Semester-Long Internship 2026

On

Industrial Robotics & Autonomous Navigation.

Submitted by

Anuj Naik

B.Tech Student

Fr. Conceicao Rodrigues College of Engineering

Under the Guidance of

Prof. Jayendran Venkateswaran

Department of Industrial Engineering and Operations Research (IEOR)

Indian Institute of Technology Bombay

Declaration

I, **Anuj Naik**, student of B.Tech at Fr. Conceicao Rodrigues College of Engineering, hereby declare that this internship report titled *"Industrial Robotics, Autonomous Navigation, and Pipeline Inspection Systems"* is a genuine record of work carried out by me during the FOSSEE Semester Internship at the IEOR Department, Indian Institute of Technology Bombay, from February 2026 to April 2026.

The work presented in this report has not been submitted to any other institution or university for the award of any degree or diploma. All sources of information have been duly acknowledged in the bibliography.

Anuj Naik
B.Tech Student
Fr. C.R.C.E, Mumbai

Prof. Jayendran Venkateswaran
Department of IEOR
IIT Bombay

Acknowledgment

I would like to express my sincere gratitude to everyone who supported and guided me throughout the course of this semester-long internship and the successful completion of all projects undertaken. Their encouragement, mentorship, and guidance were instrumental in shaping my learning experience and technical growth.

I express my sincere thanks to **Prof. Jayendran Venkateswaran**, Department of Industrial Engineering and Operations Research (IEOR), IIT Bombay, for his continuous supervision, invaluable technical guidance, and encouragement throughout every phase of the internship. His insights helped direct the projects toward meaningful and practically relevant outcomes.

I am deeply grateful to the mentors from the IEOR Department, including the M.Tech students, for their patient technical support, constructive feedback, and willingness to assist during challenging phases of project development. Their experience was indispensable in navigating complex technical problems.

I extend my heartfelt gratitude to **Prof. Kannan M. Moudgalya**, Principal Investigator of the FOSSEE project, Department of Chemical Engineering, IIT Bombay, for making this internship opportunity possible and for his ongoing commitment to promoting open-source education in India.

I am especially thankful to my FOSSEE guides, **Mr. Sumanto Kar** and **Mr. Varad Patil**, for efficiently coordinating all administrative and operational aspects of the internship, and for ensuring smooth and consistent communication throughout its duration.

I also thank the FOSSEE Managers **Ms. Usha Viswanathan** and **Ms. Vineeta Parmar**, along with their entire team, for their consistent support and for maintaining the high standard of the FOSSEE internship programme.

A special thanks to the organizing team of **IEOR Day 2026** for the opportunity to present our work to a distinguished panel of industry professionals and for creating a platform that bridges academic projects with real-world industrial relevance.

I gratefully acknowledge the support of the **National Mission on Education through**

Information and Communication Technology (NMEICT), Ministry of Education (MoE), Government of India, for supporting the FOSSEE initiative and facilitating projects such as this.

Finally, I sincerely thank **Fr. Conceicao Rodrigues College of Engineering** and its Principal for providing a supportive institutional environment and for encouraging students to pursue advanced technical internship opportunities.

Abstract

This report documents the work carried out during the FOSSEE Semester Internship at the Industrial Engineering and Operations Research (IEOR) Department, Indian Institute of Technology Bombay, from February 2026 to April 2026.

The internship comprised two major technical projects and participation in a department-level exhibition event. The first project, the **Pipeline Inspection Rover for Blockage Detection**, involved developing a smart two-tier inspection system integrating water flow, ultrasonic level, temperature, and smoke sensors into a central monitoring station. The monitoring system provides real-time data visualization, early blockage prediction, and approximate anomaly localization. A camera-equipped rover is deployed to the flagged location, streaming a live video feed that is processed in real time to detect and annotate blockages visually. A responsive real-time dashboard ties the entire system together.

The second project, **Autonomous Navigation in a Mapped Environment**, focused on developing a line-free AGV system capable of navigating a pre-mapped environment using a binary occupancy grid and the A* path planning algorithm. The robot integrates ultrasonic sensing for real-time obstacle detection and dynamically re-routes its path when unexpected obstacles are encountered, removing dependence on physical guide infrastructure.

Both projects were presented at **IEOR Day 2026**, the department's annual exhibition event, in front of industry experts and academic faculty.

Keywords: Pipeline inspection, sensor fusion, real-time monitoring, autonomous navigation, A* algorithm, occupancy mapping, obstacle avoidance, AGV, Industry 4.0, FOSSEE, IIT Bombay.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 National Mission on Education through ICT

The National Mission on Education through ICT (NMEICT) is a flagship scheme under the Department of Higher Education, Ministry of Education, Government of India. The mission was conceived to leverage the transformative potential of Information and Communication Technology (ICT) to enhance the quality, reach, and effectiveness of education in Higher Education Institutions (HEIs) across the country, operating on the principle of anytime–anywhere learning.

NMEICT aligns with the three cardinal principles of the National Education Policy: **access**, **equity**, and **quality**. By creating and disseminating digital educational content, tools, and platforms, it seeks to democratise access to high-quality education regardless of geographic location or socioeconomic background.

The mission specifically targets bridging the digital divide between urban and rural educational institutions, empowering learners and educators at all levels. It promotes inclusive participation in the knowledge economy by equipping students and faculty with modern digital tools and resources.

1.1.1 ICT Initiatives of the Ministry of Education

Under the NMEICT umbrella, the Ministry of Education has launched several impactful ICT initiatives targeting different aspects of higher education in India.

Table 1.1: Summary of ICT Initiatives by the Ministry of Education

No.	Resource	For Students / Researchers	For Institutions
1	SWAYAM	Earn credit via online courses	Develop and host courses
2	National Digital Library	Access e-content	Form NDL Clubs
3	e-PG Pathshala	Access free books	Host e-books
4	Shodhganga	Access research theses	List institutional theses
5	e-Yantra	Embedded systems training	Create e-Yantra labs
6	FOSSEE	Volunteer for open-source	Run FLOSS labs
7	Virtual Labs	Perform online experiments	Develop experiments
8	SAMARTH ERP	Manage lifecycle digitally	Enable e-governance

1.2 FOSSEE Project

The FOSSEE (Free/Libre and Open Source Software for Education) project is based at IIT Bombay and operates under the NMEICT. Its primary goal is to reduce the dependence of Indian academic institutions on expensive proprietary software by promoting the adoption of high-quality Free/Libre and Open Source Software (FLOSS) alternatives across engineering, science, and technology disciplines.

FOSSEE achieves this through a multifaceted approach:

- Developing and maintaining open-source software tools suitable for academic and research use.
- Creating educational content, tutorials, and textbook companions for FLOSS tools.
- Running internship programmes that train students in open-source technologies through hands-on project work.
- Collaborating with institutions across India to establish FLOSS-equipped laboratories.

The FOSSEE internship programme provides students with the opportunity to work on real engineering and research problems at IIT Bombay, supervised by faculty and graduate students, gaining exposure to both technical and research methodologies.

1.3 Industrial Engineering and Operations Research (IEOR) Department

The Department of Industrial Engineering and Operations Research at IIT Bombay is one of the leading departments in India for research and education in manufacturing systems, supply chain management, operations research, and industrial automation.

The IEOR laboratory is equipped with modern robotics hardware including Dobot robotic arms, autonomous vehicle platforms, sensor arrays, and embedded computing systems. The lab provides an environment where students can design, implement, and test complete automation systems under the mentorship of experienced faculty and researchers.

1.4 About the Internship

This semester-long internship was conducted at the IEOR Department, IIT Bombay from **February 2026 to April 2026**. Over the course of the internship, two major technical projects were conceptualized, designed, implemented, and demonstrated:

1. **Pipeline Inspection Rover for Blockage Detection:** A smart two-tier inspection system combining a fixed multi-sensor monitoring station with a deployable camera-equipped rover, supported by a real-time dashboard with early prediction and location estimation capabilities.
2. **Autonomous Navigation in a Mapped Environment:** A line-free AGV system using binary occupancy grid mapping, the A* shortest-path algorithm, and ultrasonic-based real-time obstacle detection and dynamic re-routing.

Additionally, both projects were exhibited and presented at **IEOR Day 2026**, the department's annual event attended by industry professionals, faculty, and researchers.

1.5 Objectives of the Internship

The internship was guided by the following objectives:

- To gain practical experience in designing and implementing complete, end-to-end intelligent automation systems.
- To develop proficiency in sensor integration, real-time data processing, and embedded systems communication.
- To apply algorithmic techniques (such as A* path planning) to real robotic navigation problems.
- To understand the principles of Industry 4.0 through hands-on project implementation.
- To develop technical communication skills by presenting projects to industry experts and academic faculty.
- To contribute meaningfully to the FOSSEE initiative and open-source engineering education.

