS architecture, effectively managed the horde of AGVs and allowed for real-time monitoring and control of the swarm. The integration of live video streaming and two-way audio communication enhanced the situational awareness of the rescue team and allowed for efficient communication with victims.

The autonomous exploration and mapping functionality enabled the AGVs to navigate challenging terrains, map the environment, and detect objects or victims. The manual control feature allowed for a human operator to take control of a specific robot when necessary. The live sensor readings provided real-time feedback on the status of each AGV, which was crucial for ensuring their proper functioning during search and rescue operations.

The project achieved the objective of reducing the risk to human life in dangerous areas by providing a quick and efficient response system for search and rescue operations. However, some limitations were observed during testing, such as the limited battery life of the AGVs and the dependency on stable network connectivity for communication.

In summary, our work offers a comprehensive solution that combines cost-effectiveness, practicality, scalability, and immense potential for various applications. With its ability to provide live video streaming, two-way audio communication, object detection, and global mapping, it holds promise for empowering sections of society prone to natural disasters such as earthquakes with increased efficiency in disaster response and providing humanitarian aid and relief efforts. Along with search and rescue operations, remote monitoring and environmental exploration in vast lands of an unknown environment and difficult terrain is also made comparatively easier.

B. Future Scope

Also, our architecture provides a solid foundation for future enhancements and advancements. The open-source nature of the software components allows for continuous development and integration of cutting-edge technologies. The proposed project can be further improved by implementing advanced machine-learning algorithms for object detection and tracking. The addition of unmanned aerial vehicles (UAVs) equipped with thermal imaging cameras can enhance search and rescue operations during nighttime or in low visibility conditions. The integration of 5G network technology can also improve the stability and speed of communication between the AGVs and the central control system.

The integration of advanced sensors and communication modules, along with autonomous exploration and mapping functionality, provided an efficient and reliable response system for search and rescue operations. The project has significant implications for improving the safety and effectiveness of search and rescue operations in challenging environments.

REFERENCES

[1] M. Schranz, M. Umlauft, M. Sende, and W. Elmenreich, "Swarm robotic behaviors and current applications," *Frontiers in Robotics and AI*, vol. 7, 2020, https://doi.org/10.3389/frobt.2020.00036.

[2] Marco Dorigo et al, "Reflections on the future of swarm robotics," *Science Robotics*, vol. 5, 2020, DOI: 10.1126/scirobotics.abe4385.

[3] W. Sun, M. Tang, L. Zhang, Z. Huo, and L. Shu, "A Survey of Using Swarm Intelligence Algorithms in IoT," *Sensors*, vol. 20, no. 5, p. 1420, 2020, doi:10.3390/s20051420.

[4] L. E. Parker, "On the design of behavior-based multi-robot teams," *Advanced Robotics*, vol. 10, no. 6, pp. 547-578, 1995, DOI: 10.1163/156855396X00228.

[5] L. H. Kim, D. S. Drew, V. Domova, and S. Follmer, "User-defined Swarm Robot Control". *In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (CHI '20). *Association for Computing Machinery*, pp. 1–13. 2020 https://doi.org/10.1145/3313831.3376814

[6] P. J. Narayana Raju, V. Gaurav Pampana, V. Pandalaneni, G. Gugapriya and C. Baskar, "Design and implementation of a rescue and surveillance robot using cross-platform application," *International Conference on Inventive Computation Technologies* (ICICT), pp. 644-648, 2022 doi: 10.1109/ICICT54344.2022.9850646.

[7] P. Skrzypczyński, "A team of mobile robots and monitoring sensors—from concept to experiment," *Advanced Robotics*, vol. 18, no. 6, pp. 583-610, 2004, DOI:10.1163/1568553041257413.

[8] G. A. Cardona and J. M. Calderon, "Robot Swarm Navigation and Victim Detection Using Rendezvous Consensus in Search and Rescue Operations," *Applied Sciences*, vol. 9, no. 8, 2019, doi: 10.3390/app9081702.

