Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 6 (2023)

Ardubot Path Finder: Road Obstacles Finder through Auto Navigation Using Artificial Intelligence and Internet of Things (Iot)

Dr.V.Alamelu Mangayarkarasi[1], Dr.A.Adhiselvam[2]

[1]Associate Professor and Head, Department of Computer Applications, Sengamala Thayaar Educational Trust Women's college (Autonomous), Sundarakkottai-614016, Thiruvarur-Dt.

[2]Professor and Head, Department of Information Technology, Dr.N.G.P. Arts and Science College, Coimbatore-48

Abstract

Obstacles avoiding robot using Arduino is a project aimed at developing an autonomous robotic system, capable of navigating through cluttered environments while avoiding obstacles. The proposed system is built around the Arduino microcontroller, which serves as the brain of the robot. The robot is equipped with an ultrasonic sensor to detect obstacles in its path. The sensor emits ultrasonic waves and measures the time taken for the waves to bounce back, enabling it to calculate the distance to the obstacles. If an obstacle is detected within a specified range, the robot initiates an avoidance maneuvered by instructing the motors to change the direction. This ensures that the robot maneuvers around the obstacle and continues along its intended path. To achieve precise control over the robot's movements, motor drivers are employed to regulate the speed and direction of the motors. Typically, a rechargeable battery powers the robot, making it portable and removing the need for ongoing tethering. The implementation of this project involves programming the Arduino microcontroller using the Arduino IDE, interfacing the ultrasonic sensor and motor drivers with the microcontroller, and integrating them into a compact chassis. The system can be further enhanced by a Bluetooth module, to enable more advanced obstacle detection and avoidance capabilities.

***Keywords**: Arduino, Colliding, Navigate, Obstacle, Robot, etc.*

1. Introduction

The Internet of Things (IoT) is a powerful technological development that allows objects, devices, and machines to communicate and exchange data with one another through interconnected networks. This technology holds the potential to revolutionize various sectors such as healthcare, manufacturing, transportation, agriculture and more, by enabling machines and devices to operate efficiently, minimize waste, save time, and reduce human intervention. The method employed for Ardubot the maze runner is based on ultrasonic distance sensing algorithm and motor control logic algorithm. The design and development of the robot, along with the hardware and software components employed, are thoroughly explained in this paper. The authors want to build a fully autonomous robot that can move around a space without assistance from a person. The robot's powers include the ability to recognize impediments and alter its course to avoid them. As a result this research emphasizes the significance of creating robots that can maneuver through complicated situations and offers a helpful illustration of an effective application of obstacle avoidance technology. The hardware and software components of the robot are thoroughly described through this paper. The three ultrasonic sensors, an Arduino and a motor controller are included in the hardware. The Arduino code used to direct the robot's movements and the obstacle detection algorithm are among the software components.

2. Review Of Literature

The idea of this research [1] in the design and manufacturing ultrasonic sensor handheld was to combine the properties of sound monitoring and that benefit the blind and vibrating alert feature, which benefit from the experience of deafness. The authors have suggested a new strategy that utilizes a combination of sensory feedback to give these people a reliable method of obstacle detection. In this paper [2], the development of an autonomous obstacle avoidance robot using ultrasonic sensors was presented which design the ultrasonic sensor serves as the eye of the robot to aid its autonomous movement. The authors proposed [3] the obstacle avoiding robot directs whenever an obstacle comes in its path. This robotic vehicle was built, using a micro-controller of AT mega 8 family. An ultrasonic sensor was used to find any obstacle ahead of it and sends a command to the micro-controller. This report [4] presented a method for autonomous navigation of a mobile robot, a concept which can be built upon with the goal of achieving a long-term robotic activity without any human intervention. This article [5] presented the implementation of a strategy for detecting, mapping and avoiding obstacles using the mobile robotic platform DaNI 2.0. This mobile robot included an ultrasonic sensor, which is modeled based on a Gaussian function with the aim of reaching higher precision of the distance position measured for each obstacle in the scene. An avoidance obstacle algorithm based on reactive path planning, allows the mobile robot navigate coherently around the environment. The path planning method analyzes the change in potential field around the environment. The paper [6] represents the design of an Obstacle Avoiding Robot with the capability of detecting objects in its course and navigating around those objects by making a proper decision. It demonstrats a robotic unit based on Arduino UNO and Adafruit Motor Shield where the code is written in Arduino IDE Software. In the paper [7] an Obstacle Avoiding Robot was designed which can detect obstacles in its path and maneuver around them without making any collision. It is a robot vehicle that works on Arduino micro-controller and employs three ultrasonic distance sensors to detect the obstacles. The main objective of this research [8] is to provide simple guidelines to the technical oriented students and beginners who are interested in this type of research.

