Hazard Scout : Multi-purpose Emergency Response Robot.

Submitted in partial fulfillment of the requirements
of the degree of

B. Tech. in Computer Engineering

By

Kruti Shah	**60004210122**
Khushi Jobanputra	**60004210147**
Manasvi Gupta	**60004210235**
Shashwat Shah	**60004220126**

Guide(s):

Name of the Guide
Prof. Ruhina Kanani

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

University of Mumbai

2024-2025

CERTIFICATE

This is to certify that the project entitled **"HazardScout: Multipurpose Emergency Response Robot"** is a bonafide work of **"Kruti Shah" 60004210122 , "Khushi Jobanputra" 60004210147, "Manasvi Gupta" 60004210235 , "Shashwat Shah" 60004220126** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of B. Tech. in Computer Engineering

Prof. Ruhina Kanani
Internal Guide

Dr. Meera Narvekar
Head of the Dept

Dr. Hari Vasudevan
Principal

Project Report Approval for B.Tech.

This project report entitled by **HazardScout : Multipurpose Emergency Response Robot** by **Manasvi Gupta , Kruti Shah, Khushi Jobanputra, Shashwat Shah** is approved for the degree of ***B.Tech. in Computer Engineering.***

Examiners

1.________________________________

2.________________________________

Date:

Place:

Declaration

I/We declare that this written submission represents my/our ideas in my/our own words and where others' ideas or words have been included, I/We have adequately cited and referenced the original sources. I/We also declare that I/We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my/our submission. I/We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

Kruti Shah-60004210122

(Signature)

Khushi Jobanputra-60004210147

(Signature)

Manasvi Gupta-60004210235

(Signature)

Shashwat Shah-60004220126

Date:

Abstract

In hazardous industrial and disaster scenarios, manual inspections often expose workers to dangerous conditions like toxic gases, chemical spills, and structural risks. HazardScout: A Multi-Purpose Emergency Response Robot aims to mitigate these challenges by combining advanced mobility, IoT integration, and real-time environmental sensing. Designed for industries such as petrochemicals and manufacturing, as well as search-and-rescue missions, the robot offers reliable inspection and emergency response capabilities.

The robot uses a wheeled and magnet-based propulsion system powered by motors, enabling movement across horizontal and vertical pipes, sharp turns, and uneven surfaces. Equipped with chemical, gas, and distance sensors, it detects leaks and structural anomalies while capturing real-time images through its onboard camera. Its compact design ensures adaptability in confined spaces and disaster zones.

Raspberry Pi-driven with Wi-Fi connectivity, the system supports remote monitoring, allowing users to analyze live data and imagery. The robot's versatile design addresses limitations of traditional systems, such as surface dependency and navigation inefficiencies, making it suitable for diverse environments. Additionally, its life-saving application in locating trapped individuals during disasters enhances its societal impact.

HazardScout bridges gaps in adaptability, efficiency, and autonomy while offering a scalable platform for future enhancements like AI-driven navigation and improved connectivity in remote areas. This innovative solution promises to redefine industrial safety and disaster management through robotics and IoT integration.

Contents

List of Figures

List of Abbreviations

Sr. No.	Abbreviation	Expanded form
1	IOT	Internet of things
2	GUI	Graphical user interface
3	LPG	Liquified Petroleum Gas
4	CAD	Computer aided diagram

1. INTRODUCTION

1.1 Description

Emergency inspections in industries like gas and chemical manufacturing face challenges such as navigating complex pipelines, detecting leaks, and ensuring worker safety in hazardous environments. Existing solutions struggle with vertical and horizontal pipe traversal, sharp turns, and obstacles like joints, while traditional methods expose workers to risks and lack adaptability.

Our robot addresses these issues with a versatile mobility system, featuring one omnidirectional and two standard wheels powered by servo motors, enabling it to climb pipes, navigate joints, and traverse various surfaces. Magnets provide stability, and a Raspberry Pi controls the camera and distance sensors for real-time hazard detection and monitoring. IoT integration ensures remote operation and data transmission, offering a safe, efficient, and cost-effective solution for emergency inspections.

1.2. Problem Formulation

In industrial and residential settings, managing emergencies such as pipeline inspections and leakage detection presents significant challenges due to limitations in current technologies. Our emergency response robot is designed to address these challenges through innovative features that ensure adaptability, efficiency, and cost-effectiveness.

Limitations & Features

1. Accessibility Issues and Navigation Limitations

Limitation: Existing technologies struggle to inspect pipelines and other hard-to-reach areas, such as confined spaces, industrial ducts, and external pipes. These environments often present challenges that traditional robots cannot overcome, leading to incomplete or inefficient inspections. Additionally, navigation obstacles like pipe joints, sharp turns, uneven terrains, and ground-based obstructions further hinder the effectiveness of conventional solutions.
Feature: Our robot overcomes these limitations with an advanced mobility system using one omnidirectional and two standard wheels powered by servo motors. It moves horizontally, climbs

vertically, and navigates joints, turns, and uneven surfaces. Magnets ensure stability on metal surfaces, enabling efficient inspections in confined spaces and challenging terrains.

2. Safety Concerns

Limitation: Human operators face significant risks during manual inspections, including exposure to hazardous gases, chemical leaks, and unsafe environments. These dangers are particularly acute in industrial settings and during emergency scenarios like earthquakes, where locating people trapped in caves or debris is critical.
Feature: The robot is equipped with a high-resolution camera and distance sensors, enabling real-time monitoring and remote operation. This eliminates the need for human intervention in high-risk environments, ensuring operator safety.

3. Inspection and Hazard Detection Challenges

Limitation: Current tools lack precision and versatility in identifying leaks or structural damage in pipelines, especially in complex industrial settings.
Feature: With its integrated camera and distance sensors, the robot offers detailed inspection capabilities, detecting leaks, blockages, and structural flaws. The use of Raspberry Pi allows for efficient processing and IoT integration for live data transmission and remote control.

4. Cost-Effectiveness

Limitation: Existing systems rely on multiple specialized tools and extensive manual labor, increasing operational costs.
Feature: By combining multiple functionalities such as mobility, hazard detection, and real-time monitoring into a single device, our robot significantly reduces the need for additional equipment and labor. Its versatility and efficiency make it a cost-effective solution for various industries.

1.3 Motivation

The given figure 1.3.1 diagram illustrates the progression of design ideas for a robot designed to climb pipes and navigate challenging surfaces.

1. Initial Setup: The robot is equipped with sensors and steel wheels but faces two problems: inability to overcome obstacles and climb pipes.

2. First Idea: Conveyor belts are introduced to tackle these issues. However, this solution struggles on horizontal surfaces and cannot effectively climb obstacles.
3. Second Idea: Friction grippers with springs and wheels are proposed but face strength and reliability issues.
4. Final Solution: Ferromagnetic omnidirectional wheels are adopted, providing enhanced capabilities, such as climbing obstacles and traversing both horizontal and vertical surfaces effectively.

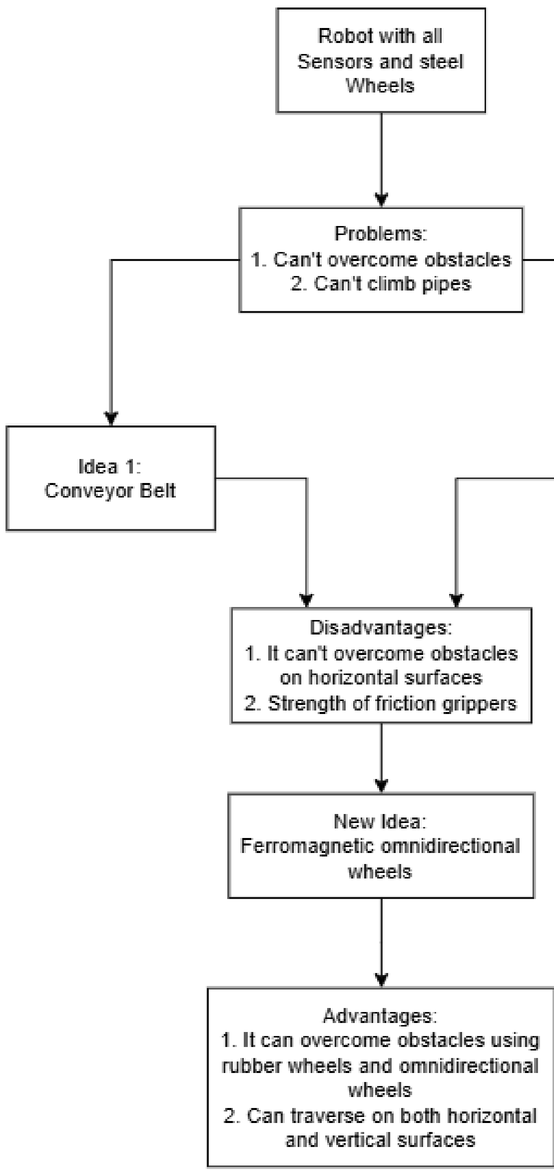

Figure 1.3.1 : Flow of Ideas

1.4. Proposed Solution

The Ferromagnetic Pipe Climbing Robot employs an advanced mobility system to tackle the limitations faced by previous approaches. Here's a breakdown of the solution based on the existing concept:

Method/Technique Used

1. Ferromagnetic Wheels with Omnidirectional Movement**:**

This robot utilizes ferromagnetic wheels, allowing it to adhere to metallic pipes and traverse with ease. These wheels are powered by servo motors, enabling omnidirectional movement, which provides flexibility to the robot as it moves along pipes horizontally and vertically.

