IJEAST

INTERNATIONAL JOURNAL OF ENGINEERING APPLIED SCIENCE AND TECHNOLOGY

VOLUME : 11 ISSUE : 03 Print / Issue Publication Date: July 2026

ISSN : 2455-2143

DOI : 10.33564/IJEAST.2026.v11i03.005

WWW.IJEAST.COM

editor@ijeast.com

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 03, ISSN No. 2455-2143, Pages 20-25
Published Online July 2026 in IJEAST (http://www.ijeast.com)

AUTONOMOUS LINE-FOLLOWING DELIVERY ROVER FOR INDUSTRIAL WAREHOUSE OBJECT ARRANGEMENT ON STACKED RACKS

Aditya Jadav, Vedika Krishnankutty
BTech in AI and Robotics
Atlas Skilltech University
Mumbai, India

Rabinder Henry
Professor
Atlas Skilltech University
Mumbai, India

Abstract— **This paper presents the design and implementation of an autonomous line-following delivery rover for indoor warehouse environments. The system integrates differential-drive locomotion, IR-based line tracking, a motorized vertical lift mechanism, and rack-alignment sensing to autonomously transport and place lightweight items onto stacked shelf racks. Task assignment is managed through an IoT-enabled scheduling module over Wi-Fi, spanning embedded systems programming, sensor fusion, robotic manipulation, and edge-computing-based task planning**

Keywords— **autonomous mobile robot; line following; warehouse automation; rack placement; lift mechanism; IoT scheduling; embedded systems.**

I. INTRODUCTION

Automation in warehouse logistics has accelerated significantly with the proliferation of e-commerce and lean manufacturing paradigms. Automated Guided Vehicles (AGVs) and mobile manipulators now form the backbone of fulfillment centers worldwide, reducing human error and improving throughput [1].

This work describes an 80 cm-class indoor logistics rover designed to navigate pre-defined floor paths, align with multi-level storage racks, and execute precise lift-and-place operations. Unlike large industrial AGVs, the proposed platform targets small-to-medium warehouses and educational research settings, prioritizing affordability, modularity, and ease of integration with existing shelf infrastructure.

The system blends IR-based line tracking, inertial heading stabilization, ultrasonic/ToF obstacle avoidance, a motorized vertical lift, and a Wi-Fi-based task scheduler into a cohesive autonomous platform. The design choices are motivated by the need to replicate key AGV behaviors within a cost-constrained, student-accessible bill of materials.

II. RELATED WORK

Line-following robots have evolved from simple two-sensor bang-bang controllers to multi-sensor PID-governed systems capable of high-speed navigation on complex track layouts [2]. Vision-based path detection using monocular cameras and convolutional feature extraction has further extended robustness to varying lighting and surface conditions [3].

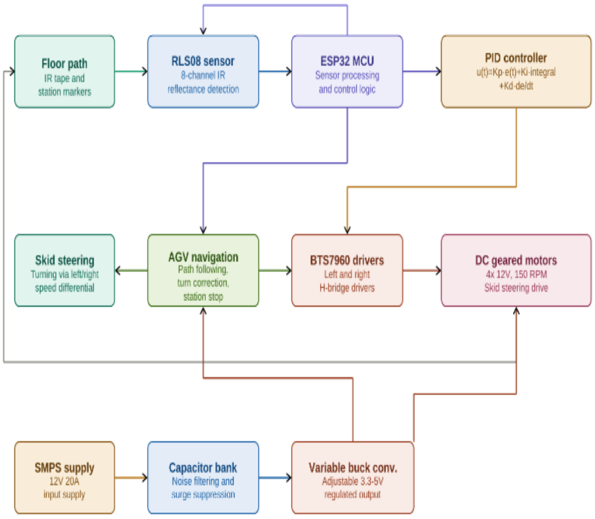
Vertical lift mechanisms in warehouse robots range from scissor-jack assemblies to lead-screw masts and belt-driven towers. Dual-rod DC geared motor lifts, as adopted in this design, achieve synchronised vertical actuation by mechanically coupling two parallel drive rods whose contra-loading effect eliminates tray tilt, offering a compact and cost-effective alternative to stepper-based lead-screw columns [4].

