

ADVANCED ADAPTIVE CRUISE CONTROL SYSTEM IN AUTONOMOUS VEHICLE

Submitted in partial fulfillment of the requirements for the award of
Master of Engineering degree in Computer Science and Engineering

By

P. MEIYAZHAGAN (Reg.No - 42310005)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

CATEGORY -1 UNIVERSITY BY UGC

**Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI -600119**

APRIL – 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

Accredited "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **P. MEIYAZHAGAN (42310005)** who carried out the Project Work entitled "**ADVANCED ADAPTIVE CRUISE CONTROL SYSTEM IN AUTONOMOUS VEHICLE**" under my supervision from November 2023 to April 2024.

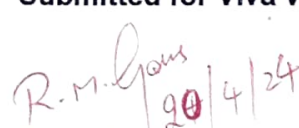

Internal Guide

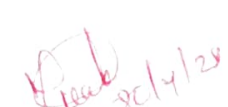
Dr.K. ASHOK KUMAR M.E., Ph.D.


Head of the Department
Dr. L. LAKSHMANAN, M.E., Ph.D.

HEAD OF THE DEPARTMENT
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Joppa Nagar, Rajiv Gandhi Salai,
Chennai - 600 119.

Submitted for Viva voice Examination held on 20/4/24


Internal Examiner


External Examiner

DECLARATION

I, P. MEIYAZHAGAN (Reg.No - 42310005), hereby declare that the Project Report entitled "ADVANCED ADAPTIVE CRUISE CONTROL SYSTEM IN AUTONOMOUS VEHICLE " done by me under the guidance of Dr.K. ASHOK KUMAR, M.E., Ph.D. is submitted in partial fulfillment of the requirements for the award of Master of Engineering degree in Computer Science and Engineering.

DATE: 20/04/2024

PLACE: CHENNAI



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the Board **of Management** of **Sathyabama Institute Of Science And Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA M.E., Ph.D., Professor and Dean, School of Computing, Dr. L. LAKSHMANAN M.E., Ph.D., Professor and Head of the Department of Computer Science and Engineering** for providing me necessary support during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.K.ASHOK KUMAR, M.E., Ph.D., Professor**, Department of Computer Science and Engineering his valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

In the realm of autonomous vehicles, achieving efficient and safe navigation remains a critical challenge. This project endeavors to address this challenge through the development of an Advanced Adaptive Cruise Control (AACC) system for a robotic mini car. The primary goal of this system is to ensure the mini car reaches its destination within a specified distance and time frame while adapting to dynamically changing environments.

The AACC system integrates a suite of sensors, including IR sensors, to detect obstacles and vehicles in the car's path. Upon detection, the system employs electric motors and a battery management system to execute appropriate maneuvers, such as deceleration or stopping, to avoid collisions. Furthermore, a WiFi-enabled microcontroller orchestrates the car's operations, receiving input parameters such as distance, start time, and end time from a cloud-based server via a WiFi IoT cloud application.

Key features of the AACC system include its ability to dynamically adjust the car's speed based on real-time sensor data and external input, thereby ensuring safe and timely navigation to the destination. Additionally, the system prioritizes safety through robust obstacle detection algorithms and fail-safe mechanisms. Through rigorous testing and evaluation, the effectiveness of the AACC system in achieving its objectives has been demonstrated across various driving scenarios. This project lays the foundation for further advancements in autonomous vehicle technology, with potential applications in transportation, logistics, and beyond.

LIST OF CONTENT

CHAPTER	TITLE	PAGE NO
	ABSTRACT	i
	LIST OF FIGURES	vi
	LIST OF ABBREVIATIONS	vii
	LIST OF TABLES	viii
1	INTRODUCTION	1
	1.1 BACKGROUND	1
	1.2 EXISTING SYSTEM: TRADITIONAL CRUISE CONTROL	1
	1.3 EXCITING SYSTEM: ADAPTIVE CRUISE CONTROL (ACC)	1
	1.4 PROPOSED SYSTEM: ADVANCED ADAPTIVE CRUISE CONTROL (AACC)	2
2	LITERATURE SURVEY	3
	2.1 OVERVIEW OF ADAPTIVE CRUISE CONTROL SYSTEMS	3
	2.2 ADVANCED TECHNOLOGIES IN AUTONOMOUS VEHICLES	3
	2.3 SAFETY BENEFITS	4
	2.4 ENVIRONMENTAL BENEFITS	4
3	REQUIREMENTS	5
	3.1 FUNCTIONAL REQUIREMENTS	5

	3.2 TECHNICAL REQUIREMENTS	5
	3.3 SOFTWARE REQUIREMENTS	6
	3.4 PERFORMANCE REQUIREMENTS	6
	3.5 USER REQUIREMENTS	7
	3.6 ENVIRONMENTAL REQUIREMENTS	7
4	SYSTEM COMPONENTS AND SPECIFICATIONS	8
	4.1 BLOCK DIAGRAM	8
	4.2 HARDWARE CONFIGURATION	9
	4.2.1 Micro Controller	9
	4.2.2 Infrared (IR) sensors	11
	4.2.3 DC Motor	13
	4.2.4 Motor Driver	15
	4.2.5 Battery	17
	4.2.6 Robotic Vehicle	19
	4.3 SOFTWARE CONFIGURATION	22
5	DESCRIPTION OF PROPOSED SYSTEM	24
	5.1 FLOW DIAGRAM	24
	5.2 IMPLEMENTATION	24
	5.2.1 Sensor Integration	24
	5.2.2 Microcontroller Programming	25
	5.2.3 Communication Interface	25
	5.2.4 User Interface Development	25
	5.2.5 Integration with Power System	25

	5.2.6 Testing and Validation	25
	5.3 EXPECTED OUTPUT	25
	5.3.1 Autonomous Navigation	25
	5.3.2 Adaptive Cruise Control Functionality	26
	5.3.3 User Interaction	26
	5.3.4 Safety and Reliability	26
	5.3.5 Efficiency and Convenience	26
	5.4 PROJECT ROBOTIC VEHICLE	27
	5.5 APPLICATION INTERFACE AND CONTROLS	28
	5.5.1 Use Of Application	28
	5.5.2 User Input and Interaction	29
6	SYSTEM RESULTS AND DISCUSSIONS	31
	6.1 TESTING	31
	6.2 TESTING OBJECTIVES	31
	6.3 TESTING METHODOLOGY	31
	6.4 TEST CASES	32
	6.4.1 Autonomous Navigation	32
	6.4.2 Adaptive Cruise Control Functionality	32
	6.4.3 Safety and Reliability	32
	6.4.4 Speed and Distance Control	33
	6.4.5 Test Results Scenarios Table 1	35
	6.4.6 Start Time and End Time Control	36
	6.4.7 Test Results Scenarios Table 2	38

