2024 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) | 979-8-3503-4846-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/SCEECS61402.2024.10482083

Mean-Based Approach To Omnidirectional Pathfinding Algorithm Using Single Ultrasonic Sensor For Robotic Car

Anvit Tandon
UG Student
Department of Computer Science and Engineering
IIIT Bhopal
anvittandon0808@gmail.com

Shubham Sah
UG Student
Department of Computer Science and Engineering
IIIT Bhopal
shubhamsah869@gmail.com

Suresh Kumar Gawre
Assistant Professor
Department of Electrical engineering
MANIT Bhopal
sgawre28@gmail.com

***Abstract*—Our project centers around creating and practically testing a pathfinding algorithm using a single ultrasonic sensor on a test RC car. The algorithm consists of two phases: normal and intensive detection. This algorithm provides a reliable method for determining an obstacle-free path. The intensive detection method uses a servo motor for a broader field of view, thus creating an omnidirectional composite sensor with just one monodirectional sensor. It reduces the number of potential false positives for the ideal path by a mean-based approach.**

***Index Terms*—Arduino UNO, DC Motors, H-W 130 Motor Shield, Servo Motor, Ultrasonic Sensor.**

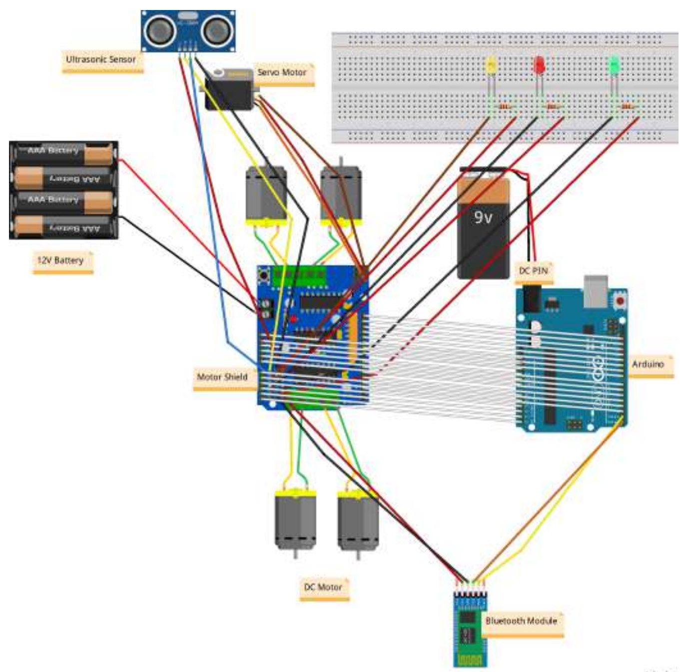

Fig. 1. Circuit Diagram

I. Introduction

A. Background:

In this modern world, pathfinding algorithms play a vital role in robotics and automation. However, Commonly used pathfinding algorithms for ultrasonic sensors are limited by the monodirectional nature of the sensor. They often rely on simple binary directional decisions (left and right), which is inaccurate [1]. This paper aims to enhance the existing algorithms.

These algorithms are crucial for finding paths and avoiding obstacles that come the car's way. They have various applications in the military to reach inaccessible or dangerous areas. Moreover, they have applications in obstacle avoidance systems on extraterrestrial surfaces with an atmosphere.

B. Motivation

The motivation behind this project stems from the limitations of the monodirectional nature of the ultrasonic sensor. This does not provide an accurate way of finding the ideal path as it gives false positives (detailed in the Algorithm section). While other wide-angle methods exist, they tend to be expensive and complicated. This motivated us to create a composite sensor using an ultrasonic sensor mounted on a servo motor. We formulated an algorithm for omnidirectional pathfinding and obstacle detection alongside this sensor.

C. Paper Organization

In the following section, we'll give a basic overview of our project, Section II tells us about the design and hardware requirements. Subsequently, Section III tells us about the algorithm and its working principle. Section IV will describe the software requirement and control mechanism, emphasizing their important role.

To conclude our discussion, Section V tells us about the advantages, real-world applications, and future potential of the project, meanwhile, Section VI will provide a detailed summary of our discoveries and inputs. Refer to Figures 12 and 13, showing the block diagram and the final model, respectively. Our overall methodology revolves around omnidirectional pathfinding using a mean-based algorithm, utilizing a single ultrasonic sensor.

