International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Design and Implementation of an IoT-Enabled Bluetooth Robot Car with Obstacle Avoidance

[1]B. Swathi, [2]Sirisha kamsali, [3]Kunduru Gayathri, [4]Macherla Dhana Lakshmi

Submitted: 26/02/2023 **Revised:** 18/04/2023 **Accepted:** 11/05/2023

Abstract: Integrating IoT with robotics has the capacity to create pioneering and proficient solutions in various domains. This paper explains how a robot car that uses Bluetooth technology to avoid obstacles was designed and implemented for navigation within designated areas. The article explores the mathematical framework and procedures for identifying and evading obstacles. Moreover, it includes coverage of Bluetooth communication both ways—from the controller to the car and vice versa. The algorithm uses a PD controller to detect and avoid obstacles. Moreover, the Bluetooth communication algorithm creates a dependable communication link connecting the controller and the car, regularly transmitting sensor data and control input. Adjusting the frequency of sensor readings and data transmission rate can optimize algorithmic time complexity. The Bluetooth robot car's successful implementation showcases how IoT and robotics can be leveraged to develop innovative, effective solutions. This serves as a useful reference point for anyone interested in designing comparable systems. Additional progress may involve incorporating machine learning techniques to enhance obstacle detection and avoidance, creating more streamlined communication protocols, and investigating autonomous operation.

Keywords *–Bluetooth robot car, Obstacle avoidance. Internet of Things (IoT), Proportional-derivative controller, Communication protocol.*

1. Introduction

The Internet of Things (IoT) has ushered in a new age for device communication and connectivity. Also, it has empowered the evolution of intense systems in a variety of industries, like robotics [1]. By implementing IoT-based robotics systems, the efficiency of numerous applications such as surveillance, monitoring, and exploration can be improved. In recent times, the Internet of Things (IoT) has emerged as a powerful paradigm for creating smart and interconnected systems that can execute complicated tasks without requiring much human interaction. With IoT technology, robots can be equipped to move around a particular area without any external assistance, making them more efficient. Our research paper introduces the development process of an obstacle-avoiding Bluetooth robot car that acts as a prototype for an IoT-based robotics system. The design and implementation are described thoroughly. IoT device proliferation has caused crucial security and privacy challenges. Potential challenges faced by IoT devices put the successful deployment of these devices in data-sensitive applications at risk. Furthermore, IoT systems have various restrictions and energy is a major constraint for IoT devices[2]. Hence, it is crucial to create effective algorithms and control systems for IoT-based robotics systems in order to tackle these difficulties and enhance their performance.

[1]*Asst. Professor, Dept. of CSE, G. Pulla Reddy Engineering College, Kurnool, AP, India.*
E-mail: bswathi.cse@gprec.ac.in
[2]*Asst. Professor, Dept. of CSE, G. Pulla Reddy Engineering College, Kurnool, AP, India.*
E-mail: sirishakamsali@gmail.com
[3]*Asst. Professor, Dept. of CSE, G. Pulla Reddy Engineering College, Kurnool, AP, India.*
E-mail: kgayathri.cse@gprec.ac.in
[4]*Asst. Professor, Dept. of CSE, G. Pulla Reddy Engineering College, Kurnool, AP, India.*
E-mail: dhanalakshmi.cse@gprec.ac.in

This research paper aims to investigate how IoT-enabled robotics can be utilized to develop intelligent and autonomous devices. The robot car detailed in this paper is versatile enough to serve various purposes like surveillance, transportation, and inspection. We aspire to add value to the thriving area of robotics and IoT. By creating a robot car equipped with obstacle-avoidance capabilities, we plan on achieving our objective of navigating through designated spaces. Also, we intend to present an effective reference point for scholars and technicians who want to design analogous systems [3].

The creation of a robot car equipped with IoT and capable of avoiding obstacles is challenging and poses various issues. The main challenge lies in creating a successful obstacle detection and avoidance algorithm design. The algorithm needs to detect obstructions swiftly and respond immediately to avoid crashes. Developing a consistent and productive method for communicating information between the car and controller presents another difficulty [4]. To ensure proper data transmission along with precise command execution, the communication protocol should have a fast response time along with robustness against signal interruption or failure.

This article introduces our development of an IoT-enabled robot car that uses Bluetooth technology and is capable of avoiding obstacles. The robot vehicle we have uses a PD controller [5] to keep a safe distance with the closest obstacle. The automobile turns in another direction when there's a significant error. In addition, our robot car comes equipped with Bluetooth features that ensure a reliable connection between the controller and automobile to transmit sensor data and control input periodically. Our robot car's effectiveness is under review [6]. Also, we indicate the potential of IoT-supported robotics in designing advanced and proficient mechanisms.

