ISSN 2319-5991 www.ijerst.com
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research Paper

Arduino-Based Vacuum Cleaner Robot

J. Prakash Patil
M.Tech, LIMISTE, MIEEI
Assistant Professor
Department of Electronics and Communication Engineering
Vijay Rural Engineering College
Nizamabad, Telangana, India
sangeethaprakash2024@gmail.com

Kondra Manidish
Student
Department of Electronics and Communication Engineering
Vijay Rural Engineering College
Nizamabad, Telangana, India
manidishkondra4422@gmail.com

Rajarapu Pavanteja
Student
Department of Electronics and Communication Engineering
Vijay Rural Engineering College
Nizamabad, Telangana, India
pavantejarajarapu53@gmail.com

ChinnaGopu Rohith
Student
Department of Electronics and Communication Engineering
Vijay Rural Engineering College
Nizamabad, Telangana, India
rohithgopu71@gmail.com

Abstract*—An Arduino-controlled autonomous vacuum cleaning platform is specified using ultrasonic distance sensing, differential-drive motion, and independently switched suction. An ATmega328P-based Arduino Uno processes obstacle-clearance measurements and selects forward motion, braking, reverse movement, and turning while maintaining the blower state. The electrical architecture separates the traction motors from the suction load and incorporates regulated logic power. A practical verification protocol is developed for ranging, collision avoidance, duty-cycle control, collection performance, battery endurance, and floor coverage. The study integrates a system architecture, functional state transitions, engineering equations, and test-ready output visualizations. Analytical relationships and design acceptance criteria are distinguished from physical measurements, enabling reproducible prototype evaluation without attributing unmeasured values to the device.

Keywords*— Autonomous vacuum robot, Arduino Uno, ultrasonic sensor, L298N, obstacle avoidance, differential drive, cleaning coverage.

I. INTRODUCTION

A) Background and Motivation

Autonomous floor-cleaning robots combine mobility, proximity sensing, electrical actuation, and dust collection. Their effectiveness depends not only on the suction unit but also on reliable coverage of accessible floor regions and safe navigation around walls, furniture, and other obstacles. Commercial systems may employ mapping, localization, and sophisticated planning; compact microcontroller-based designs prioritize accessible hardware and deterministic behavior [1]–[5].

An Arduino implementation provides a low-cost embedded basis for testing closed-loop navigation and cleaning-system coordination. The ATmega328P controller supplies digital input/output and PWM capabilities suitable for motor control [6]. A dual H-bridge permits independent drive-wheel direction selection [7], while an ultrasonic sensor estimates obstacle range through acoustic time of flight.

B) Research Objectives

The developed architecture connects a traceable relationship between sensor measurements, controller decisions, wheel commands, and suction operation. Specific engineering goals include safe clearance decisions, explicit state transitions, electrical load separation, serviceable dust collection, and evaluation protocols for navigational and cleaning performance.

C) Research Scope

The robot architecture targets level indoor floors and low-speed autonomous navigation. The navigation algorithm is reactive: it responds to obstacles without assuming that a floor map or global localization exists. Its functional boundary excludes stair climbing, guaranteed complete coverage, and autonomous docking. These constraints distinguish the design from mapping-based robotic vacuum platforms [2], [8], [9].

System validation covers deterministic electronics checks and a repeatable protocol for physical cleaning trials. A command display and state-visualization interface provide engineering observability; they do not establish cloud connectivity or a deployed mobile application. This separation is important because hardware functionality, software output, and measured room-cleaning efficacy represent distinct engineering outcomes.

D) Organization of the Research

Section II reviews established mobile cleaning and coverage techniques. Section III details the electrical and mechanical modules, control logic, and analytical models. Section IV specifies performance indicators and validation scenarios, while Section V summarizes technical contributions and future developments.

