Powered Obstacle Avoidance with Bluetooth-Controlled Navigation in Automated Vehicles Using Raspberry Pi

T. Raghavendra Gupta[1], S. Shiva Shanker Prasad[2], S. Vivek Vardan[3], S. Rakesh Reddy[4], P. Abhiram Sharma[5]
[1]*Associate Professor, Department of CSE, HITAM, Hyderabad, India*
[2,3,4,5]*Student of Computer Science and Engineering, HITAM, Hyderabad, India*

***Abstract*—One of the critical challenges in automated vehicle technology has always been reliable obstacle detection and avoidance. Initially, systems relied heavily on basic sensor technologies like infrared and simple ultrasonic sensors, which provided limited detection capabilities. As the demand for safer and more reliable automated vehicles grew, the industry saw the integration of more advanced technologies, including LiDAR, radar, and high-resolution cameras. However, these solutions often came with high costs, making them less accessible for widespread use. In response to the need for cost-effective yet reliable obstacle avoidance systems, recent innovations have focused on integrating affordable and efficient components like the Raspberry Pi and ultrasonic sensors. These systems, while initially simple, have been progressively enhanced to include additional sensors, such as metal detectors and cameras, to provide a more comprehensive detection capability. Moreover, the incorporation of Bluetooth technology has allowed for more seamless user interaction, particularly through smartphone applications, further increasing the usability and flexibility of these systems. In the advanced version of this system, additional features are being incorporated to further enhance detection capabilities and overall performance. These enhancements include the integration of a metal detector and a camera, which will provide more comprehensive detection and improve the system's ability to navigate complex environments. These improvements aim to ensure robust obstacle avoidance, making automated vehicles safer and more reliable.**

***Index Terms*—Arduino UNO, Bluetooth, Motor driver, Ultrasonic sensor, servo motor driver, SD card, Obstacle detection, camera.**

I. INTRODUCTION

Obstacle avoidance systems play a crucial role in enabling automated vehicles to navigate safely by identifying and avoiding obstacles in real time. These systems typically use sensors such as LiDAR, ultrasonic sensors, and cameras to gather data about their surroundings and make decisions on adjusting the vehicle's speed or direction. However, over time, the accuracy of these systems can decrease due to several factors, including sensor wear, environmental conditions like dust or fog, and limitations in processing large amounts of data. This reduction in accuracy can delay the vehicle's response to obstacles, posing safety risks, especially in high-traffic or complex environments.

Our project aims to address these challenges by improving the accuracy and reliability of obstacle detection in automated vehicles. We plan to enhance sensor precision and optimize the algorithms used to process sensor data. By refining these algorithms, the system can more efficiently analyze the data and make quicker, more accurate decisions. This will help maintain a high level of performance over time, reducing the risk of sensor degradation or delayed responses. Our focus is on ensuring the system performs well in both ideal and challenging environmental conditions.

In addition to improving accuracy, we are also introducing a Bluetooth-based manual control system. This feature will allow a human operator to take control of the vehicle in the event of a system malfunction or failure, providing a fail-safe option. The combination of enhanced obstacle detection and the backup manual control will increase both the safety and reliability of automated vehicles, making them better suited for real-world applications where safety and accuracy are critical.

II. RELATED WORK

Now in this related work part, we will discuss some work that has been done in this field.

A. Obstacle Detection and Avoidance Robot

According to the research paper [1] developed an obstacle detection and avoidance robot using ultrasonic sensors and servo motors. The system operates by continuously scanning the environment for obstacles and reacting when an obstacle is detected at a distance of 15 cm. The ultrasonic sensor, mounted on a servo motor, helps the robot scan the front hemisphere by rotating between 0 and 180 degrees. The robot measures distances from 10 different directions, storing the results in variables for further comparison. Based on these readings, the robot decides whether to turn left or right, depending on which side has the greater distance from obstacles. This decision-making process allows the robot to move efficiently through its environment.

Additionally, the system integrates Bluetooth control as a manual override, ensuring that the robot can be controlled remotely in case of any failure in its autonomous navigation. This combination of autonomous operation with a manual backup enhances the reliability of the system, especially in unpredictable environments. Data from the ultrasonic sensor is continuously recorded and stored on an SD card for analysis. This ensures that all the detected distances and robot movements can be monitored in real-time via the Arduino’s serial monitor.

