International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Steering Through Chaos Navigating Automated Mobility

Dr S Andrews[1], Rashmi Gowda[2], Nikhil M[3], O Hanumantha[4], Noel Thomas[5]

[1]*Professor, Department of Computer Science and Engineering, CMR University, Karnataka, India*
[2,3,4,5] *Student, Department of Computer Science and Engineering, CMR University, Karnataka, India*

---***---

Abstract - In recent years, the field of automated mobility has made significant advancements in self-driving vehicle technology. Even so, navigating complex environments such as roads with unexpected pedestrian traffic, uneven surfaces, unforeseen obstacles, and narrow pathways remains challenging. This paper focuses on an innovative AI-based solution to address the unique challenges of autonomous driving in Indian environments. Built on Intel's CARLA, an open-source autonomous driving simulator. Its focus on simulation-based testing ensures the system's reliability in diverse scenarios before real-world deployment, tailored specifically for Indian roads. The system utilizes reinforcement learning for optimal driving strategies with minimal data, computer vision for road signs and obstacle detection, and context-aware navigation to adapt to varying traffic conditions. Offering a promising solution for autonomous driving in regions with similar road conditions by being cost-effective, scalable, and adaptable.

***Key Words*:** Automated Vehicles, Simulation-based Software, Computer Vision, Obstacle Detection, Context-aware Navigation, CARLA simulator, Autonomous System Navigation

1. INTRODUCTION

Most current automated driving systems are designed for well-structured environments and struggle with the unpredictable conditions found on roads like those in India. Irregular traffic patterns, unmarked lanes, unexpected obstacles, and diverse driving behaviors pose challenges that existing systems aren't equipped to handle. In this work, we aim to address these gaps by developing a simulation-based testing setup focused on navigating such complex scenarios and evaluating different approaches to more adaptable and resilient driving systems.

1.1 Background and Context

One of the most demanding challenges in real-world navigation involves guiding vehicles through densely populated city environments. These areas are filled with complex situations such as busy intersections with multiple moving vehicles, the need to recognize various road signals and markings, and the requirement to respond to both routine and unexpected events such as sudden pedestrian movement, or reckless drivers. Navigating these scenarios often involves balancing competing demands, like deciding how to slow down safely for a pedestrian without risking a rear-end collision from a vehicle behind.

Progress in this field is often slowed by the high costs and logistical hurdles involved in testing on actual roads. Operating even a single test vehicle demands significant resources, and such limited setups rarely provide enough real-world variety to cover the wide range of unusual or rare situations needed for a comprehensive system check.

To address these constraints, simulated environments have become a practical and safe alternative. They allow for repeated trials in complex conditions without the risk and expense of real-world deployment. Tools like CARLA offer flexible platforms for building and testing virtual driving systems from the ground up. In this project, we use CARLA to explore the effectiveness of different driving strategies by creating a series of increasingly challenging navigation scenarios. By altering traffic levels, route structures, and weather or lighting conditions, we assess how various approaches ranging from rule-based pipelines to behavior-driven methods that perform across a range of urban challenges.

1.2 Problem Statement

Vehicles designed to drive themselves have shown strong performance in controlled settings like highways and well-marked city roads. However, once placed in the kinds of conditions found in many real-world locations, such as those common in Indian cities and towns often fall short. These environments are filled with unpredictable elements: uneven roads, potholes, unmarked lanes, stray animals, pedestrians weaving through traffic, and narrow or cluttered paths. The constantly shifting nature of such settings makes it difficult for most existing systems to respond with the flexibility and caution required. As a result, these vehicles are not yet ready for broad use in places where the road environment cannot be guaranteed to follow a set pattern.

1.3 Objective

This project aims to improve the reliability and adaptability of self-driving systems in real-world, unstructured environments like irregular roads, potholes, narrow lanes, and unpredictable traffic. By analyzing common navigation challenges and reviewing gaps in current solutions, the project proposes a modular software design focused on obstacle handling, route planning, and responsive movement strategies. Using simulation platforms like CARLA, the system will be tested across varied conditions to assess its performance and identify areas for enhancement. The ultimate goal is to build a robust virtual testing framework that supports the development of safer and more practical autonomous navigation in complex road settings.