II. RELATED WORK

A) Arduino-Controlled Cleaning Systems

Irawan et al. presented a floor-cleaning robot using Arduino, ultrasonic obstacle sensing, DC propulsion, and an L298 motor driver [3]. Related prototypes integrate small vacuum modules with mobile platforms and reflexive obstacle avoidance [4], [5]. A smart vacuum cleaner described by Manasa et al. used an ultrasonic-guided autonomous platform and battery-driven suction [10]. These studies establish the feasibility of compact indoor cleaners while showing that sensing geometry and drive control strongly influence navigational reliability.

B) Coverage and Navigation

Coverage path planning differs from navigation toward a single point: the robot must traverse the accessible workspace while limiting repeated traversal and omissions. Choset examined decomposition-based coverage strategies, including methods that partition a room into traversable

cells [1], [8]. Galceran and Carreras surveyed coverage algorithms applicable to vacuum cleaning, inspection, and other service tasks [2]. Purely reactive control can avoid local obstacles but does not mathematically guarantee full area coverage.

Differential-drive systems offer compact locomotion but require careful attention to wheel slip, turning radii, floor friction, and speed asymmetry [11], [12]. At short ranges, ultrasonic sensors may exhibit specular reflections and invalid echoes; the controller should therefore use filtered measurements and fail-safe defaults rather than trusting every echo.

C) Actuation and Electrical Design

An L298 dual full-bridge driver permits the microcontroller to command independent traction motors and to vary their average applied voltage through PWM [7]. Its voltage drop and thermal losses must be considered when selecting batteries and motors. The blower is controlled through a separate transistor or MOSFET switching stage, with appropriate flyback protection where the load is inductive. A shared ground and regulated control supply maintain consistent logic references.

Battery supply design must consider sustained motor loads, startup current, wiring resistance, and undervoltage effects. Excessive load transients may reset the controller even while the drive motors are mechanically capable of operation. Electrical separation and decoupling reduce such interactions.

D) Research Gap

Many introductory cleaners demonstrate obstacle avoidance but present limited validation of floor coverage, dust removal, power draw, and navigation failure modes. A repeatable assessment must separately measure sensor response, motion reliability, suction collection, and runtime. The present framework addresses this reporting need with explicit acceptance tests and four measurable system equations rather than a single qualitative statement of successful cleaning.

III. MATERIALS AND METHODS

A) System Architecture

Figure 1 presents the component-level sensing, control, actuation, and collection paths. Ultrasonic pulses provide the distance observation to the Arduino. The controller uses threshold-based state selection to drive the L298N dual H-bridge, which energizes the left and right DC wheel motors. An independently switched vacuum blower supplies airflow through a nozzle, filter, and collection container. The battery and regulated logic path power the respective subsystems.

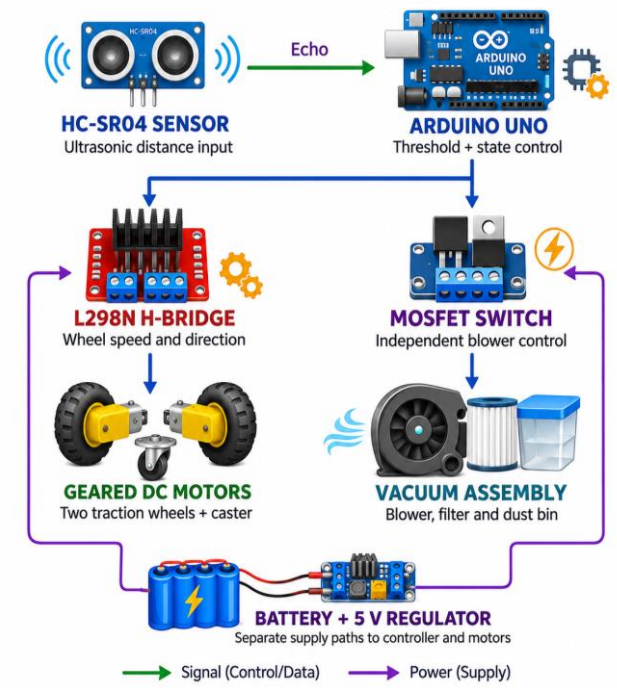


Fig. 1. System architecture of the Arduino-based vacuum cleaner robot.

