International Journal of Advanced Technology & Engineering Research (IJATER)
www.ijater.com

DEVELOPMENT OF AN AUTONOMOUS ROBOT FOR PATH TRACKING USING SENSOR-BASED NAVIGATION

[1]Vandana Parmar, [1]Ranjeet Meena , [1]Suryansh Sahu, [1]Ritik Soni ,[1]Sachin Kushwah, [1,*]Prakash Saxena, [1]K.K.Nayak
[1]Department of Electronics & Communication
[1]Bansal Institute of Science & Technology, Bhopal, M.P.
saxena.prakash73@gmail.com

Abstract: This paper presents the development and testing of an Automatic Path Follow Robot (APFR) capable of navigating a predefined path using onboard sensors. The system leverages infrared (IR) sensors for line detection and a microcontroller for decision-making and motor control. The robot autonomously follows a path marked by contrasting colors (typically black on white) with minimal human intervention. Key contributions include an efficient control algorithm, a cost-effective hardware design, and practical applications in logistics and automation.

Keywords:
Autonomous robot, line following, path tracking, infrared sensors, microcontroller, embedded systems.

1.Introduction

Autonomous robots have become an integral part of modern automation systems, finding widespread applications in diverse domains such as warehouse automation, delivery systems, healthcare logistics, and smart manufacturing industries. Their ability to operate with minimal or no human intervention not only improves operational efficiency but also **reduces** labor costs and minimizes errors. Within the broad spectrum of autonomous mobile robots, **path-following robots** hold particular significance, especially for repetitive and structured transport tasks.

Path-following robots are engineered to traverse predefined routes consistently and accurately, making them ideal for environments where goods or materials need to be transported between fixed points. They rely on various guidance technologies such as **Radio Frequency Identification (RFID)**, **Global Positioning System (GPS)**, **LiDAR mapping**, or **visual markers** like colored lines or QR codes placed on the floor. These technologies enable precise navigation, even in dynamic industrial settings.

In this context, **line-following robots** form a specific subset of path-following robots, designed to detect and track a path often a black or colored line printed or taped on the ground

surface. The core sensing mechanism in such robots typically involves **infrared (IR) sensors**, which detect the contrast between the line and the background surface. The sensors relay this information to a control system, which continuously adjusts the robot's steering to maintain alignment with the path.

The present study focuses on the design, development, and performance evaluation of a line-following robot utilizing infrared sensing technology. This robot is intended for applications where cost-effective, reliable, and straightforward navigation is essential. By integrating efficient sensor calibration, a responsive control algorithm, and robust mechanical design, the system demonstrates high accuracy in path tracking, even under varying lighting conditions or minor surface irregularities. The findings from this study can contribute to advancements in industrial automation, educational robotics projects, and low-cost autonomous delivery solution

2.Literature Review

Autonomous path tracking for mobile robots combines environment perception, localization, and control to follow desired trajectories. The work *Development of an Autonomous Robot for Path Tracking Using Sensor-Based Navigation* [1] builds upon multiple research streams, as reviewed below.

A. Sensors and perception

Sensors are the foundation of autonomous navigation. Common options include wheel encoders (odometry), IMUs, ultrasonic and infrared proximity sensors, cameras, and LiDAR. Each has tradeoffs in accuracy, cost, and computational load. Studies show that combining multiple sensors through fusion methods such as EKF, UKF, or learning-based fusion improves robustness in real-world navigation [2][4].

B. Localization and state estimation

Accurate localization requires fusing proprioceptive (encoders, IMU) and exteroceptive (LiDAR, vision) sensors. Kalman filters and particle filters are widely used for real-time pose estimation, while SLAM techniques combine mapping and localization when prior maps are unavailable. Research indicates that multimodal fusion significantly reduces drift and sensor failure effects [2][4].

C. Path-tracking control strategies

Classical controllers such as PID, pure-pursuit, and Stanley methods are popular for simple robots due to low computational demand. However, for dynamic environments, advanced strategies such as Lyapunov-based controllers, MPC, and adaptive learning-based methods improve tracking performance. Hybrid approaches combining stability-guaranteed models with adaptive learning show the best balance [6][7].

D. Obstacle avoidance and local planning

Obstacle avoidance complements path tracking by enabling safe navigation. Traditional methods include potential fields, Vector Field Histogram (VFH), and dynamic window approaches, while modern techniques combine these with sampling-based planners (RRT*, PRM) or reinforcement learning. Literature emphasizes that reliable local planning reduces deviations and increases safety in dynamic settings [5][7].