3. Existing System

1. The robot can detect obstacles only in a narrow range in front of it. It has a drawback Which it has only two sensors in its left and right side So that it will only able to sense Left and its right side. It can’t observe the in-front obstacles.
2. Even though the robot is not very precise, it is capable of detecting nearby obstacles and With the help of Arduino UNO and H-bridge, it can maneuver the car to avoid collisions.
3. The robot may not be able to detect obstacles from different directions or angles.
4. The robot can avoid obstacles, but it may not be able to navigate around them efficiently
5. May not be suitable for complex environments or situations where multiple obstacles are Present
6. The robot range of obstacle detection is limited to a single sensor, which may not be Enough in certain situations.
7. The robot may have difficulty avoiding obstacles that are at an angle or to the side of the Sensor' s field of view.
8. The robot may have difficulty navigating around obstacles that are in close proximity to Each other or are in irregular shapes.
9. The robot movements may not be as smooth or efficient as desired.

4. Proposed System

An obstacle-avoiding robot is a type of mobile robot that is designed to navigate autonomously in an environment with obstacles and avoid them while reaching its intended destination. The problem definition for an obstacle-avoiding robot involves the development of an intelligent algorithm that allows the robot to sense the environment, detect obstacles and navigate around them to reach a goal location. The robot must be equipped with sensors such as proximity sensors, infrared sensors, or ultrasonic sensors that can detect the distance and location of obstacles in the environment. The algorithm must take into account the position and orientation of the robot, the location and size of obstacles and the intended goal location. The robot must be able to make decisions in real-time and adjust its trajectory to avoid collisions with obstacles. The problem also involves optimizing the algorithm for efficiency and accuracy, as well as designing a suitable control system for the robot movements. Viewing as a whole the problem of obstacle avoidance for a robot requires a combination of hardware design, sensing and control algorithm development. Ultrasonic sensor is a non-contact level measurement method that uses sound waves to determine the process material being measured.

Objectives

1. To develop a robust algorithm for obstacle detection and avoidance, considering the factors like sensor data processing, path planning and real- time decision – making.

2. To design the physical structure of a robot and hardware components to support obstacle avoidance, including motor control and sensor mounting.

Methodology

1. Choose a robot platform with motorized wheels and sensor space. Installing an Arduino microcontroller will enable you to interface and regulate sensors. Attach ultrasonic sensors (HC-SR04, for example) to the robot and place them in various locations to achieve a wider field of vision.

2. Make a calibration procedure to ensure accurate distance measurements from the ultrasonic sensors. Consider any noise or interference present in the sensor data.

3. Utilize the micro-controller to read data from the sensors. Determine the barriers distance and orientation in relation to the robot's position by analyzing the data.

4. Determine the obstacle detection system's cutoff distance. Compare the sensor data with the threshold to identify any blockage.

5. Use an algorithm (such as a simple "move and avoid" or a more complex path planning system) to decide how the robot should move to avoid obstacles. Consider the factors like the size of the surrounding area, the turning radius, and the robot's speed.

