INTERNATIONAL JOURNAL OF ADVANCED RESEARCH
International Peer-Reviewed Research Journal
ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972
Volume 3 • Issue 5 • September 2026

A SURVEY ON LINE FOLLOWING ROBOT: NAVIGATION STRATEGIES, CONTROL ARCHITECTURES, AND APPLICATIONS IN SERVICE ROBOTICS

Vaniya Khan[1], Syed Sohail Ahmed Shah[*2], Bushra Muneer[3], Fatima Muhammad Ali[4], Deepak Sharma[5], Aisha Aslam[6]

[*2]*Assistant Professor, Government College University Hyderabad,*
[1,3,4,5,6] *Students, Government College University Hyderabad,*
Government College University, Hyderabad, Pakistan.

[1]bscs.2023.170@student.gcuh.edu.pk, [*2]sohailahmed.shah@gcuh.edu.pk,
[3]bscs.2023.103@student.gcuh.edu.pk, [4]bscs.2023.097@student.gcuh.edu.pk,
[5]bscs.2023.104@student.gcuh.edu.pk, [6]bscs.2023.127@student.gcuh.edu.pk

DOI: https://doi.org/10.5281/zenodo.23082829

Keywords
Line Following Robot (LFR), Autonomous Guided Vehicle (AGV), PID Control, Deep Learning, Internet of Things (IoT), Ziegler-Nichols Tuning, Restaurant Delivery Robot.

Article History
Received: 01 June 2026
Accepted: 23 August 2026
Published: 29 September 2026

Copyright @Author
Corresponding Author: *
Syed Sohail Ahmed Shah

Abstract
Line-following robots (LFRs) are widely studied mobile robots that are increasingly used in factories, schools, and restaurants. This paper reviews several studies published between 2022 and 2026 to show how LFR sensors, steering systems, parts, and real-world uses have improved. The reviewed research covers a variety of methods, including simple rule-based programming, smooth PID control, and advanced AI models like CNNs and LSTMs. Additionally, the paper looks at real-world uses, such as IoT systems for restaurant delivery robots using the ESP32 chip, as well as virtual testing tracks built using the Unity computer simulation program. The results show that PID control gives the best mix of accuracy and speed for small computer chips, keeping steering errors as low as 1.18 cm and working 43% better than simple ON/OFF steering. Advanced AI methods are great at adapting to different tracks, but they need too much computer power and get confused easily when the room changes. IoT systems help these robots do more, like changing routes on the spot, taking wireless orders, and delivering items in real business settings. This review points out the main design choices, unsolved problems, and future ideas to help people build cheap and useful line-following robots.

1. INTRODUCTION

1.1 Background

The last two decades have seen the field of autonomous mobile robotics make significant strides. The availability of cheap microcontrollers with small sensors and AI models that are efficient for small hardware has fueled this growth. However, in this changing field the Line Following Robot (LFR) remains as a fundamental design. The basic concept of an LFR is the detection and tracking of a contrasting line which is cast on the ground by optical sensors. In principle, it's easy, but in practice, it is applicable to a wide variety of scales from Automated

Guided Vehicles (AGVs) in warehouses to robotics in STEM education and, more recently, robotics for serving meals in restaurants.

In the past few years, the rate at which services are automated has been very high. The International Federation of Robotics estimates that the education robotics market will achieve a market value of EUR 3.1 billion by 2027 [10] . In addition, Brigido and Oliveira's meta-analysis highlights the broader impact of line-following robots across multiple domains [16]. In the hospitality sector, this technology shift is driven by rising wages, persistent labor shortages, and growing customer expectations. Especially restaurants required operational systems that are capable of continuous performance without human fatigue. Line Following Robots (LFRs) are reliable and have affordable hardware and also for service automation, they are of particular interest due to their economic feasibility.

1.2 Problem Statement :

The Line Following Robots (LFRs) have still contains a lot of challenges that needs to be addressed. Although basic ON/OFF control systems are low cost but they generate unwanted motions, which are not suitable for transporting hot liquids, where stability is critical. The Classical PID controllers offer smoother motions, but it is slow and takes time in tuning and often struggle with sharp turns. On the other hand, deep learning techniques offer more flexibility, but need more computational resources. These models often consume a lot of embedded hardware and can be less effective in unpredictable environment and with the unseen data that is not in the training set.

The real world deployment of Robotics is much more complicated than simple experiments, because it must have to deal with several aspects of payload stability, obstacle detection, table identification and wireless order processing.

