OBSTACLE AVOIDANCE ROBOT

Minor Project Report

Submitted for the partial fulfillment of the degree of

Bachelor of Technology

In

Internet of Things (IOT)

Submitted By

Ayush Gupta [0901IO221019]

UNDER THE SUPERVISION AND GUIDANCE OF

Dr. Aditya Dubey

Assistant Professor

Department of Center for Internet Of Things

माधव प्रौद्योगिकी एवं विज्ञान संस्थान, ग्वालियर

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

Deemed University

(Declared under Distinct Category by Ministry of Education, Government of India)

NAAC ACCREDITED WITH A++ Grade

Gola Ka Mandir, Gwalior (M.P.)- 474005, INDIA

Ph.:+91-751-2409300, E-mail: vicechancellor@mitsgwalior.in, Website: www.mitsgwalior.in

August – December 2024

DECLARATION BY THE CANDIDATE

We hereby declare that the work entitled **"Obstacle Avoidance Robot"** is my work, conducted under the supervision of, **Dr. Aditya Dubey Assistant Professor,** during the session Aug-Dec 2024. The report submitted by me is a record of Bonafide work carried out by me.

We further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Ayush

Ayush Gupta [0901IO221019]

Date: 19/11/24

Place: Gwalior

This is to certify that the above statement made by the candidates is correct to the best of my knowledge and belief.

Guided By:

Aditya 20/11/24

Dr. Aditya Dubey
Assistant Professor
Center for Artificial Intelligence
MITS, Gwalior

Departmental Project Coordinator

Assistant Professor
Center for Artificial Intelligence
MITS, Gwalior

Approved by HOD

19/11/24

Dr. Praveen Bansal
Associate Professor & Head
Center for Artificial Intelligence
MITS, Gwalior

PLAGIARISM CHECK CERTIFICATE

This is to certify that we, students of B.Tech. in **Internet of Things (IOT)** have checked my complete report entitled **"Obstacle Avoidance Robot"** for similarity/plagiarism using the "Turnitin" software available in the institute.

This is to certify that the similarity in my report is found to be 6 %. which is within the specified limit (30%).

The full plagiarism report along with the summary is enclosed.

Ayush

Ayush Gupta [0901IO221019]

Checked & Approved By:

Soumyajit Ghosh 20/11/24

Dr. Soumyajit Ghosh
Assistant Professor
Center for Artificial Intelligence
MITS, Gwalior

ABSTRACT

This project involves the construction of an Obstacle Avoidance Robot; a fully automated system developed to move while observing the existence of obstacles in a given territory. For distance measurement, the robot employs an HC-SR04 ultrasonic sensor while the microcontroller is an Arduino Uno for control and decision-making. BO geared motors can be driven with speed and direction settings due to the integrated L298N motor driver in the system. It is operated by a 9V battery, the algorithms to nip obstacles and to stop, back up and swerve are gross but adequate.

The possibility of further development can be essentially seen in the versatility of the structure and the mission-achieving algorithm of the obstacle avoiding robot; Therefore, the potential future application of the current obstacle avoiding robot is for creating autonomous vehicles, industrial automation systems and robotic exploration in risky terrains. PWM is implemented to economize power used to control motors, jumper wires and a castor wheel guarantees portability and easy assembling of the circuit.

This project lays the beginnings for a series of more complex projects dealing with robotics and embedded systems showing how with rather inexpensive hardware and simple programming one can achieve autonomous intelligent action. It demonstrates how it empowers using machine learning for real-time decision-making and better perceiving obstacles to scaling and integrating the device.

ACKNOWLEDGEMENT

On behalf of all the authors of this Report, we would like to say a big 'Thank you' to all the people who helped and supported us in preparing this Minor project. We are thankful to the whole Artificial Intelligence department for providing contacts in the first stages, guidance during the course of the experiments and for encouraging until the last days right before the making of the final report of this Mini skill-based project.

With regard to the above-mentioned points, special thanks are due to our colleagues for contributing great ideas when exchanging with an interest in solving problems and experience. We also extend our deep sense of gratitude to our professor Dr. Aditya Dubey for his constant support and interests that were instrumental in steering us towards enacting our dreams, and subsequently, the ability of opting for the course that is most suited to us and without his help, our project would have remained incomplete.

Finally, we would like to show our gratitude to friends who appreciated our work and encouraged us to go on

Ayush Gupta
[0901IO221019]

CONTENT

Table of Contents

ACRONYMS

1. **MCU**: Microcontroller Unit
2. **PWM**: Pulse Width Modulation
3. **IDE**: Integrated Development Environment
4. **I/O**: Input/Output
5. **DC**: Direct Current
6. **LiDAR**: Light Detection and Ranging
7. **GND**: Ground
8. **VCC**: Voltage Common Collector (or Voltage at the Common Collector, indicating the power supply for circuits)
9. **AMR**: Autonomous Mobile Robot

NOMENCLATURE

1. Arduino Uno (MCU): It is responsible for controlling signals from the sensors and motor drivers apart from processing.
2. HC-SR04 (Ultrasonic Sensor): Ultrasonic sensor which is used to calculate distance with objects surrounding it.
3. L298N (Motor Driver): An H-bridge motor driver was employed also in this design to manage the direction as well as the velocity of the two DC motors on the robot design with a comparable pace.
4. PWM (Pulse Width Modulation): A method by which the speed of a motor is regulated by changing the duration of the pulse signal in contact with the motor.
5. GPIO (General Purpose Input/Output): The ports on the Arduino board that one employed when passing signals to or from different devices.
6. VCC (Voltage Common Collector): The pin used for providing power to a component to which usually a positive voltage is connected.
7. GND (Ground): The phase from which voltage is measured and the electrical current return path..
8. Castor Wheel: A spinning wheel with no load and ensured the balancing of the robot.
9. PWM Frequency: The frequency at which the pulse width modulated signal is transmitted so as to regulate the speed of the motor.
10. Echo Pin: The hardware pin that points out that the ultrasonic wave reflected off the object has been detected.
11. Obstacle Avoidance: The process through which the robot actually identifies and avoids any obstacle that is in front of it.
12. Distance Measurement: An elaborate procedure in determining the proximity of the robot from an object by using ultrasonic sensors.
13. Backwards Motion: The actual motion of the robot reversing possibly to assist in recalibration of where it is situated next to Path Planning.
14. Motor Driver Circuit: A circuit that has the ability to control both speed and direction necessary for motion into the required path of the robot.
15. Obstacle Detection Logic: a software-based logic which determines when an obstacle is seen and how the robot has to behave.
16. Proportional Turn: A rotational apparatus that changes the angle and the turning rate depending on a specific signal.
17. Autonomous Navigation: The use of the sensors in enabling the robot to move in its environment through a set of predetermined instructions without assistance from its owner.