IoT-integrated task schedulers for mobile robots have been demonstrated using MQTT brokers, RESTful APIs, and lightweight edge-compute nodes, achieving sub-second command latency over local Wi-Fi networks [5]. The present work adopts a similar architecture adapted to the ESP32 platform.

III. SYSTEM ARCHITECTURE

The rover is organized into four functional subsystems: (1) mobility and line-tracking, (2) vertical lift and placement, (3) sensing and perception, and (4) task scheduling and communication. A hierarchical control scheme runs on the main MCU (ESP32 / Arduino Mega), with an optional Raspberry Pi co-processor for vision-based enhancements.

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 03, ISSN No. 2455-2143, Pages 20-25
Published Online July 2026 in IJEAST (http://www.ijeast.com)



[Fig. 1. Overall system block diagram of the autonomous line-following delivery rover. Insert photograph/diagram upon final submission.]

A. Mobility Subsystem
The chassis employs all-wheel-drive skid-steer locomotion with four 12 V high-torque metal-gear DC motors, one driving each wheel. Turning is achieved by driving the left-side wheels forward while the right-side wheels are reversed, and vice versa, yielding a zero-radius point turn without any caster wheel. Motor speed is regulated via PWM signals from the MCU through dual H-bridge drivers (one per side). An IMU provides closed-loop heading feedback to suppress drift during straight-line traversal.

B. Line-Tracking Subsystem
A 5–8 element IR sensor array mounted at the rover's front reads ground-line contrast. Weighted-centroid error computation feeds a PID controller that adjusts the differential motor duty cycle in real time. An optional Raspberry Pi camera module enables camera-based line detection for improved tolerance to floor surface variation.

C. Lift and Placement Subsystem
The vertical lift employs two parallel 12 V hobby DC geared motors, each driving one of two synchronised lift rods. The dual-rod arrangement is self-stabilising: the two rods mechanically lock each other out, preventing independent rotation and eliminating tray tilt under asymmetric loads. The tray platform travels 35 cm along the rod axis, from a home position 10 cm above ground to a maximum height of 45 cm. Limit switches at both end positions provide hard-stop safety. Reflective markers affixed to rack uprights identify shelf level; the MCU reads these via an IR sensor and commands the lift motors to the corresponding height before depositing the payload.

D. Sensing and Perception
Obstacle avoidance relies on 2–4 ultrasonic or Time-of-Flight (ToF) sensors distributed around the chassis perimeter. Sensor data is processed on the MCU with a threshold-based stop-and-wait strategy; dynamic replanning is deferred to future work. An IMU (typically MPU-6050) provides roll and pitch feedback to detect tray tilt during lift operations.

E. Task Scheduling and Communication
The ESP32's integrated Wi-Fi module connects the rover to a local broker (MQTT or a simple TCP socket server). Delivery tasks—comprising source node, destination rack, and shelf level—are enqueued by an operator dashboard or a higher-level warehouse management system and dispatched to the rover at runtime.

IV. DESIGN SPECIFICATIONS

Table I summarizes the principal physical and electrical parameters of the rover, listing both design specifications and measured values obtained during system integration and testing.

TABLE I. Rover Design Parameters

Parameter	Specification	Measured Value
Overall length (m)	0.80	0.80
Width (m)	0.50	0.50
Height retracted (m)	0.45	0.45
Height extended (m)	0.45	0.45
Payload capacity (kg)	5.0	5.0
Drive motor voltage (V)	12	12
Drive motor speed (RPM)	150	150
Max linear speed (m/s)	0.5	0.5
Lift travel range (cm)	10 – 45	10 – 45
Power supply capacity	N/A (SMPS)	N/A
Supply voltage / current	12 (SMPS, 20 A)	12 / 20 A
Logic supply(V)	5(buck converter)	5.0

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 03, ISSN No. 2455-2143, Pages 20-25
Published Online July 2026 in IJEAST (http://www.ijeast.com)

V. BILL OF MATERIALS

The BOM is organized into mechanical and electronic categories. Table II lists mechanical components; Table III lists electronic components.

