Design of obstacle avoiding car

Darmenkyzy Nurbakyt
Department of Computer Engineering
Astana IT University
Astana, Kazakhstan
211379@astanait.edu.kz

Amangeldi Aruzhan
Department of Computer Engineering
Astana IT University
Astana, Kazakhstan
211417@astanait.edu.kz

Abu Mariyam
Department of Computer Engineering
Astana IT University
Astana, Kazakhstan
211293@astanait.edu.kz

***Abstract*—In this research, the focus is on designing a robot-car using the Arduino, which is an open-source platform used for constructing and programming of electronics [1]. This study explores the design of a small mobile machine that uses an ultrasonic sensor to measure object distance and avoids obstacles. When the car is within a specific distance of an obstacle, the Arduino UNO will send a signal to the engines to turn the car using a servo motor. The servo motor is a rotary actuator which is used for applications that require a degree for the rotation angle [2]. The study also explores the full process of construction of a car taking advantage of all the components required for this robot. Different hardware and software requirements for the robot's functioning are detailed throughout the document. The proposed design is shown, followed by the findings and conclusions.**

***Index Terms*—Arduino Uno, obstacle detection and avoidance, mobile robot, ultrasonic sensor, motor shield.**

I. Introduction

Robotics was encouraged with the further development of human exploration activities and almost all industries now depend heavily on robots, which have cut down on human meddling by about 50%. Robots are machines that can perform a variety of tasks accurately and repeatedly in a wide range of potential applications as well as challenging and life-threatening occupations.

Robotics can frequently have issues avoiding obstacles. Brick walls or other barriers could be obstacles for robots. So, obstacle avoidance is crucial for robots that move through unfamiliar areas. There are numerous techniques for achieving obstacle avoidance and one of common is sensor-based obstacle detection. With this technique, the robot's movement can be controlled using the sensors. In this paper, we propose a design for an obstacle-avoiding car based on an Arduino Uno, an ultrasonic sensor, and a blue motor shield. By avoiding obstacles, the car can drive itself in an unknown environment.

The obstacle-avoiding car is composed of the following components:

Arduino Uno: The Arduino Uno is a microcontroller board that is used to control the car's movement.

Ultrasonic sensor: The ultrasonic sensor is used to detect obstacles in front of the car.

Motor shield: The motor shield is used to control the car's motors.

The obstacle-avoiding car works in the following way:

The ultrasonic sensor detects obstacles in front of the car.

The Arduino Uno calculates the distance to the obstacles.

If the distance to any of the obstacles is less than or equal to a certain threshold, the Arduino Uno commands the motors to turn the car.

The car continues to move forward until it detects no obstacles in front of it.

The obstacle-avoiding car is a simple and effective way to create a robot that can autonomously navigate in an unknown environment. The car can be used for a variety of applications, such as surveillance, delivery, and exploration.

In addition to the components mentioned above, the obstacle-avoiding car can also be equipped with other sensors, such as cameras, to improve its obstacle detection and avoidance capabilities. The car can also be programmed to follow a specific path or to respond to certain commands.

The obstacle-avoiding car is a low-cost and easy-to-build platform that can be used to learn about the principles of obstacle avoidance and to develop new obstacle avoidance algorithms.

II. Literature Review

For mobile robots that must move in unfamiliar or dynamic environments, obstacle avoidance is a crucial ability. Different obstacle avoidance strategies have been developed, each having pros and cons of their own.

The technique of wall following is one of the earliest methods of obstacle avoidance. In the wall following, the robot moves in a straight line against the wall after detecting it with sensors. Although this method is simple to use, it might be slow and ineffective in situations with lots of obstacles. About this type of technique, in his article Turennout [3] described that when the sensors can not provide an overview of the shape or the size of the obstacle, it is not possible to plan an evasive route. So it becomes necessary to follow the contour of the obstacle until such is possible or until the original route can be resumed.

About another common obstacle avoidance technique Koskinen mentioned in his book [4], a sensor fusion-based obstacle avoidance technique combines data from multiple sensors to create a more accurate representation of the environment. This allows robots to use more complex obstacle avoidance techniques, such as potential field and path planning. Path planning from point A to B, being able to recognize obstacles and at least stop before a danger are common requirements for mobile robots and autonomous vehicles.