List of Figures

LIST OF TABLES

CHAPTER 1: INTRODUCTION

The Obstacle Avoidance Robot is a complete system involved in avoiding obstacles on its path using the advanced ultrasonic sensors. This project achieves the use of mechanics and electronics putting together mechanical components, circuits, and code to illustrate the application of robotics and embedded systems. Successfully self-mobile, the robot stands as one of the essential base models for the application area that involves the use of self-driving technologies such as self-driving cars, self-automated factories, and space explorations.

The fundamentals of this system are the Arduino Uno, an MCU that collects data from the HC-SR04 Ultrasonic Sensor to identify obstacles by computing distances from reflected sound waves. If an obstacle is present within the set distance range, the Arduino analyses this signal and gives control signals to the L298N Motor Driver Board to stop, reverse, and turn the robot from the side to avoid the obstacle. The robot takes a 9V battery for power and it has a movement option of BO Geared Motors whose wheel diameter is 65mm. A castor wheel is also provided for stability during movement of the equipment. In the control strategy, PWM is utilized for velocity control to produce efficient energy consumption. The design is fully modular, meaning that the parts are all interconnected using jumper wires to facilitate easy construction.

This project also looks at the working of the ultrasonic sensor which works by emitting ultrasound waves and calculates the time taken for the wave to be reflected. It is converted into distance thus making the robot in a position to make decisions on the time that has been converted.The operations of the obstacle avoidance robot depend on a program that is fixed in advance and uploaded to the Arduino Uno using the Integrated Development Environment (IDE). The code tells the robot to always send ultrasonic waves, receive distances, and perform a number of movements all in relation to the sensors. This project does more than provide the basic understanding of robotics; it prepares the way for adding other aspects like machine learning to provide even better direction and determinations. This robot is a good example of how expensive computational and control systems are not necessary for satisfying and intelligent autonomy provided cost-effective components are designed and programmed effectively.

CHAPTER 2: LITERATURE SURVEY

Obstacle avoidance robots are widely researched due to their real-world applicability in the use of automation and robotics and self-navigation. The existing technologies, methodologies, and applications relevant to the development and nature of such robots are presented in this literature survey.

2.1. Ultrasonic Sensor Technology

For example, small ultrasonic sensors such as the HC-SR04 are preferred in building obstacle avoidance robots because of the parameters of high accuracy, low cost and non-contact range detection. These sensors function through production of ultrasonic waves, the time taken for the wave reflected back to the same transducer to be computed so as to facilitate computation of distances. Studies show that the HC SR04 module has a range of 2 cm to 400 cm and maximum accuracy of +/ - 3mm suggested for short range use only. Such technology is especially suitable in environments which need real-time ability to determine presence of obstacles.

2.2. Microcontroller Based Control System

Many tiny computers like Arduino Uno are used in obstacle avoidance robots for data analysis and motor control. Research has proved the Arduino Uno to be very useful because of its open-source characteristics, easy to code and adequate I/O channels to interface with sensors and motors. This is made even easier by the fact that it supports Integrated Development Environment (IDE).

2.3. Motor Driver Circuits

L298N motor driver board is most commonly used for modulating velocity and direction of DC motors in robotic systems. A search on H-Bridge circuits the L298N uses stresses their versatility in bidirectional motor control. Technique such as Pulse Width Modulation (PWM) is understood as a significant strategy of managing the speed of a motors utilized in shifting the motion of robots depending on the obstacles encountered.

2.4. Applications and Significance

Obstacle avoidance robots are basic models for more complex applications of robotic systems for self-driving cars, robotic manufacturing processes, and robotics used in extreme conditions. Research shows that these robots reduce risks and enhance performance by avoiding contact and mapping the environment on their own.

2.5. Limitations and Future Directions:

Ultrasonic sensors with microcontroller-based systems are also efficient and the research reveals deficiencies in the method of identifying reflective or translucent objects. Future work recommends the use of machine learning with improved obstacle detection elements, including LiDAR or vision-based sensors to provide appropriate decision making.

This survey gives a rather sound ground for the development of the obstacle avoidance robot that is based on the integration of the known technologies and enables defining further possibilities and directions in the given sphere.

Chapter 3: Components Used

Table of Components Used in this Project:

Component Name	Quantity
Arduino Uno	1
L298N Motor Driver Board	1
HC-SR04 Ultrasonic Sensor	3
BO Geared Motor with Wheels	2
Castor Wheels	1
9V Battery	2
On-Off Switch	1
Jumper Wires	15
3D Printed Chassis	1

Table 1. Components Required

3.1. Arduino Uno

The Arduino Uno is a microcontroller board based on the ATmega328P, designed for easy programming and prototyping. It has 14 digital I/O pins (6 for PWM) and 6 analog inputs, enabling the control of motors and sensors. The board is programmable through the Arduino IDE using a USB cable and operates on 5V logic. Its simplicity, affordability, and open-source nature make it ideal for small-scale robotics projects like obstacle avoidance robots. Itprocesses sensor data from the HC-SR04 and provides instructions to the motor driver board, enabling real-time decision-making.