7	CONCLUSION	39
	SUMMARY	39
	REFERENCES	41
	APPENDIX	44
	A SOURCE CODE	44
	B. SCREENSHOTS	62
	C. RESEARCH PAPER	67

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
4.1	Block Diagram	8
4.2.1	Micro Controller	10
4.2.2	Infrared (IR) sensors	12
4.2.3	DC Motor	14
4.2.4	Motor Driver	16
4.2.5	Battery	19
4.2.6	Robotic Vehicle	21
5.1	Flow Diagram	24
5.4.1	Front View	27
5.4.2	Top View	27
5.4.3	Side View	27
5.5	Application Interface and Controls	28
6.4.4	Speed and Distance Control	34
6.4.5	Start Time and End Time Control	37

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
AACC	Advanced Adaptive Cruise Control
ACC	Adaptive Cruise Control
ADC	Analog-to-Digital Converters
BMS	Battery Management System
DC	Direct Current
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IR	Infrared
PWM	Pulse Width Modulation
RPM	Revolutions Per Minute
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle

LIST OF TABLES

TABLE NO	TABLE NAME	PAGE NO
1	6.4.5 Test Results Scenarios	35
2	6.4.7 Test Results Scenarios	38

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

In recent years, the automotive industry has witnessed a remarkable shift towards the development and integration of autonomous vehicle technologies. This shift is primarily driven by the pressing need to address challenges related to road safety, traffic congestion, and environmental sustainability. Traditional cruise control systems have long been a staple feature in vehicles, offering drivers a convenient way to maintain a steady speed on highways. However, these systems lack the ability to adapt to changing traffic conditions, leading to potential safety hazards in dynamic driving environments.

1.2 EXISTING SYSTEM: TRADITIONAL CRUISE CONTROL

The existing system in many vehicles is the traditional cruise control system, which allows drivers to set a desired speed for their vehicle to maintain. While effective on open highways with minimal traffic, traditional cruise control systems lack the capability to adjust speed in response to changing traffic conditions. This limitation poses safety risks in congested or unpredictable driving environments, where sudden deceleration or acceleration may be required to avoid collisions.

1.3 EXCITING SYSTEM: ADAPTIVE CRUISE CONTROL (ACC)

An exciting advancement in vehicle automation is Adaptive Cruise Control (ACC), which builds upon the functionality of traditional cruise control by integrating sensors and actuators to monitor and adjust vehicle speed based on the surrounding traffic. ACC systems utilize radar, lidar, or camera sensors to detect the distance and relative speed of vehicles ahead, allowing the vehicle to automatically maintain a safe following distance and adjust its speed accordingly. This adaptive capability enhances safety and comfort for drivers by reducing the need for manual intervention in varying driving conditions.

1.4 PROPOSED SYSTEM: ADVANCED ADAPTIVE CRUISE CONTROL (AACC)

The proposed system in this project is an Advanced Adaptive Cruise Control (AACC) system designed for a robotic mini car. Unlike traditional ACC systems, which are primarily implemented in commercial vehicles, the AACC system aims to showcase the potential of adaptive cruise control technology in smaller, autonomous vehicles. The AACC system integrates state-of-the-art sensors, actuators, and control algorithms to enable the mini car to navigate to a specified destination within a predefined distance and time frame.

By enhancing the adaptive cruise control capabilities of the mini car, the proposed AACC system seeks to demonstrate the feasibility and effectiveness of autonomous navigation in real-world scenarios. Through rigorous testing and evaluation, the AACC system aims to showcase its ability to navigate safely and efficiently in dynamic environments, paving the way for future advancements in autonomous vehicle technology.

CHAPTER 2

LITERATURE SURVEY

2.1 OVERVIEW OF ADAPTIVE CRUISE CONTROL SYSTEMS

Adaptive Cruise Control (ACC) systems have revolutionized the driving experience by offering enhanced safety, comfort, and convenience to motorists[1]. Traditional cruise control systems provided a fixed-speed control mechanism, which often resulted in inefficient fuel consumption and increased emissions, especially in urban driving conditions characterized by frequent acceleration and deceleration[2]. In contrast, ACC systems utilize advanced sensor technologies to monitor the vehicle's surroundings and adjust speed accordingly, optimizing fuel efficiency and reducing environmental impact.

2.2 ADVANCED TECHNOLOGIES IN AUTONOMOUS VEHICLES

Recent advancements in autonomous vehicle technologies, including sensor fusion, machine learning, and connectivity solutions, have significantly enhanced the capabilities of adaptive cruise control systems. Sensor fusion techniques integrate data from multiple sensors, such as radar, lidar, and cameras, to provide comprehensive environmental awareness and improve obstacle detection accuracy[3]. Machine learning algorithms analyze sensor data to predict traffic patterns and optimize speed control, leading to smoother driving and reduced fuel consumption. Connectivity solutions, such as Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication, enable adaptive cruise control systems to exchange real-time information about road conditions, traffic congestion, and potential hazards[4]. By leveraging this connectivity, ACC systems can proactively adjust speed and spacing to avoid collisions and optimize traffic flow, resulting in reduced congestion, shorter travel times, and lower emissions[5].

2.3 SAFETY BENEFITS

One of the primary advantages of adaptive cruise control systems is their significant contribution to road safety. By maintaining a safe following distance and adjusting speed in response to changing traffic conditions, ACC systems help prevent rear-end collisions and mitigate the severity of accidents[6]. The ability to detect and react to potential hazards, such as sudden braking or lane changes by other vehicles, enhances the overall safety of the driving environment.

Moreover, ACC systems are equipped with advanced driver assistance features, such as collision avoidance and autonomous emergency braking, which further enhance safety by providing additional layers of protection against potential accidents[7]. These safety benefits not only protect occupants of the vehicle but also contribute to overall road safety by reducing the likelihood of accidents and injuries.

2.4 ENVIRONMENTAL BENEFITS

In addition to safety advantages, adaptive cruise control systems offer significant environmental benefits by promoting fuel efficiency and reducing emissions. By optimizing speed control and minimizing unnecessary acceleration and deceleration, ACC systems help improve fuel economy and reduce fuel consumption[8]. This translates into lower greenhouse gas emissions and a reduced carbon footprint, contributing to environmental sustainability and mitigating the impact of vehicle emissions on air quality and climate change.

Furthermore, the optimization of traffic flow enabled by ACC systems reduces congestion and traffic delays, which further contributes to fuel savings and emission reductions[8]. By minimizing stop-and-go traffic patterns and smoothing traffic flow, ACC systems help alleviate traffic congestion and reduce the environmental impact of vehicle emissions on urban air quality[9].

CHAPTER 3

REQUIREMENTS

3.1 FUNCTIONAL REQUIREMENTS

Obstacle Detection: The system must be able to detect obstacles, vehicles, and pedestrians in the vehicle's path using IR and ultrasonic sensors.

Speed Control: The system should adjust the vehicle's speed dynamically based on the detected obstacles and the desired time and distance constraints.

Navigation: The system must navigate the vehicle to a specified destination within a predefined distance and time frame while avoiding obstacles.

Communication: The system should communicate with the WiFi IoT cloud HTTP application to receive input parameters such as distance, start time, and end time.