1.6 Organization of the Report

This report is organized as follows:

- **Chapter 2** presents the Pipeline Inspection Rover project in comprehensive detail, covering system architecture, sensor integration, monitoring dashboard, rover design, live feed processing, results, and future scope.
- **Chapter 3** describes the Autonomous Navigation system, covering occupancy mapping, A* path planning, obstacle detection and avoidance, system architecture, results, and future scope.
- **Chapter 4** covers the IEOR Day 2026 event, including organizational contributions, project exhibitions, and learnings from industry interaction.
- **Chapter 5** summarizes the learning outcomes of the internship.
- **Chapter 6** concludes the report.

Chapter 2

Pipeline Inspection Rover for Blockage Detection

2.1 Introduction and Motivation

Pipeline infrastructure is one of the most critical and extensively used systems in modern industrial operations. Pipelines transport water, chemicals, petroleum products, gases, and slurry across vast distances within industrial plants, municipal water systems, oil refineries, and manufacturing facilities. The integrity and unobstructed flow within these pipelines are fundamental to operational efficiency and safety.

Blockages, leaks, and structural degradation in pipelines are common problems that lead to significant economic losses, environmental hazards, and safety risks. Traditional pipeline inspection methods rely heavily on manual intervention – sending workers with equipment into confined or hazardous spaces, scheduling planned shutdowns, or relying on periodic mechanical probing. These approaches are slow, resource-intensive, and often reactive rather than preventive.

The motivation behind this project was to develop a smart, proactive, and semi-autonomous pipeline inspection system that:

- Continuously monitors pipeline health parameters through a fixed sensor array.
- Applies data analysis to predict blockages before they become critical.
- Estimates the approximate location of anomalies automatically.
- Deploys a rover with a live camera for visual confirmation and detailed inspection of flagged areas.

- Presents all data and alerts through a clean, real-time, responsive dashboard.

This approach transforms pipeline inspection from a reactive, manual process into a proactive, data-driven, and semi-autonomous operation.

2.2 System Architecture

The pipeline inspection system was designed around a **two-tier architecture**, combining a stationary monitoring layer with a deployable inspection layer. Together, these tiers provide comprehensive pipeline health management capability.

2.2.1 Tier 1 – Stationary Monitoring Layer

The monitoring layer consists of a fixed array of sensors installed at strategic locations along the simulated pipeline. This layer is always active, continuously sampling environmental and flow parameters. Data from all sensors is transmitted to the central control computer, where the monitoring software processes it, applies the early prediction engine, estimates anomaly locations, and displays everything on the dashboard.

Key responsibilities of the monitoring layer:

- Continuous multi-parameter data acquisition.
- Real-time threshold monitoring and alert generation.
- Early blockage prediction through trend analysis.
- Approximate fault location estimation.
- Data logging for historical analysis.

2.2.2 Tier 2 – Deployable Inspection Layer

The inspection layer is activated when the monitoring layer identifies a potential blockage or critical anomaly. A rover equipped with a front-mounted camera is deployed to the approximate flagged location. The rover navigates to the area and streams live video back to the dashboard, where image processing algorithms analyze the feed in real time to detect and annotate visible blockages or pipeline damage.

Key responsibilities of the inspection layer:

- Physical navigation to flagged pipeline segments.
- Live video streaming to the monitoring dashboard.
- Real-time image processing for visual blockage detection.
- Visual confirmation of monitoring layer predictions.

2.3 Sensor Integration and Data Acquisition

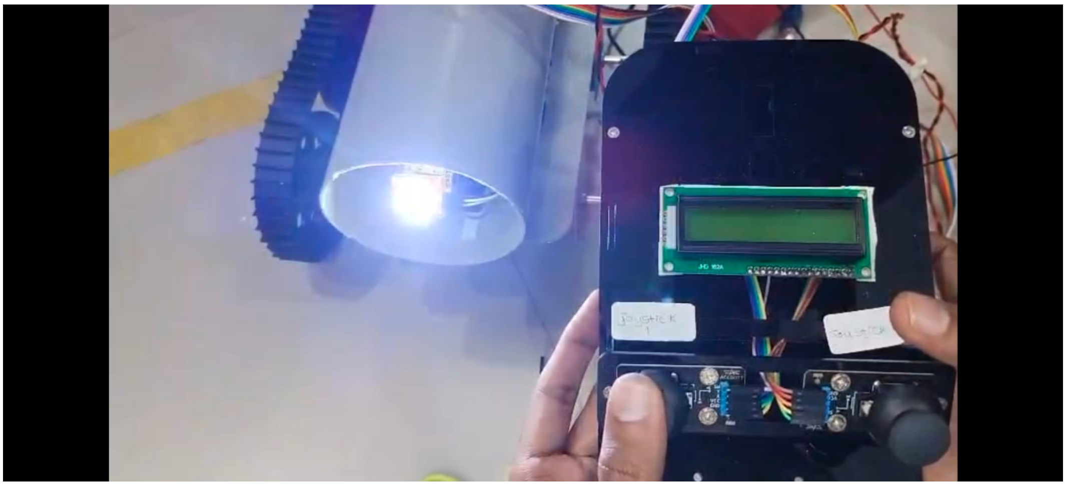

Figure 2.2: Sensor Array Installation: Water Flow, Ultrasonic, Temperature, and Smoke Sensors

Four categories of sensors were integrated into the monitoring station to capture a comprehensive picture of pipeline health.

2.3.1 Water Flow Sensor

The water flow sensor was positioned within the pipeline to measure the volumetric flow rate of liquid passing through it. A Hall-effect flow sensor was used, generating pulse signals proportional to the liquid flow rate. The microcontroller counted pulses over a timed interval and calculated flow rate in litres per minute.

A sustained, significant decrease in flow rate below the baseline is a primary indicator of a developing blockage. The system continuously compared the live flow rate against the established baseline and flagged deviations exceeding the defined threshold.

Parameters monitored:

- Instantaneous flow rate (litres per minute).
- Running average flow rate over a configurable time window.
- Percentage deviation from baseline.

2.3.2 Ultrasonic Sensor for Water Level

An ultrasonic distance sensor (HC-SR04) was mounted above the pipeline channel to measure the water level non-invasively. The sensor emits ultrasonic pulses and calculates water surface distance based on echo return time.

Abnormal water levels – whether excessively high (indicating a downstream blockage causing backflow) or low (indicating upstream supply failure) – trigger corresponding alerts. The relationship between flow rate and water level provides rich contextual information for blockage characterization.

Parameters monitored:

- Water level in centimetres from sensor reference.
- Rate of level change (rising or falling trend).
- Level status relative to normal operating band.

2.3.3 Temperature Sensor

A digital temperature sensor (DS18B20) monitored the temperature of the medium flowing through the pipeline. Anomalous temperature readings can indicate:

- Heat buildup due to pressure accumulation upstream of a blockage.
- Chemical reactions caused by material build-up within the pipe.
- Leakage of warmer or cooler external fluids into the pipeline.

The sensor provided readings with ±0.5°C accuracy, ensuring reliable detection of even gradual thermal anomalies.

2.3.4 Smoke and Gas Sensor

A smoke and gas sensor (MQ-2 type) was installed near the pipeline to detect the presence of smoke, combustible gases, or hazardous vapours in the surrounding environment. This sensor addresses safety scenarios where pipeline damage or leakage leads to gas emission or fire risk.

Smoke or gas detection triggers a high-priority critical alert regardless of other sensor readings, as it represents an immediate safety concern.

2.3.5 Microcontroller and Data Transmission

All four sensors were interfaced with an Arduino microcontroller, which performed:

- Analog-to-digital conversion for sensor readings.
- Pulse counting for the flow sensor.
- Distance calculation for the ultrasonic sensor.
- Serial communication of all readings to the host computer at a defined sampling rate.

Data packets transmitted over the serial port included a timestamp, sensor ID, and reading value, enabling the monitoring software to correctly parse and attribute each data point.

2.4 Monitoring System and Real-Time Dashboard

Figure 2.3: Real-Time Monitoring Dashboard Displaying Sensor Data, Alerts, and Predictions

The monitoring dashboard was the central interface of the pipeline inspection system. It was designed and implemented as a real-time, responsive application that consolidates all sensor data, prediction outputs, and alerts into a single, operator-friendly interface.

2.4.1 Dashboard Design Philosophy

The dashboard was designed around the following principles:

- **Clarity:** Each parameter is displayed in its own dedicated widget with appropriate units and visual indicators. Operators can assess system health at a glance.
- **Responsiveness:** Sensor data updates in real time without requiring manual refresh.
- **Actionability:** Alerts are prominently displayed and accompanied by recommended actions where applicable.
- **Historical Context:** Time-series charts display the recent history of each sensor reading, enabling operators to observe trends rather than only instantaneous values.

