Autonomous Navigation Robot with Intelligent Obstacle Avoidance

Dr. KAYALVIZHI.M [1], DURAIRAJ.N [2], PRAVEEN.M. S [3], SOMASUNDARAM.V [4]
[1]*Associate Professor, Department of Electrical and Electronics Engineering, Asian College of Engineering and Technology, Coimbatore - 641110*
[2,3] *Assistant Professor, Department of Mechanical Engineering, Asian College of Engineering and Technology, Coimbatore - 641110*
[4]*Student, Department of Electrical and Electronics Engineering, Asian College of Engineering and Technology, Coimbatore - 641110*

***Abstract*—This project focuses on the design and implementation of an obstacle-avoiding autonomous robot capable of navigating its environment without human intervention. Obstacle avoidance is a primary requirement of any autonomous robot. The robot uses ultrasonic sensors to detect obstacles in its path and employs a microcontroller-based decision-making system to determine an alternative route. The integration of real-time sensing and control algorithms allows the robot to respond dynamically to changes in its surroundings, or stop to prevent collisions thereby enhancing its ability to move safely and efficiently. By mimicking human reflexes in avoiding barriers, obstacle avoiding robots enhance the efficiency and safety of robotic systems in dynamic or unknown environments. In today's world of advancing technology, robotics plays an important role in making tasks easier and more efficient. An obstacle-avoiding robot is a simple but powerful example of how machines can be programmed to move independently while avoiding collisions with objects in their path. An intelligent obstacle avoidance robot is designed to navigate autonomously in unknown environment by detecting and avoiding obstacles in its path and resumes its running method to reach a destination without collisions. This method is applied in floor-cleaning robots for efficient navigation in long hallways. This capability is crucial for autonomous robots operating in dynamic or unknown environments. The project demonstrates the practicality and effectiveness of basic autonomous navigation using low-cost components.**

***Index Terms*—Obstacle avoiding robot senses**

I. INTRODUCTION

An obstacle-avoiding autonomous robot is designed to navigate its environment safely by detecting and avoiding collisions with obstacles, often using sensors like ultrasonic sensors or infrared sensors. These robots are programmed to intelligently sense obstacles in their path, react by maneuvering around them, and continue their movement, typically following a predetermined path or goal.
Key aspects of an obstacle-avoiding autonomous robot:
Sensing: Using various sensors to detect the presence and distance of obstacles.
Decision-making: Interpreting sensor data and deciding on the appropriate course of action to avoid obstacles.
Movement: Implementing the chosen maneuver to navigate around the obstacle.
Autonomy: Operating independently, without external control, based on its internal programming.
Applications: Used in various fields like industrial automation, healthcare, and research.

II. MECHANICAL DESIGN OF ROBOT

This includes the hardware design of the robot that is motor & wheel placement body setup. Robot uses two Robotics gear motor & wheel for the movement, which will help it to move forward, left or right. Robot uses two motor & wheel in the back side and one freewheeling ball is placed at the front which helps it to free movement. The sensor is placed in such a way that they can cover the maximum area in front of the robot and can be capable to detect an obstacle either obstacle small or big.

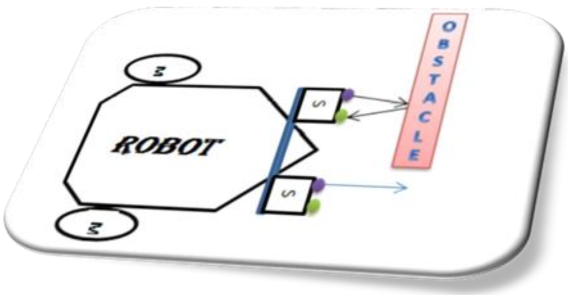

Fig 2.1 Drawing of the robot

III. STRUCTURES OF ROBOT

3.1 CIRCUIT DESIGN OF ROBOT:

Circuit design mainly consists of two parts

- Sensor part
- Control board part

Sensor part: The sensors used in this robot are Infrared sensor, consisting two-part infrared signal generator and the IR receiver designed in single PCB. There are two sensors are used as left side sensor and right-side sensor and two sensors are used to sense the obstacle on left and right side.

