International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

AI Robot with Obstacle Detector

Design and Implementation of a Low-Cost Autonomous Obstacle-Avoidance Robot Using Servo-Scanned Ultrasonic Sensing

Bishnu Gupta[1], Kalyani Suresh Mantri[1]

[1]Department of Electronics Engineering, M.S. Bidve Engineering College, Latur, Maharashtra, India Guide: **Dr. S. R. Dhanure** (Mudda)

Abstract— This paper presents the design and implementation of a compact, low-cost autonomous mobile robot that detects nearby obstacles and changes its direction without human intervention. The prototype uses an Arduino Nano as the controller, an HC-SR04 ultrasonic ranging module mounted on an SG90 servo motor, an L298N dual H-bridge motor driver, two BO geared DC motors, and a rechargeable battery pack. The controller measures the forward distance and, when an obstacle is detected within a programmed threshold of 20 cm, stops the robot, reverses briefly, scans the right and left directions, compares the measured clearances, and turns toward the side with greater available space. The software is implemented in the Arduino IDE using embedded C/C++ and the Servo and NewPing libraries. Indoor tests described in the project report indicate obstacle detection over approximately 10–70 cm for the tested objects and generally stable movement on smooth surfaces. The work demonstrates a practical embedded-systems platform for basic reactive navigation. Quantitative reaction-time, detection-error, and success-rate measurements were not recorded; these are identified as priorities for further evaluation.

Keywords— autonomous mobile robot, obstacle avoidance, Arduino Nano, HC-SR04, ultrasonic ranging, servo scanning, L298N, embedded systems.

1. INTRODUCTION

Autonomous mobile robots must sense their surroundings and modify their motion to reduce the risk of collision. A basic obstacle-avoidance platform combines a sensing unit, a controller, a motor driver, and a locomotion mechanism. The sensing unit estimates the distance to nearby objects; the controller interprets the measurements and selects a movement action; and the motor driver translates low-power control signals into the current required by the motors. The reliability of the complete system therefore depends on the coordination of sensing, decision-making, and actuation.

Low-cost ultrasonic modules are widely used in educational mobile robotics because they estimate distance without physical contact. A single fixed sensor, however, mainly observes the region directly ahead of the robot. This project addresses that limitation by mounting the ultrasonic sensor on a small servo motor so that the sensing direction can be changed to inspect the front, right, and left regions. The controller then selects a turn based on the side with the larger measured clearance.

The term “AI” appears in the original project title. The implemented controller described here is a deterministic, rule-based reactive controller; it does not use a trained machine-learning model, neural network, or learning-based policy. The paper therefore describes the demonstrated system accurately as an autonomous obstacle-avoidance robot.

2. RELATED WORK

Ultrasonic sensing and threshold-based decisions are established approaches in small mobile robots. Khaleel and Oleiwi describe a servo-mounted ultrasonic sensor and a distance-based decision process for maze navigation, illustrating how directional scanning can support reactive movement [1]. Other implementations use multiple fixed ultrasonic sensors or infrared sensors to expand the observed area. Compared with a fixed single-sensor arrangement, servo scanning reuses one ranging module across several directions, reducing sensor count at the cost of sequential measurements and servo settling time. The present prototype adopts this compact scanning approach and a simple clearance-comparison rule.

3. SYSTEM ARCHITECTURE AND HARDWARE

The robot follows the sequence **sense–process–decide–act**. The HC-SR04 measures the distance to an object by transmitting an ultrasonic burst and timing the returned echo. The SG90 servo positions the sensor at selected angles. The Arduino Nano processes the measured distance and issues direction commands. The L298N provides two H-bridge channels for controlling the left and right BO motors. Two 3.7 V lithium cells connected in series provide a nominal battery voltage of approximately 7.4 V, as described in the project report.



Distance feedback □ decision □ motor actuation

Fig. 1. Functional signal flow of the servo-scanned obstacle-avoidance robot.

3.1. Main components

Arduino Nano: acts as the central controller. It generates the ultrasonic trigger pulse, measures the echo duration, commands the servo position, evaluates the threshold conditions, and controls motor direction. The classic Nano is based on the ATmega328P platform [2].