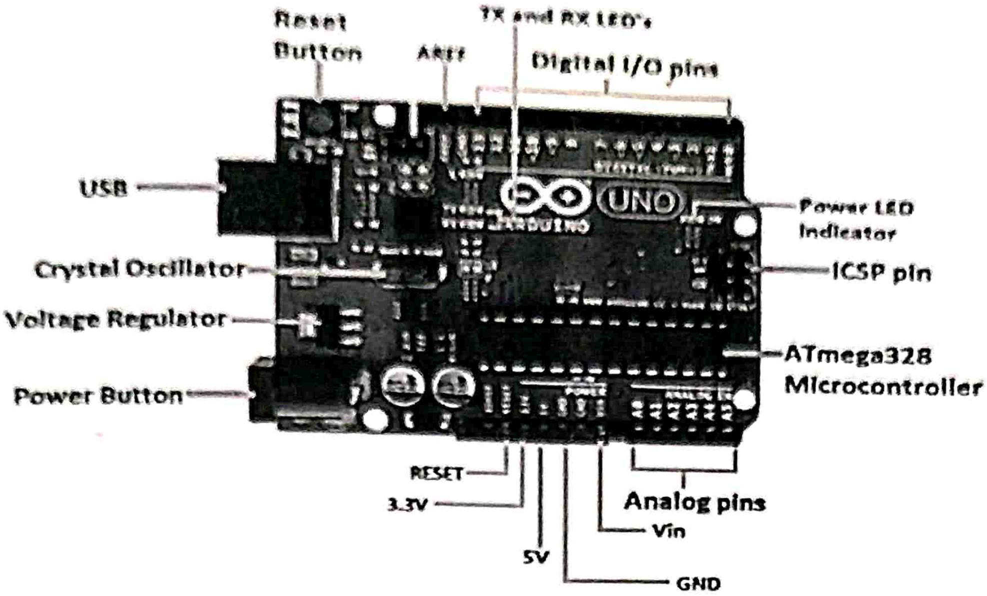

Figure 1. Arduino Uno

3.2. L298N Motor Driver Board

The L298N is a dual H-Bridge motor driver that controls the speed and direction of two DC motors. It supports motor voltages between 5V and 35V and a peak current of 2A per channel. Using PWM signals from the Arduino, the L298N regulates motor speed, while its H-Bridge configuration allows forward and backward motion. This module is compact and features onboard heatsinks for thermal management. Its ability to control two motors independently is critical for navigation and obstacle avoidance in the robot.

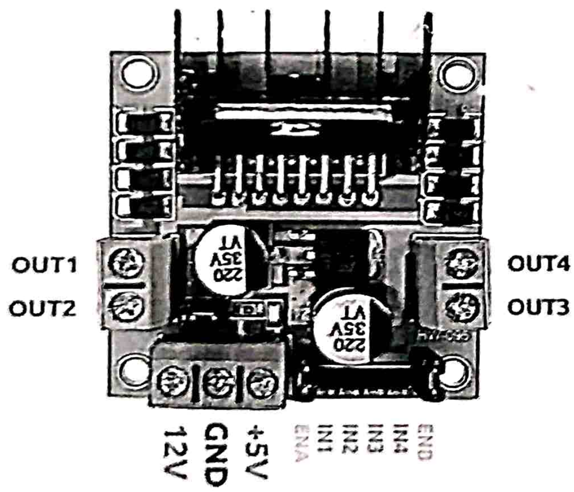

Figure 2. L298 Motor Driver Board

3.3. HC-SR04 Ultrasonic Sensor

The HC-SR04 ultrasonic sensor measures distances ranging from 2 cm to 400 cm with an accuracy of ±3 mm. It emits ultrasonic waves via a transmitter and calculates the time taken for their reflection to return. The Arduino uses this time to compute the distance to obstacles. This sensor is reliable, cost-effective, and ideal for detecting obstacles in dynamic environments. Its data is essential for the robot to avoid collisions by steering clear of detected objects.

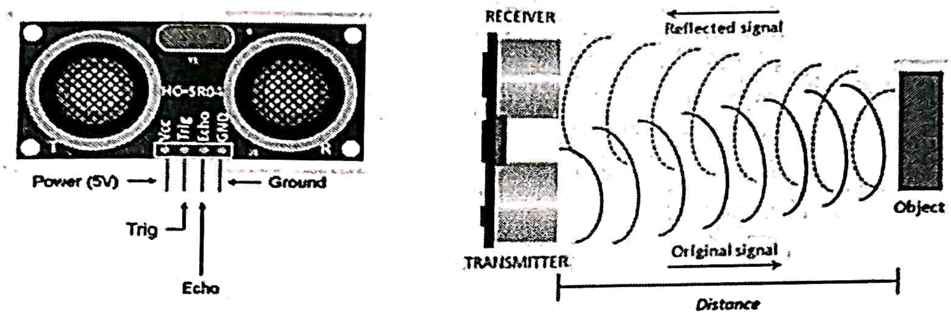

Figure 3. Ultrasonic Sensor with its working

3.4. BO Geared Motor with Wheels

BO geared motors are lightweight DC motors equipped with internal gearboxes for high torque and low speed. They are available in various RPM ratings, making them suitable for precise movement in small robots. These motors are paired with 65mm rubber-grip wheels for traction and stability. The high torque enables the robot to navigate different terrains, while the wheels ensure smooth movement. These motors work in conjunction with the L298N motor driver for directional control.

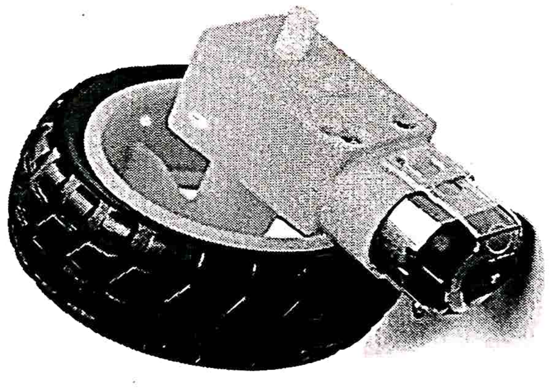

Figure 4. BO Geared Motor Wheels

3.5. Castor Wheel

The castor wheel acts as a support wheel to stabilize the robot's chassis and allow multidirectional movement. Positioned at the rear of the robot, it reduces friction while turning or changing direction. Its low weight and durable build make it suitable for small robots, ensuring smooth operation without interfering with the powered wheels.

3.6. 9V Battery

A 9V battery powers the Arduino Uno and motor driver, ensuring uninterrupted operation of the robot. It provides a convenient and portable power source, with sufficient voltage to support the sensors, motors, and microcontroller. A battery connector simplifies integration, while its compact size allows for easy placement within the robot chassis.