[9] J. Penders, L. Alboul, U. Witkowski, A. Naghsh, J. Saez-Pons, S. Herbrechtsmeier, and M. El-Habbal, "A Robot Swarm Assisting a Human Fire-Fighter," *Advanced Robotics*, vol. 25, no. 1-2, pp. 93-117, 2011, DOI: 10.1163/016918610X538507.

[10] P. G. F. Dias, M. C. Silva, G. P. Rocha Filho, P. A. Vargas, L. P. Cota, and G. Pessin, "Swarm Robotics: A Perspective on the Latest Reviewed Concepts and Applications," *Sensors*, vol. 21, no. 6, 2021, doi: 10.3390/s21062062.

[11] M. Dorigo, G. Theraulaz and V. Trianni, "Swarm Robotics: Past, Present, and Future [Point of View]," in Proceedings of the IEEE, vol. 109, no. 7, pp. 1152-1165, 2021, doi: 10.1109/JPROC.2021.3072740.

[12] M. Brambilla, et al, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, pp. 1–41, 2013, https://doi.org/10.1007/s11721-012-0075-2.

[13] N. Bredeche, and N. Fontbonne, "Social learning in swarm robotics," Philos. Trans. of The Royal Society B., vol. 377, no. 1843, doi: 10.1098/rstb.2020.0309.

[14] D. C. Zapata, E. Milner, J. Hird, G. Tzoumas, P. J. Vardanega, M. Sooriyabandara, M. Giuliani, A. F. T. Winfieild, and S. Hauert, "Mutual shaping in swarm robotics: User studies in fire and rescue, storage organization, and bridge inspection," *Frontiers in Robotics and AI*, vol. 7, 2020, https://doi.org/10.3389/frobt.2020.00053

[15] S. Na, et al, "Bio-inspired artificial pheromone system for swarm robotics applications," *Adaptive Behavior*, vol. 29, no. 4, pp. 395-415, 2021, doi: 10.1177/1059712320918936.

[16] R. Gross, M. Bonani, F. Mondada and M. Dorigo, "Autonomous Self-Assembly in Swarm-Bots," in *IEEE Transactions on Robotics,* vol. 22, no. 6, pp. 1115-1130, 2006, doi: 10.1109/TRO.2006.882919.

[17] M. Coppola, et al, "Provable self-organizing pattern formation by a swarm of robots with limited knowledge," *Swarm Intell*, pp. 59–94, 2019, https://doi.org/10.1007/s11721-019-00163-0

[18] Y. Liu, and K. Lee, "Probabilistic consensus decision making algorithm for artificial swarm of primitive robots," *SN Appl. Sci.*, 2019, https://doi.org/10.1007/s42452-019-1845-x

[19] C. Coquet, A. Arnold, and P.-J. Bouvet, "Control of a Robotic Swarm Formation to Track a Dynamic Target with Communication Constraints: Analysis and Simulation," *Applied Sciences,* vol. 11, no. 7, p. 3179, Apr. 2021, doi: 10.3390/app11073179.

[20] Z. Qiao, J. Zhang, X. Qu and J. Xiong, "Dynamic Self-Organizing Leader-Follower Control in a Swarm Mobile Robots System Under Limited Communication," in *IEEE Access,* vol. 8, pp. 53850-53856, 2020, doi: 10.1109/ACCESS.2020.2980778.

[21] D. Shah and L. Vachhani, "Swarm Aggregation Without Communication and Global Positioning," in *IEEE Robotics and Automation Letters,* vol. 4, no. 2, pp. 886-893, April 2019, doi: 10.1109/LRA.2019.2893413.

[22] M. Oliveira, et al, "Uncovering the social interaction network in swarm intelligence algorithms," *Appl Netw Sci*, 2020 https://doi.org/10.1007/s41109-020-00260-8

[23] T. Blackwell and J. Kennedy, "Impact of Communication Topology in Particle Swarm Optimization," in *IEEE Transactions on Evolutionary Computation,* vol. 23, no. 4, pp. 689-702, Aug. 2019, doi: 10.1109/TEVC.2018.2880894.