Potential field is a popular obstacle avoidance technique that uses a repulsive force to push the robot away from obstacles and an attractive force to pull the robot towards its goal. In the potential field approach, obstacles are assumed to carry electric charges, and the resulting scalar potential field is used to represent the free space. Collisions between the obstacles and the robot are avoided by a repulsive force between them, which is simply the negative gradient of the potential field. [5]

Path planning is a more complex obstacle avoidance technique that involves planning a path from the robot's current location to its goal. It is taking care of the generation of obstacle free paths taking into consideration geometric characteristics of obstacles and the kinematic constraints of the robot. This technique can be used to avoid obstacles that are not detected by the robot's sensors. [6]

III. Theoretical Background

The project's goal was to build a car that could navigate unfamiliar or dynamic environments by avoiding obstacles. As a central controller for this project, the Arduino platform, an open-source electronics platform, was used.

Required components such an ultrasonic sensor, servo motor, blue motor shield, wheels, and a 6V battery were included along with to the Arduino. These elements were successfully combined to create a functioning obstacle avoidance car that could make decisions about its location in real time using data collected by the ultrasonic sensor.

A. Arduino Board

The Arduino board is a flexible and well-known microcontroller platform that allows users to build interactive projects involving electronics. It is made up of many main components and functions on the input, processing, and output concept. As shown in Figure 1.

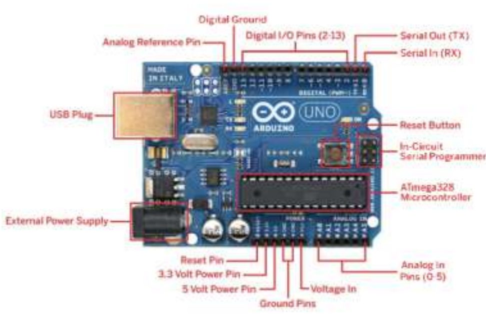

Fig. 1. Arduino Uno Board [1]

The ATmega328 microprocessor at the heart of the Arduino board is in charge of executing program instructions and controlling the board's functionality.

The following digital input/output (I/O) pins are used to read digital signals and control external devices. The Arduino programming environment allows you to set up these pins as input or output.

Analog input ports on Arduino boards allow the measurement of analog signals such as sensor readings or variable voltages. These pins transform analog signals into digital values that the microcontroller can process.

Arduino boards can be powered by a variety of methods, such as USB connections, external power supplies, or batteries. Voltage regulators are often used to maintain a stable power supply for the components.

The Arduino board operates on a basic input, processing, and output model. [1] First, users can attach many different sensors, switches, or other devices to the Arduino board's input pins. These sensors detect changes in the physical or environmental environment and turn them into electrical signals. To obtain data about the external world, the microcontroller examines these input signals, which might be digital or analog.

The microcontroller then processes the input data in accordance with the program instructions uploaded to the device. Arduino programs, written in a simplified form of the C++ programming language, control the microcontroller's behavior and capability. The logic of the program specifies how the input data should be processed, analyzed, and responded to. The microcontroller generates output signals after processing the input data, which operate external devices or interact with the environment. This includes operating motors, LED lights, transmitting signals to displays, and interfacing with other devices via serial connection or wireless modules.

The Arduino programming environment has a user-friendly interface that makes creating and uploading code to the board easier. This enables users, even those with no programming experience, to fully utilize the Arduino board's capabilities and construct a wide range of interactive projects.

B. Motor Shield

The Arduino Motor Shield L293D is a popular motor driver shield that is created exclusively for Arduino boards. It is an essential component in robotics and automation tasks because it provides a quick and convenient approach to operate DC motors and stepper motors. As shown in Figure 2.

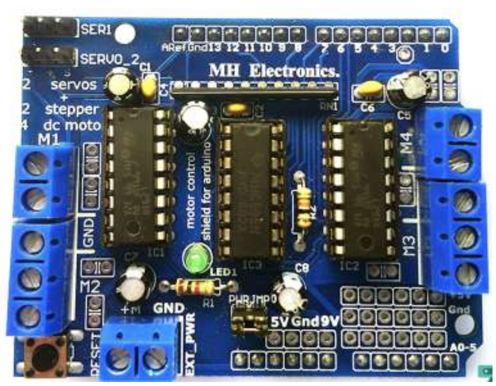

Fig. 2. L293D Motor Shield

And it has following features:

1. It can control up to four DC motors or two stepper motors at the same time. It gives each motor independent control over its speed and direction.