Fig. 2. Chassis, including motors and wheels

Fig. 3. Arduino UNO

Fig. 4. Bluetooth Module HC-05

II. DESIGN AND HARDWARE

A. CIRCUIT

- The Arduino UNO is connected to the motor shield either by mounting it or by using a jumper cables. The four DC motors are connected to an external power supply. Remove the jumper on the motor shield to prevent damaging the Arduino and motor shield.
- The HC-05 Bluetooth module is connected to the Arduino. The RX pin of the Arduino is connected to the TX pin of HC-05, and vice-versa. The HC-05 is connected to the 5V and ground pins.
- The HC-SR04 Ultrasonic sensor is attached to the Arduino, with the Trig pin connected to the A0 pin and the Echo pin to the A1 pin of the Arduino. The ultrasonic sensor is attached to the 5V and ground pins.
- The servo motor is connected to the servo pins provided in the motor shield.
- A 9V battery of cells is used to power the Arduino. (Fig. 1)

B. Components

1) Chassis, including motors and wheels: The car's body is made of a metal chassis, and four DC motors attached to the four wheels that will propel the vehicle. (Fig. 2)

2) Arduino UNO: The Arduino UNO is an open-source microcontroller board based on the Microchip ATmega328P microcontroller, developed by Arduino. Cc. The board has a set of pins that may be interfaced with various expansion boards and other circuits. (Fig. 3)

3) Bluetooth Module HC-05: HC-05 Wireless Serial Bluetooth Module can be used with any microcontroller. It utilizes a universal asynchronous receiver/ transmitter (UART) for wireless data transfer and can connect to most phones and computers using Bluetooth technology.[2] (Fig. 4)

4) Ultrasonic sensor HC-SR04: An ultrasonic sensor is an electronic device that measures the distance or detects an object. The ultrasonic sensor has a sender and a receiver. The sender emits ultrasonic waves, and the receiver receives the waves. In our case, the transmitted ultrasonic wave propagates through the given medium, air, and it returns to the receiver after hitting an object. The Arduino calculates the time taken by the ultrasonic wave to reach the receiver from the sender. (Fig. 5)

5) HW-130 Motor shield: The HW-130 Motor shield is based on the L293D motor driver IC and is normally used with Arduino UNO and Arduino MEGA. It is a motor driver shield used to drive more than two and less than six motors. In our case, we have the use of four motors, and it has some sets of pins available for servo motors.[3] (Fig. 6)

6) Jumper wires: A jumper wire is used in electrical works, with a connector or pin at each end and some tinned. It can be

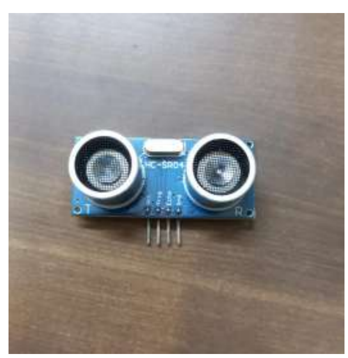

Fig. 5. Ultrasonic Distance HC-SR04

Fig. 6. HW-130 Motor shield

Fig. 7. Jumper wires

used to connect the components of the breadboard breadboard internally or with other equipment or components without soldering. (Fig. 7)

7) Servo Motor: A servo motor is a type of motor that can turn with an excellent level of precision. It usually consists of three parts: a controlled device that controls the motors, an output sensor, and a feedback system. (Fig. 8)

III. ALGORITHM AND WORKING

A. Algorithm

1) Normal Detection Method: The normal detection method uses the ultrasonic sensor to actively measure the distance between the car and the nearest object in its line of sight. This detection method is not omnidirectional but is the most commonly used one [4], and it does not use the servo motor.

2) Intensive Detection Method: The intensive detection method uses the servo motor to move the ultrasonic sensor to get a 180-degree view. In this omnidirectional detection method, the ultrasonic sensor starts by rotating the sensor to the extreme left, 0 degrees. Hereafter, it turns to the extreme

Fig. 8. Servo Motor

right in 3-degree steps, that is, to 180 degrees. At each step, a new reading is taken and associated with the degree of the sensor turn. Once all 60 readings (180 degrees divided by 3-degree steps) are obtained, the sensor is returned to its original position. This collected data is passed on to the analyzing function to determine the further plan of action.