Our review indicates that the existing studies have focuses on these issues in order to fill these identified gaps, we highlight the necessity to develop a comprehensive analysis, which combines sensing technologies, control strategies, and hardware design, with the trade-offs of performance, cost, and reliability.

1.3 Objectives and Significance:

This study examines the recent developments in Line Following Robots (LFRs) and addresses many gaps found in the previous studies and research that can be a part of an integrated system. We will compare sensing technologies, control strategies and hardware platforms based on the recent literature. The purpose of this analysis is to emphasize the trade-offs between performance and cost, to pinpoint where there are continuing performance limitations, and to recommend future development directions. The talk is meant to be useful for both research and engineers in the area of service robots. We have combine seven papers from peer-reviewed journals and conference papers published between 2022 and 2026 for more timely and critical update on the current state of LFR technologies.

2. Literature Review

2.1 Evolution of Line Following Robot Technology

The earliest Automated Guided Vehicle systems from the 1950s and 1960s, which used electromagnetic induction to follow wire-embedded floor tracks, where the line-following robot's conceptual roots may be found. The substitute of embedded wires with painted or taped surface lines, integrated with introduction of affordable infrared (IR) optical sensors, popularized and enabled compact robotic designs best for academic research. The subsequent growth of open-source microcontroller platforms, In particular, Arduino family introduced by Banzi et al. in 2005, dramatically decrease the entry barrier to LFR development and creative community.

In the early, LFR research focused almost entirely on design of sensor array and basic control logic switching. A typical first-generation system paired a microcontroller with two to five IR sensors

arranged transversely across the robot's undercarriage, which produce binary reflectance signals that the control mapped directly to motor commands through conditional logic. Hasan et al. [1] represent a refined version of this paradigm, which illustrate an Arduino Uno system with a 5-bit digital IR sensor array and an L293D motor driver capable of following multi-directional paths and carrying payloads up to 1,000 grams at a maximum speed of 2.19 km/h. Their performance of work established for the baseline standard of rule-based LFR control and which confirmed the practical viability of the platform for industrial transport work.

2.2 Application in Restaurant and Food Delivery Contexts

The application based on LFR technology, restaurant food delivery which attracts the high interest as labour costs and service efficiency requirement which have been boosted in the hospitality sector. By using a two-level food rack delivery robot, Aisyah et al. [2] developed Arduino Uno and a differential drive configuration, To iteratively design, implement, observe, and refine the system, which follow the action research methodology of Kurt Lewin. Their study highlights: the robot's speed must be very carefully measured relative to its payload to and must avoid path deviation or mechanical stalling, without restriction in unloaded academic demonstrations, so this is the crucial practical finding of research.

A detailed restaurant delivery system, which is more mechanically important, was documented in [3], whereby, an Arduino MEGA-based robot integrated with IR sensors, Light Dependent Resistors (LDRs), and a barcode scanner to get fully automated order delivery which works from kitchen to table. This system presented PWM motor control for smooth differential drive transitions, the mechanical oscillation reduce risks spilling of beverage which is used in simpler designs and a limitation of the sudden on/off switching. Confirmation of table was got through a barcode-triggered LED at each table, when the LDR detected as an optical arrival signal, which is low cost and provide a practical and alternative to active radio-frequency or vision-based table localization.

Rehman [4] proposed a complete functional restaurant delivery system as compared to the reviewed works, made an IoT-enabled waiter robot using the ESP32 microcontroller. Unlike track-constrained designs, this system used an A* graph search algorithm for dynamic path planning in open floor environments and communicated with kitchen staff through a Wi-Fi network, which receiving table dispatch commands in real time. A TCRT5000 IR sensor is innovatively redeploy to detect tray removal at each table, and an automatic return to the kitchen. With a documented payload capacity of 6 kg and a touchscreen interface for customer food selection, So, this system describes a commercial based service robot prototype built from low-cost embedded components.

Other robots for restaurant deliveries have been suggested recently by Gayitri et al. [3], Soon & Abdullah [11], Al-Mamur Sayor et al. [11], Patil et al. [13] and Kaur & Ghelotia [14] wherein they have confirmed the interest in the concept of service automation. In a similar fashion, Asha Wise et al. [15] demonstrated the adaptability of LFR in applications to hospitals.