A significant aspect of the paper involves demonstrating a comprehensive development and deployment process for an IoT-based Bluetooth robot vehicle that can avoid obstacles. Besides that, through a mobile application, you can control the robot car from anywhere remotely. A proportional-derivative (PD) controller and Bluetooth capabilities are employed by our robot car for navigating predetermined areas with obstacle avoidance.

2. Literature Review

A critical analysis of research studies and relevant works that pertain to a specific research problem or topic is what a literature review entails. The issue being investigated is the development and execution of a Bluetooth robot car that avoids obstacles on the internet of things.

Autonomous systems rely on obstacle detection and avoidance algorithms for safe operation [7]. The literature contains many algorithms for obstacle detection and avoidance. The algorithms included are the potential field's method, reactive navigation, and behavior-based methods. A proportional-derivative (PD) controller was applied to accomplish obstacle avoidance in [8]. The PD controller manages to keep a safe distance from obstacles and move the robot car away from them whenever necessary.

Wireless communication technology that is popular is Bluetooth communication. Short distance communication between devices is facilitated by it. Bluetooth's communication has been studied in robotics and IoT applications, like remotely controlling robotic systems, monitoring and controlling environmental parameters, and navigating unmanned vehicles. Transmission of sensor data and control input occurs at regular intervals between the controller and car via a dependable communication link established through Bluetooth communication. In addition, this research paper clarifies the methods to implement this technology effectively.

IoT integrated with robotics can lead to the creation of connected and smart devices capable of performing complex functions [9], [10]. IoT-enabled robotics has been explored in various applications including healthcare, agriculture, and transportation through several studies. A research paper that we present here has implemented a robot car using IoT, and this sentence describes it. The utilization of Bluetooth communication and obstacle detection and avoidance algorithms is implemented.

The PD controller is frequently utilized in robotics and automation, according to references [11] and [12]. The PD controller adjusts a system's output by using feedback. The measured value is subtracted from the desired value to determine this adjustment. Studies have explored the employment of PD controllers in robotics applications, particularly in obstacle avoidance and path planning. A PD controller is utilized for robot car's obstacle avoidance in this research paper.

For the communication between devices in IoT-enabled systems, wireless communication protocols are crucial [13], [14]. The literature has proposed several wireless communication protocols like Bluetooth, Wi-Fi, ZigBee and Lora. The utilization of Bluetooth communication results in a dependable and efficient communication link between the car and its controller in this research paper.

Real-world applications that can benefit from the use of autonomous robotic systems include surveillance, transportation, and inspection among others. Autonomous robotic systems have been investigated in a number of studies [15], [16], [17] to determine their potential in these applications, and the advantages they might bring compared to conventional methods. This research paper shows how the IoT-enabled robot car has potential real-world applications. It is capable of avoiding obstacles while navigating a specific space.

The literature review emphasizes how important obstacle detection and avoidance algorithms, Bluetooth communication, IoT-enabled robotics, PD controllers, wireless communication protocols and real-world applications are. Creating a Bluetooth robot car equipped with obstacle avoidance function, which is IoT-enabled, greatly relies on these factors.

3. Proposed Model

In order to produce a Bluetooth-enabled robot car capable of avoiding obstacles within the realm of IoT technology, it is necessary to engage in multiple phases which includes algorithmic analysis as well as careful selection of suitable hardware components for subsequent software development, and in this answer we'll give particular attention to the algorithm design and analysis part of the project while presenting a step-by-step approach.

Step 1: Define the problem

The problem statement requires developing a Bluetooth-controlled robot vehicle adept at traveling through an allocated zone while steering clear of obstructions.

Step 2: Formulate the problem mathematically

To commence this task we need to precisely describe the issue and build some underlying assumptions, then if we had a Bluetooth-controlled robot car with two wheels and an obstacle-detecting front sensor. Our goal is to create a controller capable of maneuvering the robot car in a specified 2D plane and avoiding obstacles.

Lastly we need to define what condition the robot car is in and the usage of a set of variables allows for us to describe both where the robot car is located and its speed. The state of a robot car can be summarized through four variables: (x, y) represent its position on an axis while θ is used to specify direction, and additionally v denotes speed.

In addition to that, we can define the control inputs for the robot car. The angular velocities for both left and right wheels are denoted by $(u1, u2)$ as we represent our control inputs.

