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

Design and Evaluation Framework for Ultrasonic Obstacle Detection and Collision Avoidance in an Arduino-Based Mobile Robot

Prof. Shivaprasad[1], Akshaya Gulvady[2], Dhanush Raj BR[3], Manikanta BM[4]

[1]Assistant Professor, Department of Information Science and Engineering, East West Institute of Technology, Bengaluru, Karnataka, India.
[2]UG Scholar (1EW23IS004), Department of Information Science and Engineering, East West Institute of Technology, Bengaluru, Karnataka, India.
[3]UG Scholar (1EW23IS025), Department of Information Science and Engineering, East West Institute of Technology, Bengaluru, Karnataka, India.
[4]UG Scholar (1EW23IS053), Department of Information Science and Engineering, East West Institute of Technology, Bengaluru, Karnataka, India.

***Abstract*— Low-cost autonomous mobile robots commonly avoid collisions with a single ultrasonic sensor and a fixed distance threshold, yet such prototypes are usually reported as working demonstrations rather than as systems whose measurement behaviour has been characterised. This paper documents an implemented Arduino-based mobile robot built from an Arduino Uno R3, an HC-SR04 ultrasonic sensor, an L293D motor driver, two DC geared motors, a battery supply and a wheeled chassis, and defines a structured framework for its evaluation. The paper is therefore a design, implementation and evaluation-framework contribution, not an experimental study. The robot moves forward, derives the distance ahead from the echo duration, compares it with a programmed threshold and, when an obstacle lies within the threshold, stops, performs a programmed turning action and resumes movement. The mechanism is purely threshold-based; no artificial intelligence, camera, additional sensor or sensor fusion is used. The contribution is neither a new algorithm nor a new device. It is (i) a description of the platform with component limits taken from manufacturer documentation, (ii) a proposed reference wiring and control sequence, kept separate from the prototype records, and (iii) a six-experiment protocol with defined error and success metrics covering distance-measurement error, obstacle material, obstacle angle, sudden obstacle appearance, repeatability, and stopping and turning behaviour. The six experiments are proposed procedures; quantitative validation remains pending, and no measured performance is reported. The prototype's pin mapping, threshold, turning parameters and battery rating are not documented in the records available for this paper. Practical limitations of single-sensor ultrasonic avoidance are discussed.**

***Keywords*— Ultrasonic sensor, HC-SR04, Arduino Uno, mobile robot, obstacle detection, collision avoidance, L293D motor driver, embedded systems, evaluation framework.**

I. INTRODUCTION

A. Background

Autonomous mobile robots (AMRs) move through their environment without continuous human control and are used in logistics, inspection, service and education [1]. A basic requirement is obstacle detection: the robot must perceive an object in its path early enough to avoid it. Collision avoidance then closes the control loop by turning a measurement into a motion decision.

Ultrasonic time-of-flight sensors are widely used for this purpose because they are inexpensive, simple to interface and unaffected by ambient lighting, although their readings depend on the reflecting surface [2], [3]. The Arduino platform suits educational and low-cost prototypes because of its open tools, low price and large community, and several Arduino robots with ultrasonic sensing have been reported [4]–[6].

B. Motivation and Problem Statement

An ultrasonic reading depends on the beam geometry, the reflecting surface and the angle of incidence [2], [3]. A robot that stops before a flat board in a demonstration may behave differently with soft or inclined obstacles, or with an obstacle that appears suddenly in its path. For a low-cost robot, knowing the conditions under which the sensor and the fixed threshold can be trusted is as useful as the build itself, which is why measured evaluation is needed.

Problem statement. Single-sensor threshold robots are rarely characterised for measurement error, obstacle material, obstacle angle, response behaviour and repeatability together. This work therefore asks how reliably the implemented prototype can detect and avoid obstacles when these factors are varied under controlled conditions, and how that should be measured.

C. Research Objectives

The paper has two aims: to document the design of the implemented robot, and to define an evaluation protocol for it. The protocol has six objectives:

1) measure the HC-SR04 distance error at known reference distances;
2) compare detection for different obstacle materials;
3) compare detection at different obstacle angles;
4) evaluate the response to an obstacle that appears suddenly;
5) quantify the repeatability of distance readings; and
6) evaluate stopping and turning behaviour over repeated trials.

D. Contributions and Scope

- A description of the implemented single-sensor Arduino robot, limited to what the available project records support.
- Manufacturer-documented limits of the Arduino Uno R3, HC-SR04 and L293D, with the L293D logic supply (V_{CC1}) and motor supply (V_{CC2}) distinguished.
- A proposed reference wiring and control sequence, clearly separated from the prototype details.
- A six-experiment evaluation protocol with defined metrics and calculation rules.

The hardware and the threshold algorithm are not claimed to be new, and no superiority over other work is claimed. Four kinds of statement are kept apart: (i) the description of the implemented robot; (ii) component specifications from manufacturer documentation; (iii) proposed reference designs,

such as the example wiring and control sequence; and (iv) experiments that are proposed and pending validation. Prototype details that are not documented in the available records (pin mapping, threshold, comparison operator, turning parameters, battery rating and motor ratings) are stated as such in Tables II and III and in Section IX.

Type of paper. This is a design, implementation and evaluation-framework paper. It documents what the prototype is and specifies how its sensing and avoidance behaviour should be measured; it does not report measurements. The title therefore says "evaluation framework" and not "experimental evaluation", and a measured-results version can follow once Experiments 1–6 are carried out.

The remainder of the paper is organised as follows. Section II reviews related work, Section III states the research gap, Section IV gives the methodology and equations, Sections V–IX describe the architecture, hardware, software, circuit connections and algorithm, Section X defines the experimental methodology, and Sections XI–XIV cover the status of results, limitations, conclusion and future scope.

II. RELATED WORK AND LITERATURE REVIEW

The works below are a small, non-exhaustive sample of foundational sonar and obstacle-avoidance papers and of Arduino-based ultrasonic robots; this is not a systematic review.

Ultrasonic sensing and its limits. Kleeman and Kuc review sonar principles and the phenomena that limit ranging, such as wide beams and specular reflection [3]. Borenstein and Koren discuss the limitations of ultrasonic sensors for mobile-robot obstacle avoidance, including poor directionality and erroneous readings from reflecting surfaces [2].