The robot has been calibrated to ensure a movement accuracy of 92.5%, based on multiple trials. Furthermore, speed calculations were made using a wheel diameter of 7 cm, with the motor running at 210 rotations per minute. The robot achieves a speed of 76.93 cm per second, ensuring efficient movement [1]
.

B. Obstacle Avoidance of a Mobile Robot Using Hybrid Learning Approach

According to the research paper [2] as robotics technology continues to grow, robots are being used in more complex environments and given more challenging tasks that require them to operate with high levels of independence. To effectively move around in these dynamic and unpredictable settings, robots need to process information from their surroundings and react quickly in real-time. The ultimate aim in mobile robotics is to achieve autonomous behavior, allowing robots to learn from their experiences and improve their actions over time. One promising way to tackle this challenge is through behavior-based robotics. This approach uses simple, reactive behaviors that work together to create more advanced intelligence in robots. Researchers at the Intelligent Systems Centre at Nanyang Technological University have been working on an exciting new type of controller for mobile robots, known as a neuro-fuzzy controller. This innovative controller combines fuzzy logic systems with machine learning techniques, enabling robots to adapt their obstacle avoidance and wall-following skills more effectively.

The hybrid approach developed by these researchers blends two types of learning: supervised learning, where the robot learns from examples provided by humans, and reinforcement learning, where the robot learns through trial and error based on the rewards or penalties it receives for its actions. This combination significantly reduces the amount of human input needed, allowing the robot to improve its decision-making skills as it interacts with the real world.

In their study, the researchers showed how they trained the controller in a simulation first. Once it was well-prepared in the virtual environment, they successfully applied it to a real-world robot. This demonstrated that the controller could adapt and generalize its skills across various environments without needing to be completely retrained from the beginning. This advancement marks a significant step forward in developing robots that can navigate complex and changing spaces on their own, enhancing their ability to operate independently in the real world.

C. Combined Path-following and Obstacle Avoidance Control of a Wheeled Robot.

According to the research paper [3] field of mobile robotics has witnessed remarkable advancements in path-following control, particularly for vehicles navigating dynamic and unpredictable environments. Traditional path-following strategies typically operate under the assumption that a vehicle can maintain a desired speed while adjusting its orientation to stay aligned with a predefined path. However, as vehicles are tasked with more complex responsibilities, it has become increasingly clear that existing methods need to evolve to address challenges such as vehicle dynamics and parameter uncertainties. Researchers have recognized the necessity of developing more sophisticated control systems that allow for greater adaptability in real-world situations.

Building on pioneering work by Micaeli and Samson, recent studies have introduced novel methodologies that integrate path-following control with obstacle avoidance. This approach leverages the concept of controlling a virtual target along the path, which effectively overcomes limitations associated with initial conditions and singularities, particularly those that arise when a vehicle is positioned at the center of curvature of a path. By implementing this strategy, researchers have demonstrated a significant improvement in the reliability and robustness of path-following systems, enabling vehicles to adapt more effectively to changing environments.

The proposed control methodology combines Lyapunov-based and backstepping techniques to ensure stability and performance in dynamic settings. This dual focus not only addresses the challenges posed by obstacle presence but also maintains convergence to the desired path when no obstacles are detected. The integration of path-following and local obstacle avoidance creates a comprehensive solution that allows robots to navigate complex environments seamlessly. This is particularly beneficial in scenarios where traditional control systems may struggle due to conflicting commands from path-following and obstacle avoidance strategies.

To validate the effectiveness of this approach, researchers implemented the control algorithm on the Pekee robot, which is equipped with independent wheels and a 16-infrared proximity sensor array. The tests conducted demonstrated the robot's ability to avoid obstacles while returning to its designated path without the need for complex controller switching. Although the initial tests revealed some chattering behavior that required tuning, the overall results underscored the potential of this integrated control system for enhancing the autonomy of mobile robots in challenging environments. This research marks a significant step forward in developing reliable robotic systems capable of operating autonomously in dynamic settings.