This setup allows the robot to navigate even pipe joints and turns without losing grip or stability, which was a limitation in earlier designs like conveyor belts that couldn't handle such maneuvers effectively.

2. Integration of Raspberry Pi and Sensors**:**

A Raspberry Pi board serves as the central control unit, enabling remote operation through Wi-Fi. It processes inputs from various sensors (camera, gas detectors, distance sensors, and environmental sensors), providing real-time data and decision-making capabilities for efficient inspection tasks.

The integration of sensors makes it suitable for pipe inspection, detecting gas leaks, and navigating confined spaces effectively.

3. Lightweight and Modular Design**:**

The system is designed to be lightweight, leveraging the magnetic mechanism and servo-powered wheels to ensure stability without adding excess weight, improving overall maneuverability.

Unlike previous solutions, this robot does not rely on heavy machinery or cumbersome setups like conveyor belts, making it more accessible for various industries, including chemical and gas industries.

How It Overcomes Drawbacks

1. Overcoming Obstacles**:**

The use of magnetic wheels helps overcome obstacles and ensures the robot can cling firmly to pipe surfaces, overcoming traditional mobility issues like climbing, joint traversal, and turn navigation.

2. Improved Efficiency and Cost**:**

The robot's lightweight design and the use of servo motors offer greater efficiency while reducing overall cost and energy consumption, addressing the drawbacks of previous systems that were heavy or energy-inefficient.

End-User Benefits

1. Industrial Use for Continuous Monitoring:

The robot is suited for continuous pipe maintenance and monitoring, especially in gas, chemical, and drainage systems, significantly reducing the need for human intervention in hazardous environments.

2. Emergency Response in Confined Spaces:

The robot can be used in disaster response situations like earthquakes, where it can help search for trapped individuals in confined spaces, providing crucial rescue and recovery data.

3. Cost-Effectiveness and Versatility:

The robotic system is designed to be more cost-effective than previous heavy-duty robotic systems, providing a scalable solution for both large and small-scale industrial and emergency operations.

This Ferromagnetic Pipe Climbing Robot provides a practical, versatile, and efficient solution that overcomes many limitations of previous systems, offering a reliable tool for industrial inspections and emergency rescues.

1.5 Scope of the project

The proposed robot is designed to address a wide range of applications, including industrial inspections and disaster response. However, its functionality is subject to certain constraints and limitations:

1. Domain-Specific Constraints:

The robot is optimized for environments with metallic or stable surfaces where magnetic propulsion and grippers can be utilized effectively. It may face challenges in non-metallic pipelines or highly irregular terrains.

2. Application Constraints:

The robot's navigation depends on distance sensors and a camera, which may struggle in extremely low visibility or environments with high interference (e.g., dense debris or reflective materials).
While designed for various terrains, its performance in highly cluttered or unstable areas, such as landslides or collapsing structures, may require further testing and adaptation.
The reliance on servo motors and magnets might limit its climbing ability on pipes or surfaces with unusual geometries or non-magnetic materials.

3. Power and Connectivity:

The robot's performance depends on battery capacity and stable IoT connectivity for remote control. Prolonged operations or weak signals in remote areas may reduce its efficiency.

While these limitations exist, the robot provides significant capabilities for confined-space inspections and emergency responses, addressing critical gaps in current technologies.

2. REVIEW OF LITERATURE

Recent advancements in robotics have focused on developing systems capable of operating in hazardous environments, such as industrial plants, chemical factories, and post-disaster situations. One notable approach is the use of magnetic-wheeled climbing robots for non-destructive testing, as discussed by Zhang et al. (2022). These robots employ magnetic adhesion combined with adaptive mechanisms, enabling them to climb vertical metal surfaces and perform critical tasks like detecting gas leaks. However, the primary limitation of this technology is its restriction to metallic surfaces. To expand the operational range of robots, HazardScout integrates friction-based grippers, which allow it to climb and operate on both metallic and non-metallic surfaces, making it a more versatile solution in diverse industrial environments.

Smith et al. (2023) tackled the issue of robotic navigation through complex and flexible environments, particularly pipelines, by using articulated joints. Their system allows robots to navigate through narrow and winding pipelines with ease. However, the lack of autonomous decision-making in their system led to slower operations in unpredictable environments. HazardScout improves on this by incorporating AI-driven navigation, which enables real-time decision-making and more efficient navigation through difficult spaces, such as pipelines or during emergency response situations.

The integration of multiple sensors for gas leak detection is another critical area of research, as highlighted by Kumar et al. (2021). Their multi-sensor system proved highly effective in detecting gas concentrations in real-time, but the high energy consumption of the sensors limited the robot's operational duration. To overcome this limitation, HazardScout incorporates energy-efficient sensor management, allowing for longer operational periods without frequent recharging, thus improving the robot's ability to conduct prolonged inspections in hazardous environments.

Patel et al. (2023) explored the use of sensor fusion in autonomous firefighting robots, integrating data from smoke, gas, and temperature sensors to enhance the robot’s fire detection capabilities. Despite improved accuracy, the robots still faced challenges with response times, which remained relatively slow in critical situations. HazardScout addresses this by implementing faster real-time processing, allowing for quicker responses to gas leaks, fires, or other hazardous events.

Diaz et al. (2021) investigated snake-like robots designed for confined space navigation, making them ideal for use in post-disaster scenarios. Their serpentine movement allowed them to move through tight and obstructed spaces but was limited by the unpredictable nature of the environment. HazardScout combines the benefits of flexible movement with omnidirectional wheels and friction grippers, ensuring more reliable operation in confined spaces, even when navigating uneven or unpredictable terrain.

Wilson et al. (2022) introduced magnetic wall-climbing robots for industrial inspections, specifically targeting environments like petrochemical plants. These robots proved effective in scaling vertical tanks and pipes, but their application was confined to magnetic surfaces. To increase the versatility of the HazardScout, friction-based grippers allow it to operate on both magnetic and non-magnetic surfaces, making it suitable for a wider range of industrial and emergency settings.

The field of fire detection and suppression has also seen the introduction of robotic systems, such as those studied by Lee et al. (2022). These robots use thermal and infrared imaging to detect fires and autonomously suppress them. However, their use is limited to flat, open spaces, making them ineffective in more complex environments. HazardScout, on the other hand, can operate in multi-level and confined spaces, offering better flexibility in handling fire emergencies in industrial and confined areas.

Wireless sensor networks for continuous monitoring of hazardous environments, as discussed by Li et al. (2022), offer autonomous detection and alarming capabilities for gas leaks. Despite their effectiveness, the high maintenance cost of these sensor systems poses a barrier to widespread use. HazardScout utilizes more cost-effective sensor networks, enhancing its affordability and making it a more accessible solution for routine inspections and emergency operations.

In the field of industrial safety, real-time data transmission is crucial, as demonstrated by Brown et al. (2021), who developed robots capable of transmitting high-resolution video and sensor data in hazardous environments. While effective, these systems can struggle with connectivity in low-signal areas. To address this, HazardScout improves data transmission through advanced wireless technologies, ensuring reliable communication even in remote industrial zones or disaster-stricken areas.

Finally, Tanaka et al. (2023) investigated firefighting robots equipped with water jet propulsion and thermal imaging for fire suppression. Although these robots can extinguish fires in challenging environments, they are limited by their water supply, which restricts the duration of operations. HazardScout, being a multi-functional robot, does not focus on fire suppression but instead offers a versatile solution for a variety of emergency response scenarios, including gas leak detection and search-and-rescue operations.

The development of robots for industrial inspection, emergency response, and hazardous environment navigation has greatly advanced through the integration of AI, flexible movement mechanisms, and sensor fusion. The HazardScout robot builds on existing research by addressing key limitations such as energy consumption, slow response times, and restricted mobility, offering a more comprehensive and adaptable solution for use in industrial inspections, firefighting, and disaster recovery operations.

3. SYSTEM ANALYSIS

3.1. Functional Requirements

The HazardScout robot is designed to perform emergency response and industrial inspection tasks. Below are the core functional requirements:

1. Gas and Chemical Leak Detection:

The robot shall detect hazardous gases (e.g., methane, carbon monoxide) and chemicals (e.g., ammonia, chlorine) using a combination of gas and chemical sensors, including MQ-2, MQ-5, MQ-7, and chemical-specific sensors.