HC-SR04 ultrasonic module: provides non-contact distance measurement through TRIG and ECHO signals. The measured round-trip time is converted to distance using $d = (v \times t)/2$, where v is the speed of sound and t is the echo travel time. Dividing by two accounts for the outgoing and returning path.

SG90 servo motor: rotates the ultrasonic module to predefined positions. In the program, the sensor is directed toward approximately 50° for right scanning, 115° for the forward/central position, and 170° for left scanning.

L298N motor driver: is a dual full-bridge driver used to control the two DC motors independently. Its input signals determine motor direction; the enable inputs can be used for speed control where PWM wiring is implemented [3].

BO geared DC motors: provide differential-drive movement. Forward motion is produced by driving both motors in the forward direction. Opposite directions on the two sides produce a pivoting turn.

3.2. Power and integration

The project uses a shared ground reference between the controller, sensor, servo, and motor-driver electronics. The battery pack supplies the motor-driver stage, while regulated logic power is required for the controller and low-voltage modules. Motor current should not be drawn directly from an Arduino I/O pin. The power arrangement must be checked against the exact motor-driver module and Nano board variant before publication or replication; the original report describes a 7.4 V pack and a regulated 5 V logic supply.

3.3. Software configuration

The software is developed in the Arduino IDE using embedded C/C++. The report identifies the Servo.h and NewPing.h libraries. The code listing uses the ultrasonic trigger and echo pins A1 and A2, the servo signal on digital pin 10, and motor direction pins 7, 6, 4, and 5. The report contains a separate wiring subsection with a different pin assignment; therefore, the code pin map should be treated as the implementation listing and reconciled with the final physical wiring diagram before submission.

4. CONTROL METHOD AND IMPLEMENTATION

The controller uses a threshold-based reactive algorithm. Let d_f denote the forward distance, d_r the right-side distance, and d_l the left-side distance. The programmed obstacle threshold is T = 20 cm. A reading greater than T permits forward movement. When d_f is at or below T, the robot stops and performs a short reverse movement to create turning clearance. It then scans right and left and selects the direction associated with the larger measured distance.

4.1. Algorithm

Step 1: Initialize motor pins, ultrasonic ranging, servo control, and the run timer.
Step 2: Position the servo at the forward angle and obtain a distance reading.
Step 3: If the forward distance is greater than 20 cm, command both motors to move forward.
Step 4: If the forward distance is 20 cm or less, stop the motors and reverse briefly.
Step 5: Rotate the servo to the right, allow it to settle, and record d_r.
Step 6: Rotate the servo to the left, allow it to settle, and record d_l.
Step 7: If d_r □ d_l, turn right; otherwise, turn left.
Step 8: Return to forward scanning and repeat the control cycle. The code listing also includes a 20-second run-time stop condition.

4.2. Pseudocode

```
INITIALIZE controller, sensor, servo, motors WHILE run_time < 20
seconds:
    d_forward=measure_distance() IF
    d_forward > 20 cm:
        move_forward()
    ELSE:
        stop_motors()
        move_backward_briefly()
        d_right=scan_right_and_measure() d_left =
        scan_left_and_measure() IF d_right>=d_left:
        turn_right() ELSE: turn_left()
STOP motors
```

4.3. Distance estimation and decision rule

The ultrasonic distance is estimated from the echo duration according to:

$$d = (v \times t)/2 \quad (1)$$

where d is the one-way distance, v is the speed of sound in air, and t is the measured round-trip time. In practice, the NewPing library performs the timing and conversion. The code maps a zero centimetre return to a large default value (250 cm); this avoids treating a missing echo as an immediate obstacle, but it can also interpret a failed measurement as an open path. A safer implementation should distinguish a timeout from a valid long-distance reading.

5. EXPERIMENTAL OBSERVATIONS

The project report describes indoor tests using cardboard blocks, bottles, walls, and angled surfaces. The tests considered obstacle detection, reaction, movement stability, and collision avoidance. The sensor reportedly detected the tested objects over an observed range of approximately 10–70 cm. Flat surfaces produced more consistent readings, while soft or curved surfaces sometimes returned weaker echoes. The robot moved smoothly on flat flooring and changed direction when an obstacle was placed in front of it. In tight spaces, the reverse-then-turn sequence was used to create additional clearance.