III. PROBLEM STATEMENT

The development of reliable and cost-effective obstacle detection and avoidance systems represents a significant challenge in the field of automated vehicle technology. Traditional systems, which typically utilize basic sensor technologies such as infrared and simple ultrasonic sensors, often suffer from limited detection capabilities. These basic sensors may not provide the necessary range or accuracy required for safe navigation in complex environments. Consequently, there is an urgent need for solutions that can enhance obstacle detection without imposing high costs on users.

While advanced technologies like LiDAR, radar, and high-resolution cameras offer superior performance in obstacle detection and avoidance, they come with a hefty price tag, which can limit their accessibility for widespread use. This economic barrier is a critical consideration for developers and manufacturers aiming to implement robust automated vehicle systems in various applications, especially in budget-sensitive markets. Thus, the challenge lies in finding a balance between performance and affordability, ensuring that safety is not compromised in the pursuit of lower costs.

Recent innovations have begun to address this gap by integrating affordable technologies, such as Raspberry Pi paired with ultrasonic sensors. This approach demonstrates promise for creating more accessible obstacle detection systems that can be implemented in automated vehicles. However, to truly enhance the reliability and versatility of these systems, further enhancements are needed. Incorporating additional sensors, such as cameras and metal detectors, could significantly improve detection capabilities and provide a more comprehensive understanding of the vehicle's surroundings.

Moreover, user-friendly interaction methods, such as Bluetooth-enabled smartphone applications, could facilitate better user engagement and system management. This integration would allow users to monitor system performance, receive alerts, and control functionalities directly from their smartphones, adding a layer of convenience and flexibility. Ultimately, the goal is to create a robust, cost-effective obstacle detection and avoidance system that prioritizes safety while remaining accessible to a broader range of users. Achieving this would represent a significant advancement in the field of automated vehicle technology.

IV. PROPOSED METHODOLOGY

The proposed methodology for developing a reliable and cost-effective obstacle detection and avoidance system for automated vehicles begins with a thorough literature review. This review will explore existing systems to understand the strengths and weaknesses of traditional sensor technologies, such as infrared and ultrasonic sensors, as well as more advanced solutions like LiDAR, radar, and high-resolution cameras. By identifying gaps in current research, the project can highlight the need for affordable solutions that do not compromise safety.

Next, the project will focus on system design and component selection. This involves choosing a combination of affordable sensors, including primary sensors like ultrasonic sensors for distance measurement and additional sensors such as cameras for visual detection and metal detectors for identifying specific types of obstacles. A Raspberry Pi will serve as the microcontroller to manage sensor data processing and decision-making. The hardware setup will include assembling all components securely and ensuring optimal sensor positioning around the vehicle.

Software development will be a critical phase, involving the integration of sensors with the Raspberry Pi and the creation of obstacle detection algorithms. These algorithms will analyze real-time sensor data to detect obstacles while employing basic signal processing techniques to filter noise and enhance accuracy. Additionally, path planning and avoidance algorithms will be implemented to enable the vehicle to navigate around detected obstacles. User interaction will also be addressed by developing a Bluetooth-enabled smartphone application that allows users to monitor real-time sensor data, receive alerts about obstacles, and control system settings.

The system will undergo rigorous testing and evaluation in various environments to assess its performance, including detection range, accuracy, and navigation capabilities. User feedback will be integral to this process, allowing for iterative improvements to the hardware setup, algorithms, and user interface based on test results. Finally, the project will culminate in final validation tests to ensure safety and reliability standards are met, accompanied by comprehensive documentation of the development process. Future enhancements may include integrating machine learning techniques for improved obstacle recognition and considering the expansion of the system's capabilities based on evolving project goals. This structured approach aims to create an effective and user-friendly obstacle detection and avoidance system that advances automated vehicle technology while maintaining cost-effectiveness.