2.3 Advanced Control: From PID to Deep Learning

PID control is now the top method for making line-following robots move smoothly. It works much better than simple on/off switching. System constantly checks how far the autonomous robot is from the line. Then it uses three parts at the same time to fix the steering: integral part clean the past mistakes so the robot can stays centered, the proportional part fixes the current mistake and the derivative part guesses future mistakes by seeing how fast the robot is deviating .A study by Shahi et al. [5] and his team proved this works best. A setup method is called Ziegler-Nichols Type 2, robot missed the line by an average of just 1.18 cm. It was obvious 43%

improvement over basic on/off control across 200 tests.

Standard PID control has problem with complex, twisting tracks, so researchers experimented with the deep learning artificial intelligence (AI) as an alternative. A study compared three different control methods—standard PID, an LSTM network (which learns from patterns over time), and a CNN network (which looks at pictures)—using the exact same Raspberry Pi 4 computer hardware.The LSTM learned by watching how the PID steered. It successfully copied the PID's behavior just by looking at data from the sensors over time. The CNN looked at small, simple black-and-white pictures of the track. It used a basic AI design called LeNet-5 to choose between eight different driving directions. Because the CNN could see the whole track ahead in pictures, it handled sharp 90-degree turns much better than the PID, which can only see what is directly beneath its infrared sensors. However, the CNN only made the right choice 79.17% of the time. It was also much slower, running at just 24 updates per second compared to the PID's fast 80 updates per second. This slowdown occurred because of taking pictures, preparing them, and running them through the AI take too much computer power[6][9].

Optimization approaches such as Rakhmat & Sukma [8] is also demonstrate about how PID tuning and fuzzy logic can further improve its performance.

2.4 Simulation and Virtual Testing Environments

Computer simulations become a useful tool to build physical robots, which allow engineers to test safely the code and track the shapes without breaking any hardware or spending money to build real tracks. A research used in [7] Unity game engine to make a virtual lab that accurately copied how a car-like robot moves and how its virtual infrared sensors see the ground. They did two driving patterns: 3-command drive and 5-command drive and four track shapes: straight tracks, 90 degree corners, curved tracks. The 5-command mode reduced steering errors up to 65% compared with the 3-command mode on hard tracks, and continued to be on track much better. But the 3 command mode provided a smoother ride and less wobbling. This is the same as a principle of engineering that I know: If a controller reacts more quickly, it corrects errors quickly, but it shakes more. This Simulation demonstrates that there is a balance between steering accuracy and ride smoothness; this way the engineers don't have to make and build several real robots to learn the same lesson.

2.5 Research Gap

Whenever we take a look at the past researchs and that explains four big things which are missing in this field. Firstly, no robot with the smart track-reading has the ability of camera-based AI that merge the fast speed of a PID controller Second, in real time no restaurant delivery robots can automatically change their motor speed or steering strength when the food is added or removed, So , this is a critical real-world need. Third, there is no information about how real customers feel regarding these robots, or how well they are safe to use around crowds in busy places. Fourth, no one has successfully tested that, at the same time multiple robots which works together to deliver food to different tables. These missing parts shows the limitations of current line-following robot research and which explain why we need the new project ideas outlined in Section 7 [16].

3. Methodology

3.1 Research Design

This study uses a systematic literature review, carried out in three clear phases. First, we found and choose the main related sources from well-known databases. During phase 2, the key technical details from each source was identified and assigned to several categories. In phase 3, the extracted data from sources were summarized and analyzed. By combining and analyzing all the relevant information, common trends, key compromises in performance and directions for

future work were identified. All contributions to the survey from the primary sources identified can be verified using standard engineering practices.

3.2 Source Selection Criteria

To provide a meaningful survey, seven major studies were selected for detailed consideration for the following reasons: Firstly, it was necessary to restrict the studies under consideration to those directly pertinent to the design, control, or applications of line-following robots, and secondly, it was necessary to limit the time period of publication (from 2022 to 2026) and to have formal documentation available. International conference papers, peer-reviewed journal articles, preprints on the arXiv and technical reports were the primary sources for the primary studies. A compendium of additional sources was provided to give background information and context to the discussion [1]–[16].

3.3 Data Extraction Framework

The following technical attributes were systematically extracted for the each reviewed source,: microcontroller platform and computational resources, sensor type and configuration, control algorithm and tuning methodology, motor driver and actuation system, application domain and deployment context, quantitative performance metrics where reported, obstacle detection capability, and identification of limitations and suggestions of future work. This fully structured extraction framework supported direct cross-study comparison and served as the foundation for the comparative tables listed in Section 4.