The system shall send real-time alerts to the operator or monitoring system if gas or chemical concentrations exceed a predefined threshold.

2. Pipe Navigation and Inspection:

The robot shall be capable of climbing pipes, overcoming joints, and navigating turns within industrial and confined spaces.

It should be able to move both vertically on pipes and horizontally on the ground.

3. Search and Rescue Operations:

Equipped with a camera and distance sensors, the robot shall locate and identify humans trapped in confined spaces, such as caves or collapsed buildings.

It should provide real-time visual and distance data to rescuers, enabling better coordination during rescue operations.

4. Real-Time Communication:

The robot shall be capable of transmitting sensor data, images, and videos to a remote operator in real-time, even in low-connectivity environments.

The system must support stable wireless communication over Wi-Fi or LTE.

5. Autonomous Navigation:

The robot shall feature AI-driven navigation, allowing it to autonomously navigate through complex environments, detect and avoid obstacles, and make route adjustments based on real-time environmental data.

6. Environmental Monitoring:

The robot shall be equipped with environmental sensors, including temperature and humidity sensors (e.g., DHT11), to monitor the surroundings and provide essential data for decision-making.

7. Battery Management:

The robot shall be capable of continuous operation for at least 4 hours on a single charge, with a power-saving mode to optimize battery life.

3.2. Non-Functional Requirements

The HazardScout system must meet several non-functional criteria to ensure reliable, efficient, and user-friendly operation:

1. Performance:

The robot should travel at a speed of 1-2 m/s on both ground and vertical surfaces.
Gas and chemical sensors should provide a response time of no more than 2 seconds when detecting hazardous materials.

2. Reliability:

The robot must be capable of functioning for 4+ hours without failure under harsh environmental conditions. The system shall have an uptime of 99% in normal operating conditions.

3. Usability:

The robot’s control system must be intuitive, with a graphical user interface (GUI) for the operator to monitor and control the robot, view data, and receive alerts.Training for operators should not exceed 1 hour.

4. Safety:

The robot must include fail-safe mechanisms to stop operations in case of critical system failures (e.g., loss of power or communication).Safety protocols shall be in place to avoid harm to human operators and the environment.

5. Scalability:

The system must be scalable for future upgrades, such as integrating additional sensors or AI-driven features.

6. Environmental Impact:

The robot shall be energy-efficient, utilizing low-power components to minimize its environmental footprint. It should be constructed from environmentally friendly materials.

7. Maintainability:

The robot must be easy to maintain, with modular components that can be replaced or upgraded without specialized tools. Remote software updates should be supported.

3.3. Specific Requirements

Hardware Requirements:

1. Core System:

Processor: Raspberry Pi 4 Model B or equivalent for managing sensors, motors, and communication systems.
Battery: Li-ion or Li-polymer battery (capacity of at least 10,000mAh) for 4+ hours of operation.
Frame: Lightweight, durable materials such as aluminum or carbon fiber for the robot's body.

2. Mobility System:

Wheels: One omnidirectional wheel and two standard wheels, driven by servo motors for movement on both ground and vertical surfaces.
Magnets: For propulsion and stability on metal surfaces.

3. Sensors:

Gas Sensors: MQ-2, MQ5,MQ-7, and other chemical-specific sensors for detecting gases such as methane, carbon monoxide, ammonia, and chlorine.
Temperature and Humidity Sensors: DHT11 or similar for environmental monitoring.
Distance Sensors: Ultrasonic or LiDAR sensors for obstacle detection and mapping.
Camera: High-definition camera with night vision for inspection and search-and-rescue operations.

4. Communication System:

Wi-Fi or LTE Modem: For stable communication with remote operators.

Microcontroller: For managing the sensor data, motor control, and communication interfaces.

Software Requirements:

1. Operating System:

Raspberry Pi OS (or equivalent Linux-based system) for handling sensor data processing, motor control, and communications.

2. Control Software:

Autonomous Navigation Software: AI-based system for route optimization and real-time decision-making.

Real-Time Communication Software: For transmitting data (video, sensor readings) to the control center.

Sensor Data Processing Software: For analyzing gas, chemical, temperature, humidity, and distance data in real-time.

3. User Interface:

Graphical User Interface (GUI): For controlling the robot and visualizing real-time data.

Alert System: To notify operators of hazardous conditions or system failure.

3.4. Use-Case Diagrams and description

The use case diagram provided illustrates the functional interactions between the user (actor) and the robot system. Here's a breakdown of the key elements:

1. Actor (User): The actor in this system is the user, who interacts with the robot. The user controls the robot by performing various actions such as powering on the robot, connecting to Wi-Fi, and inspecting pipes.
2. Power on Robot: The user starts by turning on the robot, which is the initial step before it can begin performing its functions.

3. Connect to Wi-Fi: Once the robot is powered on, the user establishes a connection to Wi-Fi to enable remote control and data transmission between the robot and the user.
4. Inspect Pipes: The user instructs the robot to inspect pipes, triggering the robot to engage its sensors, camera, and other functionalities related to inspection tasks.
5. Sensors: The robot uses a variety of sensors, which include:

Chemical Sensors: These sensors detect chemical substances in the environment.

Industrial Sensors: These are specialized sensors for detecting specific conditions within industrial settings.

Gas Leakage Sensors: The robot can detect gas leaks through dedicated sensors.

Distance Sensor: A sensor that measures the proximity of objects around the robot, aiding in navigation and obstacle detection.

6. Camera: The robot's camera captures images or video during inspections, which can be analyzed later for detailed inspection data.
7. Detect Obstacles: The robot detects obstacles in its path using its sensors. This triggers the system to decide on a course of action to overcome these obstacles.
8. Overcome Obstacles: The robot moves using its wheels and magnets to navigate around obstacles and continue forward movement. It uses its mobility system (wheels and magnets) to overcome physical barriers in its path.
9. Detects Turn: The robot is also capable of detecting turns in the pipe or path using its sensors, enabling it to navigate corners or bends.

This use case diagram reflects how the robot, equipped with multiple sensors and cameras, interacts with the user for effective pipe inspections, hazard detection, and obstacle navigation in real-time. The user remotely controls the robot and receives data for decision-making while the robot autonomously handles navigation and inspection tasks.

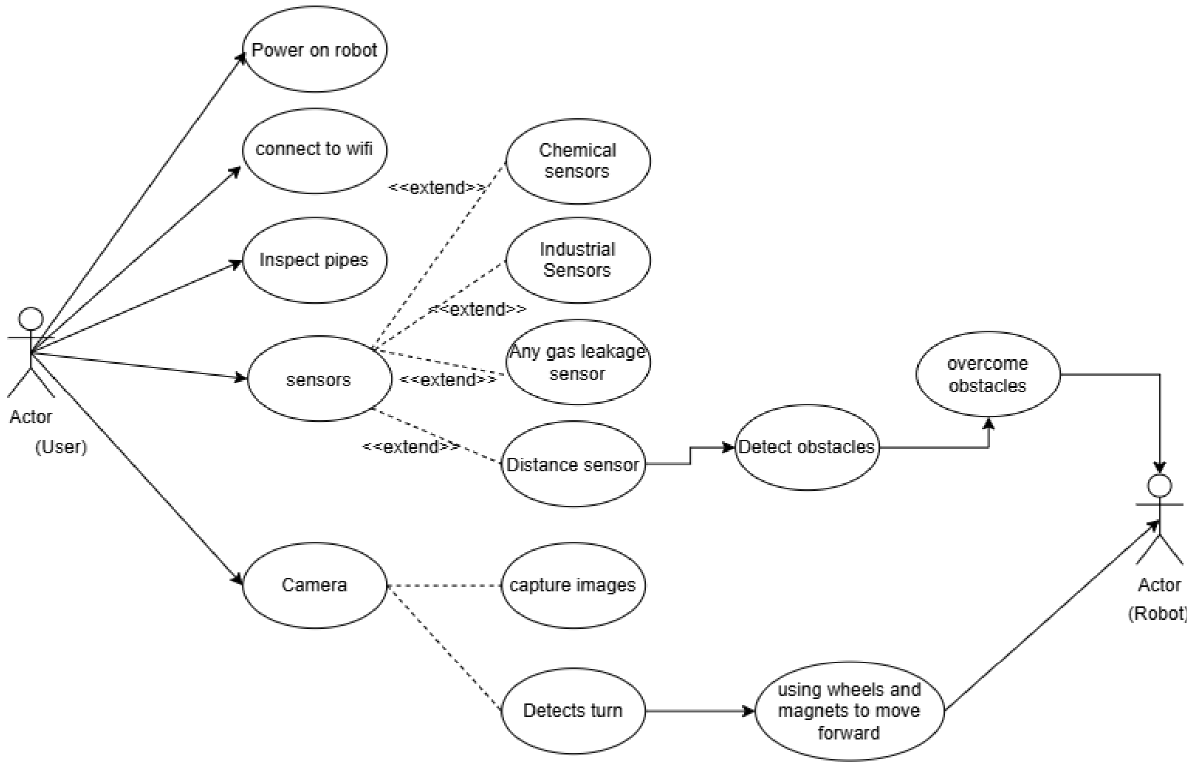

Figure 3.4.1 : Use case Diagram

This figure 3.4.1 illustrates the use cases of the project and simplifies the process of understanding for the end user.

