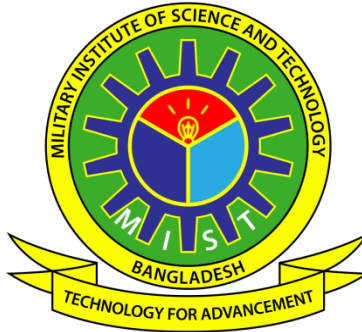


PROJECT REPORT



MILITARY INSTITUTE OF SCIENCE AND TECHNOLOGY (MIST) COMPUTER SCIENCE AND ENGINEERING (CSE)

COURSE CODE

CSE-405(Computer Interfacing Sessional)

GROUP A6

TITLE

AUTONOMOUS SURVEILLANCE ROBOT

SUBMITTED BY

Lt Sk Saier Islam	202014073
Lt Akkhoy Areng Joy	202014074
Lt Khalid Md. Arafatzaman Miraj	202014076
Jawad Rahman	202114049

Introduction

In today's rapidly evolving technological landscape, security concerns in public and private spaces are becoming increasingly critical. Autonomous surveillance systems present a promising solution, leveraging robotics and artificial intelligence to enhance safety and situational awareness. This project aims to develop an advanced security solution in the form of an Autonomous Surveillance Robot, designed to autonomously navigate, monitor, and respond to environmental stimuli.

The robot incorporates indoor mapping, real-time object detection, environmental monitoring, and remote-control functionalities. Utilizing robotics and computer vision technologies, it can autonomously navigate complex indoor environments, detect and track objects in real-time, and provide continuous situational updates. The system's versatility is enhanced through its integration of various sensors, including temperature, gas, smoke, and sound detectors. These sensors allow the robot to identify hazardous conditions, such as fires, gas leaks, and unusual noises, alerting users to potential threats.

The robot's functionality is extended by a web-based dashboard, offering users the flexibility to remotely monitor and control the system from any location with internet access. This feature not only increases usability but also ensures that immediate actions can be taken in response to real-time events. The project underscores the practical application of cutting-edge robotics and computer vision technologies, advancing the development of autonomous systems that contribute to public safety and security.

By combining real-time data collection, autonomous navigation, and remote access, this project demonstrates the potential of autonomous systems in various domains. It sets the stage for further innovations in security technology, showcasing how robotics can be deployed to address complex safety challenges efficiently and effectively.

Background Study

This paper [1] aims to develop a smart surveillance bot for highly restricted outdoor areas. The bot is designed with features including automatic surveillance and obstacle detection, human detection using an IR thermal camera, identification of friend or foe using RFID tags, and live video surveillance with manual remote control. The methodology involves using a PixyCam for line-following and ultrasonic sensors for obstacle detection, IR-based thermal cameras for human detection, and UHF RFID tags for identification. The bot uses Wi-Fi for data transfer and is built with a Raspberry Pi and other hardware components. In terms of accuracy, the bot has an 87% success rate in obstacle detection, a 91% success rate in human detection, and 100% success in RFID identification up to 12 meters, with reduced efficiency at 13-15 meters. Limitations include reduced RFID efficiency at longer distances and potential environmental interferences with

sensors. Overall, the bot is effective for surveillance and security but may face challenges in extreme conditions.

This paper [2] aims to develop and evaluate a system for remote control and autonomous operation of a wheeled mobile robot (WMR) for indoor patrol and surveillance. This system integrates smartphone control, Bluetooth, Wi-Fi wireless communication, image processing for object tracking, and infrared sensors with fuzzy control for wall-following and obstacle-avoidance. The methodology involves three main components: smartphone remote control where the WMR is manually operated via Zigbee wireless communication; object tracking using the EYECAM camera with image processing techniques to convert RGB images to HSV color space and then to binary images for object identification, followed by distance calculation and fuzzy control for dynamic tracking; and the integration of infrared sensors and fuzzy systems for autonomous wall-following and obstacle avoidance. The accuracy and output of the system are demonstrated through successful dynamic tracking of moving objects, precise indoor patrol with consistent wall-following and obstacle avoidance, and effective remote surveillance with real-time image transmission to remote devices. However, the system has limitations including the Boe-Bot Idrobot's limited memory capacity (2K Bytes), potential communication delays or interference due to the use of Zigbee and 24 GHz camera frequency, reduced performance in highly dynamic environments with numerous obstacles and varying light conditions, and potentially insufficient processing speed for real-time operation in all scenarios.