[24] J. Wilson, and S. Hauert, "Information transport in communication limited swarms," *Artif Life Robotics*, 2022, https://doi.org/10.1007/s10015-022-00768-5

[25] A. Sarin and A. T. Avestruz, "Code Division Multiple Access Wireless Power Transfer for Energy Sharing in Heterogenous Robot Swarms," in *IEEE Access,* vol. 8, pp. 132121-132133, 2020, doi: 10.1109/ACCESS.2020.3010202.

[26] J. Guan, W. Zhou, S. Kang, Y. Sun and Z. Liu, "Robot Formation Control Based on Internet of Things Technology Platform," in *IEEE Access,* vol. 8, pp. 96767-96776, 2020, doi: 10.1109/ACCESS.2020.2992701.

[27] X. Wang et al, "Machine Learning Empowered Spectrum Sharing in Intelligent Unmanned Swarm Communication Systems: Challenges, Requirements and Solutions," *in IEEE Access*, vol. 8, pp. 89839-89849, 2020, doi: 10.1109/ACCESS.2020.2994198.

[28] Y. Baldemir , S. İyigün , O. Musayev and C. Ulu , "Design and Development of a Mobile Robot for Search and Rescue Operations in Debris," *International Journal of Applied Mathematics Electronics and Computers*, vol. 8, no. 4, pp. 133-137, 2020, doi:10.18100/ijamec.800840.

[29] S. Habibian, et al, "Design and implementation of a maxi-sized mobile robot (Karo) for rescue missions," *Robomech Journal*, 2021, https://doi.org/10.1186/s40648-020-00188-9.

[30] R. P. Saputra et al, "ResQbot 2.0: An Improved Design of a Mobile Rescue Robot with an Inflatable Neck Securing Device for Safe Casualty Extraction," *Applied Sciences*, vol. 11, no. 12, p. 5414, 2021, doi: 10.3390/app11125414.

[31] M. Kocaoğlu , Y. E. Işıkdemir , M. A. Uzun , K. Turgut , M. O. Tas and B. Kaleci, "A Mobile Robot Application for Constructing Semantic and Metric Maps of Search and Rescue Arenas with Point-Based Deep Learning", *Journal of Science, Technology and Engineering Research*, vol. 2, no. 1, pp. 11-22, 2021, doi: 10.5281/zenodo.4589489.

[32] H. Fan, V. Hernandez Bennetts, E. Schaffernicht, and A. Lilienthal, "Towards Gas Discrimination and Mapping in Emergency Response Scenarios Using a Mobile Robot with an Electronic Nose," *Sensors*, vol. 19, no. 3, p. 685, 2019, doi: 10.3390/s19030685.

[33] H. Wang, C. Zhang, Y. Song, B. Pang, and G. Zhang, "Three-Dimensional Reconstruction Based on Visual SLAM of Mobile Robot in Search and Rescue Disaster Scenarios," *Robotica*, vol. 38, no. 2, pp. 350–373, 2020. doi: 10.1017/S0263574719000675.

[34] T. Kamegawa, T. Akiyama, S. Sakai, K. Fujii, K. Une, E. Ou, Y. Matsumura, T. Kishutani, E. Nose, Y. Yoshizaki, and A. Gofuku, "Development of a separable search-and-rescue robot composed of a mobile robot and a snake robot," *Advanced Robotics*, vol. 34, no. 2, pp. 132-139, 2020, DOI: 10.1080/01691864.2019.1691941.

[35] D. Chatziparaschis, M. G. Lagoudakis, and P. Partsinevelos, "Aerial and Ground Robot Collaboration for Autonomous Mapping in Search and Rescue Missions," *Drones*, vol. 4, no. 4, p. 79, 2020, doi: 10.3390/drones4040079.

[36] Q. Tang, F. Yu, Z. Xu and P. Eberhard, "Swarm Robots Search for Multiple Targets," *in IEEE Access*, vol. 8, pp. 92814-92826, 2020, doi: 10.1109/ACCESS.2020.2994151.