2. The L293D chip employs H-bridge circuitry, allowing bidirectional motor control. This means that simply altering the direction of the current flow, you can easily control the motors to move forward, backward, or halt.

3. The motor shield includes a separate power input terminal for connecting an external power source to the motors. This guarantees that sufficient power is sent to the motors without overloading the Arduino board.

4. The motor shield is designed to stack on top of an Arduino board, allowing for a simple plug-and-play connection. It makes use of standard Arduino headers, making it compatible with a variety of Arduino boards such as the Uno, Mega, and others.

5. It can also be utilized for an extended period of time after the program has been uploaded to the Arduino board. It will not be reset until a new set of commands is uploaded.

Also Pandey [8] in his work mentioned some advantages of this motor shield. They are easily available in the local market and very cost effective. Operation of Arduino shield is completely based on embedded C language coding, therefore, coding is independent of the operating platform.

C. Ultrasonic Sensor

Sound is used by ultrasonic sensors to detect the distance between the sensor and the closest object in its path. Ultrasonic sensors are similar to sound sensors, except they operate at a higher frequency than human hearing.

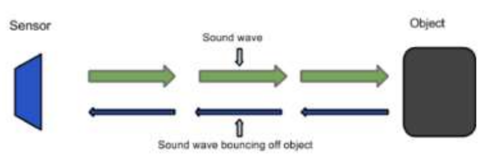

Fig. 3. Illustration of sound wave to bounce off of an object and come back [9]

A sound wave at a specified frequency is sent out by the sensor. It then waits for that exact sound wave to bounce off an item and return, as illustrated in Figure 3. The sensor measures the time between sending the sound wave and receiving it. Equation 1 can be used to calculate the distance traveled if you know how fast something is moving and how long it is traveling. [9]

$$d = v * t \tag{1}$$

To convert the Equation 1 to a distance, we need the speed of sound at standard temperature and pressure, which is 343 meters per second. And the time is mostly in microseconds or less. So, we use Equation 2 to convert meters per second to microseconds per centimeter and Equation 3 to calculate the distance in centimeters. [9]

$$D = \frac{Speed}{170.15m} \times \frac{Meters}{100cm} \times \frac{1e6\mu S}{170.15cm} \times \frac{58.772\mu S}{cm} \tag{2}$$

$$D = \frac{time}{58} = \frac{\mu s}{\mu s/cm} = cm \tag{3}$$

The HC-SR04 consists of four pins: VCC, GND, TRIG, and ECHO, each with a different function. The VCC and GND pins are the most basic they supply power to the HCSR04. These pins must be connected to a +5 volt supply and ground, respectively. The TRIG pin is the only control pin. The ultrasonic blast is transmitted through the TRIG pin. When this pin is set to HIGH for 10 seconds, the HC-SR04 will send off an eight cycle sonic burst at 40 kHz.

Fig. 4. Ultrasonic sensor HC-SR04.

The ECHO pin will go HIGH after a sonic blast is sent. The ECHO pin is a data pin that is used to measure distances. When an ultrasonic burst is sent, the pin becomes HIGH and stays high until another ultrasonic burst is detected, at which point it goes LOW.

Additionally, ultrasonic sensors consider a cone of detection, and cone's angle changes with distance. The orientation of the object in relation to the sensor affects a sensor's capacity to detect it. It is possible that the sound wave will bounce off an item so that it does not return to the sensor if the object does not present the sensor with a flat surface.

D. Servo Motor

Fig. 5. Servo Motor SG90

A servo motor is a popular micro servo motor that is commonly used in robotics that require precise control of rotational or angular movements and that moves an object at a fixed angle. It is a small, lightweight motor that provides relatively high torque for its size.

The SG90 servo motor has attracted interest among people in the field of robotics and automation due to its small size, low cost, and ease of use. It is frequently used to operate robot limbs, steering mechanisms in remote-controlled cars, camera pan-tilt systems, and a variety of other applications that demand precise and controlled motion.

The width of a pulsed signal corresponding to the angle at which the servo motor is rotated precisely controls the servo motor. The SG90 servo rotates to angles ranging from 0° to 180° when provided signals with pulse widths ranging from 0.5ms to 2.5ms and pulse intervals of 20ms.