IR Generator: -This is a Monostable multivibrator using NE555 IC generating Infrared Signal of 38KHz frequency for better determination of the object. By using a variable resistance, we can adjust the frequency of the IR signal detector TSOP1738 gives a high output.

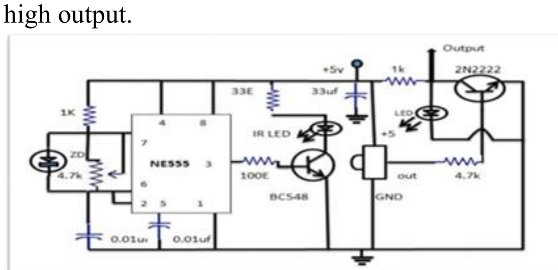

Fig 3.1 Diagram of Sensor circuit

3.2 CONTROL BOARD:

There are two sensor S1and S2 placed at the left and right side of the Robot to sense the obstacle. These sensors may be infrared sensors or ultrasonic sensor depending upon the application. Sensors sense the object then generate a signal high or low then signal is processed by the microcontroller AT80C2051

For an obstacle-avoiding robot the control board serves as its central nervous system orchestrating the actions of its various components to navigate autonomously. Typically a microcontroller like an Arduino Uno, ESP32, or Raspberry Pi Pico forms the brain of the system. This microcontroller receives input signals from the robot's sensors such as ultrasonic sensors or infrared (IR) sensors which detect the presence and distance of obstacles. Based on the data received from these sensors the microcontroller executes a programmed algorithm.

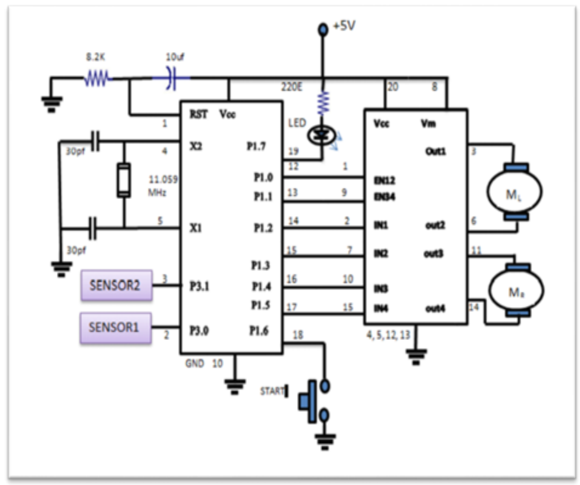

Fig 3.2 Circuit diagram of the control board

IV. WORKING PRINCIPLE

The robot in this project detects obstacles with the help of three ultrasonic distance sensors to measures the distance to surrounding objects. Although the project is started with a single ultrasonic sensor, two more sensors is added since the robot had blind spots in its right and left direction for which it was having collision while maneuvering. Unlike the projects discussed above our project concentrates on coordinating multiple ultrasonic sensors for maneuvering without collision and also maintaining a minimum travel distance. Fig. 2 describes this algorithm in a flow chart.

The robot was designed to detect the presence of any object within the specified threshold distance. If any object is found within this distance, it is designated as an obstacle and the robot will turn away from it. The three ultrasonic sensors are placed in the frontal section of the robot at the right, middle and left position. The three sensors emit an ultrasonic pulse every 300 millisecond which echoes from the neighbouring objects. Using time difference between the input and echo, the Arduino calculates the distance to the obstacle from which the echo is coming by using the constant speed of sound 340 m/s. When one of the sensors detects obstacle within the threshold distance, the robot changes its direction.

Along with these basic movements, the robot is designed to handle a more complex situation when all three sensors have obstacles within the specified range. In this case, the robot will move backward for 10 ms and again check the distance to objects with the help of right and left sensors.

The robot will then compare the two distances and move in the direction where the distance is larger.

