Autonomous Search and Rescue Robot

Ms.Tintu George, Razi Ilyas M.K, Ashish Arun, P Aswanth Mohan, Pranav KC Associate Professor, EEE Department, Vimal Jyothi Engineering College, Kannur,India
UG Scholars, EEE Department, Vimal Jyothi Engineering College, Kannur, India

---

Abstract - In disaster-stricken environments, traditional search and rescue methods often encounter significant challenges due to adverse conditions such as smoke, fog, darkness, or complex terrains. Advancements in autonomous robotics and sensor technologies provide effective solutions to enhance the efficiency and reliability of rescue operations. This work focuses on the development of an autonomous search and rescue robot equipped with LiDAR based mapping and thermal imaging technology. The LiDAR system enables real-time 2D mapping of the environment, ensuring precise navigation and obstacle avoidance in dynamic and unstructured terrains. At the same time, the thermal imaging system detects heat signatures, allowing the identification and localization of humans or animals that may be obscured by environmental hazards.

The proposed robotic system is designed to operate in disaster scenarios where human intervention is difficult or hazardous. By utilizing LiDAR for navigation and thermal imaging for survivor detection, the robot can autonomously move through challenging terrains while effectively identifying potential survivors. The integration of advanced sensing technologies with intelligent navigation algorithms enhances adaptability and reliability in critical situations. This approach aims to improve the speed and accuracy of locating victims, ultimately contributing to more effective and life-saving search and rescue missions.

Key Words: Autonomous, ROS Neotic, LIDAR, Teleoperation, Ubuntu Mate, Ubuntu Server, Navigation

1.INTRODUCTION

In disaster-stricken environments, locating survivors quickly and efficiently is crucial for successful search and rescue operations. Traditional methods often face limitations due to hazardous conditions such as smoke, fog, debris, and low visibility, making it difficult for rescue teams to identify and reach victims in a timely manner. Advancements in autonomous robotics and sensor technologies offer innovative solutions to overcome these challenges, enhancing the speed, accuracy, and safety of rescue missions.

This project focuses on the development of an autonomous search and rescue robot equipped with a thermal camera and LiDAR technology. The thermal camera detects heat signatures, allowing the identification of survivors even in visually obstructed environments, such as collapsed buildings or areas with heavy smoke. Meanwhile, the LiDAR system provides real-time 2D mapping of the surroundings, enabling precise navigation and tele-op keyboard control, ensuring the robot can be manually guided through complex terrains when necessary.

By integrating these technologies, the proposed robotic system can effectively support rescue teams in disaster scenarios where human access is restricted or dangerous. The robot’s ability to detect heat signatures and navigate autonomously reduces the time required to locate survivors, increasing the chances of successful rescue operations. Additionally, the system minimizes risks to human responders by allowing them to assess the situation remotely before direct intervention.

This approach aims to improve the overall efficiency and reliability of search and rescue operations by leveraging advanced sensing and navigation capabilities. The development of such autonomous robotic systems represents a significant step toward enhancing disaster response efforts, ultimately contributing to saving lives in critical situations.

Specific objectives include:

- Design an autonomous robot for efficient navigation in search and rescue missions
- Integrate sensors and algorithms to enable real-time environment mapping and thermal data for effective search and rescue missions.

- Ensure the robot can communicate its findings and coordinate with human rescuers.

3. LITERATURE REVIEW

The concept for our autonomous search and rescue robot was inspired by the study on "LiDAR-Stereo Thermal Sensor Fusion for Indoor Disaster Environments." This research emphasizes integrating LiDAR for precise 3D mapping with stereo thermal sensors for detecting heat signatures, enabling effective navigation and survivor detection in disaster zones. Real-time data processing and advanced decision-making algorithms further enhance the robot’s autonomy and situational awareness.

Low-power MEMS(Micro-Electro-Mechanical Systems) mirrors for LiDAR applications highlighted their compact size and efficiency. However, MEMS mirrors are highly sensitive to environmental factors like temperature, humidity, and vibrations, making them unsuitable for our robot, which requires robust performance in unpredictable disaster conditions.

LiDAR-based 2D mapping is crucial for autonomous navigation, offering precise spatial data for obstacle detection and localization. Despite its strengths, challenges such as environmental influences and data processing complexities must be addressed to ensure reliable operation in dynamic environments.