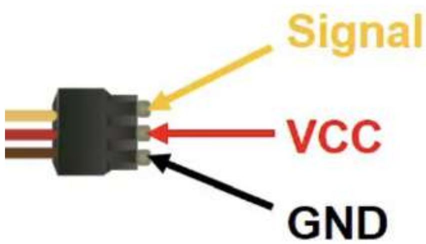

Fig. 6. Servo Motor SG90 connections

As shown in Figure 6, the servo motor has three connections: red for power, brown or black for ground, and orange or white for signal. [10] A servo motor operates at 5V and can consume hundreds of milliamps while the rotor is moving, which is greater than the 40mA maximum output of the Arduino pins.

On both ends of the voltage regulator, decoupling capacitors smooth both the voltage supply and the voltage demand.

IV. Experiments

A. Constructing a car

To conduct tests on our robot, we must first establish the processes for the machine's design. We used and implemented the following steps during the construction process.

a) Put the chassis together: Creating a platform for the car by building a base construction. Attaching the two wheels to the motors, making sure they are properly aligned and securely fastened. Shown in Figure 7.

b) Connect the motors and wheels to the Motor Shield: Placing the Motor Shield L293D on top of the Arduino UNO Board, making sure that the pins are properly aligned. Then connecting the motor wires to the Motor Shield. Each motor's red wire should be connected to the "+" terminal, while the black wire should be connected to the "-" terminal.

c) Connect the Ultrasonic sensor: Connecting the Ultrasonic sensor's VCC pin to the Arduino UNO Board's 5V pin. And connecting the GND (ground) pin to the Arduino UNO Board's GND pin. Then, connect the Trig pin to an A0, and Echo pin to an A1 digital pin on the Arduino UNO Board.

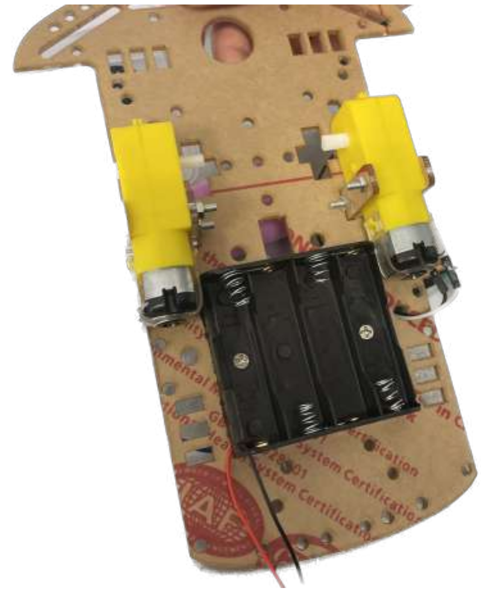

Fig. 7. Putting together the chassis

d) Connect the servo motor: Connecting the VCC pin of the servo motor to the 5V pin, and the GND (ground) pin of the servo motor to the GND pin on the Arduino Uno Board. Then connect the control pin of the servo motor to a digital pin on the board.

e) Power the Arduino and Motor Shield: Inserting the 4 batteries with 1.5V into a battery holder. And connecting the positive and negative terminal of the battery holder to the VIN pin on the Motor Shield. Shown in Figure 8.

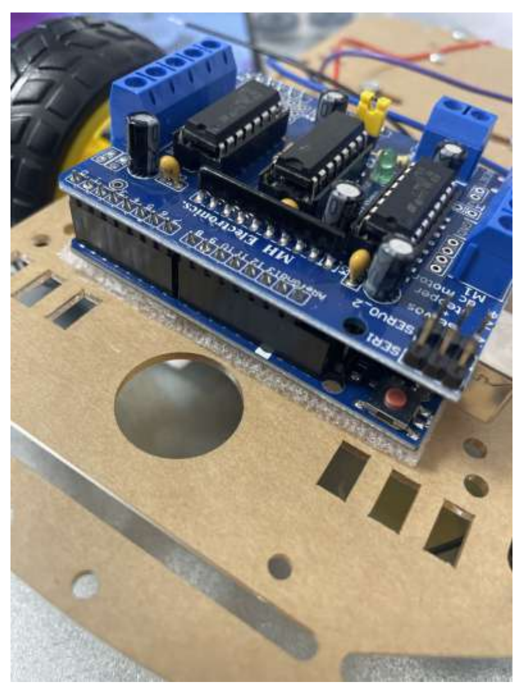

Fig. 8. Connecting Arduino Board and Motor Shield

f) Upload the code: Opening the Arduino IDE on the computer and connecting the Arduino Uno Board to the computer using a USB-A USB-B cable. Uploading a code using Arduino IDE to the car.