This paper [3] aims to design and implement an autonomous dam surveillance robot system that ensures the long-term safety and operational stability of dams by replacing human surveillance. The methodology involves developing a robust skid-steered mobile robot equipped with various sensors, including a GNSS receiver, stereo camera, LiDAR, monocular camera, and an onboard IMU for accurate data collection. A two-node Extended Kalman Filter (EKF) structure fuses data from these sensors for robust 3D localization, while the Dynamic Window Approach (DWA) algorithm is used for local motion planning. The control system includes a customized MCU board and a PID controller for precise motor control, and a YOLO v3-based recognition network is deployed for real-time detection of cracks and pedestrians. The system demonstrates high accuracy and robust performance in localization and navigation, with an origin RMSE of 0.3135 over a 70-meter path and effective real-time detection capabilities. However, the system's performance is heavily dependent on the availability and accuracy of multiple sensors, and it may face challenges in extremely rugged or inaccessible areas. Additionally, real-time data processing and sensor fusion can introduce latency, affecting the robot's responsiveness in dynamic

environments, necessitating improvements in computational efficiency and real-time processing capabilities.

This paper [4] aims to develop an advanced indoor security system that enhances surveillance coverage by using a mobile patrolling vehicle, providing real-time remote monitoring and control capabilities, and implementing face detection technology to identify potential intruders. The methodology includes designing the system architecture with components like wireless IPCAMs, RFID position detection, and remote monitoring interfaces; implementing the hardware with a DFRduino RoMeo 328 microcontroller, RFID tags, and WiFi modules; developing software using OpenCV for face detection; and conducting experiments to evaluate the system's performance in patrolling routes, detecting faces, and sending alerts. The accuracy and outcomes of the system are validated through experimental results, demonstrating effective face detection within specific range and angle limitations, accurate RFID-based position tracking, and reliable remote monitoring and alert capabilities, ensuring continuous operation with automatic recharging. However, the system has limitations, including constraints in face detection range and angles, dependency on stable network and RFID systems, environmental constraints, limited battery life, indoor applicability, complexity and cost of implementation, and reliance on user response to alerts, highlighting areas for further improvement.

This paper [5] aims to design and implement a real-time system for human detection, tracking, security, and verification in complex environments using a Kinect RGB-D camera mounted on a wireless-controlled mobile robot integrated with the Robot Operating System (ROS). The methodology involves capturing visual data through the Kinect, processing it on a laptop to normalize point cloud data, forming histograms, applying PCA for dimensionality reduction, and classifying using SVM. Human tracking is achieved through the nearest neighbor search algorithm, and wireless control is facilitated via HTTP protocol, enabling remote operation through a cell phone. Experimental results demonstrated the system's accuracy in detecting and tracking humans in real-time scenarios, with successful single and multiple human interactions observed. However, limitations include increased complexity in crowded environments, dependency on a laptop for processing, limited detection scope, and potential issues with occlusions. Future improvements suggest implementing the system on ARM-based embedded platforms like Raspberry Pi for better portability and communication and incorporating more RGB-D cameras for 360-degree video capturing.