A wheeled locomotion system was chosen for its energy efficiency, simplicity, and cost-effectiveness, making it ideal for search and rescue operations. This ensures stable, controlled movement while keeping manufacturing and maintenance feasible.

Various sensor technologies were reviewed, including cameras, ultrasonic sensors, radar, and LiDAR. While cameras offer high-resolution imagery, they are affected by lighting conditions. Ultrasonic sensors are cost-effective but have limited range, and radar lacks precision for detailed mapping. LiDAR emerged as the best choice for our project due to its accuracy, real-time mapping, and resilience to lighting conditions.

Sound-based human detection using microphone arrays can be useful in search and rescue but is unreliable in noisy environments and for non-vocal individuals. Instead, our system integrates LiDAR and thermal sensors, providing robust mapping and heat-based survivor detection, ensuring effective performance even in low-visibility conditions.

4. METHODOLOGY

The development of the thermal-based search and rescue robot follows a structured methodology to ensure efficient navigation and survivor detection in disaster-stricken environments. The methodology consists of several key phases, including system design, sensor integration, mapping and control mechanisms, and performance evaluation.

1. System Design and Hardware Development

The robot's hardware framework is designed to support autonomous and manual operations in complex terrains. It consists of a rugged mobile platform equipped with LiDAR, a thermal camera, and essential electronic components. The LiDAR system is used to generate a real-time 2D map of the surroundings, while the thermal camera detects heat signatures of survivors. The robot is also integrated with a Raspberry Pi, which processes sensor data and facilitates communication with the operator.

2. Sensor Integration and Data Processing

The thermal camera captures infrared radiation emitted by humans or animals, allowing the system to detect survivors in low-visibility conditions. LiDAR generates environmental data, which is processed to construct a real-time spatial representation of the terrain. The integration of these sensor inputs enables accurate localization of survivors while providing navigational support.

3. Navigation and Teleoperation Control

The robot is designed to operate in both autonomous and manual modes. In autonomous mode, it follows a predefined path while continuously scanning the surroundings for heat signatures. In teleoperation mode, the operator manually controls the robot using a keyboard interface to navigate challenging terrains. The teleop keyboard control provides flexibility in areas where autonomous navigation may not be reliable, ensuring that the robot can be precisely directed toward potential survivors.

4. Testing and Performance Evaluation

The final phase involves extensive testing in simulated disaster environments to evaluate the robot’s performance under different conditions. Metrics such as mapping accuracy, thermal detection efficiency, response time, and mobility in rough terrains are assessed. The collected data is analyzed to optimize sensor calibration, improve navigation algorithms, and refine the teleoperation control for better responsiveness.

This methodology ensures that the search and rescue robot is capable of effectively detecting and locating survivors while being adaptable to diverse and challenging environments. The combination of thermal imaging, LiDAR-based mapping, and teleoperation control enhances its usability in real-world disaster response scenarios.

5) SYSTEM ARCHITECTURE

1) BLOCK DIAGRAM

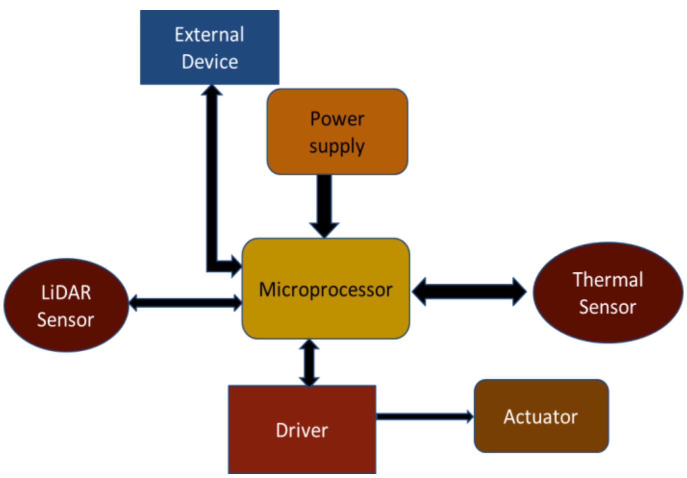

Fig 1. Block diagram of proposed system

The block diagram represents the architecture of a thermal-based search and rescue robot, showcasing the interaction between different components. Each block plays a crucial role in ensuring efficient operation, navigation, and detection of survivors.

1. Raspberry Pi (Microprocessor)

The Raspberry Pi serves as the central processing unit, handling data from the LiDAR sensor and the thermal sensor while communicating with other components such as the driver and external device.

