Self-Driving RC Car

R. Sai Venkat
Assistant Professor,
Department of IT,
Chaitanya Bharathi
Institute of Technology
Hyderabad, India
saivenkatr_it@cbit.ac.in

Karthik Reddy
Student,
Department of IT,
Chaitanya Bharathi
Institute of Technology
Hyderabad, India
pesarukarthikreddy1@gmail.com

Saketh Raju
Student,
Department of IT,
Chaitanya Bharathi
Institute of Technology
Hyderabad, India
sakethrajualluri@gmail.com

Varun Singh
Student,
Department of IT,
Chaitanya Bharathi
Institute of Technology
Hyderabad, India
the.varun216@gmail.com

***Abstract*— The advent of autonomous vehicles has revolutionized the way people commute, providing a safer and more cost-effective alternative to traditional modes of transportation. In this project, we present a prototype for a self-driving RC car that utilizes a Raspberry Pi and IOT sensors for self-navigation. The project aims to demonstrate the feasibility of self-driving technology in a miniature car setting and the potential for its scalability to full-sized vehicles. The self-driving RC car is designed to reduce the cost of commuting and decrease the number of accidents caused by human error. The Raspberry Pi serves as the brain of the car, processing sensor data and executing commands to control the car's movements. The car is equipped with IOT sensors, including ultrasonic sensors and a camera, that allow it to detect obstacles and navigate around them autonomously. The RC car prototype demonstrates the potential for self-driving technology in real world applications. The use of IOT sensors enables the car to gather real-time data on its surroundings and make informed decisions about its movements. This technology has the potential to reduce accidents caused by human error, as the car can react to its surroundings more quickly and accurately than a human driver. Furthermore, the self-driving RC car prototype demonstrates the potential for cost savings in transportation. The use of self-driving technology has the potential to reduce the need for human drivers, which could lead to lower labour costs and reduced fares for passengers. This technology has the potential to make transportation more accessible and affordable for people around the world. In conclusion, the self-driving RC car prototype is a promising demonstration of the potential for self-driving technology in the transportation industry. With the continued development of this technology, we may see a future where autonomous vehicles are the norm, providing safer, more efficient, and more cost-effective transportation for people around the world.**

***Keywords*— Self driving, RC Car, IOT, Self-navigation.**

I. INTRODUCTION

The concept of self-driving cars has gained widespread attention over the past few years. The self-driving car is an innovative approach to transportation that aims to reduce the cost and increase the safety of commuting. The self-driving car is designed to operate without human intervention, using advanced technologies such as machine learning algorithms, artificial intelligence, and computer vision. This project report describes the design and implementation of a prototype self-driving RC car, which is based on the use of a Raspberry Pi and IOT sensors. The self-driving RC car is a miniature model of a car that is designed to navigate itself without human intervention. This model car uses various sensors and cameras to detect obstacles, traffic lights, and other road conditions to navigate on its own. The primary objective of this project is to develop a self-driving RC car that can serve as a prototype for a larger model of a self-driving car. The self-driving RC car is equipped with various sensors, including ultrasonic sensors, infrared sensors, and a camera module, which allows it to navigate itself autonomously. The Raspberry Pi microcontroller is used to control the car's movements, and the IOT sensors are used to collect data about the car's surroundings. The implementation of a self-driving RC car has several advantages over conventional cars. Firstly, it reduces the cost of commuting as it eliminates the need for a human driver. Secondly, it increases safety by reducing the number of accidents caused by human error. Thirdly, it provides an efficient way of transportation, as the car can navigate itself using advanced technologies. In conclusion, this project report describes the design and implementation of a self-driving RC car prototype, which is based on the use of a Raspberry Pi and IOT sensors. The self-driving RC car serves as a prototype for a larger model of a self-driving car that has the potential to revolutionize the transportation industry. The implementation of this technology has several benefits, including cost reduction, increased safety, and efficient transportation.