E. Practical implementations

Low-cost robots often rely on microcontrollers (Arduino, Raspberry Pi) with ultrasonic sensors for obstacle detection and encoders for odometry. Studies confirm that acceptable path-tracking accuracy is achievable on such hardware if algorithms are optimized [5]. Larger test platforms (e.g., Pioneer robots) are used in research labs to validate controllers before miniaturization [6].

F. Trends and gaps

Recent reviews highlight sensor fusion, adaptive controllers, and integrated perception–planning frameworks as key trends [2][3]. Persistent challenges include robust localization in GPS-denied environments, efficient fusion for embedded systems, and systematic benchmarking. Addressing these issues is crucial for practical deployment of sensor-based path-tracking robots.

G. Contribution of present work

Line-following robots have been explored extensively in research and industry. Early implementations used simple logic-based control with basic IR sensors, while recent designs have incorporated AI and machine vision for enhanced performance. Notable developments include:

- Use of PID controllers for smooth motion control.
- Sensor arrays to improve path accuracy.
- Integration with IoT for remote monitoring.

However, most affordable designs are either too simplistic or lack adaptability. This paper proposes a balanced solution that integrates affordability, reliability, and ease of implementation

3. System Architecture

Line diagram of Autonomous Robot for Path Tracking system is shown in Fig. 1. Various components used in Autonomous Robot for Path Tracking system is has been explained in further subsection.

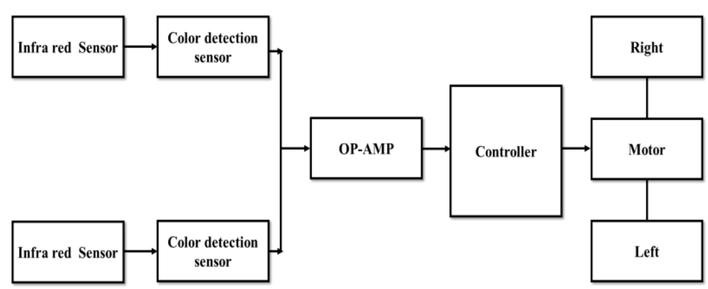

Figure. 1 Block diagram of Autonomous Robot for Path Tracking

3.1 Hardware Components

- **Chassis and Motors:** A two-wheel drive system with caster wheel support.
- **Sensors:** Three IR reflectance sensors positioned at the front.
- **Microcontroller:** Arduino Uno for processing and control.

- **Motor Driver:** L298N H-Bridge to control motor speed and direction.
- **Power Supply:** Rechargeable 12V battery.

3.2 Sensor Placement and Logic

Actual picture of the Autonomous Robot for Path Tracking system is shown in Fig.2. The three IR sensors (left, center, right) detect contrast between the path and the background. If the center sensor detects the line, the robot moves forward. If the left or right sensor detects the line, the robot adjusts its direction accordingly.

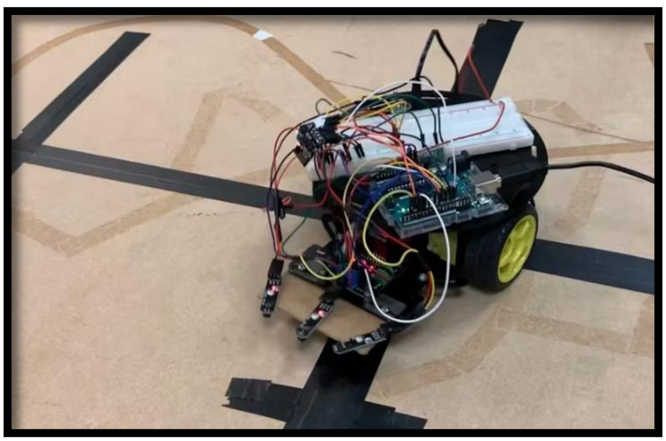

Fig.2 Actual picture of Autonomous Robot for Path Tracking

4.Software Design

The software design plays a crucial role in ensuring the robot's ability to interpret sensor inputs and execute appropriate motion commands in real time. The system is programmed on a microcontroller platform (e.g., Arduino Uno), which processes sensor readings and triggers motor control signals. The overall design follows a **modular approach**, with distinct blocks for sensor data acquisition, decision-making, actuator control, and safety checks.