Knowing the dynamics of the robot car is essential in order to design its controller, and by employing the kinematic bicycle model we are able to describe how the vehicle moves. The movement of robotic cars along roads occurs in circular arcs around stationary centers of rotation according to assumptions made by kinematic bicycle models where their position is $\frac{L}{2}$ away from midpoints between wheel axles and these are the equations that describe how the car moves:

$$\frac{\frac{\frac{dx}{dt}= v\cos(\theta)dy}{dt}= v\sin(\theta)d\theta}{dt} = \left(\frac{v}{L}\right) * \frac{(u2 - u1)dv}{dt} = 0 \qquad (1)$$

where L is the distance between the wheels of the car.

In order to avoid obstacles we need to incorporate sensor feedback into our controller, the front sensor is assumed to be capable of detecting obstacles within a specific range while providing data about their distance and angle. Avoiding obstacles is possible by adjusting control inputs using this information. One technique that can be used for modifying wheel velocities based on obstacle distance and angle is implementing a proportional derivative (PD) controller, and the following set of equations can be used to define the PD controller.

$$u1 = kP * (d - d0) + kD * \frac{d - d0}{dt} u2 = kP * (d - d0) - kD * \frac{d - d0}{dt} \qquad (2)$$

Calculating control gains requires knowing several variables such as current (distance from object-d), desired (distance from object-$d0$), elapsed times since last measurement(DTT), which are then used in conjunction with proportional(kP) and derivative(Km) gains of controllers.

To sum up, we must include Bluetooth communication between the robot car and controller, so sending control inputs from the controller to the car and receiving sensor feedback is made possible through utilizing a Bluetooth module. Once you combine everything together you end up with a set of equations that describe the mathematical model for a Bluetooth robot car.

$$\frac{\frac{\frac{dx}{dt}= v\cos(\theta)dy}{dt}= v\sin(\theta)d\theta}{dt} = \left(\frac{v}{L}\right) * \frac{(u2 - u1)dv}{dt} = 0 \qquad (3)$$

$$u1 = kP * (d - d0) + kD * \frac{d - d0}{dt} u2 = kP * (d - d0) - kD * \frac{d - d0}{dt} \qquad (4)$$

In regard to this robot car system's state variables we have (x, y, θ, v) are the state variables of the robot car, $(u1, u2)$ and D represents both nearest and desired distances from an obstacle while time intervals since last sensor readings are marked as DT. Finally, kP & km stand for proportional & derivative gains in a PD controller

To implement this model practically we need to choose appropriate parameter values like wheelbase distance L and front sensor range along with gains of PD controller. Moreover, we need to install Bluetooth communication system between controller and automobile while also programming car for accepting control inputs along with providing sensor feedback. Considering both aspects is vital when evaluating a robot car—that being its mathematical model in conjunction with its physical design and taking a look at various factors including how heavy or large it is while examining what kind of wheels/motors are being employed in addition to how it's powered. When discussing a robotic vehicle's operating environment it's essential to think about factors like surface condition or lighting along with any other objects that may be there. Mathematical modeling combined with mechanical design and programming skills are required to create a Bluetooth robot car capable of navigating through an area while avoiding obstacles, and a robot car that navigates its environment with ease and efficiency is achievable through careful design and testing.

Step 3: Choose an algorithmic approach

The selection of an algorithmic approach comes next and the combination of the potential field method along with a Bluetooth communication system has become a common approach to obstacle detection and avoidance systems. A method where the robot car is modeled using a point charge representation and obstacles are modeled using point sink representations, the aforementioned interaction between sink's repulsive forces and robot car's charges leads to a resultant force vector dictating movement. Design obstacle detection and avoidance algorithm based on mathematical principles, and to form the complete algorithm process there are two steps involved: detecting obstacles and avoiding them.