As a result, we get a car with 2 wheels and 2 eyes(ultrasonic sensor). Result is shown in Figure 9.

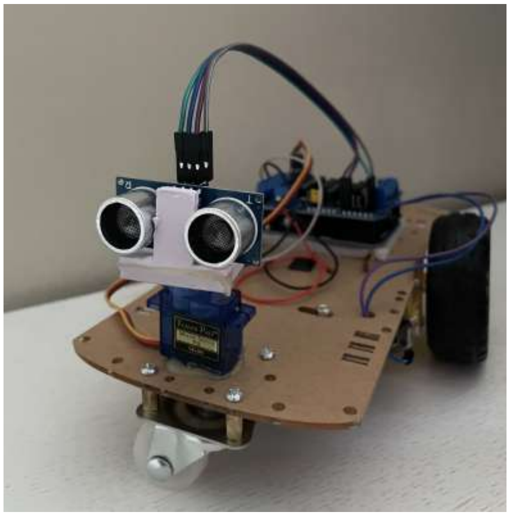

Fig. 9. Final result of the car

B. Obstacle Avoidance Algorithm

The ultrasonic sensor is used by the Arduino robot car's obstacle avoidance algorithm to identify obstacles in front of the car. A sound wave is released by the ultrasonic sensor, which then tracks how long it takes for the wave to return. The speed of sound is then used to determine the distance to the obstacle.

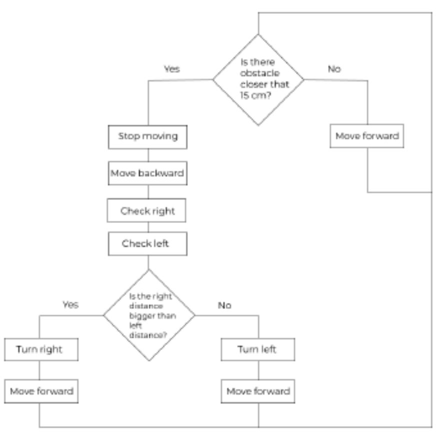

Fig. 10. Flow chart of obstacle avoidance robot

The algorithm works in the following steps:

The servo motor is turned to the right and the ultrasonic sensor is used to measure the distance to the obstacle on the right side. The servo motor is turned to the left and the ultrasonic sensor is used to measure the distance to the obstacle on the left side. The distance to the obstacle in front is measured. If the distance to any of the obstacles is less than or equal to 15 centimeters, the robot will stop moving. The robot will then move backwards for 300 milliseconds. The robot will then turn to the right or left, depending on which side of the obstacle is closer. The robot will then resume moving forward.

C. Testing a car

The car demonstrated its capabilities and validated the successful integration of the components during these testing. The car demonstrated consistent forward and backward movement, precise control of the servo motor angles, and effective obstacle detection and avoidance functions. These tests evaluate the built obstacle-avoiding car's functionality and performance. In total, there were performed 3 tests:

1. Checking the move forward and backward functions:

The goal of this test was to validate the car's basic movement capabilities, specifically its forward and backward movement. The test consisted of activating the motor functions using Arduino code and watching the car's response. The car moved forward and backward as intended, suggesting that the motor connections and control were correctly configured.

2. Checking the servo motor for correct functioning:

The servo motor's functioning and range of motion were investigated in this test. The code was run to spin the servo motor at various angles, and the actual position of the servo motor was recorded. It was proven that the servo motor was running correctly and accurately adapting to the appropriate angles by comparing the expected angles with the observed positions. This test ensured that the servo motor could identify and avoid obstructions accurately.

3. Final test with all: check and avoid obstacles, move forward and back, turn left and right functions.

This test was designed to assess the effectiveness of the obstacle detection and avoidance mechanisms. The car's reaction to detecting impediments was observed, particularly how it changed direction to avoid collisions. The test results showed that the car identified obstacles and quickly moved away from them, demonstrating the accuracy of all functionalities.

Finally, the obstacle-avoiding car was successfully tested and demonstrated its capabilities. The car is able to move forward and backward consistently, the servo motor is able to control the car's direction accurately, and the obstacle detection and avoidance mechanisms are effective.

The car can be used for a variety of applications, such as surveillance, delivery, and exploration. It is a valuable tool for researchers and students who are interested in robotics and autonomous systems.