3.4 Tools and Procedures

The reviewed studies mention the tools which consist of wide hardware and software stack. On the hardware side, microcontroller platforms used for the implementation of Arduino Uno (ATmega328P, 16 MHz), Arduino MEGA (ATmega2560), Arduino Nano (ATmega328P), ESP32 (dual-core Xtensa LX6, 240 MHz, integrated Wi-Fi), and Raspberry Pi 4 Model B (Cortex-A72, 1.5 GHz, with NVIDIA RTX 3060 GPU used for offline training)[6][9]. The Sensers components are digital and analog IR sensors (TCRT5000, 5-bit IR arrays), ultrasonic distance sensors (HC-SR04), light dependent resistors (LDR), barcode scanners with USB Host Shield interfaces, and a 5-megapixel camera module (Raspberry Pi Camera Rev 1.3) [3][4]. Motor actuation was implemented through L293D and L298N dual H-bridge motor drivers, and a 4-channel relay module. Software tools referenced are the Arduino IDE (C/C++), Python 3.7.4 with TensorFlow 1.14.0 and Keras 2.2.4 [6], and the Unity game engine for virtual simulation [7].

3.5 Analysis Techniques

The quantitative performance parameters taken from the analyzed studies range from mean tracking error (cm), root mean square error (RMSE), control loop frequency (Hz), obstacle detection reliability (%), classification accuracy (%), payload capacity (kg), maximum operating speed (km/h or m/s), to cost ($ USD). If statistical data was available, like confidence intervals and p-values from Shahi et al. [5] they were included in the comparative analysis. Descriptive characterization was employed for studies providing qualitative results, but not numerical. The virtual simulation study was not a physical study, so it was analyzed independently, in order to provide background information for discussion of the trade-offs necessary for the design of the controllers, but not to serve as directly comparable data of controller performance in a physical setting [7].

4. Results and Discussion

4.1 Overview of Reviewed Systems

Table I summarizes the key attributes of all seven reviewed LFR systems, providing a direct cross-comparison of microcontroller platform, sensor suite, control algorithm, application domain, obstacle avoidance capability, and key performance metrics. This structured overview

forms the empirical foundation for the discussion that follows[1]–[7].

Table I: Comparative Summary of Reviewed LFR Systems

Study / System	Year	Controller	Sensors	Microcontroller	Application	Obstacle Avoidance	Key Performance Metric
Hasan et al. [1]	2022	Rule-Based ON/OFF	5-bit Digital IR Array	Arduino Uno	Industrial Load Carrier	No	Max speed 2.19 km/h; 1 kg payload
Aisyah et al. [2]	2024	Differential Drive	2 IR Sensors Sensors	Arduino Uno R3	Restaurant Food Delivery	No	Speed-load calibration critical
Barcode/LDR Restaurant Robot [3]	2022	PWM Motor Control	IR + LDR + Barcode Scanner	Arduino MEGA	Automated Waiter Robot	No	Smooth turn via PWM; LDR table detection
Rehman [4]	2025	A* Path Planning + IoT	Ultrasonic + TCRT5000 IR	ESP32	IoT Waiter Robot	Yes (Ultrasonic)	6 kg payload; WiFi order dispatch
Shahi et al. [5]	2026	PID (Ziegler-Nichols)	Dual TCRT5000 + HC-SR04	Arduino Nano	Educational Robotics	Yes (96.7%)	1.18 cm error; 43% better than ON/OFF
PID vs DL Study [6]	2023	PID / LSTM / CNN	4-ch IR Array + 5 MP Camera	Raspberry Pi 4	Autonomous Mobile Robot	No	PID: 80 Hz; CNN: 79.17% accuracy
Virtual Simulation Study [7]	2023	3-Mode vs 5-Mode Maneuver	8 Virtual IR Sensors	Unity Simulation	ITS / Pre-deployment Testing	N/A	5-Mode reduces deviation by up to 65%

Table II compares the major control strategies identified across the reviewed studies in terms of tracking precision, update frequency, hardware complexity, path adaptability, and cost suitability.