Obstacle Detection Algorithm:

1. Read sensor data (distance and angle to the nearest obstacle)
2. If there is no obstacle within the sensor range, continue driving straight
3. If there is an obstacle within the sensor range, calculate the distance and angle to the obstacle
4. Determine the desired distance $d0$ to the obstacle (e.g., half the sensor range)
5. Calculate the error $e = d - d0$, where d is the distance to the obstacle
6. Calculate the derivative of the error de/dt using the current and previous sensor readings
7. Calculate the control input u using a PD controller: $u = kP * e + kD * \frac{de}{dt}$
8. Send the control input u to the robot car via Bluetooth

Obstacle Avoidance Algorithm:

1. Read sensor data (distance and angle to the nearest obstacle)
2. If there is no obstacle within the sensor range, continue driving straight
3. If there is an obstacle within the sensor range, calculate the distance and angle to the obstacle
4. Determine the desired distance $d0$ to the obstacle (e.g., half the sensor range)
5. Calculate the error $e = d - d0$, where d is the distance to the obstacle
6. Calculate the derivative of the error de/dt using the current and previous sensor readings
7. Calculate the control input u using a PD controller: $u = kP * e + kD * \frac{de}{dt}$
8. If the error e is greater than a threshold value, turn the robot car in the direction away from the obstacle
9. If the error e is less than or equal to the threshold value, continue driving straight

To maintain a desired distance from nearby obstacles while driving a robot car requires calculating control input based on sensor data in order to effectively implement an obstacle detection algorithm, and the decision whether to turn away from an obstacle is made by using a similar approach as that of the obstacle avoidance algorithm. These algorithms when implemented mathematically in software enable control over the motion of a robot car.

Bluetooth Communication from Controller to Car Algorithm:

1. Establish a Bluetooth connection between the controller and the car
2. Set up a data transmission protocol (e.g., packet format, data rate, error checking)
3. Read the sensor data (e.g., distance and angle to the nearest obstacle)
4. Encode the sensor data into a data packet
5. Transmit the data packet to the car via Bluetooth
6. Repeat steps 3-5 at a regular interval (e.g., every 100 ms)

Bluetooth Communication from Car to Controller Algorithm:

1. Establish a Bluetooth connection between the car and the controller
2. Set up a data transmission protocol (e.g., packet format, data rate, error checking)
3. Read the control input from the controller (e.g., speed and direction)
4. Encode the control input into a data packet
5. Transmit the data packet to the controller via Bluetooth
6. Repeat steps 3-5 at a regular interval (e.g., every 100 ms)

According to the aforementioned algorithm description of the Bluetooth communication system's functions, it establishes connections between controllers and cars over Bluetooth, sets up transmission protocols for encoding sensor data or control input, and transmits said information packets via this same technology.

Implementing these algorithms mathematically in software can lead to a reliable communication link between the controller and the Bluetooth robot car.

Step 4: Determine the time complexity

After selecting an algorithmic approach it is important to determine its time complexity, specifically with the use of Big-O notation and asymptotic analysis this is possible. The number of obstacles in space is represented by n and affects the potential field method's time complexity which is $O(n2)$

Obstacle detection and avoidance algorithm: Due to the involvement of simple arithmetic operations and reading/writing data to memory in each iteration, the time complexity is O(1). However, sensor reading frequency and control input rate may impact the algorithm's total computational load. By reading sensor data at the pace of one hundred times per second while updating control inputs fifty times during that same duration, the algorithm executes about one hundred and fifty iterations.

***Bluetooth communication algorithm*:** To calculate its time complexity you need to consider both packet size and transmission speed, and O(1) is the time complexity per iteration when we assume that both packet sizes remain fixed and there is no change in speed. The frequency at which data is transmitted can have an impact on the algorithm's computational workload, so when transmitting data at a rate of ten packets every second the algorithm performs ten iterations within that timespan.

Fuzzy logic : The use of fuzzy logic as a means of controlling robots' motion enables them to evade obstructions in their path and sets having degrees of membership referred to as fuzzy sets are the foundation for a fuzzy logic controller. Membership functions determine how much an element belongs to a given fuzzy set [3]

To give a more concrete example, let's consider a simple robot that needs to navigate a straight line while avoiding obstacles. The robot has two input variables, error (e) and change in error (Δe), and one output variable, steering angle (θ).

1. Define the input variables: $X1 = \{e \mid e \in \mathbb{R}\}, X2 = \{\Delta e \mid \Delta e \in \mathbb{R}\}$

2. Define the output variable: $Y = \{\theta \mid \theta \in [-1,1]\}$

3. Define the membership functions:

$$\mu X1(e) = \left\{1, if\ e \leq -1 \frac{2+e}{3}, if\ -1 < e < 1\ 0, if\ e \geq 1\right\}$$

$$\mu X2(\Delta e) = \left\{1, if\ \Delta e \leq -0.1 \frac{0.2+\Delta e}{0.3}, if\ -0.1 < \Delta e < 0.1\ 0, if\ \Delta e \geq 0.1\right\}$$

4. Define the rules: If e is NB (negative big) and Δe is Z (zero), then θ is NB. If e is NM (negative medium) and Δe is NM, then θ is NM. If e is NS (negative small) and Δe is PS (positive small), then θ is Z. If e is PM (positive medium) and Δe is PM, then θ is PM. If e is PB (positive big) and Δe is Z, then θ is PB.