4. ANALYSIS MODELING

4.1. Activity Diagrams

The activity diagram illustrates the functional flow of the pipe-climbing robot system. The process begins with the user powering on the robot and ensuring essential components, like the camera for inspection, are installed. Once the camera is operational, the Raspberry Pi serves as the control unit, coordinating all components. The user connects additional sensors, such as the distance sensor, which becomes operational for environmental detection.

Next, magnets are integrated to enable the robot to adhere to the metal pipe surfaces, and high-torque servo motors are installed for mobility. After successful installation, the robot begins its tasks: it detects obstacles on its path and employs the magnets and motors to overcome these obstacles. It also identifies pipe joints and turns, ensuring it can navigate bends and sharp angles efficiently.

Through continuous sensing and adjustment, the robot is capable of climbing pipes, overcoming challenging terrains, and maintaining consistent performance. This system ensures robust operation for inspection and hazard-detection tasks in industrial settings.

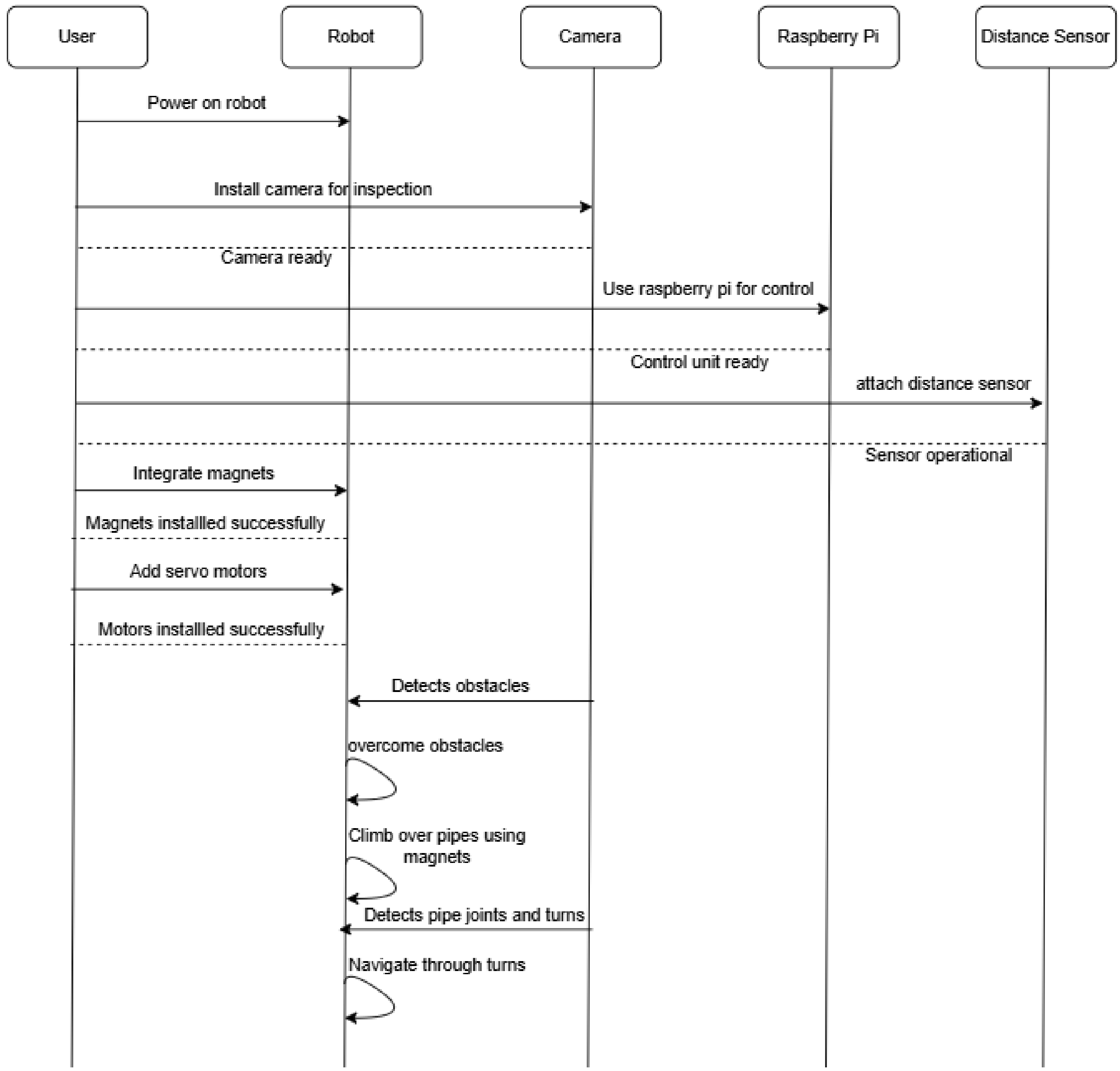

Figure 4.1.1 : Activity Diagram

The figure 4.1.1 illustrates the activities of the robot.

4.2. TimeLine Chart

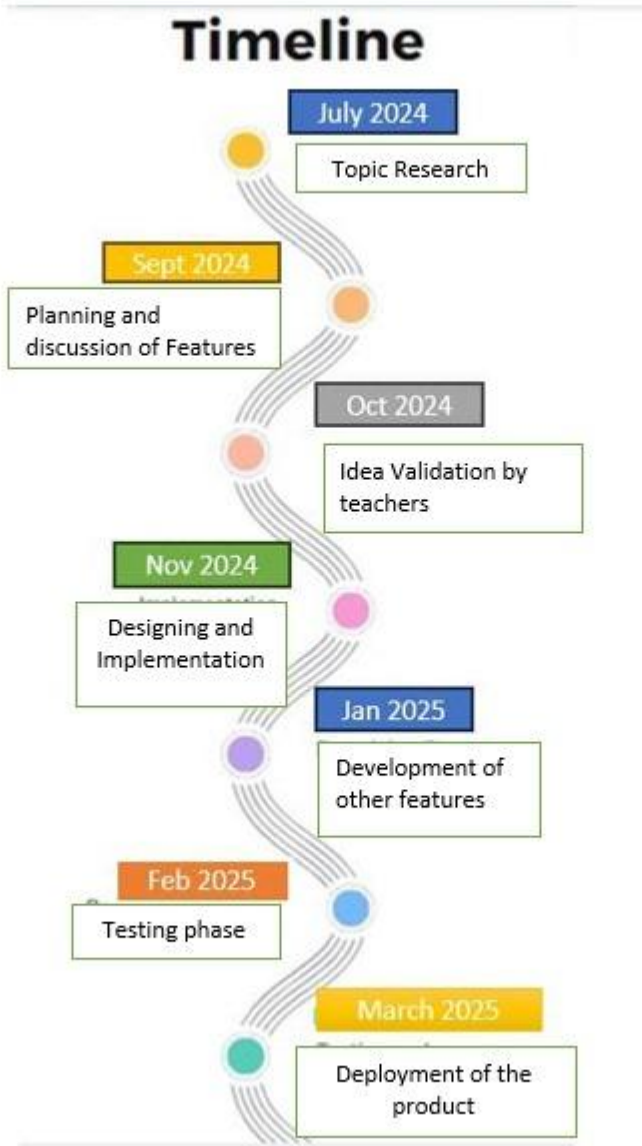

Figure 4.2.1 : Time Line Chart

5. DESIGN

5.1. Architectural Design

1. User Interaction (Start to Authentication)

Login/Signup Using App: The process begins with the user accessing a mobile or web app to log in or sign up.

Ensure Wi-Fi Connectivity: Stable Wi-Fi is a mandatory requirement to fetch and communicate data.

Authenticate User via Firebase Authentication: Firebase ensures secure user authentication, allowing only authorized users to control the robot.

2. Robot Control and Data Management

Control the Directions for the Robot: Through the app, users can control the robot's movement and direction on the pipeline.

Firebase Connectivity: Used as a real-time database for seamless communication between the app and the Raspberry Pi. It ensures that commands and sensor data are transmitted reliably.

3. Robot Mechanisms

Raspberry Pi 4B: Acts as the main processing unit, handling inputs from sensors, camera, and motors.

Motors Help Move the Robot Using Ultrasonic Sensors: Ultrasonic sensors guide the robot in avoiding collisions and navigating the pipeline.

Omnidirectional Wheels: Assist in overcoming obstacles by providing maneuverability on uneven surfaces or around obstructions.

Obstacle Detection Using Camera: The camera module, integrated with obstacle detection algorithms, ensures the robot avoids hazards.