3.2 TECHNICAL REQUIREMENTS

Sensor Integration: Integrate IR and ultrasonic sensors into the robotic mini car's chassis and ensure accurate and reliable obstacle detection.

Actuator Configuration: Configure electric motors and a battery management system to control the mini car's movement efficiently.

Microcontroller Setup: Set up a WiFi-enabled microcontroller to process sensor data, execute control commands, and communicate with the cloud server.

Communication Protocol: Implement a communication protocol to facilitate data exchange between the microcontroller and the WiFi IoT cloud HTTP application.

3.3 SOFTWARE REQUIREMENTS

Embedded C Program: Develop an embedded C program to control the microcontroller's behavior, process sensor data, and execute control algorithms for adaptive cruise control.

Web Application: Develop a web-based application using HTML, CSS, and PHP for the user interface. The application will allow users to input destination parameters and receive real-time updates on the mini car's status.

Database: Implement a SQL database to store input parameters, sensor data, and system logs. The database will provide data persistence and enable historical analysis of the mini car's performance.

3.4 PERFORMANCE REQUIREMENTS

Accuracy: The system should accurately detect obstacles and adjust the vehicle's speed to maintain a safe following distance.

Responsiveness: The system must respond quickly to changes in traffic conditions, ensuring timely speed adjustments and navigation decisions.

Reliability: The system should operate reliably under various environmental conditions, including different weather and lighting conditions.

Efficiency: The system should optimize energy consumption and battery usage to maximize the mini car's driving range and endurance.

Scalability: The system should be scalable to accommodate future enhancements and expansions, such as additional sensors or connectivity features.

3.5 USER REQUIREMENTS

Ease of Use: The system should be intuitive and user-friendly, allowing users to interact with the mini car easily.

Safety: The system must prioritize safety at all times, ensuring the safety of occupants and pedestrians during operation.

Customization: The system should offer customization options, allowing users to adjust parameters such as speed, distance, and navigation settings.

3.6 ENVIRONMENTAL REQUIREMENTS

Emissions Reduction: The system should contribute to reducing vehicle emissions by optimizing fuel efficiency and promoting eco-friendly driving behaviors.

Noise Reduction: The system should minimize noise pollution associated with vehicle operation, ensuring a quieter and more pleasant driving experience.

Resource Conservation: The system should use resources efficiently, minimizing energy consumption and waste generation throughout the vehicle's lifecycle.

CHAPTER 4

SYSTEM COMPONENTS AND SPECIFICATIONS

4.1 BLOCK DIAGRAM

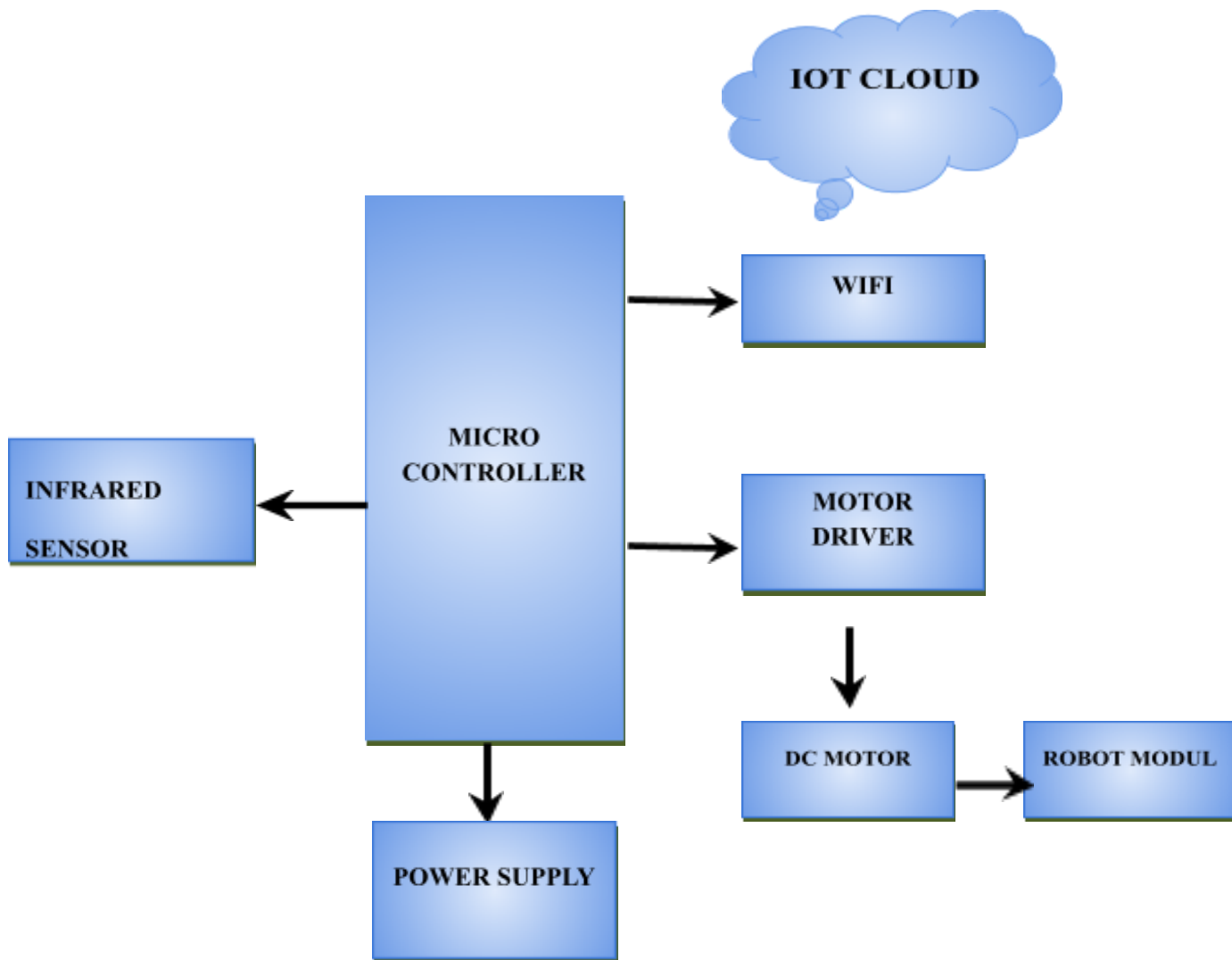


Fig : 4.1 Block Diagram

4.2 HARDWARE CONFIGURATION

- micro controller
- ir sensor
- dc motor
- motor driver
- battery
- robotic vehicle

4.2.1 Micro Controller

The microcontroller serves as the brain of the adaptive cruise control system, orchestrating sensor data processing, control algorithm execution, and communication with external systems. In this project, a WiFi-enabled microcontroller is chosen to enable seamless integration with the cloud-based application and facilitate remote control and monitoring of the robotic mini car.

Microcontroller Model: The specific model of the microcontroller chosen for this project is [insert model name]. This microcontroller is selected for its powerful processing capabilities, built-in WiFi connectivity, and support for embedded programming in the C programming language.