3.7. On-Off Switch

The on-off switch provides a simple way to control the power supply to the robot. It allows users to start or stop the robot without disconnecting the battery. This enhances usability and ensures energy conservation when the robot is not in use.

3.8. Jumper Wires

Jumper wires are used to establish electrical connections between components on a breadboard or directly to the Arduino. They come in male-to-male, female-to-male, and female-to-female configurations, providing flexibility in connecting pins. These wires facilitate quick prototyping without soldering, making them essential for projects like the obstacle avoidance robot.

3.9. 3D Printed Chassis A 3D printed chassis gives a lightweight and rigid structure along with an opportunity to customize it according to need. This section is supposed to embrace all subsystems within the Arduino, motor driver, selected sensors, and a battery. They have also provided some slots and mounts on the basis of their designs to have chassis where the motors, castor wheel and wires can be locked and placed. Applying 3D printing increases accuracy and, if required, plane changes to fit the project requirements can be made quickly. The sleek and well-fit structure improves on the sturdiness and the appearance of the robot.

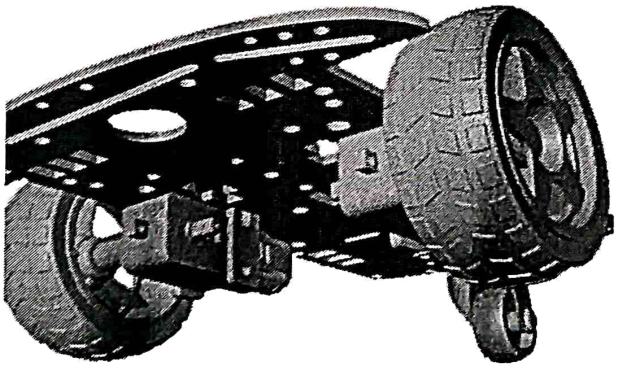

Figure 5. 3D Printed Chassis (using Fusion360)

Chapter 4: Hardware Assembly and Construction

4.1 Assembly of Components for Obstacle Avoidance Robot:

This part should explain how the various parts mentioned earlier are physically incorporated into the context of the robot and therefore it should cover matters concerning motor fixing, the sensor fixing, the fixing of Arduino, fixing of a motor driver and other peripherals on the chassis of the robot. It may contain procedures for orienting parts to function correctly, ways of making the connections safe and proper placing of the robot design layout so as not to experience problems like interferences by parts.

4.2. Arduino Uno Pin Connections:

- **Pin 4 (MLa)**: Connect to **Motor A (Left)** motor driver control pin.
- **Pin 5 (MLb)**: Connect to **Motor B (Left)** motor driver control pin.
- **Pin 6 (MRa)**: Connect to **Motor A (Right)** motor driver control pin.
- **Pin 7 (MRb)**: Connect to **Motor B (Right)** motor driver control pin.d
- **Pin 8 (EchoPin)**: Connect to **Echo Pin** of **HC-SR04 Ultrasonic Sensor** (North).
- **Pin 9 (TrigPin)**: Connect to **Trig Pin** of **HC-SR04 Ultrasonic Sensor** (North).
- **Pin 10 (EchoPinNE)**: Connect to **Echo Pin** of **HC-SR04 Ultrasonic Sensor** (North-East).
- **Pin 11 (TrigPinNE)**: Connect to **Trig Pin** of **HC-SR04 Ultrasonic Sensor** (North-East).
- **Pin 12 (EchoPinNW)**: Connect to **Echo Pin** of **HC-SR04 Ultrasonic Sensor** (North-West).
- **Pin 13 (TrigPinNW)**: Connect to **Trig Pin** of **HC-SR04 Ultrasonic Sensor** (North-West).

4.3. L298N Motor Driver Board Pin Connections:

- **IN1 (Motor A Forward)**: Connect to **Pin 4 (MLa)** on Arduino.
- **IN2 (Motor A Backward)**: Connect to **Pin 5 (MLb)** on Arduino.
- **IN3 (Motor B Forward)**: Connect to **Pin 6 (MRa)** on Arduino.
- **IN4 (Motor B Backward)**: Connect to **Pin 7 (MRb)** on Arduino.
- **ENA (Motor A Enable)**: Connect to 5V pin on Arduino for enabling the left motor.
- **ENB (Motor B Enable)**: Connect to 5V pin on Arduino for enabling the right motor.
- **VCC**: Connect to **9V Battery** for motor power.
- **GND**: Connect to **GND** on Arduino.

4.4. HC-SR04 Ultrasonic Sensor Connections:

For the three ultrasonic sensors, the connections are as follows:

- **HC-SR04 (North):**
- **Trig Pin:** Connect to **Pin 9 (TrigPin)** on Arduino.
- **Echo Pin:** Connect to **Pin 8 (EchoPin)** on Arduino.
- **HC-SR04 (North-East):**
- **Trig Pin:** Connect to **Pin 11 (TrigPinNE)** on Arduino.
- **Echo Pin:** Connect to **Pin 10 (EchoPinNE)** on Arduino.
- **HC-SR04 (North-West):**
- **Trig Pin:** Connect to **Pin 13 (TrigPinNW)** on Arduino.
- **Echo Pin:** Connect to **Pin 12 (EchoPinNW)** on Arduino.

4.5. Power Connections:

- **Cubed together:** This means connecting the **positive terminals** of both 9V batteries together and also the **negative terminals** of both batteries together.
- **Positive Terminal:** Connect this combined positive terminal to the **Vin** pin of the Arduino Uno (through the power port).
- **Negative Terminal:** Connect this combined negative terminal to both the **GND pin on the Arduino** and the **GND pin on the L298N motor driver.**