2. RPLiDAR (LiDAR Sensor)

The RPLiDAR sensor captures environmental data by generating a 2D map of the surroundings. This helps the robot understand its environment and aids in navigation by identifying obstacles and terrain features. The data from the LiDAR is sent to the Raspberry Pi for further processing.

3. AGM8866 (Thermal Sensor)

The AGM8866 thermal sensor detects heat signatures emitted by humans or animals, allowing the robot to locate survivors in low-visibility conditions, such as smoke, fog, or darkness. The Raspberry Pi processes this data to determine potential rescue locations.

4. Power Supply

The power supply provides the necessary electrical energy to all components, ensuring the continuous operation of the system. It powers the Raspberry Pi, sensors, and actuators.

5. External Device

The external device serves as a communication interface for the operator. It can be a laptop, remote control, or a monitoring system used to receive live data, issue commands, or control the robot manually if needed.

6. Driver

The driver acts as an interface between the Raspberry Pi and the actuators. It converts control signals from the processor into appropriate commands that drive the actuators, enabling movement and interaction with the environment.

7. Actuator

The actuator executes physical movements based on the commands received from the driver. It could be motors controlling the robot's wheels or robotic arms used for object manipulation.

Overall Functionality

The Raspberry Pi collects and processes data from the RPLiDAR and AGM8866 thermal sensors, providing real-time navigation and survivor detection capabilities. The operator can control or monitor the system through an external device. The driver and actuators execute necessary actions to maneuver the robot based on sensor inputs. The power supply ensures that all components function efficiently.

This architecture enables the search and rescue robot to operate effectively in disaster-stricken environments, improving the chances of locating and assisting survivors.

2) CIRCUIT DESIGN

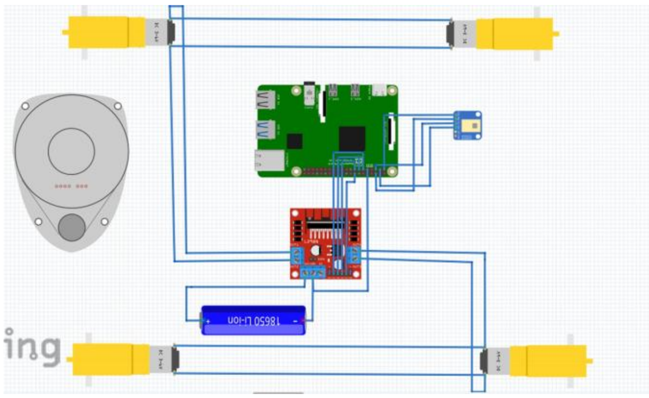

Fig 3. Circuit diagram of proposed system

The circuit diagram provides an exact electrical wiring representation of the project. It shows how each component is physically connected, including specific GPIO pin connections on the Raspberry Pi for the LiDAR (UART/USB), AMG8833 (I2C), Motor Driver (PWM & Direction Control), and Power Supply (Voltage Regulators, Battery, and Motors). It ensures proper operation.

3) SOFTWARE REQUIREMENT

1. Operating System & Development Environment

- Ubuntu 20.04 Server (Installed on Raspberry Pi 4)
- Ubuntu MATE Core (For GUI access and lightweight operations)

2. Robotics & SLAM Software

- ROS (Robot Operating System) Noetic (Middleware for integrating different components)
- Hector SLAM (For mapping using RPLIDAR)
- RPLIDAR SDK (Drivers for LiDAR integration)
- Navigation Stack (Partial usage)

3. Thermal Imaging Software

- AMG8833IR Python Library (To interface with the AMG8833 IR sensor)

5. Hardware Interface & Control

- RPi.GPIO (For motor and sensor control via Raspberry Pi)
- I2C (For sensor communication)
- USB (lidar communication)

6. Additional Tools & Libraries

- Vino & VNC (For remote access and debugging)

4) ALGORITHM

i) Subscriber script for 2D navigation goal and function Step 1:

Initialize the System

1. Start the ROS node (navigate_with_pwm).
2. Initialize GPIO pins for motor control.
3. Set up PWM signals for motor speed adjustment.
4. Subscribe to the /move_base_simple/goal topic to receive goal positions.