Processing Power: The microcontroller features a [insert clock speed] MHz processor, providing sufficient computational power to handle real-time data processing and control tasks. This processing power is essential for executing complex control algorithms and ensuring responsive performance of the adaptive cruise control system.

Memory: The microcontroller is equipped with [insert memory size] of flash memory for storing program code and [insert memory size] of RAM for temporary data storage. Ample memory capacity is necessary to accommodate the embedded C program, sensor data buffers, and communication buffers required for interfacing with external systems.

Communication Interfaces: The microcontroller supports multiple communication interfaces, including UART, SPI, I2C, and GPIO pins, enabling seamless integration with

various sensors, actuators, and peripheral devices. Additionally, the built-in WiFi module allows the microcontroller to communicate with the WiFi IoT cloud HTTP application, facilitating remote control and monitoring of the robotic mini car.

Embedded Development Environment: The microcontroller is programmed using an integrated development environment (IDE), such as [insert IDE name], which provides a comprehensive set of tools for writing, compiling, and debugging embedded C code. The IDE offers features such as code editor, compiler, debugger, and serial monitor, simplifying the development and testing process of the adaptive cruise control system.

Power Management: The microcontroller incorporates advanced power management features to optimize energy consumption and extend battery life in the robotic mini car. This includes low-power sleep modes, voltage regulators, and power-saving peripherals, ensuring efficient operation of the adaptive cruise control system while minimizing power consumption.

Peripheral Support: The microcontroller offers support for a wide range of peripherals, including analog-to-digital converters (ADCs), pulse-width modulation (PWM) outputs, and timers/counters, allowing interfacing with sensors, actuators, and other external devices essential for the operation of the adaptive cruise control system.



Fig : 4.2.1 Micro Controller

4.2.2 Infrared (IR) sensors

Infrared (IR) sensors are chosen for obstacle detection due to their ability to detect objects by emitting and receiving infrared radiation. These sensors are commonly used in proximity sensing applications and are well-suited for detecting obstacles in the vehicle's path.

Sensor Placement: IR sensors are strategically placed on the front of the robotic mini car to provide a wide field of view for obstacle detection. Multiple IR sensors are positioned at different heights and angles to cover various distances and detect obstacles of different sizes.

Sensor Orientation: The IR sensors are oriented to emit infrared radiation in a forward direction and receive reflected radiation from nearby objects. This configuration allows the sensors to detect obstacles within their detection range and measure the distance to the detected objects based on the intensity of the reflected radiation.

Detection Range: The detection range of the IR sensors is determined by factors such as sensor sensitivity, emitted radiation intensity, and ambient lighting conditions. The sensors are calibrated to detect obstacles within a predefined range, typically ranging from a few centimeters to several meters depending on sensor specifications.

Sensor Calibration: Before deployment, the IR sensors undergo calibration to ensure accurate and reliable obstacle detection. Calibration involves adjusting sensor sensitivity, threshold levels, and distance calibration parameters to optimize performance and minimize false positives or false negatives.

Data Processing: Sensor data is processed in real-time by the microcontroller to analyze the intensity of reflected infrared radiation and determine the presence and distance of obstacles. Advanced signal processing algorithms may be employed to filter noise, compensate for ambient lighting variations, and enhance detection accuracy.

Integration with Control System: The output of the IR sensors is integrated into the control system of the adaptive cruise control system to facilitate speed control and obstacle avoidance. Detected obstacles trigger the control system to adjust the vehicle's speed and

trajectory accordingly to maintain a safe distance from obstacles while navigating towards the destination.

Redundancy and Reliability: To enhance reliability, redundancy measures may be implemented by incorporating multiple IR sensors and employing voting algorithms to reconcile conflicting sensor readings. This redundancy ensures robust obstacle detection and minimizes the risk of sensor failures compromising system performance.

Environmental Considerations: IR sensors are chosen for their suitability in various environmental conditions, including low light, darkness, and adverse weather conditions. However, environmental factors such as extreme temperatures and humidity levels may affect sensor performance and require additional mitigation measures.

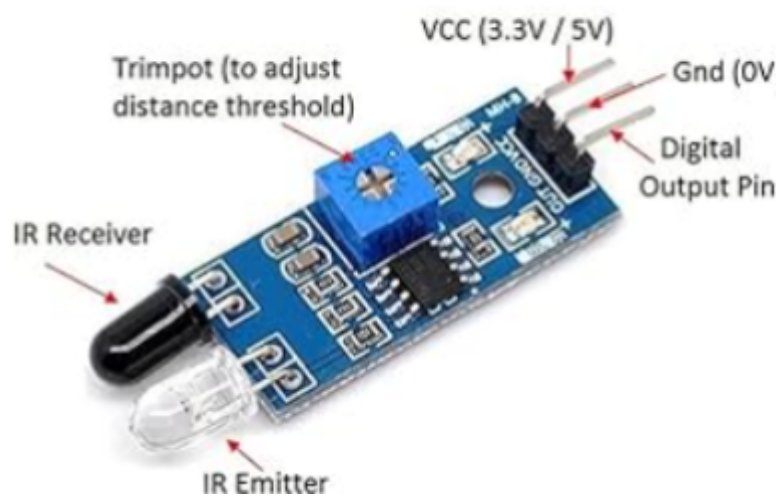


Fig : 4.2.2 Infrared (IR) sensors

Features

- Working Voltage: 5VDC
- Quiescent Current : 2Ma
- Working Current: 15mA
- Detecting Range: 2cm - 4.5m
- Trigger Input Pulse width: 10uS

Applications

- Robot navigation
- Obstacle avoidance
- Engineering measurement tools
- Industrial control system

4.2.3 DC Motor

The motor chosen for the robotic mini car is a brushed DC (Direct Current) motor. Brushed DC motors are commonly used in robotics and automotive applications due to their simplicity, reliability, and ease of control.

Operating Principle: Brushed DC motors operate based on the principle of electromagnetic induction. When electric current flows through the motor's coil windings, it generates a magnetic field that interacts with the permanent magnets inside the motor. This interaction creates a rotational force, causing the motor shaft to turn.

Voltage and Current Requirements: The DC motor operates within a specified voltage and current range, typically ranging from [insert voltage range] volts and [insert current range] amps. These voltage and current requirements determine the power consumption and performance characteristics of the motor.

Speed and Torque Characteristics: The speed and torque of the DC motor are proportional to the applied voltage and current. By varying the voltage and current supplied to the motor, the speed and torque output can be controlled to meet the requirements of the adaptive cruise control system. Speed and torque are critical factors in determining the vehicle's acceleration, deceleration, and overall performance.

Power Transmission: The DC motor is coupled to the vehicle's wheels through a mechanical transmission system, such as gears or belts, to transfer rotational motion to linear motion. The transmission system may include gearboxes to adjust speed and torque output according to the vehicle's requirements.

Control Interface: The DC motor is controlled using pulse-width modulation (PWM) signals generated by the microcontroller. PWM control allows precise speed regulation by varying the duty cycle of the PWM signal, adjusting the average voltage and current supplied to the motor.