Table II: Comparison of Control Strategies for Line Following Robots

Control Method	Typical Tracking Error	Update Frequency	Hardware Complexity	Adaptability	Cost Suitability

Control Method	Typical Tracking Error	Update Frequency	Hardware Complexity	Adaptability	Cost Suitability
Rule-Based ON/OFF	2.8 – 3.5 cm	High (>50 Hz)	Low	Low	Very Low Cost
PWM Control	~2.0 – 2.5 cm	High (>50 Hz)	Low–Medium	Low–Medium	Low Cost
PID (Classical)	1.18 – 1.3 cm	High (80 Hz)	Medium	Medium	Low–Medium Cost
LSTM (Deep Learning)	~PID level	Medium (25 Hz)	High	High (sequential)	Medium–High Cost
CNN (Vision-Based)	N/A (classification)	Low (24 Hz)	High	High (geometric)	High Cost
A* Path Planning (IoT)	Global optimal path	Dynamic (event-driven)	Very High	Very High	Medium Cost (ESP32)

4.2 Sensing Hardware Performance

The most popular sensors used in the studies reviewed were infrared (IR) sensors. The analog IR sensors gave a continuous output of reflectances, from which algorithms could be derived to be applied as weighted error corrections, whereas the digital IR sensors gave a simple binary output. The number of sensors used in the studies also differed, from two sensors in Shahi et al. [5] to eight sensors in a simulation study [7] and the most typical number was five (Hasan et al. [1]). The accuracy of tracking is not only a function of the number of sensors but also of the type of control algorithm as shown by Shahi et al. [5].

For obstacle detection there were two studies which included ultrasonic sensors. Shahi et al. [5] found that an HC-SR04 ultrasonic sensor can detect the presence of the object with minimal false positive with 96.7% reliability, through a set of 100 trials. To reduce transient noise, they applied a three-sample debounce filter. The same principle was applied in restaurant navigation as done by Rehman [4] with ultrasonic sensing, where the operating range is up to 21 meters. Besides, Rehman created a TCRT5000 sensor which could be used in a different application than just line tracking, to verify that the tray was removed before the robot moved back to the kitchen.

The third sensing technique used were Light Dependent Resistors (LDRs). An LED combined with an LDR was used in study [3] to create a passive optical sensor, which had the advantage of being less expensive than an active sensor for use in a fixed stop-trigger detection system.

4.3 Control Algorithm Performance

There is literature evidence proving that Proportional-Integral-Derivative (PID) control is effective, and consistently better than using ON/OFF switching alone. Shahi et al. [5] showed that the Ziegler-Nichols tuned PID controller achieved an average tracking error of 1.18 cm, which is 43% less than the ON/OFF logic ($p < 0.001$). The Root Mean Square Error (RMSE) was reduced from 1.84 cm (ON/OFF control) to 0.32 cm (PID) on a curved 10 second track. The results emphasize the capability of the derivative term to counter the sharp overcorrections common to “bang-bang” control.

Different strengths were provided by deep learning approaches. A Convolutional Neural Network (CNN) was able to navigate complex paths with an accuracy of 79.17% but was not

able to navigate at higher speeds for safety margins, reducing the speed of navigation[6][9]. Many CNN errors were associated with the ambiguous manual labels at transition points. A model based on Long Short Term Memory (LSTM) was trained to predict motor outputs comparable to those of PID by training a model of the relationships between sequential errors and speeds.

Another simulation study [7] more fully demonstrated some of the aforementioned trade-offs between accuracy and smoothness. On complex routes a five-maneuver mode resulted in up to 65% less lateral deviation than a three-maneuver mode. But repeated fine adjustments resulted in a noticeable wobbling of the ball around the mark. Both operation modes stayed within the 0.2 m autonomous driving limit on straight tracks. In one instance the three maneuver mode went over this limit on twisty routes, the five maneuver mode was always under the limit.

Optimization studies such as Rakhmat & Sukma [8] has further confirmed that the PID tuning combined with fuzzy logic can enhance performance of food delivery.

4.4 Hardware Platform Analysis

Microcontrollers like Uno, MEGA, and Nano are the Arduino family, dominated the studies which are reviewed for rule-based and PID control applications, low cost, extensive community support, and sufficient I/O for sensor arrays and PWM motor control. So, The Arduino Nano used in the Line Master Pro (total system cost USD $28.50) get a 94% cost reduction relative to commercial educational robotics kits, demonstrating that high-quality PID control can be used at very low cost with careful component selection. The Arduino MEGA was selected for the barcode guided restaurant robot specifically for its 54 digital I/O pins, which accommodated the USB Host Shield for barcode scanning alongside motor and sensor interfaces [3].

In Rehman's IoT system [4] the ESP32 microcontroller is used, explained a qualitatively more capable platform with dual-core 240 MHz processing, which is integrated with Wi-Fi and Bluetooth, and hardware-accelerated floating-point arithmetic. These capabilities allowed for the A* path planning algorithm, the ability to receive orders wirelessly in real-time, and the ability to manage multiple sensors from a single microcontroller as opposed to multiple coordinated Arduino boards trying to approximate the same effect. Raspberry Pi 4 in deep learning comparative study [6] also provided the highest computational resources, 1.5 GHz quad-core processor, Linux OS, and GPU-accelerated TensorFlow, which allowed real-time CNN inference on the Raspberry Pi that would not be possible with the microcontroller-class devices, but with higher power consumption and larger physical size.