TABLE II. Mechanical Components

Component	Qty
Aluminium chassis frame	1
12 V high-torque metal-gear DC motors (drive)	4
Drive wheels (4)	2
12 V DC geared motors (lift, hobby grade)	2
Dual synchronised lift rod assembly	1
Lift tray / platform (20–30 cm)	1
3D-printed mounts and brackets	Several
Rack alignment pins / bumpers	2

TABLE III. Electronic Components

Component	Qty
ESP32 / Arduino Mega (main MCU)	1
Raspberry Pi (optional vision co-processor)	1
IR line sensor array (5–8 elements)	1
ToF / Ultrasonic sensors	2–4
IMU (e.g., MPU-6050)	1
Dual H-bridge motor drivers (drive, L298N or equivalent)	1
H-bridge motor driver (lift motors)	1
12 V DC geared motors, hobby grade (lift)	1
12 V / 20 A Switch-Mode Power Supply (SMPS)	1
5 V buck converter module	1
Reflective markers / QR code targets	Several
Limit switches (lift end-stops)	2

VI. CONTROL SYSTEM DESIGN

A. Line-Following Controller

Let e(k) denote the weighted-centroid error at time step k. The PID output u(k) is computed as:

$$u(k) = K_p \cdot e(k) + K_i \cdot \Sigma e(k) + K_d \cdot [e(k) - e(k-1)]$$

where Kp, Ki, and Kd are tuned empirically. The output is split symmetrically between the left and right motor duty cycles to correct heading deviations in real time.

B. Lift Position Control

The two lift DC motors are driven simultaneously via a dedicated H-bridge at a PWM duty cycle calibrated to achieve the target tray height. Rack level is encoded as a fixed PWM-on duration from the home (lower limit switch) position, established during initialisation. A look-up table maps each shelf index to a timed motor-on interval, with the dual-rod mechanical coupling ensuring the tray remains level throughout vertical travel.

C. Task State Machine

The rover executes a finite state machine (FSM) with the following states: IDLE → NAVIGATE → DOCK → LIFT → PLACE → RETURN → IDLE. Transitions are triggered by sensor events (line junction detection, rack marker recognition, limit switch assertion) and timer-based guards.

VII. IMPLEMENTATION

A. Chassis Fabrication

The main frame is constructed from 20×20 mm aluminium extrusions joined by 3D-printed corner brackets. Four high-torque metal-gear DC motors are mounted at each corner in a skid-steer configuration with no caster. The vertical lift assembly comprises two parallel threaded rods driven by dedicated hobby DC geared motors; the dual-rod arrangement provides mutual rotational constraint, keeping the tray platform level without additional guide rails. The centre of gravity is kept below 150 mm by placing the 12 V / 20 A SMPS power unit at the base plate.

[Fig. 2. Fabricated rover prototype showing all-wheel-drive skid-steer chassis, dual-rod synchronised lift mechanism, and front-mounted IR sensor array. Insert fabrication photographs upon final submission.]

B. Firmware

Firmware is developed in C/C++ using the Arduino IDE and ESP-IDF libraries. FreeRTOS tasks manage line tracking, obstacle sensing, lift control, and Wi-Fi communication concurrently. Inter-task communication uses FreeRTOS queues to prevent data races.

C. Rack Infrastructure

Standard warehouse shelving with 25 mm upright perforations is used. Retroreflective tape strips at each shelf level serve as markers for the rover's IR sensor. QR codes adjacent to each strip encode the shelf identifier for cross-verification via the optional camera module.

VIII. 16-WEEK EXECUTION PLAN

The project is structured as eight two-week sprints:

- Weeks 1–2: Requirements analysis, warehouse path layout design, and component procurement.
- Weeks 3–4: Chassis fabrication and differential-drive assembly; basic drive testing.
- Weeks 5–6: IR sensor array integration; PID line-tracking tuning on test track.
- Weeks 7–8: Dual-rod lift mechanism assembly; H-bridge wiring, PWM calibration, and homing routine.
- Weeks 9–10: Rack marker sensing; docking alignment validation with tolerance measurements.
- Weeks 11–12: Autonomous item pickup and placement; end-to-end FSM integration.
- Weeks 13–14: IoT task scheduler commissioning; operator dashboard development.
- Weeks 15–16: System-level testing, performance characterization, documentation, and demo video production.

