DEMO VIDEO LINK:

https://youtu.be/VErMT7hzMCk

OBSTACLE AVOIDANCE ROBOT USING ARDUINO UNO

Group Member Names

18BCE0582 - Aditya Ruhatiya

18BCE0586 - Aditya Agrawal

Faculty Name

Dr. RAJAKUMAR K

Associate Professor, SCOPE,
VIT Vellore

Winter Semester

Dec 2019–Jun 2020

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

ABSTRACT

Trajectory planning is one of the most important pivotal point in pick and place tasks done by robotic manipulators. In this work, we have presented a robot, which is compact, autonomous and fully functional.

This robot or a smart car is built to sense any obstacle in its path, to avoid it and resume its running involving the pre-computation of an obstacle free path. Ultrasonic sensors were adapted to implement a real-time obstacle avoidance system for wheeled robots, so that the robot can continually detect surroundings, avoid obstacles, and move toward the target area.

This model has tremendous applications in vacuum cleaners, avoiding concealed paths, parking systems, assembling automobiles and in chemical industries, in scientific exploration, emergency rescue and in other isolated environments.

We use an Arduino UNO with a Motor Shield along with Stepper Motors to make the car, and for sensing we incorporate an Ultrasonic Sensor which accurately and efficiently detects any obstacles in the smart car's path

Throughout the construction of this model, we educated ourselves to the Arduino coding language, the Motor Shield functionality, and comprehensively, with the working of an ultrasonic sensor and its features.

In conclusion, through this project, we aim to construct a model of a smart car that is beneficial to the quotidian problems of the present generation.

INTORDUCTION

Obstacle avoidance Robot is designed in order to navigate the robot in unknown environment by avoiding collisions. Obstacle avoiding robot senses obstacles in the path, avoid it and resumes its running. There are some very popular methods for robot navigation like wall-following, edge detection, line following and many more. A more general and commonly employed method for obstacle avoidance is based on edge detection. A disadvantage with obstacle avoidance based on edge detecting is the need of the robot to stop in front of an obstacle in order to provide a more accurate measurement. All mobile robots feature some kind of collision avoidance, ranging from primitive algorithms that detect an obstacle and stop the robot in order to avoid a collision, using some sophisticated algorithms that enable the robot to detour obstacles. The latter algorithms are more complex, since they involve detection of an obstacle as well as some kind of quantitative measurements concerning the obstacle's dimensions.

LITERATURE SURVEY

"Urban Search and Rescue Robots: From Tragedy to Technology" has been designed and developed by A. David to create an autonomous robot which intelligently detects the obstacle in its path and navigates according to the actions that user set for it. So this system provides an alternate way to the existing system by replacing skilled labour with robotic machinery, which in turn can handle more patients in less time with better accuracy and a lower per capita cost [2]. "Obstacle-avoiding robot with IR and PIR motion Sensors" has been designed and developed by Aniket D. Adhvaryu et al has proposed that developed robot platform

was not designed for specific task but as a general wheeled autonomous platform. It can therefore be used for educational, research or industrial implementation. We can use it to learn the microcontroller programming using C++, Arduino Uno 1.6.5 compiler, IR and PIR sensors characteristics, motor driving circuit and signal condition circuit design. Research on obstacle avoidance robot at the polytechnic level can help students to develop communication, technical skills and teamwork. The design of such robot is very flexible and various methods can be adapted for another implementation. It shows that PIR sensors are more sensitive compared to IR sensors while detecting human being [1]. “A surveillance robot with hopping capabilities for home security” has been designed and developed by G. Song, Y. Kaijian, Z. Yaoxin, and C. Xiuzhen mentioned that enormous amount of work has been done on wireless gesture controlling of robots. Various methodologies have been analysed and reviewed with their merits and demerits under various operational and functional strategies.