Objective

- I. To develop an autonomous surveillance robot capable of mapping indoor environments and navigating autonomously using SLAM techniques.
- II. To implement real-time motion detection.
- III. To integrate temperature, humidity, LPG gas, air quality, CO, and smoke sensors to enable environmental monitoring and early detection of hazardous conditions.
- IV. To incorporate a microphone for detecting and recording suspicious sounds, enhancing situational awareness.
- V. To design and develop a user-friendly dashboard for remote monitoring and control of the robot, allowing seamless integration with existing security systems.
- VI. Fire detection in real-time

Methodology

a. Components Description

Hardware Components

1. **RPLIDAR A2:** This component is responsible for precise distance measurement and environmental mapping. It plays a crucial role in enabling the robot to understand its surroundings for autonomous navigation. It uses laser and collects all the object around it.
2. **12V Battery:** The main power source for the robot, providing energy to all its components.
3. **Jetson Xavier NX:** The primary processing unit used for handling the overall computations, including data from sensors and running algorithms for navigation and detection.
4. **Gas Sensor:** Detects hazardous gases in the environment, allowing the robot to monitor air quality and issue warnings if harmful gases are present. We have used MQ series MQ-2, MQ-6, MQ-9, MQ-135.
5. **Ultrasonic Radar Sensor:** This sensor is used for motion detection, helping the robot avoid obstacles and navigate its environment safely. Instead of sniffing the blackbody radiation from a moving person, this sensor uses a “microwave Doppler radar” technique to detect moving objects.
6. **Pololu Motor Driver:** Manages the power supplied to the motors, enabling the robot to move by driving the wheels.
7. **Web Camera:** Provides visual data for the robot by capturing images or video of the indoor environment, supporting object detection and real-time monitoring.
8. **Arduino Mega:** Functions as a microcontroller that takes input from various sensors and processes the data for communication with the main processing unit.
9. **Teensy 4.1 :** It is a microcontroller of higher clock speed around 600 Mhz reliable for motor controlling.

10. **Voltage Meter:** Measures the voltage levels in the robot's power system to ensure all components are receiving adequate power.
11. **DHT Sensor:** Measures indoor temperature and humidity, providing essential data for environmental monitoring and control.

Software Components

1. **Robot Operating System (ROS):** The communication framework that connects all hardware and software components. It facilitates sensor data processing, control algorithms, and communication between various modules.
2. **Ubuntu 20.04:** The main operating system that runs on the Jetson Xavier. It provides a stable and flexible environment for executing the robot's software stack.
3. **Flutter:** A development framework used to create the mobile application or web-based interface for remote monitoring and control of the robot.
4. **OpenCV:** An image processing library used for video capturing and analyzing visual data, enabling object detection and recognition in real-time.
5. **Socket.IO:** A server used for real-time communication between the robot and its remote interface, ensuring data is transmitted instantly for effective monitoring.
6. **YOLO V8:** An object detection model that helps identify and classify objects within the robot's visual range, enhancing its ability to detect suspicious activities or threats(local).