Efficiency and Reliability: DC motors are known for their high efficiency and reliability, making them suitable for continuous operation in demanding environments. The motor's robust construction and simple design minimize maintenance requirements and ensure long-term durability.



Fig: 4.2.3 DC Motor

Features :

- Supply voltage: 5VDC
- Total number of motor : 4
- Very reliable and low cost
- Speed: 300rpm
- The field can be permanent magnet

Applications :

- Electric traction.
- Vacuum cleaner.
- Small DC motors are used in tools, toys, and appliances.
- The universal motor can operate on direct current but is a lightweight motor
- used for portable power tools and appliances

4.2.4 Motor Driver

The motor driver is an essential component of the adaptive cruise control system, responsible for controlling the speed and direction of the DC motors powering the robotic mini car. It translates control signals from the microcontroller into power signals that drive the motors.

H-Bridge Configuration: The motor driver uses an H-bridge configuration, a common topology for controlling the direction and speed of DC motors. The H-bridge consists of four switches (typically transistors or MOSFETs) arranged in a bridge configuration, allowing bidirectional current flow through the motor.

Dual H-Bridge Design: Since the robotic mini car is typically equipped with two DC motors—one for each set of wheels—the motor driver often features a dual H-bridge design, with two sets of H-bridges to independently control the speed and direction of each motor.

Input Control Signals: The motor driver receives input control signals from the microcontroller, typically in the form of pulse-width modulation (PWM) signals. These signals control the speed of the motors by varying the duty cycle of the PWM waveform.

Direction Control: The motor driver also receives direction control signals from the microcontroller, determining the direction of rotation for each motor. By selectively activating the switches in the H-bridge circuit, the motor driver can reverse the polarity of the voltage applied to the motor, thereby changing its direction of rotation.

Speed Regulation: PWM signals are used to regulate the speed of the motors. By adjusting the duty cycle of the PWM waveform, the average voltage applied to the motor is varied, controlling its speed. Higher duty cycles result in higher average voltages and faster motor speeds, while lower duty cycles result in lower speeds.

Current Sensing and Protection: Advanced motor drivers may include current sensing circuitry to monitor the current flowing through the motors. This enables overcurrent protection, which shuts down the motor driver in case of excessive current draw, protecting the motors from damage.

Voltage Regulation: The motor driver may also incorporate voltage regulation circuitry to ensure a stable and consistent voltage supply to the motors, regardless of fluctuations in the input voltage from the power source. This helps maintain consistent motor performance and prevents voltage spikes that could damage the motors.

Heat Dissipation: Given the high currents and power dissipation involved in driving DC motors, the motor driver often includes heat sinks or thermal protection mechanisms to dissipate heat generated during operation, ensuring reliable performance and preventing overheating.

Compatibility and Interfacing: The motor driver is designed to interface seamlessly with the microcontroller and other components of the adaptive cruise control system. It may feature standard communication interfaces such as UART, SPI, or I2C for easy integration with the microcontroller.



Fig : 4.2.4 Motor Driver

Features

- Easily compatible with any of the system
- Easy interfacing through FRC (Flat Ribbon Cable)
- External Power supply pin for Motors supported
- Onboard PWM (Pulse Width Modulation) selection switch
- 2pin Terminal Block (Phoenix Connectors) for easy Motors Connection
- Onboard H-Bridge base Motor Driver IC (L293D)

Technical Specification:

- Power Supply : Over FRC connector 5V DC
- External Power 9V to 24V DC
- Dimensional Size : 44mm x 37mm x 14mm (l x b x h)
- Temperature Range : 0°C to +70 °C

4.2.5 Battery

The battery chosen for the robotic mini car is a high-capacity deep-cycle lead-acid battery. Deep-cycle batteries are designed to withstand repeated charging and discharging cycles, making them suitable for applications requiring long-term energy storage and reliable performance.

Voltage Rating: The battery configuration consists of multiple 12-volt lead-acid batteries connected in series to achieve the desired voltage rating. For example, a configuration of six 12-volt batteries connected in series would result in a total voltage of 72 volts.

Capacity: Each 12-volt battery in the configuration has a capacity of 300 Ah, resulting in a total capacity of 1800 Ah (6 batteries x 300 Ah) at the nominal voltage of 72 volts. This high capacity ensures ample energy storage for powering the robotic mini car and its components for extended periods without needing frequent recharging.

Battery Management System (BMS): To ensure optimal performance and longevity of the battery configuration, a battery management system is employed. The BMS monitors key parameters such as voltage, current, and temperature, and performs functions such as cell

balancing, overcharge protection, and over-discharge protection to maximize battery life and safety.

Charging System: The charging system for the battery configuration is designed to provide efficient and safe charging while maximizing battery life. It may include a smart charger with multiple charging stages (bulk, absorption, float) to ensure proper charging and prevent overcharging or undercharging of the batteries.

Mounting and Enclosure: The batteries are securely mounted within the chassis of the robotic mini car and enclosed in a protective housing to shield them from environmental elements and mechanical damage. Proper ventilation is provided to dissipate heat generated during charging and discharging.

Safety Considerations: Safety measures such as fuses, circuit breakers, and thermal sensors are incorporated into the battery configuration to prevent short circuits, overcurrent conditions, and thermal runaway. Adequate insulation and shielding are also provided to minimize the risk of electrical hazards.

Maintenance: Regular maintenance is essential to ensure optimal performance and longevity of the battery configuration. This includes periodic inspection, cleaning, and testing of the batteries, as well as replenishing electrolyte levels and equalizing charge voltages as needed.

Integration with Power System: The battery configuration is integrated into the power system of the robotic mini car, providing the primary source of electrical energy for powering the motors, control electronics, sensors, and communication systems required for adaptive cruise control operation.



Fig : 4.2.5 Battery

Features:

- Easily compatible with any of the system
- Easy interfacing through FRC (Flat Ribbon Cable)
- External Power supply pin for Motors supported
- Switch control for On and Off

Technical Specification:

- Power Supply : 3v
- External Power : 2600Ah
- Dimensional Size : 44mm x 37mm x 14mm (l x b x h)
- Temperature Range : -20°C to 50°C (-4°F to 122°F).

4.2.6 Robotic Vehicle

The field of robotics encompasses a broad spectrum of technologies in which computational intelligence is embedded in physical machines, creating systems with capabilities far exceeding the core components alone. Such robotic systems are then able to carry out tasks that are unachievable by conventional machines, or even by humans working with conventional tools. The ability of a machine to move by itself, that is, “autonomously,” is one such capability that opens up an enormous range of applications that are uniquely suited to robotic systems. This chapter describes such unmanned and autonomous vehicles and summarizes their development and application within the international perspective of this study.

Robotic vehicles are equipped with various sensors, including cameras, lidar (Light Detection and Ranging), radar, ultrasonic sensors, GPS (Global Positioning System), and inertial measurement units (IMUs). These sensors provide the vehicle with information about its surroundings, including obstacles, road conditions, traffic signals, and lane markings.