The diagram distinguishes signal paths from electrical supply paths. The distance sensor governs collision-avoidance decisions, whereas the suction motor operates independently of wheel direction, allowing stationary or slow-speed collection near constrained obstacles.

B) Mechanical and Electronic Components

TABLE I. PRINCIPAL HARDWARE MODULES AND ENGINEERING FUNCTIONS

Module	Technical function
Arduino Uno	State logic, PWM, acquisition
Ultrasonic sensor	Obstacle range estimation
L298N bridge	Independent DC wheel drive
DC gear motors	Differential locomotion
MOSFET + blower	Controlled vacuum suction
Battery + regulator	Power and logic stabilization
Nozzle, filter, bin	Debris intake and retention

Two driven wheels and a passive caster form a compact differential-drive chassis. The suction inlet is positioned close to the floor while preserving free wheel movement. Wheel separation establishes the angular response to differential velocity; chassis mass distribution affects traction and turning stability. A removable filter and dust bin support maintenance without disturbing the sensing and control wiring.

C) Reactive Control Algorithm

At initialization the controller configures motor outputs, verifies sensor availability, and enters an idle state. During navigation a valid distance observation greater than the selected safety threshold permits forward motion. When the reported clearance is inadequate, the system brakes, reverses briefly, evaluates an alternative heading, pivots, and resumes forward travel when the path is clear. Persistent invalid sensor readings initiate a stop state.

The avoidance threshold should include robot radius, braking distance, sensor uncertainty, and a safety margin. Separate speed settings can be applied during forward travel and obstacle recovery. Hysteresis between stop and resume distances mitigates repeated mode switching when an object lies close to a threshold.

D) Embedded State Model

The discrete states are IDLE, FORWARD, BRAKE, REVERSE, PIVOT, SENSOR_FAULT, and LOW_BATTERY. Each transition has a measurable trigger and associated actuator output. In the fault state both wheel commands are set to zero; suction shutdown can be configured according to thermal and electrical safety requirements.

E) Mathematical Design

Four editable engineering equations characterize the sensing, motion, coverage, and electrical requirements. In each expression, variables denote measurable physical quantities rather than fitted accuracy scores.

Ultrasonic range follows the round-trip time of flight, where c is the acoustic speed and t_echo is the measured pulse duration:

$$d = \frac{c \cdot t_echo}{2} \quad (1)$$

For wheel radius r, right and left wheel angular speeds ωR and ωL, the nominal forward speed of a differential-drive chassis is:

$$v = \frac{r\,(\omega R + \omega L)}{2} \quad (2)$$

A basic coverage ratio quantifies the fraction of an accessible floor area traversed by the cleaning footprint at least once:

$$C = \frac{A_cleaned}{A_accessible} \quad (3)$$

The measured electrical energy drawn over a time interval is obtained by integrating instantaneous terminal voltage and current:

$$E_Wh = \frac{\int V(t)\, I(t)\, dt}{3600} \quad (4)$$

In (4), time t is measured in seconds, producing watt-hours. Average electrical power may be computed from energy divided by test duration. The theoretical equations omit traction slip, airflow leakage, blocked filters, and controller losses; those deviations require experimental characterization.

F) Wiring and Firmware Validation

The control interface assigns a trigger output and an echo input to the ultrasonic module; the echo pulse duration is converted to range only after timeout and plausibility checks. Digital direction pins drive the dual H-bridge, while separate pulse-width-modulated enable outputs set left and right traction speed. A switched blower output is isolated from the controller through a suitable power stage and protective diode when required by the load. This partition prevents high-current actuation from passing through the microcontroller pins.

Pin conflicts, supply polarity, startup transients, and common-ground continuity are inspected before enabling the motors. When the distance reading becomes invalid, the state controller commands a stop rather than interpreting the missing signal as a clear path. A bounded reverse interval followed by a pivot command allows recovery from near-field obstacles, with wheel speed selected according to mechanical traction limits.