The working principle of an obstacle-avoiding robot revolves around a continuous cycle of sensing, processing, and acting to navigate its environment without colliding with obstacles.

Sensing the Environment the robot utilizes various sensors to perceive its surroundings. In an obstacle-avoiding autonomous robot**,** several common sensors are used to detect and avoid obstacles.

Ultrasonic Sensors these emit high-frequency sound waves and measure the time it takes for the echoes to return. This time is used to calculate the distance to objects in front of the sensor.

Infrared (IR) Sensors: These sensors typically consist of an IR emitter and an IR receiver. The emitter sends out an infrared light beam, and the receiver detects the reflected light. The intensity of the reflected light can indicate the presence and proximity of an object particularly those that are reflective to IR.

V. WORKING DIAGRAM

The sensor operates on the echo ranging principle, which is similar to how bats navigate in the dark. It involves the transmission and reception of sound waves.

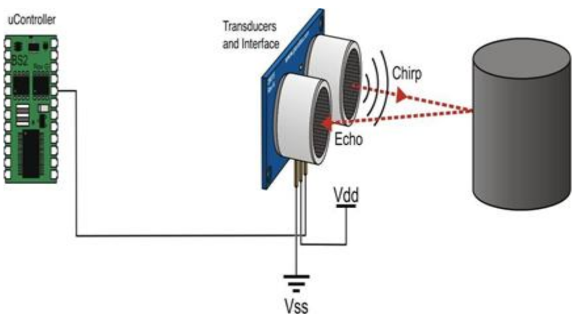

Fig. 5.1 Working diagram of ultrasonic sensor

- Emission: The sensor has a transmitting transducer that emits an ultrasonic pulse (sound wave).
- Reflection: When this pulse hits an object, itreflects back.
- Reception: A receiving transducer detects the reflected sound wave (echo).
- Time Measurement: The time interval between the sending of the pulse and receiving of the echo is measured.
- Distance Calculation: Using the known speed of sound in air (approximately 343 m/s), the distance to the object is calculated.
- DISTANCE = (SPEED OF SOUND * TIME) / 2.

5.1 H BRIDGE MOTOR DRIVER:

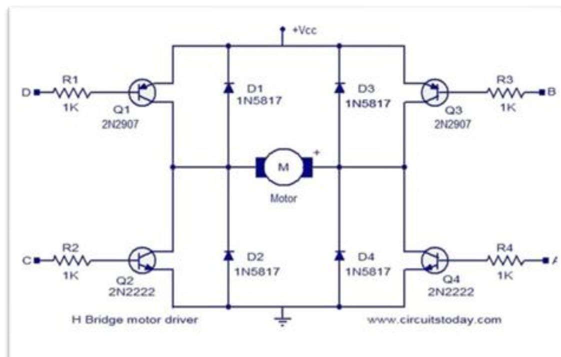

Fig5.2 Circuit diagram of H bridge

MOTOR DRIVER

The circuit given here is of a simple H bridge motor driver circuit using easily available components. H Bridge is a very effective method for driving motors and it finds a lot of applications in many electronic projects especially in robotics. The circuit shown here is a typical four transistor H Bridge. The diodes D1 to D4 provide a safer path for the back emf from the motor to dissipate and thus it protects the corresponding bipolar transistors from damage. Resistors R1 to R4 limit the base current of the corresponding transistors. When terminal D is grounded and A is pulled to +Vcc, transistors Q1 and Q4 will be on and current passes through the motor from left to right. When terminal B is grounded and C is pulled to +Vcc, transistors Q3 and Q2 will be on and current passes through the motor from right to right making the motor to rotate in the opposite direction.