Step 2: Receive Goal Coordinates

1. When a new goal is received via PoseStamped, extract:
   - Goal X, Y coordinates from the message.
2. Retrieve the robot's current position and orientation using tf.TransformListener().
3. Calculate the difference in X and Y positions (delta_x, delta_y).
4. Compute the target angle using: θgoal=atan2(Δy,Δx)\theta_{goal} = \text{atan2}(\Delta y, \Delta x)θgoal=atan2(Δy,Δx)

Step 3: Determine Robot Orientation and Adjust Motion

1. Convert the robot's quaternion orientation to Euler angles.
2. Extract the current yaw (θ_robot).
3. Calculate the angle difference: θdiff=θgoal−θrobot\theta_{diff} = \theta_{goal} - \theta_{robot}θdiff=θgoal−θrobot
4. Normalize the angle difference to ensure it stays within [−π,+π]

Step 4: Move the Robot Towards the Goal

1. If the robot is not aligned with the goal direction:
   - Adjust the angular speed (angular_speed = 0.5 * θ_diff).
   - Rotate left if θ_diff > 0, otherwise rotate right.
2. Once the robot is aligned within a threshold (±π/6 rad):
   - Move forward with PWM-based speed control.
   - Set linear_speed = 0.5.
3. Maintain movement until the goal position is reached.

Step 5: Monitor Movement and Stop at Goal

1. Continue adjusting speed and direction while the goal is not reached.
2. If the robot overshoots or detects an obstacle (extendable with sensors), stop motion.
3. Once the robot is at the goal position, execute stop_motors().
4. Log the completion and wait for a new goal.

Step 6: Cleanup on Exit

1. If the program is terminated, clean up all GPIO resources using GPIO.cleanup().
2. Stop the ROS node execution.

ii) Thermal imaging python script

Step 1: Initialize Hardware & Libraries

1. Import required libraries:
   - time (for delays)
   - board & busio (for I2C communication)
   - adafruit_amg88xx (AMG8833 sensor library)
   - matplotlib.pyplot & numpy (for visualization)
   - matplotlib.colors (for color mapping)
   - scipy.ndimage.zoom (for interpolation)
2. Initialize the I2C bus (SCL, SDA) to communicate with AMG8833.
3. Create an AMG8833 object to read sensor data.

Step 2: Set Up Heatmap for Live Display

1. Enable plt.ion() for interactive live plotting.
2. Create a matplotlib figure (fig) and axes (ax).
3. Define a custom colormap (coolwarm) for temperature visualization:
   - Blue for cooler areas
   - Yellow for moderate temperatures
   - Red for temperatures above 30°C
4. Set up normalization (colors.Normalize()) between 0°C and 35°C.
5. Create an empty 8×8 heatmap to store sensor data.

Step 3: Capture and Process Sensor Data

1. Enter an infinite loop to continuously capture temperature data.
2. Read 8×8 pixel temperature values from AMG8833.
3. Convert data into a NumPy array for easier processing.
4. Interpolate the 8×8 data into a higher-resolution 32×32 grid using zoom().

Step 4: Update Heatmap in Real-Time

1. Update the heatmap with new temperature values.
2. Autoscale the color range dynamically to reflect temperature variations.
3. Redraw the heatmap (plt.draw()) and pause for 0.1 seconds to refresh the display.

Step 5: Exit Cleanly on User Interruption

1. If the user presses Ctrl+C, handle the KeyboardInterrupt gracefully.
2. Close the matplotlib window and stop execution.

iii) Subscriber script for teleopt Step 1:

Import Required Libraries

1. Import ROS (rospy) for handling ROS nodes and messages.
2. Import Twist from geometry_msgs.msg to receive velocity commands.
3. Import RPi.GPIO to control the motor pins on the Raspberry Pi.
4. Import Timer from threading to implement a timeout function for stopping the robot if no commands are received.

Step 2: Define GPIO Motor Pins and Initialization

1. Set GPIO mode to BCM.

2. Define motor GPIO pins for:
   - Left Motor Forward (LEFT_MOTOR_FORWARD)
   - Left Motor Backward (LEFT_MOTOR_BACKWARD)
   - Right Motor Forward (RIGHT_MOTOR_FORWARD)
   - Right Motor Backward (RIGHT_MOTOR_BACKWARD)
3. Set up GPIO pins as outputs.

