Sinhgad Institutes

DEPARTMENT OF COMPUTER ENGINEERING

SUBMITTED BY

Jaywardhan Thorat
C33317

ABSTRACT

I worked on the design and development of a high-speed line follower robot. The primary goal of this project was to build a robot capable of accurately following a black line on a white surface at high speed, with minimal deviation and quick response during sharp turns or curves.

The robot was built using the Arduino Nano microcontroller and programmed via the Arduino IDE. For line detection, a LAS08 line sensor array was used, which provides high-resolution feedback using multiple IR sensors in a single module. This allowed the robot to precisely determine the line's position in real time, even at higher speeds.

Based on the sensor readings, the robot controlled its movement using DC motors, with direction and speed handled through appropriate motor driver logic. The control algorithm was optimized using PID (Proportional-Integral-Derivative) logic to ensure smooth navigation, fast recovery from errors, and efficient cornering at high velocity.

Special focus was given to sensor calibration, PID tuning, and achieving balance between speed and stability. This hands-on project enhanced my knowledge in sensor interfacing, real-time control systems, and embedded robotics programming.

INTRODUCTION

Being Head Of College Robotics Team, I was assigned a project to design and build a high-speed line follower robot. The primary objective was to develop a robot capable of accurately detecting and following a predefined path (typically a black line on a white surface) at high speed, with smooth turns and minimal deviation. The robot was built using an Arduino Uno microcontroller, programmed via Arduino IDE, and employed an LAS08 IR sensor array for real-time line detection.

To ensure precise and responsive movement, I implemented PID (Proportional-Integral-Derivative) control logic which enabled the robot to continuously adjust its position based on sensor feedback. This logic helped in maintaining stability and reducing overshoot or lag, especially at corners or during rapid directional changes.

This project allowed me to enhance my understanding of sensor calibration, motor control, and real-time feedback systems. It also provided practical exposure to embedded C programming, robotic algorithms, and troubleshooting hardware-software integration issues, giving me a strong foundation for future work in automation and robotics

PROBLEM DEFINITION

1. Title of the Project: -High-Speed Line Follower Robot Using Arduino

2. Problem Definition:

In industrial automation, research, and robotics competitions, the need for precise and fast autonomous movement is crucial. Line follower robots are fundamental in these domains, where robots must navigate accurately along a predefined path. However, many conventional line followers suffer from instability at high speeds, delayed response to sharp turns, or imprecise path tracking, especially when the environment changes or the sensors are not well-calibrated.

The challenge addressed during this project was to develop a robot capable of maintaining high speed while still being responsive to path deviations and stable during turns. Using basic IR sensors often leads to delays and overshooting at corners, limiting performance. Thus, the goal was to use a precise sensor array (LAS08) and efficient control logic, including PID (Proportional–Integral–Derivative) algorithms, to optimize line detection and maintain smooth, accurate motion.

The solution was developed on Arduino Uno and programmed through the Arduino IDE, ensuring a low-cost and flexible embedded system capable of meeting high-speed performance demands.

3. Aim & Objectives:

Aim:

To design and build a high-speed line follower robot using Arduino Uno and LAS08 sensor array, capable of accurately detecting and following a line with minimal error and smooth control, even at high speeds.

Objectives:

a. Hardware Integration:

Use Arduino Uno as the control unit for processing sensor data and controlling motors.

Interface the LAS08 IR sensor array to accurately detect the line's position.

Connect and configure DC motors and motor driver circuitry to support responsive movement.

b. Programming and Control:

Develop embedded C code using the Arduino IDE.

Implement PID control logic to adjust motor speed and direction dynamically based on sensor feedback.

Calibrate sensors to handle variations in lighting and surface conditions.

c. Performance Optimization:

Test and refine the robot's speed without compromising on line-following accuracy.

Ensure the bot maintains smooth turning and stability at junctions or curves.

Troubleshoot and improve code for real-time response and minimal delay in control logic.

d. Learning Outcome:

Gain hands-on experience with embedded systems, sensor calibration, and motor control techniques.

Understand real-world applications of automation and robotics.

Develop a strong foundation for future robotic projects and control system designs

METHODOLOGICAL DETAILS

The development of the high-speed line follower robot was carried out through a structured process involving hardware selection, circuit design, software development, and performance testing. The following steps summarize the overall methodology:

- 1. Requirement Analysis & Planning

Defined the core objective: to build a fast and accurate line-following robot.
Identified essential components: Arduino Uno, LAS08 IR sensor array, DC motors, motor driver, and power supply.
Studied the working environment (black line on white surface) to guide sensor and control logic design.

- 2. Hardware Selection & Assembly

Arduino Uno was chosen for its simplicity and sufficient I/O pins.
LAS08 IR Sensor Array (8-channel analog line sensor) was used for accurate and wide-range line detection.
DC Motors were selected for high-speed operation with torque suitable for stable movement.
Motor Driver (L298N or similar) was used to control the direction and speed of the motors.
Components were mounted on a custom chassis, ensuring balance and low weight for faster motion.

- 3. Sensor Calibration

Calibrated the LAS08 sensor to distinguish between black and white surfaces in varying light conditions.
Analyzed sensor readings to map sensor values to specific robot actions (e.g., turn left, right, or go straight).