b. System Architecture

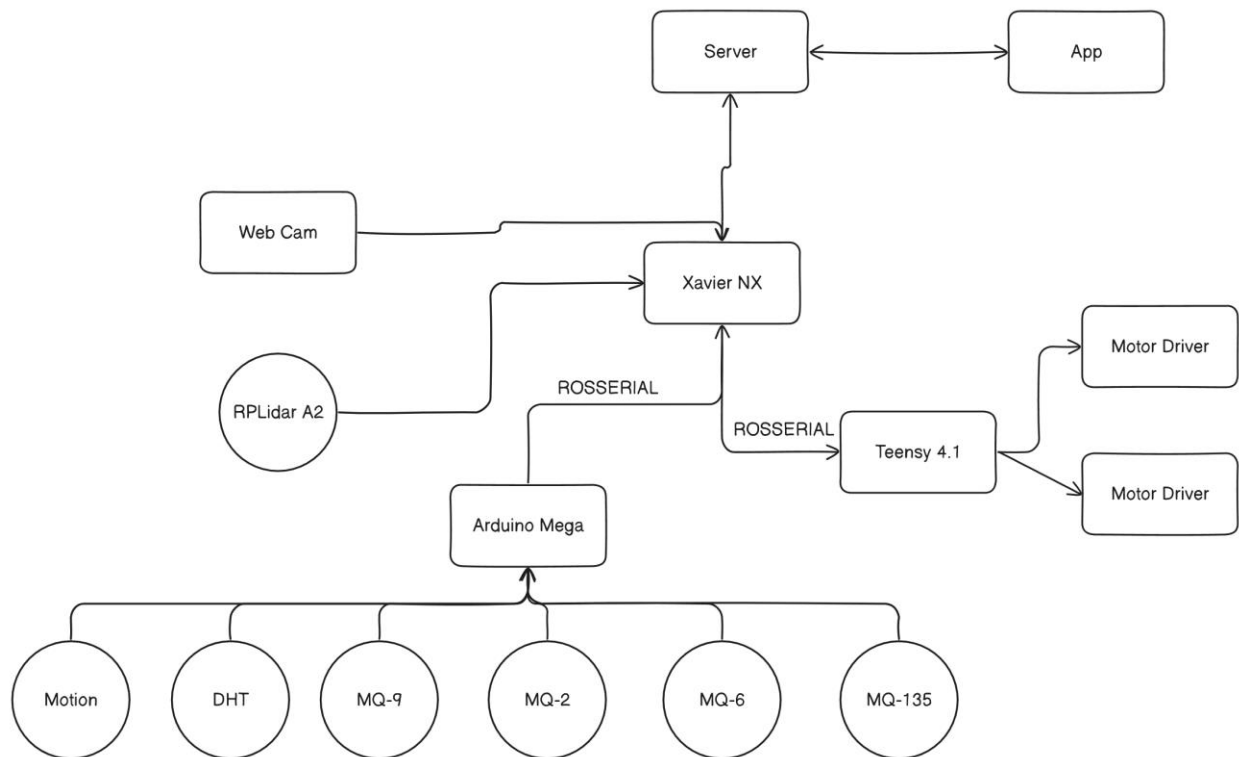


Fig1: System Architecture

The system architecture figure likely illustrates the overall structure and components of the system you're developing. It depicts how various subsystems, such as the frontend (user interface), backend (servers) and hardware interact with one another.

For example, in a mobile application setup, this architecture could show how the user interacts with the app, which sends requests to a backend server which is deployed in Xavier NX. The server handles the inter node communication and seamless integration among all the scripts. The architecture diagram highlights the flow of data and control across these components, including communication protocols frameworks used (Flutter for frontend), and the core logic that ties everything together. The system architecture diagram also usually includes the integration of any machine learning models, such as YOLO V8, and how these models are embedded in the app to perform predictions or analysis on-device. This figure outlines the overall structure of the Autonomous Surveillance Robot, showing how the different hardware and software components are connected. It highlights the relationship between the sensors, processing units (e.g., Jetson Xavier, Arduino, Teensy), power systems (battery), and communication framework (ROS). The architecture diagram provides a high-level view of how data flows through the system and how each module interacts to achieve the robot's goals of navigation, detection, and remote monitoring.