II. LITERATURE SURVEY

The development of Self driving RC Car has been credited to numerous writers. The following list of recent inventions that have been put out in papers:

In paper [1] the author describes Self-Driving car as a car capable of sensing its surrounding and moving on its own through traffic and other obstacles with minimum or no human input. This is the current upcoming technology in the automobile industry and even though it has been discussed and worked on for a long time, it was successfully manufactured by TESLA. In recent years, these cars began to roll out in foreign markets as private and public vehicles (taxis etc.). Many companies like Waymo, UBER, Nissan, Nvidia are involved in this product development. With this type of car, the whole automotive transportation's safety, security, efficiency is increased, and the human errors can be eradicated whilst the drive is made to its best. This project has infused the idea of traffic signal responding which is absent in the current models and the above-mentioned advantages can be achieved with much more ease and at a low cost. This type of system can bring a revolution in transporting for differently abled people and help blind people travel independently. This project plans to planning to provide a self-driving car with a system that can navigate between two places on the map, detect any obstacles, lane detection, accident avoidance and emergency services. And our project's uniqueness is we are implementing traffic signal responding which is not present in Tesla and other companies' car. But the system key fault is that it is fully autonomous car and hence does not pass the legal system of India.

In paper [2] the author states that Nowadays, technology of self-driving cars become important as it will make the transportation more brilliant and safer. This technology involves many aspects of science such as computer vision, artificial intelligence, deep learning control system, image processing and many of other technology. The thing which is more interesting is how the car can control itself to follow the right path. Self-driving cars is a complicated system. It needs a massive information to know and understand all the process of designing and making a self-driving car. However, the one can dive and study the aspect of controlling the self-driving car. Designing a control system for a self-driving car requires some important information. There are many different algorithms that are used to design a control system such as proportional integral derivative (PID) controller, model reference adaptive controller (MRAC), Model Predictive Control (MPC), Fuzzy Logic Control and many others.

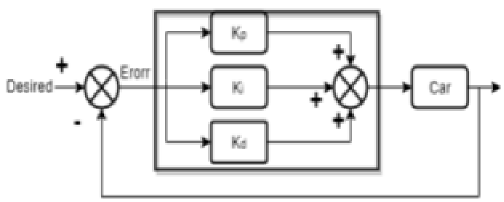

Figure 1. PID Controller

The output of the car is subtracted from the desired, then the error multiplied by Model reference adaptive Controller. After that the car will take the output of the PID controller as an input. Those steps go in a cycle till the error become Zero. This is very simple explanation of a PID controller.

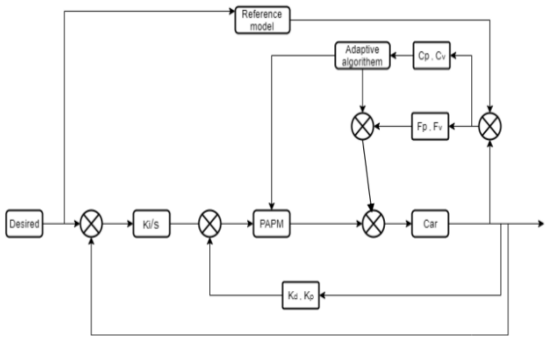

Figure 2 MRAC

In MRAC controller, first there is an input to the system to get the desired path. Then the output of the car goes on comparison with the model reference. After subtraction, the error will go as an input to the adaptive algorithm. The adaptive algorithm will result in an output used as an input to the car. At the same time, the output of the car will be subtracted from the desired path. The error will be multiplied by the integral controller Ki then, the result will be subtracted from a feedback Kp, Kd to control the velocity and position. The output will be multiplied by the output of the adaptive algorithm, the result will go in Precaution Adoption Process Model (PAPM). Then, the output will be added to the output of the adaptive algorithm again. The result will be the input to the car. These loops will continue till the output of the reference model becomes 0. For Fp, Fv, Cp and Cv. They are used to balance the car stability. Those two controllers are combined result in a PID+MRAC controlled. The important thing of this controller is that the error will be an input to the PID controller and the adaptive algorithm, these two outputs will be multiplied, and the result will be added to the adaptive algorithm output again. All these explanations are so simple for more information and math derivation.

In paper [3] the authors describe self-driving Car technology as a vehicle that guides itself without human conduction and states that "The first truly autonomous cars appeared in the 1980s with projects funded by DARPA (Defence Advance Research Project Agency). Since then, a lot has changed with the improvements in the fields of Computer Vision and Machine Learning. We have used the concept of behavioural cloning to convert a normal RC model car into an autonomous car using Deep Learning technology". Behavioural cloning is a method by which sub-cognitive skills like -recognizing objects, experience while performing an action can be captured and reproduced in a computer program. The skills performed by human actors are recorded along with the situation that gave rise to the action. The input to the learning program is the log file generated from the above data collection. The learning program outputs a set of rules that reproduce skilled behaviour. The application of this method can be to construct automatic control systems for complex tasks for which classical control theory is inadequate.