5.1 OTHER COMPONETS

- H-BRIDGE CONFIGURATION
- DUAL CHANNELS
- BO MOTOR
- SERVOMOTOR
- SOLAR PANEL:

VI. GRID MAP AND ADDITIONAL COMPONENTS

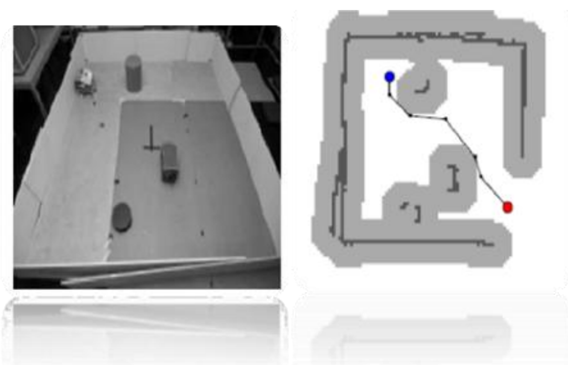

Fig.6.1 Obstacle Avoidance Setup and the Resultant Grid Map with the Collision Free Path

The robot developed in this project uses ultrasonic sensors to detect obstacles in real time and requires no path planning. Its processing unit is based on the Arduino platform.

The Autonomous Surface Vehicle (ASV) developed by Henderson and Sukhumi (2011), employed a single-beam mechanically-scanning profiling sonar to detect obstacles under water. The profiling sonar has the ability to produce cone- shaped beam which is ideal for detecting near surface obstacles. One of the objectives of their work was to investigate the suitability of using sonar near the water air boundary for which the study found promising results.

Although similar detection technology is used, our robot is designed to navigate on the ground and identify obstacles above the surface. It is using the Arduino software which enables to upload a code written in C programming language.

There were other works using multiple sensors to make the robot more accustomed to its surroundings by employing both range and appearance-based obstacle detection (Shah dib, Ullah, Hasan, & Mahmud, 2013; Grey, 2000).

Their obstacle detection also includes a combination of global and local avoidance. In one of these projects, Shah dib, Ullah, Hasan and Mahmud (2013) fused the strengths of an image and an ultrasonic sensor to detect objects and measure its size.

A grid map is a common way to represent a robot's environment for obstacle avoidance. It divides the space into a grid of cells, where each cell can be marked as either occupied (obstacle) or free (safe to navigate). Obstacle avoidance algorithms then use this grid map to plan paths that avoid the occupied cells.

VII. IMPLEMENTATION

PROGRAM CODE:

```
#include <Servo.h> Servo Myservo;
#define trigPin 9          // Trig Pin Of HC-SR04
#define echoPin 8 // Echo Pin Of HC-SR04
#define MLa 4          //left motor 1st pin
#define MLb 5  //left motor 2nd pin
#define MRa 6  //right motor 1st pin
#define MRb 7 //right motor 2nd pin long
duration, distance;
void setup() { Serial.begin(9600);
pinMode(MLa, OUTPUT);     // Set Motor Pins
As O/P pinMode(MLb, OUTPUT);
pinMode(MRa, OUTPUT); pinMode(MRb,
OUTPUT);
pinMode(trigPin, OUTPUT);         // Set
Trig Pin As O/P To Transmit Waves
pinMode(echoPin, INPUT);     //Set Echo Pin
As I/P To Receive Reflected Waves
Myservo.attach(10);
}
void loop()
{
Serial.begin(9600); digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);     //    Transmit
Waves For 10us delayMicroseconds(10);

duration = pulseIn(echoPin, HIGH);
     // Receive Reflected Waves distance =
```

```
duration / 58.2; // Get Distance
Serial.println(distance);
delay(10);
if (distance > 15) // Condition For Absence Of Obstacle
{
Myservo.write(90);
digitalWrite(MRb, HIGH); // Move Forward
digitalWrite(MRa, LOW);
digitalWrite(MLb, HIGH); digitalWrite(MLa, LOW);
}
else if ((distance < 10)&&(distance > 0)) // Condition For Presence Of Obstacle
{
digitalWrite(MRb, LOW); //Stop
digitalWrite(MRa, LOW); digitalWrite(MLb, LOW); digitalWrite(MLa, LOW); delay(100);
Myservo.write(0); delay(500);
Myservo.write(180); delay(500);
Myservo.write(90); delay(500);
digitalWrite(MRb, LOW); // Move Backward digitalWrite(MRa, HIGH);
digitalWrite(MLb, LOW); digitalWrite(MLa, HIGH); delay(500);
digitalWrite(MRb, LOW); //Stop
digitalWrite(MRa, LOW); digitalWrite(MLb, LOW); digitalWrite(MLa, LOW); delay(100);
digitalWrite(MRb, HIGH); // Move Left digitalWrite(MRa, LOW); digitalWrite(MLa, LOW); digitalWrite(MLb, LOW);
delay(500);
}
}
```