[37] M. B. Alatise and G. P. Hancke, "A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods," *in IEEE Access*, vol. 8, pp. 39830-39846, 2020, doi: 10.1109/ACCESS.2020.2975643.

[38] Z. Kashino, G. Nejat and B. Benhabib, "A Hybrid Strategy for Target Search Using Static and Mobile Sensors," *in IEEE Transactions on Cybernetics*, vol. 50, no. 2, pp. 856-868, 2020, doi: 10.1109/TCYB.2018.2875625.

[39] Y. Zhang, T. Liu, H. Zhang and Y. Liu, "LEACH-R: LEACH Relay With Cache Strategy for Mobile Robot Swarms," *in IEEE Wireless Communications Letters*, vol. 10, no. 2, pp. 406-410, 2021, doi: 10.1109/LWC.2020.3033039.

[40] R. Semenas and R. Bausys, "Modelling of Autonomous Search and Rescue Missions by Interval-Valued Neutrosophic WASPAS Framework," *Symmetry*, vol. 12, no. 1, p. 162, 2020, doi: 10.3390/sym12010162.

[41] B. A. D. M. Menezes, H. Kuchen, and F. B. D. L. Neto, "Parallelization of Swarm Intelligence Algorithms: Literature Review," *International Journal of Parallel Programming*, vol. 50, pp. 486–514, 2022, https://doi.org/10.1007/s10766-022-00736-3.

[42] T. Saba, et al, "Cloud-edge load balancing distributed protocol for IoE services using swarm intelligence," *Cluster Comput*, vol. 26, pp. 2921-2931, 2023. https://doi.org/10.1007/s10586-022-03916-5.

[43] M. Starks, A. Gupta, S. S. O V and R. Parasuraman, "HeRoSwarm: Fully-Capable Miniature Swarm Robot Hardware Design With Open-Source ROS Support," *2023 IEEE/SICE International Symposium on System Integration* (SII), 2023, pp. 1-7, doi: 10.1109/SII55687.2023.10039174.

[44] P.K. Das, and P.K. Jena, "Multi-robot path planning using improved particle swarm optimization algorithm through novel evolutionary operators," *Applied Soft Computing*, vol. 92, ISSN 1568-4946, 2020, https://doi.org/10.1016/j.asoc.2020.106312.

[45] L. Chen, et al, "Obstacle Avoidance and Multitarget Tracking of a Super Redundant Modular Manipulator Based on Bezier Curve and Particle Swarm Optimization," *Chinese Journal of Mechanical Engineering*, 2020, https://doi.org/10.1186/s10033-020-00491-x.

[46] E.Villemure, et al, "SwarmUS: An open hardware and software on-board platform for swarm robotics development," 2022, https://doi.org/10.48550/arXiv.2203.02643.

[47] Y. Wu, M. Li, G. Li and Y. Savaria, "Persistence Region Monitor With a Pheromone-Inspired Robot Swarm Sensor Network," *in IEEE Internet of Things Journal*, vol. 9, no. 14, pp. 12093-12110, 2022, doi: 10.1109/JIOT.2021.3133501.

[48] M. Chandarana, et al, "Planning and Monitoring Multi-Job Type Swarm Search and Service Missions," *Journal of Intelligent & Robotic Systems*, 2021, https://doi.org/10.1007/s10846-020-01272-3.

[49] J. K. Verma, and V. Ranga, "Multi-Robot Coordination Analysis, Taxonomy, Challenges and Future Scope," *Journal of Intelligent & Robotic Systems*, 2021, https://doi.org/10.1007/s10846-021-01378-2.

[50] K. Harikumar, J. Senthilnath and S. Sundaram, "Mission Aware Motion Planning (MAP) Framework With Physical and Geographical Constraints for a Swarm of Mobile Stations," in *IEEE Transactions on Cybernetics,* vol. 50, no. 3, pp. 1209-1219, 2020, doi: 10.1109/TCYB.2019.2897027.