Step 3: Define Motor Control Functions

1. Function to stop all motors (stop_motors()):
   - Set all GPIO motor pins to False (low state).
2. Function to control the robot's motion (control_motors(linear_x, angular_z)):
   - First, stop all motors to prevent unwanted motion.
   - Check linear velocity (linear_x):
     - Positive: Move forward (activate both forward pins).
     - Negative: Move backward (activate both backward pins).
   - Check angular velocity (angular_z):
     - Positive: Turn left (activate right motor forward only).
     - Negative: Turn right (activate left motor forward only).

Step 4: Implement a Timer for Automatic Stopping

1. Define a global stop_timer variable.
2. Function (reset_timer()) to restart the stop timer every time a new command is received:
   - If stop_timer exists, cancel the existing timer.
   - Create a new timer (Timer(0.5, stop_motors)) that stops the robot if no new commands arrive within 0.5 seconds.
   - Start the timer.

Step 5: Define ROS Callback Function

1. Function (move_robot(data)) to process incoming Twist messages:
   - Reset the stop timer when a new message is received.
   - Call control_motors(data.linear.x, data.angular.z) to execute movement.

Step 6: Initialize ROS Node and Start Listening

6. Initialize ROS node (teleop_listener).
7. Subscribe to /cmd_vel topic to receive velocity commands.
8. Register stop_motors() to run when the node shuts down.
9. Keep the node running indefinitely (rospy.spin()) to keep receiving messages.

Step 7: Handle ROS Shutdown and Cleanup GPIO

1. If a ROS interrupt exception (Ctrl+C) occurs:
   - Stop all motors.
   - Cleanup GPIO pins to prevent issues when restarting the script.

6) WORKING AND IMPLEMENTATION

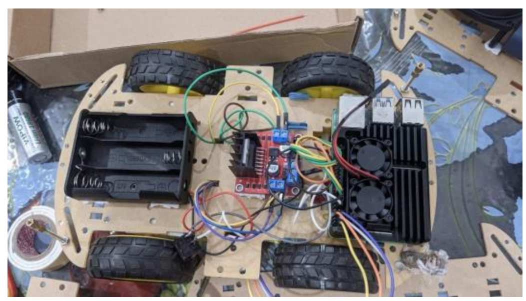

Fig 4. Circuit building

Fig 5. 3D Model

- The autonomous robot operates by using a mobile chassis equipped with a Raspberry Pi 4, RPLIDAR for SLAM-based mapping, and an AMG8833IR thermal sensor for heat detection.
  - It runs on Ubuntu MATE Core with ROS Noetic, utilizing Hector SLAM for real-time

2D mapping and localization.

- The robot receives a navigation goal from the user and uses the ROS navigation stack to navigate into that direction of the goal. Simultaneously, it collects thermal data, which is processed in real-time.
- Once the robot reaches its desired direction, it stops and the collected map data can be saved for further analysis.

7) RESULT AND VERIFICATION

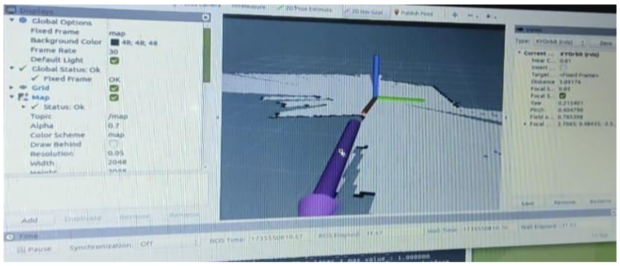

Fig 6. Defining target

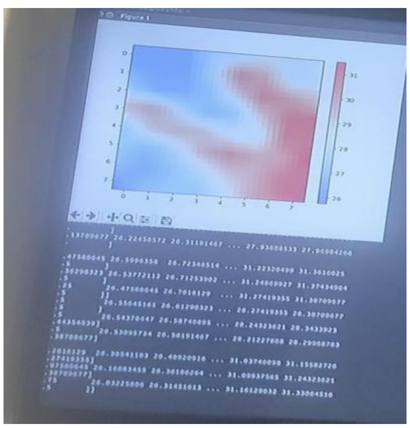

Fig 7.Thermal Data

Fig 8. Lidar Hector-slam mapping

The 4-wheeled autonomous robot was successfully developed, integrating LiDAR (RPLIDAR) for mapping and navigation, a Raspberry Pi for processing, and an AMG8833 thermal camera for thermal imaging. The robot accurately mapped its surroundings using Hector SLAM and navigated through nav goals effectively . The real-time thermal imaging system successfully detected heat sources and displayed corresponding thermal maps. Additionally, the motion control system, implemented using ROS and PWM motor drivers, provided smooth and precise movement. Remote monitoring and control were achieved via wireless communication, ensuring seamless interaction with the robot.