Although recent researches in this field have made wireless gesture controlling a ubiquitous phenomenon, it needs to acquire more focus in relevant areas of applications like home appliances, wheelchairs, artificial nurses, table top screens etc. in a collaborative manner [3]. “Autonomous robot navigation with fuzzy logic in uncontrolled environment” has been designed and developed by Reck R and Antoun S. By implementing this model and using Continuous Transmitted Frequency Modulated (CTFM) ultrasonic sensors, along with more advanced localization and mapping algorithms, a more efficient form of autonomous navigation can be created. Through utilizing the CTFM ultrasonic sensor echo data analysis techniques and combining them with Simultaneous Localization and Mapping (SLAM) methodology, it is theorized that the mobile robot will be able to identify obstacles, as well as characteristics and features in the environment [4]. “The arc-transversal median algorithm: a geometric approach to increasing ultrasonic sensor azimuth accuracy” has been designed and developed by H. Choset, K. Nagatani, and N.A. Lazar have mentioned that Obstacle Avoidance Car successfully detects and avoids obstacles. Simple algorithms used to steer and reducing the turning radius, successfully navigated the vehicle. In conclusion, the group successfully interfaced every component that was originally planned. Timer interrupts for IR pulse generation. Obstacle detection using IR transceiver. Servo mechanism using PWM. Steering system using Lego and Servo. [5].

TABLE

YEAR	AUTHOR AND TITLE	METHODOLOGY	OBSERVATION
2020	Reck R, Antoun S “Autonomous robot navigation with fuzzy logic in uncontrolled environment” References:	Using Continuous Transmitted Frequency Modulated (CTFM) ultrasonic sensors, along with more advanced localization and mapping algorithms, a more efficient form of autonomous navigation can be created	The CTFM ultrasonic sensor echo data analysis techniques and combining them with Simultaneous Localization and Mapping (SLAM) methodology, it is theorized that the mobile robot will be able to identify obstacles
2013	**1.** Antoun, S.M., McKerrow, P.J.: Issues in wall tracking with a ctfm ultrasonic sensor. IEEE Sens. J. 13, 4671–4681		
1998	**2.** Yata, T., Kleeman, L., Yuta, S.: Wall following using angle information measured by a single ultrasonic transducer. In: Proceedings of 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146), vol. 2, pp. 1590–1596. IEEE		

2009	G. Song, Y. Kaijian, Z. Yaoxin, and C. Xiuzhen, "A surveillance robot with hopping capabilities for home security",	A six-bar linkage leg system to enable hopping locomotion. It can also roll freely on flat floors and change its directions by the two-wheeled differential drive system	Experimental results verify that the prototype robot is a powerful home security device that can patrol in cluttered home environments with ease
2005	Aniket D. Adhvaryu et al “Obstacle-avoiding robot with IR and PIR motion Sensors”	Infrared and passive infrared motion sensors for detecting obstacles in the path and making decisions.	PIR sensors are more sensitive as compared to IR sensors while detecting human being.
2003	H. Choset, K. Nagatani, and N.A. Lazar, "The arc-transversal median algorithm: a geometric approach to increasing ultrasonic sensor azimuth accuracy"	Simple algorithms used to steer and reducing the turning radius, successfully navigated the vehicle.	Timer interrupts for IR pulse generation. Obstacle detection using IR transceiver. Servo mechanism using PWM. Steering system using Lego and Servo
2002	A. David, “Urban Search and Rescue Robots: From Tragedy to Technology”,	Create an autonomous robot which intelligently detects the obstacle in its path and navigates according to the actions that user set for it.	Alternate way to the existing system by replacing skilled labour with robotic machinery,

METHODOLOGY

HARDWARE COMPONENTS USED

1. Arduino UNO
2. 2 Wheel Drive robotic chassis
3. Two DC BO motors
4. L293 motor driver
5. HC-SR04 Ultrasonic sensor
6. Switch
7. 9v Batteries and connector
8. Jumper wires
9. Caster
10. Nut-Bolts, Spacer
11. Other supporting components and tools

Ultrasonic sensor: The ultrasonic sensor has a signal generator and a receiver. The signal generator generates an ultrasonic wave and transmits in the forward direction. The transmitted wave strikes any obstacle in its path and a huge part of it gets reflected. The receiver receives the reflected wave.

The obtained values from the ultrasonic sensor need to be calibrated in order to get a meaningful data (distance). The distance of the object is calculated on the basis of the time taken by the wave in the process of transmission, reflection and collection

Arduino: Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs (Like-light on a sensor, a signal of a button, or a signal from sensor etc) and turn it into an output (Like- activating a motor, turning on an LED, publishing something online etc). We can tell our board what to do by sending a set of instructions to the microcontroller on the board. To do so we use the Arduino programming language and the Arduino Software (IDE).

