Cognitive Robotics Lab
Lab 4 - Obstacle Avoidance Using Smart Car Kit

Srikar Ganesh 22BAI1080

25-08-2025

Aim:

To design and implement an Arduino-based smart car system capable of autonomously detecting and avoiding obstacles in its path using sensors, thereby enabling smooth navigation without human intervention and also implement a line following smart car

Components:

1. **Arduino Uno (or compatible board)** – acts as the main microcontroller to control the smart car
2. **Smart Car Chassis Kit** – includes the frame, wheels, and motor mounts
3. **DC Motors (typically 2 or 4)** – to drive the wheels for movement
4. **Motor Driver Module (L298N or L293D)** – controls the direction and speed of the motors
5. **Battery Pack (e.g., 4x AA or Li-ion)** – provides power to the motors
6. **Jumper Wires** – for making electrical connections between components and the Arduino
7. **USB Cable (Type A to B)** – to upload code from PC to Arduino
8. **Wheels (2 or 4)** – attached to the motors to enable movement
9. **Caster Wheel** – provides balance and smooth turning for 3-wheel cars
10. **Breadboard (optional)** – for quick prototyping of connections
11. **Switch (optional)** – to turn the power supply on/off
12. **Ultrasonic Sensors**
13. **LEDs (optional)** – for basic status indicators

OBSTACLE AVOIDANCE
Output:

Source Code:

```
#include <Servo.h>

// --- Pin Definitions ---
#define ENA 5
#define ENB 6
#define IN1 7
#define IN2 8
#define IN3 9
#define IN4 11

#define TRIG_PIN A5
#define ECHO_PIN A4
#define SERVO_PIN 3

// --- Constants ---
```

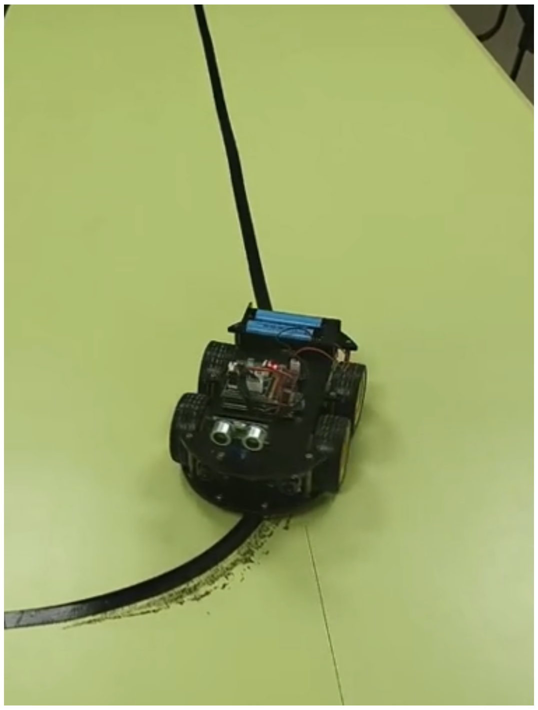
```
#define SAFE_DISTANCE 25   // cm - safety margin front
#define SIDE_MIN 18        // cm - keep away from side walls
#define SCAN_LEFT 150
#define SCAN_RIGHT 30
#define SCAN_CENTER 90

Servo ultrasonicServo;

void setup() {
  Serial.begin(9600);

  pinMode(ENA, OUTPUT);
  pinMode(ENB, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);

  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);

  ultrasonicServo.attach(SERVO_PIN);
  ultrasonicServo.write(SCAN_CENTER);
  delay(500);
}

void loop() {
  long forwardDist = getDistanceAt(SCAN_CENTER);

  if (forwardDist <= SAFE_DISTANCE && forwardDist > 0) {

    stopMotors();
    Serial.println("Obstacle ahead!");

    moveBackward(140);
    delay(250);
    stopMotors();
    delay(150);

    // Scan left and right
    long leftDist = getDistanceAt(SCAN_LEFT);
    delay(150);
    long rightDist = getDistanceAt(SCAN_RIGHT);
    delay(150);

    if (leftDist > rightDist && leftDist > SAFE_DISTANCE) {
      turnLeft(160);
      delay(400);
    } else if (rightDist > SAFE_DISTANCE) {
      turnRight(160);
      delay(400);
    } else {
      // Both sides blocked → reverse
      moveBackward(140);
      delay(600);
    }
    stopMotors();
    delay(150);
  }
  else {
    long leftDist = getDistanceAt(SCAN_LEFT);
    long rightDist = getDistanceAt(SCAN_RIGHT);

    if (leftDist < SIDE_MIN) {
      // Too close to left wall → steer right slightly
      turnRight(160);
      delay(150);
    }
    else if (rightDist < SIDE_MIN) {
      // Too close to right wall → steer left slightly
      turnLeft(160);
      delay(150);
    }
    else {
      // Both sides okay → go straight
      moveForward(140);
      delay(300);   // small cautious step
    }

    stopMotors();
    delay(100);   // rescan often
  }
}

// --- Motor functions ---
void setMotorSpeed(int speedA, int speedB) {
  analogWrite(ENA, speedA);
  analogWrite(ENB, speedB);
}

void moveBackward(int speed) {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
  setMotorSpeed(speed, speed);
}

void moveForward(int speed) {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  setMotorSpeed(speed, speed);
}

void turnRight(int speed) {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  setMotorSpeed(speed, speed);
}

void turnLeft(int speed) {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
  setMotorSpeed(speed, speed);
}

void stopMotors() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
  setMotorSpeed(0, 0);
}

// --- Distance functions ---
long getDistanceAt(int angle) {
  ultrasonicServo.write(angle);
  delay(200);
  return getDistance();
}

long getDistance() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  long duration = pulseIn(ECHO_PIN, HIGH, 20000);
  long distance = duration / 2 / 29.1;
  return distance;
}
```