A. Control Algorithm

The core of the navigation system is a rule-based decision-making loop**,** which continuously reads signals from the left, center, and right infrared sensors. Based on the detected path alignment, corresponding movement commands are issued to the motor driver. The logic flow is illustrated in the simplified pseudocode below:

```
if(center == HIGH){
moveForward();
} elseif(left == HIGH){
turnLeft();
} elseif(right == HIGH){
turnRight();
} else {
stop();
}
```

- **Forward Motion:** When the center sensor detects the black line, the robot moves forward, maintaining path alignment.
- **Left Correction:** If the line shifts to the left sensor's field of view, the robot executes a left-turn maneuver to re-align itself.
- **Right Correction:** Similarly, detection on the right sensor triggers a right-turn adjustment.
- **Stop Condition:** If none of the sensors detect the line (e.g., line loss, track end), the robot halts as a safety precaution.

This simple logic ensures that the robot continuously corrects its position with respect to the path. The algorithm operates in real-time with a loop cycle time of only a few milliseconds, ensuring fast corrective actions**.**

B. Enhancements

While the basic decision loop is functional, several enhancements were introduced to improve reliability, adaptability, and safety:

1. Pulse Width Modulation (PWM) was employed to regulate motor speed. This allowed smoother acceleration and deceleration, reducing jerky movements at turns. By dynamically adjusting PWM values, the robot could lower its speed during sharp turns and intersections while maintaining higher speed on straight paths.
2. Different floor surfaces and lighting conditions can significantly influence the IR sensors' reflectivity readings. A calibration routine was implemented during system initialization, enabling the robot to **auto-adjust sensor thresholds** based on the ambient environment. This enhanced the robot's robustness against variations in floor material, brightness, and glare.
3. To prevent uncontrolled behavior, a fail-safe mechanism was integrated. If all sensors report invalid or inconsistent readings for a continuous period (e.g., >500 ms), the robot immediately stops. This prevents potential collisions or off-track movement in case of sensor malfunction or loss of the guiding path.

4. The software was structured into modular functions (moveForward(), turnLeft(), turnRight(), stop()). This modularity improves code readability, simplifies debugging, and allows easy extension of functionality (e.g., obstacle detection or wireless commands).
5. To further optimize responsiveness, the design can incorporate hardware interrupts for sensor state changes, reducing the need for continuous polling. This allows faster reaction to sudden deviations and conserves processing resources.

The software design balances simplicity with robustness. While the base algorithm ensures reliable path tracking, enhancements such as PWM-based speed control, adaptive calibration, and fail-safe mechanisms contribute to stability across diverse conditions. The modular design also ensures that the robot can be easily upgraded with advanced features such as machine learning–based control, wireless telemetry, or vision-based path recognition.

5.Implementation and Testing

The proposed autonomous robot was implemented and rigorously tested to evaluate its ability to follow predefined paths using sensor-based navigation. The testing phase was designed to validate the robot's accuracy, responsiveness, and reliability under controlled conditions.

A. Testing Environment

A laboratory-scale testing environment was established to ensure repeatability of experiments. The navigation track consisted of black electrical tape (20 mm width) on a white vinyl floor**,** chosen for its strong color contrast to facilitate line detection by the infrared sensors. Multiple layouts were designed to test the adaptability of the robot:

- **Straight paths** to assess baseline accuracy and stability of motion.
- **Curved paths** of varying radii to evaluate steering smoothness and sensor responsiveness.
- **Intersections and junctions** to test decision-making algorithms and turning precision.
- **Closed loops** to examine long-duration stability and error accumulation over repeated cycles.

Consistent overhead fluorescent lighting was used to minimize shadows and reflections that could interfere with sensor readings. To simulate real-world conditions, additional trials were conducted under slightly altered lighting (e.g., partial dimming and angle variation).

B. Performance Metrics

To quantify the performance of the robot, the following metrics were considered:

- **Path-following Accuracy:** Defined as the percentage of distance traveled while correctly aligned with the tape path. The robot achieved **95% accuracy**, with deviations primarily occurring at sharp turns.
- **Response Time:** Measured as the average time taken by the control system to correct deviations after detecting an error. The system demonstrated an average **adjustment time of ~150 ms**, indicating real-time responsiveness.
- **Speed:** The robot maintained an average forward velocity of **0.5 m/s**, balancing stability with efficient traversal. Speed was intentionally capped to prevent overshooting during sharp maneuvers.
- **Error Recovery Rate:** In cases where the robot deviated from the path (e.g., at intersections), it successfully realigned itself in **over 90% of instances** through corrective maneuvers.
- **System Robustness:** Performance was stable over repeated trials ($n = 20$ runs per track layout), indicating consistency of the control strategy.