V. Results and Discussion

Using the Arduino platform, a robot automobile capable of obstacle avoidance was created and built in this project. An ultrasonic sensor, servo motor, motor shield, wheels, and a 6V battery were all included in the automobile. Assembling the chassis, connecting the motors and wheels to the motor shield, adding the ultrasonic sensor, connecting the servo motor, and powering the Arduino and motor shield were all part of the construction process.

The functioning and performance of the obstacle-avoiding car were evaluated through tests. The initial test focused on the car's basic mobility skills, such as forward and backward movement. The car moved in the desired directions, demonstrating that the motor connections and control were properly set up.

The servo motor's functionality and range of motion were tested in the second test. The servo motor was rotated using the code at various angles, and the actual positions were recorded. The servo motor's capacity to recognize and avoid obstructions was noticed as it was correctly set to the desired angles and operated.

The effectiveness of the obstacle detection and avoidance methods was then evaluated through an extensive evaluation. It was possible to see how the car responded to obstacles by changing direction in order to avoid collisions. The test results demonstrated that the car correctly detected obstacles and immediately avoided them, confirming the accuracy and dependability of all commands.

During the tests, the constructed robot car successfully showed its ability to avoid obstacles and functioned as expected. For accurate control and effective navigation in unknown environments, the Arduino platform, ultrasonic sensor, servo motor, and motor shield were integrated. The car demonstrated its capacity to autonomously move over unfamiliar environments by avoiding obstacles.

This project's obstacle avoidance technology is built on the idea of ultrasonic ranging. A sound wave is transmitted by the ultrasonic sensor, which then tracks how long it takes for the wave to return. The speed of sound is then used to determine the distance to the obstruction. The ultrasonic sensor is rotated by the servo motor so that it may search the surroundings for impediments.

The Arduino code includes the obstacle avoidance algorithm. The algorithm searches the environment continuously for obstacles. The car stops motors when an obstruction is seen. The car will turn to avoid the obstruction if it is close enough.

This project's obstacle avoidance system is useful for moving through unfamiliar environments. The system does have some disadvantages though. Obstacles that are too far away cannot be detected by the ultrasonic sensor due to its short range. The obstacle avoidance algorithm is also not flawless. On occasion, it might miss an obstacle or steer the automobile in the wrong direction.

The obstacle avoidance mechanism in this project is an important achievement despite these disadvantages. It shows how Arduino may be used to build robots that can independently move through unknown environments.

The obstacle avoidance system could be enhanced in the future through the use of an improved ultrasonic sensor with a longer range. The obstacle avoidance algorithm should also be strengthened to increase its dependability. These advancements would make it possible to build robots that can safely navigate over even the most challenging surfaces.

VI. Arduino Pseudo-Code

Here is the Arduino pseudo code for our robot. An obstacle avoidance algorithm was used and described in this pseudo-code.

```
#include libraries and define
constants and variables
setup():
  attach the servo motor to pin 10
  turn the servo motor to the front
  position read the distance from the
  ping sensor 4 times
loop():
  determine the distance on the right
  and on the left side using the servo
  motor if the distance in front
  is <= 15 centimeters:
    stop moving
    move backwards for 300 milliseconds
    stop moving
    turn the robot to the right if the
    distance on the right is greater
    than or equal to the distance
    on the left, and turn left otherwise
  else:
    move forward
lookRight():
  turn the servo motor to the right
  read the distance from the ping sensor
  turn the servo motor back to the
  front position return the distance
lookLeft():
  turn the servo motor to the left
  read the distance from the ping sensor
  turn the servo motor back to the
  front position return the distance
readPing():
  delay for 70 milliseconds
  use the ping sensor to read the
  distance
  if the distance is zero:
    set the distance to 250 centimeters
  return the distance
moveStop():
  stop both motors
moveForward():
  if the robot is not already
  moving forward:
    set the boolean goesForward
    to true
    move both motors forward
    gradually up to the maximum
```

```
  speed and incrementing by 2
moveBackward():
  set the boolean goesForward
  to false
  move both motors backward
  gradually up to the maximum
  speed and incrementing by 2
turnRight():
  turn the robot to the right by
  doing a half pivot turn
turnLeft():
  turn the robot to the left by
  doing a half pivot turn
```

VII. Conclusion

To conclude, using the Arduino platform, this study successfully designed and built an obstacle-avoiding robot car. An ultrasonic sensor detected obstacles a servo motor controlled navigation, and a motor shield powered the motors. The car demonstrated constant forward and backward functions, accurate servo motor control, and successful obstacle recognition and avoidance in a series of tests.