[51] O. Saha, P. Dasgupta, and B. Woosley, "Real-time robot path planning from simple to complex obstacle patterns via transfer learning of options," *Auton Robot* vol. 43, pp. 2071–2093, 2019, https://doi.org/10.1007/s10514-019-09852-5

[52] Y. Huang, and M. Jafari, "Risk-aware Vehicle Motion Planning Using Bayesian LSTM-Based Model Predictive Control," 2023, https://doi.org/10.48550/arXiv.2301.06201.

[53] F. H. Ajeil, I. K. Ibraheem, A. J. Humaidi, and Z. H. Khan, "A novel path planning algorithm for mobile robot in dynamic environments using modified bat swarm optimization," *The Journal of Engineering*, pp. 37–48, 2021, https://doi.org/10.1049/tje2.12009.

[54] R. Almadhoun, et al, "A survey on multi-robot coverage path planning for model reconstruction and mapping," *SN Applied Sciences,* 2019, https://doi.org/10.1007/s42452-019-0872-y.

[55] N. Toufan, and A. Niknafs, "Robot path planning based on laser range finder and novel objective functions in grey wolf optimizer," *SN Applied Sciences,* 2020, https://doi.org/10.1007/s42452-020-3093-5.

[56] M. Vaidis, and M. J. D. Otis, "Toward a robot swarm protecting a group of migrants," *Intelligent Service Robotics,* 2020, https://doi.org/10.1007/s11370-020-00315-w.

[57] J. R. C. Leon, et al, "Raspberry pi and arduino uno working together as a basic meteorological station," 2017, https://doi.org/10.5121/ijcsit.2017.9508.

[58] P. J. Narayana Raju, V. G. Pampana, V. Pandalaneni, G. G and C. Baskar, "Multipurpose Adaptable Robot," 2022 8th *International Conference on Advanced Computing and Communication Systems* (ICACCS), 2022, pp. 1074-1077, doi: 10.1109/ICACCS54159.2022.9785181.

[59] D.Singh, and A. Nandgaonkar, "IOT-Based Wi-Fi Surveillance Robot with Real-Time Audio and Video Streaming," *Computing, Communication and Signal Processing*, vol. 810, 2019, https://doi.org/10.1007/978-981-13-1513-8_65.

[60] Gallastegi, and Akaitz, "Web-based Real-Time Communication for Rescue Robots," 2014.

[61] https://webrtc.org/

[62] T.P. Fowdur, N. Ramkorun, and P. K. Chiniah, "Performance Analysis of WebRTC and SIP-based Audio and Video Communication Systems," *SN Computer Science*, 2020, https://doi.org/10.1007/s42979-020-00380-z.

[63] B. Yamauchi, "A frontier-based approach for autonomous exploration," *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation'*, 1997, pp. 146-151, doi: 10.1109/CIRA.1997.613851.

[64] W. G. Teame , Dr. Y. Yu, and W. Zhongmin, "Optimization of SLAM Gmapping based on Simulation," *International Journal of Engineering Research & Technology*, vol. 09, no. 04, 2020.

[65] http://wiki.ros.org/gmapping

[66] A. Salunke, C. Patil, R. Mude and R. D. Joshi, "Simultaneous Localization and Mapping (SLAM) in Swarm Robots for Map-Merging and Uniform Map Generation Using ROS," *2023 15th International Conference on Computer and Automation Engineering* (ICCAE), pp. 411-415, 2023, doi: 10.1109/ICCAE56788.2023.10111365.

[67] P. Y. Lajoie, and G. Beltrame, "Swarm-SLAM: Sparse Decentralized Collaborative Simultaneous Localization and Mapping Framework for Multi-Robot Systems," 2023, https://doi.org/10.48550/arXiv.2301.06230

[68] W. Y. G. Louie, and G. Nejat, "A victim identification methodology for rescue robots operating in cluttered USAR environments," *Advanced Robotics*, vol. 27, no.5, pp. 373-384, 2013, DOI: 10.1080/01691864.2013.763743.

[69] S. Jia, Y. Hada, and K. Takase, "Human-assistance robotic system based on distributed computing technology," *Advanced Robotics*, vol. 18, no. 5, pp. 515-532, DOI: 10.1163/156855304774195055.