6. Utilizing the algorithm's output, regulate the robot's motors. Assume accurate and seamless motions in reaction to sensor data.

7. Test the robot in a controlled environment with various obstacles. Optimize the algorithm for better performance, considering factors like speed, accuracy and efficiency.

8. Add more sensors or modules (such as encoders for precise movement) to enhance the robot's capabilities. Consider incorporating wireless communication for applications requiring autonomous or remote control.

5. ALGORITHM

Ultrasonic Obstacle Avoidance Algorithm

Step 1: *Initialize the robot's orientation, position, and goal location.*

Step 2: *Read data from the ultrasonic sensors continuously in order to identify impediments.*

Step 3: *Examine the sensor data to determine whether obstacles exist and where they are located. Utilize a threshold distance to classify objects as obstacles.*

Step 4: *Determine an evasive path based on the locations of the obstacles. In this calculation, take into account the robots present orientation and position.*

Step 5: *Change the movement orders to make sure the robot stays on the estimated course. Make sure that changes are applied smoothly and consistently to prevent roadblocks in real time.*

Step 6: *Depending on its movements, update the robot's orientation and position. Verify whether you've arrived at the desired spot or whether any new barriers have emerged.*

Step 7: *Optimize the algorithm to reach the highest levels of precision and effectiveness. Think about things like robot speed, sensor noise, and the intricacy of the surroundings.*

Step 8: *Repeat the reading of sensor data, obstacle detection, and movement adjustment process until the desired outcome is achieved.*

Pseudo Code of the Algorithm

```
Initialize robot, sensors, goal location
while not at goal location do
read sensor data
obstacles = detect_obstacles(sensor_data)
if obstacles is not empty then
avoid_obstacles(obstacles)
else
move_towards_goal()
end if
end while
function detect_obstacles(sensor_data)
obstacles = []
for each sensor in sensors do
if sensor detects obstacles then
obstacles.append(obstacles_location)
end if
end for
return obstacles
end function
function avoid_obstacles (obstacles )
calculate_avoidance_trajectory(obstacles)
adjust_movement()
end function
function move_towards_goal()
calculate_trajectory_to_goal()
adjust _movement()
end function
```

An obstacle-avoiding robot's fundamental algorithm is described in the provided code. The goal location, sensors, and robot are initialized. Until it reaches the destination, the robot keeps performing the subsequent actions, Examine sensor data to find obstructions. To identify obstacles based on sensor data, call the detect obstacles function.

Call the avoid obstacles function to generate an avoidance trajectory and modify movement to avoid obstacles if they are found. Call the move_towards_goal function to compute a trajectory towards the goal location and modify movement if no obstacles are found. The function repeatedly checks to see if each sensor picks up any impediments. When an impediment is found, it's the list of barriers now includes the location. The function "avoid obstacles" (avoid_obstacles): The list of barriers is the input for this function. Using the locations of the obstacles as a foundation, it computes an avoidance trajectory. It modifies the robot's motion to stay clear of the obstructions.
Function Move towards Goal (move_towards_goal): This function determines a path that leads to the desired destination. It modifies the robot's motion to adhere to the determined path. All things considered, this algorithm enables the robot to navigate toward a destination on its own while dodging impediments that are picked up by its sensors. When impediments are identified, the avoid_obstacles function is triggered, guaranteeing the robot's safe navigation through its surroundings.