c. Workflow Diagram

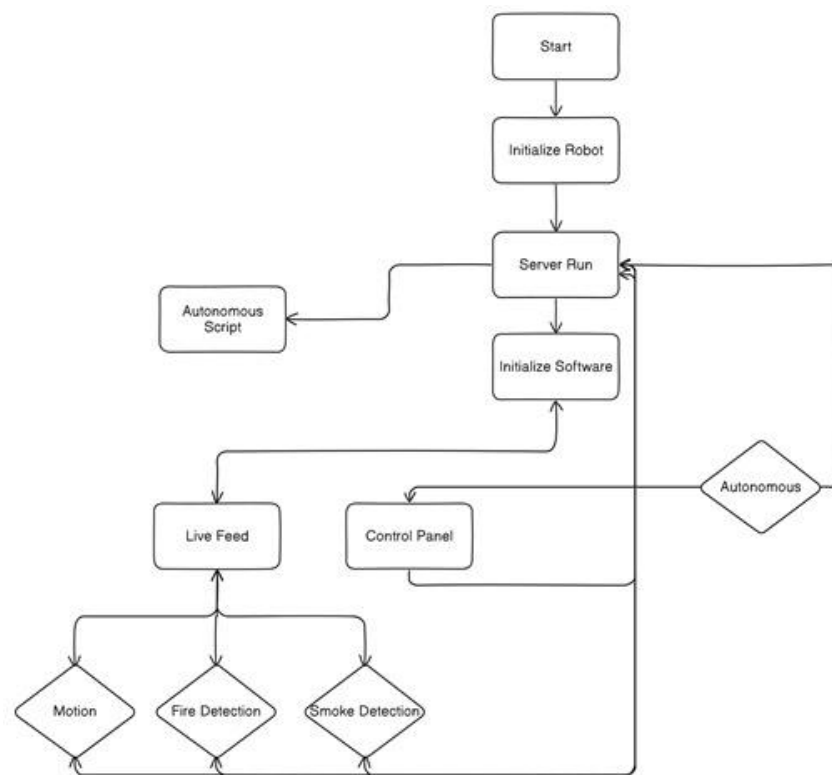


Fig2: Workflow

The workflow diagram visualizes the sequence of actions and interactions in your system, focusing on how tasks progress from start to finish. This diagram typically captures the user journey, illustrating the step-by-step flow of activities the system performs in response to user inputs or internal processes.

Breakdown:

1. **Start:**
 - The process begins with the initiation of the robot system.
2. **Initialize Robot:**
 - Once the system starts, the robot is initialized. This likely involves booting the hardware and preparing the necessary components for operation.
3. **Server Run:**
 - After the robot is initialized, a server is initiated. This server probably manages the robot's data, operations, and communications with other systems.
4. **Initialize Software:**
 - Once the server is running, the software required for the robot's tasks is initialized. This software may include drivers, algorithms, and control systems for autonomous functioning.
5. **Autonomous Decision:**
 - The system checks whether the robot should operate autonomously. This is indicated by a decision node labeled "Autonomous," which directs the system flow:
 - **If autonomous:** The system activates an autonomous script.
 - **If not autonomous:** The system operates under manual control or supervision.
6. **Autonomous Script:**
 - In the case of autonomous operation, the robot runs an autonomous script, allowing it to perform tasks without human intervention. This script likely dictates movement, environment scanning, and detection algorithms.
7. **Live Feed:**
 - The system captures a live feed from the robot's sensors and cameras, feeding into the detection processes for various scenarios.
 - **Motion Detection:** The robot checks for movement in its environment.
 - **Fire Detection:** It analyzes the environment for signs of fire.
 - **Smoke Detection:** It scans for any smoke present, indicating potential fire hazards.
8. **Control Panel:**
 - Alternatively, the system can be managed via a control panel, enabling a human operator to monitor and control the robot manually, rather than in autonomous mode.

Summary of the Workflow:

- The process initiates with robot hardware and server setup.
- Based on the operating mode (manual or autonomous), the robot follows either a scripted autonomous flow or operator-controlled procedures.
- Key operations involve live feed monitoring and detection (motion, fire, and smoke).