2. LITERATURE REVIEW

2.1 Existing Systems

Several prominent automotive companies and technology providers have developed advanced systems to support autonomous driving, each focusing on different strategies and components to enhance safety, perception, and navigation across a range of driving environments.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Tesla (Autopilot and Full Self-Driving): Tesla's driver-assist technologies are among the most recognized in the industry. The Autopilot system offers features such as adaptive cruise control and lane centering, relying on a network of cameras, ultrasonic sensors, and radar to assist with highway driving. Tesla's Full Self-Driving (FSD) system is designed for more intricate scenarios, particularly in urban areas. It includes functions like automatic lane changes, stop sign and traffic light recognition, and route navigation through city streets. The system is powered by Tesla's in-house processing hardware, which handles visual data from multiple cameras to interpret surroundings and support driving decisions.

Waymo (Alphabet Inc.): Waymo has developed one of the most sophisticated autonomous mobility platforms to date. Its vehicles are outfitted with a combination of LiDAR, radar, and high-resolution cameras to capture a complete 360-degree view of their environment. The system operates using detailed pre-mapped data of its deployment areas, covering lane structures, intersections, and traffic infrastructure. Waymo's decision-making capability draws from constant updates and analysis of surrounding objects, allowing it to anticipate behaviour patterns of other road users, including cyclists and pedestrians.

General Motors (Super Cruise and Ultra Cruise): General Motors offers Super Cruise as a highway-focused hands-free driving solution. It leverages LiDAR-based map data, GPS positioning, cameras, and radar to manage speed, lane discipline, and vehicle following on selected routes. The system includes real-time driver monitoring for added safety. Ultra Cruise, GM's upcoming system, is expected to extend beyond highways and cover a larger percentage of real-world conditions, including city driving. It aims to integrate deeper sensor integration and more expansive map data for improved decision-making across varied driving conditions.

Mobileye (Intel): Mobileye specializes in vision-based solutions for both partial and full automation. Its systems primarily use camera inputs processed by EyeQ chips to interpret surroundings, including lane boundaries, vehicles, and pedestrians. One of Mobileye's key offerings is its REM (Road Experience Management) system, which builds detailed maps using data collected from everyday drivers. This enables more accurate localization and better adaptability in different road settings. Additionally, Mobileye promotes a mathematically grounded model for vehicle behaviour to ensure safety in unpredictable traffic situations.

NVIDIA Drive: NVIDIA provides an open development platform combining high-performance computing with modular tools for vehicle autonomy. It's hardware, such as Drive AGX, and accompanying software stack, Drive Works, support perception, mapping, decision-making, and vehicle control tasks. NVIDIA also emphasizes simulation for developing and testing navigation strategies in diverse and high-risk scenarios, helping teams to iterate and improve without real-world deployment. This platform enables customized development suited to varying levels of vehicle autonomy.

2.2 Gaps and Challenges

Current autonomous systems still face significant challenges in unpredictable environments. Many of these technologies rely on detailed maps for navigation, which limits their functionality in areas that are either poorly mapped or frequently changing, such as construction zones or rural roads. These systems struggle to detect irregular obstacles like animals, pedestrians, or debris, particularly in low-visibility conditions or cluttered surroundings. Poor road conditions, such as gravel or faded lane markings, further reduce the effectiveness of these systems, making them less reliable in real-world scenarios.

Navigating tight spaces like narrow streets or parking areas remains a difficult task for many autonomous platforms, which are primarily optimized for highways or well-marked city roads. In rapidly changing environments, the system's ability to respond quickly is hindered by the high data demands from multiple sensors, slowing down decision-making when immediate action is needed. While advanced sensors like LiDAR provide precise measurements, their high costs limit widespread adoption, whereas more affordable camera-based solutions often fall short in challenging situations.

2.3 Proposed Approach

Our approach combines simulation techniques, data-driven modeling, and the integration of multiple sensor inputs to develop a reliable and adaptable system. We use CARLA to evaluate the effectiveness of three distinct approaches to autonomous driving: modular pipeline, imitation learning, and reinforcement learning. In CARLA, we create controlled navigation scenarios with specific goals, progressively increasing their difficulty. These scenarios vary in terms of route complexity, traffic presence, and environmental conditions. The outcomes of these experiments provide valuable insights into how each approach performs under different circumstances.