4.5 Application Context and Load Performance

Load carrying capacity and its impact on speed and stability was a common constraint in the design of the restaurant delivery systems. Although Hasan et al. [1] showed maximum speed dropped from 2.19 km/h for zero load to 1.87 km/h for 1 kg; in Rehman's system [4] the navigation function was not lost at 6 kg and the speed dropped from 2.17 m covered in 20 s at 2 kg to 1.50 m at 6 kg, while current draw increased from 1.68 mA to 1.91 mA. Aisyah et al. [2] found that the speed should be adjusted manually for various loads in order to avoid path deviation. Closed-loop adaptive load compensation was not found to be implemented by any of the reviewed systems, which is a significant difference between current implementations and a production-ready delivery robot.

The two systems mentioned above, one system based on the LDR in [3] and the second on the TCRT5000 in [4] are both practical innovations, which have extended the use of the line following algorithm to take care of the whole operation of a restaurant delivery system: departure from kitchen, navigation to table, confirmation of the delivery, and return. These are not present in the

more basic academic LFR studies that only consider path following performance without the context of the operation. Rehman's system also features a touchscreen menu interface that serves to further the human-robot interaction aspect, though no quantitative data for customer satisfaction was reported [3][11][12][13][14].

4.6 Comparison with Prior Literature

The reviewed studies contribute to the advancement of line-following robot research in several important areas. Shahi et al. [5] is more rigorous than many previous PID LFR studies that have reported qualitative observations instead of the more statistically thorough comparisons. Most earlier deep learning robotics studies relied on more powerful hardware than PID-based approaches. The comparison presented in [6] provides a direct evaluation of the computational cost of deep learning by testing PID, LSTM, and CNN methods on the same platform. Rehman's [4] implementation of A* planning on a $50 ESP32 system proves that the complex navigation algorithms that have been known using high cost LiDAR-based systems can also be implemented on these micro controllers. The virtual simulation architecture presented in [7] extends the use of Unity as a pre-deployment testing tool beyond the previously mentioned simulations of the more expensive system.

5. Challenges and Limitations

Despite its tremendous development, LFRs still have a number of unsolved challenges that are important to be addressed to enable line-following robots' use in real world application other than laboratory and test environments. One problem with infrared (IR) based sensing systems is that they are very sensitive to ambient light. Bright overhead lights or direct sunlight will quickly overload the photodetectors, and therefore lead to incorrect readings and the robot becoming unusable.

Similarly, with vision based systems like CNNs, the same applies; often surface reflection and room light are set by hand .In particular, it was found that performance of the Line Master Pro system becomes significantly worse when the light levels exceed 50,000 lux [5].

All the above problems demonstrate the need for more powerful solutions such as adaptive thresholding algorithms, optical filtering or other sensors such as time of flight to ensure reliable operation .Cheap embedded systems' insufficient processing capability is one of the main problems[8][16].

Due to this performance difference, most industrial applications still use conventional PID controllers. One thing that still continues to be a problem is the load predictability, especially for food delivery services to restaurants. The robot's speed will slow down as it loads more weight, and a greater risk of it getting off course. Unfortunately, none of the methods presented in this book provide any active compensation in the case of real-time varying loads. This restriction might negatively impact the service's dependability and efficacy in real-world situations, particularly when meal orders are of different weights.

The application of deep learning technologies is limited by data related problems. Often models are trained with relatively small, environment-specific examples; for instance, one CNN was trained with only 1,800 hand-labeled photos of one track. This results in poor transfer to new layout/track configurations. In addition, moderate F1-score 0.7360 was obtained due to inconsistencies in the hand labeling, especially at the transitions. By contrast, the traditional control systems do not need any training data, which is another advantage that they have in terms of simplicity and flexibility.

Lastly, for IR based system, constraints in track layout are difficult. The fact that minimum turning radius is 15cm (which is 3cm from the sensors) and it is hard to be able to take sharp 90 degree turns makes it impossible for CNN's to achieve a reliability of 100%. As a result, these robots tend to demand considerable re-organization of the restaurant floor layouts, as pre-laid tracks are often needed to position the

robot correctly, which limits flexibility and increases the cost of deployment. In order to make LFR viable beyond the prototype phase, we need to tackle this bundle of issues.