d. Circuit Diagram

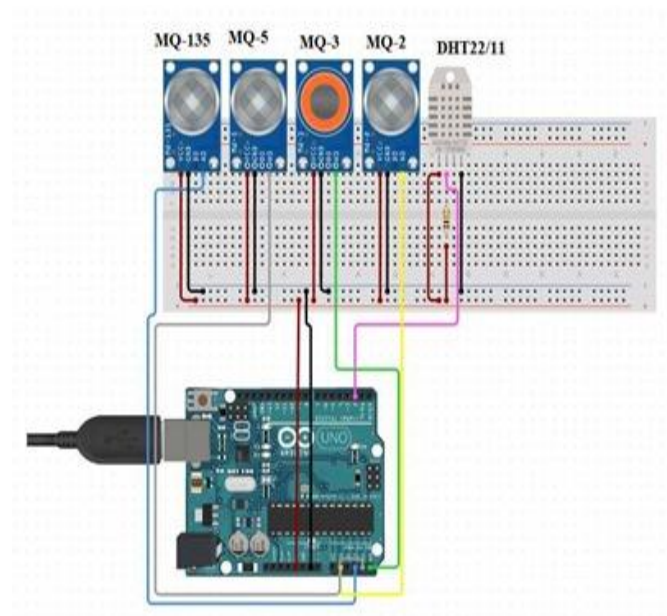


fig3: Gas Circuit

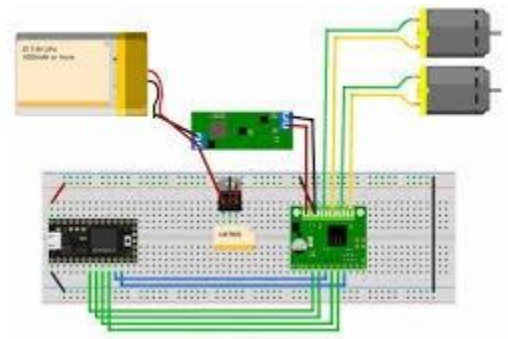


fig4: motor Circuit

We have implemented some modules and there are separate circuits for them. We have plenty of gas sensors so it is a gas stack. Moreover we have a dedicated motor controlling circuit with teensy 4.1

e. Autonomous Algorithm

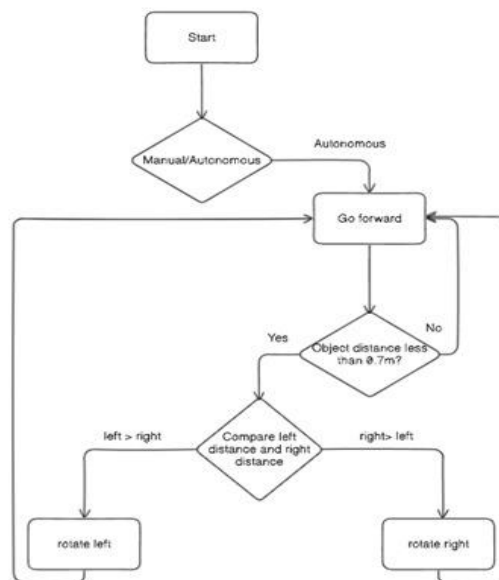


Fig5: Autonomous custom algorithm

Explanation:

1. **Start:**

- The process begins when the robot's system is activated.

2. **Manual/Autonomous Decision:**

- The robot checks if it should operate in **manual** or **autonomous** mode.
- If **manual mode** is selected, the system likely expects human input to guide the robot.
- If **autonomous mode** is selected, the robot proceeds with self-navigation.

3. **Go Forward:**

- Once in **autonomous mode**, the robot begins moving forward. During this phase, it uses sensors to monitor the distance of obstacles in its path.

4. **Object Distance Check (Less than 0.7m?):**

- As the robot moves forward, it continuously checks the distance to objects in its path.
- If there is an obstacle within **0.7 meters**, the robot stops and proceeds to compare the distances on its left and right sides.
- If no obstacle is detected (distance > 0.7 meters), the robot continues moving forward.

5. **Compare Left and Right Distances:**

- If an obstacle is detected, the robot measures the distance to the left and the right of the obstacle. This comparison helps the robot determine which direction has more space to navigate around the object.

6. **Left > Right:**

- If the **left side has more space** (distance to the left is greater than the distance to the right), the robot **rotates left** to avoid the obstacle.

7. **Right > Left:**

- If the **right side has more space** (distance to the right is greater than the distance to the left), the robot **rotates right** to navigate around the obstacle.

8. **Return to "Go Forward":**

- After rotating either left or right, the robot continues the process of moving forward and checking for obstacles again.

f. App Architecture

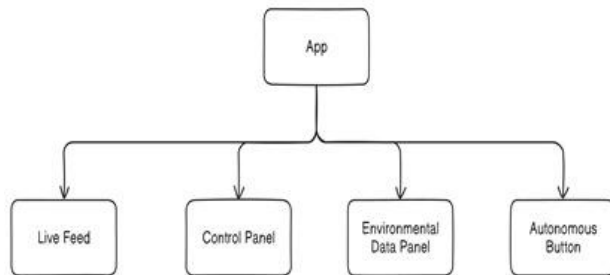


Fig6 : App architecture and design

The app architecture diagram lays out the technical design and components of the app we built.