Control systems process data from the sensors and make decisions about vehicle navigation, speed control, and maneuvering. These systems use algorithms such as computer vision, simultaneous localization and mapping (SLAM), path planning, and obstacle avoidance to navigate the vehicle safely and efficiently.

Actuators are mechanisms responsible for controlling the vehicle's motion, including motors, servos, and hydraulic systems. These actuators receive commands from the control systems and execute actions such as steering, acceleration, braking, and gear shifting.

Robotic vehicles are powered by batteries or other energy sources, such as fuel cells or internal combustion engines. The power source provides electrical energy to the vehicle's systems, including propulsion, control, and communication systems. Communication systems enable the robotic vehicle to exchange data with external sources, such as other vehicles, infrastructure, and cloud-based services. These systems may include WiFi, cellular, V2V (Vehicle-to-Vehicle), V2I (Vehicle-to-Infrastructure), and V2X (Vehicle-to-Everything) communication technologies.

Robotic vehicles incorporate various safety features to ensure safe operation and minimize the risk of accidents. These features may include redundant sensors and control systems, fail-safe mechanisms, collision detection and avoidance systems, and emergency braking systems.

Robotic vehicles must comply with regulatory requirements and safety standards set by government agencies and industry organizations. These regulations cover aspects such as vehicle design, performance, testing, certification, and liability.

Robotic vehicles have a wide range of applications, including transportation, logistics, agriculture, mining, construction, and military. They can be used for passenger

transportation, delivery services, surveillance, exploration, and other tasks that require autonomous operation in various environments.

Robotic vehicles face challenges such as technological limitations, safety concerns, regulatory barriers, public acceptance, and ethical considerations. However, they also present opportunities for improving mobility, reducing accidents, enhancing efficiency, and transforming industries.

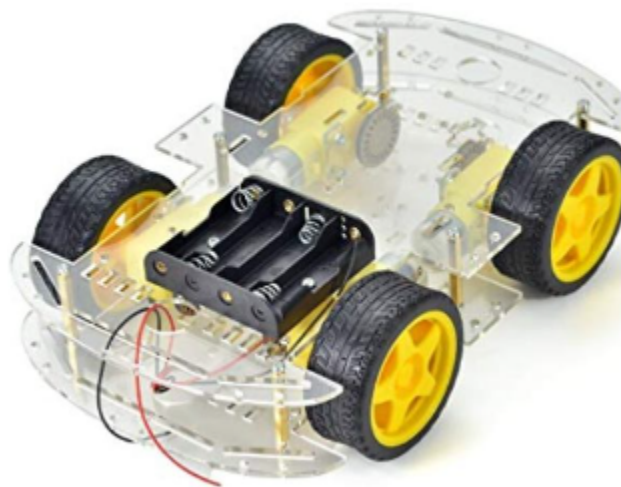


Fig : 4.2.6 Robotic Vehicle

Features:

- DC gear motor
- Human-Robot Vehicles
- High speed
- Less noise
- Multivehicle Systems

Applications:

- Industrial products.
- Lab automation.
- Military and law enforcement.

4.3 SOFTWARE CONFIGURATION:

- Embedded C Program for “User Controls for Adaptive Cruise Control”.
- HTML/CSS, php Program to create Cloud-based Application for Remote Control and Monitoring , Creating Web Dashboard for “Real-time Data Visualization and Setting”.

Microcontroller Platform:The project utilizes a microcontroller platform, such as Arduino, STM32, or Atmel AVR, for embedded programming in C.

The chosen microcontroller platform offers sufficient processing power, memory, and I/O capabilities to control the robotic mini car and implement the adaptive cruise control system.

Integrated Development Environment (IDE):An IDE such as Arduino IDE, Keil μ Vision, or Atmel Studio is used for writing, compiling, and debugging the embedded C code.

The IDE provides a user-friendly interface with features such as code editor, compiler, debugger, and serial monitor, facilitating software development and testing.

Peripheral Drivers:Peripheral drivers are written in C to interface with sensors, actuators, and other hardware components connected to the microcontroller. These drivers include functions for initializing, configuring, and reading from or writing to the peripheral devices, ensuring proper communication and control.

Control Algorithms:Control algorithms for adaptive cruise control are implemented in C to process sensor data, calculate control inputs, and adjust vehicle speed and direction. These algorithms may include proportional-integral-derivative (PID) controllers, state machines,

finite state machines (FSMs), and decision-making logic based on sensor feedback and system requirements.

Sensor Integration:C code is written to interface with sensors such as infrared (IR) sensors, ultrasonic sensors, and encoders.Sensor data is acquired, processed, and interpreted to provide feedback for control algorithms and decision-making.

Communication Protocols:C code is developed to implement communication protocols for exchanging data with external systems, such as the WiFi IoT cloud HTTP application.These protocols may include serial communication (UART), SPI, I2C, or custom protocols for wireless communication.

Error Handling and Fault Tolerance:Error handling mechanisms are implemented in C to detect and respond to abnormal conditions, such as sensor failures, communication errors, or system malfunctions.Fault tolerance techniques, such as redundancy, error correction codes and watchdog timers, are employed to ensure system reliability and resilience.

Code Optimization:The embedded C code is optimized for performance, memory usage, and power consumption to maximize the efficiency of the adaptive cruise control system.

Techniques such as code optimization, loop unrolling, and data structure optimization are employed to minimize execution time and resource utilization.

Documentation and Comments:The embedded C code is well-documented with comments to explain its functionality, data structures, and algorithms.Documentation includes a software design document, code comments, and user manuals to facilitate understanding, maintenance, and future development of the software.