- 4. Control Logic & Programming

Developed the main control algorithm using Arduino IDE in embedded C.
Implemented PID (Proportional–Integral–Derivative) control for real-time speed and direction adjustments:
P (Proportional): Responds to the current error (distance from the line center).
I (Integral): Corrects accumulated errors over time.
D (Derivative): Predicts future errors based on the rate of change.
Adjusted PID constants (Kp, Ki, Kd) through multiple test iterations to achieve optimal performance.

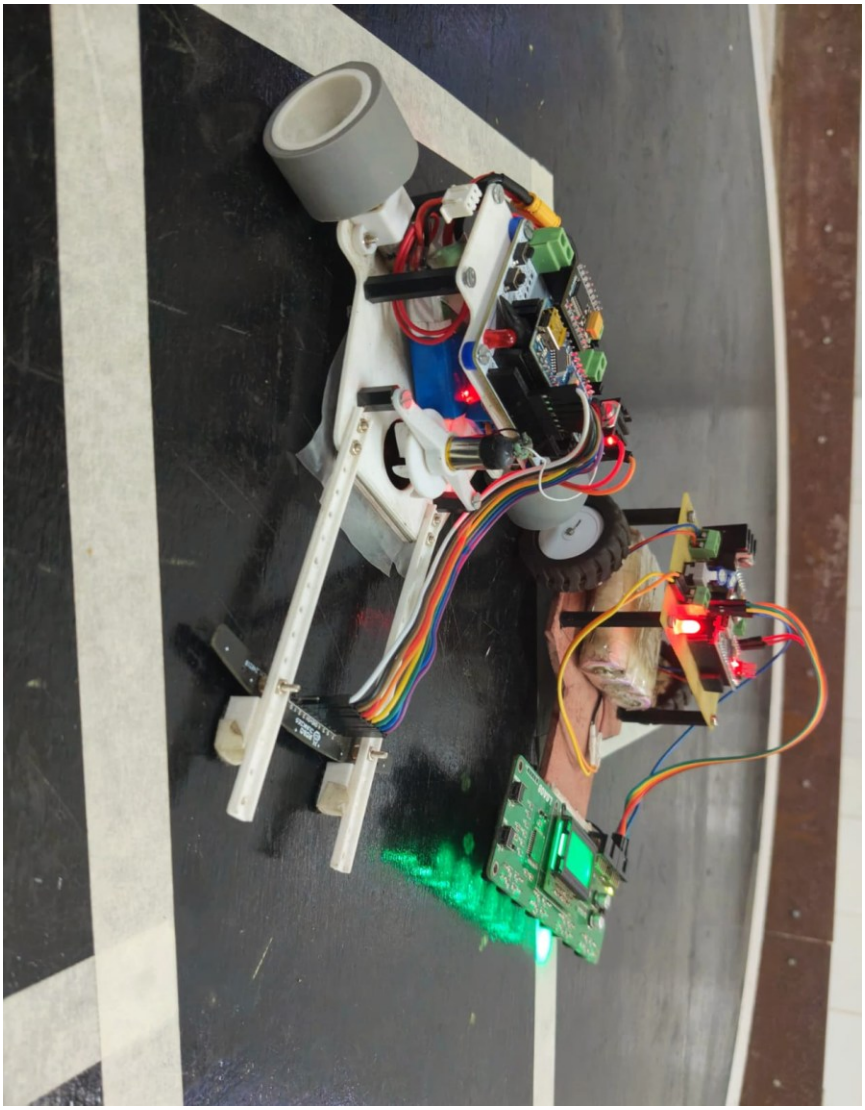
- 5. Testing & Optimization

Conducted trial runs to test speed, accuracy, and cornering performance.
Tuned PID values and motor speeds to minimize deviation and maximize stability.
Ensured consistent behavior on different sections of the track (straights, curves, and intersections).
Debugged sensor noise and adjusted the threshold for precise detection.

- 6. Final Implementation

Verified the robot could:
Follow the line smoothly at high speed.
Adjust dynamically to curves.
Maintain control without overshooting or losing track.

RESULTS:-




Laqshya'11
SOLVATION

CONCLUSION

,I gained comprehensive practical knowledge in the field of robotics and automation. I was trained in essential areas such as PCB design, PLC programming, Arduino-based embedded systems, and the fundamentals of robotics logic.

As a part of the internship, I successfully designed and developed a high-speed line follower robot, which involved integrating hardware components, programming in Arduino IDE, and applying control logic using a PID algorithm with the LAS08 sensor array for accurate and stable line tracking.

This experience greatly enhanced my skills in sensor interfacing, motor control, circuit design, and real-time problem solving. It also allowed me to bridge the gap between theoretical knowledge and its real-world application.

REFERENCES

- Arduino Uno Documentation. (n.d.). Retrieved from https://www.arduino.cc/en/Main/ArduinoBoardUno

- LASER Line Sensor (LAS08) Datasheet – For specifications and interfacing guidelines.

- PID Control Basics – Control Tutorials for Arduino and Robotics. Retrieved from https://www.controleng.com or https://www.ni.com

- L298N Motor Driver Module Guide. Retrieved from https://components101.com Getting Started with Arduino." Banzi, M. (Book/Resource)