Collision-avoidance techniques. Khatib formulated real-time obstacle avoidance with artificial potential fields [7]. Occupancy grids built from sonar data support mapping [8], and the vector field histogram steers a robot using ultrasonic data held in a histogram grid [9]. These approaches need more computation, and usually more sensors, than a threshold rule.

Arduino-based robots and low-cost evaluation. Sukri *et al.* report an Arduino Uno robot with an HC-SR04 and an L298N driver, and tabulate stopping-distance errors for set distances of 5–30 cm [4]. Varma *et al.* describe an Arduino Uno obstruction-avoiding robot with a servo-mounted ultrasonic sensor [5]. Sharma *et al.* describe an ultrasonic object-sensing robot with an Arduino Uno, a Wi-Fi module and a motor-shield driver [6]. Table I compares these works.

TABLE I
COMPARISON OF RELATED WORK

Ref.	Technique / hardware	Main content	Limitation relevant here
[2] 1988	Ultrasonic range sensors on a mobile robot	Identifies sensor limitations such as poor directionality and specular reflection	Wider avoidance scheme, not a low-cost threshold robot
[3] 2008	Sonar sensing review	Explains time-of-flight ranging and reflection phenomena	General survey; no Arduino platform evaluation
[7] 1986	Artificial potential fields	Continuous real-time avoidance formulation	Needs obstacle-position knowledge and computation beyond a threshold rule
[8] 1989	Occupancy grids; sonar	Probabilistic spatial map from sonar readings	Mapping overhead; not a simple reactive controller
[9] 1991	Vector field histogram; ultrasonic sensors	Fast avoidance with statistical handling of sonar inaccuracy	Designed around multiple sensors and more processing
[4] 2021	Arduino Uno, HC-SR04, L298N; threshold stop	Stopping tests at set distances of 5–30 cm; mean relative error 0.118 as tabulated (≈11.8 %); largest error at 30 cm, where the robot stopped at 24 cm	Frontal obstacle only; no material, angle or response-time study
[5] 2019	Arduino Uno; ultrasonic sensor on a servo	Obstruction-avoiding robot; servo for a wider field of view	Abstract reports no quantitative sensor characterisation
[6] 2025	Arduino Uno, Wi-Fi module, motor shield	Object sensing and distance measurement; good performance under various illumination conditions	Qualitative in abstract; no material or angle analysis stated

Entries for [5] and [6] are based on the published abstracts; the entry for [4] is based on the full text. The figures for [4] were checked against its full text.

III. RESEARCH GAP AND CONTRIBUTION

In the Arduino-based works reviewed here, measurement error is reported at most for frontal flat obstacles, and obstacle material, obstacle angle, sudden appearance and repeatability are not jointly characterised for a single-sensor robot. This paper addresses that gap by defining an evaluation protocol for an implemented prototype, not by proposing a new algorithm. The implemented robot uses ordinary, widely available hardware and a conventional threshold-based avoidance rule; neither is presented as an invention. The contribution lies in documenting the platform with its component limits, in separating documented prototype details from proposed ones, and in specifying the experiments and metrics by which its behaviour can be characterised.

IV. PROPOSED METHODOLOGY

The implemented robot works as follows: (1) the robot moves forward; (2) the ultrasonic sensor detects an obstacle and the Arduino measures the echo duration; (3) the distance is calculated; (4) the distance is compared with a programmed threshold; (5) when the obstacle is within the threshold, the robot stops; (6) it performs a programmed turning action; and (7) it resumes movement. The study proceeds in ten steps:

1) *Hardware assembly:* mount the motors and wheels on the chassis and connect the L293D to the motors and battery.
2) *Sensor installation:* fix the HC-SR04 at the front, facing forward.
3) *Arduino configuration:* connect the trigger, echo and driver lines and load the program with the Arduino IDE.
4) *Distance measurement:* trigger the sensor and time the echo.

5) *Threshold comparison:* compare the distance with the programmed threshold.
6) *Motor control:* set the L293D inputs.
7) *Obstacle avoidance:* stop and turn when an obstacle is within the threshold.
8) *Experimental testing:* run Experiments 1–6 (Section X).
9) *Data collection:* record readings and trial outcomes.
10) *Performance evaluation:* compute the metrics below.

The distance follows from the echo time:

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

where d is the one-way distance to the obstacle (m), v the speed of sound (m/s) and t the echo time (s). Division by two is necessary because the pulse travels to the obstacle and back. The speed of sound varies with air temperature T (°C):

$$v \approx 331.3 + 0.606\ T \quad (2)$$

With t in microseconds and d in centimetres,

$$d_{cm} = v \times t_{\mu s} / (2 \times 10^4) \quad (3)$$

The measurement error, absolute error and percentage error of a reading d_m at a known reference distance d_{ref} (cm) are

$$e = d_m - d_{ref} \quad (4)$$

$$|e| = |d_m - d_{ref}| \quad (5)$$

$$PE = (|e| / d_{ref}) \times 100 \quad (6)$$

For N readings, the mean absolute error (cm) and the sample standard deviation (cm) are

$$MAE = (1/N)\ \Sigma\ |e_i|,\ \ i = 1 \ldots N \quad (7)$$

$$\sigma = \sqrt{[\ (1/(N-1))\ \Sigma\ (d_i - \bar{d})^2\ ]} \quad (8)$$

where d_i is reading i and $\bar{d}$ their mean (cm). The detection rate and the avoidance success rate are

$$DR = (N_{det} / N) \times 100\ \% \quad (9)$$

$$SAR = (N_{succ} / N) \times 100\ \% \quad (10)$$

where N_{det} is the number of readings or trials in which the obstacle was detected and N_{succ} the number of trials in which the robot stopped and turned without collision. For the filmed experiments (4 and 6) three frame numbers are read from the video: n_0, the first frame in which the obstacle is in the robot's path; n_{react}, the first frame in which the robot visibly begins to slow or stop; and n_{stop}, the first frame in which the wheels are stationary. With the frame rate f in frames/s, the times in milliseconds are the obstacle-to-reaction delay, the stopping time and the total response time:

$$t_{det} = 1000\ (n_{react} - n_0) / f \quad (11)$$

$$t_{stop} = 1000\ (n_{stop} - n_{react}) / f \quad (12)$$

$$t_r = t_{det} + t_{stop} = 1000\ (n_{stop} - n_0) / f \quad (13)$$

The obstacle-to-reaction delay t_{det} is what the video shows between the obstacle being in the robot's path and the robot visibly beginning to react. It contains the sensor measurement cycle, the program's decision and the driver and motor response, so it is not the internal detection latency of the ultrasonic sensor, which the video does not show, and must not be read as such. Each time read from video is uncertain by about one frame (16.7 ms at 60 frames/s).