6. Future Research Directions

The problems identified stimulate future research directions frequently suggested by the authors themselves. The addition of Simultaneous Localization and Mapping (SLAM) is a paradigm-changing upgrade to service-oriented LFR systems, where no fixed tracks are required [4]. For deployable, for free-navigation systems, Rehman [4] notes that SLAM integration with lightweight planners, such as A*, is a key goal on device such as ESP32 or RASPBERRY Pi 4, is important.

The architecture that has emerged as a near research direction and is important is "hybrid control", which is the combination of the fast calculating PID and the geometrically flexible CNN-based vision. Practical design uses the PID controller as tracking method on regular path corners, only by using CNN inference where the IR array senses a potential line loss. The strategy enhances the strength of each method while reducing their weakness; the geometrical discontinuity is not a strong point of the PID, and the simple segments are not a strong point of the CNN. The comparative study [6] indicates this mixed approach as a prospective future one, which is what the deep learning does.The Transfer learning and synthetic data generation can drastically lower the data dependency of deep learning LFR systems. Rather than training models from scratch on small, manually labeled real-world datasets, future research could fine-tune pre-trained convolutional models on LFR-specific imagery. Alternatively, researchers could use the Unity virtual environment to generate large, diverse, and automatically labeled training datasets. Synthetic-to-real transfer is already proven in complex robotics perception tasks and offers a practical path toward generalizable LFR vision systems.

Delivery robots often slow down, steer poorly, or struggle to turn when they are carrying heavy packages. To fix this, robots can use their own electric motors as a built-in scale. When a package is heavy, the motors have to work harder and draw more electricity. By measuring this electrical current, the robot instantly figures out how heavy its cargo is. It then automatically adjusts its steering math and motor speeds to match the weight. For even better accuracy, the robot can use an IMU sensor—the same type of sensor that lets your smartphone know when you rotate it. This sensor tracks the robot's position which, provide the extra movement of data needed to keep the robot driving smoothly and safely, whether it is carrying an empty or heavy box.

By Managing a group of food-delivery robots in a large restaurant requires an efficient central system which keep everything moving. This central server connects to all the robots using ESP32 chips which is cheap, wireless microchips . When customers order food, the server instantly decides which robot should deliver to which table to get the food out as fast as possible. By keeping the restaurant running smoothly, this central system solves three major challenges: it prevent the robots from crashing into each other in narrow pathways, and automatically assign new delivery tasks as orders come in, and schedule battery recharging breaks so no robot runs out of power in the middle of the dining room. By Solving these problems, allows a large group to work together easily and serve multiple tables at the same time.

Delivery robots can be used in real restaurants, researchers must study how they interact with real people. Nowadays, engineering studies ignores how everyday people feel about and interact with these machines. Before these robots can be sold commercially, researchers from different fields must work together to test three key things. First, they need to take customer surveys to see if people actually enjoy being served by a machine or if it makes them uncomfortable. Second, they should prove the robots are

completely safe when they are driving through crowded rooms full of walking adults, unpredictable kids, and busy waiters. Finally, they need to test the robot's screens and buttons to make sure they are easy and comfortable for people of all ages and heights to use.

7. Conclusion

Seven primary studies were selected for detailed comparative analysis, while additional references were included to provide background and contextual support, to address various aspects from basic to advanced AI. Collectively, these studies explain that line-following technology has evolved into a practical tool for use in hospitality settings. The main take-home message is that there is no universal optimal design; that is, each strategy has its own set of compromises. The traditional PID control method has lower costs, higher accuracy, and a deviation of only a few degrees on a relatively simple path[5][8]. while the advanced Deep learning approaches such as CNNs and LSTMs [6][9] provide the more adaptability to complex path, but requires more computational resources and a large amount of training data. While, Internet-enabled robots can move around in crowded, changing environments with complex code, and virtual simulations enable engineers to test their software in video games without creating physical robots. The missing pieces are needed now to make sensors work in challenging lighting, to assist robots to adapt heavy loads, to train AI to cope with new rooms, and to test people's reactions in the vicinity of the robots. Addressing these limitations will require integrating optimization techniques [8], advanced sensing, and hybrid control strategies that combine the speed of PID with the adaptability of AI.

REFERENCES

[1] M. M. Hasan, S. Shahria, M. H. Ullah, and M. M. Rahman, "Building an Industrial Autonomous Line-Following Robot through the Design and Construction Processes," in Proc. 5th Int. Conf. on Industrial & Mechanical Engineering and Operations Management (IEOM), Dhaka, Bangladesh, Dec. 26–28, 2022, pp. 390–397.