2.4.2 Sensor Data Visualization

Each sensor was allocated a dedicated widget on the dashboard:

- **Flow Rate Widget:** Displays current flow rate as a numeric value and as a gauge indicator. A time-series graph shows flow rate over the past 10 minutes.
- **Water Level Widget:** Shows current water level with a visual bar indicator representing low, normal, and high zones.
- **Temperature Widget:** Displays current temperature and its deviation from baseline. A trend line shows temperature changes over time.
- **Smoke/Gas Widget:** Displays sensor output level and a binary "Safe / Alert" status indicator.

Color-coded status indicators (green = normal, yellow = warning, red = critical) were used consistently across all widgets to provide immediate visual feedback.

2.4.3 Alert and Notification System

The alert system operated across three severity levels:

Table 2.1: Alert Severity Levels and Trigger Conditions

Level	Trigger Condition	System Response
Normal	All readings within baseline	No action required
Warning	Reading approaching threshold (within 15%)	Visual highlight, log entry
Critical Alert	Reading exceeds threshold	Audio alert, pop-up notification, rover deployment prompt

2.4.4 Early Prediction Engine

The early prediction engine was a core component of the monitoring system. Unlike simple threshold monitoring, which only triggers when a problem has already reached a critical level, the prediction engine analyzed patterns across multiple sensors to forecast developing blockages *before* they became critical.

The engine implemented the following logic:

1. **Trend Analysis:** Each sensor reading was tracked over a sliding time window. A consistent downward trend in flow rate, combined with a rising water level, was flagged as a developing blockage pattern.
2. **Multi-Sensor Correlation:** The engine correlated readings from all four sensors simultaneously. For example, a simultaneous decrease in flow rate, increase in water level, and rise in temperature together formed a high-confidence blockage prediction.
3. **Prediction Confidence Score:** Each prediction was assigned a confidence score based on the strength and consistency of the observed pattern. Scores above a defined threshold triggered a formal early warning alert.

This approach transformed the system from a reactive alarm system into a proactive predictive maintenance tool.

2.4.5 Approximate Location Estimation

To assist operators in quickly identifying *where* within the pipeline the problem is likely developing, the system implemented an approximate location estimation algorithm.

Each sensor in the array was mapped to a specific segment of the pipeline in the system's configuration. When anomalous readings were detected by sensors at specific positions, the system highlighted the corresponding pipeline segments on a schematic diagram displayed on the dashboard.

For example, if the flow sensor at Segment 3 reported reduced flow while the ultrasonic level sensor at Segment 2 reported elevated levels, the system inferred a blockage in the region between Segments 2 and 3. This estimate guided the rover deployment decision.

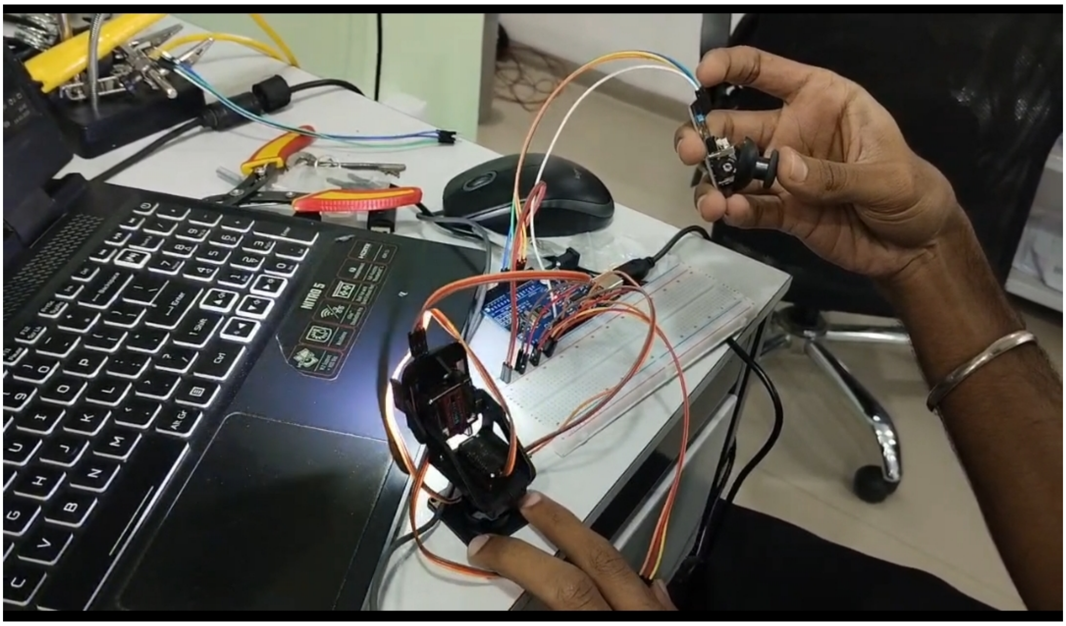

Figure 2.4: Early Prediction Display and Approximate Blockage Location Highlighting on Dashboard

2.5 Rover Design and Deployment

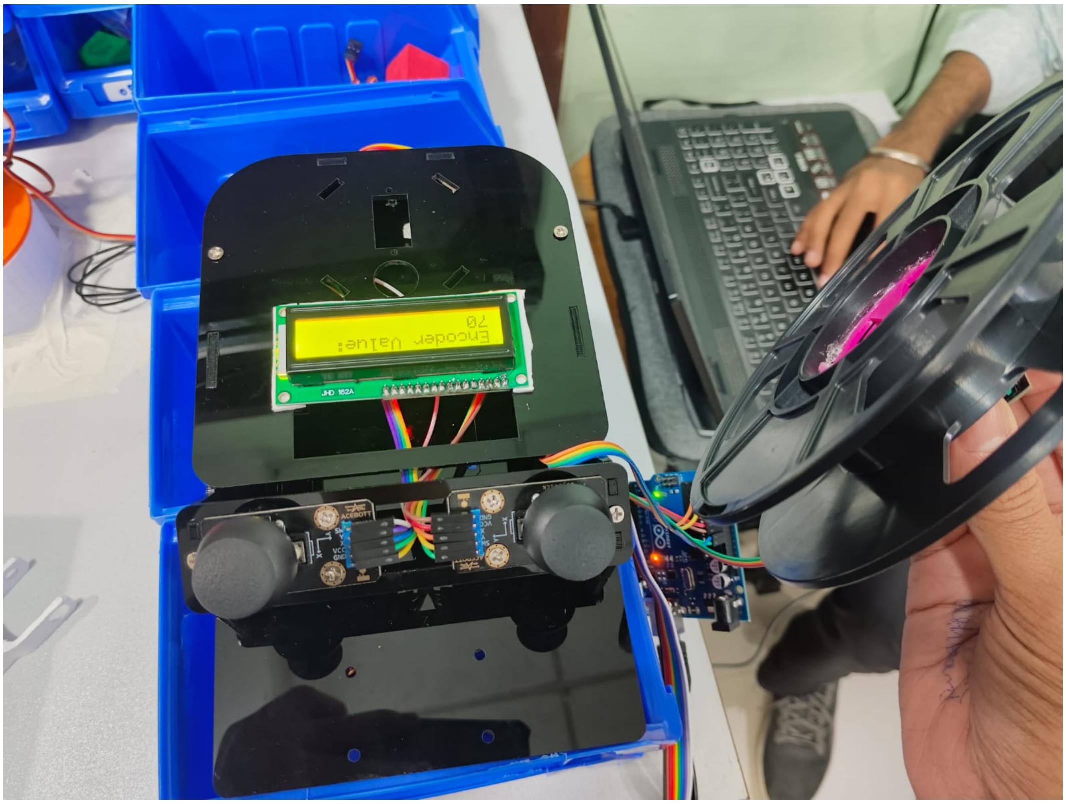

Figure 2.5: Pipeline Inspection Rover with Front-Mounted Camera and Drive System

2.5.1 Mechanical Design and Construction

The inspection rover was designed for compact, stable locomotion within the simulated pipeline environment. Key design considerations included:

- **Low Profile:** A low chassis height to facilitate movement in confined or narrow pipeline-like spaces.
- **Stable Drive:** A four-wheel differential drive configuration providing reliable traction and smooth directional control.
- **Modular Mounting:** A front-facing camera mount designed for easy adjustment of camera angle to optimize the field of view inside the pipeline.
- **Cable Management:** Careful routing of power and communication cables to prevent tangling during rover movement.