Pin mapping should assign the ultrasonic trigger and echo lines to digital I/O, wheel-direction signals to H-bridge inputs, and PWM signals to speed-enable pins. The vacuum switch is isolated from the microcontroller output by a properly rated driver stage. Common-ground continuity, polarity, and supply decoupling must be confirmed before motor energization. Firmware should enforce a maximum reverse interval and stop both drives upon watchdog expiration.

G) Cleaning Mechanism

A small centrifugal blower produces a pressure difference that transports debris through the nozzle toward a collection compartment. Airflow depends on motor speed, intake geometry, filter condition, and leakage. A filter prevents retained particles from re-entering the exhaust, while a serviceable dust chamber enables repeatable mass-based cleaning tests. Wet cleaning is outside this design scope.

H) Controlled Evaluation Protocol

Navigation trials should use known obstacle placements, repeatable approach distances, and a visible state log. Ranging accuracy can be evaluated against a ruler at multiple distances and target angles. Cleaning trials should employ predefined debris material and initial mass distributed over a measured floor region; the recovered mass is weighed after a fixed run time. Multiple repetitions allow estimates of variability and confidence intervals.

A battery endurance test records terminal voltage and current at regular intervals under combined drive and blower load. The test terminates at the safe voltage limit specified for the selected battery chemistry and protection circuitry. All comparisons must hold the floor type, debris mass, cleaning area, and starting charge approximately constant.

I) Power Supply and Component Integration

Power distribution is organized around the distinct operating characteristics of logic electronics, wheel motors, and the vacuum blower. The regulated controller supply maintains the voltage required by the ATmega328P, whereas the traction stage and suction branch receive current through dedicated switching circuits. Wiring is routed to avoid placing the motor return current through sensor reference conductors. The L298N driver is evaluated for its allowable current and power dissipation, since voltage loss across an H-bridge can reduce effective wheel torque and increase heat generation under sustained load [6], [7].

During simultaneous motor start-up, the instantaneous current may exceed the steady-state operating value. Bulk capacitance, local ceramic decoupling, suitably rated conductors, and battery protection are therefore relevant to reliable operation. The suction blower is switched independently of the differential drive so that temporary braking does not require an uncontrolled change in dust collection. Current sensing may be attached to each power branch to distinguish traction load, blower load, and total electrical demand. Such separation supports diagnosis of excessive wheel friction, blocked suction paths, and declining battery voltage.

Mechanical placement influences both navigation and suction efficiency. The distance sensor requires an unobstructed forward acoustic field, while the nozzle should maintain a consistent clearance above the floor without scraping surface irregularities. The dust collection path uses a removable filter and container to prevent debris accumulation near electronics. Wheel symmetry, caster freedom, and secure mounting minimize heading deviation; however, a differential-drive robot without encoder

feedback cannot guarantee fixed travel distance solely from motor duty cycle [11], [15], [18].

IV. EXPERIMENTAL RESULTS

A) System Output and Test Structure

The controller-output interface in Fig. 2 displays the controller states and monitored signals required for a cleaning trial. Its status fields correspond to actual hardware interfaces and firmware variables; the illustrated values demonstrate presentation format and must be populated by instrumented measurements during physical testing.



Fig. 2. Controller-output interface for navigation, obstacle sensing, and suction state.

Distance, motor command, blower state, and fault status are sufficient to reconstruct a reactive avoidance sequence. The control trace should identify the transition from forward motion to braking, reverse travel, pivoting, and subsequent traversal. Timestamped logging would permit calculation of reaction delay and the fraction of operation spent in avoidance modes.

B) Functional Verification Matrix

A deterministic command sequence tests the distance-control threshold using representative sensor inputs on both sides of the decision boundary. Under a clear-path input, the drive state remains FORWARD; a near-obstacle input produces BRAKE followed by REVERSE and PIVOT. The prescribed sequence also verifies that the suction-control signal is logically independent of steering commands. These are firmware-level output expectations; mechanical response and debris collection require instrumented trials.