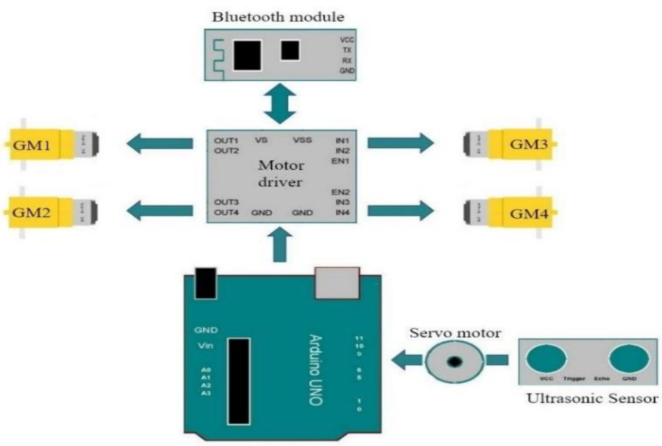
7. System Design And Analysis

Ultrasonic sensor is used to sense in three different directions and avoid the obstacles. So that the object in-front of the robot will be detected easily and precisely. Here the speed will be high when compared with the existing robot.

- The robot range of obstacle detection is significantly increased by using servo motor, allowing it to detect obstacles from multiple angles and directions.
- The robot can better avoid obstacles that are at an angle or to the side of the sensors' field of view.
- The robot movements can be smoother and more efficient due to better obstacle detection and avoidance.
- Using the ultrasonic sensor to detect obstacles from different directions (front, left and right).
- The robot can navigate around obstacles efficiently, improving its overall movement and speed.
- Suitable for complex environments or situations where multiple obstacles are present.

- May require more complex programming and wiring than the existing system.

8. PHYSICAL STRUCTURE OF ARDUINO ROBOT



9.

Figure 1: System Architecture

The system architecture (figure 1) of a Maze runner typically includes the following components:

Sensors: The robot is equipped with sensors such as ultrasonic or infrared sensors to detect obstacles in its path. An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic waves and then converting the reflected waves into an electrical signal. Ultrasonic waves travel way faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the receiver and the transmitter. The transmitter emits the sound using piezoelectric crystals and the receiver encounters the sound after it has travelled to and from the target. The sensor directions are given in Table 2.

Microcontroller: A micro-controller, such as an Arduino, is used to control the robot movements and process the sensor data. Arduino is an open-source electronics platform that is based on the easy-to-use software and hardware. Arduino board is capable of reading inputs (like light glowing on a sensor, a finger touched on a button, or a message of Twitter) and turn it into an output (like activating motors, turning on an LED, publishing some content online). We can tell our board what to do by sending a group of instructions to the microcontroller on the board. Over many years, in the past Arduino has been the brain of hundreds and thousands of projects, from day-to-day objects to complex scientific instruments. It was developed at the Ivrea Interaction Design Institute, which aims to help students who are without a background in electronics and programming for fast prototyping.

Motor driver: A motor driver is used to control the speed and direction of the robot motors. It is an integrated circuit chip. It is mostly used to control motors in an autonomous robot. Motor driver IC acts as an interface between the motors and the microprocessors in the robot. L293 series such as L293D, L293NE are the most commonly used motor drivers. These motor drivers are designed to control the above two DC motors together. L293D consists of two H-bridge. H- Bridge is the simplest of all the circuits for controlling a low current rated motor. The L293D IC receives signals from the microprocessor and transmits a relative signal to the motors. It has two voltage pins, one of which is used to apply voltage to the motors and the other to draw current for the working of the L293D.

Gear motor: The type and size of the motors depend on the size and weight of the robot. A gear motor is a mechanical system consisting of an electric motor and a gearbox containing a series of gears. The function of the gearbox coupled to the motor is to reduce its speed and increase its torque to do a given job at a given speed.

Communication module (Bluetooth Module): A communication module such as Wi-Fi or Bluetooth is required to communicate with the robot wirelessly. The HC-05 has two operating modes, one is the Data mode in which it can send and receive data from other Bluetooth devices and the other is the AT Command mode where the default device settings can be changed. We can operate the device in either of these two modes by using the key pin as explained in the pin description.

Servo motor: A servo motor is a type of electric motor that is commonly used in robots, automation and other applications that require precise control of position, speed and torque. It consists of a small DC motor connected to the gearbox, along with a control circuit which determines the position and speed of the motor. Then the computer processes that information and sends a signal to the servo attached to the throttle to adjust the engine speed.