4. Environmental Sensing

Sensors for Gas and Chemical Detection: These detect the presence of harmful gases like LPG or other leaks.
Temperature and Humidity Sensor: Monitors the environmental conditions inside the pipeline for safety and assessment.

5. Visual Insights

Camera Module for Capturing Real-Time Images: Captures the pipeline's internal structure and potential leaks.
Flashlight for High-Quality Output: Ensures the camera performs well even in low-light conditions.
Provide Visual Insights: The captured images are relayed back to the user for analysis.

6. Localization and Insights

Live Location: Tracks the robot's position in real-time, helping identify the exact location of issues.
Helps Detect Accurate Location of Leakage: Combines live location and sensor data for pinpoint accuracy in leakage detection.

7. Data Collection and Machine Learning

Access All Readings and Train Model with the Datasets: Sensor data and images are collected and fed into a machine learning model.
ML Algorithms for Detection and Improvement: The model improves detection accuracy and adapts to new pipeline conditions over time.

8. End of Workflow (Stop)

The workflow ends when all necessary inspections are performed, and the robot stops its operation.

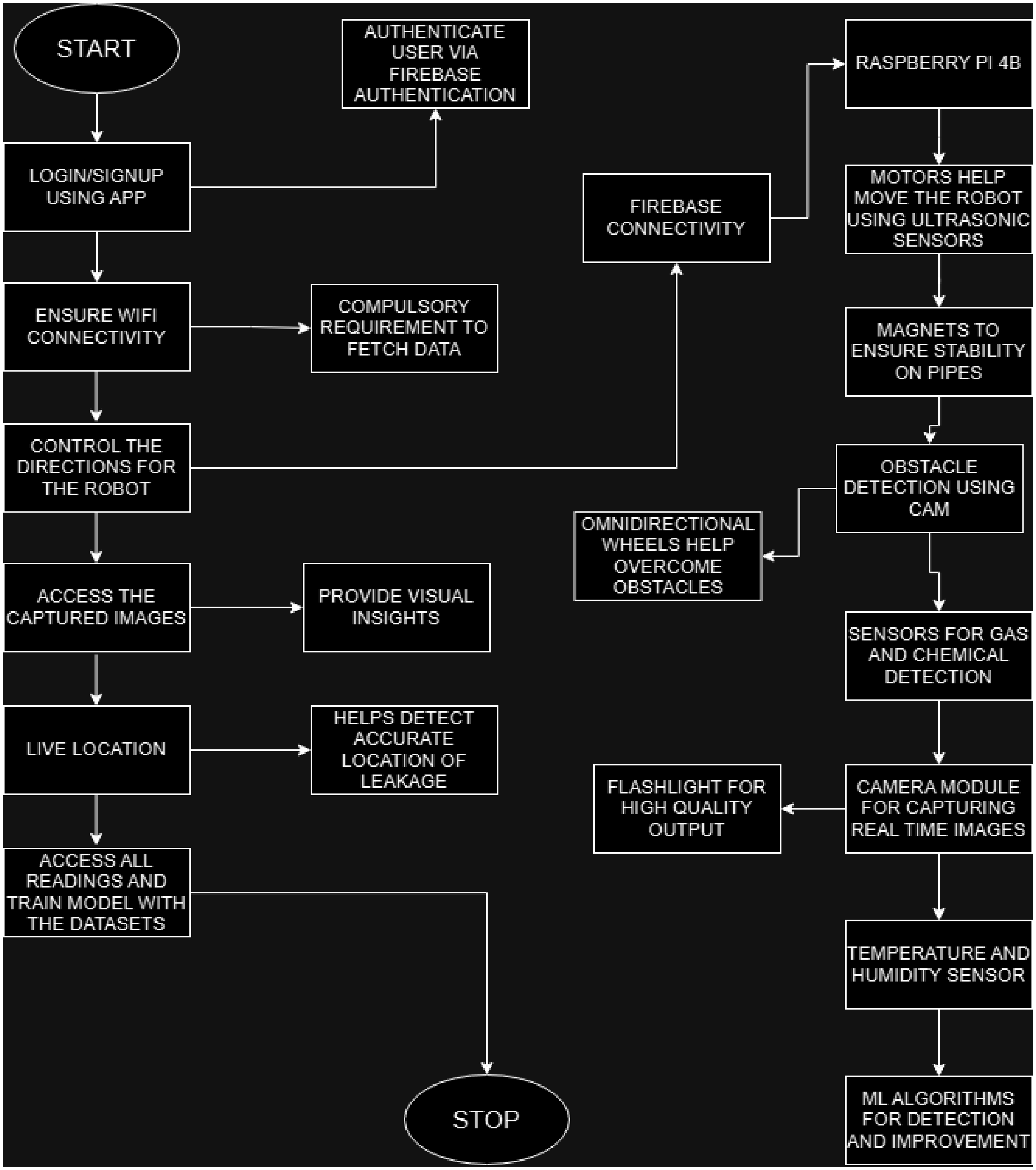

Figure 4.3.1: Architectural Diagram

The figure 4.3.1 illustrates the architectural diagram of the system.

5.2. User Interface Design

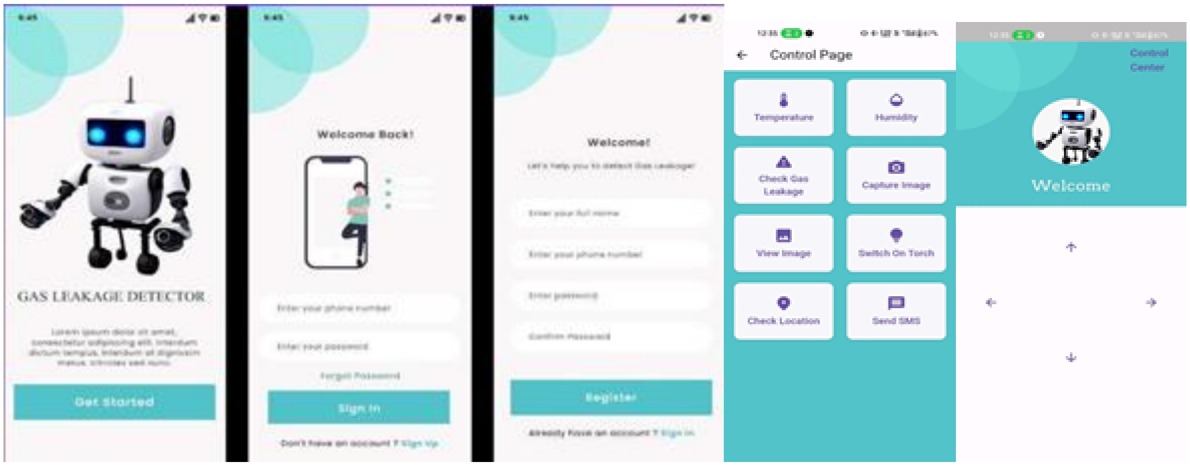

Figure 5.2.1: User Interface Design

This figure is the ui of the app that will be IOT controlled.

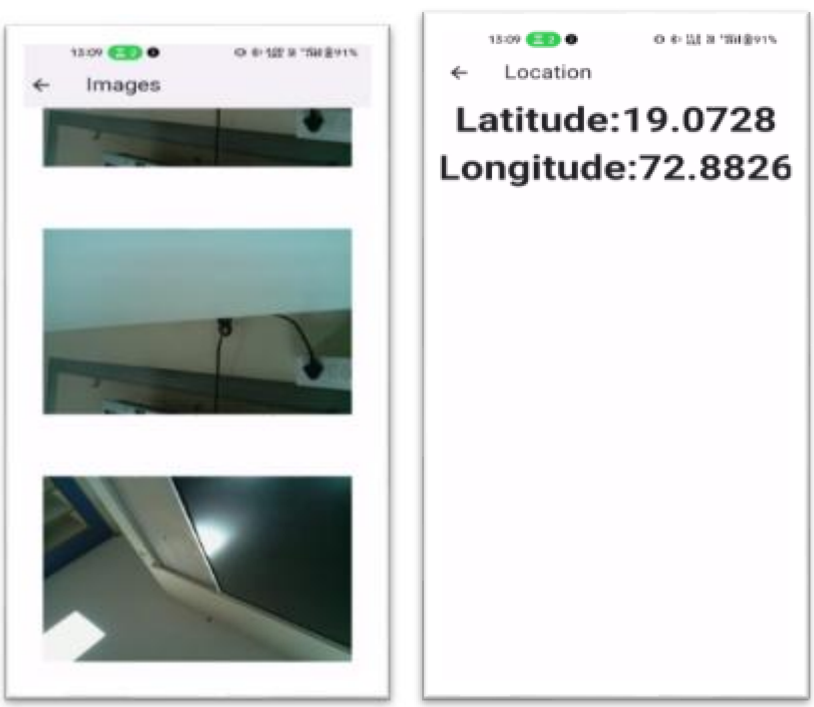

Figure 5.2.2: Mobile Application interface

This figure illustrates the working of the mobile application.

