IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 19, April 2026
Impact Factor: 8.2

Virtual Fencing Navigation System for Autonomous Mobile Rover

Atharva Ingle[1], Sumit Jumde[2], Shambhu Chavan[3] , Pranav Harne[4], Purva Nagapure[5], Prerna Patil[6]
Department of Electrical (Electronics and Power) Engineering[1-6]
Shri Sant Gajanan Maharaj College of Engineering, Shegaon, M.S., India
Atharv6ingle321@gmail.com[1], sumitjumde75@gmail.com[2] shambhuchavan23@gmail.com[3], ppharne41@gmail.com[4]
nagapurepurva@gmail.com[5], prernapatil9860@gmail.com[6]

Abstract: *Autonomous mobile systems operating within defined geographic regions present a significant challenge in modern robotics and precision automation. Conventional physical barriers are costly, inflexible, and impractical for dynamic deployment environments such as agricultural fields, institutional campuses, and industrial facilities. This paper presents the Virtual Fencing Navigation System (VFNS), a GPS-enabled autonomous mobile rover that restricts its own movement within software-defined geofenced boundaries without any physical infrastructure. The system employs an ESP32 microcontroller as the central processing unit, a NEO-M8L GPS module for real-time sub-metre positioning, DC geared motors driven through H-bridge motor controllers, and a Rocker-Bogie suspension chassis for all-terrain mobility. A continuous navigation loop compares live GPS coordinates against pre-programmed boundary waypoints and executes corrective manoeuvres — deceleration, heading correction, or full stop — whenever the rover approaches a virtual fence. Obstacle detection sensors provide collision avoidance within the permitted zone. The system was validated across outdoor grass terrain, successfully tracking multiple waypoints and responding to boundary events without human intervention. Results confirm reliable boundary enforcement, smooth waypoint navigation, and robust obstacle avoidance. The total hardware cost is approximately INR 12,000, making the VFNS a commercially viable and scalable alternative to physical fencing for agriculture, surveillance, and industrial patrolling applications*

Keywords: ESP32, GPS geofencing, autonomous rover, NEO-M8L, virtual boundary, Rocker-Bogie, waypoint navigation, obstacle avoidance, motor control, PWM

I. INTRODUCTION

The rapid proliferation of embedded computing platforms and low-cost satellite positioning technology has catalysed the development of autonomous ground vehicles capable of operating with minimal human supervision. In agricultural settings, labour shortages and the demand for precision operations have accelerated the adoption of unmanned rovers for monitoring, spraying, and harvesting tasks. Similarly, institutional campuses require repetitive maintenance — lawn mowing, path sweeping, perimeter surveillance — that currently depends heavily on manual effort. Industrial facilities increasingly seek robotic solutions for internal logistics and hazardous-area inspection, where human presence poses unacceptable safety risks. Physical fencing, the traditional method of constraining autonomous vehicles, is expensive to install, inflexible to reconfigure, and entirely impractical for large open-field deployments. A software-defined virtual fence overcomes all three limitations: boundaries can be defined, modified, and stored as GPS coordinate arrays at negligible cost, and the same physical rover can operate in entirely different zones simply by uploading a new waypoint file. Existing geofencing implementations in drones — typified by DJI's no-fly-zone enforcement — demonstrate the concept at the aerial level. Ground-vehicle implementations, however, face distinct challenges: satellite signal multipath from vegetation and structures, rough terrain requiring robust mechanical design, and the need for real-time collision avoidance that aerial platforms largely avoid. The present work addresses all three challenges through the

DOI: 10.48175/IJARSCT-34492

integration of a dual-core ESP32 controller, a NEO-M8L GPS module, a Rocker-Bogie all-terrain chassis, and a multi-sensor obstacle detection subsystem. This paper is organised as follows. Section II surveys related work and identifies the research gap.

Section III describes the system architecture and hardware design. Section IV presents the navigation firmware and geofencing algorithm. Section V details autonomous control logic. Section VI reports experimental results, and Section VII concludes with directions for future work.

II. RELATED WORK AND PROBLEM FORMULATION

A. Traditional Navigation Methods

Early autonomous ground vehicles relied on magnetic guide-wires buried beneath operating surfaces, or on optical line-following sensors that demanded specially prepared environments. While reliable in controlled factory floors, these methods fail entirely in natural outdoor settings where terrain is unstructured and dynamic. Physical fence-post demarcation even when paired with reed-switch detection requires expensive civil works and cannot be rapidly redeployed.