2.5.2 Camera Module Integration

A wide-angle USB camera module was mounted on the front face of the rover. The camera captured a continuous video stream at a target resolution sufficient for reliable visual inspection. The stream was transmitted to the host computer over a USB/wireless link and displayed live on the monitoring dashboard.

The camera mounting was adjustable, allowing the field of view to be directed at the pipeline interior walls, floor, or ceiling as required.

2.5.3 Rover Control System

The rover was controlled from the dashboard interface. Operator commands (forward, backward, left, right, stop) were transmitted to the rover's onboard microcontroller via a wireless communication link. The microcontroller translated these commands into appropriate motor driver signals, controlling the drive wheels accordingly.

2.5.4 Deployment Protocol

The deployment workflow was as follows:

1. The monitoring dashboard identifies a critical or high-confidence warning alert.
2. The dashboard displays the approximate location estimate for the anomaly.
3. The operator initiates rover deployment from the dashboard interface.
4. The rover is guided to the flagged pipeline segment.
5. Live video feed begins streaming to the dashboard.
6. The operator inspects the feed with image processing annotations active.
7. Based on visual confirmation, the operator determines the appropriate maintenance response.

2.6 Live Feed Processing and Computer Vision

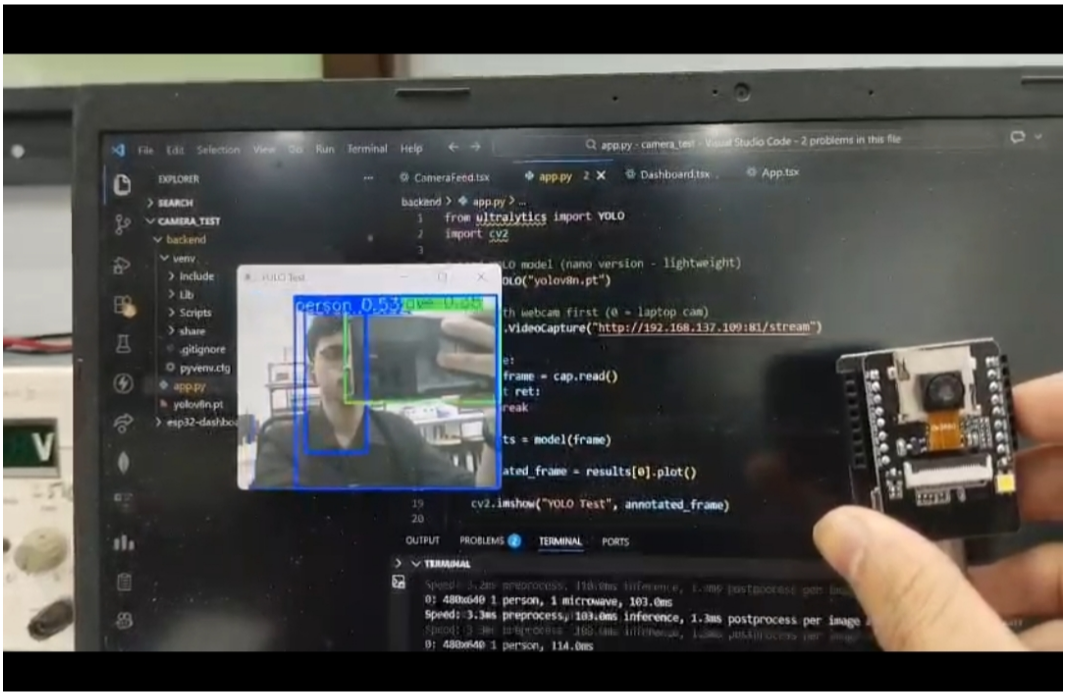

Figure 2.6: Live Camera Feed from Rover with Real-Time Blockage Detection Annotation Overlay

2.6.1 Image Processing Pipeline

The live video feed from the rover's camera was processed frame-by-frame in real time on the host computer. The image processing pipeline consisted of the following sequential stages:

1. **Frame Capture:** Continuous extraction of frames from the video stream at a consistent frame rate.
2. **Grayscale Conversion:** Conversion to grayscale to reduce computational load and simplify subsequent processing.
3. **Gaussian Blurring:** Application of a Gaussian blur to reduce noise and smooth the image before edge detection.
4. **Edge Detection:** Canny edge detection was applied to identify boundaries and structural features within the pipeline interior.
5. **Contour Analysis:** Contours were extracted and analyzed to identify regions of unusual shape, size, or density that may indicate blockage material or structural damage.

6. **Anomaly Classification:** Detected contours were classified as potential blockages, sediment accumulation, or normal pipeline features based on size and shape criteria.

7. **Annotation Overlay:** Regions classified as anomalies were annotated on the displayed frame with bounding rectangles and descriptive labels, providing immediate visual feedback to the operator.

2.6.2 Processing Performance

The image processing pipeline was implemented to balance detection accuracy with processing latency. Lightweight classical computer vision techniques were selected specifically to enable real-time performance within the available computational environment, maintaining a sufficient frame rate for smooth, actionable video display.

2.7 Results and Observations

The pipeline inspection system was tested across multiple scenarios, simulating different types and severities of pipeline conditions.

Table 2.2: Test Scenarios and System Response Summary

Test Scenario	Sensor Response	System Action
Partial blockage	Flow rate drop 30%, water level rise	Warning alert, prediction triggered
Full blockage	Flow rate drop >70%, high water level	Critical alert, rover deployed
Temperature anomaly	5°C above baseline	Warning alert, trend monitored
Gas/smoke detection	MQ-2 threshold exceeded	Critical alert, priority notification
No anomaly	All readings normal	Normal status, no alert

Key observations from testing:

- The early prediction engine successfully issued warnings **prior to** threshold breaches in all partial blockage scenarios, demonstrating genuine predictive capability.
- Location estimation correctly identified the affected pipeline segment in all tested scenarios.

- The live feed processing pipeline successfully detected and annotated simulated blockage materials placed within the test pipeline.
- The dashboard maintained reliable real-time data refresh throughout all tests.

2.8 Challenges Encountered

The development of the pipeline inspection system presented several technical challenges:

- **Sensor Calibration:** Establishing accurate baseline readings for all sensors required careful calibration, particularly for the flow sensor whose output varied with fluid viscosity and temperature.
- **Serial Communication Stability:** Maintaining stable, continuous serial communication between the Arduino and the host computer required careful buffer management and error handling in the software.
- **Image Processing Tuning:** Finding the right parameters for edge detection and contour analysis that minimized false positives while reliably detecting genuine anomalies required extensive iterative testing.
- **Dashboard Performance:** Ensuring the dashboard remained responsive while simultaneously rendering real-time charts, processing live video, and handling alert logic required efficient data flow architecture.

2.9 Future Scope

The pipeline inspection system developed during this internship represents a solid foundation that can be extended significantly for more advanced and scalable real-world deployment.

2.9.1 Machine Learning-Based Prediction

The current early prediction engine uses rule-based logic. Replacing it with a trained machine learning model – such as a time-series anomaly detection model or a supervised classifier trained on historical sensor data – would significantly improve prediction accuracy, reduce false positives, and enable detection of more complex, subtle blockage patterns.

2.9.2 Autonomous Rover Navigation

The rover currently requires manual operator control for navigation. Integrating autonomous navigation capabilities – using the A* algorithm and occupancy mapping approach developed in Project 2 – would enable fully autonomous rover deployment to flagged pipeline segments without human steering.

2.9.3 Deep Learning-Based Visual Inspection

The current image processing pipeline uses classical computer vision techniques. Integrating a deep learning-based object detection model (such as YOLO or a custom CNN) trained specifically on pipeline inspection imagery would dramatically improve blockage detection accuracy and enable classification of blockage type (sediment, debris, corrosion, etc.).

2.9.4 IoT Integration and Remote Monitoring

Connecting the monitoring system to an IoT cloud platform would enable remote monitoring of pipeline health from anywhere, push notifications to mobile devices, and integration with broader facility management systems. This aligns directly with Industry 4.0 principles of connected, smart manufacturing.

2.9.5 Scalability to Real Pipeline Networks

Scaling the sensor array to cover longer pipeline segments, adding more sensor nodes, and implementing a networked sensor mesh architecture would allow the system to monitor entire industrial pipeline networks in real time.

2.9.6 Drone-Based Inspection for Large Pipelines

For large-diameter industrial pipelines, a drone-based aerial inspection platform could replace the wheeled rover, providing greater maneuverability and the ability to inspect pipes that are inaccessible to ground vehicles.