6. IMPLEMENTATION

6.1. Module Implemented

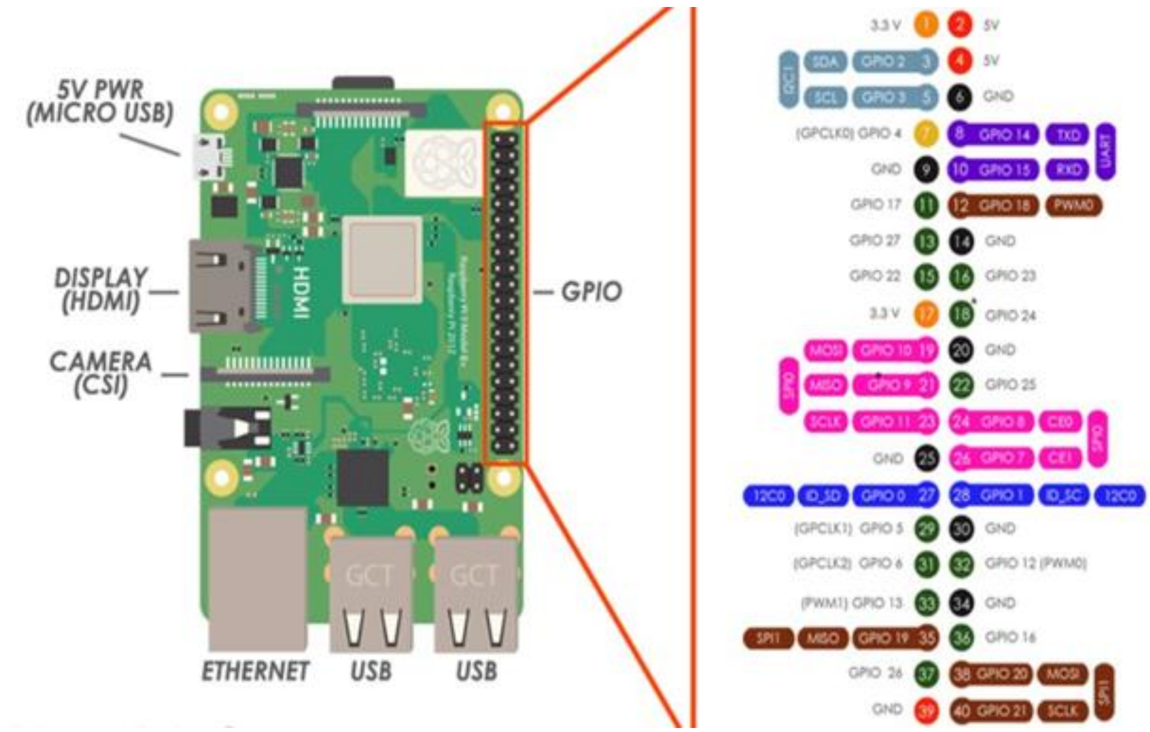

Figure 6.1.1 : Raspberry Pi Pinout

The figure 6.1.1 illustrates the pins present in raspberry pi which can help understand the circuit diagram.

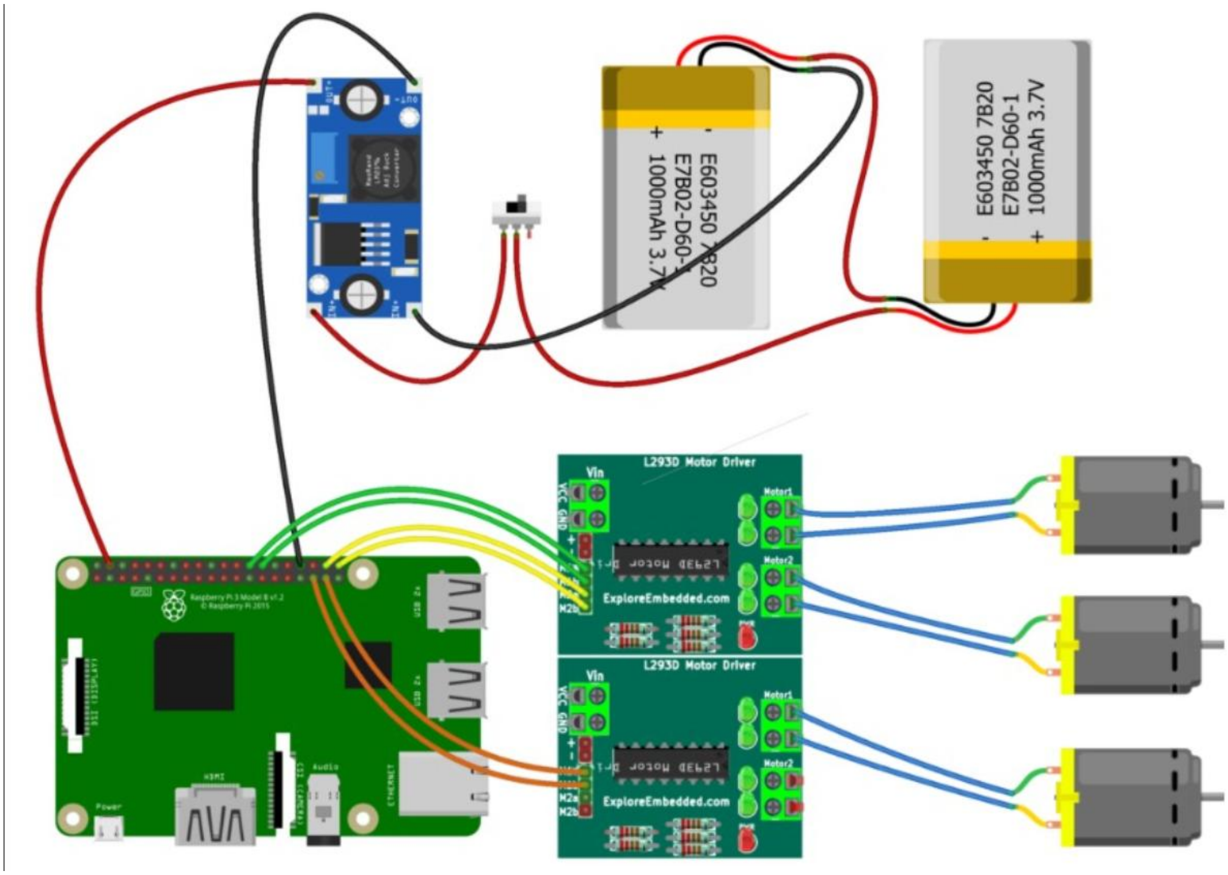

Figure 6.1.2 : Circuit Diagram

The figure 6.1.2 illustrates the connections in the circuit for the robot.