For distances, let x be the position of the front of the robot along its direction of travel, measured with the tape from a marked start line and increasing in the direction of travel (cm). Let x_0, x_{react} and x_{stop} be its positions at frames n_0, n_{react} and n_{stop}, and x_{board} the position of the near face of the board on the same axis. The distance travelled during the obstacle-to-reaction delay, the stopping distance, the total travel after the obstacle appears, and the clearance at standstill are

$$s_{det} = x_{react} - x_0 \quad (14)$$

$$s = x_{stop} - x_{react} \quad (15)$$

$$s_{tot} = x_{stop} - x_0 = s_{det} + s \quad (16)$$

$$g = x_{board} - x_{stop} \quad (17)$$

Because the robot moves forward, x increases and s_{det}, s and s_{tot} are non-negative; a negative value indicates a reading error or backward motion and is reported, not discarded. A clearance $g > 0$ means that the robot stopped without touching the board; $g \le 0$ is a collision. No value of DR, SAR, MAE, σ, t_{det}, t_{stop}, t_r, s_{det}, s, s_{tot} or g is computed in this paper, because the experiments have not yet been performed.

V. SYSTEM ARCHITECTURE

The signal chain is: HC-SR04 ultrasonic sensor → Arduino Uno R3 → obstacle-detection logic → L293D motor driver → DC motors → robot movement (Fig. 1).

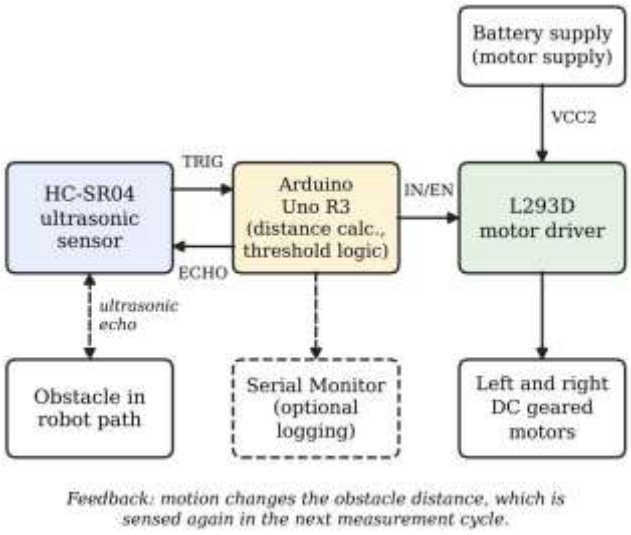


Fig. 1. System and hardware block diagram of the robot. Serial logging is optional and is required only for the evaluation protocol.

Hardware architecture. The sensor is mounted at the front and faces forward. The Arduino is the only processing unit. The L293D sits between the Arduino logic outputs and the motors, which draw their current from the battery rather than from Arduino pins. Sensor-to-controller communication uses two digital lines, a trigger output and an echo input [10]; controller-to-driver communication uses digital direction inputs and enable inputs of the L293D [12].

Software architecture. One program consists of an initialisation routine and a repeating loop. The loop contains the functional blocks of distance measurement, threshold decision and motor control. Reporting the measured distance and state over the serial port is an optional block that the evaluation protocol requires for

recording readings; whether the prototype already provides it is not documented in the available records. Data flow: echo duration → distance value → decision (clear or obstacle) → motor-driver input states → motion → new echo. Fig. 2 shows the working sequence of one control cycle.

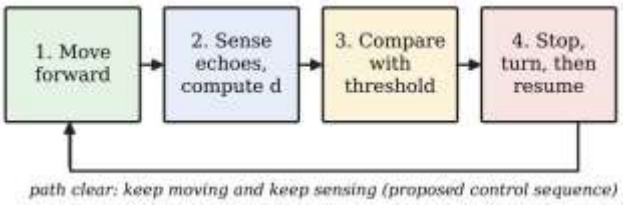


Fig. 2. Robot working sequence (one control cycle).

VI. HARDWARE REQUIREMENTS

The prototype is built from an Arduino Uno R3, an HC-SR04 ultrasonic sensor, an L293D motor driver, two DC geared motors, a battery supply and a wheeled chassis. Table II lists the component specifications and the basis of each entry. Values taken from manufacturer documentation describe the components, not the measured performance of the robot.

TABLE II
COMPONENT SPECIFICATIONS

Component	Specification	Basis
Arduino Uno R3	ATmega328P microcontroller; 5 V operating voltage; recommended input 7–12 V (limit 6–20 V); 14 digital I/O pins, 6 with PWM (D3, D5, D6, D9, D10, D11); 20 mA DC recommended per I/O pin [11]	Manufacturer documentation
HC-SR04	40 kHz; supply 5 V DC (4.5–5.5 V); working current 15 mA typical (10–20 mA); range 2–400 cm; resolution 0.3 cm; effectual angle below 15°, measuring angle 30°; trigger pulse of at least 10 µs; module size 45 × 20 × 15 mm [10]	Manufacturer documentation
L293D	Quadruple half-H driver (two full H-bridges). Logic supply V_{CC1}: 4.5–7 V. Motor supply V_{CC2}: V_{CC1} up to 36 V. Continuous output current 600 mA per channel; peak 1.2 A (non-repetitive, t ≤ 100 µs). Logic-high input threshold 2.3 V minimum. Internal clamp diodes; enable input per driver pair [12]	Manufacturer datasheet (TI SLRS008D)
Two DC geared motors	Rated voltage, speed and current are not documented in the available records; the running and stall current of the fitted motors must be compared with the 600 mA L293D limit	Not documented
Battery supply	Type, voltage and capacity are not documented. Per the project description, the battery supplies the motors through the L293D rather than through Arduino pins; the supply of the Arduino itself is not documented	Project description (partial)
Chassis and wheels	Wheeled chassis with two driven wheels; dimensions and any caster wheel are not documented	Project description (partial)

The datasheet values describe the components and are not measured performance of the robot.