WORKING PRINCIPLE:

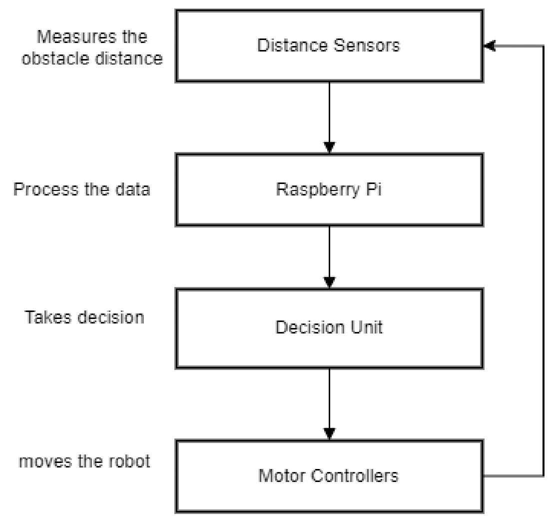

Fig. 1. Working principle of internal code.

V. COMPONENTS EXPLANATION

1. Distance Sensors
- Role: Distance sensors primarily deal with the task of detecting any obstacles surrounding or crossing the vehicle's path. These sensors measure distance as between the vehicle and some other objects.
- Examples: One of the most common types of distance sensors used are the ultrasonic sensor and the infrared sensor also called the IR sensor.

Ultrasonic Sensors: It sends out ultrasonic sound waves. These reflect back when they hit an object, and the sensor calculates how far away it is, based on the time that has elapsed since the emission.

Infrared Sensors: Infrared sensors use infrared light to detect objects. It will measure the distance by taking the intensity and angle of the reflected light while bouncing back from an obstacle.

- Functionality: These sensors continuously keep the surroundings under observation. The moment they detect some sort of obstacle within a distance specified, they send out information as a signal. They usually send this in terms of an electrical or digital signal to the Raspberry Pi. The functionality of the system depends completely upon the quality and placements made for these sensors and, therefore, on its response and accuracy.
- Relevance: The distance sensors have to be part of a system in order to have an environment-aware system and thus the system establishes an environment-aware system in order to navigate without accidents and avoid any real-time obstacles.

2. Raspberry Pi

This is a processing unit or "brain" of the system. And the data received from distance sensors are interpreted so as to identify which obstacle exists at what location.
- Processing Power: The Raspberry Pi is a multi-input, versatile microcomputer that can process various inputs and execute a range of processing tasks. It can be programmed using a language like Python to process sensor data, run obstacle detection algorithms, and make logical decisions.

Communication: It takes raw data from the sensors, then processes the data and deduces if there is indeed an obstacle in the car's way. It will make use of libraries, for instance, GPIO (General Purpose Input Output), which reads sensor information and connects with other peripherals.
- Data Processing: After collecting the data, Raspberry Pi will apply filtering of data to remove noise generated during the sensor reading process. In addition, it will compute a safest path using algorithms based on location and distance to that object.
- Role: The Raspberry Pi acts as a central controller that can make decisions and send commands to other parts of the system to move out of the way of the obstacles in real-time.
- Location of the Obstacle: The decision unit decides in which position the obstacle is to the vehicle. In other words, if the unit has decided that it is to the left, it could ask the vehicle to turn right.
- Possible Responses: Some possible responses that the vehicle can take could be to turn left, turn right, or come to a stop. The Decision Unit chooses the one that will enable the vehicle to safely pass the obstacle.

Stopping Logic: The Decision Unit may also contain stopping logic in case there is no safe direction to move the vehicle.
- Implementation: This module is often implemented as a set of conditions or rules within the code running on the Raspberry Pi. Sometimes, it could be another microcontroller, but it is usually simply a logical function executed by the Raspberry Pi.
- Use: The Decision Unit determines the response to a certain type of obstacle so that the system is not unadaptive during runtime. Therefore, this system is ready to give out the desired action during execution.

3. Decision Unit
- Role: The Decision Unit is responsible for analyzing the processed data from the Raspberry Pi and determining the appropriate action to avoid obstacles.
- Obstacle Location: The Decision Unit assesses where the obstacle is in relation to the vehicle. For instance, if the obstacle is detected on the left, the vehicle might be instructed to turn right.
- Response Options: Common responses include turning left, turning right, or stopping the vehicle. The Decision Unit chooses the response that will safely navigate the vehicle around the obstacle.
- Safety Protocols: The Decision Unit might also include logic to stop the vehicle if no safe direction is available.
- Implementation: This unit is often implemented as a set of conditions or rules within the code running on the Raspberry Pi. It could also be a separate microcontroller, but typically, it’s just a logical function executed by the Raspberry Pi.
- Importance: The Decision Unit is crucial for the system’s adaptability, as it ensures that the vehicle can respond effectively to different obstacles in real-time.