Chapter 3

Autonomous Navigation in a Mapped Environment

3.1 Introduction and Motivation

Autonomous navigation is a foundational capability for modern industrial mobile robotics. The ability of a robot to move from one location to another without continuous human guidance unlocks transformative possibilities in warehouse automation, intralogistics, manufacturing line supply, and hazardous environment operation.

Conventional Automated Guided Vehicles (AGVs) navigate by following physical guide infrastructure: painted lines on the floor, magnetic strips, or RFID markers. While effective in stable, dedicated environments, this approach has significant limitations:

- Inflexible – any change in the desired route requires physical modification of the guide infrastructure.
- Costly – installation and maintenance of physical guides is expensive.
- Unable to handle unexpected obstacles – if something blocks the path, the AGV stops.
- Unsuitable for dynamic environments where layout changes frequently.

This project aimed to design and implement an autonomous navigation system that addresses all these limitations. The system navigates using a **software-defined map**, computes the **optimal path** algorithmically using A*, and handles **unexpected obstacles** in real time through ultrasonic sensing and dynamic path re-planning – all without any physical guide infrastructure.

3.2 Environment Mapping

The first step in enabling autonomous navigation is creating a digital representation of the physical space through which the robot will navigate.

Figure 3.1: Binary Occupancy Grid Map Representation of the Navigation Environment

3.2.1 Binary Occupancy Grid Representation

The physical navigation environment was represented as a **binary occupancy grid**. In this representation, the continuous physical space is discretized into a uniform grid of square cells. Each cell stores a binary value:

- **0 (Free):** The cell is unoccupied and safe for the robot to traverse.
- **1 (Occupied):** The cell contains a static obstacle (wall, shelf, post, etc.) and is not traversable.

This binary matrix forms a complete, computationally tractable digital map of the navigable environment.

3.2.2 Occupancy Matrix Construction

The occupancy matrix was constructed by physically surveying the navigation environment and encoding all known static obstacles into the grid. The process involved:

1. Dividing the physical environment floor into a grid with a defined cell resolution (e.g., 20cm $\times$ 20cm per cell).
2. Mapping each physical obstacle to its corresponding grid cell(s).
3. Populating the matrix with 1s for occupied cells and 0s for free cells.
4. Incorporating a safety margin (inflation) around obstacles – cells immediately adjacent to known obstacles were also marked as occupied to account for the robot's physical footprint and avoid grazing contacts.

3.2.3 Map Visualization

The occupancy matrix was visualized as a 2D grid graphic displayed in the navigation software. Free cells were shown in white, occupied cells in black, and the computed path was overlaid in a distinct colour. This visualization provided an intuitive representation of both the environment and the robot's planned trajectory.

3.3 Path Planning Using the A* Algorithm

Figure 3.2: A* Algorithm Path Planning – Computed Shortest Path Overlaid on Occupancy Grid

3.3.1 Why A*?

Several path planning algorithms were considered for this project, including Dijkstra's algorithm, Breadth-First Search (BFS), and Greedy Best-First Search. A* was selected for the following reasons:

- **Optimality:** A* guarantees finding the shortest path when used with an admissible heuristic.

- **Efficiency:** The heuristic guides the search toward the goal, dramatically reducing the number of cells explored compared to uninformed search algorithms like BFS or Dijkstra.
- **Completeness:** A* will always find a path if one exists.
- **Flexibility:** A* can be applied to any grid-based or graph-based environment without modification to its core structure.

3.3.2 A* Algorithm Formulation

The A* algorithm assigns each cell n a total cost $f(n)$ computed as:

$$f(n) = g(n) + h(n)$$

Where:

- $g(n)$ = actual cost from the start cell to cell n (number of steps taken).
- $h(n)$ = estimated cost from cell n to the goal (heuristic function).
- $f(n)$ = total estimated cost of the path through cell n.

The algorithm maintains an **open set** (cells to be evaluated) and a **closed set** (cells already evaluated). At each step, the cell with the lowest $f(n)$ is selected from the open set for expansion.

3.3.3 Heuristic Function

The **Manhattan distance** heuristic was selected as it is admissible and consistent for a 4-directional grid (movement restricted to up, down, left, right):

$$h(n) = |x_n - x_{goal}| + |y_n - y_{goal}|$$

This heuristic never overestimates the true cost to the goal, preserving the optimality guarantee of A*.

3.3.4 Path Computation Process

Given a start cell S and a goal cell G on the occupancy grid:

1. Initialize the open set with S. Set $g(S) = 0$ and $h(S) =$ Manhattan distance to G.

2. Select the cell with minimum $f(n)$ from the open set.

3. If this cell is G, reconstruct and return the path.

4. Otherwise, expand the cell: for each free neighbour, compute g, h, f; add to open set if not already evaluated.

5. Move the expanded cell to the closed set.

6. Repeat from step 2.

The result is an ordered list of grid cells forming the optimal path from S to G, avoiding all occupied cells.

3.3.5 Path-to-Motion Translation

The computed sequence of grid cells was translated into motor control commands:

- Consecutive cells in the same direction $\rightarrow$ continue forward motion.
- Direction change $\rightarrow$ compute required turn angle, execute turn, resume forward motion.
- Each cell traversal corresponded to a calibrated physical distance, enabling metric path execution.

3.4 Real-Time Obstacle Detection and Dynamic Re-routing

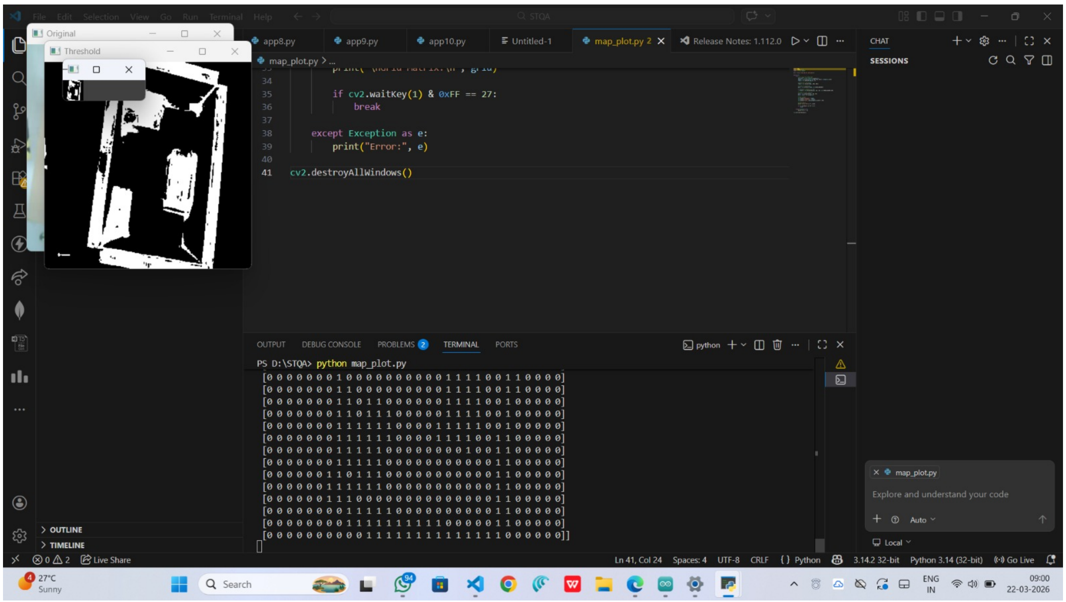

Figure 3.3: Autonomous Navigation Robot with Mounted Ultrasonic Sensor for Obstacle Detection

3.4.1 Limitations of Static Map-Based Navigation

While the pre-computed A* path handles all static obstacles encoded in the occupancy map, real industrial environments are dynamic – objects are moved, personnel walk across paths, and temporary obstacles appear unexpectedly. A purely static-map-based navigation system would fail whenever it encountered an unmodelled obstacle.

3.4.2 Ultrasonic Sensor Integration

An HC-SR04 ultrasonic distance sensor was mounted on the front of the robot to provide continuous real-time range measurements directly ahead of the robot's direction of travel. The sensor:

- Emits an ultrasonic pulse at 40kHz.
- Measures the time elapsed before the echo returns.
- Calculates the distance to the nearest object using: $d = \frac{v \cdot t}{2}$, where v is the speed of sound and t is the round-trip echo time.

- Updates the distance reading at a sampling rate sufficient for responsive obstacle detection at the robot's operating speed.

3.4.3 Obstacle Detection Logic

The obstacle detection module continuously compared the ultrasonic sensor reading against a predefined **safety threshold distance**. When an obstacle was detected within this threshold:

1. Motor control signals were immediately set to halt the robot.
2. The estimated position of the obstacle in the grid was computed based on the robot's current grid position and heading direction.
3. The occupancy matrix was updated to mark the obstacle's estimated cell(s) as occupied.
4. A re-planning call was issued to the A* algorithm with the updated map.
5. A new optimal path from the robot's current position to the goal was computed, now accounting for the newly detected obstacle.
6. Navigation resumed along the updated path.