Explanation:

The Arduino-based obstacle avoiding car uses an ultrasonic sensor mounted on a servo motor to continuously scan its surroundings. The car measures distances in front, left, and right directions to detect obstacles or nearby walls. Based on these readings, the Arduino controls the motor driver to move forward, turn left/right, or reverse. This ensures that the car can navigate safely, avoid collisions, and adjust its path when it gets too close to side walls.

LINE FOLLOWING

OUTPUT 2



SOURCE CODE 2

```
#include <Servo.h>

// Motor driver pins
#define ENA 5   // Speed control for Motor A (Left)
#define ENB 6   // Speed control for Motor B (Right)
#define IN1 7   // Motor A direction
#define IN2 8
#define IN3 9   // Motor B direction
#define IN4 11

// Line sensor pins
#define LEFT 2
#define MID 4
#define RIGHT 10

void setup() {
  pinMode(ENA, OUTPUT);
  pinMode(ENB, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);

  pinMode(LEFT, INPUT);
  pinMode(MID, INPUT);
  pinMode(RIGHT, INPUT);
}


void loop() {
  int leftVal = digitalRead(LEFT);
  int midVal = digitalRead(MID);
  int rightVal = digitalRead(RIGHT);

  if (midVal == LOW) {          // Middle on black → go straight
    forward();
  }
  else if (leftVal == LOW) {    // Left on black → turn left
    left();
  }
  else if (rightVal == LOW) {   // Right on black → turn right
    right();
  }
  else {                        // No sensor sees black → stop
    stopMotors();
  }
}

// ----- Motor functions -----
void forward() {
  analogWrite(ENA, 120);
  analogWrite(ENB, 120);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}

void left() {
  analogWrite(ENA, 200);
  analogWrite(ENB, 200);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}

void right() {
  analogWrite(ENA, 200);
  analogWrite(ENB, 200);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}

void stopMotors() {
  analogWrite(ENA, 0);
  analogWrite(ENB, 0);
}
```

Explanation

In the line following mode, three infrared (IR) sensors (left, middle, right) detect a black strip on the ground. The car adjusts its direction based on sensor feedback: moving straight when the middle sensor is on the line, turning left or right when side sensors detect the line, and stopping when no line is detected.

Result:

The smart car successfully followed the black line when operated in line following mode, accurately turning left and right as required. In obstacle avoidance mode, the car detected obstacles in its path, turned away, or reversed to prevent collisions while also maintaining safe margins from side walls. The combined implementation proved effective in demonstrating autonomous robotic navigation, showcasing both guided movement (line following) and self-driven navigation (obstacle avoidance).