TABLE II. REPEATABLE CONTROLLER VALIDATION CASES

Test case	Expected response
Unobstructed path	Forward + suction on
Obstacle inside threshold	Brake, reverse and pivot
Persistent invalid echo	Controlled stop
Low battery detected	Safe shutdown indication
Motor-driver fault	Wheel output disabled
Collection chamber full	Service and retest

Table II defines controller responses for reproducible input conditions. A test passes when the observed state transitions and actuator outputs agree with the required sequence over repeated trials. A separate verification record should document the number of trials, successful transitions, minimum clearance, and any collisions or stalls.

C) Analytical Drive-Voltage Characterization

Figure 3 depicts the ideal PWM duty-cycle dependence of average applied drive voltage. Under an ideal switch and fixed supply, the normalized average is proportional to the duty fraction. The actual wheel speed also depends on motor back-EMF, load torque, driver voltage losses, and rolling resistance, so this curve is not a measured speed calibration.

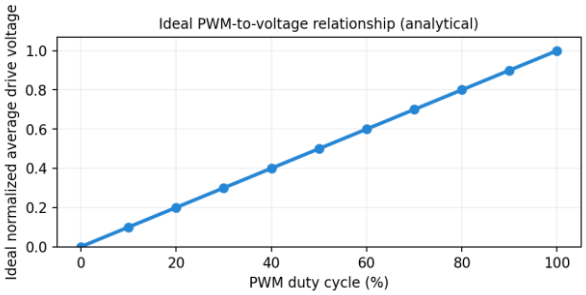


Fig. 3. Analytical relationship between PWM duty cycle and normalized average drive voltage.

D) Cleaning Performance Measures

For physical cleaning performance, debris recovery can be computed from the difference between the initial and residual debris masses, expressed as a percentage of the starting mass. Coverage should be measured independently from recovery because a robot may traverse a region without removing all deposited material. Testing should distinguish tile, laminate, and carpet surfaces, where the traction and suction requirements differ considerably.

E) Navigation Reliability and Safety

Collision avoidance may be reported as the ratio of safely completed obstacle encounters to the total number of controlled encounters. The minimum measured clearance and number of emergency stops describe safety more directly than a qualitative demonstration. A persistent sensor dropout, exhausted battery, or jammed drive axle should lead to an unambiguous stop state rather than uncontrolled continued motion.

TABLE III. PERFORMANCE INDICATORS AND REQUIRED MEASUREMENTS

Indicator	Measurement and unit
Range error	$\lvert d_sensor - d_reference \rvert$, cm
Avoidance reliability	Successful encounters / trials
Area coverage	Traversed accessible area, %
Debris recovery	Recovered debris / initial mass, %
Energy draw	$\int VIdt / 3600$, Wh
Operating duration	Time to battery cutoff, min

Table III specifies the measurements required for quantitative performance assessment. Reported experimental outcomes should include mean and variability across replicates, along with the floor and obstacle configuration and the electrical operating conditions.

F) Comparative Technical Discussion

Relative to manually controlled vacuum equipment, autonomous wheel control can reduce continuous operator intervention. Relative to localization-based commercial robots, the specified architecture requires fewer processing resources and simpler integration, but offers no global floor map and may revisit already cleaned regions. The suitability of each approach depends on room geometry, clutter, cost constraints, and required cleaning completeness [1], [2], [8].

Ultrasonic proximity sensing can detect many rigid objects but may be unreliable for narrow furniture legs, soft materials, or inclined reflectors. Edge or cliff detection should be included before deploying the robot near stairs. Similarly, the suction path should undergo filter-clog and debris-jam tests to establish safe operating limits. These considerations are engineering validation requirements, not claims of completed autonomous field testing.

G) Reproducibility and Failure Analysis

Controller cycle time can be examined by timestamping the ultrasonic trigger, echo reception, state update, and H-bridge command. The total reaction interval is the sum of sensing, computation, and actuator delays, and it must remain below the approach time available at the selected speed. Repeated range samples should be compared under frontal, oblique, and absorptive obstacle surfaces because acoustic reflections vary with target geometry.