3. SYSTEM ARCHITECTURE

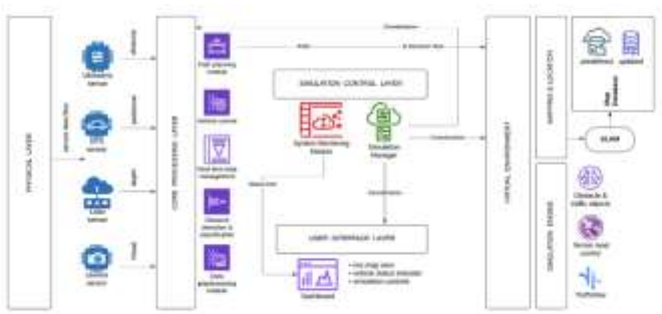

Fig -1: System Architecture

4. SIMULATION

CARLA is a high-fidelity simulation platform designed for testing self-driving technologies in virtual urban settings. Developed using Unreal Engine, it offers detailed environments with configurable parameters such as weather conditions, lighting, and traffic flow. A range of built-in towns and road layouts are available, supporting diverse scenarios that mirror real-world traffic dynamics.

In this study, CARLA was utilized to replicate driving environments that include moving vehicles, complex road geometries, and unexpected elements like pedestrians and changes in surface conditions. Virtual sensors such as cameras, LiDAR, and GPS modules were attached to the vehicle models to enable data collection for tasks involving perception, positioning, and movement control.

The simulation framework supports interaction through Python and integrates well with robotics middleware and machine learning tools. This made it possible to implement, test, and refine driving logic in a repeatable and risk-free setting, while evaluating system performance across various environmental and traffic challenges.

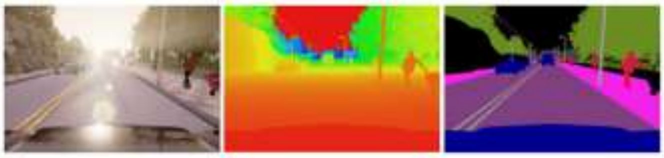

Fig -2: CARLA Sensors

5. IMPLEMENTATION

5.1 System Overview

This study makes use of CARLA, an advanced urban driving simulator designed to support the development and assessment of self-driving technologies in realistic environments. CARLA replicates cityscapes with considerable detail, including diverse road structures, pedestrians, vehicles, and varying weather conditions. To ensure a consistent framework for comparison, all methods under evaluation are designed to operate under the same setup—receiving visual data from a forward-facing camera and a navigation cue indicating the intended turn direction. Their task is to process this information and output decisions for steering, acceleration, and braking.

The simulator acts as the testing ground for three distinct approaches to autonomous navigation: a classical rules-based system, a model based on learning from driving demonstrations, and a method that refines behavior through repeated trial and feedback. Despite their differences in internal logic, all approaches share a common objective: to operate a vehicle safely and efficiently under a wide range of traffic and environmental conditions. The shared interface and structured layout of the simulation provide a level playing field, allowing for direct performance comparisons and deeper insight into each method's strengths and shortcomings.

5.2 Simulation Setup

The evaluation is carried out in two virtual towns created within the CARLA simulator. Town 1 is used exclusively for training, while Town 2 is reserved for testing. The two towns differ in layout, complexity, and road topology, enabling a clear separation between familiar and unfamiliar driving conditions. This distinction is critical in assessing whether a system can function reliably in areas beyond those it was originally exposed to.

To introduce environmental diversity, five different weather settings are included: clear daylight, overcast skies, wet roads, intense rainfall, and sunset with light rain. These variations simulate real-world driving conditions that affect visibility and road behaviour. Additionally, four levels of task complexity are defined ranging from simple straight-line driving to full city navigation that includes avoiding pedestrians and other vehicles. Each task is tested multiple times, and the vehicle must reach its destination within a time window calculated from an expected average speed.

The simulation setup also ensures the presence of dynamic elements. Vehicles and pedestrians appear unpredictably, especially in the more complex scenarios. These variables create challenges similar to those faced in actual urban driving, where road users do not always behave in predictable or orderly ways. This setup ensures that the systems are not only capable of basic navigation but also of handling unpredictable events.