3) Analyzing Function: The analyzing function accepts the data collected by the intensive detection method. This data is represented by a 1-D array of distance values stored as the long integer data type. This paper refers to this array as the distance map. A conventional approach would be to select the maximum distance value and move in the appropriate direction. However, in our implementation of this approach, several issues emerged. A single maximum value does not ensure that the car will fit through the gap (Fig. 9). It leaves the possibility for false positives. To address this issue, we developed a mean-based algorithm that determines the desired movement. Firstly, the algorithm calculates the minimum angle (defined as angle 'a') required to guarantee the successful passage of the car. Given the predefined minimum obstacle distance (defined as length 'y') for detection and car dimensions(width defined as length 'x'), the algorithm calculates angle 'a' (Fig. 10). Once all the required calculations are complete, the algorithm calculates the mean of every consecutive cluster of values. The length of each cluster is static and is determined by the angle 'a'. The angle with the maximum mean value is chosen as the ideal moving space. This dramatically reduces the number of false positives as the mean of the distances would not meet the threshold (Fig. 11).

1) While an object is not detected in the direct line of view, the algorithm does not give the command to stop the car. It continues to navigate its path directly based on the user's command. Further, it continues to detect the obstacles using the normal detection method.
2) In case an obstacle is detected by the normal detection method, it stops the motors and moves backward briefly.

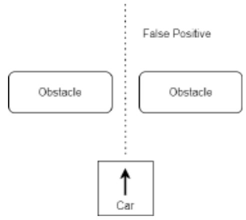

Fig. 9. False Positive

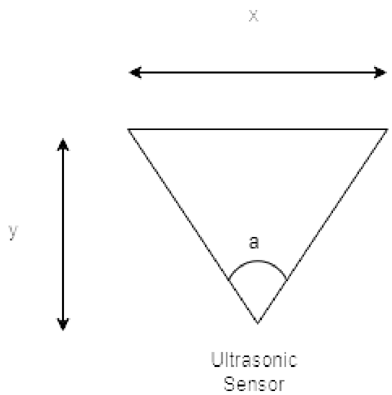

Fig. 10. Calculation of angle

The detection method is elevated to the intensive detection method. Once the ideal direction is determined, the car is turned and moved as required.

3) The detection method is de-escalated to the normal detection method.
4) The car resumes the execution of direct user commands. The algorithm loops back to step 1.

IV. SOFTWARE REQUIREMENT

A. Arduino IDE

The Arduino Integrated Development Environment (IDE) is a cross-stage application for Windows, macOS, and Linux that is; it uses a simplified programming language based on C and C++. It is an open source Software. The IDE includes a serial monitor that enables developers to communicate with and debug their Arduino projects by sending and receiving data between the computer and the Arduino board. This is particularly useful for troubleshooting and real-time data monitoring.

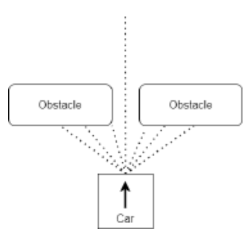

Fig. 11. Intensive scanning

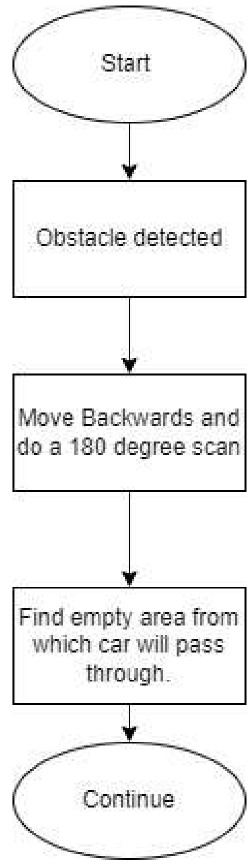

Fig. 12. Block Diagram

B. To Control The Robotic Car

- Send the command using a mobile app to communicate with Arduino.
- This app will act as a controller by sending control signals to the car via Bluetooth. [5]
- Commands are entered in this Android application.[6]

V. ADVANTAGES AND FUTURE SCOPE

Some real-world applications of this robot car are:

1) The robot can reach those places where humans find it difficult to reach, such as in fire situations or highly toxic areas, as seen in the Chernobyl power plant incident. [7]
2) The robotic car can be easily driven by anyone by using commands with the help of an Android application on a smartphone.
3) The robot can be utilized for remote monitoring or investigation.
4) We can further calibrate the car for different surfaces by adjusting the motor speeds and changing the predefined set of values.
5) We can add cameras to use face detection algorithms.
6) We can add mini computers like Rasberry Pi to further increase the car's functionality.