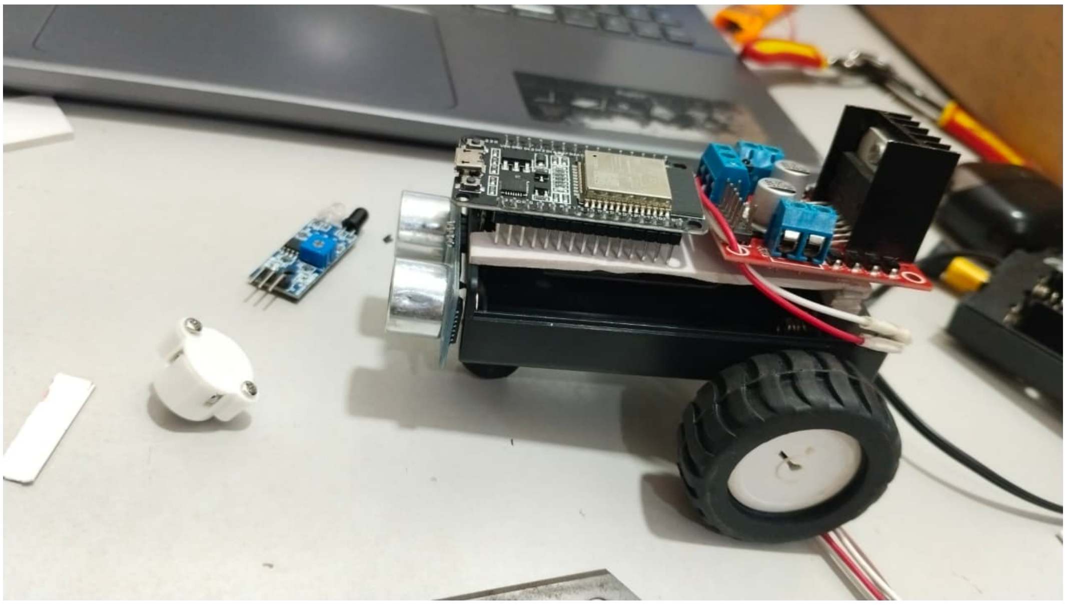

Figure 3.4: Dynamic Re-routing: Original Path (Blue), Detected Obstacle (Red), Re-planned Path (Green)

3.4.4 Re-routing Performance

The re-routing process was designed to complete within a minimal time window to limit the operational interruption caused by unexpected obstacles. The computational efficiency of A* on the grid sizes used in this project enabled re-planning to complete well within a second in all tested scenarios.

3.4.5 Multi-Obstacle Handling

The system handled multiple sequential dynamic obstacles correctly. Each new obstacle detection triggered a fresh re-planning call with the accumulated updated occupancy map, ensuring the robot could navigate past multiple unexpected obstacles in sequence during a single navigation run.

3.5 Navigation Without Line Dependency

A central achievement of this project was demonstrating fully functional autonomous navigation in an environment without any physical guide lines, tracks, or markers on the floor. The robot relied entirely on:

- The pre-loaded binary occupancy map.
- Real-time ultrasonic range sensing.
- Algorithmic path planning and re-planning (A*).
- Motor encoder feedback for accurate distance tracking.

This line-free approach offers significant practical advantages:

- Route changes require only a software update to the occupancy map – no physical floor modification.
- Multiple robots can share the same space with different assigned goals, each computing its own optimal path.
- The system can accommodate workspace layout changes quickly and at no physical infrastructure cost.

3.6 System Components and Architecture

Table 3.1: Hardware and Software Components of the Autonomous Navigation System

Component	**Type**	**Function**
Arduino Uno	Microcontroller	Sensor interfacing, motor control
HC-SR04	Ultrasonic Sensor	Real-time obstacle detection
L298N	Motor Driver	Differential drive wheel control
DC Gear Motors	Actuators	Robot locomotion
Host Computer	Control System	A* path planning, occupancy map, navigation logic
USB Serial	Communication	Bidirectional command/data link
Wheel Encoders	Feedback	Distance and position tracking

3.6.1 Software Architecture

The software system was structured in three functional layers:

1. **Perception Layer:** Continuous ultrasonic sensor data acquisition and obstacle detection logic.
2. **Planning Layer:** A* path planning on the occupancy grid, triggered at start and on each obstacle detection event.
3. **Execution Layer:** Translation of planned path into motor commands, transmitted to the robot via serial communication.

Communication between layers operated on an event-driven model: the perception layer raised an obstacle event that triggered re-planning, and the planning layer generated updated path commands for the execution layer.

3.7 Path Visualization Module

A real-time visualization module was developed to display the robot's navigation state graphically. The visualization showed:

- The occupancy grid map with free and occupied cells clearly distinguished.
- The A*-computed path overlaid on the grid in a distinct colour.
- The robot's current estimated position on the grid, updated in real time.
- Detected obstacles highlighted on the grid at their estimated positions.
- The re-planned path displayed immediately after each obstacle detection event.

This visualization served both as a diagnostic and debugging tool during development and as a demonstration aid during the IEOR Day exhibition.

3.8 Results and Performance Evaluation

The autonomous navigation system was evaluated across a series of test runs with varying start and goal positions and different obstacle placement configurations.

Table 3.2: Navigation Test Results Summary

Test	**Scenario**	**Outcome**	**Re-routes**
1	Open path, no obstacles	Goal reached	0
2	One static obstacle in path	Goal reached	1
3	Two sequential dynamic obstacles	Goal reached	2
4	Dense obstacle field	Goal reached via extended path	3
5	Goal unreachable (blocked)	Correct infeasibility report	–

Key findings:

- The A* algorithm consistently found the optimal (shortest) path for all feasible navigation scenarios.
- Dynamic re-routing correctly handled all unexpected obstacle insertion events.
- The system correctly identified and reported infeasible navigation requests (when the goal was completely surrounded by obstacles).
- Path visualization accurately reflected the robot's real-time position and re-planned trajectories.

3.9 Comparison with Traditional AGV Systems

Table 3.3: Comparison: Traditional Line-Following AGV vs. This Autonomous Navigation System

Feature	**Traditional AGV**	**This System**
Navigation method	Physical line/track	Digital occupancy map
Path flexibility	Fixed at installation	Dynamically re-plannable
Route change cost	High (physical work)	None (software update)
Obstacle avoidance	None / stops	Real-time re-routing
Shortest path guarantee	No	Yes (A*)
Infrastructure cost	High	Low
Environment dependency	Line required	Line-free
Multi-robot support	Complex	Scalable

3.10 Challenges Encountered

- **Odometry Drift:** Accumulated position error from wheel encoder-based odometry over longer navigation runs required periodic correction strategies.
- **Grid Resolution vs. Computation:** Higher grid resolution improves path accuracy but increases A* computation time. Balancing these was critical for real-time re-planning performance.

- **Obstacle Size Estimation:** Accurately estimating the physical size and position of a detected obstacle from a single ultrasonic reading presented challenges; the safety margin inflation approach was used to compensate.
- **Corner and Turning Precision:** Executing precise turns at grid cell boundaries required careful motor timing calibration.

3.11 Future Scope

3.11.1 SLAM Integration

Simultaneous Localization and Mapping (SLAM) would allow the robot to build its occupancy map autonomously in real time as it explores an unknown environment, eliminating the need for a pre-surveyed map. Integrating a LiDAR or depth camera with a SLAM algorithm would transform this system into a fully autonomous explorer.

3.11.2 Multi-Robot Coordination

Extending the system to coordinate multiple robots navigating the same shared environment would require a centralized path planning server that issues non-conflicting paths to each robot, avoiding inter-robot collisions. This is directly applicable to real warehouse automation scenarios.

3.11.3 3D Navigation

The current system navigates in a 2D plane. Extending the occupancy representation to 3D and applying a 3D-capable path planner would enable navigation in multi-level environments or for drones navigating in three-dimensional airspace.

3.11.4 Dynamic Map Updates via Camera

Integrating a camera or LiDAR sensor to continuously update the occupancy map in real time – not just at detected obstacle positions but across the entire visible field – would significantly improve the system's robustness in highly dynamic environments.

3.11.5 ROS Integration

Migrating the navigation system to the Robot Operating System (ROS) framework would provide access to a rich ecosystem of ready-made navigation, sensing, and visualization packages, and would make the system interoperable with a wide range of commercially available robotic platforms.