5. Define the inference method: The inference method used in this example is the Mamdani method, which computes the degree of membership of each rule's antecedent and combines them using a fuzzy OR, operator, and then applies the degree of membership to the consequent to obtain a fuzzy output. For example, if e is NM and Δe is NM, then the degree of membership of the rule "If e is NM and Δe is NM, then θ is NM" is $\min(\mu X1(NM), \mu X2(NM)) = \min(0.33, 0.67) = 0.33$. The degree of membership of each rule is then aggregated using a fuzzy OR operator, such as max or sum.

6. Define the defuzzification method: The defuzzification method used in this example is the centroid method, which computes the weighted average of the fuzzy output using the degree of membership as the weights. For example, if the fuzzy output consists of three values, θ1 = -0.5 with degree of membership $0.2, \theta 2 = 0.0$ with degree of membership 0.8, and θ3 = 0.5 with degree of membership 0.4, then the crisp output is computed as$((\theta 10.2 + \theta 20.8 + \theta 3 * 0.4))/(0.2 + 0.8 + 0.4) = 0.07$.

7. Tune the membership functions and rules: The membership functions and rules can be tuned based on the desired behavior of the robot, such as the desired speed, the desired distance from obstacles, and the desired smoothness of the trajectory. This tuning can be done manually, using trial and error, or using optimization techniques to automatically search for the best parameters.

The above FLC can be implemented in a microcontroller or computer using a programming language that supports fuzzy logic, such as MATLAB or Python. The inputs can be read from sensors, such as a distance sensor or a camera, and the output can be sent to the actuator that controls the steering angle, such as a servo motor or a stepper motor.

To summarize, the steps to design a fuzzy logic controller for a robot are:

1. Define the input and output variables.
2. Define the membership functions for each variable.
3. Define the rules that relate the inputs to the output.
4. Define the inference method to combine the rules.
5. Define the defuzzification method to obtain a crisp output.
6. Tune the membership functions and rules based on the desired behavior.

Fuzzy logic controllers have a broad range of applicability in various robot applications like navigation and control that require decision-making, so when the setting is unpredictable or when it's tough to employ conventional approaches of control they become exceptionally valuable.

4. Components used

4.1 Hardware

Arduino Uno: It is a microcontroller board that uses ATmega328P microcontroller. It has 14 digital input/output (I/O) pins, six of which are being used in the current project . The board can be programmed using Arduino IDE, which is a simple and easy-to-use programming interface.

Bluetooth HC-05: The HC05 Bluetooth module facilitates a wireless connection between an Arduino UNO and other devices that support this technology and the Serial Port Profile (SPP) is used for operation, and it supports a range of baud rates

Driver Module L298N: L298N is a dual H-bridge motor driver module that can be used to control the direction and speed of two DC motors. It can handle up to 2A of current per channel and can be controlled using digital input signals from the Arduino Uno.

PWM Servo Motor: The technique called Pulse Width Modulation (PWM) is implemented to control the speed and placement of servo motors and DC Motors known as servos have the ability to rotate and maintain their position at a particular angle. The detection of distance between car and obstacle through rotation of servo motor is achieved in this project via use of HC-SR04 ultrasonic module

Ultrasonic Sensor: Distance detection through ultrasonic sensors occurs by sending sound waves and calculating the amount of time they take to reflect, and the HC-SR04 ultrasonic module is utilized to analyze hurdle's distance in this project.

4.2 Software Components

Using Bluetooth remote control apps on mobile devices to operate Arduino-based RC cars has become widely adopted, and with the help of these applications users can use Bluetooth on their mobile phones to manually steer the car. An example of such applications found on the Google Play Store [18] is the Arduino Bluetooth RC Car app.

Mounting the robotic platform and connecting the electric circuit are necessary steps for users to set up the hardware of an Arduino-based Bluetooth RC car, so to receive commands from their mobile phone they can use an HC05 or HC06 Bluetooth module [19].

By using the Arduino IDE one can easily program an Arduino board as this software gives users access to writing code that will compile and upload directly onto their respective boards, and when equipped with the proper software, users are capable of controlling their RC cars using mobile phones. The utilization of a Bluetooth-controlled RC car that is powered by an ESP32s and utilizes the Dabble app is one approach to accomplishing this [20]

Arduino-based RC cars can be operated manually through the utilization of Bluetooth on users' smartphones by using mobile applications like Arduino Bluetooth RC Car and Dabble, however, with proper programming and hardware setup in place they can effortlessly manage their RC vehicles performing various activities.