6.2. Output

```
import firebase_admin
from firebase_admin import credentials, storage
from firebase_admin import firestore
import dht11
import RPi.GPIO as GPIO
from time import sleep
from gpiozero import DistanceSensor
import busio
import digitalio
import board
```

```
import adafruit_mcp3xxx.mcp3008 as MCP
from adafruit_mcp3xxx.analog_in import AnalogIn
import geocoder
from picamera import PiCamera
import datetime
from twilio.rest import Client


############# Setup Starts #############

# GPIO Mode (BOARD / BCM)
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

# Set GPIO Pins
PWMA = 4 #7
AIN1 = 18 #12
AIN2 = 17 #11
PWMB = 24 #18
BIN1 = 22 #15
BIN2 = 23 #16
LED_PIN = 27
GPIO.setup(PWMA, GPIO.OUT)
GPIO.setup(AIN1, GPIO.OUT)
GPIO.setup(AIN2, GPIO.OUT)
GPIO.setup(PWMB, GPIO.OUT)
GPIO.setup(BIN1, GPIO.OUT)
GPIO.setup(BIN2, GPIO.OUT)
GPIO.setup(LED_PIN, GPIO.OUT)

# Setup the DHT11 Sensor
DHT11PIN = 21
```

```
instance = dht11.DHT11(pin=DHT11PIN) # temperature & humidity

# Setup the Distance Sensor
sensor = DistanceSensor(26,20) # Distance sensor

# Setup the MQ5 Sensor
spi = busio.SPI(clock=board.SCK, MISO=board.MISO, MOSI=board.MOSI) # create the spi bus
cs = digitalio.DigitalInOut(board.D5) # create the cs (chip select)
mcp = MCP.MCP3008(spi, cs) # create the mcp object
chan = AnalogIn(mcp, MCP.P0) # create an analog input channel on pin 0

# Connect to the Firebase
bucket_name = "gas-leakage-f87ff.appspot.com"
cred = credentials.Certificate('serviceAccountKey.json')
firebase_admin.initialize_app(cred, {'storageBucket': bucket_name})

# Create a Cloud Firestore client
database = firestore.client()
new_collection = database.collection("users")
document = new_collection.document('pi')

# Twilio Setup
account_sid = ''
auth_token = ''
client = Client(account_sid, auth_token)

iterations = 0

############# Setup Complete #############

# Turn off all the motors
def turnOffMotors():
```

```
    GPIO.output(AIN1, GPIO.LOW)
    GPIO.output(AIN2, GPIO.LOW)
    GPIO.output(PWMA, GPIO.LOW)
    GPIO.output(BIN2, GPIO.LOW)
    GPIO.output(BIN1, GPIO.LOW)
    GPIO.output(PWMB, GPIO.LOW)
    print("Motors are off")
    return

# Forward motion
def forward():
    GPIO.output(AIN1, GPIO.LOW)
    GPIO.output(AIN2, GPIO.HIGH)
    GPIO.output(PWMA, GPIO.HIGH)
    GPIO.output(BIN2, GPIO.LOW)
    GPIO.output(BIN1, GPIO.HIGH)
    GPIO.output(PWMB, GPIO.HIGH)
    print("Moving forward")
    return

# Left motion
def left():
    GPIO.output(AIN1, GPIO.LOW)
    GPIO.output(AIN2, GPIO.HIGH)
    GPIO.output(PWMA, GPIO.HIGH)
    GPIO.output(BIN2, GPIO.HIGH)
    GPIO.output(BIN1, GPIO.LOW)
    GPIO.output(PWMB, GPIO.HIGH)
    print("Moving left")
    return

# Right motion
```

```
def right():
    GPIO.output(AIN1, GPIO.HIGH)
    GPIO.output(AIN2, GPIO.LOW)
    GPIO.output(PWMA, GPIO.HIGH)
    GPIO.output(BIN2, GPIO.LOW)
    GPIO.output(BIN1, GPIO.HIGH)
    GPIO.output(PWMB, GPIO.HIGH)
    print("Moving right")
    return

# Backward motion
def backward():
    GPIO.output(AIN1, GPIO.HIGH)
    GPIO.output(AIN2, GPIO.LOW)
    GPIO.output(PWMA, GPIO.HIGH)
    GPIO.output(BIN2, GPIO.HIGH)
    GPIO.output(BIN1, GPIO.LOW)
    GPIO.output(PWMB, GPIO.HIGH)
    print("Moving backward")
    return

# Stop motion
def stop():
    GPIO.output(AIN1, GPIO.LOW)
    GPIO.output(AIN2, GPIO.LOW)
    GPIO.output(PWMA, GPIO.LOW)
    GPIO.output(BIN2, GPIO.LOW)
    GPIO.output(BIN1, GPIO.LOW)
    GPIO.output(PWMB, GPIO.LOW)
    print("Stop")
    return
```

```
# Read the temperature and humidity
def readHumidityTemperature():
    result = instance.read()
    if result.is_valid():
        print("Temp: %d C" % result.temperature +' '+"Humid: %d %%" % result.humidity)
        # setHumidityTemperature(result)
        return result

# Read the distance
def readDistance():
    distance = sensor.distance
    print("Distance: %.1f cm" % (distance * 100))
    if distance < 0.06:
        stop()
    return distance

# Read the Gas Quality
def readGasQuality():
    gas = str(chan.voltage)
    print("ADC Voltage: " + gas + "V")
    return gas

# Send SMS
def sendSMS():
    message = client.messages.create(
    from_='+12184026613',
    to='+919820077642',
    body= "GAS Leakage Detected",
    )

    print(message.sid)
```

```
    print("Sending SMS using Twilio")

# Click Image
def clickImage():
    print("Clicking Image")
    # Initialize camera
    camera = PiCamera()
    camera.resolution = (1024, 768)

    # Generate image path
    image_path = f'images/image_{datetime.datetime.now().isoformat()}.jpg'

    # Capture image
    camera.start_preview()
    sleep(2)  # Camera warm-up time
    camera.capture(image_path)
    camera.stop_preview()
    camera.close()

    # Upload to Firebase Storage
    bucket = storage.bucket()
    blob = bucket.blob(image_path)
    blob.upload_from_filename(image_path)
    print(f'Image uploaded to {image_path}')

    # # Generate a URL to access the image
    # url = blob.generate_signed_url(datetime.timedelta(seconds=300), method='GET')  # URL valid for 5 minutes
    # print(f'Access URL: {url}\n')
```

```
    public_url =
f'https://firebasestorage.googleapis.com/v0/b/{bucket_name}/o/{image_path.replace("/",
"%2F")}?alt=media'

    return public_url

# Get all values from pi
def getAllData():
    print("getAlldata Called")
    data = document.get().to_dict()
    print("Completed | getAllData")
    print(data)
    return data

def readAndSetAllData(data):
    try:
        dht = readHumidityTemperature()
        data["hum"], data["temp"] = str(dht.humidity), str(dht.temperature)
    except Exception as e:
        print(f"Error reading humidity and temperature: {e}")

    try:
        data["dist"] = str(readDistance())
    except Exception as e:
        print(f"Error reading distance: {e}")

    try:
        data["leakage"] = str(readGasQuality())
    except Exception as e:
        print(f"Error reading gas quality: {e}")

    if iterations % 5 == 0:
```

```
        try:
            geoLocation = geocoder.ip('me')
            data["lat"] = str(geoLocation.latlng[0])
            data["long"] = str(geoLocation.latlng[1])
        except Exception as e:
            print(f"Error obtaining geolocation: {e}")

    try:
        document.set(data)
    except Exception as e:
        print(f"Error setting document data: {e}")

    return data

# Main Loop
while True:
    print("\nMain Loop Started")

    data = getAllData()

    if data['forward'] == True:
        forward()
    elif data['left'] == True:
        left()
    elif data['right'] == True:
        right()
    elif data['backward'] == True:
        backward()
    else:
        stop()

    if data["torch"] == True:
```

```
        GPIO.output(LED_PIN, GPIO.HIGH)
    else:
        GPIO.output(LED_PIN, GPIO.LOW)

    if iterations % 4 == 0 :
        if data["sms_enabled"] == True and float(data["leakage"]) > 1.5:
            try:
                sendSMS()
                data["sms_enabled"] = False
            except Exception as e:
                print("Error sending SMS: ", e)

        if data["click_image"] == True:
            try:
                imageUrl = clickImage()
                data["view_images"].append(imageUrl)
                data["click_image"] = False
            except Exception as e:
                print("Error clicking image: ", e)
        print(readAndSetAllData(data))

    iterations += 1
    print("Main Loop Ended:",iterations, end="\n")
    sleep(1)
```