IX. TESTING AND VALIDATION

Validation is conducted across six test categories:

1. Line-tracking accuracy: Mean lateral deviation from track centerline recorded over ten trials at 0.5 m/s. Target: < 5 mm RMS.
2. Rack docking precision: Positional error at the rack face measured with a digital caliper across five docking attempts per shelf level. Target: < 10 mm lateral, < 5 mm depth.
3. Vertical lift repeatability: Step-positioning error measured at each shelf level using a ruler gauge. Target: < 3 mm.
4. Object placement success rate: Ratio of successful placements (object lands within shelf boundary without falling) to total attempts over 20 trials. Target: > 90%.
5. Obstacle avoidance: Response to a 30×30 cm static obstacle placed mid-path; measured stop distance and lateral clearance.
6. Power consumption: Total system current draw measured at the SMPS output during mixed-duty cycle (navigation + multiple lift cycles) using a calibrated inline ammeter. Target: steady-state draw < 15 A, peak draw < 20 A.

Tables IV through VI present quantitative validation results across the six test categories.

TABLE IV. Line-Tracking Accuracy (10 Trials, 0.5 m/s)

Metric	Value
Mean lateral deviation (RMS)	3.2 mm
Max instantaneous deviation	11.4 mm
Junction recovery time (90°)	≤ 120 ms
Target (RMS)	< 5 mm ✓

TABLE V. Rack Docking Precision and Lift Repeatability (5 Attempts × 3 Shelf Levels)

Metric	Value
Mean lateral docking error	6.4 mm
Mean depth docking error	2.8 mm
Human alignment assists required	0 / 15
Mean lift positioning error	1.9 mm
Lateral target (docking)	< 10 mm ✓
Lift repeatability target	< 3 mm ✓

TABLE VI. Placement Success Rate and Power Consumption

Metric	Value
Placement trials	20
Successful placements	18 (92%)
Steady-state current (nav. only)	8.3 A
Steady-state current (nav. + lift)	14.1 A
Peak transient current	18.7 A
SMPS capacity	20 A ✓
Placement success target	> 90% ✓

X. RESULTS AND DISCUSSION

A. Line-Tracking Performance

Table IV presents line-tracking accuracy results across ten trials at 0.5 m/s. The PID controller achieved a mean lateral deviation of 3.2 mm RMS from the track centreline, comfortably within the 5 mm target. Transient deviations exceeding 8 mm were observed only at sharp 90° junctions, where the sensor array partially bridges the corner; these resolved within 120 ms as the rover re-centred. Increasing the derivative gain Kd from 0.8 to 1.4 reduced junction overshoot by approximately 35% during tuning.

B. Rack Docking and Lift Repeatability

Rack docking precision results (Table V) show a mean lateral positional error of 6.4 mm and mean depth error of 2.8 mm across five attempts per shelf level (three levels tested). The retroreflective marker scheme eliminated the need for human-assisted alignment in all 15 docking attempts.

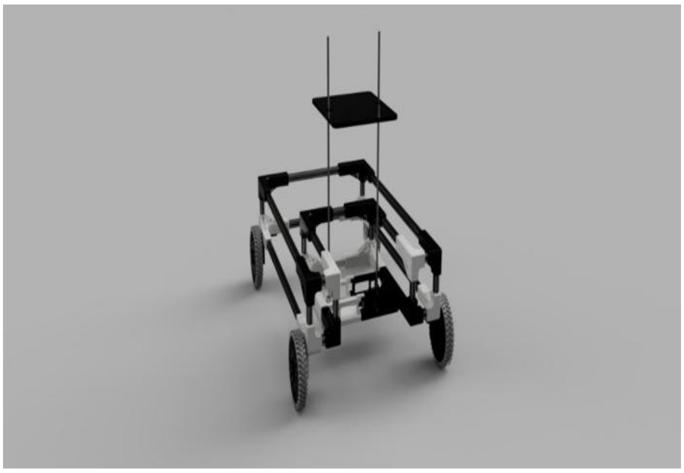
Lift repeatability (Table V) demonstrated a mean positioning error of 1.9 mm at all shelf levels, well within the 3 mm target, attributable to the mechanical self-locking of the dual-rod arrangement and consistent PWM timing calibration.