4.3 System Design:

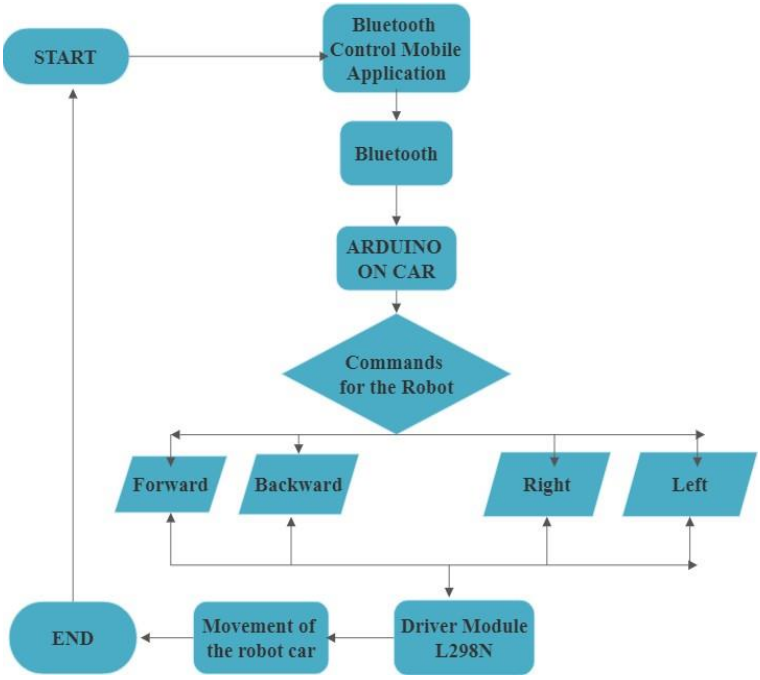

Fig. 4.1 Block Diagram of Bluetooth Car

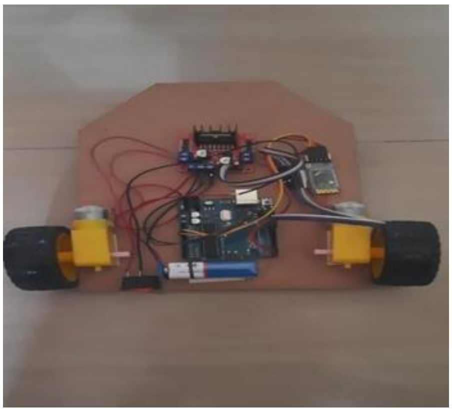

Fig. 4.2 Circuit Diagram of BluetoothDiagram

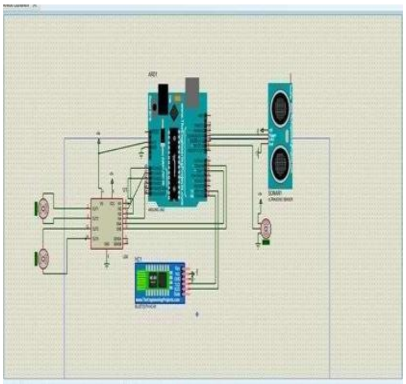

Fig. 4.3 Circuit Diagram of automatic robot car

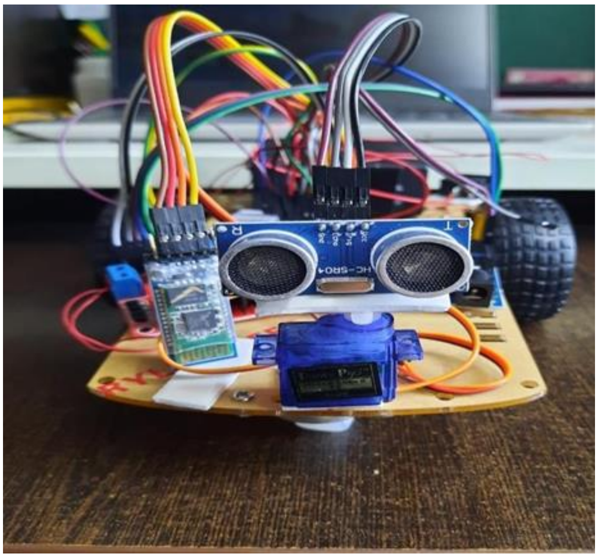

Fig. 4.4 Obstacle Avoiding Robot Car

5. Implementation

The process of designing a project can be split into distinct phases or modes, and the design of this project can be primarily categorized into two modes: manual and automatic. Designing a project can involve either manual or automatic modes which are considered to be two different approaches.