[2] R. Aisyah, A. Latifah, E. P. Hidayatulloh, Nurhidayat, J. Kifti, M. F. Sodikin, and B. N. Alfiyah, "Implementation of Line Follower Robot for Food Delivery Robot," Journal of Electrical, Electronic, Information, and Communication Technology (JEEICT), vol. 6, no. 2, pp. 55–58, Oct. 2024, doi: 10.20961/jeeict.6.2.92211.

[3] H. M. Gayitri, S. G. Adhikari, and J. R. Kumar, "Smart Robot for Restaurant Food Delivery," Mathematical Statistician and Engineering Applications, vol. 71, no. 4, pp. 2747–2753, 2022, doi: 10.17762/msea.v71i4.833.

[4] S. U. Rehman, "An IoT-Based Autonomous Waiter Robot," Memoria Investigaciones en Ingeniería, núm. 28, pp. 154–167, 2025, doi: 10.36561/ING.28.11.

[5] J. Shahi, A. Shah, and A. S. M. Ahsanul Sarkar Akib, "LineMaster Pro: A Low-Cost Intelligent Line Following Robot with PID Control and Ultrasonic Obstacle Avoidance for Educational Robotics," arXiv preprint arXiv:2603.13907v1, Mar. 14, 2026.

[6] H. M. Leal, R. S. Barbosa, and I. S. Jesus, "Control of a Mobile Line-Following Robot Using Neural Networks," Algorithms, vol. 18, no. 1, p. 51, Jan. 2025, doi: 10.3390/a18010051.

[7] M. Bakirci, "Simulation of Autonomous Driving for a Line-Following Robotic Vehicle: Determining the Optimal Manoeuvring Mode," Elektronika ir Elektrotechnika, vol. 29, no. 6, pp. 4–11, Dec. 2023.

[8] G. A. Rakhmat and M. R. Sukma, "Optimization of Food Delivery Robot Using the PID Method Ziegler Nichols Type 2 and Fuzzy Logic Mamdani," E3S Web of Conferences, vol. 484, p. 01020, 2024.

[9] C. Minaya, R. Rosero, M. Zambrano, and P. Catota, "Application of Multilayer Neural Networks for Controlling a Line-Following Robot," Journal of Automation, Mobile Robotics and Intelligent Systems, vol. 18, no. 1, pp. 35–42, 2024.

[10] International Federation of Robotics, "World Robotics 2023 — Service Robots," IFR Press Conference, Frankfurt, Germany, 2023.

[11] J. L. J. Soon and A. R. bin Abdullah, "Development of Restaurant Waiter Assistant Line Follower Robot," International Journal of Electrical Engineering and Applied Sciences (IJEEAS), vol. 1, no. 1, pp. 1–7, Sep. 2021.

[12] J. J. Al-Mamur Sayor, N. T. Shishir, B. B. Barmon, S. Ahemed, Nurjahan, and M. Whaiduzzaman, "RoboServe: Design and Implementation of a Robotic Agent for Serving Food at Restaurants," in Proc. 3rd Int. Conf. on Computing Advancements (ICCA 2024), Dhaka, Bangladesh, Oct. 17–18, 2024, pp. 588–597, doi: 10.1145/3723178.3723256.

[13] L. Patil, C. Wadbudhe, P. Negi, P. Kale, D. Agrawal, A. Fule, and G. Saryakar, "Computerized Serving Robot," International Journal of Engineering Research and Technology (IJERT), vol. 12, no. 2, pp. 59–63, Feb. 2023.

[14] H. Kaur and S. Ghelotia, "Line Following Humanoid Robot for Restaurant Applications," International Journal of Creative Research Thoughts (IJCRT), vol. 12, no. 4, pp. l231–l235, Apr. 2024, ISSN: 2320-2882.

[15] C. T. Asha Wise, S. Fouziya Sulthana, and A. Sundaramoorthy, "Line Follower Robot based Room Service Automation in Hospital Environment," AIP Conference Proceedings, vol. 2460, p. 020002, Aug. 2022, doi: 10.1063/5.0097033.

[16] W. J. H. Brigido and J. M. Parente de Oliveira, "The Line Follower Robot: A Meta-Analytic Approach," PeerJ Computer Science, vol. 11, p. e2744, Mar. 2025, doi: 10.7717/peerj-cs.2744.

International Journal of Advanced Research