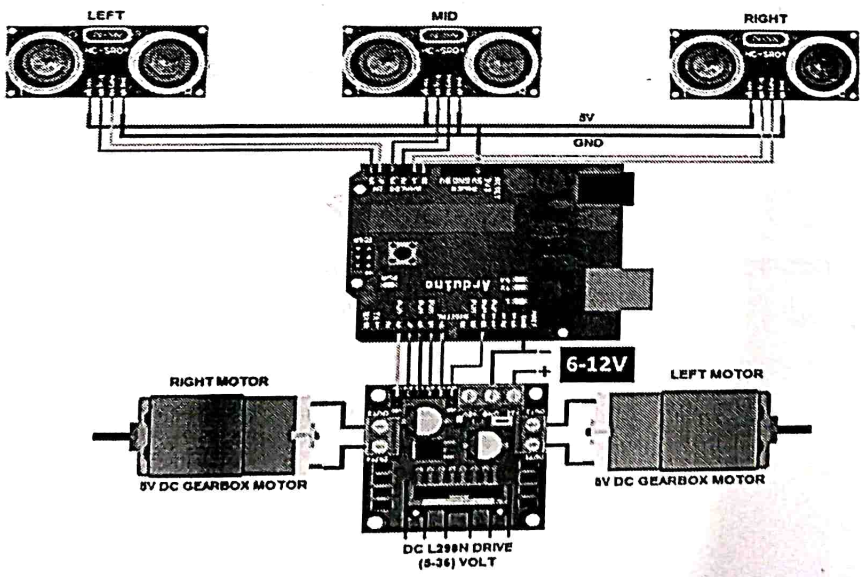

Figure 6. Circuit Diagram and Connections

Chapter 5: Working Of Model

5.1. Obstacle Avoidance in the Obstacle Avoiding Robot

The obstacle avoiding robot is an AMR developed to move in its operational environment while detecting real time obstacles and avoiding them. To obtain the aforementioned capabilities the system incorporates the following key components: ultrasonic sensors, a motor driver, DC motors, and an Arduino Uno controller. The robot's working can be divided into several steps: recognition of the environment, planning and selection of response approaches, and control of movement signals.

5.2. Obstacle Detection:

The robot uses three HC-SR04 ultrasonic sensors, each mounted at different positions on the robot: One of them is a flap facing front (North), the other is at the North East and the third one faces North West. Each sensor measures distance from any object by sending audible signals in the form of pulse & waves that will be traced back by persons or objects in its path. The time it takes for the sound waves to return is measured and converted into a distance using the formula:

$$\textit{Distance} = \textit{Duration} \ X\ 58.2$$

Duration is the time taken for the sound wave to travel to the object and back.

Since the sound travels to the obstacle and back, the formula divides the total time by 2 to get the one-way distance. The **minimum distance** is typically set around **15 cm** in the code, meaning if any sensor detects an object within 15 cm, the robot will consider it as an obstacle and initiate avoidance behavior.

5.3. Obstacle Avoidance Strategy:

When there is an obstacle in front of the robot (the ultrasonic sensor returns a value smaller than 15 cm), the system action to avoid it is to stop for a while, turn backward for a short distance, and then turn around to search for a path. In the avoidance process one is supposed to check the sensors in the North-East and North-West direction.

1. **Moving Forward:**

Firstly, the robot goes straight. Arduino constantly reads distance values that are received from ultrasonic sensors all the time. Again, if all the distances from the forward-facing sensor, North-East Sensor and the North-West Sensor are more than 15 cm the robot continues to move forward.

2. **Obstacle Detection:**

If the distance becomes less than 15 cm, it means that there is an obstacle in front of the robot – any of the detectors is set to alarm – then the forward movement is stopped. The robot then withdraws a little backward by taking a small step to create more space for it to move in and to help it to capture a better view of the environment.

3. **Scanning and Turning:**

After the backup, the robot navigates the surroundings forwarding by the help of North-East and North-West Ultrasonic sensors. As constructed from the readings, the robot makes decisions whether to turn left or to turn right. The specific output is that, if the side either North-East or North-West has more available space, the robot turns toward that side.

5.4 Working of void loop in the Code:

Void loop() function is the most important function of the Arduino code of the obstacle avoidance robot. It also patrols the environment by continuously viewing through the three ultrasonic sensors and makes decisions instantly regarding the direction of movement of the robot. The logic in this function allows the robot to move and avoid obstacles within its path on its own depending on information received from the ultrasonic sensors. Here's a detailed breakdown of the working of void loop() for your setup with 3 ultrasonic sensors (North, North-East, and North-West):

Step 1: Distance Read by Ultrasonic Sensors

The void loop () begins with the call to the function get Distance() for the three ultrasonic sensors; North, North-East, and North-West. These sensors determine the distance between the robot and any object that is in front of them by continuously emitting ultrasonic waves and then measuring the amount of time it takes for the pulses to get back.

- **North Sensor (Front):** This sensor is positioned to detect obstacles directly in front of the robot.
- **North-East Sensor:** This sensor is angled to detect obstacles at a 45-degree angle to the robot's right.
- **North-West Sensor:** This sensor is angled to detect obstacles at a 45-degree angle to the robot's left.

Each sensor sends a pulse, and the time it takes for the pulse to return is converted into a distance value using the standard formula. This formula calculates the one-way distance based on the time of flight of the ultrasonic pulse. These distance values are then stored in the variables distance Front, distance NE, and distance NW.

Step 2: Obstacle Detection

Following the data acquisition from the three ultrasonic sensors, the program processes the distance values and scans whether any of the distances is below a defined minimum value which is usually 15 cm depending on the application. If the distance is smaller than this threshold, it means that there is an object in front of the robot which will force the robot to halt, then reverse and do another scan.• If all the sensors detect distances above 15 cm the robot remains to move forward.• If the from See_days_distance distanceFront is less than 15 from an object, the robot will halt and then go back one step in order to increase the scanning distance. Whenever the robot starts backing up it makes another check on both the North-Eastand the North-West sensors to determine which way to turn.hecks if any of the distances are below a predefined threshold, typically 15 cm (you can adjust this as needed). If the distance is smaller than this threshold, it indicates the presence of an obstacle in front of the robot, and it will trigger the robot to stop, back up, and re-scan the environment. Here's how it works:

- **If all sensors report distances greater than 15 cm**, the robot continues to move forward.
- **If the front sensor (distanceFront) detects an obstacle** (distance < 15 cm), the robot stops moving and then takes a backward step to allow more space for scanning.
- After backing up, the robot re-checks the **North-East** and **North-West** sensors to decide which direction to turn.

Step 3: Backward Motion and Re-scan

The moment an obstacle is spotted by the front sensor the robot shifts backward. This backward movement is by using the L298N motor driver that stops the motors and then reverses their direction for some few seconds (approximately 500ms). The rationale for this movement is so that the space around the robot can be scanned again to enable a proper decision when it comes to the direction to take.