7) FUTURE SCOPE

The thermal-based search and rescue robot has significant potential for future enhancements to improve its effectiveness in disaster response. Advancements in autonomous navigation using AI-driven path planning can enable independent operation, while machine learning models can refine obstacle detection and terrain adaptation. Multi-sensor integration with ultrasonic sensors, RGB cameras, and gas detectors will enhance situational awareness, while combining thermal imaging with visual spectrum cameras can improve heat signature verification. Swarm robotics can expand large-scale search operations by coordinating multiple robots for efficient mapping and survivor detection. Enhancing long-range communication with 5G, LoRa, or satellite connectivity will extend operational range, allowing real-time data monitoring via cloud-based systems. Mobility can be improved by incorporating legged or hybrid robotic designs for better terrain adaptability, and advanced shock-absorbing mechanisms can enhance stability on rough surfaces. Lastly, optimizing energy efficiency through solar-assisted power and high-capacity batteries will extend operational duration, making the robot more sustainable for prolonged rescue missions.

8) CONCLUSION

The development of a thermal-based search and rescue robot, incorporating RP LiDAR for mapping and AGM8866 thermal imaging for heat signature detection, has proven its effectiveness in disaster response by assisting rescue teams in locating survivors. Powered by a Raspberry Pi, the system integrates advanced sensing and teleoperation control, enabling real-time mapping, navigation, and human detection in hazardous environments where direct human intervention is challenging. The ability to remotely operate the robot using a teleoperation keyboard enhances its adaptability in complex terrains. This project underscores the critical role of robotic automation in emergency response, demonstrating how intelligent sensing technologies can significantly improve the speed and accuracy of search and rescue operations. The findings of this work, along with its potential advancements, reinforce the growing importance of robotics in disaster management, ensuring safer, faster, and more effective rescue missions.

REFERENCES

[1] S. Rho, S. M. Park, J. Pyo, M. Lee, M. Jin and S. -C. Yu, "LiDAR-Stereo Thermal Sensor Fusion for Indoor Disaster Environment," in IEEE Sensors Journal, vol. 23, no. 7, pp. 7816-7827, 1 April 1, 2023, doi: 10.1109/JSEN.2023.3245619.

[2] D. Wang, L. Thomas, S. Koppal, Y. Ding and H. Xie, "A Low-Voltage, Low-Current, DigitalDriven MEMS Mirror for Low-Power LiDAR," in IEEE Sensors Letters, vol. 4, no. 8, pp. 1-4, Aug. 2020, Art no. 5000604, doi: 10.1109/LSENS.2020.3006813.

[3] M. J. Gallant and J. A. Marshall, "Two-Dimensional Axis Mapping Using LiDAR," in IEEE Transactions on Robotics, vol. 32, no. 1, pp. 150-160, Feb. 2016, doi: 10.1109/TRO.2015.2506162.

[4] 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.

[5] L. Reddy Cenkeramaddi, J. Bhatia, A. Jha, S. Kumar Vishkarma and J. Soumya, "A Survey on Sensors for Autonomous Systems," 2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA), Kristiansand, Norway, 2020, pp. 1182-1187, doi: 10.1109/ICIEA48937.2020.9248282.

[6] B. Zhang, K. Masahide and H. Lim, "Sound Source Localization and Interaction based Human Searching Robot under Disaster Environment," 2019 SICE International Symposium on Control Systems (SICE ISCS), Kumamoto, Japan, 2019, pp. 16-20, doi: 10.23919/SICEISCS.2019.8758766.

[7] X. Chen, H. Zhang, H. Lu, J. Xiao, Q. Qiu and Y. Li, "Robust SLAM system based on monocular vision and LiDAR for robotic urban search and rescue," 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), Shanghai China, 2017, pp. 41-47, doi: 10.1109/SSRR.2017.8088138. 20

[8] M. Y. Chuttur and P. Rungen, "Design and Implementation of an Autonomous Wheeled Robot Using IoT with Human Recognition Capability," 2022 3rd International Conference on Computation, Automation and Knowledge Management (ICCAKM), Dubai, United Arab Emirates, 2022, pp. 1-5, doi: 10.1109/ICCAKM54721.2022.9990183.