[70] https://pypi.org/project/cvlib/

[71] Y. Chen, S. Huang and R. Fitch, "Active SLAM for Mobile Robots With Area Coverage and Obstacle Avoidance," *in IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 3, pp. 1182-1192, 2020, doi: 10.1109/TMECH.2019.2963439.

[72] http://wiki.ros.org/multirobot_map_merge

[73] Şen, H, and Can, F.C, "Remote Control of Swarm Mobile Robots," DEStech Transactions on Engineering and Technology Research, 2018, https://doi.org/10.12783/dtetr.

[74] S. Jia, Y. Hada, and K. Takase, "Human-assistance robotic system based on distributed computing technology," *Advanced Robotics*, vol. 18, no. 5, pp. 515-532, DOI: 10.1163/156855304774195055.

[75] K. Ito, and A. Gofuku, "Hybrid autonomous control for multi mobile robots," *Advanced Robotics*, vol. 18, no. 1, pp. 83-99, 2004 DOI: 10.1163/156855304322753317.

[76] A. Barciś, M. Barciś and C. Bettstetter, "Robots that Sync and Swarm: A Proof of Concept in ROS 2," *2019 International Symposium on Multi-Robot and Multi-Agent Systems* (MRS), pp. 98-104, 2019 doi: 10.1109/MRS.2019.8901095.

[77] E. Poberezkin, et al, "Development of a robust Wi-Fi/4G-based ROS communication platform for an assembly and repair mobile robot with reliable behavior under unstable network or connection failure," *Artificial Life and Robotics*, vol. 27, pp. 786–795, 2022, https://doi.org/10.1007/s10015-022-00792-5.

[78] C. Crick, G. Jay, S. Osentoski, B. Pitzer, and O. C. Jenkins, "Rosbridge: ROS for Non-ROS Users," *Robotics Research*, *Springer Tracts in Advanced Robotics*, vol. 100, 2017, https://doi.org/10.1007/978-3-319-29363-9_28.

[79] J. E. Hurtado, et al, "Decentralized Control for a Swarm of Vehicles Performing Source Localization," *Journal of Intelligent and Robotic Systems*, 2004, https://doi.org/10.1023/B:JINT.0000049161.62303.e5.

[80] D. Tardioli, R. Parasuraman, and P. Ögren, "Pound: A multi-master ROS node for reducing delay and jitter in wireless multi-robot networks," *Robotics and Autonomous Systems*, vol. 111, pp. 73-87, 2019, https://doi.org/10.1016/j.robot.2018.10.009.

[81] Kamilaris, Andreas, and Botteghi, Nicolò. "The Penetration of Internet of Things in Robotics: Towards a Web of Robotic Things," vol. 12, no. 6, pp. 491 – 512, 2020,10.3233/AIS-200582.

[82] M. Mukhandi, D. Portugal, S. Pereira and M. S. Couceiro, "A novel solution for securing robot communications based on the MQTT protocol and ROS," *2019 IEEE/SICE International Symposium on System Integration* (SII), pp. 608-613, 2019, doi: 10.1109/SII.2019.8700390.

[83] K. Takaya, T. Asai, V. Kroumov and F. Smarandache, "Simulation environment for mobile robots testing using ROS and Gazebo," *2016 20th International Conference on System Theory, Control and Computing* (ICSTCC), pp. 96-101, 2016 doi: 10.1109/ICSTCC.2016.7790647.

[84] R. K. Megalingam, C. R. Teja, S. Sreekanth, and A. Raj, "ROS based autonomous indoor navigation simulation using SLAM algorithm," *International Journal of Pure and Applied Mathematics*, vol. 118, no. 7, pp. 199-205, 2018.

[85] S. G. Tzafestas, "Mobile Robot Control and Navigation: A Global Overview," *Journal of Intelligent & Robotic Systems* vol. 91, pp. 35–58, 2018, https://doi.org/10.1007/s10846-018-0805-9.