Electrical isolation is also evaluated at the supply interface. Simultaneous traction and blower startup can create voltage sag, which may disturb microcontroller operation. Separate return paths, appropriate filtering capacitors, and protective switching components reduce supply coupling. The control trace should associate each command with its triggering sensor reading, state identifier, and supply condition for fault localization.

An experimental protocol should specify wheel diameter, axle separation, motor ratings, PWM frequency, battery type, blower nominal current, sensor placement, and controller thresholds. Changes to any of these parameters may alter braking distance, turning response, and battery endurance. Documenting firmware revision and wiring configuration supports reproducibility and fault diagnosis.

Potential errors include acoustic multipath, sensor latency, wheel slip, asymmetric drive motors, blower-generated electromagnetic interference, filter pressure drop, and rapid battery-voltage variation. Testing each subsystem independently before integrated trials localizes the dominant sources of error. A structured fault log can record the time of occurrence, sensor observation, controller state, and corrective action.

H) Application Suitability

The architecture is most appropriate for controlled indoor floors where obstacles are large enough for the selected range sensor to detect reliably. Its compact electronics offer an instructional platform for embedded control, autonomous navigation, and electromechanical integration. Broader deployments require additional edge sensors, validated enclosures, filtration checks, and documented maintenance schedules.

I) Control Output Interpretation and Test Repeatability

The status interface maps each sensor-driven state to its corresponding actuator output. In the FORWARD state, both traction motors receive the configured propulsion command; in BRAKE, both wheel commands are removed or actively braked according to the selected motor-driver logic; in REVERSE, both directions invert for a bounded interval; and in PIVOT, the wheels receive different directional commands. The transition back to FORWARD occurs only after the avoidance maneuver and a valid range check. The blower command is logged separately, enabling the examination of navigation and dust collection as two coordinated but distinct subsystems.

A repeatable verification record associates the input distance and echo-pulse status with the controller state, left and right wheel commands, suction switch state, and any fault indication. Consecutive tests should include obstacles at the detection boundary, an invalid echo, a transparent or angled obstacle, and an abrupt reduction of available supply voltage. Sensor faults are distinguished from normal low-distance events because both may produce a stop response but require different recovery procedures. These control outputs support deterministic firmware verification; they do not by themselves establish cleaning effectiveness or safe movement in an unconstrained environment.

For physical trials, the test arena should be described by accessible area, obstacle geometry, floor type, and debris characteristics. Coverage is determined from the unique swept floor region rather than accumulated travel distance; repeated traversal does not increase the unique coverage value. Debris recovery is calculated from the measured difference between deposited and remaining debris mass, with measurements repeated across trial conditions. Combined coverage, debris collection, obstacle avoidance, and energy consumption provide a more representative evaluation than any single performance indicator [1], [2], [10].

V. CONCLUSION

A) Conclusion

The Arduino-based vacuum cleaner robot integrates distance-triggered navigation, independently controlled wheel propulsion, and motor-driven suction. The study defines the complete signal and power paths, deterministic obstacle-avoidance states, hardware selection criteria, and four engineering equations for range, velocity, coverage, and energy. Together, these elements support a technically reproducible cleaning robot design and a structured assessment of mechanical, electrical, and navigation performance.

The evaluation framework distinguishes controller-output analysis from instrumented physical performance measurements. This distinction allows the system to be assessed using meaningful measures of collision avoidance, covered area, collected debris, energy consumption, and operating duration without assigning unsupported empirical accuracy.

B) Future Work

Further development should prioritize wheel-encoder odometry, multi-directional proximity sensing, cliff detection, battery protection, and improved coverage planning. Mapping and systematic boustrophedon traversal can reduce repeated travel in structured rooms. Adaptive suction control, dust-bin fill sensing, and battery-aware return-to-base strategies offer additional extensions requiring independent experimental verification.