Step 4: Scanning for Clear Path

Upon backward movement, the robot halts and scans only the environment with the help of NE and NW ultrasonic sensors again. These facilitate helping the robot decipher which side is more open to allow it to turn to which side in a bid to avoid the obstacle. If the North-East sensor differences with the North-West sensor suggest that the latter has seen more space, the robot will turn right in the direction of North-East. If the North-West sensor provides a greater number of space than the North-East sensor, the robot will then move left (in North-West direction). If both directions have nearly equal amounts of space the robot may turn in either direction depending on the implantation or turn in proportion to the amount of space provided. These sensors help the robot determine if there is more space to the left or rightand which direction it should turn to avoid the obstacle.

- **If the North-East sensor detects more space than the North-West sensor**, the robot will turn right (towards the North-East direction).
- **If the North-West sensor detects more space than the North-East sensor**, the robot will turn left (towards the North-West direction).
- **If both directions have roughly equal space**, the robot might turn in either direction based on the implementation or turn proportionally based on the available space.

Step 5: Turning

This robot uses the motor driver (L298N) to take control of the motors and as well as perform the turn. Instead, it controls the power supplied to the motors, an aspect that makes the robot turn in the required direction. The turn is dependent upon the amount of space toward which it is turning, that is, the robot will turn through a greater angle if less space is available or through a smaller angle if more is available.

- **Right Turn (if North-East is clearer):** The robot will move the left motor forward and the right motor backward to rotate right.
- **Left Turn (if North-West is clearer):** The robot will move the right motor forward and the left motor backward to rotate left.

Step 6: Resuming Forward Motion

After it has turned, the robot continues to move and of course scans the new environment for more obstacles. The void loop() repeats so that the robot is informed of new obstacles as they emerge, and is able to avoid them accordingly.

The entire process of detecting obstacles, backing up, scanning and turning is done as long as the robot is switched on and hence making the robot move on its own.

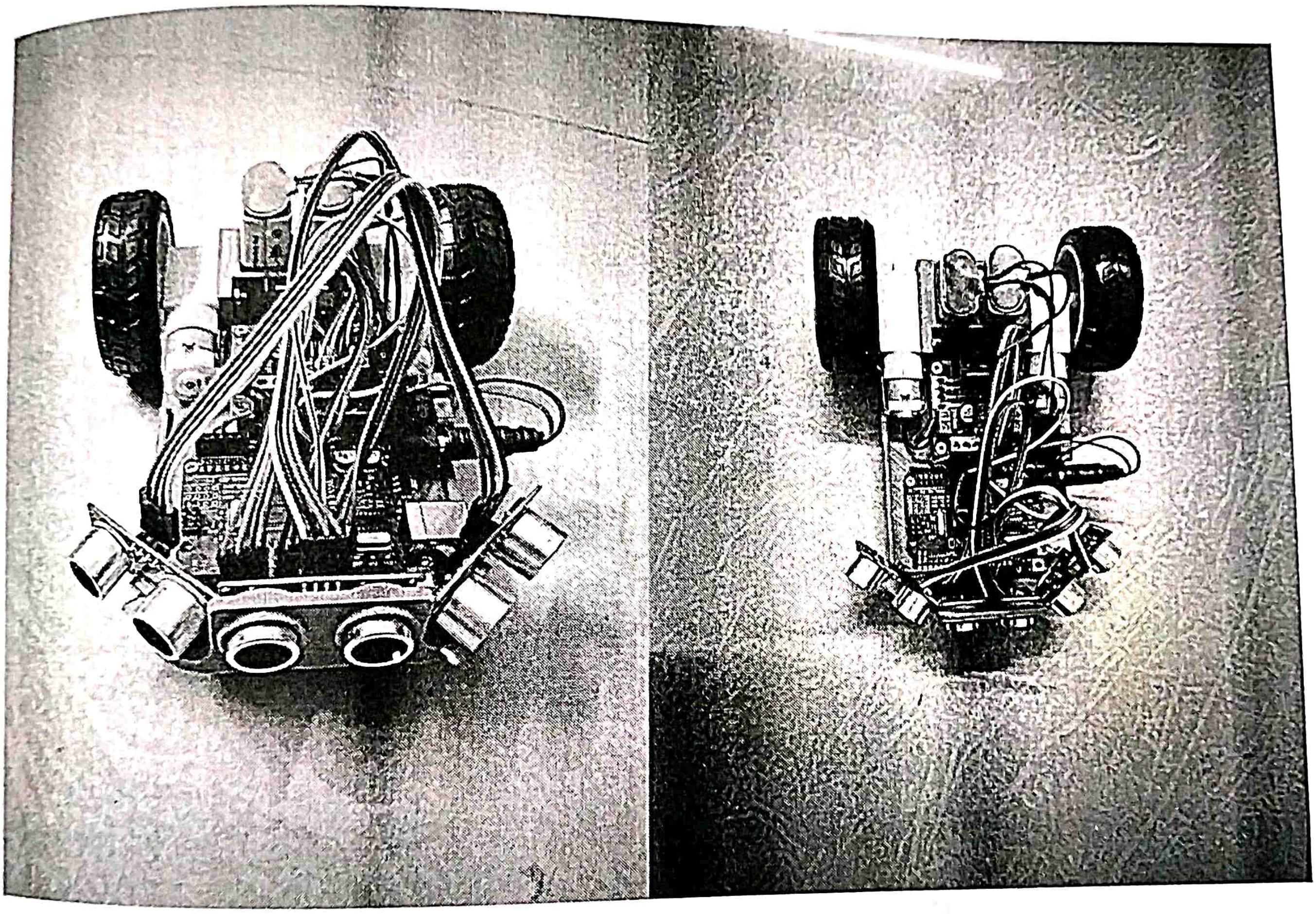

Figure 7. Final Setup of the Model (integration of all sensors along with code)