The implementation comprises of Hardware stack which consists of the RC Car, Raspberry Pi model 3B, Pi Camera, jumper wires and the Software Stack consists of Raspbian OS, Keras, CV2 and gpio library of python. It consists of a normal RC car, which is controlled wirelessly using the remote. By reverse-engineering its remote and soldered the-forward, reverse, right and left pins with male to female jumper wires 8 such that it can relate to the appropriate gpio pins of the Raspberry Pi. After this, the Raspberry Pi relates to the PC and stored the weight file for prediction. The Pi Camera is placed on the car and the images are sent to the Raspberry Pi. The Pi then signals the next command-straight, right and left according to the prediction of the Deep Learning model. The Software Implementation is divided into three main sections data generation, training phase, and testing phase. In data generation, the Pi Camera module records video feed and generates images of resolution 648 X 648. A .csv data file is generated along with the steering details (0: straight, 1: right and -1: left) with respect to images in each frame.

In training phase, the video feed recorded by the camera was of resolution 648 X 648. For training, we kept the batch size to 32. We used CNN sequential layers for training and compiled the model file. Sequential layers are basically a linear stack of layers. The first layer added to sequential layer used is a two-dimensional convolution layer. We also used the activation function called “eLu” to the first layer. ELU is very similar to RELU except for negative inputs. The second layer added was a MaxPooling layer. The pooling layer basically reduces computation by sending inputs with fewer parameters. The third layer was to flatten the inputs from previous layers. The fourth, fifth, sixth layers are dense layers. For the third and fourth layers we used ReLu activation function. For the fifth layer, SoftMax activation function was used for multiclass classification. We split the dataset using pandas and sklearn libraries of python. The dataset split was of the ratio 80/20 where 80% is for training and 20% for testing. Adam optimizer is also required during the training process and used “mean-square error” method to reduce the loss, as a loss function. We set epoch as 20 and a batch size of 8 for training. The car performed satisfactorily. But there was an observable lag due to the low computing capability of Raspberry Pi model 3b. Hence, the steering decisions made by the car, although correct were not made in a suitable time frame. Thus, this method is suitable for training and implementing an autonomous car for a set environment, that does not change unpredictably.