3.11.6 Machine Learning-Based Path Optimization

Reinforcement learning approaches could be applied to learn navigation policies that improve over time based on operational experience, potentially outperforming static A* in complex, frequently changing environments.

Chapter 4

IEOR Day 2026 – Project Exhibition and Presentation

4.1 Overview of IEOR Day

IEOR Day is the annual flagship event of the Industrial Engineering and Operations Research (IEOR) Department at IIT Bombay. The event serves as a showcase of the department's academic and research achievements, bringing together a diverse audience including industry professionals, corporate representatives, faculty, alumni, PhD and M.Tech researchers, and undergraduate students.

IEOR Day 2026 provided an exceptional platform to present both projects developed during the semester-long internship to a distinguished panel of industry experts and academic professionals. The event created a unique opportunity to receive real-world feedback, engage in professional technical discussions, and experience the intersection of academic project work and industrial practice.

Figure 4.1: Project Exhibition Area Setup at IEOR Day 2026

4.2 Organizational Contributions

Active participation in the organization of IEOR Day 2026 was an integral component of the internship experience. Organizational responsibilities undertaken included:

- **Exhibition Area Setup:** Coordinating the physical arrangement of demonstration hardware, display tables, poster boards, and electronic equipment for the project exhibition zone.

- **Poster and Display Preparation:** Assisting in the design and printing of project posters and informational display materials.
- **System Setup and Testing:** Setting up and testing the pipeline monitoring dashboard, rover systems, and autonomous navigation demonstration hardware prior to the event to ensure reliable live operation.
- **Cross-Team Coordination:** Liaising with other student project teams to synchronize demonstration schedules and ensure smooth, non-overlapping visitor flow.
- **Visitor Guidance:** Welcoming and guiding event visitors through the project demonstration area, explaining the exhibition structure and pointing them to relevant projects based on their interests.

Participation in these organizational activities provided valuable insight into event management, professional coordination, and the logistics of presenting technical work in a formal institutional setting.

4.3 Project Exhibition Details

Figure 4.2: Presenting the Pipeline Inspection Rover System to Industry Experts at IEOR Day

4.3.1 Pipeline Inspection Rover – Live Exhibition

The Pipeline Inspection Rover system was set up for a fully live demonstration during IEOR Day 2026. The monitoring dashboard was operational, receiving live data from the sensor array and displaying real-time readings, alerts, and prediction outputs.

During the exhibition:

- Visitors observed the real-time sensor data visualization on the dashboard.
- The alert system was demonstrated by artificially triggering threshold breaches to show warning and critical alert responses.
- The early prediction engine was explained and demonstrated with sensor data trending toward blockage conditions.
- The rover was deployed to a simulated blocked pipeline segment, and the live camera feed with image processing annotations was displayed on the dashboard.

- Industry visitors were invited to interact with the dashboard interface and observe the system's response to different simulated pipeline conditions.

4.3.2 Autonomous Navigation System – Live Exhibition

Figure 4.3: Live Demonstration of Autonomous Navigation with Dynamic Obstacle Avoidance

The Autonomous Navigation system was demonstrated live on a physical test track representing the mapped grid environment. The demonstration sequence was:

1. The robot was positioned at the start cell with a goal cell defined on the dashboard.
2. The A* path was computed and displayed on the occupancy grid visualization.
3. The robot began navigating autonomously along the computed path.
4. A physical obstacle was manually introduced in the robot's path mid-navigation.
5. The ultrasonic sensor detected the obstacle; the robot halted.
6. A* recomputed a new path around the obstacle.
7. The robot resumed navigation along the updated path and successfully reached the goal.

The real-time path visualization displayed alongside the physical robot made the demonstration visually compelling and easy to follow for visitors with varying levels of technical background.

4.4 Formal Presentations to Industry Experts

Formal structured presentations were delivered to panels of industry experts and academic faculty in a seminar room setting. Each presentation followed the structure:

1. **Problem Statement:** Clear articulation of the industrial challenge being addressed.
2. **Proposed Solution:** Overview of the system architecture and design approach.
3. **Technical Implementation:** Detailed walkthrough of key technical components, algorithms, and hardware.
4. **Live Demonstration:** Video demonstration of the system in operation.
5. **Results and Validation:** Summary of test results and performance metrics.
6. **Industrial Relevance and Future Scope:** Discussion of real-world applicability and future development directions.
7. **Q&A:** Open question and answer session with the expert panel.

4.5 Industry Expert Feedback and Discussion

Figure 4.4: Team Photograph at IEOR Day 2026 – Post Exhibition

Engaging with industry professionals provided genuinely valuable feedback that offered perspectives not typically encountered in academic project development. Key discussion points and feedback received included:

4.5.1 Pipeline Inspection Rover Feedback

- Industry professionals highlighted the potential to scale the sensor network to cover longer real pipeline segments using wireless sensor nodes, eliminating the need for wired sensor connections.
- Questions were raised about the robustness of the prediction engine under noisy real-world sensor conditions, prompting discussion of machine learning-based approaches.
- Experts from the manufacturing sector noted the direct applicability of the monitoring dashboard concept to existing SCADA (Supervisory Control and Data Acquisition) system architectures.

- Suggestions were made regarding integrating pressure sensors as an additional parameter, as pressure drop is often a more direct indicator of blockage than flow rate alone.

4.5.2 Autonomous Navigation System Feedback

- Industry experts expressed strong interest in the line-free navigation approach, noting its direct relevance to flexible warehouse automation applications where fixed AGV tracks are a significant operational constraint.
- Questions were raised about navigation accuracy over longer distances and the management of cumulative odometry error, leading to discussion of correction mechanisms and potential LiDAR integration.
- The A* algorithm's performance in large, complex grid environments was discussed, with experts suggesting Dijkstra variants or hierarchical planning for very large warehouse maps.
- Safety aspects of autonomous navigation near human workers were highlighted as a critical consideration for real-world deployment, including the need for 360° sensing and speed regulation in human proximity.

4.6 Learnings from IEOR Day

Participation in IEOR Day 2026 contributed deeply to the overall internship experience beyond the technical project work. Key takeaways included:

- **Audience Awareness:** Adapting technical explanations to different audience backgrounds – from highly technical engineers to business professionals – required real-time assessment of the audience and flexible communication.
- **Demonstration Preparedness:** Live demonstrations at a public event require extensive preparation, including rehearsal, contingency planning for technical failures, and clear explanation scripts.
- **Professional Confidence:** Presenting in front of a formal industry panel and fielding critical questions developed significant professional confidence and composure.
- **Industrial Perspective:** Direct interaction with working industry professionals revealed practical considerations – scalability, fail-safes, integration with existing

systems, cost – that academic project development does not always naturally surface.

- **Networking:** IEOR Day provided a professional networking opportunity with industry representatives and researchers, broadening awareness of career opportunities in industrial automation and robotics.

Chapter 5

Learning Outcomes

The semester-long internship at IIT Bombay under the FOSSEE initiative provided a comprehensive and multi-dimensional learning experience spanning technical skills, engineering methodology, professional development, and industry awareness. This chapter summarizes the key learning outcomes from the internship period.

5.1 Multi-Sensor System Design and Integration

The Pipeline Inspection Rover project provided intensive hands-on experience in integrating multiple heterogeneous sensors into a unified, reliable data acquisition system. Key learnings included:

- Understanding the working principles, output characteristics, and interfacing requirements of water flow sensors, ultrasonic sensors, digital temperature sensors, and gas/smoke sensors.
- Implementing accurate analog-to-digital conversion and pulse-counting logic in microcontroller firmware for different sensor types.
- Managing the challenges of multi-sensor data synchronization and consistent sampling rate maintenance.
- Calibrating sensors against known reference values to ensure accurate baseline readings.
- Designing robust serial communication protocols between the microcontroller and host computer for reliable, continuous data transmission.

5.2 Real-Time System Design and Dashboard Development

Developing the real-time monitoring dashboard introduced the principles and practical challenges of real-time data-driven application development:

- Designing data pipelines that ingest, parse, process, and display sensor data with minimal latency.
- Implementing time-series data visualization with live-updating charts and status indicators.
- Structuring alert logic that correctly classifies sensor conditions across multiple severity levels.
- Developing a responsive UI that remains performant while simultaneously handling real-time data rendering and live video display.
- Implementing data logging for historical sensor reading storage and retrieval.

5.3 Computer Vision and Real-Time Image Processing

The live feed processing component of the pipeline inspection system introduced foundational computer vision skills:

- Applying classical image processing techniques (Gaussian blur, Canny edge detection, contour analysis) for real-world visual inspection tasks.
- Understanding the performance-accuracy trade-offs involved in real-time image processing.
- Designing annotation overlays that convey detected anomalies clearly to operators.
- Integrating a live camera feed into a larger software application with minimal latency impact.