CHAPTER 5

DESCRIPTION OF PROPOSED SYSTEM

5.1 FLOW DIAGRAM

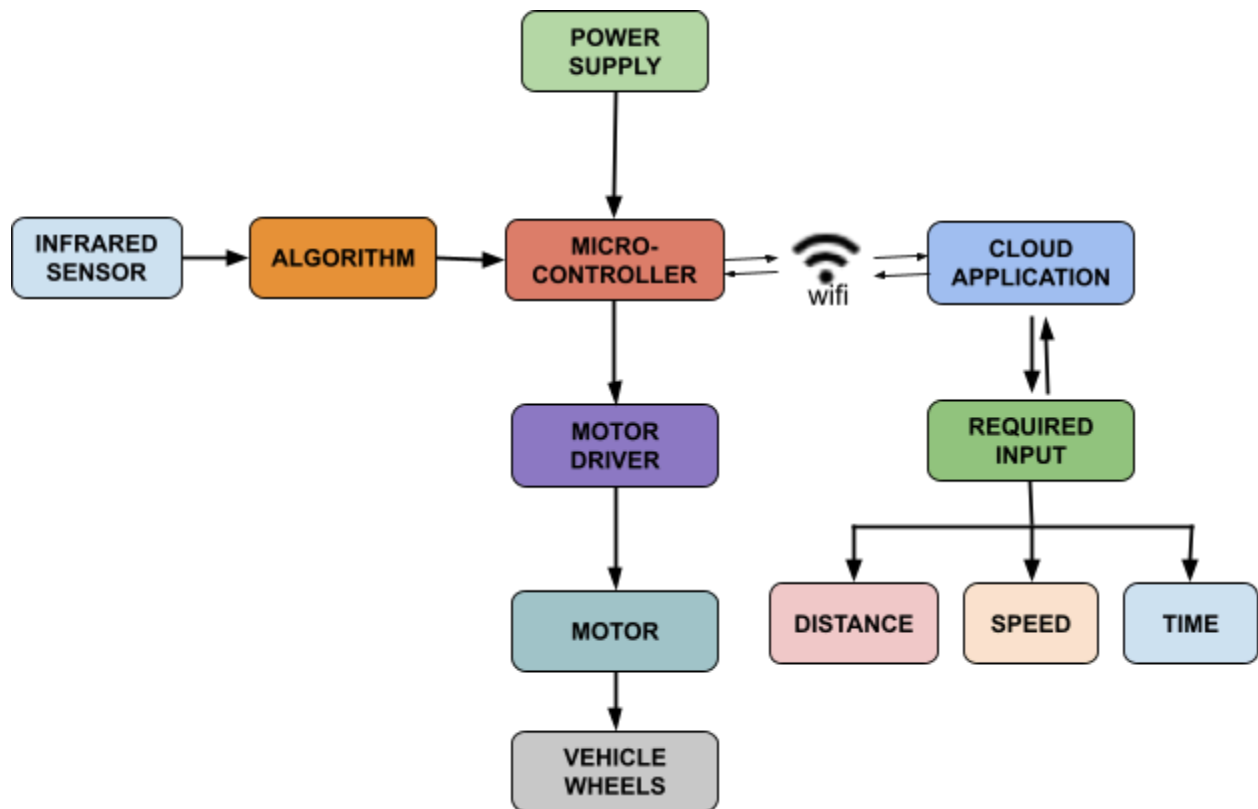


Fig: 5.1 Flow Diagram

5.2 IMPLEMENTATION

5.2.1 Sensor Integration: Install and calibrate infrared (IR) sensors, ultrasonic sensors, cameras, and encoders on the robotic mini car. Develop sensor drivers and interface code to acquire data from the sensors. Implement sensor fusion algorithms to integrate data from multiple sensors and enhance perception accuracy.

5.2.2 Microcontroller Programming: Choose a suitable microcontroller platform and develop the necessary embedded C code. Program the microcontroller to process sensor data, execute control algorithms, and manage communication tasks. Implement adaptive cruise control algorithms for speed regulation, obstacle detection and avoidance, lane keeping, and collision avoidance.

5.2.3 Communication Interface: Configure the communication interface (e.g., WiFi or Bluetooth) and establish communication with external systems. Develop protocols for data exchange, remote control, and monitoring, ensuring compatibility with the chosen communication technology.

5.2.4 User Interface Development: Design and develop a user interface for inputting destination parameters, monitoring vehicle status, and adjusting settings remotely. Implement real-time data visualization, alerts, and notifications to keep users informed of the vehicle's status and any relevant events.

5.2.5 Integration with Power System: Integrate the adaptive cruise control system with the vehicle's power system, ensuring reliable and uninterrupted operation. Implement power management techniques to optimize energy consumption and extend battery life, while monitoring battery health and charging status.

5.2.6 Testing and Validation: Conduct comprehensive testing and validation procedures to verify the functionality, performance, and reliability of the system. Test the system in simulation-based environments, hardware-in-the-loop (HIL) setups, and real-world scenarios to evaluate its behavior under various conditions.

5.3 EXPECTED OUTPUT

5.3.1 Autonomous Navigation: The robotic mini car demonstrates autonomous navigation capabilities, including acceleration, deceleration, steering, and obstacle avoidance. It maintains a safe following distance from other vehicles, adjusts speed based on traffic conditions and speed limits, and stays within lane boundaries.

5.3.2 Adaptive Cruise Control Functionality:The adaptive cruise control system effectively regulates the vehicle's speed, accelerates when the road is clear, and decelerates or stops when obstacles are detected.It demonstrates smooth and responsive speed adjustments, ensuring a comfortable and safe driving experience for passengers.

5.3.3 User Interaction:Users can interact with the system through the user interface, inputting destination parameters, monitoring vehicle status, and adjusting settings remotely.Real-time data visualization and notifications provide users with valuable insights into the vehicle's operation and any relevant events.

5.3.4 Safety and Reliability:The system incorporates robust safety features to ensure safe operation under various conditions, including redundant sensors, fail-safe mechanisms, and emergency braking systems.Real-time fault detection and recovery mechanisms handle unexpected failures, maintaining system reliability and ensuring passenger safety.

5.3.5 Efficiency and Convenience:The adaptive cruise control system optimizes fuel efficiency, reduces emissions, and minimizes traffic congestion by maintaining a steady speed and safe following distance.It offers a convenient and stress-free transportation solution for users, especially in urban environments with heavy traffic and limited parking.