L293D supplies and limits. The L293D has two separate supplies. V_{CC1} (pin 16) powers the internal logic and is limited to 4.5–7 V, so the Arduino 5 V rail is suitable. V_{CC2} (pin 8) supplies the output stage, hence the motors, and may range from V_{CC1} up to 36 V; it must come from the battery, not from an Arduino pin [12]. Each driver channel is rated at 600 mA continuous and 1.2 A peak only for non-repetitive pulses of at most 100 µs, so motors whose running or stall current exceeds 600 mA would overload the device. Because the output stage has a voltage drop (the high-level output is typically V_{CC2} − 1.4 V and at worst V_{CC2} – 1.8 V at 0.6 A [12]), the voltage across the motor is lower than the battery voltage. Pins 1 (1,2EN) and 9 (3,4EN) enable the driver pairs; the inputs 1A, 2A, 3A and 4A set the outputs 1Y–4Y, and a motor is connected between two outputs of the same pair [12]. Whether the battery voltage of the prototype lies within the permitted V_{CC2} range cannot be checked, because the battery rating is not documented.

Hardware implementation. The prototype follows the architecture of Section V on a wheeled chassis. Fig. 3 is an illustrative representation of such a robot. **It is an AI-generated illustration, not a photograph of the actual prototype**; the chassis layout, wheel type, wiring and battery holder shown in it are not documented project details.

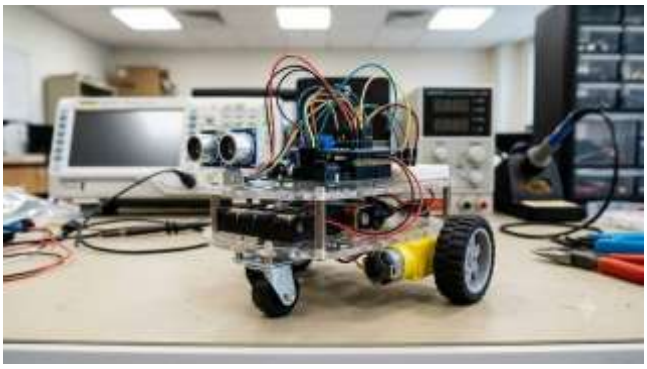


Fig. 3. Illustrative representation of an Arduino-based autonomous mobile robot (AI-generated illustration; not a photograph of the prototype).

VII. SOFTWARE REQUIREMENTS

The program is written in Embedded C/Arduino (C/C++) and developed in the Arduino IDE. The IDE version and the source code were not available for this paper and are not reproduced. The Arduino core provides digital input and output, microsecond-level delays and pulse-width measurement, which are the functions needed for the HC-SR04 trigger–echo protocol; the specific functions used in the prototype are not documented in the available records. The evaluation protocol additionally needs (i) serial logging of the measured distance and state, (ii) frame-accurate video analysis at 60 frames/s or higher for Experiments 4 and 6, and (iii) a spreadsheet or script to compute Eqs. (4)–(17). None of these tools forms part of the robot's control logic.

VIII. CIRCUIT CONNECTIONS

The actual pin mapping of the prototype is not documented in the available records and must not be inferred from the example below. To show that the described architecture is realisable, Table III gives a technically valid reference wiring for an Arduino Uno R3 and an L293D, with L293D pin numbers taken from the manufacturer pinout [12]. **It is proposed reference wiring and is not the verified wiring of the prototype.** The last column records what is documented about the prototype; it

contains no verified pin assignments, and it can be completed only from the final source code and the physical prototype. Fig. 4 shows the same reference wiring.

TABLE III
PROPOSED REFERENCE WIRING AND PROTOTYPE RECORD

Signal	Proposed reference wiring (not the prototype)	Actual prototype record
HC-SR04 Trig	D8 (output) → Trig	Arduino pin not documented in the available prototype records.
HC-SR04 Echo	D9 (input) ← Echo	Arduino pin not documented in the available prototype records.
HC-SR04 supply	V_{CC} → 5 V; GND → GND	Connection not documented in the available prototype records. Manufacturer supply: 5 V DC [10].
Left motor	D2 → 1A (pin 2); D3 → 2A (pin 7); motor across 1Y (pin 3) and 2Y (pin 6)	Arduino pins and L293D channel not documented in the available prototype records.
Right motor	D4 → 3A (pin 10); D7 → 4A (pin 15); motor across 3Y (pin 11) and 4Y (pin 14)	Arduino pins and L293D channel not documented in the available prototype records.
Enable inputs	D5 → 1,2EN (pin 1); D6 → 3,4EN (pin 9); driven HIGH to enable	Enable wiring and any PWM use not documented in the available prototype records.
Logic supply V_{CC1}	Arduino 5 V → pin 16	Not documented in the available prototype records. Permitted range: 4.5–7 V [12].
Motor supply V_{CC2}	Battery + → pin 8 (V_{CC2}; motor supply, up to 36 V); not an Arduino pin	Battery voltage and capacity not documented. Per the project description, the motors are supplied from the battery through the L293D.
Ground	Battery −, Arduino GND, L293D pins 4, 5, 12, 13 and HC-SR04 GND joined (common ground)	Common-ground connection not documented in the available prototype records; it is a design requirement for any such wiring.

The last column states what the available records document. "Not documented" means that no source code, wiring record or hardware specification was available; it does not mean that the prototype lacks the connection.