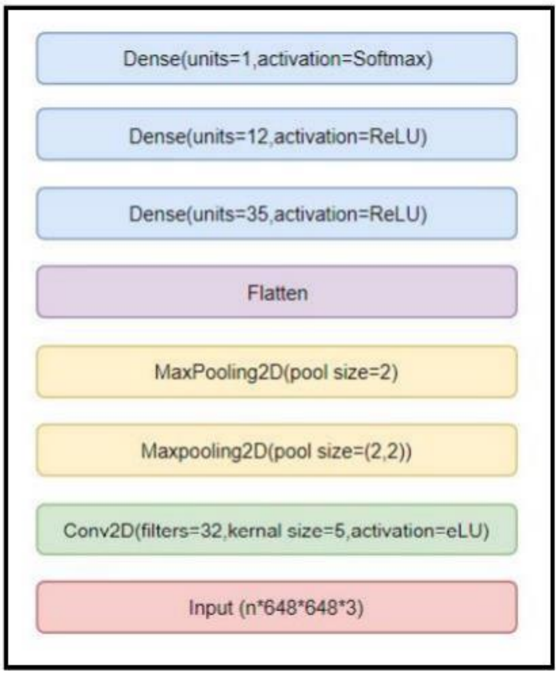

Figure 3 architecture of network

In paper [4] The methodology used in the paper describing the self-driving RC car with adaptive cruise control, lane assist, and parking assist can be divided into several key steps. Firstly, the researchers designed and built the prototype car, incorporating the necessary hardware and software components required for the autonomous driving features. Next, the team conducted extensive testing and evaluation of each feature separately, as well as in combination, to ensure their proper functioning and accuracy. They also collected data on the performance of the car in various scenarios and used this information to refine and improve the algorithms used for each feature. Finally, the team validated the performance of the self-driving car using real-world scenarios, demonstrating its effectiveness in navigating complex driving situations. Overall, the methodology used in this paper provides a comprehensive and systematic approach for developing and testing autonomous driving features in a prototype RC car. In conclusion, there are many strong socio-economic motivators for the adoption of autonomous vehicles. Human safety, infrastructure efficiency, quality of life and a ready customer base are just a few of the key factors that will help make self-driving cars a reality. Technology is converging rapidly, both incrementally from existing vendors and from new entrants. A car equipped with existing systems can take in more information quickly and reliably, and then process it to implement a correct decision about a complex situation. Yet to be solved are the complex issues associated with the legal and liability infrastructure. Gradual introduction of these features combined with strong economic motivators are sure to overcome such obstacles.

III. PROPOSED METHODOLGY

A. Hardware and Software Requirements:

- Python: Python is used as the programming language for a self-driving RC car due to its versatility, ease of use, and large community of developers.
- Micro Python: Micro Python is lightweight and efficient, making it ideal for use in embedded systems such as RC cars.
- RPi Module: The Raspberry Pi (RPi) module can be a powerful tool for programming a self-driving RC car.
- Servo Motor: It will be used to control the steering mechanism of the car, allowing it to turn left or right as required. The servo motor receives signals from the car's onboard computer system, which then adjusts its angle of rotation to direct the wheels in the desired direction.
- Geared Motors: Geared motors are used to control the movement and steering of the vehicle. Typically, two of the geared motors are used to drive the rear wheels, while the other two are used to steer the front wheels.
- Arduino: Arduino is used to provide the necessary computing power and programming capabilities to control the car's movements and functions.
- Motor controller: The motor controller receives signals from the car's sensors and sends appropriate commands to the motors to control the car's speed and direction.
- GPS: GPS is used to enable them to navigate through their surroundings autonomously.
- Camera: The camera is used to capture images of the car's surroundings.
- Ultrasound sensor: The sensor emits high-frequency sound waves that bounce off objects in the car's path and return to the sensor.
- IR sensor: The sensor sends out infrared signals and measures the time it takes for the signal to bounce back to the sensor.

Figure 4: Block Diagram of proposed system

B. Methodology:

Step 1: Assemble the physical components of the car, including the geared motors, servo motor, Arduino, motor controller, camera, GPS, ultrasound, and IR sensors.

Step 2: Program the Arduino and motor controller to control the movement of the car. This involves writing code to interpret signals from the ultrasound and IR sensors and adjust the speed and direction of the motors accordingly.

Step 3: Connect the Arduino to the Raspberry Pi module and program the Raspberry Pi to receive data from the car's sensors and control its movements.

Step 4: Use computer vision libraries to process the images from the camera and identify obstacles and other objects in the car's path.

Step 5: Integrate GPS data to improve the car's ability to navigate to specific locations.

Step 6: Develop a machine learning algorithm to train the car to recognize different objects and respond appropriately (e.g., stop at stop signs, avoid pedestrians, etc.).

Step 7: Test the car. The car will be moving forward if the road is empty without traffic. In case of any intruder or obstacle, the car will slow down and check at left directions and right direction and take the path which has a clear, in case of both clear routes, it will check the traffic and choose the relatively lower one. If the car is blocked from all the ways, it will halt the journey.

Step 8: Continuously test and refine the car's performance in different environments and scenarios, adjusting the code and hardware as necessary.