5.4 Algorithmic Path Planning

Implementing the A* algorithm for autonomous navigation developed both theoretical understanding and practical implementation skills in algorithmic robotics:

- Deep understanding of the A* algorithm: data structures (open/closed sets), cost functions, heuristic selection, and path reconstruction.
- Analysis of admissible and consistent heuristics and their impact on algorithm optimality.
- Implementation of the algorithm in an embedded software context with real-time performance requirements.
- Translating abstract algorithmic path outputs into physical robot motor command sequences.

5.5 Occupancy Grid Mapping and Spatial Reasoning

Working with binary occupancy grids developed foundational skills in digital environment representation for robotics:

- Understanding how physical environments are discretized and represented as computational data structures.
- Designing grid resolution strategies that balance spatial accuracy with computational load.
- Implementing obstacle inflation to account for robot physical footprint in path safety.
- Dynamically updating the occupancy map during navigation in response to real-time sensor input.

5.6 Autonomous System Design and Closed-Loop Control

Designing the autonomous navigation system provided experience in closed-loop robotic system engineering:

- Structuring a multi-layer autonomous system (perception, planning, execution) with clean interfaces between layers.
- Implementing event-driven system architecture where obstacle detection events trigger re-planning.

- Managing the interaction between real-time sensor input and longer-running algorithmic path planning.
- Handling edge cases: infeasible paths, consecutive obstacle encounters, path convergence issues.

5.7 Embedded Systems and Serial Communication

Both projects involved extensive embedded systems work:

- Microcontroller programming for sensor interfacing, motor control, and serial data transmission.
- Serial communication protocol design and implementation between Arduino and host computer.
- Managing serial buffer overflow, timing, and synchronization in continuous-operation systems.
- Integrating hardware motor drivers with software-generated control signals for smooth, precise robot motion.

5.8 Technical Communication and Presentation Skills

IEOR Day 2026 provided invaluable development of professional communication skills:

- Structuring clear, compelling presentations of complex technical systems for diverse audiences.
- Delivering live demonstrations under time constraints and public observation pressure.
- Responding confidently to critical technical questions from industry experts.
- Adapting communication style in real time to match the technical background of the audience.
- Producing professional display materials (posters, demonstration scripts) for a formal institutional event.

5.9 Industry 4.0 and Smart Manufacturing Awareness

Both projects embodied core Industry 4.0 principles, providing practical exposure to the technologies and concepts that define modern smart manufacturing:

- **Cyber-Physical Systems:** Integration of digital software systems with physical sensors, actuators, and robotic hardware.
- **Real-Time Data-Driven Decision Making:** Using live sensor data to drive autonomous decisions in both monitoring and navigation systems.
- **Predictive Maintenance:** The pipeline inspection system's early prediction engine demonstrated the principles of proactive, data-driven maintenance.
- **Autonomous Operation:** Both systems incorporated significant autonomous capability, reducing the need for continuous human intervention.
- **Modular Architecture:** Both systems were structured as modular, layered architectures that mirror real industrial automation design patterns.

5.10 Problem-Solving, Debugging, and Engineering Judgment

Perhaps the most transferable learning from the internship was the development of practical engineering judgment through problem-solving:

- Systematic debugging of embedded hardware-software integration issues.
- Diagnosing and resolving sensor calibration errors and communication faults.
- Making informed design trade-off decisions (grid resolution vs. computation time; detection sensitivity vs. false positive rate).
- Iteratively refining systems based on test results and observed failure modes.
- Managing project scope and priorities within a defined time window.

5.11 Overall Technical and Professional Growth

The internship demonstrated the complete engineering lifecycle – from problem identification and system conceptualization through design, implementation, testing, refinement, and public presentation – within a structured, professional academic environment. This holistic exposure is difficult to replicate through coursework alone and represents significant technical and professional growth.

Chapter 6

Conclusion

The FOSSEE Semester Internship at the IEOR Department, IIT Bombay, from February to April 2026, provided a rich, comprehensive, and deeply formative engineering experience. Over the course of the internship, two major technical projects were developed from conception to working implementation and public demonstration, and both were presented to an audience of industry professionals at IEOR Day 2026.

The first project, the **Pipeline Inspection Rover for Blockage Detection**, addressed a genuine and significant industrial challenge. By combining a multi-sensor monitoring station – integrating water flow, ultrasonic level, temperature, and smoke sensors – with an early prediction engine, an approximate location estimation algorithm, and a deployable camera-equipped rover, the system demonstrated how proactive, intelligent pipeline inspection can be achieved with practical, accessible hardware and software tools. The real-time responsive dashboard tied the entire system into a coherent operator interface, making the system's capabilities immediately accessible and actionable. The project demonstrated the principles of sensor fusion, predictive analytics, real-time data visualization, and computer vision in a unified industrial application.

The second project, the **Autonomous Navigation System**, demonstrated that line-free, map-based autonomous navigation is achievable with well-designed software, standard hardware, and carefully implemented algorithms. By encoding the environment as a binary occupancy grid, computing optimal paths with the A* algorithm, and handling unexpected obstacles through real-time ultrasonic detection and dynamic path re-planning, the system delivered robust autonomous navigation without any physical guide infrastructure. The comparison with traditional line-following AGVs highlighted the significant practical advantages of the algorithmic approach in terms of flexibility, adaptability, and scalability.

Participation in **IEOR Day 2026** elevated the internship beyond technical project work into a professional experience. Presenting both systems to industry experts, receiving constructive critical feedback, and engaging in substantive technical discussions provided a perspective on how academic projects relate to real-world industrial needs – a perspective that is invaluable for a student transitioning toward a professional engineering career.

Both projects embodied the principles of **Industry 4.0**: cyber-physical systems integration, real-time data-driven decision making, autonomous operation, and modular system architecture. They demonstrated that these principles are not abstract – they can be implemented practically by engineering students with appropriate guidance, hardware access, and commitment.

The technical skills developed – sensor integration, real-time system design, computer vision, algorithmic path planning, embedded systems programming, autonomous system architecture – complement the theoretical knowledge gained through formal coursework and lay a strong, practical foundation for graduate study or professional work in robotics, automation, and intelligent manufacturing systems.

The FOSSEE programme and IIT Bombay's IEOR Department provided an exceptional environment for this learning: access to well-equipped laboratories, experienced mentors, a supportive research community, and the opportunity to present work in a professional forum. This internship experience has been a defining chapter in the academic journey, and its influence will be felt throughout the professional career that follows.

Project Resources

The complete source code, firmware, implementation files, wiring diagrams, and project documentation for all projects developed during the internship are publicly available at:

GitHub Repository:
https://github.com/SmartDeve/IIT-IEOR-Lab

Demonstration videos, system walkthroughs, testing recordings, and presentation materials are available at:

Google Drive Folder:
https://drive.google.com/drive/folders/1I836Ifn8df5Zflaje_ec24CngTVtIiff

The repository contains the following content organized by project:

- `/pipeline-inspection/` – Arduino firmware, dashboard source code, image processing scripts, wiring schematics.
- `/autonomous-navigation/` – A* algorithm implementation, occupancy map generator, robot control firmware, path visualization code.
- `/documentation/` – System architecture diagrams, component datasheets, test logs, and this report.

Bibliography

[1] National Mission on Education through ICT (NMEICT), Department of Higher Education, Ministry of Education, Government of India.
https://www.nmeict.ac.in

[2] FOSSEE Project, Department of Chemical Engineering, Indian Institute of Technology Bombay.
https://fossee.in

[3] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[4] H. P. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, St. Louis, MO, USA, 1985, pp. 116–121.

[5] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[6] M. Srinivasan and A. Bhaskaran, "A review of autonomous guided vehicles in smart manufacturing," *Journal of Manufacturing Systems*, vol. 52, pp. 110–125, 2019.

[7] B. Ranjit and P. K. Singh, "Pipeline inspection technologies: A comprehensive review," *International Journal of Advanced Manufacturing Technology*, vol. 98, pp. 1123–1145, 2018.

[8] K. Schwab, *The Fourth Industrial Revolution*. World Economic Forum, Geneva, Switzerland, 2016.

[9] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
https://docs.opencv.org

[10] Arduino LLC, *Arduino Reference Documentation*.
https://docs.arduino.cc

[11] Cytron Technologies, *HC-SR04 Ultrasonic Ranging Module Datasheet*, 2020.

[12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, Cambridge, MA, USA, 2005.

[13] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[14] D. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, 1997.

[15] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel, "Prognostics and health management design for rotary machinery systems – reviews, methodology and applications," *Mechanical Systems and Signal Processing*, vol. 42, pp. 314–334, 2014.