10. Result And Discussion

Obstacle detection based on distance and Angle of sensors

The distance values from the ultrasonic sensors are usually used as sample sensor data for ultrasonic obstacle avoidance. The distance between any given sensor and the closest obstruction inside its range of vision is indicated by the distance measurement that each sensor offers. The undermentioned table is an illustration of how the sensor data may appear

Sensor number	Angle (degrees)	Distance(cm)
1	0	30
2	90	25
3	180	40
4	270	15

Table 2: Angle Based Sensor Data

In Table2. The numbers show that impediments are present at three, two, three, and four centimeters, respectively, from sensors 1, 2, 3, and 4. With the use of this information, the robot is able to identify and avoid obstacles.

The robot uses a servo motor to rotate the ultrasonic sensor and detect obstacles from multiple angles. Each entry in the table corresponds to a sensor reading at a specific angle (in degrees). If no obstacle is detected at a particular angle, it is indicated by "No obstacle".

Finding the safe path from the angle and distance based data using test cases

In each test case, the robot uses its ultrasonic sensors mounted on a servo motor to detect obstacles from different angles. Test cases for a robot using ultrasonic sensors for obstacle avoidance can be applied to verify and validate the functionality of the robot's navigation system. Prioritizing the execution order of the test cases can enhance the software testing efficiency. Test case prioritization is the process of scheduling the order of test case execution in order that the higher priority test cases are executed first[9].

Robot Behavior and the Expected Outcome

1. No obstacles

The robot moves directly towards the destination without any deviation.

2. Single obstacle

The robot detects the obstacle, plans a path around it, and successfully reaches the destination.

3. Multiple obstacles

The robot detects and avoids each obstacle, adjusting its path as necessary to reach the destination.

4. Obstacle at close range

The robot detects the close –range obstacle, plans a precise maneuver to avoid it, and continues towards the destination.

5. Obstacle at different angles

The robot successfully detects and avoids obstacles at various angles while navigating towards the destination.

6. Path planning efficiency

The robot efficiency plans its path, avoiding obstacles and reaching the destination in a timely manner.

Test case prioritization can be beneficial for testing the robot’s obstacle avoidance algorithm, especially when there are limited resources or time constraints. Prioritizing test cases ensures that the most critical and impactful scenarios are tested first, increasing the likelihood of identifying important issues early in the development process.

Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 6 (2023)

Test cases of sensor data

Test case	Sensor	Angle (degrees)	Distance (cm)
1	1	45	30
1	2	135	25
1	3	225	40
1	4	315	No
2	1	60	35
2	2	150	30
2	3	240	No
2	4	330	45
3	1	90	No
3	2	180	25
3	3	270	30
3	4	360	35
4	1	45	30
4	2	135	25
4	3	225	40
4	4	315	35
5	1	60	35
5	2	150	30
5	3	240	25
5	4	330	No
6	1	90	40
6	2	180	35
6	3	270	No
6	4	360	30
7	1	45	25
7	2	135	30
7	3	225	30
7	4	315	40
8	1	60	30
8	2	150	35
8	3	240	40
8	4	330	No
9	1	90	No
9	2	180	25
9	3	270	30
9	4	360	35
10	1	45	30

10	2	135	25
10	3	225	40
10	4	315	35

Table 1: Test cases of sensor data

A series of test scenarios for ultrasonic obstacle detection employing four sensors positioned at various angles are shown in the table. The sensor number, the angle at which the sensor is positioned, and the centimeter-based distance to the identified obstruction are all included in each test case.