In this project, to avoid the obstacle in the path, a condition is put in the system which says: if the distance between the robot and the object gets below a certain level, stop the robot and take a backward motion and then turn the robot into other direction and continue the loop. This logic is applied to the system by writing the code in the arduino.

L293 motor driver: We now have the conditions set up for the robot car but it needs to be executed/implemented on the hardware. The hardware used is the DC motor. To drive these DC BO Motors, we need motor driver. In this project we are using L293 motor driver. Motor driver is used to send the commands to motors according to signal received from Arduino.

BO Motors: Two motors are used in this process: left motor and right motor. To move the robot car forward, both the motors are turned on. For backward step, both motors need to run in opposite direction. To turn the robot car to avoid obstacle, one of the motor is reversed for a while, keeping the other motor forward

CONNECTIONS

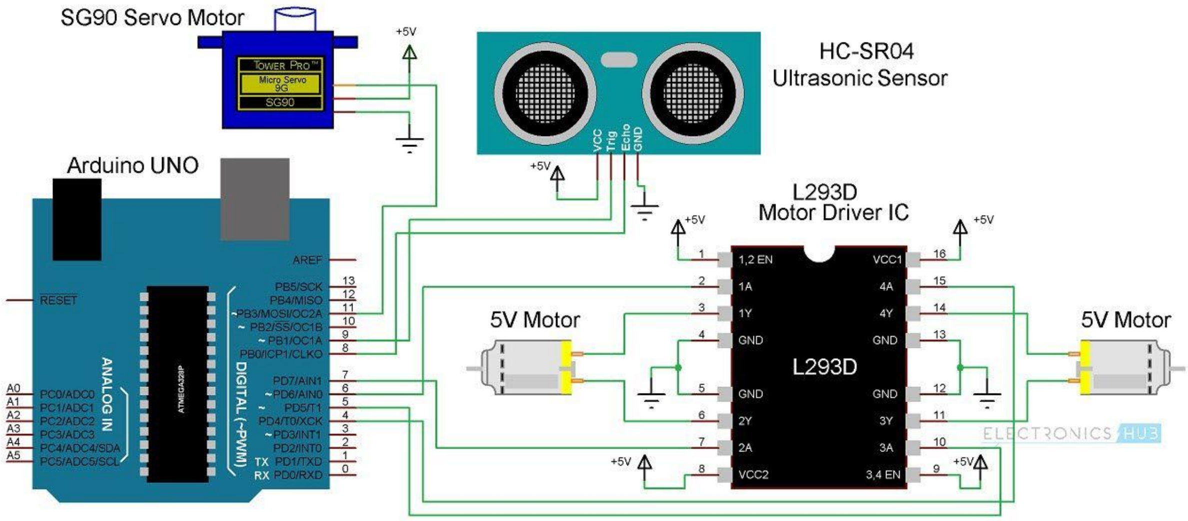
Motor Driver connection:

Vin → 9v Battery (+)ve
G → 9v Battery (-)ve
Motor2 → Left Motor connection
Motor1 → Right Motor connection
M1a and M1b → Arduino 7 and 6 (If motor runs in wrong direction, connection is swapped)
M2a and M2b → Arduino 5 and 4 (If motor runs in wrong direction, connection is swapped)

Ultrasonic connection:

Gnd: Arduino GND
Echo: Arduino A2
Trig: Arduino A1
Vcc: Arduino 5V

ARCHITECTURE DIAGRAM



CODE

```
#include <Servo.h>          //Servo motor library. This is standard library, we are not using servo in this Robot
#include <NewPing.h>        //Ultrasonic sensor function library. You must install this library

//our L298N control pins
const int LeftMotorForward = 7;
const int LeftMotorBackward = 6;
const int RightMotorForward = 4;
const int RightMotorBackward = 5;

//sensor pins
#define trig_pin A1 //analog input 1
#define echo_pin A2 //analog input 2

#define maximum_distance 200
boolean goesForward = false;
int distance = 100;

NewPing sonar(trig_pin, echo_pin, maximum_distance); //sensor function
Servo servo_motor; //our servo name


void setup(){
  pinMode(RightMotorForward, OUTPUT);
  pinMode(LeftMotorForward, OUTPUT);
  pinMode(LeftMotorBackward, OUTPUT);
  pinMode(RightMotorBackward, OUTPUT);


}

void loop(){

  int distanceRight = 0;
  int distanceLeft = 0;
  delay(50);

  if (distance <= 25){
    moveStop();
    delay(100);
    moveBackward();
    delay(400);
    moveStop();
    delay(300);

    if (distance >= distanceLeft){
      turnRight();
      moveStop();
    }
    else{
      turnLeft();
      moveStop();
    }
  }
  else{
    moveForward();
  }
    distance = readPing();
}

int lookRight(){
  servo_motor.write(50);
  delay(500);
  int distance = readPing();
  delay(100);
  servo_motor.write(115);
  return distance;
}

int lookLeft(){
  servo_motor.write(170);
  delay(500);
  int distance = readPing();
  delay(100);
  servo_motor.write(115);
  return distance;
  delay(100);
}

int readPing(){
  delay(70);
  int cm = sonar.ping_cm();
  if (cm==0){
    cm=250;
  }
  return cm;
}

void moveStop(){

  digitalWrite(RightMotorForward, LOW);
  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(RightMotorBackward, LOW);
  digitalWrite(LeftMotorBackward, LOW);
}

void moveForward(){

  if(!goesForward){

    goesForward=true;

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorBackward, LOW);
  }
}

void moveBackward(){

  goesForward=false;

  digitalWrite(LeftMotorBackward, HIGH);
  digitalWrite(RightMotorBackward, HIGH);

  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(RightMotorForward, LOW);

}

void turnRight(){

  digitalWrite(LeftMotorForward, HIGH);
  digitalWrite(RightMotorBackward, HIGH);

  digitalWrite(LeftMotorBackward, LOW);
  digitalWrite(RightMotorForward, LOW);

  delay(900);

  digitalWrite(LeftMotorForward, HIGH);
  digitalWrite(RightMotorForward, HIGH);

  digitalWrite(LeftMotorBackward, LOW);
  digitalWrite(RightMotorBackward, LOW);
```

```
}

void turnLeft(){

  digitalWrite(LeftMotorBackward, HIGH);
  digitalWrite(RightMotorForward, HIGH);

  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(RightMotorBackward, LOW);

  delay(900);

  digitalWrite(LeftMotorForward, HIGH);
  digitalWrite(RightMotorForward, HIGH);

  digitalWrite(LeftMotorBackward, LOW);
  digitalWrite(RightMotorBackward, LOW);
}
```

IMPLEMENTATION SCREENSHOTS:

1 The robot detects the obstacle

2. robot goes back

3. robot turns left and avoids obstacle

CONCLUSION:

This project develops an obstacle avoiding robot to detect and avoid obstacles in its path. The robot is built on the Arduino platform for data processing and its software counterpart helped to communicate with the robot to send parameters for guiding movement. For obstacle detection three ultrasonic distance sensors were used that provided a wider field of detection. The robot is fully autonomous and after the initial loading of the code, it requires no user intervention during its operation. When placed in unknown environment with obstacles, it moved while avoiding all obstacles with considerable accuracy. In order to optimize the movement of the robot, we have many considerations for improvement. However, most of these ideas will cost more money and time as well. In future cameras can be used to detect the obstacle. Even the ones we mentioned in the camera holder part will be better because of the special software.

REFERENCES:

1. Aniket D. Adhvaryu et al “Obstacle-avoiding robot with IR and PIR motion Sensors” IOP Conference Series: Materials Science and Engineering, vol. A247, pp. 529-551, April 2005.
2. A. David, “Urban Search and Rescue Robots: From Tragedy to Technology”, IEEE Intelligent systems 17.2, 2002, pp. 81-83.
3. G. Song, Y. Kaijian, Z. Yaoxin, and C. Xiuzhen, "A surveillance robot with hopping capabilities for home security", IEEE Transactions on Consumer Electronics 55.4, 2009.
4. Reck R, Antoun S “Autonomous robot navigation with fuzzy logic in uncontrolled environment”, Advances in Intelligent Systems and Computing 1070, pp. 275-283
5. H. Choset, K. Nagatani, and N.A. Lazar, "The arc-transversal median algorithm: a geometric approach to increasing ultrasonic sensor azimuth accuracy", IEEE Transactions on Robotics and Automation 19.3, 2003, pp. 513-521.