VIII. ALGORITHMS AND ARCHITECTURE

8.1 ALGORITHMS

Start

Step:1 Check either switch (p1.6) is on or off.

Step:2 If on then go to next step4, otherwise rotate at the same step.

Step:3 Initialize the input port (P3) & output port(P1).

Step:4 Set the bit of port pin 1.0 and pin1.1.

Step:5 Read data from port 3.

Step:6 Check the bit on p3.0

Step:7 If bit is present move left motors in Forward direction and stop the right motor, else go to next step9

Step:8 Check the bit in p3.2

Step:9 If bit is present on pin p3.2, then move right motor in forward direction until we get high signal on pin p3.2&stop left motor.

Step:10 Again go to step 6

Sensor Detection:

Sensors: Robots use various sensors to detect obstacles, such as ultrasonic sensors, infrared (IR) sensors, or cameras.

Distance Measurement: Sensors measure the distance to nearby objects. For example, ultrasonic sensors emit sound waves and calculate the distance based on the time it takes for the echoes to return.

Thresholds: The robot uses pre-defined thresholds to determine if an object is considered an obstacle.

8.2 FLOWCHART:

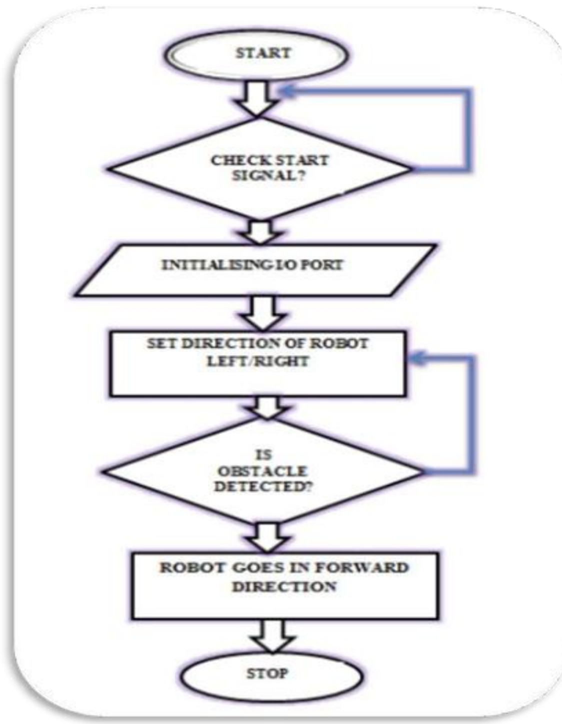

Fig 8.2 Flowchart for the software

8.3 ROBOT ARCHITECTURE:

The Arduino Platform:

There are numerous hardware platforms are used as the foundation for building obstacle-avoiding and other types of mobile robots. We have selected the Arduino board as the microcontroller platform and its software counterpart to carry out the programming. Arduino is an open-source platform which is an integration of hardware (microcontroller) and software components. The microcontroller can read input in the form of light or sound through a sensor and convert it into an output (e.g., driving a motor) according to the instruction given by the Arduino programming (Arduino, 2015). The Arduino microcontroller can only be functional with the help of a code. To write this code Arduino Integrated Development Environment or Arduino Soft- ware (IDE) is used which is also open source like the Arduino Uno board (Arduino, 2015). It is much popular software used by many for its simplicity and the ability to communicate with all Arduino boards.