- **App:** The app serves as the central control hub for the robot. It provides access to various functions needed to monitor and manage the robot's operations.
- **Live Feed:** This feature allows the user to view a live video stream from the robot's camera, providing real-time surveillance or monitoring of the robot's environment.
- **Control Panel:** The control panel provides manual control of the robot, allowing the user to adjust movement and actions. This likely includes options such as moving the robot forward, backward, left, or right.
- **Environmental Data Panel:** This panel likely displays important environmental metrics, such as temperature, humidity, or other sensor data, helping users understand the robot's surroundings.
- **Autonomous Button:** This button toggles the robot's autonomous mode on or off. When activated, the robot will operate based on pre-programmed scripts, making decisions without manual input.

g. Communication Architecture

- **ROS :** It is involved into inter node communication.
- **Websocket Integration :** It is used to transmit and receive data.
- **Socket.io :** For deploying server.

h. Detection Modules

- **Fire Detection (YOLO v8):** Provide a deep dive into the methodology, including the model training (if applicable), detection pipeline, and results.
- **Smoke Detection:** Explain the gas sensor setup, threshold levels for detecting smoke, and how this data is processed and used for decision-making.

- **Motion Detection:** Describe the motion detection logic, sensor usage, and its impact on the robot's decision-making.

i. Environmental Setup/Implementation

The Autonomous Surveillance Robot project integrates a diverse range of hardware and software components to achieve its advanced security and monitoring functionalities. The hardware aspect of the robot starts with its mobility system, which includes durable rubber wheels designed to provide traction on different surfaces, ensuring smooth navigation. High-torque motors are used to drive the wheels, enabling precise control of speed and direction, crucial for obstacle avoidance and maneuverability in complex environments. To further enhance motion control, encoders are attached to the motors to measure rotational speed and direction, offering critical feedback for accurate distance measurement and movement detection. The Pololu motor driver manages the power flow to the motors, ensuring efficient performance while interfacing with the processing unit to maintain control over the robot's movements.

On the sensory side, the robot is equipped with a variety of devices to collect data from its environment. Web Cam module provides visual data essential for surveillance and real-time video streaming, supporting image processing tasks crucial for security purposes. The RPLIDAR A2 sensor enables precise distance measurement and environmental mapping using 2D LIDAR technology. This sensor is vital for the Simultaneous Localization and Mapping (SLAM) algorithm, which allows the robot to map its surroundings while navigating autonomously. Additionally, microphones have been installed to capture ambient audio, enabling sound-based event detection, while sensors for temperature and gas detection allow the robot to monitor environmental conditions and detect potential hazards such as fires or gas leaks.

The robot's processing tasks are handled by the Jetson Xavier NX, which serves as the main control unit. This processing unit is responsible for running the operating system, navigation algorithms, and user interfaces, and is capable of handling complex computations while maintaining energy efficiency. Power to the system is supplied by rechargeable 5000 mAh batteries, offering extended operational time and promoting sustainability through their reusability.

On the software side, a combination of operating systems and frameworks are employed to ensure seamless operation. The Teensy and Arduino mega microcontroller uses a Real-Time Operating System (RTOS) to efficiently manage real-time tasks such as sensor data acquisition and control signal generation. Meanwhile, Ubuntu 20.04, a lightweight Linux distribution, is installed on the Jetson Xavier NX to provide a stable environment for running the robot's software stack. Robot Operating System (ROS) serves as the middleware, facilitating communication between hardware and software components, while algorithms Hector SLAM and AMCL enable autonomous navigation, path planning, and obstacle avoidance. A user interface that includes a remote control and dashboard for data visualization allows users to monitor and control the robot remotely, enhancing flexibility and usability.

j. Manual Control System

- **Control Interface:** From control interface command goes to the backend server running in the jetson Xavier nx.
- **Collision Avoidance in Manual Mode:** We have protection system regarding manual control as well. In no case it will collide with any object even anyone tries to do it with manual control.