5.3 Approach

The first approach follows a modular architecture that divides the driving task into three stages: detecting elements in the environment, choosing how to act based on those elements, and issuing movement instructions. The perception stage uses trained models to recognize features like lanes and intersections. A set of defined rules then decides the appropriate course of action, whether to continue forward, turn, or stop. Finally, the control system adjusts the vehicle's motion using tuning techniques that ensure smooth operation. While this method is transparent and can be inspected easily, it relies heavily on the accuracy of its visual recognition models and the completeness of its predefined rules.

The second method uses recorded examples of human driving to learn how to respond to different situations. A large collection of driving data is gathered from a human-operated vehicle in the simulator. This data is then used to train a network to associate visual input and navigation cues with correct driving responses. To prepare the system for a range of scenarios, the images are altered to simulate changes in lighting, blur, and obstruction making it less likely to be thrown off by weather changes or visual noise. The goal here is not to manually specify how to drive but to teach the system through observation and exposure.

The final method attempts to improve performance through self-guided exploration. The system begins with no prior knowledge and repeatedly attempts to complete navigation tasks. Feedback is given in the form of numerical scores encouraging safe and efficient driving and penalizing mistakes like collisions or driving off the road. Over time, the system adjusts its decisions to favor behaviors that lead to higher scores. However, because the system must discover good behavior through trial alone, and because useful feedback is rare in many situations, this method can be slow to develop effective strategies. In this study, the system was trained for the equivalent of millions of driving steps to allow patterns to emerge.

6. PERFORMANCE EVALUATION

Criteria used to measure performance, focuses not only on task completion but also on the vehicle's ability to adhere to road rules and avoid hazards. The simulation scenarios are structured to test the systems across a spectrum of challenges, from routine drives to complex urban navigation involving unpredictable elements.

6.1 Methodology

The main indicator of success is whether the vehicle reaches its destination within the time limit for each test run. This success rate reflects how reliably each approach completes assigned tasks under different weather conditions and in both towns. Additional observations such as collisions, veering off the road, and crossing lane markings are also recorded. These help to assess the safety and rule compliance of each method, which are vital for real-world driving.

Testing each system in a town it has never encountered before along with unfamiliar weather settings, offers a strong measure of adaptability. A method that performs well in this context can be considered more likely to handle unforeseen situations outside of the training environment. This setup therefore moves beyond simple benchmarking to test how broadly applicable and dependable each strategy is.

6.2 Simulation Scenario Performance

The classical modular approach performs well in straightforward conditions, particularly in the town used during training. It handles tasks like straight-line driving and single turns with a high degree of accuracy. However, in the second town and during poor weather, performance declines. This is due to its reliance on segmentation and rule systems that are not flexible enough to deal with novel layouts or visual noise.

The model trained on driving demonstrations shows solid results across a broader set of challenges. It navigates complex routes more consistently than the modular system and is less affected by lighting and weather variations. This robustness comes from the variety introduced during its training phase. Nevertheless, its performance still dips slightly when placed in the second town, suggesting that while it adapts well to visual changes, it may be sensitive to unfamiliar road layouts.

The trial-based method performs the weakest. In nearly all scenarios, the vehicle fails to reach its target, either by becoming stuck, colliding with obstacles, or driving off course. Even after extensive training, this method struggles with the structured nature of urban driving. Its reward system, designed to encourage progress and penalize unsafe behavior, often leads to confusing or ineffective results, especially when feedback is delayed or ambiguous.

7. RESULT

7.1 Analysis

Comparing the three systems reveals both the potential and the current limitations of automated driving technologies in controlled virtual settings. The modular system offers clarity and performs reliably in basic conditions. However, its rule-bound nature makes it brittle in the face of unfamiliar situations. It is suited to environments where layout and rules are well understood in advance.

The second method offers a more flexible solution. By learning from example, it avoids the need for hand-written rules and adapts better to changes in the environment. Its performance in more difficult tasks and under different weather conditions supports this conclusion. Still, it falls short in entirely new areas, revealing a tendency to learn specific routes or visual patterns rather than more abstract strategies.

The final method is promising in theory but does not deliver strong results here. The learning process is slow, and even after substantial time spent interacting with the simulation, the system remains unreliable. It frequently fails to make meaningful progress toward its goals and struggles with basic tasks like avoiding collisions. This suggests that improvements in reward design, training strategy, or input interpretation will be needed before such methods can be considered viable.