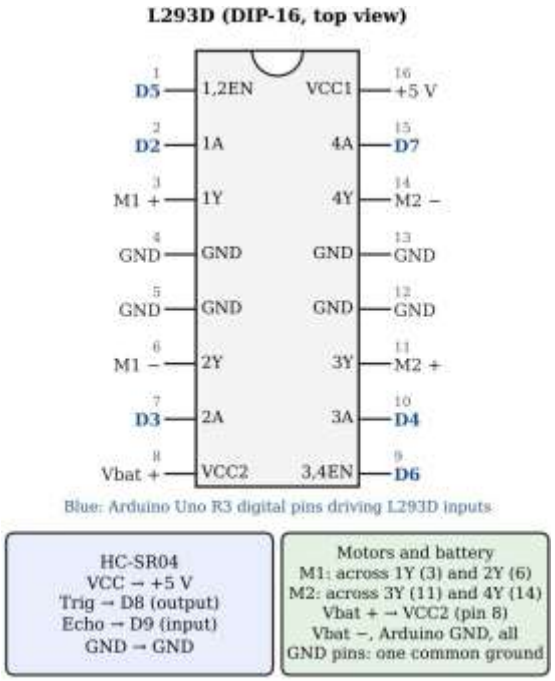


Fig. 4. Proposed reference connection diagram of the Arduino-based ultrasonic obstacle-avoidance system (consistent with Table III, column 2). It is not the verified wiring of the prototype.

Three design points apply to any wiring of this kind. First, the battery negative terminal, the Arduino GND, the L293D ground pins and the HC-SR04 GND must share one common ground. Second, V_{CC1} and V_{CC2} are distinct supplies with different limits (Section VI). Third, the HC-SR04 echo output is a 5 V TTL signal, which an Uno input accepts directly [10], [11].

IX. OBSTACLE DETECTION AND AVOIDANCE ALGORITHM

A. HC-SR04 Operation

According to the manufacturer manual, a trigger pulse of at least 10 µs makes the HC-SR04 transmit a burst of eight cycles at 40 kHz, after which the echo line goes high for a time proportional to the distance [10]. The Arduino measures this pulse width and converts it with (3); at about 20 °C, $v \approx 343.4$ m/s and (3) gives approximately $t/58.2$ cm, consistent with the $t/58$ rule given in the manual [10]. Fig. 5 illustrates the detection principle.

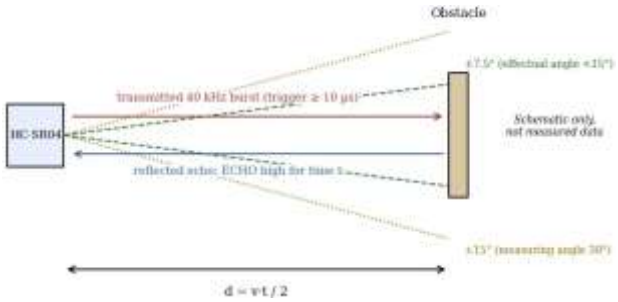


Fig. 5. Schematic of ultrasonic obstacle detection with the HC-SR04: transmitted burst, reflected echo and the manufacturer-stated beam angles [10]. Schematic only; not measured data.

B. Detection Threshold and Comparison Operator

The detection threshold programmed into the prototype is not documented in the available records. Any threshold figure used elsewhere in the paper is a design example for the proposed experiments, not the threshold of the implemented robot. A detection threshold must exceed the distance the robot travels between the sensor reading and complete standstill, plus the sensor's minimum range of 2 cm [10] and a safety margin, which is why the total travel s_{tot} (16) is measured in Experiment 4 and the stopping distance s (15) in Experiments 4 and 6.

The comparison operator is also not documented. The two candidate rules differ only at equality: with "distance < threshold" the robot continues forward when a reading equals the threshold exactly, whereas with "distance ≤ threshold" it stops. Because the sensor output is quantised (resolution 0.3 cm [10], or one integer centimetre if the program truncates), the choice changes the effective stopping distance by at most one quantisation step. The difference is small but matters for exact reproduction; the generic algorithm below assumes ≤ as a design choice only.

C. No-Echo Handling

A measurement may return no valid echo when the target lies beyond the sensor's range, when a soft or porous surface absorbs the burst, or when an inclined or smooth surface reflects it away from the receiver [2], [3], [10]. A program that waits for the echo line without a limit can then stall, and a program that treats a missing echo as "no obstacle" can drive towards an undetected object. A timeout and an explicit rule for the no-echo case therefore matter for safety. Whether the prototype has a timeout, and how it treats a missing echo, is not documented; the proposed sequence below includes one as a design recommendation only.

D. Motor Control and Turning Action

The L293D has an enable input for each pair of drivers and a direction input per driver [12]. For one motor, driving the two inputs to opposite levels with enable high runs the motor in one sense, and reversing the levels reverses it. Equal input levels with enable high put the same level on both outputs and stop the motor rapidly, and a low enable switches the outputs off so that the motor free-runs to a stop [12]. Forward movement drives both motors in the same sense; stopping removes or short-circuits the drive; turning drives the two motors so that the heading changes, either as a spot turn (motors in opposite senses) or a pivot turn (one motor stopped).

The turning direction, programmed turning duration, delay values, motor speed (including any PWM value) and the rule for resuming movement are not documented in the available records. Because turning is open loop (no encoder or gyroscope), the turn angle depends on duration, battery voltage and floor friction, and is not known unless measured.