Experiment Result

The experiments conducted with the Autonomous Surveillance Robot demonstrated its ability to successfully navigate indoor environments, detect objects in real-time, and monitor environmental conditions. The robot's Simultaneous Localization and Mapping (SLAM) algorithm effectively created a dynamic map of its surroundings while tracking its position. Path planning and obstacle avoidance algorithms ensured the robot could safely maneuver around obstacles. Environmental sensors, including gas and temperature sensors, were able to detect hazardous conditions, with the system providing timely alerts. The web-based dashboard enabled smooth remote control and real-time monitoring of the robot's status, proving the system's usability and reliability.

Impact

Environmental Effect

The use of rechargeable 5000 mAh batteries contributes to environmental sustainability by reducing the need for disposable power sources, which helps lower electronic waste. Moreover, the robot's ability to monitor hazardous environmental conditions such as gas leaks can prevent potential damage to ecosystems within its operational area.

Economical Effect

The implementation of autonomous surveillance robots could lead to significant cost savings over time. By reducing the reliance on human security personnel, operational costs can be minimized. Additionally, the automation of continuous surveillance and early detection of hazards reduces the risks associated with damage or loss, contributing to long-term cost efficiency.

Social Effect

On a societal level, the robot's deployment can enhance public safety by providing continuous, real-time monitoring without the limitations of human operators. Its autonomous operation in security-sensitive environments can mitigate risks and prevent accidents, contributing to the well-being of people in its operating areas. The project also demonstrates the potential of robotics to transform security operations, leading to greater social trust in autonomous systems.

Limitations & Future Work

The current system, while functional, has several limitations. The robot's performance is constrained by battery life, limiting extended operational periods. The reliance on indoor mapping means the robot is not suited for outdoor use or highly dynamic environments. Additionally, while the sensors used are effective, improvements could be made in detecting a wider range of hazardous materials.

Future work will focus on extending battery life and integrating solar charging systems to further enhance operational sustainability. The robot's design can also be adapted for outdoor environments by incorporating 3D LIDAR and enhanced environmental sensors for more dynamic monitoring. Finally, improvements in AI and machine learning algorithms could improve the robot's ability to recognize complex patterns of suspicious activity, enhancing overall functionality.

Conclusion

In conclusion, the Autonomous Surveillance Robot project demonstrates the practical application of cutting-edge technologies in improving security and safety. By integrating autonomous navigation, real-time object detection, environmental monitoring, and remote-control capabilities, the robot offers an efficient, scalable, and reliable solution for surveillance tasks. While limitations exist, they provide valuable opportunities for future enhancements. This project represents a significant step forward in the development of autonomous systems capable of transforming how security operations are conducted across various industries and public spaces.

References

1. T.-H. S. Li, C.-Y. Chen, Y.-C. Yeh, et al., "An autonomous surveillance and security robot team," in *Proc. IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO '07)*, Hsinchu, China, Dec. 2007, pp. 1–6.
2. T. Theodoridis and H. Hu, "Toward intelligent security robots: A survey," *IEEE Trans. Syst. Man Cybern. Part C (Appl. and Rev.)*, vol. 42, pp. 1219–1230, 2012.
3. G. A. Hollinger, B. Englot, F. S. Hover, U. Mitra, and G. S. Sukhatme, "Uncertainty-driven view planning for underwater inspection," *Auton. Robots*, vol. 33, no. 1-2, pp. 89–106, 2012.
4. B. Song, C. Ding, A. Roy-Chowdhury, and J. Farrell, "Multi-camera tracking of a vehicle in a structured environment," in *Proc. 2008 IEEE Int. Conf. on Robotics and Automation (ICRA)*, Pasadena, CA, USA, 2008, pp. 561–566.
5. R. T. Oropesa, P. A. López, and L. P. Cuellar, "Autonomous surveillance robot with low-cost ultrasonic localization system," in *Proc. 2012 IEEE Colombian Conf. on Communications and Computing (COLCOM)*, Bogotá, Colombia, 2012, pp. 1–6.