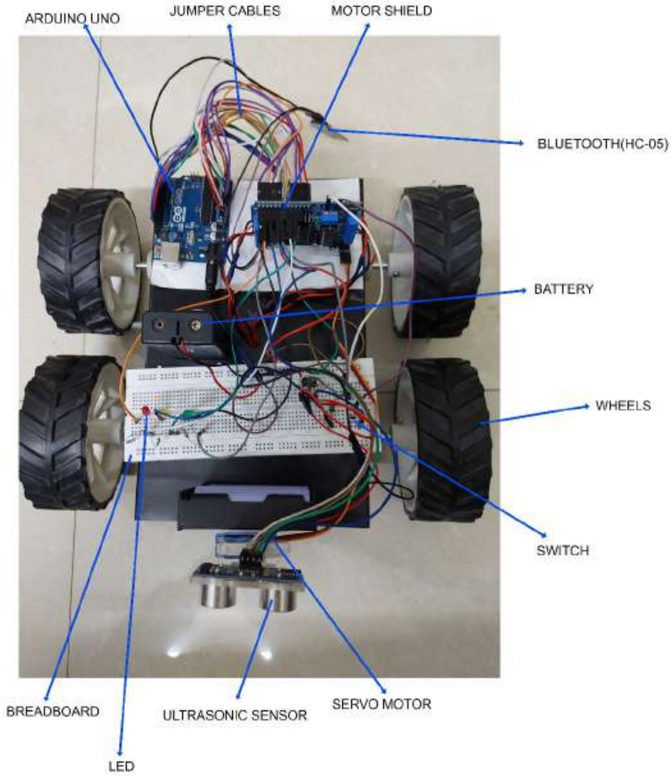

Fig. 13. Final Model

7) We can introduce a Thermal camera to detect the warmth produced by bodies, which would be valuable for military purposes to distinguish foes on the lines.

VI. Conclusion

Considering all the aforementioned points, we have practically tested and demonstrated better pathfinding and obstacle-avoiding algorithms on a test model car. This algorithm can detect and find its path with a higher level of precision, as tested practically. The dual-phase nature of the algorithm allows for a low power consumption implementation while still providing an accurate and reliable ideal path. It can be used in military applications, research endeavors, and places where human interference may be dangerous.

References

[1] T. Mahmud et al., “Design and Implementation of an Ultrasonic Sensor-Based Obstacle Avoidance System for Arduino Robots,” 2023 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD), Dhaka, Bangladesh, 2023, pp. 264-268, https://doi.org/10.1109/ICICT4SD59951.2023.10303550.

[2] “Bluetooth Module HC-05 — Sensors & Modules,” https://www.electronicwings.com/sensors-modules/bluetooth-module-hc-05.

[3] “Control DC, Stepper & Servo with L293D Motor Driver Shield & Arduino,” Last Minute Engineers, Dec. 17, 2018. https://lastminuteengineers.com/l293d-motor-driver-shield-arduino-tutorial.

[4] L. Chen, J. Zhang and Y. Wang, “Wireless Car Control System Based on ARDUINO UNO R3,” 2018 2nd IEEE Advanced Information Management, Communications, Electronic and Automation Control Conference (IMCEC), Xi’an, China, 2018, pp. 1783-1787, https://doi.org/10.1109/IMCEC.2018.8469286.

[5] R. Sissodia, M. S. Rauthan and V. Barthwal, “Arduino Based Bluetooth Voice-Controlled Robot Car and Obstacle Detector,” 2023 IEEE International Students’ Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 2023, pp. 1-5, https://doi.org/10.1109/SCEECS57921.2023.10063092.

[6] AnvitT, “SonicRover,” GitHub, Oct. 26, 2023. https://github.com/AnvitT/SonicRover (accessed Dec. 30, 2023).

[7] T. Akilan, S. Chaudhary, P. Kumari and U. Pandey, “Surveillance Robot in Hazardous Place Using IoT Technology,” 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 2020, pp. 775-780, https://doi.org/10.1109/ICACCCN51052.2020.9362813.

[8] Fritzing, “Fritzing,” https://fritzing.org.

[9] Draw.io, “Flowchart Maker & Online Diagram Software,” https://app.diagrams.net.