C. Algorithm:

The algorithm used in the proposed system is A* algorithm which is a popular search algorithm used to find the shortest path between two points in a graph. The algorithm works by using a heuristic function to estimate the distance between the current node and the goal node, along with the actual cost of getting to the current node. This makes it a popular choice for pathfinding in autonomous vehicles, including self-driving RC cars. To use the A* algorithm in a self-driving RC car, the first step is to create a map of the environment. This can be done using sensors such as cameras and ultrasonic sensors. The map should contain information about obstacles, road signs, traffic signals, and other relevant details. Once the map has been created, the A* algorithm can be used to find the shortest path between the car's current location and the destination. The algorithm works by evaluating each possible path, starting from the current location, and selecting the path that minimizes the total cost, which includes the actual cost of getting to the current node and the estimated cost of reaching the goal node. In the context of a self-driving RC car, the actual cost of getting to a particular node could be determined by considering factors such as the distance to the next waypoint, the speed of the car, and any obstacles in the way. The estimated cost of reaching the goal node could be determined by using a heuristic function that considers the distance to the goal, the road conditions, and other relevant factors. The A* algorithm can also be used to adjust the car's speed and direction based on the current road conditions. For example, if the 16 algorithm detects a traffic jam or roadblock, it can reroute the car to avoid the obstacle or slow down to avoid a collision. Another important consideration when using the A* algorithm in a self-driving RC car is safety. The algorithm must be designed to ensure that the car does not violate traffic laws or endanger pedestrians or other vehicles on the road. This can be achieved by incorporating safety constraints into the algorithm, such as obeying speed limits and stopping at red lights. In conclusion, the A* algorithm can be a valuable tool for pathfinding in self-driving RC cars. By using a combination of actual cost and heuristic estimates, the algorithm can help the car navigate through complex environments and adjust its speed and direction based on changing conditions. However, it is important to design the algorithm with safety in mind to ensure that the car operates in a safe and responsible manner.

IV. RESULTS AND DISCUSSIONS

The car was able to detect interventions on roads and navigate around them by checking both the left and right sides and choosing the best path for the car. The car was also able to stop if it encountered obstacles on all sides of travel, indicating a high level of safety. Through experimentation, it was found that the combination of IR and ultrasound sensors provided a robust perception system that enabled the car to navigate complex environments with ease. The discussion of the results highlights the potential of the self-driving RC car to be used as a platform for further research in autonomous driving technology. Future work could focus on improving the car's ability to detect and respond to different types of obstacles, as well as developing more sophisticated control algorithms that enable the car to operate in more complex scenarios.

V. CONCLUSION

In conclusion, the self-driving technology has seen significant progress in recent years, and developing such a car allows for experimentation with state-of-the-art tools and techniques in artificial intelligence, computer vision, and robotics. The project provides an opportunity to learn about various components, including sensors, microcontrollers, motors, and actuators, and how they work together to create a functioning autonomous vehicle. The skills learned during this project can be applied to a wide range of real-world applications in the automotive, aerospace, and robotics industries. Moreover, this project has immense potential for making driving safer and more efficient, as self-driving cars can reduce the number of accidents caused by human error and improve the overall transportation system's efficiency. With the advancements in technology and increasing demand for self-driving vehicles, this project's scope is vast and can open numerous opportunities for researchers and engineers. It is a project that combines technology, innovation, and creativity to produce something that has the potential to revolutionize the transportation industry.

VI. LIMITATIONS

The proposed model uses more battery and cannot run perfectly on uneven surfaces. Hence making it less safe on roads.

VII. FUTURE SCOPE

The project of a self-driving RC car using IoT sensors has immense future scope in the field of autonomous vehicles. With advancements in IoT technology and artificial intelligence, the scope of this project can be expanded to include more advanced sensors and algorithms to improve the car's performance. The project can be integrated with cloud-based systems, which can allow for remote monitoring and control of the car. This can be useful in situations where the car needs to be used in hazardous or dangerous environments. The project can also be used as a prototype for developing larger autonomous vehicles, such as driverless cars and trucks. With more research and development, this project can pave the way for a safer and more efficient future of transportation.

VIII. REFERENCES

[1] Self-driving Car, Dr.T. Manikandan, J.S. Jayashwanth, S. Harish and K.N. Harshith Sivatej, April 2020.

[2] Self-Driving Cars, Aiad Asaad, August 2020.

[3] 2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP) IEEE Artificial Intelligence based Self-Driving Car

[4] Self-Driving RC Car using Behavioural Cloning, Aliasgar Haji, Priyam Shah, Srinivas Bijoor, March 2021.

[5] 2018 4th International Conference on Green Technology and Sustainable Development (GTSD) Real-Time Self-Driving Car Navigation Using Deep Neural Network Truong-Dong Do, Minh-Thien Duong, Quoc-Vu Dang and My-Ha Le*

[6] Self-Driving Cars, Ratan Hudda, Clint Kelly, Garrett Long.

[7] A Study on the Obstacle Recognition for Autonomous Driving RC Car Using LiDAR and Thermal Infrared Camera 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN).

[8] 2018 IEEE Intelligent Vehicles Symposium (IV) Changshu, Suzhou, China, June 26-30, 2018 Autonomous RC-Car for Education Purpose in iSTEM Projects.