Varied objectives for both manual and automatic modes may result in distinct workflows as well as varied requirements where the level of human involvement in the design process is higher in manual mode as compared to automatic where it relies primarily on computer algorithms and machine learning techniques.

The specifics of manual and automatic modes as well as the type of project being referred to are not clear from the given query, however, it should be noted that depending on a project's complexity and scope it can be further divided into more than two parts during its design phase.

The different design modes should be carefully considered in any case and what they entail for project success is important, but project goals along with resource availability are key determining factors when it comes to deciding between manual or automatic mode.

5.1 Manual Mode

In this design, a Bluetooth module is used to connect with an Android phone to control the steering of a robot car. Specifically, the HC-05 Bluetooth module is connected to the RX and TX pins of an Arduino microcontroller. The communication between the Android phone and the HC-05 Bluetooth module is established using an application called Bluetooth RC.

To set up the communication between the Android phone and the HC-05 Bluetooth module, the Bluetooth module must be configured as either a master or a slave device using AT commands. The Android phone must also have Bluetooth enabled and scan for available devices. The HC-05 Bluetooth module should appear in the list of available devices, and the phone can be paired with the module by providing the correct pin code or passkey.

Once the communication between the Android phone and the HC-05 Bluetooth module is established, the mobile application can be used to control the robot car's steering. The application allows the user to send commands to the Bluetooth module, which in turn sends signals to the Arduino microcontroller. The microcontroller interprets these signals and controls the robot car's steering accordingly.

Overall, using a Bluetooth module to connect with an Android phone to control a robot car's steering is a relatively straightforward process. By properly configuring the HC-05 Bluetooth module and using an appropriate mobile application, the user can remotely control the robot car's steering with ease.

5.2 Automatic Mode

In this design it is possible to control the steering of a robot car by connecting it with an Android phone using a Bluetooth module, and the specific way of connecting an Arduino microcontroller with a HC05 Bluetooth module is through its RX and TX pins. The establishment of communication between the HC05 Bluetooth module and Android phone requires an application known as Bluetooth RC .

Configuring the HC -05 Bluetooth module as a master or slave device through AT commands is crucial for establishing communication between an android phone and it, so in order to search for any available device the android phone needs an enabled Bluetooth and scanning capability. Find and select HC -05 Bluetooth Module from available device list & provide correct pin code/passkey in order to pair it with your phone.

When there is a connection established between Android phone and HC 05 Bluetooth module, it becomes feasible to use mobile application for controlling robot car's steering and users of this application are able to issue commands through Bluetooth module which will then transmit signals over to Arduino microcontroller. Controlling the robot car's steering appropriately requires interpretation of these signals by the microcontroller.

In general terms, the process that you follow when you want to use your Android device and Bluetooth module for controlling the direction of your robotic vehicle is quite straightforward and controlling the steering of the robot car from a distance is made simple by using an appropriate mobile app and configuring the HC05 Bluetooth module correctly.

6. Conclusion

A Bluetooth robot car capable of navigating a given space while avoiding obstacles is designed and implemented in this paper. A reliable communication link between the controller and the car is established by the Bluetooth communication algorithm while the obstacle detection and avoidance algorithm uses a proportional-derivative controller. Researchers and engineers can use the successful implementation of the Bluetooth robot car as a valuable reference for creating innovative and efficient solutions in various fields through the integration of IoT and robotics. The obstacle detection and avoidance algorithm can be improved in the future by integrating machine learning techniques. Research could be carried out to develop more effective and dependable communication protocols for IoT and robotics applications, as well. The possibility of enabling the robot car to navigate and avoid obstacles without human input through autonomous operation could finally be explored.

References

[1] Prof. Swati Khawate, Komal Prajapati, Yash Anand, & Kakshil Chandodwala. (2022). Voice Controlled Robotic Car using Arduino. International Journal of Scientific Research in Science, Engineering and Technology, 119–129. https://doi.org/10.32628/ijsrset122934

[2] Vieira, R., Argento, E., & Revoredo, T. (2022). Trajectory Planning For Car-like Robots Through Curve Parametrization And Genetic Algorithm Optimization With Applications To Autonomous Parking. IEEE Latin America Transactions, 20(2), 309–316. https://doi.org/10.1109/tla.2022.9661471

[3] M, P., & K, D. S. D. (2023). ICN Scheme and Proxy re-encryption for Privacy Data Sharing on the Block Chain. *International Journal of Computer Engineering in Research Trends*, *10*(4), 172–176.