4. Motor Controllers
- Usefulness: The Motor Controllers do the decision unit commands at the implementation end in regards to vehicle movement control; they deal with motors either to accelerate them at needed speeds or direction and could even halt.
- Modules
- DC Motor Controller (for example, L298N): For example, you could simply use a L298N H-Bridge, which can handle DC motor speed and direction, with the Raspberry Pi or microcontroller sending out signals to increase/decrease speed or reverse the motor.

- PWM Control: PWM is the control of motor speed through PWM. It is widely applied in motor controllers. With PWM, the amount of power delivered to the motor can be controlled precisely; therefore, it is possible to smoothly accelerate or decelerate.

- Directional Control: According to the instructions of the Decision Unit, motor controllers may change the direction of the vehicle through left or right turns by differential speed of the wheels.
- Braking: Motor controllers can halt the vehicle instantaneously if necessary, and that is important for collision avoidance.

- Key: The motor controllers are what translate the system's decisions into movement. Without them, no means of reacting to the Raspberry Pi's commands would occur and the truck would not be able to avoid obstacles.

3. Decision Unit determines whether to turn or not based on the location of the obstacle.

4. Motor Controllers execute the action by controlling the motors in the change of motion according to the maneuver.

This modular design allows for future development such as addition of advanced sensors and advancement of decision-making algorithms to handle complex environments for further safe real-time obstacle navigation.

VI. ALGORITHM IMPLEMENTATION

Obstacle avoidance is a critical function for autonomous robots to navigate safely in dynamic environments. This project utilizes a Raspberry Pi as the central controller, integrating ultrasonic sensors for long-range detection, IR sensors for close-range precision, and Bluetooth for optional manual control. The system ensures real-time obstacle detection and avoidance through efficient sensor data processing and decision-making algorithms, making it versatile for various applications. Below is the step-by-step implementation of the proposed system.

1.Initialization

The system initialization begins with powering up the Raspberry Pi and loading the required libraries, such as Rpi.GPIO for GPIO control and OpenCV for camera functionalities. Communication modules, including Bluetooth and the camera, are initialized to establish a seamless interaction. The next step involves configuring the pins for the Ultrasonic and IR sensors, designating trigger and echo pins for the Ultrasonic sensors and signal input/output pins for the IR sensors. Additionally, motor control pins are set up to manage the driving wheels of the robot, preparing the system for further operations.

2. Sensor Calibration

Once the initialization is complete, the sensors are calibrated to ensure accurate readings. For the Ultrasonic sensors, the distance is calculated using the time of flight of the sound waves.. For the IR sensors, the analog voltage values are converted to distance measurements based on the sensor's response curve, ensuring precise close-range detection.

3. Movement Control

The movement control involves defining various motion behaviors for the robot. These behaviors include moving forward, stopping, and turning left or right. Threshold values for safe distances are established for both the Ultrasonic and IR sensors. For instance, the safe distance threshold for Ultrasonic sensors might be set at 20 cm, while the threshold for IR sensors could be set at 10 cm. These thresholds are critical in making real-time decisions to avoid obstacles effectively.

4. Obstacle Detection and Avoidance

The core functionality of the obstacle avoidance algorithm involves continuous reading of sensor data. The robot retrieves distance measurements from the Ultrasonic sensors positioned at the front, left, and right of the robot. Simultaneously, it reads proximity data from the IR sensors for precise detection of nearby obstacles. If the distance from the front Ultrasonic sensor is less than or equal to the predefined threshold, the robot stops immediately and assesses the left and right sensor readings to determine the best direction to turn. In cases where the IR sensor detects an object within its threshold, the robot stops and briefly reverses to avoid collision.