Table -1: Quantitative Evaluation

	Training Conditions	New Town	New Weather	New Town and Weather
Straight	98	97	100	80
One Turn	89	61	95	50
Navigation	86	40	94	47
Nav. Dynamic	83	38	89	44

7.2 Limitations and Future Developments

Despite the detailed environment and structured testing process provided by CARLA, none of the current systems are able to consistently handle the full range of challenges presented. Even the most successful approach falls short when asked to navigate unfamiliar roads under poor visibility. This points to a broader issue: while current methods can be tuned to perform well in known settings, they are still far from being fully adaptable or universally reliable.

Future developments should explore techniques that make better use of past experiences over time, such as models that remember sequences of actions or changes in the environment. Including different types of input such as depth estimation or external signals like GPS could also improve situational awareness. Additionally, the simulation itself could be expanded to include a wider variety of locations, interactions, and unpredictable behavior from other road users. Addressing these areas will be crucial for advancing the next generation of vehicle autonomy.

8. CONCLUSIONS

This work introduces a comprehensive simulation platform designed for the study and advancement of self-driving systems. Beyond the core software framework and protocols, the project makes available a rich set of digital resources custom-built for this environment can be reused and adapted for a variety of research and development purposes. The simulator has been employed to evaluate three different strategies for vehicle autonomy: a structured, rule-based architecture; a model shaped through behavioral observation; and another developed through iterative learning driven by outcomes. Each of these methods was tested within urban layouts populated with both vehicular and pedestrian traffic, providing a realistic and controlled setting to assess decision-making and navigation capabilities. By facilitating in-depth diagnostics and performance reviews, the simulator offers valuable insights into system limitations and design challenges, laying the groundwork for future enhancements.

REFERENCES

1. CARLA, "CARLA Simulator," [Online]. Available: https://carla.org
2. Li, X., and Wong, J. "Road Quality Detection in Autonomous Vehicles Using Convolutional Neural Networks," IEEE Transactions on Intelligent Transportation Systems, vol. 12, 6, 2021, pp. 481-492
3. An, B., and Xiao, L. "Enhanced Vision-Based Obstacle Detection for Real-Time Autonomous Navigation," International Journal of Autonomous Systems, vol. 10, 8, 2021, pp. 381-393
4. Zhu, T., and Li, W. "Design and Implementation of Simulated Road Conditions for Autonomous Vehicles," Computers in Industry, vol. 45, 12, 2021, pp. 310-325
5. H. A. Ignatious, H. El Sayed, M. Khan, "An overview of sensors in autonomous vehicles," Procedia Computer Science, vol. 198, no. 2021, pp. 736–741, 2021
6. J. Alfred Daniel, C. Chandru Vignesh, B. A. Muthu, R. Senthil Kumar, C. B. Sivaparthipan, C. E. M. Marin, "Fully convolutional neural networks for LIDAR–camera fusion for pedestrian detection in autonomous vehicle," Multimedia Tools and Applications, vol. 82, no. 16, pp. 1–24, 2023
7. Bauer, Thomas, et al. "Reference Architectures for Automotive Software." Reference Architectures for Critical Domains: Industrial Uses and Impacts. Cham: Springer International Publishing, 2022. 73-111
8. D. Parekh, N. Poddar, A. Rajpurkar, M. Chahal, N. Kumar, G. P. Joshi, et al., "A review on autonomous vehicles: Progress, methods and challenges," Electronics, vol. 11, no. 14, p. 2162, 2022
9. Dekkati, Sreekanth. "Automotive Software Engineering: Real-World Necessity and Significance." Engineering International 10.1 (2022): 33-44
10. Saha, Sujoy Kumar. "Past, Present and Future of Automotive Software Engineering." Quest, Journal of Research in Mechanical Engineering 8 (2022): 29-36
11. J. Alfred Daniel, C. Chandru Vignesh, B. A. Muthu, R. Senthil Kumar, C. B. Sivaparthipan, C. E. M. Marin, "Fully convolutional neural networks for LIDAR–camera fusion for pedestrian detection in autonomous vehicle," Multimedia Tools and Applications, vol. 82, no. 16, pp. 1–24, 2023
12. Tengilimoglu, O. Carsten, Z. Wadud, "Implications of automated vehicles for physical road environment: A comprehensive review," Transportation Research Part E: Logistics and Transportation Review, vol. 169, p.102989, 2023