[4] Chachin, P. (2017). IoT in transportation: connected and autonomous cars. ELECTRONICS: Science, Technology, Business, (8), 154–158. https://doi.org/10.22184/1992-4178.2017.169.8.154.158

[5] Lu, C. H., & Hu, Y. L. (2013). Speech Control System for Robot Based on Raspberry Pi. Advanced Materials Research, 791–793, 663–667. https://doi.org/10.4028/www.scientific.net/amr.791-793.663

[6] Yeom, K. (2022). Design of Deep Neural Network Based Model Predictive Controller for a Car-like Mobile Robot. International Journal of Mechanical Engineering and Robotics Research, 606–613. https://doi.org/10.18178/ijmerr.11.8.606-613

[7] Obstacle Detection and Avoidance Methods for Autonomous Mobile Robot. (2016). International Journal of Science and Research (IJSR), 5(1), 1245–1249. https://doi.org/10.21275/v5i1.nov152937

[8] Baldovino, R. G., & Dadios, E. P. (2017). A Fuzzy Proportional-Derivative (PD) Algorithm for Programmable Logic Controller (PLC). Advanced Science Letters, 23(11), 11400–11403. https://doi.org/10.1166/asl.2017.10292

[9] Rudra Kumar, M., Rashmi Pathak, and Vinit Kumar Gunjan. "Machine Learning-Based Project Resource Allocation Fitment Analysis System (ML-PRAFS)." *Computational Intelligence in Machine Learning: Select Proceedings of ICCIML 2021*. Singapore: Springer Nature Singapore, 2022. 1-14.

[10] Manikandan, G., & Srinivasan, S. (2012). Traffic Control by Bluetooth Enabled Mobile Phone. International Journal of Computer and Communication Engineering, 66–69. https://doi.org/10.7763/ijcce.2012.v1.19

[11] Suneel, Chenna Venkata, K. Prasanna, and M. Rudra Kumar. "Frequent data partitioning using parallel mining item sets and MapReduce." *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* 2.4 (2017).

[12] M. M. Venkata Chalapathi, M. Rudra Kumar, Neeraj Sharma, S. Shitharth, "Ensemble Learning by High-Dimensional Acoustic Features for Emotion Recognition from Speech Audio

Signal", Security and Communication Networks, vol. 2022, Article ID 8777026, 10 pages, 2022. https://doi.org/10.1155/2022/8777026

[13] M, P., & K, D. S. D. (2023). ICN Scheme and Proxy re-encryption for Privacy Data Sharing on the Block Chain. *International Journal of Computer Engineering in Research Trends*, *10*(4), 172–176. https://doi.org/10.22362/ijcertpublications.v10i4.11

[14] Benaziza, W., Mallem, A., & Slimane, N. (2018). PD Terminal Sliding Mode Control Using Fuzzy Genetic Algorithm for Mobile Robot in Presence of Disturbances. Journal of Automation, Mobile Robotics and Intelligent Systems, 12(2), 52–60. https://doi.org/10.14313/jamris_2-2018/11

[15] B. Swathi, M. R. Kumar, M. Riyaz Belgaum and Z. Alansari, "Emotion Recognition Using Bayesian Learning from a Multi-Label Data Corpus," *2022 International Conference on Computing, Networking, Telecommunications & Engineering Sciences Applications (CoNTESA)*, Skopje, North Macedonia, 2022, pp. 17-22, doi: 10.1109/CoNTESA57046.2022.10011330

[16] A, M., & G.V, S. (2017). Zigbee Transmitter for IoT Wireless Devices. International Journal of VLSI Design & Communication Systems, 8(5), 1–13. https://doi.org/10.5121/vlsic.2017.8501

[17] Andersen, R., Brogaard, R., & Boukas, E. (2023). Autonomous Robotic Inspection for Remote Inspection Technique Systems: A Review. Field Robotics, 3(1), 69–96. https://doi.org/10.55417/fr.2023002

[18] Bezawada , M., & P, V. K. (2023). Comparative Study on Techniques Used for Anomaly Detection in IoT Data. *International Journal of Computer Engineering in Research Trends*, *10*(4), 177–181. https://doi.org/10.22362/ijcertpublications.v10i4.12