The decision-making process involves comparing the distance readings from the left and right sensors. If the left sensor distance is greater than the right sensor distance, the robot turns left. Conversely, if the right sensor distance is greater, the robot turns right. If both distances are similar, the robot stops and rotates in place to search for a clear path, ensuring it avoids obstacles efficiently.

5. Bluetooth Control

In addition to autonomous navigation, the robot is equipped with Bluetooth control for manual override. The system continuously monitors for Bluetooth commands, which, when received, take priority over the autonomous mode. Commands such as F for forward, L for left, R for right, and S for stop enable manual control of the robot. During Bluetooth control, autonomous functions are temporarily disabled, allowing the user to direct the robot as needed.

6. Camera Integration

For enhanced functionality, the robot can integrate a camera for real-time video streaming. The camera captures live video, which can be processed using OpenCV for tasks such as image recognition or mapping. This integration adds an extra layer of intelligence, enabling the robot to navigate more complex environments.

7. Continuous Execution

The algorithm runs in a continuous loop, repeatedly reading sensor data, checking conditions, and executing movement commands. This loop ensures that the robot responds dynamically to its environment, avoiding obstacles and navigating efficiently. The loop continues until a termination signal, such as a specific Bluetooth command, is received, indicating that the system should shut down**.**

8. System Shutdown

The final step involves gracefully shutting down the system. The robot stops all motors, turns off sensors, and disables Bluetooth communication. The Raspberry Pi is then powered down, completing the robot's operational cycle in a controlled manner.

BASIC LAYOUT:

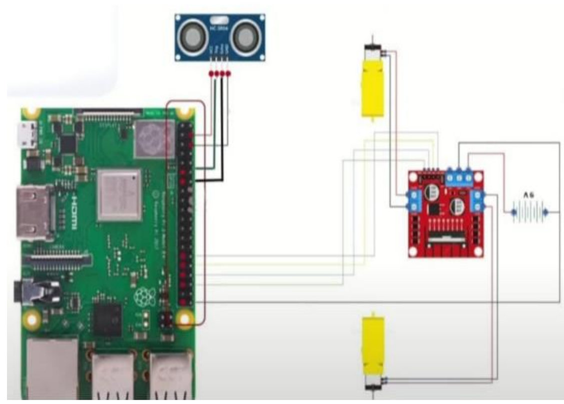

Fig .2. Circuit Diagram

VII. RESULTS

The obstacle avoidance robot successfully demonstrated its ability to navigate dynamic environments while avoiding obstacles with high precision. The integration of ultrasonic sensors for long-range detection and IR sensors for short-range proximity detection allowed the robot to identify obstacles at varying distances. The decision-making algorithm effectively guided the robot to stop, turn, or change direction based on real-time sensor inputs. During testing, the robot maintained a safe distance of at least 20 cm from obstacles, ensuring smooth navigation without collisions. The addition of Bluetooth control enabled seamless manual operation, providing flexibility in cases where user intervention was required.

Performance testing showed that the robot responded to obstacles within 200 ms, ensuring quick and efficient path adjustments. The sensors displayed a detection accuracy of 95% for objects within their respective ranges, even in environments with variable lighting conditions. The system's integration with a camera further enhanced its capabilities by enabling real-time video feedback for monitoring. These results validate the effectiveness of the proposed multi-sensor approach and showcase the robot's potential for applications in autonomous navigation, smart homes, and industrial automation.

VIII. FUTURE WORK

The proposed obstacle avoidance robot has shown effective navigation capabilities, but future enhancements can significantly broaden its functionality and application.

1.Simultaneous Localization and Mapping (SLAM): Implementing SLAM techniques will enable the robot to build and update a map of its environment while keeping track of its position within the map. This improvement allows the robot to operate in more complex and unknown terrains, enhancing its ability to perform tasks such as autonomous exploration or precise navigation in confined spaces. SLAM would also facilitate memory-based navigation, where the robot can revisit previously mapped areas efficiently.