A final hardware evaluation should include replicated tests on multiple floor materials, debris types, obstacle layouts, and battery charge levels. Statistical treatment of the resulting measurements would enable quantitative comparison against conventional reactive cleaners and coverage-planning systems.

C) Engineering Discussion

The architecture uses a deterministic control loop rather than a computationally intensive global navigation pipeline. This trade-off simplifies embedded execution and makes sensor-to-actuator causality observable. Reactive obstacle avoidance nevertheless provides local safety behavior only. Coverage completeness requires workspace memory, external markers, systematic trajectories, or localization feedback, and cannot be inferred from obstacle avoidance alone.

The cleaning module creates a second independent functional axis. A mobile robot may navigate reliably while producing insufficient suction or losing debris at the inlet.

Mechanical nozzle geometry, filter pressure drop, fan efficiency, and flow leakage should therefore be evaluated independently of wheel command accuracy. Coordinating the blower and drive supplies is also necessary to prevent controller brownouts during simultaneous starting currents.

D) Quantitative Validation and Reporting

A future validation dataset should maintain trial-level observations for obstacle distance, state transitions, surface type, debris specification, starting and recovered mass, supply voltage, current, and elapsed time. Replicate tests reduce the risk that one favorable demonstration is interpreted as general performance. When measurement resolution is limited, uncertainty should be explicitly indicated rather than reporting excessive decimal precision.

The system architecture establishes a coherent embedded-control and verification framework. Existing cleaning-robot studies establish feasibility of the component selection, but their reported numerical performance is not transferable to a new chassis without matching test conditions [3]–[5], [10]. Accordingly, deployment decisions should be supported by independent integrated-hardware testing.

E) Hardware Integration and Safety

A removable collection bin, accessible switchgear, mechanically protected sensor wiring, and adequate spacing around the suction intake improve maintainability. Emergency shutoff, thermal protection of the blower stage, and the correct battery protection circuitry must be incorporated before long-duration operation. The cleaning footprint should not extend beyond the range in which obstacles can be detected and avoided safely.

A structured firmware state machine also improves traceability. Each actuator command can be associated with the latest validated sensor measurement and the applicable safety threshold. This improves troubleshooting when obstacle avoidance and dust-collection functions operate simultaneously and supports subsequent upgrades without changing the baseline evaluation definitions.

The resulting evaluation hierarchy is explicit: sensor and state logic are assessed through deterministic input–output checks; mechanical mobility is assessed by obstacle clearance and area coverage; dust collection is assessed by recovered mass; and electrical efficiency is assessed by power draw and operating time. Each metric provides independent evidence of system behavior and avoids conflating successful navigation with effective cleaning.

F) Operational Reliability and Maintenance

Operational reliability requires a test sequence that begins with independent subsystem checks and progresses to integrated operation. The distance sensor is examined over its intended detection range; wheel drive is assessed for direction and speed consistency; the suction module is examined for inlet obstruction and filter loading; and the battery supply is monitored under concurrent actuation. Integration tests subsequently examine whether sensor-processing latency, motor switching, and blower startup interfere with one another. These procedures produce reproducible information for evaluating the safety and maintainability of the assembled electromechanical system.

Regular maintenance consists of inspecting the dust container, cleaning the filter, confirming the alignment of the front proximity sensor, tightening wheel assemblies, and examining wires for abrasion. The controller should enter a defined safe state following a sensor timeout or critical supply condition. An event record containing state transitions and fault codes supports root-cause analysis while minimizing unnecessary intervention. Together, electrical protection, controlled navigation, consistent sensing, and accessible mechanical components establish a basis for reliable indoor vacuum-cleaning operation.

REFERENCES

[1] H. Choset, “Coverage for robotics—A survey of recent results,” Annals of Mathematics and Artificial Intelligence, vol. 31, pp. 113–126, 2001.

[2] E. Galceran and M. Carreras, “A survey on coverage path planning for robotics,” Robotics and Autonomous Systems, vol. 61, no. 12, pp. 1258–1276, 2013.