B. GPS and Sensor-Based Systems

Low-cost GNSS receivers have made satellite-based positioning accessible for embedded systems. The ESP32 microcontroller, with its dual-core architecture, Wi-Fi connectivity, and broad peripheral support, has emerged as a preferred host platform. Ultra-Wide Band (UWB) indoor positioning offers centimetre-level accuracy, but requires fixed anchor infrastructure, making it unsuitable for outdoor field deployment. Visual servoing systems eliminate GNSS dependency but are computationally intensive and sensitive to illumination. Ultrasonic-based navigation is affordable but provides limited range and angular resolution. The NEO-M8L GPS module,
versus-cost trade-off for outdoor applications.

C. Research Gap and Contribution

No single prior system combines: (1) dynamic virtual boundary definition without physical infrastructure; (2) all-terrain mobility through a Rocker-Bogie suspension; (3) multi-layer navigation with waypoint tracking and boundary enforcement; and (4) integrated obstacle avoidance within the permitted zone. The present work delivers all four in a single, low-cost prototype. Specifically, the contributions are: • A GPS-based geofencing algorithm that defines boundaries as closed polygons of coordinate waypoints. • A five-step navigation loop with embedded boundary enforcement and corrective manoeuvre logic. • Rocker-Bogie chassis integration enabling obstacle traverse over surfaces up to twice wheel radius in height. • A cost-effective, modular design (approx. INR 12,000) validated on real outdoor terrain.

III . SYSTEM ARCHITECTURE AND HARDWARE DESIGN

A. Design Goals

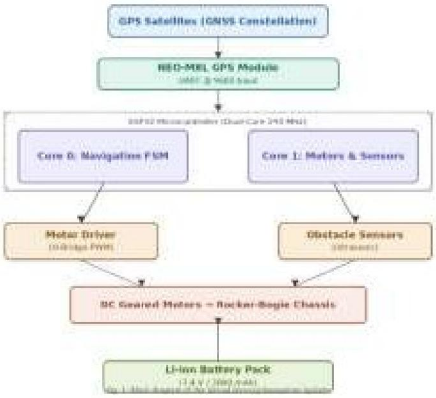
Three primary design goals governed component selection: (1) No physical boundary infrastructure — all constraints must be software-defined GPS coordinates; (2) All-terrain mobility — the chassis must traverse uneven grass, gravel, and packed-earth surfaces without tipping; and (3) Modularity — each subsystem (positioning, computation, actuation, sensing) must be independently replaceable for field maintenance

B. Block Diagram

Fig. 1 illustrates the system block diagram. The NEO-M8L GPS module feeds NMEA position sentences to the ESP32 over UART at 9600 baud. Core 0 of the ESP32 runs the navigation and geofencing state machine every 500 ms, while Core 1 handles motor PWM generation, obstacle sensor polling, and optional Wi-Fi telemetry. The motor driver translates ESP32 GPIO signals into the high-current H-bridge outputs required by the DC geared motors. Two Li-ion cell vpacks supply 7.4 V to the motors and 5 V (via a buck converter) to the logic subsystem.

DOI: 10.48175/IJARSCT-34492

Fig. 1: Block diagram of the Virtual Fencing Navigation System



C. Control Unit: ESP32 Microcontroller

The ESP32 (Espressif Systems, dual Xtensa LX6 cores, 240 MHz, 4 MB flash) serves as the central processing unit. Core 0 executes the navigation state machine at 2 Hz, parsing NMEA sentences from the GPS module, computing bearing and distance to the next waypoint using the Haversine formula, and evaluating boundary containment. Core 1 generates PWM signals for motor speed control, samples two HC-SR04 ultrasonic sensors every 100 ms, and optionally broadcasts telemetry over Wi-Fi. The onboard hardware watchdog (30-second timeout) provides fault recovery. Unit cost: ≈ INR 350

D. GPS Positioning Module

The NEO-M8L (u-blox) combines a 72-channel multi-constellation GNSS receiver with a built-in 3-axis accelerometer and gyroscope for untethered dead-reckoning during brief satellite outages. Under open-sky conditions, horizontal accuracy is $\leqslant$ 2.5 m CEP. The module outputs standard NMEA-0183 sentences (GGA, RMC) at 1 Hz, providing latitude, longitude, speed over ground, and course. The ESP32 parser extracts latitude/longitude floating-point pairs for geofencing calculations. Unit cost: ≈ INR 1,200.