Prioritized Test cases

Test case	Sensor	Angle (degrees)	Distance (cm)
1	1	45	30
1	2	135	25
1	3	225	40
1	4	315	No
4	1	45	30
4	2	135	25
4	3	225	40
4	4	315	35
2	1	60	35
2	2	150	30
2	3	240	No
2	4	330	45
7	1	45	25
7	2	135	30
7	3	225	30
7	4	315	40
6	1	90	40
6	2	180	35
6	3	270	No
6	4	360	30
3	1	90	No
3	2	180	25
3	3	270	30
3	4	360	35
5	1	60	35
5	2	150	30
5	3	240	25
5	4	330	No
8	1	60	30
8	2	150	35

8	3	240	40
8	4	330	No
10	1	45	30
10	2	135	25
10	3	225	40
10	4	315	35
9	1	90	No
9	2	180	25
9	3	270	30
9	4	360	35

Table 2: Prioritized Test cases

In table 2 represents a prioritized test cases of sensor data collected by the robot during the obstacle detection.

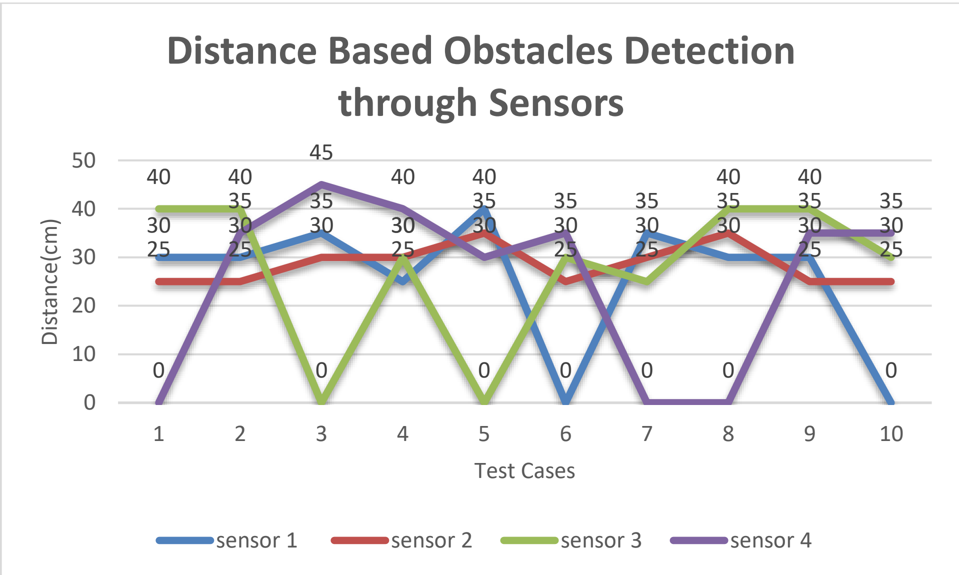


Figure 2 Distance Based Obstacles Detection

This graph (Figure 2) represents the distances of obstacles detected by sensors of robot during the process. No obstacles path clearly represented by this graph using sensor and distance data.

System to be implemented in following areas

- Warehouse robotics: Robots can use this system to navigate through warehouse shelves and avoid collisions with obstacles or other robots
- Autonomous vehicles: This system can be adapted for autonomous cars to detect obstacles from multiple angles and safely navigate roads
- Search and rescue robots: Robots used in search and rescue missions can use this system to navigate through debris and find survivors.
- Home cleaning robots: Robots like vacuum cleaners can use this system to avoid obstacles and efficiently clean different areas of a house.
- Industrial automation: Robots in manufacturing plants can use this system to navigate around machinery and avoid collisions

10. Conclusion

In conclusion, obstacle-avoiding robots are a promising innovation that can revolutionize industries and their day to day performance. With advanced sensors and intelligent algorithms, these robots can navigate complex environments, avoid obstacles and improve productivity and safety in the industries like manufacturing, logistics and transportation. They also have potential applications in household chores, healthcare and search and rescue operations. Despite the challenges, advancements in robotics and AI are driving the development of more capable obstacle-avoiding robots. To have an overall view these robots represent a significant advancement in technology with the potential to transform industries and improve our lives. Further research and development will lead to even more sophisticated robots that shape the future of automation and robotics.