Arduino Software version 1.6.5 is used to write the code in C programming language which is then uploaded to the Arduino microcontroller through an USB cable. The software saves the code in a file with .in extension. While there are many other microcontroller platforms available, Arduino gained much popularity which attributed to its distinctive features such as Economical. The software is compatible with multiple platforms, including Windows, Linux, and macOS, and features an intuitive programming environment.

Both software and hardware are open source and can be customized to meet specific needs in this project, the Arduino board will take input from ultrasonic sensor, calculate the distance to the obstacle and control rotation of the servo motor as an output response. Hardware Components and Assembly

The following flowchart in Fig.3 shows the hardware used to build the robot and explains relationship (input and output) among them. The hardware was assembled to form the obstacle avoiding robot in Fig.4 with the help of a chassis, wheels and connecting cables.

An obstacle-avoiding robot's architecture typically involves a sensing system, a processing unit (like a microcontroller), and actuators (like motors) to control movement. The sensing system detects obstacles, the processing unit analyzes the data and makes decisions, and the actuators respond by changing the robot's direction or speed.

ALGORITHM DIAGRAM

An obstacle-avoiding robot can use a straight forward but efficient algorithm that relies on sensors to detect nearby objects and change direction to avoid collisions. This can be achieved through a basic "if-then-else" approach where the robot checks for obstacles and changes its direction accordingly. More advanced algorithms, like the Follow the Gap method, can also be used to calculate the best path around obstacles. A basic algorithm for an obstacle-avoiding robot using ultrasonic sensors is implemented.

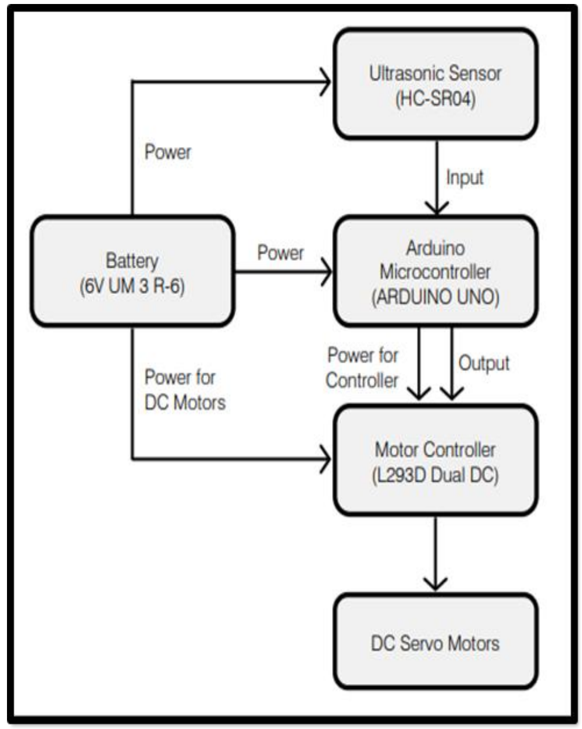

Fig.7.3: Algorithm for Obstacle Avoiding

IX. RESULT AND KIT OVERVIEW

9.1 RESULT:

Distance: 12 cm

Obstacle Detected! Working of robot As the robot is switched on, First, it will check either start signal is received or not, if not then the program counter will not go to the next address it will remains on the same address until it gets a negative signal. Then the robot continuously monitors any obstacle in path, if there is no obstacle then robot will go straight. If any obstacle will find in left side, then the controller sends a command to the motor drive to stop the right motor & move the left motor and just opposite as obstacle found in right side

Initializing...