[3] Y. Irawan, Muhardi, R. Ordila, and R. Diandra, “Automatic floor cleaning robot using Arduino and ultrasonic sensor,” Journal of Robotics and Control, doi:10.18196/jrc.2485.

[4] M. Kumar, Nikhil, U. Maurya, S. Giri, and P. K. Verma, “Arduino based smart vacuum cleaner robot,” Int. J. Novel Research and Development, vol. 8, no. 5, pp. h143–h145, 2023.

[5] J. M. S. Waworundeng, C. Deeng, and G. R. Deeng, “Prototype of mobile robot vacuum cleaner based on microcontroller,” Int. Scholars Conf., vol. 5, no. 1, 2017.

[6] Arduino, “UNO R3: Technical documentation,” Arduino Documentation. https://docs.arduino.cc/hardware/uno-rev3.

[7] STMicroelectronics, “L298 dual full-bridge driver,” product datasheet. https://www.st.com/en/motor-drivers/l298.html.

[8] H. Choset and P. Pignon, “Coverage path planning: The boustrophedon decomposition,” Proc. Int. Conf. Field and Service Robotics, pp. 216–222, 1997.

[9] H. Choset, “Coverage of known spaces: The boustrophedon cellular decomposition,” Autonomous Robots, vol. 9, no. 3, pp. 247–253, 2000.

[10] M. Manasa, T. S. Vidyashree, V. Bindushree, S. Rao, and P. S. Gowra, “Smart vacuum cleaner,” Global Transitions Proceedings, vol. 2, no. 2, pp. 553–558, 2021.

[11] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, Introduction to Autonomous Mobile Robots, 2nd ed. MIT Press, 2011.

[12] S. M. LaValle, Planning Algorithms. Cambridge University Press, 2006.

[13] S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics. MIT Press, 2005.

[14] B. Siciliano and O. Khatib, eds., Springer Handbook of Robotics, 2nd ed. Springer, 2016.

[15] K. M. Lynch and F. C. Park, Modern Robotics: Mechanics, Planning, and Control. Cambridge University Press, 2017.

[16] P. Corke, Robotics, Vision and Control: Fundamental Algorithms in MATLAB, 2nd ed. Springer, 2017.

[17] J. J. Craig, Introduction to Robotics: Mechanics and Control, 3rd ed. Pearson, 2005.

[18] M. W. Spong, S. Hutchinson, and M. Vidyasagar, Robot Modeling and Control. Wiley, 2006.

[19] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, Robotics: Modelling, Planning and Control. Springer, 2009.

[20] H. R. Everett, Sensors for Mobile Robots: Theory and Application. A K Peters, 1995.

[21] J. Borenstein, H. R. Everett, and L. Feng, Navigating Mobile Robots: Systems and Techniques. A K Peters, 1996.

[22] R. Murphy, Introduction to AI Robotics, 2nd ed. MIT Press, 2019.

[23] G. Dudek and M. Jenkin, Computational Principles of Mobile Robotics, 2nd ed. Cambridge University Press, 2010.

[24] A. Kelly, Mobile Robotics: Mathematics, Models, and Methods. Cambridge University Press, 2013.

[25] R. C. Arkin, Behavior-Based Robotics. MIT Press, 1998.

[26] K. J. Åström and R. M. Murray, Feedback Systems: An Introduction for Scientists and Engineers. Princeton University Press, 2008.

[27] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," Trans. ASME, vol. 64, pp. 759–768, 1942.

[28] A. V. Oppenheim and A. S. Willsky, Signals and Systems, 2nd ed. Prentice Hall, 1997.

[29] F. Pratama Anamendra and O. Candra, "Dust cleaning device based on ultrasonic sensors using Arduino Mega," Journal of Information Technology and Computer Engineering, doi:10.24036/w479d291.

[30] I. Mokashi et al., "Automated cleaning robot with smart navigation and control," Journal of Mechatronics and Robotics, 2025.