E. Proposed Control Sequence

Fig. 6 and the pseudocode below elaborate the seven-step working principle of Section IV into a complete control loop, consistent with Fig. 2. **It is a proposed reference algorithm, not verified source-code behaviour of the prototype.** T_{STOP}, T_{TURN}, T_{PAUSE} and T_{OUT} are design parameters; for the proposed experiments a stop threshold of 30 cm is used purely as a design example.

```
params: T_STOP, T_TURN, T_PAUSE, T_OUT (design)
setup:
  TRIG = output; ECHO = input; driver pins = output
  motors stopped
loop:
  1 send 10 us pulse on TRIG
  2 t <- width of ECHO pulse; give up after T_OUT
  3 if no echo: stop motors; go to 1
  4 d <- t * v / (2 * 10^4)          (Eq. 3)
  5 if d <= T_STOP:
      stop motors; turn for T_TURN; pause T_PAUSE
    else:
      drive both motors forward
  6 (optional) report d and state over Serial
  7 go to 1
```

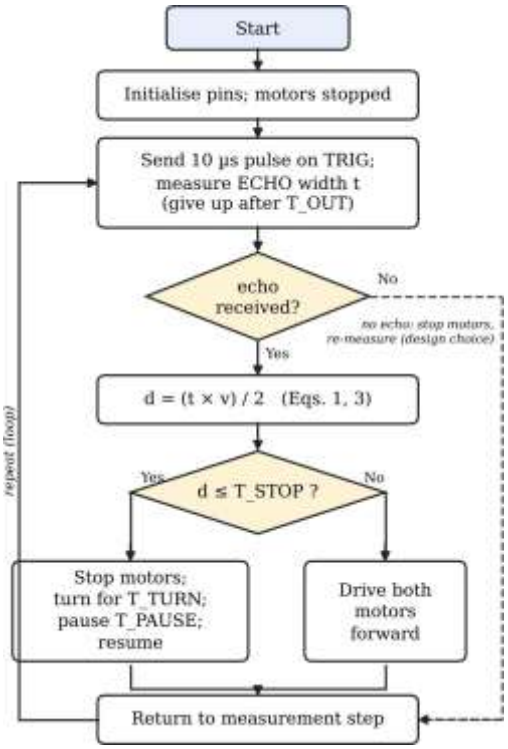


Fig. 6. Obstacle detection and avoidance flowchart (proposed control sequence; not verified prototype source code).

X. EXPERIMENTAL METHODOLOGY

This section defines the proposed experiments; none has been completed. Values labelled "proposed" are recommended test conditions, not reports of completed tests, and no results are tabulated. Table V summarises the six experiments and Table VI defines how each quantity is calculated.

A. Common Setup

Tests are performed indoors in still air on a flat floor. A measuring tape (1 mm resolution) fixes d_{ref}, measured from the sensor face to the target. Targets are flat boards of equal size. Each reading is taken from the serial output. Videos at 60 frames/s or higher are used for time and distance in Experiments 4 and 6 (one frame is then at most 16.7 ms, so each time read from video is uncertain by about one frame). Distances and times are defined in Section IV. The following should be recorded for every session: room temperature (for (2)), battery voltage before and after the session, the actual detection threshold, the robot speed (measurable from video over a known distance), the target

material and size, and the sensor height. Fig. 7 shows the arrangement for the distance and angle experiments and Table IV the common settings.

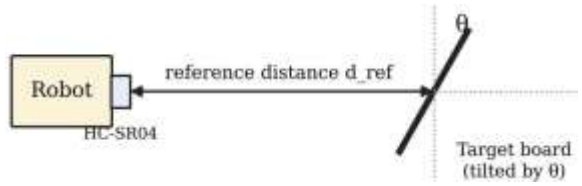


Fig. 7. Test arrangement for the distance and angle experiments.

TABLE IV
EXPERIMENTAL SETUP (PROPOSED)

Item	Setting
Robot	As implemented; actual detection threshold to be recorded before testing
Robot speed	To be measured from video
Targets	Flat boards of equal size; material per experiment
Reference distance	Tape measurement, sensor face to target
Readings or trials	10 per condition (Experiments 1–3); 10 trials (Experiment 4); 30 per distance (Experiment 5); 20 trials (Experiment 6)
Ambient temperature	To be recorded each session
Battery condition	To be recorded before and after each session

TABLE V
SUMMARY OF THE PROPOSED EXPERIMENTAL PROTOCOL

Exp.	Factor varied; fixed conditions	Trials	Recorded quantities and metrics
1	Reference distance (proposed 5, 10, 20, 30, 50, 100, 150, 200 cm); robot stationary; board perpendicular	10 per distance	Mean reading, |e|, PE, MAE: Eqs. (4)–(7)
2	Material (cardboard, wood, plastic, metal, glass, fabric-covered surface, optional extra); distance fixed (proposed 30 cm)	10 per material	Reading, detected (yes/no), |e|; DR: Eq. (9)
3	Board angle θ = 0°, 15°, 30°, 45°, 60°; cardboard at proposed 30 cm	10 per angle	Reading, detected (yes/no), |e|; DR: Eq. (9)
4	Sudden insertion of a board in the path at a proposed 30 cm (must exceed the actual threshold); robot at normal speed	10	t_{det}, t_{stop}, t_r, s_{det}, s, s_{tot}, g, collision: Eqs. (11)–(17)
5	Reference distance 20, 50, 100 cm; robot and board stationary	30 per distance	Mean, σ, minimum, maximum: Eq. (8)
6	Board across the path, perpendicular to heading; fixed start position; start from rest	20	Reaction distance $x_{board} - x_{react}$, s, g, turning success, collision; SAR: Eqs. (15), (17), (10)

TABLE VI
METRIC DEFINITIONS

Metric	Definition	Eq.	Exp.
Absolute error	$\|e\| = \|d_m - d_{ref}\|$, in cm	(5)	1–3
Percentage error	$PE = \|e\| / d_{ref} \times 100$, in %	(6)	1
Mean absolute error	Mean of |e| over N readings, in cm	(7)	1
Standard deviation	Sample σ of N readings (N−1), in cm	(8)	5
Minimum, maximum	Smallest and largest reading, in cm; outliers reported	—	5
Detection rate	Detected readings or trials / N × 100, in %	(9)	2, 3
Obstacle-to-reaction delay	$1000 (n_{react} - n_0) / f$, in ms	(11)	4
Stopping time	$1000 (n_{stop} - n_{react}) / f$, in ms	(12)	4
Total response time	$t_{det} + t_{stop} = 1000 (n_{stop} - n_0) / f$, in ms	(13)	4
Distance during obstacle-to-reaction delay	$x_{react} - x_0$, in cm	(14)	4
Stopping distance	$x_{stop} - x_{react}$, in cm (≥ 0)	(15)	4, 6
Total travel after appearance	$x_{stop} - x_0$, in cm	(16)	4
Reaction distance	$x_{board} - x_{react}$, in cm. x is the position of the robot front along the direction of travel from the marked start line, increasing forward; x_{react} is its position at frame n_{react}; x_{board} is the near face of the board on the same axis. It is the gap between robot and board when the robot first visibly reacts (≥ 0 while the board is ahead); in Exp. 4 it equals $d_{ins} - s_{det}$	—	4, 6
Clearance at standstill	$x_{board} - x_{stop}$, in cm (> 0: no contact)	(17)	4, 6
Success rate	Trials with stop, clearing turn and no contact / N × 100, in %	(10)	6