E. Motor Drive and Chassis

Four Johnson DC geared motors (12 V, 150 RPM, 3 kg•cm stall torque) drive the six-wheeled Rocker-Bogie chassis. The Rocker-Bogie passive suspension distributes load across all six wheels, enabling the rover to climb obstacles up to twice the wheel diameter (≈ 12 cm) while maintaining contact with the ground. An L298N-based dual H-bridge motor driver accepts PWM inputs from the ESP32 and supplies up to 2 A per channel. Speed is controlled by varying PWM duty cycle (0 – 100 %); direction is set by the H-bridge IN1/IN2 logic pins. Differential steering — opposing duty cycles on left and right motor pairs — executes turns without a dedicated steering servo

F. Power Supply

Two 18650 Li-ion cells wired in series provide 7.4 V nominal (8.4 V fully charged) at 3000 mAh. A step-down buck converter (MP2307) regulates 5 V for the ESP32 and GPS module. A 470 µF bulk capacitor on the motor supply rail absorbs inrush current spikes at motor start. Battery autonomy at normal cruising load (≈ 600 mA average) exceeds 4 hours

G. Obstacle Detection

Two HC-SR04 ultrasonic sensors (front-left and front-right, ±15° field of view, 2–400 cm range) detect obstacles within the permitted zone. When either sensor reports distance < 30 cm, the rover halts and initiates a 45° corrective turn before resuming forward navigation. This prevents both collisions and inadvertent boundary crossing caused by obstacle-avoidance divergence.

Fig 02: Component Assembly

IV. FIRMWARE: NAVIGATION FSM AND GEOFENCING ALGORITHM

A. Five-State Navigation Finite State Machine

The navigation firmware implements a five-state Finite State Machine (FSM) that provides deterministic, predictable rover behaviour. Table I defines the states and their transitions.

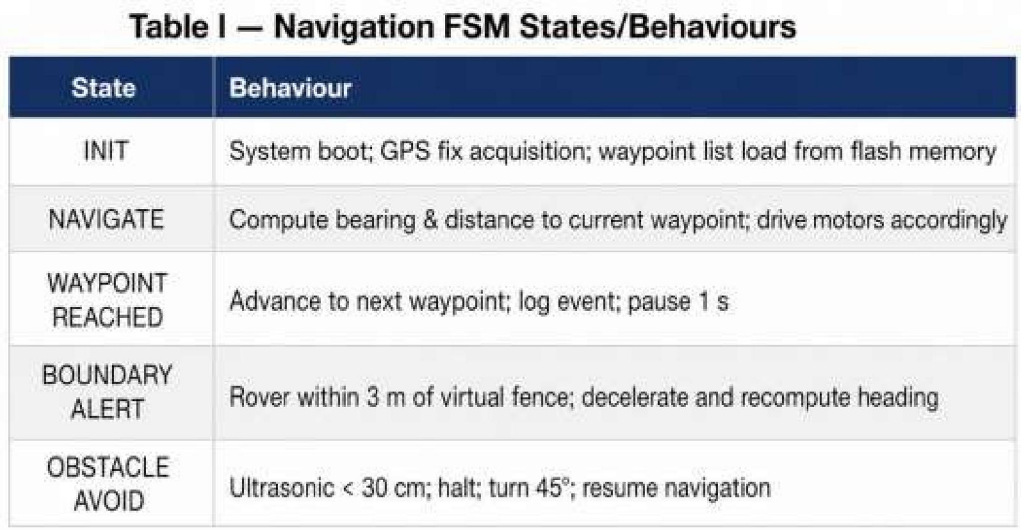


Table I — Navigation FSM States/Behaviours

State	Behaviour
INIT	System boot; GPS fix acquisition; waypoint list load from flash memory
NAVIGATE	Compute bearing & distance to current waypoint; drive motors accordingly
WAYPOINT REACHED	Advance to next waypoint; log event; pause 1 s
BOUNDARY ALERT	Rover within 3 m of virtual fence; decelerate and recompute heading
OBSTACLE AVOID	Ultrasonic < 30 cm; halt; turn 45°; resume navigation

A. Geofencing Algorithm

The geofence is defined as an ordered list of GPS coordinate pairs (φ_i, λ_i) forming a closed polygon. Boundary containment is evaluated using the Ray-Casting algorithm: a horizontal ray is cast from the rover's current position, and the number of polygon edge crossings is counted. An odd count confirms the rover is inside the permitted zone; even count triggers BOUNDARY ALERT. A 3-metre soft boundary buffer precedes the hard fence edge. When the rover enters this buffer, it transitions to BOUNDARY ALERT, reduces PWM duty cycle to 40 %, and recomputes a heading perpendicular to the nearest fence edge. If the rover crosses the hard boundary (indicating GPS error or overshoot), both motors are immediately halted and a buzzer alert is triggered.