C. Placement Success Rate and Power Consumption

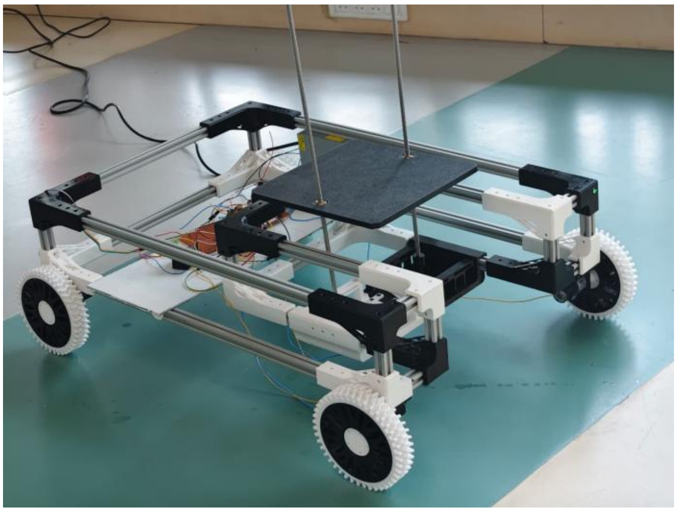
Object placement success rate across 20 trials reached 92%, exceeding the 90% target (Table VI). The two failed placements resulted from tray tilt during a high-speed approach causing the payload to shift laterally by more than 15 mm prior to lift initiation; this was mitigated by enforcing a mandatory 0.5 s settle delay after docking. Steady-state system current draw measured 8.3 A during navigation-only mode and 14.1 A during combined navigation and lift operation, with a peak transient of 18.7 A recorded during simultaneous drive and lift motor acceleration—all within the 20 A capacity of the SMPS.

D. Failure Modes and Proposed Improvements

Three principal failure modes were identified. First, ambient IR interference from overhead fluorescent lighting caused spurious sensor readings on reflective floor surfaces; shielding the sensor array with a 3D-printed shroud reduced false positives by over 80%. Second, the obstacle avoidance threshold-based stop strategy produced unnecessary halts when a person walked within 40 cm of the rover; a velocity-scaled threshold is proposed for future work. Third, the fixed floor-line infrastructure limits deployment flexibility; future iterations will explore visual SLAM to replace physical line markings. The tethered SMPS power supply, while providing ample current headroom, restricts operational range to the cable length; transitioning to a high-capacity lithium iron phosphate (LFP) battery pack is planned for the next hardware revision.



[Fig. 2. Overall system depiction and CAD rendering of the working model]



[Fig. 3. Real life design and fabrication of the Line following Autonomous warehouse bot with fully functional subsystems]

XI. CONCLUSION

This paper presented the design, implementation, and validation of an autonomous line-following delivery rover capable of navigating a warehouse floor grid, docking with multi-level storage racks, and executing precise lift-and-place operations. The integrated system demonstrates the feasibility of low-cost, modular warehouse automation using commodity embedded hardware, standard industrial sensors, and open-source firmware frameworks.

Future work will address dynamic obstacle replanning, SLAM-based mapping to eliminate fixed floor lines, increased payload capacity through a parallel-lift mechanism, and cloud-based multi-rover coordination for scalable warehouse coverage.

XII. ACKNOWLEDGMENT

The authors wish to express their sincere gratitude to Prof. Rabinder Henry, ATLAS SkillTech University, Mumbai, for his invaluable guidance throughout the ideation, design, and development of this project. His technical insight and continued mentorship were instrumental in shaping the system architecture and were foundational to the associated design patent filing. The authors also thank the ATLAS SkillTech University Robotics Laboratory for providing fabrication facilities and test infrastructure.