B. Experiment 1: Ultrasonic Distance Measurement

Objective: determine the measurement error. *Variables:* reference distance (proposed 5, 10, 20, 30, 50, 100, 150 and 200 cm). *Procedure:* with the robot stationary (motors disabled or wheels lifted), place a board perpendicular to the sensor axis at each distance and record 10 readings. *Record:* actual distance, mean measured distance, absolute error (5), percentage error (6) and MAE (7). *Interpretation:* an error that grows with distance suggests a temperature effect, since a temperature change of 10 °C alters *v* by about 1.8 % through (2), whereas a constant offset suggests a mounting or timing effect. Results are to be compared with the 0.3 cm resolution in [10] and with the stopping-distance errors in [4].

C. Experiment 2: Obstacle Material

Objective: compare detection by material. *Variables:* material (cardboard, wood, plastic, metal, glass, a fabric-covered surface and an optional additional material). The distance is fixed at a proposed 30 cm and is identical for all materials, as are target size, sensor orientation and environmental conditions. *Procedure:* 10 readings per material at perpendicular incidence;

a reading counts as detected when the sensor returns a valid echo. *Record:* measured distance, detection (yes/no) and absolute error. *Interpretation:* no outcome is assumed; the manufacturer notes only that acoustically soft materials such as cloth can be difficult to detect [10].

D. Experiment 3: Obstacle Angle

Objective: evaluate orientation effects. *Variables:* angle θ of the board relative to the sensor axis: 0°, 15°, 30°, 45° and 60°, with a flat cardboard target at a proposed reference distance of 30 cm. *Procedure:* rotate the board with a protractor about the point where the sensor axis meets its front face, so that the distance along the axis stays constant, and record 10 readings per angle. *Record:* measured distance, detection and absolute error. *Interpretation:* missing or erratic readings at larger angles would indicate that the echo is deflected away from the receiver [2], [3]; this is to be confirmed by measurement, and no result is assumed.

E. Experiment 4: Sudden Obstacle Appearance

Objective: evaluate the response when an obstacle enters the path. *Variables:* insertion distance $d_{ins} = x_{board} - x_0$, proposed as 30 cm; it must exceed the actual threshold so that the detection event is observable, and is to be adjusted if it does not. *Procedure:* with the robot moving forward at its normal speed, place a board in its path at the insertion distance; 10 trials, filmed at 60 frames/s or higher. *Record:* from the video, n_0, n_{react} and n_{stop}, and from the tape x_0, x_{react}, x_{stop} and x_{board}; then the obstacle-to-reaction delay t_{det} (11), the stopping time t_{stop} (12), the total response time t_r (13), the distances s_{det} (14), s (15) and s_{tot} (16), the clearance g (17) and whether a collision occurred. *Interpretation:* the three times are reported separately because they have different causes: t_{det} depends on the measurement cycle, the program and the driver, whereas t_{stop} depends on speed, battery state and floor friction. Since $g = d_{ins} - s_{tot}$, the travel after the obstacle appears shows whether the threshold leaves a safe margin.

F. Experiment 5: Repeatability

Objective: quantify reading variation. *Procedure:* at three reference distances (20, 50 and 100 cm) take 30 readings with the robot and board stationary. *Record:* the mean, the standard deviation σ (8), the minimum and the maximum. *Interpretation:* a small σ relative to the mean indicates stable readings; outliers are reported, not removed [13].

G. Experiment 6: Collision Avoidance Performance

Objective: evaluate stopping and turning over 20 controlled trials. *Procedure:* place a flat board of fixed size and material across the robot's path, perpendicular to its heading, at an initial distance larger than the actual threshold; start the robot from rest at the same marked position in every trial; film each trial; and reset the robot and check the battery before the next. *Record:* the initial obstacle distance, the reaction distance $x_{board} - x_{react}$, the stopping distance s (15), the clearance g (17), turning success (the turn clears the board's line without contact), collision (any contact) and the trial outcome. A trial is successful when the robot stops and turns without collision, and the success rate is SAR (10) with $N = 20$. The actual turning direction and duration should be noted from the video and not assumed.

XI. RESULTS AND DISCUSSION

Status. Quantitative evaluation is pending. No experiment of Section X has been completed, so no accuracy, detection rate, response time, stopping distance or success rate is reported for the robot and no conclusion about its performance is drawn. The authors report that the prototype performs the working sequence of Section IV in functional operation; this observation is qualitative, is not supported by recorded measurements, and is therefore not used to claim any performance. The remainder of this section sets out how each result is to be analysed once the data are collected. Statements about results must name the experiment and the number of trials supporting them, and must not generalise beyond the tested conditions.

Distance measurement (Experiment 1). Plot the measured against the actual distance and report MAE (7) and the percentage error (6) at each distance. An error that grows in proportion to distance points to a wrong speed of sound; recomputing with the recorded temperature through (2) separates this from a fixed offset. Errors should be read against the 0.3 cm resolution in [10], not against the datasheet range alone.

Material (Experiment 2). Compare DR (9) and the absolute error across materials with the same number of readings, distance and orientation. A material with a lower DR or a larger error than the reference board indicates a surface that the threshold rule cannot be trusted with at that distance.

Angle (Experiment 3). Report the largest angle at which detection remains complete, if any, and the angle at which readings first fail or become erratic. If the board is detected at 0° but not at larger angles, the fixed threshold would let the robot approach inclined obstacles unnoticed; this can be stated only if it is measured.

Sudden obstacle (Experiment 4). Summarise t_{det}, t_{stop} and t_r (11)–(13) separately, and s_{det}, s and s_{tot} (14)–(16) against the insertion distance. If s_{tot} is a large fraction of the insertion distance, the margin is thin; if $g \leq 0$ (17) in any trial, the threshold or the speed is inadequate for the tested insertion distance.