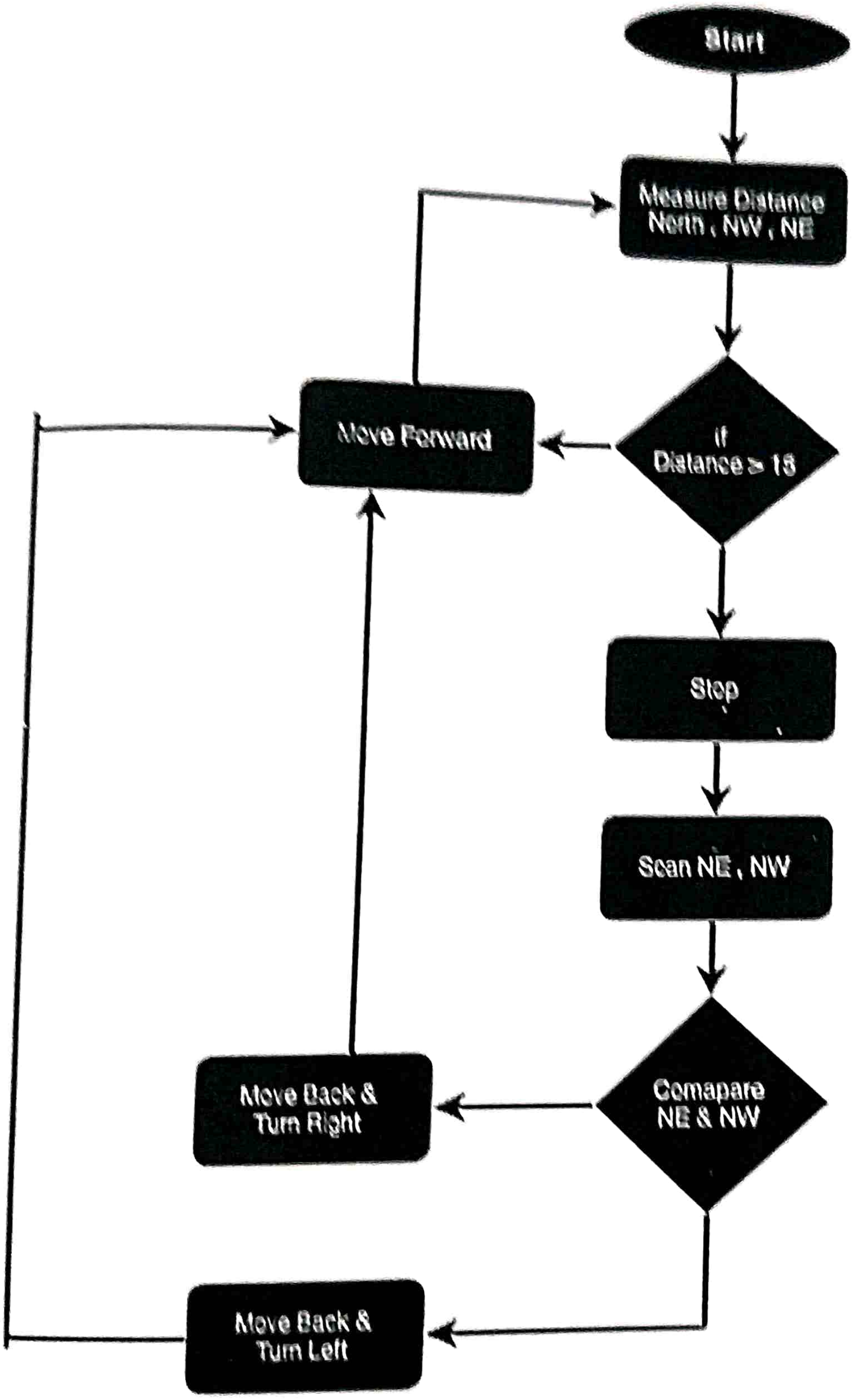

Figure 8. Control Flow of the Code

CHAPTER 7: CONCLUSION AND FUTURE SCOPE

7.1. Conclusion

Thus in this minor project, I have been able to achieve my aim and build an obstacle avoidance robot using an Arduino Uno, motor L298N driver and three ultrasonic sensors. As a result, the primary goal of the project is to ensure that the autonomous robot identifies obstacles within the environment to avoid in order to reach its goal. The robot comprises several sensors and motors thus allowing it to move around the environment and make decisions based on the output of the sensors. The ultrasonic sensors positioned to face forward, North-East, and North-West are favorable for the robot since they widen its field of vision and thus improve the recognition of the obstacle in all directions.

It is full of simple, but effective commands, allowing the robot to navigate around obstacles, such as stop at an obstacle, back up to gain clearance, scan the possible paths, and turn in the correct direction. Based on the sensor readings, turning ratio proportional motion enables the robot to respond to its environment and move in which direction is best suitable. To fuse the decisions made, the algorithm guarantees that the robot reasons and comes up with the best decisions depending on the available space and other objects.

Control is equipped with Arduino Uno board, while L298N motor module is employed for proper motor control and energy consumption. The incorporation of the ultrasonic sensors provides the robot efficient distance measurements, thus allowing the robot to respond to dynamic objects in its path. The successful completion of these components proves the overall goal of the robot to achieve mobile navigation all on its own, understanding the fundamentals of robotic movement and sensorial decision making.

7.2. Future Scope

The current configuration and organization, while sufficient to offer a practical means of achieving minimal obstacle avoidance, leaves ample room for improvement and improvement in robotic performance. One of the future improvements could be to switch totally from Arduino to Raspberry Pi 5 (8GB ram) as it will offer more computational power, instant memory and space for implementing more algorithms such as ML and Fuzzy logic.

7.2.1. What we can Do with Raspberry Pi 5 (8GB RAM)

The Raspberry Pi 5 which has 8GB RAM provides significant improvement in processing power than the Arduino. This would allow for higher powered algorithms and the associated ML models to be executed directly on the robot, greatly improving its prims and sophisticated decision making abilities. Moreover, the computation ability of the Raspberry Pi indicates it can perform other complicated processes like image processing, data analysis in real-time, which is very important in terrain classification.

Incorporating the Raspberry Pi 5, the robot will not be constrained again to the constrained computational platform of the Arduino. It will be able to perform Python based ML models, using the libraries like TensorFlow or PyTorch for image analysis for classification and terrains. This will make it easier for the robot to understand changes in conditions in its environment with better precision.