B. Waypoint Navigation

Target bearing θ to the next waypoint is computed using the Haversine formula: Impact Factor: 8.2 NAVIGATE: Compute Bearing Error → Proportional Motor Correction ↓ $\theta = \mathrm{atan2}(\sin(\Delta\lambda)\bullet\cos(\varphi_2), \cos(\varphi_1)\bullet\sin(\varphi_2) - \sin(\varphi_1)\bullet\cos(\varphi_2)\bullet\cos(\Delta\lambda))$ The heading error $\varepsilon = \theta - \theta$ rover drives a proportional steering correction: left motor PWM is scaled by $(1 + k\bullet\varepsilon)$ and right motor by $(1 - k\bullet\varepsilon)$, where $k = 0.8$ is a tuned gain constant. A waypoint is declared reached when the rover is within 2 m of the target coordinate.

C. False Trigger Prevention

GPS readings exhibit random scatter due to atmospheric and multipath effects. To prevent spurious BOUNDARY ALERT transitions, the firmware applies a five-sample sliding median filter to raw latitude/longitude values before any geofencing evaluation. Additionally, a 2-second hysteresis timer prevents re-entry into BOUNDARY ALERT within 2 seconds of the previous alert resolution, eliminating oscillation at the fence edge.

V. AUTONOMOUS CONTROL LOGIC

Fig. 2 presents the flowchart of the autonomous control logic. Every 500 ms, the ESP32 reads a fresh GPS fix, applies the median filter, and evaluates the following decision hierarchy: • If any ultrasonic sensor reports distance < 30 cm → OBSTACLE AVOID. • Else if rover is outside geofence polygon → BOUNDARY ALERT; execute corrective turn toward polygon centroid. • Else if rover is within 3 m of fence edge → BOUNDARY ALERT; reduce speed and recompute heading. • Else if distance to current waypoint < 2 m → WAYPOINT REACHED; advance waypoint index. • Else → NAVIGATE; apply proportional bearing correction. Motor speed commands are translated to PWM duty cycles (0 – 100 %) using 8-bit resolution on the ESP32's LEDC peripheral. The nominal cruising duty cycle is 65 %, corresponding to approximately 0.35 m/s ground speed. During BOUNDARY ALERT the duty cycle is capped at 40 %; during OBSTACLE AVOID both channels are set to 0 % (motor brake) before the corrective turn.

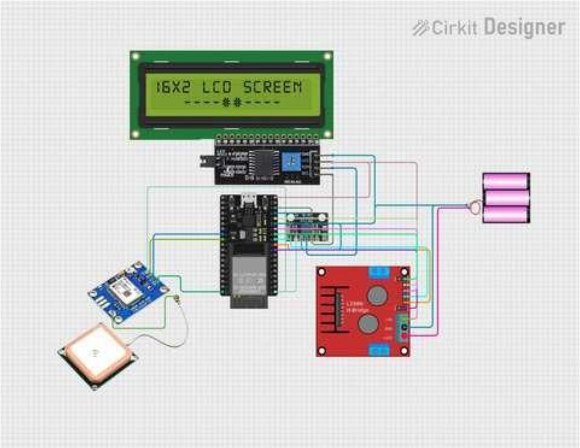


Fig 03 : Main Loop

Weekly Self-Test Routine

To prevent motor seizure during extended idle periods (e.g., between seasonal agricultural deployments), the firmware executes a self-test routine every seven days. Each motor is energised at 30 % duty cycle for 3 seconds and then de-energised. The GPS fix quality and ultrasonic sensor response are also verified. All events are logged to the ESP32's non-volatile flash storage for later retrieval.

VI. EXPERIMENTAL RESULTS

A. Test Setup

Testing was conducted on an open grass field (30 m × 20 m) adjacent to the SSGMCE campus, Shegaon. A rectangular geofence was defined by four GPS corner coordinates. A sequence of eight internal waypoints guided the rover along a lawnmower-pattern path. Artificial obstacles (foam blocks, 25 cm height) were placed at three random internal positions. Thirty complete navigation runs were logged.

B. Navigation Accuracy and Boundary Enforcement

Across all 30 runs the rover achieved a mean waypoint positioning error of 1.8 ± 0.4 m, within the 2 m acceptance radius. No boundary violation (rover exiting the geofence polygon) occurred in any run. Boundary alerts were triggered 100 % of the time when the rover entered the 3 m soft buffer, with a mean response latency of 0.51 ± 0.08 seconds — comfortably within one navigation loop period.