11. Future Enhancement

- Advanced sensor technology: Sensor technology is constantly evolving and new sensors are being developed that can provide more accurate and reliable obstacle detection. For example, depth sensors such as Microsoft's Kinect or Intel's Real Sense could be used to provide more detailed information about the environment.
- Multi-robot coordination: Coordinating multiple robots to work together could enable them to cover more grounds and share information about the environment. This could involve developing communication protocols between the robots or using machine learning algorithms to enable the robots to learn from each other.
- Machine learning: Incorporating machine learning algorithms could enable the robot to learn from its experiences and improve its obstacle avoidance capabilities over time. This could involve using reinforcement learning to train the robot to avoid obstacles more effectively or using computer vision algorithms to recognize objects and learn how to avoid them.
- Autonomous charging: Incorporating autonomous charging capabilities would enable the robot to operate for longer periods of time without human intervention. This could involve using wireless charging technology or developing a docking station that the robot can return to when it needs to recharge.
- More advanced locomotion: Enhancing the robot locomotion capabilities could enable it to navigate through more challenging terrain. This could involve adding additional legs or wheels, or developing more advanced walking or climbing are such other movement capabilities.
- For long-distance navigation, other sensor technologies like LiDAR (Light Detection and Ranging) or cameras with advanced computer vision algorithms are more suitable. LiDAR can detect objects at longer ranges with high accuracy, making it ideal for outdoor environments or applications requiring precise long-distance object detection. Cameras combined with computer vision algorithms can also provide long-range detection and recognition of objects, but they may be affected more by lighting and environmental conditions compared to LiDAR

12. References

[1] Mahdi Safaa, A., Muhsin Asaad, H., & Al-Mosawi Ali, I. (2012). Using ultrasonic sensor for blind and deaf persons combines voice alert and vibration properties. *Research Journal of Recent Sciences*

[2] Agbeyangi, A. O., Alashiri, O., Odiete, J. O., & Adenekan, O. A. (2020). An autonomous obstacle avoidance robot using ultrasonic sensor. *Journal of Computer Science and Its Application, 27*(1).

[3] T. Aravinthan, G.V. Barani Krishnan, 0, An Obstracle Avoiding Robot, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Confcall – 2018 (Volume 06 – Issue 14),

[4] Dube, B., Kazi, R., Malya, A., & Gala, N. (2019, March). IR-Based Obstacle Avoiding and Self-Navigating Robot. In *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)* (pp. 1-4). IEEE.

[5] Ojeda, D. L. A., Vera, Y. G., & Manzano, M. A. I. (2015). Obstacle detecction and avoidance by a mobile robot using probabilistic models. *IEEE Latin America Transactions, 13*(1), 69-75.

[6] Efaz, E. T., Al Mamun, A., Salman, K., Kabir, F., Sakib, S. N., & Khan, I. (2019, December). Design of an indicative featured and speed controlled obstacle avoiding robot. In *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)* (pp. 1-6). IEEE.

[7] Faiza Tabassum, Susmita Lopa, Muhammad Masud TarekandDr.Bilkis Jamal Ferdosi (2017). Obstacle Avoiding Robot. Global. Journal of Researches in Engineering: H, Robotics & Nano-Tech, vol. 17, no. 1, pp. 18-23.

[8] Ismail, R., Omar, Z., & Suaibun, S. (2016, October). Obstacle-avoiding robot with IR and PIR motion sensors. In *IOP Conference Series: Materials Science and Engineering* (Vol. 152, No. 1, p. 012064). IOP Publishing.

[9] Alamelu Mangayarkarasi,V Srinath, M.V(2016 OCTOBER) ”An Efficient DAGbM-KSJS Algorithm for Agile Software Testing, International Review on Computers and Software (I.RE.CO.S.), (Vol. 11, N. 10 ISSN 1828-6003)