C. Observations

The robot exhibited reliable path-tracking performance under controlled lighting and track conditions. Specific observations include:

- **Straight Path Tracking:** The robot maintained near-perfect alignment, with minimal oscillations after PID tuning.
- **Curves and Sharp Turns:** Slight lateral deviations were noted when navigating curves of radius < 20 cm. PID gain adjustments improved correction speed, reducing overshoot and stabilizing motion.
- **Intersections:** At four-way junctions, decision logic allowed the robot to choose the correct path based on programmed instructions. Occasional hesitation (~200–250 ms) was observed before committing to a turn, attributed to sensor re-calibration at junction edges.
- **Lighting Variations:** Performance remained robust under slight changes in illumination. However, significant glare on the floor surface occasionally interfered with sensor readings, suggesting the need for sensor shielding or adaptive thresholding in future iterations.
- **Mechanical Considerations:** Wheel slippage on smooth surfaces occasionally introduced minor

trajectory errors. Using rubberized tires reduced this effect, improving stability.

6. Applications

The development and testing of an autonomous robot capable of reliable path tracking using sensor-based navigation has broader implications in several domains. The simplicity, cost-effectiveness, and adaptability of such systems make them suitable for both industrial and educational contexts.

A. Warehouse Automation

Modern warehouses demand efficiency in material handling and goods movement. Autonomous robots equipped with sensor-based navigation can be deployed to transport items along predefined routes, reducing dependence on human labor.

- **Path Following:** Robots can move goods from storage shelves to packing stations along fixed tape or painted lines, ensuring collision-free operation.
- **Scalability:** Multiple robots can operate simultaneously on different paths, enhancing throughput.
- **Flexibility:** Low-cost sensor-based robots can be easily reprogrammed when warehouse layouts change, unlike permanently embedded conveyor systems.
  The tested robot's ability to maintain **95% path accuracy** and respond within **150 ms** demonstrates suitability for warehouse environments where speed and precision are crucial.

B. Educational Robotics

Robotics education emphasizes hands-on learning of sensors, control algorithms, and real-world problem solving. Path-tracking robots are widely used in STEM curricula, robotics competitions, and laboratory experiments.

- **Learning Tool:** Students gain practical exposure to sensor integration (IR, ultrasonic), PID tuning, and algorithm design.
- **Low-Cost Implementation:** Affordable components (microcontrollers, sensors, motors) make such systems accessible to schools and universities.
- **Research Platform:** Advanced learners can modify basic path-following robots to implement sensor fusion, vision-based navigation, or AI-driven decision-making.
  The project's modular design makes it well-suited for **educational demonstrations**, helping learners bridge the gap between theory and application.

C. Automated Guided Vehicles (AGVs)

AGVs are widely used in factories, logistics hubs, and hospitals to transport materials efficiently. While high-end AGVs often use LiDAR or vision-based navigation, tape-guided or sensor-based systems remain popular due to their reliability and low maintenance costs.

- **Structured Environments:** The robot can navigate predictable paths with minimal infrastructure changes.
- **Safety and Precision:** Consistent path tracking ensures safe operations around humans and equipment.
- **Cost Efficiency:** Compared to fully autonomous LiDAR-equipped vehicles, sensor-based AGVs are economical, making them attractive for small and medium enterprises.
  The performance metrics obtained in testing (stable 0.5 m/s speed, rapid correction times) align with the requirements of AGVs in controlled industrial environments.

D. Line Inspection Systems

In industries such as manufacturing, energy distribution, and railways, autonomous inspection robots are increasingly used to monitor conditions along fixed routes.

- **Pipeline and Cable Inspection:** Robots following predefined lines can carry sensors for detecting cracks, leaks, or temperature anomalies.
- **Quality Control:** In production plants, robots can be programmed to move along assembly lines, inspecting product quality with cameras or other sensors.
- **Infrastructure Monitoring:** Path-tracking robots can be adapted to inspect rail tracks, conveyor belts, or utility lines, reducing downtime and improving safety.
  The tested robot's ability to handle **sharp turns with PID-tuned correction** shows potential for use in inspection systems that must navigate complex paths with precision.