7.2.2. Camera Module Integration for Terrain Classification

Among the essential upgrades that should be done to the robot we should mention the camera, for example Raspberry Pi Camera Module. This camera can be used to provide real time images of the environment surrounding the robot, images that can be used to categorise the kind of terrain. Classification of terrain types such as grass, mud, gravel, and concrete will enable the robot to take proper decisions for right path, right speed and right turning. There are two options to do this – the use of image processing algorithms or deep learning models that constantly input the data from the camera into the Raspberry Pi so that the robot is aware of the different surfaces and characteristics of the area it is operating in.

7.3.3. Fuzzy Logic to Support the Trip on Dynamically Changing Terrains

Fuzzy is going to be a part of our control strategy to help the robot make decisions where information may be either vague or incomplete. While binary decision-making is common, in the framework of fuzzy logic, inputs are valued from a set of continuous variables, and outputs are also integer-valued. This will be particularly useful whenever the robot is faced with various surfaces which could range from smooth to rough, or flat and irregular.• If the robot interprets its camera data and perceives a grassy environment, it may slow down its pace so that it is lighter and subsequently, tweak its ability to turn on its axis so as to handle the looser, unstable ground.• For instance when moving on surfaces such as mud or gravel, then the speed will have to be even lower with high accuracy of turning to avoid sliding.• That is why for concrete, if the surface is smooth and firm enough the robot can go at full speed and rarely change its turning strategy.odule, such as the Raspberry Pi Camera Module. This camera can be used to capture real-time images of the robot's surroundings, which can then be processed to classify the terrain. The ability to classify terrain types such as grass, mud, gravel, and concrete will allow the robot to make more informed decisions regarding its movement, speed, and turning behavior. By using image processing algorithms or deep learning models, the camera can feed visual data into the Raspberry Pi, enabling the robot to recognize different surfaces and adjust its characteristics accordingly.

7.2.4. Fuzzy Logic for Dynamic Terrain Adaptation

Incorporating Fuzzy Logic will allow the robot to make decisions based on uncertain or imprecise information. Unlike traditional binary decision-making (e.g., obstacle detected or not detected), fuzzy logic uses a range of values to evaluate inputs and provide more nuanced outputs. This will be particularly useful when the robot encounters various terrain types with varying degrees of roughness or unevenness.

For example:

- If the robot detects grass, it may decrease its speed and adjust its turning radius to accommodate the softer, less stable surface.
- In the case of mud or gravel, the robot may need to reduce its speed even further and increase the turning precision to avoid slipping.
- For concrete, where the surface is smooth and firm, the robot can operate at its maximum speed with minimal adjustments to its turning behavior.

References

References:

1. D. L. McCauley, *Arduino Robotics*, 1st ed. New York, NY, USA: McGraw-Hill, 2013.

2. S. S. L. Yang, *Raspberry Pi Cookbook: Software and Hardware Problems and Solutions*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2016.

3. A. J. Van den Bossche, *Robotic Systems and Autonomous Platforms*, 2nd ed. Oxford, UK: Oxford University Press, 2017.

4. S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Upper Saddle River, NJ, USA: Pearson Prentice Hall, 2009.

5. D. Cohn, *Fuzzy Logic: A Practical Approach*, 1st ed. London, UK: Academic Press, 2000.

6. L. He, Z. Yu, and L. Zhang, "A review on image classification using convolutional neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3368-3378, Nov. 2019.

7. G. G. T. D. H. Nguyen and T. N. H. Nguyen, "Terrain Classification for Autonomous Robots Using Machine Learning and Fuzzy Logic," in *Proceedings of the 2019 IEEE International Conference on Robotics and Automation (ICRA)*, Montreal, Canada, May 2019, pp. 1234-1241.

8. C. L. Liu, S. K. Sarma, and A. J. Ziegler, "Terrain-adaptive Autonomous Robots Using Multi-Sensor Fusion and Machine Learning," *Journal of Robotics and Automation*, vol. 29, no. 2, pp. 115-124, Feb. 2021.

9. B. D. Patterson and L. R. Smith, *Introduction to Machine Learning with Python: A Guide for Data Scientists*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2018.

10. J. H. S. Tran, *Fuzzy Control of Autonomous Systems*, Springer, 2018.

PAPER NAME

AYUSH minor file 1 (1).pdf

AUTHOR

ADITYA DUBEY

WORD COUNT

4875 Words

CHARACTER COUNT

25153 Characters

PAGE COUNT

16 Pages

FILE SIZE

941.5KB

SUBMISSION DATE

Nov 19, 2024 2:59 PM GMT+5:30

REPORT DATE

Nov 19, 2024 3:00 PM GMT+5:30

6% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 2% Internet database
- 2% Publications database
- Crossref database
- Crossref Posted Content database
- 5% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Small Matches (Less then 10 words)

- 6% Overall Similarity

Top sources found in the following databases:

- 2% Internet database
- 2% Publications database
- Crossref database
- Crossref Posted Content database
- 5% Submitted Works database

OP SOURCES

he sources with the highest number of matches within the submission. Overlapping sources will not be isplayed.

1	Vibhav Sachan, Shahid Malik, Ruchita Gautam, Kumar Parvin. "Advanc... Publication	<1%
2	The London College UCK on 2017-06-01 Submitted works	<1%
3	Asia Pacific University College of Technology and Innovation (UCTI) on... Submitted works	<1%
4	P. Dharani Devi, C. A. Pavithra. "PIR Sensor based Blaze Barricade usin... Crossref	<1%
5	East Delta university on 2020-11-17 Submitted works	<1%
6	City University of Hong Kong on 2023-10-09 Submitted works	<1%
7	renesas.com Internet	<1%
8	Savitribai Phule Pune University on 2016-07-20 Submitted works	<1%

9	newporttraining on 2024-11-11 Submitted works	<1%
10	unimal on 2024-10-27 Submitted works	<1%
11	American University of the Middle East on 2023-01-05 Submitted works	<1%
12	Asia Pacific University College of Technology and Innovation (UCTI) on... Submitted works	<1%
13	Jared M. Moore, Anthony J. Clark, Philip K. McKinley. "Evolution of stat... Crossref	<1%
14	University of Central Lancashire on 2024-04-21 Submitted works	<1%
15	ir.kluniversity.in Internet	<1%
16	solarbotics.com Internet	<1%
17	berrybase.de Internet	<1%