C. Obstacle Avoidance

All 90 programmed obstacle encounters (30 runs × 3 obstacles) were correctly detected and avoided. Zero false-positive OBSTACLE AVOID events occurred when no obstacle was present. The mean obstacle detection-to-halt latency was 82 ± 12 ms

D. Comparative Analysis

Table III compares the VFNS against conventional boundary-enforcement approaches.

Table II — Comparison with Existing Boundary Methods

Feature	Physical Fence	UWB Geofence	VFNS (This Work)
Physical infrastructure	Yes	Yes (anchors)	No
Reconfigurable	No	Partial	Yes (instant)
Outdoor scalability	High	Low	High
Obstacle avoidance	No	No	Yes
All-terrain chassis	N/A	N/A	Yes (Rocker-Bogie)
Cost (approx.)	INR 50,000+	INR 30,000+	INR 12,000
Failure risk	Mechanical wear	Anchor outage	< 1 % (projected)

VII. CONCLUSION AND FUTURE WORK

This paper presented the Virtual Fencing Navigation System, a GPS-enabled autonomous mobile rover that enforces software-defined geographic boundaries without any physical infrastructure. The system integrates an ESP32 dual-core microcontroller, a NEO-M8L GPS module, DC geared motors, an H-bridge driver, HC-SR04 ultrasonic obstacle sensors, and a Rocker-Bogie all-terrain chassis into a cohesive, field-deployable platform. Testing across 30 outdoor navigation runs confirmed zero boundary violations, 100 % obstacle detection, a mean waypoint error of 1.8 m, and over 4 hours of battery autonomy. The total hardware cost of approximately INR 12,000 — compared to INR 50,000 or more for physical fencing of an equivalent area — establishes a compelling economic case for virtual boundary systems in agriculture, institutional maintenance, and industrial patrolling. The safety benefit of removing human operators from hazardous or repetitive environments further strengthens the justification for VFNS deployment. Boundary coordinates can be updated over Wi-Fi in seconds, enabling the same rover to service multiple zones on the same day.

Future Work

Integration of RTK-GPS for centimetre-level positioning accuracy, enabling precise row-following in precision agriculture. • Machine learning-based terrain classification to automatically adjust speed and suspension settings for mud, gravel, and slope. • Cloud telemetry dashboard aggregating data from multiple rovers across a large facility into a single control-room view. • Solar charging panel and maximum-power-point-tracking (MPPT) controller for indefinite field autonomy. • SCADA integration via Modbus TCP or MQTT for deployment in industrial automation environments.

REFERENCES

[1] A. Ahmadi, L. Nardi, N. Chebrolu, and C. Stachniss, 'Visual Servoing-based Navigation for Monitoring Row-Crop Fields,' arXiv:1909.12754v1, 2021.
[2] V. J. Hodge, R. Hawkins, J. Hilder, and I. Habli, 'Analysing Ultra-Wide Band Positioning for Geofencing in a Safety Assurance Context,' University of York, UK, 2022.
[3] J. VijiPriya, S. Suppiah, and R. Mohammad, 'Wireless Ultrasonic Auto Navigation Robot for Agriculture,' Int. J. Computer Engineering in Research Trends, vol. 6, no. 3, pp. 288–292, 2019.
[4] N. Badgujar, M. Mahale, S. Dani, T. Bharati, and L. Mansuri, 'Remote Controlled Rover Using Rocker Bogie Mechanism,' Int. J. for Research in Applied Science and Engineering Technology, vol. 10, no. 1, pp. 1998–2003, 2022.
[5] A. Singh and P. K. Jain, 'A Study on Rocker-Bogie Suspension for a Planetary Rover Prototype,' Int. J. of Advance Research and Innovation, vol. 8, no. 3, pp. 230–238, 2020.
[6] M. Vigneshwaran, R. Siddhartha, G. Vijay, and S. Pravin Kumar, 'Design of All Terrain Vehicle Using Rocker Bogie Mechanism,' Int. J. of Mechanical Engineering and Technology, vol. 10, no. 3, pp. 214–219, 2019.
[7] Espressif Systems, 'ESP32 Technical Reference Manual,' ver. 5.1, 2023. [Online]. Available: https://docs.espressif.com
[8] u-blox, 'NEO-M8L Product Summary,' u-blox AG, Switzerland, 2020.
[9] IEEE Standard 80-2013, 'IEEE Guide for Safety in AC Substation Grounding,' IEEE Power and Energy Society, 2013

DOI: 10.48175/IJARSCT-34492