Distance: 52 cm Moving Forward Distance: 38 cm Moving Forward Distance: 18 cm Obstacle Detected! Stopping

Looking Right: 45 cm Looking Left: 25 cm Turning Right Distance: 60 cm Moving Forward

Stopping
Looking Right: 20 cm Looking Left: 55 cm Turning Left.
9.2 PIN DIAGRAM:

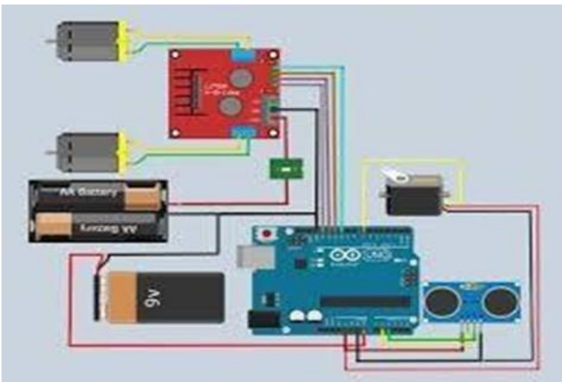

Fig 9.2 Pin diagram

9.3. OUTPUT:

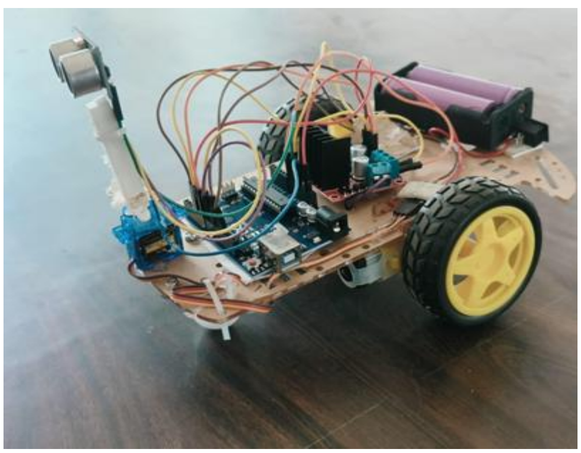

Fig 8.3 OUTPUT OF ROBOT

X. CONCLUSION

The development of an obstacle-avoiding autonomous robot demonstrates the effective integration of sensor technologies, control algorithms, and embedded systems to enable intelligent navigation in dynamic environments. These robots utilize ultrasonic sensor to detect obstacles and autonomously make navigation decisions, promoting safe and efficient movements. The robot successfully detects and avoids obstacles in its path without human intervention, showcasing real-time decision-making capabilities. In conclusion, obstacle-avoiding robots offer a promising solution for autonomous navigation in various environments. This type of robotic system has significant applications in fields such as industrial automation, security, automated transport, surveillance, service robotics and exploration where autonomous operation in unpredictable environments is essential. Future enhancements can include machine learning-based path optimization, improved sensor fusion, and adaptability to complex terrains for wider deployment in real-world scenarios.

REFERENCES

[1] Dawn M. Tilbury: Professor at the University of Michigan, with research interests including robotics and automation. Her work has touched upon shared control methods for obstacle avoidance in mobile robots.

[2] Andre Platzer: Professor at Carnegie Mellon University, known for his work on formal verification of cyber-physical systems, including aspects of autonomous robot navigation and obstacle avoidance.

[3] S. K. Tso: Formerly at the University of Hong Kong, contributed to research on sensor-based fuzzy reactive navigation for mobile robots.

[4] O. Castillo: Researcher in fuzzy logic and its applications in robotics, including the optimization of fuzzy reactive controllers for obstacle avoidance.

[5] Researchers at the GRVC (Grupo de Robótica, Visión y Control) at the University of Seville, Spain: This group has a strong focus on mobile robotics and has published work on reactive navigation and obstacle avoidance.

[6] Brooks, R. A. (1986) A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation, 2(1), 14-23. Introduces the subsumption architecture, a reactive approach to robot controls that heavily influenced early obstacle avoidance strategies.

[7] Latombe, J. C. (1991) Robot motion planning Kluwer Academic Publishers.A comprehensive textbook covering various path planning algorithms, including those that consider obstacles.

[8] Borenstein, J., Everett, H. R., & Feng, L. (1996) Navigating mobile robots' systems and techniques. AK Peters, Ltd. A widely used

textbook providing a broad overview of mobile robot navigation, with significant sections on obstacle avoidance using different sensor modalities and techniques like Vector Field Histogram (VFH).