7. Conclusion and future work

The Automatic Path-Follow Robot developed in this study demonstrates that a **sensor-based navigation system** can provide a reliable, low-cost solution for autonomous path tracking in structured environments. By combining simple infrared sensors for line detection, a PID controller for corrective action, and a microcontroller-based architecture, the robot achieved a path-following accuracy of approximately **95%**, with fast response times and stable operation across straight and curved tracks. Its successful performance under varied layouts highlights the effectiveness of sensor-based navigation as an entry-level approach to autonomy.The strength of the proposed design lies in its affordability, simplicity, and accessibility**.** The system requires minimal hardware investment, is straightforward to assemble, and can be replicated in laboratories, classrooms, or small-scale industrial applications. Although its scope is presently limited to structured environments with clear path markings, it provides a **solid foundation for further innovation** in autonomous robotics.

Future Directions

While the robot fulfills its intended purpose, future research can enhance its adaptability, intelligence, and usability in more complex scenarios:

- Traditional PID control ensures stability, but it struggles in unpredictable conditions (e.g., floor irregularities, worn tape, or environmental noise). Machine learning algorithms such as reinforcement learning or neural-network-based controllers could enable the robot to self-learn corrective behaviors, adapt to different floor textures, and optimize navigation over time.
- At present, the robot relies solely on tape-based navigation without dynamic interaction with its surroundings. Integrating wireless modules (e.g., Wi-Fi, ZigBee, or Bluetooth) would allow real-time monitoring and control from remote stations, enhancing scalability in multi-robot systems. Similarly, adding ultrasonic or LiDAR-based obstacle detection would allow the robot to **avoid unexpected obstructions**, making it suitable for semi-structured industrial settings.
- Sensor-based tracking is robust in controlled conditions but may falter in environments with faded lines, intersections, or variable lighting. Incorporating computer vision (using cameras and OpenCV/AI-based image processing) would enable the robot to detect paths dynamically, recognize symbols or markers, and even navigate in GPS-denied environments such as warehouses and underground facilities. Vision-guided navigationalso opens possibilities for unstructured environments, where physical line marking is impractical.

In conclusion, the developed prototype stands as an effective proof-of-concept for basic autonomous navigation**.** It offers a balance between simplicity and functionality, making it a useful platform for both educational training **and** low-cost automation tasks. With the suggested enhancements, particularly in machine learning, wireless communication, and vision-based navigation, such systems can evolve into more intelligent, flexible, and industrially relevant robotic platforms**.**

References

[1] Bar-Shalom, Y., Li, X. R., & Kirubarajan, T. (2001). *Estimation with applications to tracking and navigation*. Wiley.

[2] Borenstein, J., & Koren, Y. (1991). The vector field histogram—Fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation, 7*(3), 278–288.

[3] Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., & Leonard, J. J. (2016). Past, present, and future of SLAM. *IEEE Transactions on Robotics, 32*(6), 1309–1332.

[4] Singh, N., Sharma, A. K., Sarkar, I., Prabhu, S., & Chadaga, K. (2024). IoT-based greenhouse technologies for enhanced crop production: a comprehensive study of monitoring, control, and communication techniques. Systems Science & Control Engineering, 12(1). https://doi.org/10.1080/21642583.2024.2306825.

[5] Coulter, R. C. (1992). *Implementation of the pure pursuit path tracking algorithm* (CMU-RI-TR-92-01). Carnegie Mellon University, The Robotics Institute.

[6] Dissanayake, G., Newman, P., Clark, S., Durrant-Whyte, H., & Csorba, M. (2001). A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation, 17*(3), 229–241.

[7] Fox, D., Burgard, W., & Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine, 4*(1), 23–33.

[8] Singh, N., Tamrakar, S., Mewada, A., & Gupta, S.K., Artificial Intelligence Techniques in Power Systems Operations and Analysis (1st ed.). 1-72, 2023, CRC(Auerbach) Publications. https://doi.org/10.1201/9781003301820

[9] Hoffmann, G. M., Tomlin, C. J., Montemerlo, M., & Thrun, S. (2007). Autonomous automobile trajectory

tracking for off-road driving: Controller comparisons. In *Proceedings of the American Control Conference* (pp. 2296–2303).

[10] Kanayama, Y., Kimura, Y., Miyazaki, F., & Noguchi, T. (1990). A stable tracking control method for an autonomous mobile robot. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 384–389).