Repeatability (Experiment 5). Report σ and the range at each distance and list outliers explicitly [13]. A σ that grows with distance, or isolated spurious values, would indicate noise or interference and may call for filtering, such as a median of several readings, which is not part of the present design.

Collision avoidance (Experiment 6). Report SAR (10) with the collision count, the mean and spread of the reaction and stopping distances, and a note of failures (no stop, no clearing turn). With 20 trials a single failure changes the rate by 5 percentage points, so the rate should be stated with N and a confidence interval for a proportion (for example, a Clopper–Pearson interval), and not as a general reliability figure.

XII. LIMITATIONS

The following are limitations of the approach in general; they are not claimed to have been observed experimentally unless Section XI reports it.

- Dependence on ultrasonic reflections: soft or irregular surfaces may return weak echoes [3], [10].
- Possible measurement variation with temperature and noise, since the speed of sound depends on air temperature (2).

- Narrow effective sensor angle (below 15° [10]); obstacles outside the beam are not sensed.
- Angled or irregular surfaces may deflect the echo away from the receiver [2].
- Environmental interference from other ultrasonic sources or air movement.
- Single-sensor coverage: only the forward direction is sensed, so side and rear obstacles are invisible.
- Dependence on a fixed programmed threshold, which does not adapt to speed, battery state or surface.
- Limited avoidance decisions (stop and turn) and open-loop turning with no feedback on the heading achieved.
- No sensor fusion, so a missed or false reading cannot be cross-checked.
- Undocumented source code, wiring, threshold and hardware ratings limit exact reproduction until they are recorded.

XIII. CONCLUSION

This paper described an Arduino-based mobile robot that uses an HC-SR04 sensor, an L293D driver and two DC geared motors to detect obstacles by a distance threshold and to avoid collisions by stopping, turning and resuming movement. It documented the platform with manufacturer-documented component limits, distinguished the L293D logic and motor supplies, and proposed a reference wiring and control sequence that are kept separate from the prototype records. Because the title promises a framework and not a measured study, its contribution is a design description together with a six-experiment evaluation framework covering distance error, material, angle, sudden obstacles, repeatability and avoidance performance, which is intended to help judge a low-cost platform. Applications such as educational robotics and simple indoor mobile platforms are plausible only after validation. Because the experiments have not yet been performed, no performance is claimed. The limitations of single-sensor sensing and the undocumented prototype details identified in Sections VI–IX remain to be addressed.

XIV. FUTURE SCOPE

The immediate next steps are to record the source code, wiring, threshold and hardware ratings and to carry out the six experiments of Section X. The following further improvements are possible and are not implemented: multiple ultrasonic sensors for wider coverage; adaptive threshold selection; more advanced navigation algorithms; improved collision avoidance; sensor fusion; real-time monitoring; testing in more complex environments; and improved energy efficiency.

ACKNOWLEDGMENT

The authors thank the Department of Information Science and Engineering, East West Institute of Technology, Bengaluru, for the facilities and support provided. The illustration in Fig. 3 was generated with an AI image-generation tool; it is illustrative only and is neither experimental evidence nor a photograph of the prototype.

REFERENCES

[1] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd ed. Cambridge, MA, USA: MIT Press, 2011.

[2] J. Borenstein and Y. Koren, "Obstacle avoidance with ultrasonic sensors," *IEEE J. Robot. Autom.*, vol. 4, no. 2, pp. 213–218, Apr. 1988, doi: 10.1109/56.2085.

[3] L. Kleeman and R. Kuc, "Sonar sensing," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Berlin, Germany: Springer, 2008, pp. 491–519.

[4] H. Sukri, D. T. Laksono, D. T. Laksono, and M. Ulum, "Ultrasonic signal implementation in Arduino-based obstacle robot control system," in *Proc. Int. Conf. Sci. Technol. (ICST 2021), E3S Web Conf.*, vol. 328, 2021, Art. no. 02008, doi: 10.1051/e3sconf/202132802008.

[5] A. Varma, Ashwath A., A. Verma, A. Bagubali, and K. V. Krishnan, "Implementation of obstruction avoiding robot using ultrasonic sensor and Arduino UNO," *Int. J. Recent Technol. Eng.*, vol. 8, no. 4, pp. 7335–7339, Nov. 2019, doi: 10.35940/ijrte.D5295.118419.

[6] H. Sharma, S. Sharma, H. Kumar, M. Yadav, K. Singh, and F. Moreira, "Object sensing robot and distance measurement using Arduino UNO," in *Information Technology and Systems (ICITS 2025)*, Vol. 2, A. Rocha, C. Ferrás, and H. Calvo, Eds., Lecture Notes in Networks and Systems, vol. 1448. Cham, Switzerland: Springer, 2025, pp. 39–48, doi: 10.1007/978-3-031-93106-2_4.

[7] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1986, doi: 10.1177/027836498600500106.

[8] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, Jun. 1989, doi: 10.1109/2.30720.

[9] J. Borenstein and Y. Koren, "The vector field histogram—Fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 7, no. 3, pp. 278–288, Jun. 1991, doi: 10.1109/70.88137.

[10] Cytron Technologies Sdn. Bhd., "Product user's manual – HC-SR04 ultrasonic sensor," Ver. 1.0, May 2013. [Online]. Available: https://web.eece.maine.edu/~zhu/book/lab/HC-SR04%20User%20Manual.pdf (hosted copy). Accessed: Oct. 31, 2026.

[11] Arduino, "Arduino Uno Rev3," Arduino Documentation. [Online]. Available: https://docs.arduino.cc/hardware/uno-rev3. Accessed: Oct. 31, 2026.

[12] Texas Instruments, "L293x quadruple half-H drivers," Datasheet SLRS008D, Sep. 1986 (rev. Jan. 2016). [Online]. Available: https://www.ti.com/product/L293D/datasheet. Accessed: Oct. 31, 2026.

[13] Joint Committee for Guides in Metrology, "Evaluation of measurement data—Guide to the expression of uncertainty in measurement," JCGM 100:2008, 2008.