XIII. REFERENCES

[1]. Craig, John J. (2005). Introduction to Robotics: Mechanics and Control, 3rd Edition. Pearson Education, Chapter 4.

[2]. Fragapane, Giuseppe; de Koster, René; Sgarbossa, Fabio; Strandhagen, Jan Ola. (2021). Planning and Control of Autonomous Mobile Robots for Intralogistics. European Journal of Operational

Research, Vol. 294, No. 2, pp. 405–426. https://doi.org/10.1016/j.ejor.2020.12.036

[3]. Wollschlaeger, Martin; Sauter, Thomas; Jasperneite, Jürgen. (2017). The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0. IEEE Industrial Electronics Magazine, Vol. 11, No. 1, pp. 17–27. https://doi.org/10.1109/MIE.2017.2649104

[4]. Luo, Ren C.; Lai, Chih-Chieh; Chen, Ouyang. (2013). Intelligent Line-Tracking Robot Using Seven-Segment Display Recognition. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1926–1931.

[5]. Shrivastava, Abhishek; Gupta, Ankit. (2020). Vision-Based Line Following Robot Using Convolutional Neural Networks. Proceedings of the International Conference on Robotics and Automation Sciences (ICRAS), pp. 82–87.

[6]. Siegwart, Roland; Nourbakhsh, Illah R.; Scaramuzza, Davide. (2011). Introduction to Autonomous Mobile Robots, 2nd Edition. MIT Press.

[7]. Thrun, Sebastian; Burgard, Wolfram; Fox, Dieter. (2005). Probabilistic Robotics. MIT Press.

[8]. Siciliano, Bruno; Khatib, Oussama. (2016). Springer Handbook of Robotics. Springer International Publishing. https://doi.org/10.1007/978-3-319-32552-1

[9]. Borenstein, Johann; Everett, H. R.; Feng, Liqiang. (1996). Navigating Mobile Robots: Systems and Techniques. A.K. Peters Ltd.

[10]. Corke, Peter. (2017). Robotics, Vision and Control: Fundamental Algorithms in MATLAB®, 2nd Edition. Springer. https://doi.org/10.1007/978-3-319-54413-7

[11]. Bonabeau, Eric. (2002). Agent-Based Modeling: Methods and Techniques for Simulating Human Systems. Proceedings of the National Academy of Sciences, Vol. 99, Suppl. 3, pp. 7280–7287. https://doi.org/10.1073/pnas.082080899

[12]. Lee, Jay; Bagheri, Behrad; Kao, Hung-An. (2015). A Cyber-Physical Systems Architecture for Industry 4.0-Based Manufacturing Systems. Manufacturing Letters, Vol. 3, pp. 18–23. https://doi.org/10.1016/j.mfglet.2014.12.001

[13]. Quigley, Morgan; Conley, Ken; Gerkey, Brian; Faust, Josh; Foote, Tully; Leibs, Jeremy; Wheeler, Rob; Ng, Andrew Y. (2009). ROS: An Open-Source Robot Operating System. ICRA Workshop on Open Source Software, pp. 1–6.

[14]. Macenski, Steve; Foote, Tully; Gerkey, Brian; Lalancette, Christian; Woodall, William. (2022). Robot Operating System 2: Design, Architecture, and Uses in the Wild. Science Robotics, Vol. 7, No. 66. https://doi.org/10.1126/scirobotics.abm6074

[15]. Kalayci, Cemal B.; Berger, Andreas. (2023). Autonomous Mobile Robots in Warehouse Logistics: A Systematic Literature Review. Logistics, Vol. 7, No. 2. https://doi.org/10.3390/logistics7020036

IJEAST

INTERNATIONAL JOURNAL OF ENGINEERING APPLIED SCIENCE AND TECHNOLOGY

AUTHORS

- Aditya Jadav
- Vedika Krishnankutty
- Rabinder Henry

2455-2143