The study's findings indicate that the proposed design and implementation are possible for developing a small mobile robot capable of going through unusual environment autonomously. The Arduino platform proven to be a dependable and adaptable platform for controlling the robot's functions. The ultrasonic sensor allowed for precision obstacle detection, while the servo motor allowed for exact steering control.

This study advances the field of robotics by providing insights into the design and operation of an obstacle-avoiding robot car. The established system can be improved and expanded to deal with more complicated tasks and conditions. Future development could include adding more sensors for better perception, developing advanced path planning algorithms, and improving the robot car's overall intelligence and autonomy.

The insertion of a camera is one potential area for development. A camera would increase the robot's capacity to avoid impediments and navigate its environment by allowing it to perceive its surroundings in greater detail. A second ultrasonic sensor could also be added as a potential upgrade. A second ultrasonic sensor would allow the robot to identify obstacles over an expanded range, improving its ability to avoid them.

Aside from the previously suggested areas for improvement, there are many of other potential applications for obstacle-avoiding robot automobiles. Robot automobiles, for example, may be used to carry food and supplies to those living in far away or dangerous places. They might also be employed to perform surveys or check out the infrastructure. It's probable that we'll see even more innovative and exciting uses for obstacle-avoiding robot automobiles as technology progresses.

The obstacle-avoiding car can be made even more capable, adaptable, and dependable by making these changes, which could lead to new opportunities for applications in a variety of industries, including surveillance, exploration, logistics, and robotics.

The intelligence and flexibility of the car can be improved by integrating machine learning methods and artificial intelligence (AI) technologies. The car may learn from past mistakes and enhance its ability to recognize and avoid obstacles by being trained with large datasets. The car may be able to make wise decisions in real-time while accounting for environmental changes and unexpected situations thanks to reinforcement learning algorithms.

The durability and mobility of the car can be increased by enhancing its physical design and mechanics. Its capacity for navigating challenging environments can be improved by upgrading the motor system for more power and torque. The car may adjust to uneven surfaces or obstacles with various heights by utilizing a suspension system or changeable height mechanism.

Creating complex path planning algorithms can enhance the navigation and decision-making capabilities of the vehicle. To determine the most effective and obstacle-free pathways, these algorithms can take into account variables including obstacle proximity, path smoothness, and energy efficiency. Simultaneous localization and mapping (SLAM) approaches can help the car make precise maps of its surroundings while navigating.

The car can communicate with other systems or devices by adding wireless communication capabilities like Bluetooth or Wi-Fi. This may make it easier to integrate with centralized control systems, share data remotely, or perform remote control. The car can also access cloud-based resources, communicate with other smart devices, and use real-time data for better decision-making by being connected to the Internet of Things (IoT).

The ability of the car's automation and autonomy to do complex tasks with little assistance from humans may be enhanced. The stability and reactivity of the car can be increased by integrating innovative control systems, such as closed-loop feedback control. Object tracking, voice commands, and gesture recognition are further technologies that can improve how the car interacts with its surroundings and its passengers.

Overall, this research demonstrates the possibilities of Arduino-based robot automobiles for applications such as monitoring exploration, and help in risky and challenging areas. The provided construction serves as a foundation for future robotics breakthroughs and contributes to the continued development of autonomous systems.

VIII. Outlook

The study described above showcases the successful design and construction of an obstacle-avoiding robot car using the Arduino platform. It highlights the effectiveness of utilizing an ultrasonic sensor for obstacle detection and a servo motor for precise steering control. The findings of the study indicate the feasibility of developing a small mobile robot capable of autonomously navigating through challenging environments.

The established system can serve as a starting point for further advancements in the field of robotics. Several areas for improvement and future development have been identified. One potential avenue is the integration of a camera to enhance the robot's perception of its surroundings. This addition would provide detailed visual information and improve obstacle avoidance capabilities. Another suggested enhancement is the incorporation of a second ultrasonic sensor to expand the range of obstacle detection.

Furthermore, there are numerous potential applications for obstacle-avoiding robot cars beyond the scope of this study. These vehicles could be utilized for tasks such as delivering supplies to remote or hazardous locations, conducting surveys, or inspecting infrastructure. As technology progresses, the potential uses for obstacle-avoiding robot cars are likely to become even more innovative and diverse.