The report characterizes the response as quick and the turns as stable under normal test conditions, but it does not provide instrumented reaction-time values, sample counts, distance-error statistics, battery-runtime measurements, or a numerical obstacle-avoidance success rate. Accordingly, no quantitative performance claim is made in this paper. The 20 cm threshold is a programmed control parameter, not a statistically optimized value.

6. DISCUSSION

The prototype demonstrates that a low-cost microcontroller, a servo-mounted ultrasonic module, and a dual-motor drive can provide basic autonomous obstacle avoidance. The main design advantage is directional reuse of one ultrasonic sensor: the servo allows the controller to compare free space on both sides without installing separate ranging modules. The algorithm is computationally simple and can run on a small 8-bit microcontroller. The motor driver isolates the low-power controller outputs from the motor load and supports bidirectional movement through its two H-bridge channels [3].

Several limitations should be considered. First, the sensor samples directions sequentially rather than simultaneously; an obstacle moving during the scan may not be represented by a consistent set of measurements. Second, ultrasonic performance depends on target shape, surface orientation, and acoustic reflection. Third, the fixed 20 cm threshold and fixed turning duration may not be suitable for every speed, chassis geometry, or floor condition. Fourth, the reported software contains a 20-second automatic stop, so the demonstrated program is a short-duration prototype rather than a continuous-duty navigation controller. Finally, the original project documentation contains inconsistent pin assignments between the wiring description and code listing, which must be corrected in the final publication package.

For stronger experimental validation, future tests should report the number of trials, obstacle materials and angles, approach speeds, measured reaction time, distance error against a reference instrument, successful avoidance percentage, and battery operating time. A timeout-aware sensor routine, median filtering, calibrated turning duration, PWM speed regulation, and additional IR or ultrasonic sensors could improve robustness. More advanced work may incorporate mapping, SLAM, fuzzy logic, or learning-based navigation; these are proposed extensions and are not features of the present implementation.

7. CONCLUSION

A compact autonomous obstacle-avoidance robot was designed and implemented using an Arduino Nano, an HC-SR04 ultrasonic sensor mounted on an SG90 servo, an L298N motor driver, and two BO geared motors. The controller measures forward clearance, applies a 20 cm obstacle threshold, scans the right and left directions when the path is blocked, and turns toward the side with the larger measured clearance. The project report records successful basic indoor obstacle-avoidance behavior and stable movement on smooth surfaces. The design is appropriate as an educational embedded-robotics prototype. Further work should provide repeatable quantitative testing, resolve the documented pin-map inconsistency, and improve sensing reliability before claims of high-speed, intelligent, or safety-critical navigation are made.

ACKNOWLEDGMENT

The authors acknowledge the guidance of Dr. S. R. Dhanure (Mudda) and the support of the Department of Electronics Engineering, M.S. Bidve Engineering College, Latur.

REFERENCES

[1] H. Z. Khaleel and B. K. Oleiwi, “Ultrasonic sensor decision-making algorithm for mobile robot motion in maze environment,” *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 1, pp. 109–116, Feb. 2024, doi: 10.11591/eei.v13i1.6560.

[2] Arduino, “Arduino Nano—Hardware Documentation.” Available: https://docs.arduino.cc/hardware/nano (accessed Nov. 18, 2026).

[3] STMicroelectronics, “L298: Dual Full-Bridge Driver—Datasheet,” DS0218, Rev. 5, Oct. 2023. Available: https://www.st.com/resource/en/datasheet/l298.pdf.

[4] HC-SR04 Ultrasonic Ranging Module, “Technical Datasheet.” The exact manufacturer/version of the module used in the prototype should be confirmed and cited in the final camera-ready manuscript.

[5] TowerPro, “SG90 Micro Servo,” product documentation. The exact revision used in the prototype should be confirmed before final submission.

Publication note: This is a manuscript draft prepared from the supplied project report. Before submitting it to a journal or conference, confirm author order and affiliations, verify component datasheet versions, add measured experimental data, and follow the target venue’s template and reference style. The paper has not been peer reviewed and publication is not guaranteed.