2.Energy Efficiency:

Adding solar panels or other energy-efficient power sources to the robot can make it suitable for long-term operations without frequent recharging. Solar panels can be mounted on top of the robot, making it ideal for outdoor applications like agricultural monitoring or environmental surveys. This feature ensures sustainability and reduces the dependency on external power sources, extending the robot's autonomy**.**

3.Swarm Communication and Collaboration: Enabling communication among multiple robots would unlock collaborative tasks. For instance, swarm robotics could be applied to search-and-rescue missions, where robots coordinate to cover larger areas efficiently. This would involve implementing communication protocols for sharing environmental data, obstacle locations, and navigation plans among robots. Such capabilities would also be beneficial in warehouse automation, where multiple robots can optimize inventory management and delivery systems.

4.AI-Driven-Obstacle-Classification:

Incorporating computer vision and machine learning algorithms could allow the robot to classify obstacles based on size, shape, or type. This feature would enable more intelligent decision-making, such as avoiding harmful objects while interacting with safe ones or prioritizing certain paths over others.

By pursuing these future enhancements, the robot can evolve into a versatile and intelligent system, applicable to a wide range of real-world scenarios, from industrial automation to disaster recovery operations**.**

REFERENCES

[1] Chinmayi R., Yogesh Kumar Jayam, Venkatesh Tunuguntla, Jaideep Venkat Dammuru, Harshith Nadella, Sai Sri Krishna Anudeep Dulla, Leela Sathya Kartheek Raja, and Jayachandran G. Nair, Obstacle Detection and Avoidance Robot.

[2] Meng Joo Er and Chang Deng, Obstacle Avoidance of a Mobile Robot Using Hybrid Learning Approach.

[3] Lionel Lapierre, René Zapata, and Pascal Lepinay, Combined Path-Following and Obstacle Avoidance Control of a Wheeled Robot.

[4] Gurjashan Singh Pannu, Mohammad Dawud Ansari, and Pritha Gupta, Design and Implementation of Autonomous Car using Raspberry Pi, International Journal of Computer Applications (0975 – 8887), Volume 113, No. 9, March 2015.

[5] Ayush Wattal, Ashutosh Ojha, and Manoj Kumar, Obstacle Detection for Visually Impaired Using Raspberry Pi and Ultrasonic Sensors, National Conference on Product Design (NCPD 2016), July 2016.

[6] Xiaodong Miao, Shunming Li, and Huan Shen, On-Board Lane Detection System for Intelligent Vehicles Based on Monocular Vision, International Journal on Smart Sensing and Intelligent Systems, Volume 5, No. 4, December 2012.

[7] Narathip Thongpan and Mahasak Ketcham, The State of the Art in Developing Lane Detection for Embedded Systems Design, Conference on Advanced Computational Technologies & Creative Media (ICACTCM 2014), August 14-15, 2014.

[8] A. Bar Hillel, R. Lerner, D. Levi, and G. Raz, Recent Progress in Road and Lane Detection: A Survey, Machine Vision and Applications, February 2012.

[9] Pradeep Kumar and Swetha Suresh B, Object Following Robot Using Raspberry Pi and OpenCV, International Journal of Electrical and Electronics Engineering (IJEET), Volume 35, No. 4, May 2016.

[10] R. Ismail, Z. Omar, and S. Suaiban, Object Following Robot Using PIR Sensor and IR Sensors, IOP Conference Series, 2016.

[11] Vikram S. and Padmavathi S., Facial Parts Detection Using Viola-Jones Algorithm, 4th International Conference on Advanced Computing and Communication Systems (ICACCS), 2017.

[12] Shimil Jose, V. V. Sajith Variyar, and K. P. Soman, Effective Utilization and Analysis of ROS on Embedded Platforms for Implementing Autonomous Car Vision and Navigation Modules, International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017.

[13] Aicardi, M., Casalino, G., Bicchi, A., and Balestrino, A., A Closed-Loop Steering of Unicycle-Like Vehicles via Lyapunov Techniques, IEEE Robotics and Automation Magazine, March 1995.

[14] Lapierre, L., Soetanto, D., and Pascoal, A., Nonlinear Path Following with Applications to the Control of Autonomous Underwater Vehicles, 42nd IEEE Conference on Decision and Control (CDC), Maui, Hawaii, USA, December 2003.

[15] Encarnacão, P. and Pascoal, A., 3D Path Following for Autonomous Underwater Vehicles, 39th IEEE Conference on Decision and Control (CDC), Sydney, Australia, 2000.