To further improve the capabilities of the obstacle-avoiding car, various approaches can be taken. Integrating machine learning methods and artificial intelligence (AI) technologies would enhance the car's intelligence and flexibility. By training the car with large datasets, it could learn from past experiences and improve its ability to recognize and avoid obstacles in real-time. Reinforcement learning algorithms could enable the car to make informed decisions while adapting to changing environmental conditions.

Additionally, enhancing the car's physical design and mechanics would increase its durability and mobility. Upgrading the motor system for more power and torque would improve the car's navigation through challenging terrains. Implementing a suspension system or adjustable height mechanism would allow the car to adapt to uneven surfaces and obstacles of varying heights.

The navigation and decision-making capabilities of the car could be enhanced by developing complex path planning algorithms. These algorithms could consider factors such as obstacle proximity, path smoothness, and energy efficiency to determine optimal and obstacle-free routes. Simultaneous localization and mapping (SLAM) approaches could aid in creating accurate maps of the car's surroundings for effective navigation.

Adding wireless communication capabilities like Bluetooth or Wi-Fi would enable the car to communicate with other systems or devices. This would facilitate integration with centralized control systems, remote data sharing, and even remote control. Connecting the car to the Internet of Things (IoT) would allow it to access cloud-based resources, communicate with other smart devices, and make decisions based on real-time data.

Improving the car's automation and autonomy would enable it to perform complex tasks with minimal human assistance. Innovative control systems, such as closed-loop feedback control, could enhance stability and reactivity. Technologies like object tracking, voice commands, and gesture recognition could further improve the car's interaction with its environment and passengers.

In conclusion, the study on the obstacle-avoiding robot car demonstrates the potential of Arduino-based robotics for various applications, including monitoring, exploration, and assisting in risky and challenging areas. The research provides a foundation for future breakthroughs in the field and contributes to the ongoing development of autonomous systems. By implementing the suggested improvements and exploring the possibilities outlined, obstacle-avoiding robot cars can become more capable, adaptable, and reliable.

References

[1] Y. A. Badamasi, "The working principle of an Arduino," 2014 11th International Conference on Electronics, Computer and Computation (ICECCO), Abuja, Nigeria, 2014, pp. 1-4, doi: 10.1109/ICECCO.2014.6997578.

[2] A. Bhargava and A. Kumar, "Arduino controlled robotic arm," 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2017, pp. 376-380, doi: 10.1109/ICECA.2017.8212837.

[3] P. van Turennout, G. Honderd and L. van Schelven, "Wall-following control of a mobile robot," in Proceedings 1992 IEEE International Conference on Robotics and Automation, Nice, France, 1992 pp. 280-285, doi:10.1109/ROBOT.1992.220250.

[4] Koskinen, S. (2010). Sensor Data Fusion Based Estimation of Tyre-Road Friction to Enhance Collision Avoidance: Dissertation. [Dissertation, Tampere University of Technology (TUT)]. VTT Technical Research Centre of Finland. https://publications.vtt.fi/pdf/publications/2010/P730.pdf

[5] Hwang, Y. K., & Ahuja, N. (1992). A potential field approach to path planning. IEEE transactions on robotics and automation, 8(1), 23-32.

[6] Kunchev, V., Jain, L., Ivancevic, V., & Finn, A. (2006). Path planning and obstacle avoidance for autonomous mobile robots: A review. In Knowledge-Based Intelligent Information and Engineering Systems: 10th International Conference, KES 2006, Bournemouth, UK, October 9-11, 2006. Proceedings, Part II 10 (pp. 537-544). Springer Berlin Heidelberg.

[7] M. Molina, A. Vera, C. Molina and P. Garzon, "Design and Construction of an Obstacle Avoiding Robot Based on Arduino Platform and Programming Too," 2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 2018, pp. 788-791, doi: 10.1109/UEMCON.2018.8796577.

[8] Pandey, A., Kumar, A., Singh, M. K., Marimuthu, N., & Singh, V. (2016). Automation for measuring the surface resistivity of rpc by arduino motor shield. In Proceedings of the DAE-BRNS Symp. on Nucl. Phys (Vol. 61, p. 1060).

[9] Morgan, Elijah J. HC-SR04 ultrasonic sensor. Nov (2014).

[10] Cameron, N., & Cameron, N. (2019). Servo and Stepper Motors. Arduino Applied: Comprehensive Projects for Everyday Electronics, 157-176.