6.3 CAD Diagram

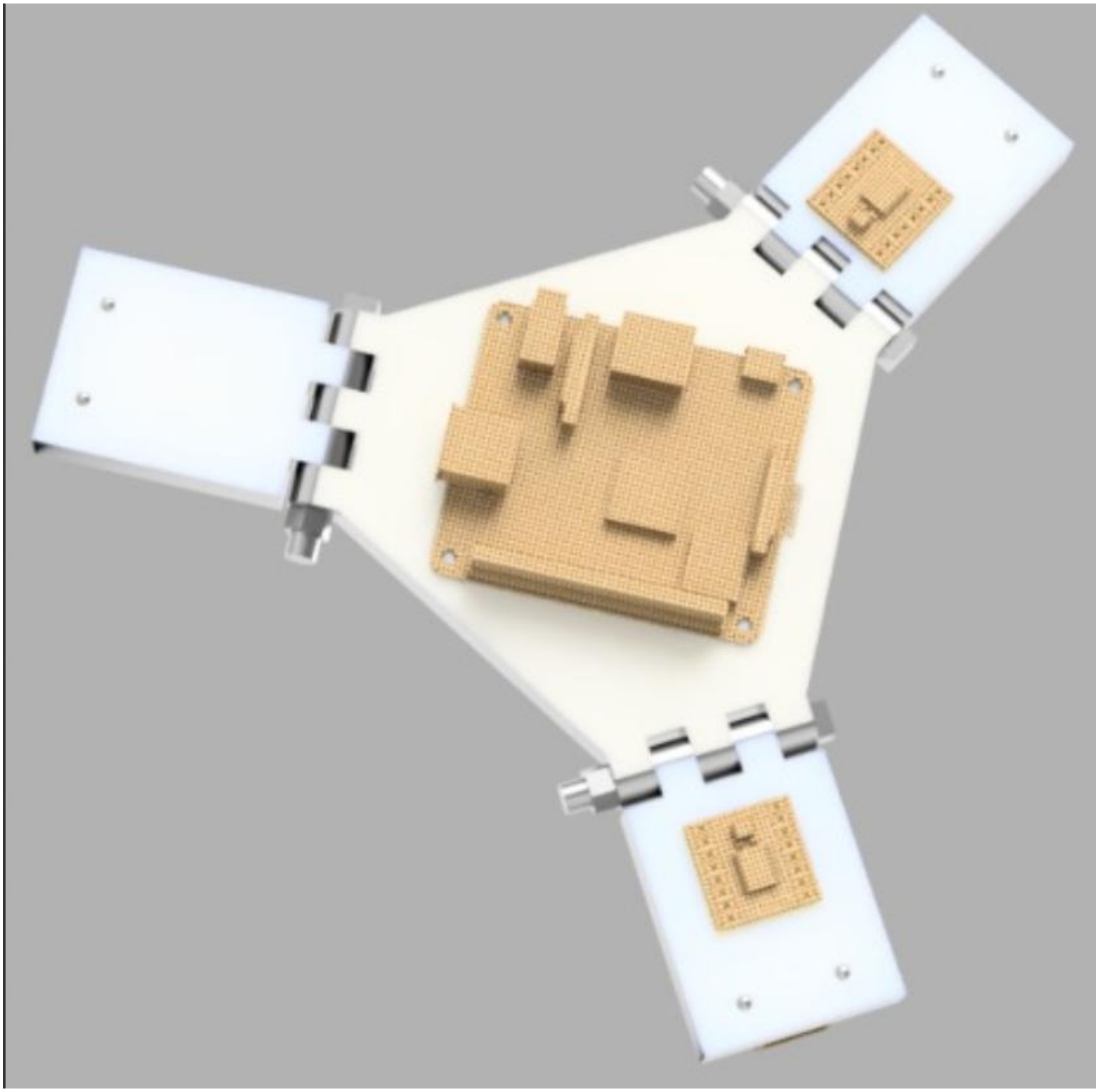

FIgure 6.3.1 : CAD Diagram

This figure 6.3.1 represents the design of the robot.

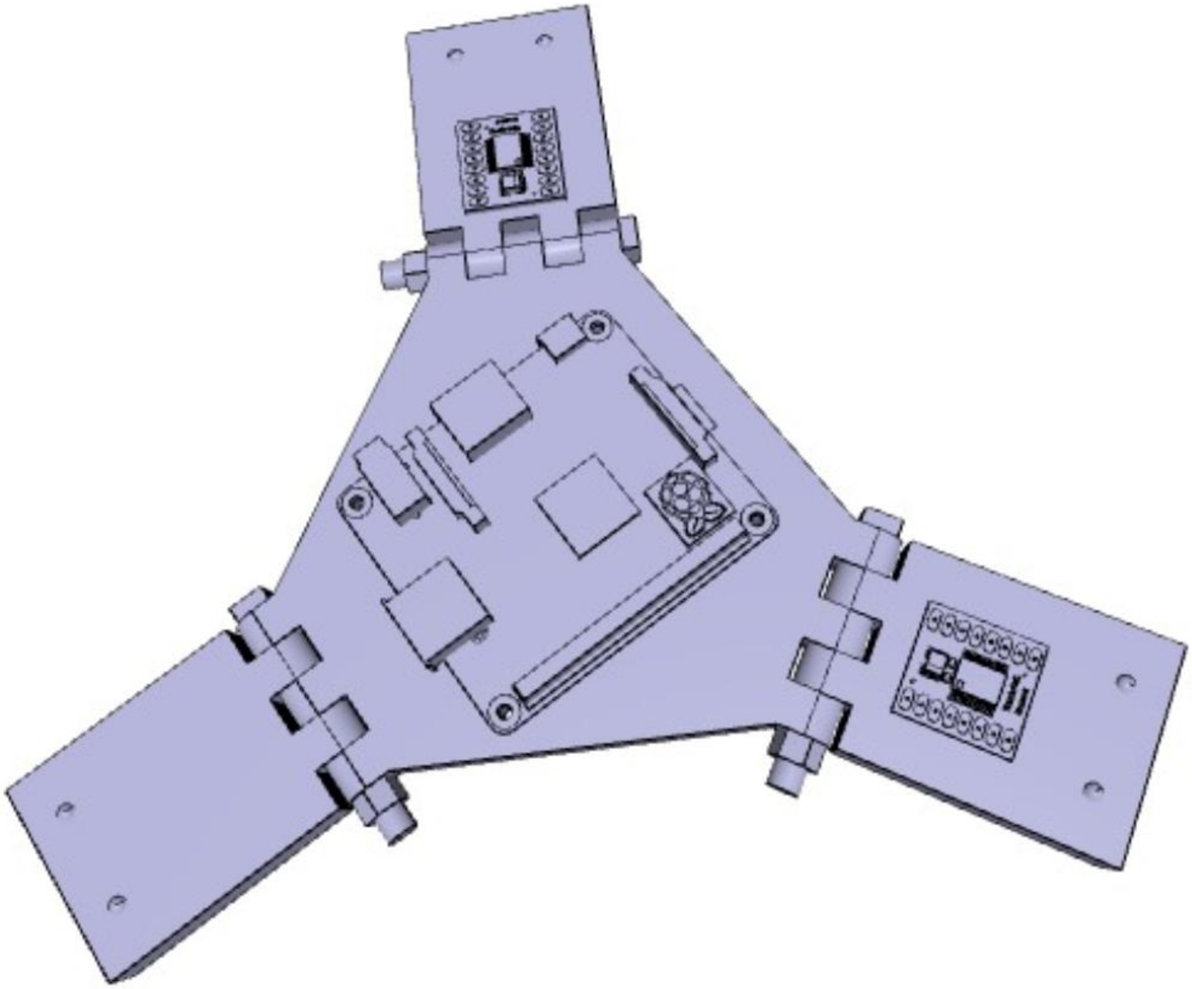

Figure 6.3.2 : 3D Print Design

This figure 6.3.2 represents the design that has to be 3d printed

Reference link : https://www.emachineshop.com/free-online-step-file-viewer/?share=ySvC7hS

7. CONCLUSION

This project aims to develop a versatile and robust robot capable of addressing critical industrial challenges, such as pipe inspection and leakage detection, particularly in gas and chemical industries. By integrating an omnidirectional wheel, two normal wheels, and magnets for propulsion, the robot achieves superior mobility, enabling it to climb pipes, navigate horizontal surfaces, overcome pipe joints, and traverse turns. The use of Raspberry Pi and a camera ensures real-time monitoring and data analysis, while distance sensors enhance obstacle detection for seamless navigation. Servo motors drive the wheels for efficient movement across various surfaces, making the robot adaptable to diverse environments. This innovative solution is designed to minimize human intervention in hazardous conditions, ensuring enhanced safety, efficiency, and reliability in emergency scenarios.

8. REFERENCES

1. X. Zhang, et al., "Magnetic-Wheeled Climbing Robots for Non-Destructive Testing," *Journal of Robotics and Automation*, 2022.
2. J. Smith, et al., "Flexible Pipe Navigation in Robotics," *International Journal of Advanced Robotics Systems*, 2023.
3. A. Kumar, et al., "Gas Leak Detection Using Multi-Sensor Robotic Systems," *Journal of Sensors and Actuators B: Chemical*, 2021.
4. N. Patel, et al., "Autonomous Firefighting Robots with Sensor Fusion," *Robotics and Autonomous Systems*, 2023.
5. R. Diaz, et al., "Snake Robots for Confined Space Navigation," *IEEE Transactions on Robotics and Automation Letters*, 2021.
6. F. Wilson, et al., "Robots for Petrochemical Industry Inspections," *Journal of Industrial Robotics*, 2022.
7. M. Lee, et al., "Advanced Fire Detection and Suppression Robots," *Fire Safety Journal*, 2022.
8. D. Li, et al., "Wireless Sensor Networks for Industrial Safety," *International Journal of Industrial Engineering*, 2022.
9. A. Brown, et al., "Real-Time Data Transmission in Hazardous Environments," *Journal of Industrial Technology and Engineering*, 2021.
10. Y. Tanaka, et al., "Firefighting Robots with Water Jet Propulsion," *International Journal of Robotics and Automation*, 2023.
11. H. Wang, et al., "Hybrid Locomotion Systems for Robotics in Complex Terrains," Robotics and Biomimetics, 2023.
12. G. Silva, et al., "Obstacle Avoidance Strategies for Pipeline Robots," Journal of Intelligent and Robotic Systems, 2022.
13. K. Yamada, et al., "Structural Health Monitoring Using Climbing Robots," Sensors and Materials, 2023.
14. P. Johnson, et al., "Magnetically Adhesive Robots for Vertical Surfaces," IEEE Robotics and Automation Magazine, 2021.
15. L. Chen, et al., "Multi-Modal Sensing in Hazardous Environment Robots," Robotics and Autonomous Systems, 2022.

16. E. Garcia, et al., "Energy Optimization Techniques for Industrial Robots," Journal of Industrial and Engineering Chemistry, 2023.
17. S. Gupta, et al., "AI-Driven Navigation Algorithms for Autonomous Robots," IEEE Transactions on Artificial Intelligence, 2023.
18. T. Nguyen, et al., "Pipeline Inspection Robots with Adaptive Clamping Mechanisms," Journal of Mechanical Systems and Signal Processing, 2022.
19. J. Alvarez, et al., "Thermal Imaging in Firefighting Robots," International Journal of Thermal Sciences, 2021.
20. D. Park, et al., "Innovative Designs in Gas Detection Robots," Sensors and Actuators A: Physical, 2023.