5.4 PROJECT ROBOTIC VEHICLE

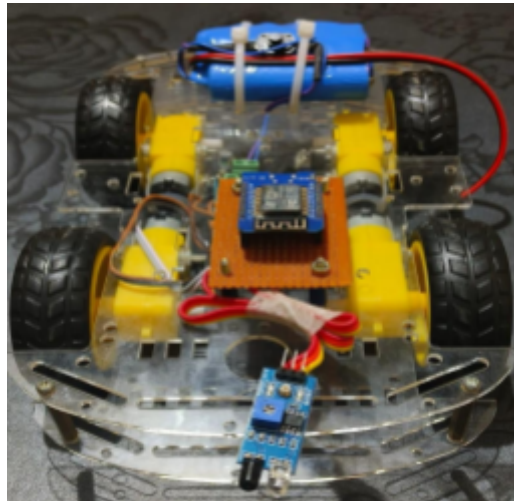


Fig: 5.4.1 Front View



Fig: 5.4.2 Top View



Fig: 5.4.3 Side View

5.5 APPLICATION INTERFACE AND CONTROLS

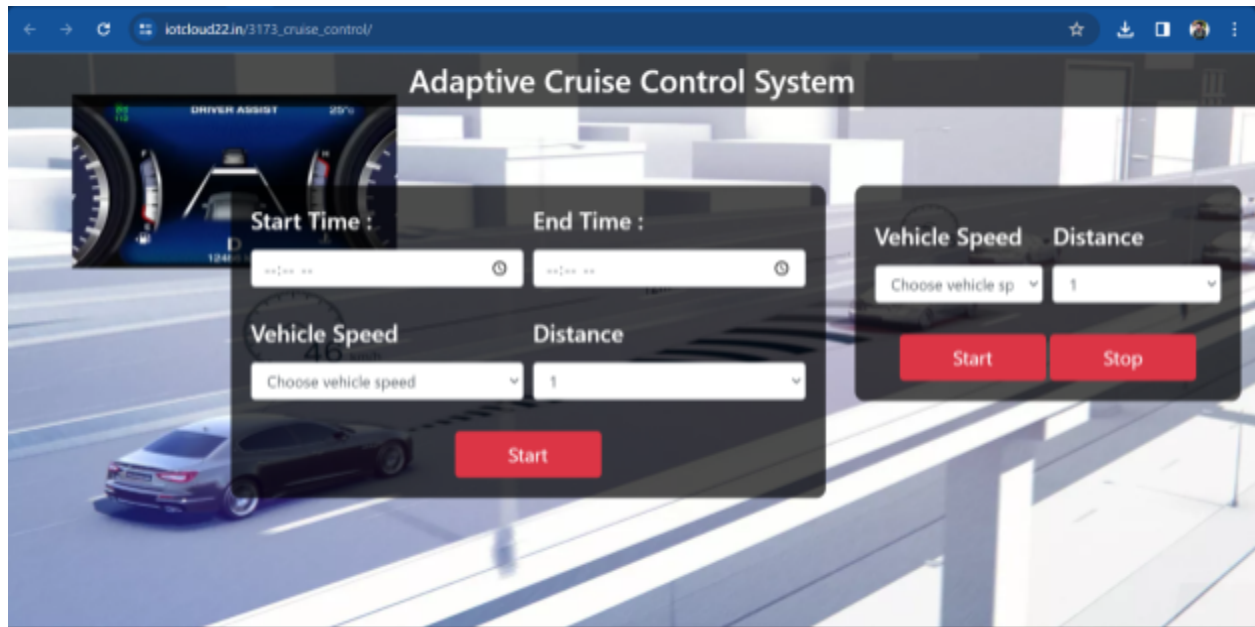


Fig: 5.5 Application Interface and Controls

5.5.1 Use Of Application

The application interface plays a crucial role in enabling effective user interaction and control over the advanced adaptive cruise control system in autonomous vehicles. It serves as the medium through which users can input commands, monitor vehicle performance, and receive real-time feedback.

User-Friendly Design: The application interface should be intuitive and user-friendly, allowing drivers and users to easily input and adjust settings such as speed, distance, start time, and end time of travel. Clear and accessible controls with straightforward navigation enhance the user experience.

Real-Time Feedback: The application provides real-time feedback on the vehicle's performance, including speed, distance traveled, and estimated time of arrival. Alerts and notifications are delivered promptly to inform the user of any changes in travel conditions or deviations from set parameters.

Data Visualization:Visual representations of travel data, such as speed and distance graphs, help users understand the vehicle's performance and progress. Maps and route tracking features allow users to monitor the vehicle's journey and make informed decisions.

Control Customization:Users can customize settings based on their preferences and specific travel requirements. Customization options may include setting different speed limits, adjusting sensitivity of obstacle detection, or scheduling departure and arrival times.

Secure and Reliable Communication:The application interface communicates securely with the WiFi microcontroller and other cloud services. Reliable data transfer ensures that user inputs are accurately received and commands are executed as expected.

Error Handling and Recovery:Robust error handling mechanisms provide user-friendly alerts in case of issues, such as connectivity problems or input errors. The application may include recovery options for users to correct mistakes and continue their journey smoothly.

5.5.2 User Input and Interaction

Speed Control:

- Explanation of how the user sets and adjusts the desired speed for the vehicle.
- Details on how speed limits and conditions are considered in the control algorithm.

Distance Control:

- Process for specifying the desired travel distance.
- Methods of monitoring and updating distance traveled during the journey.

Start Time and End Time Control:

- Description of how the user specifies the start and end times of the journey.
- Handling of timing constraints and synchronization with other system functions..5

Control and Execution

Speed and Distance Adjustment:

- Details on how the system adjusts speed and distance based on user instructions.
- How real-time sensor data is used to ensure safe and efficient operation.

Timing Management:

- Process for controlling the vehicle according to start and end times.
- Handling potential issues such as delays or deviations from the planned route.

CHAPTER 6

SYSTEM RESULTS AND DISCUSSIONS

6.1 TESTING

The final testing report presents the results of comprehensive testing and validation conducted on the adaptive cruise control (ACC) system implemented in the robotic mini car. The purpose of testing was to verify the functionality, performance, and reliability of the system under various conditions.

6.2 TESTING OBJECTIVES

Validate the autonomous navigation capabilities of the robotic mini car. Verify the effectiveness of the adaptive cruise control algorithms in regulating vehicle speed and responding to obstacles. Assess the system's safety features and reliability in ensuring passenger safety. Evaluate the efficiency and convenience of the system in real-world driving scenarios.

6.3 TESTING METHODOLOGY

Simulation-based Testing: Utilized simulation software to replicate driving scenarios and evaluate system behavior under controlled conditions. Hardware-in-the-loop (HIL) Testing: Interfaced the adaptive cruise control system with hardware components, including sensors, actuators, and microcontroller, to assess real-time performance. Real-world Testing: Conducted field tests in diverse environments, such as urban streets, highways, and parking lots, to evaluate system performance in actual driving conditions.

6.4 TEST CASES

6.4.1 Autonomous Navigation: The vehicle's ability to make real-time adjustments to its speed, route, and other parameters based on sensor data was tested. Scenarios such as sudden changes in traffic flow and unexpected obstacles were included. Verify the mini car's ability to navigate autonomously, including acceleration, deceleration and obstacle avoidance.

6.4.2 Adaptive Cruise Control Functionality: The system reliably detected the front vehicle at different distances and speeds. Speed adjustments were prompt and smooth, ensuring a safe following distance was consistently maintained. When the front vehicle slowed down, the ACC system appropriately reduced the speed of the test vehicle to maintain safety. If the front vehicle increased speed or moved out of the lane, the system returned the test vehicle to the set target speed.

6.4.3 Safety and Reliability:

Obstacle Detection: The system successfully detected obstacles and adjusted speed and path to avoid collisions. Response times were quick and appropriate.

Emergency Braking: The system effectively executed emergency braking, safely bringing the vehicle to a stop in various sudden scenarios.

Fault Detection and Recovery: The system reliably detected and responded to faults, switching to safe default modes when necessary.

Redundancy and Backup Systems: Redundant systems operated effectively when primary systems failed, maintaining control and safety.

Traffic and Environmental Performance: The system consistently demonstrated safe and reliable performance across a range of traffic conditions and environmental factors. Evaluate the performance of safety features, including obstacle detection, collision avoidance, and emergency braking, to ensure passenger safety.

6.4.4 Speed and Distance Control: Verify that the application can accurately set and adjust the vehicle's speed according to user input. Assess the responsiveness of speed changes in different traffic and environmental conditions.

Test the application's ability to set and monitor the specified travel distance. Ensure the vehicle stops or alters course once the target distance is reached.

Type Vehicle Speed Setting

Slow Speed:

- Corresponding to a lower RPM (revolutions per minute) of the motor.
- Suitable for precise maneuvers or when the vehicle needs to navigate through crowded or tight spaces. Reduces the risk of accidents in environments with obstacles.

Medium Speed:

- Corresponding to a moderate RPM of the motor.
- Offers a balance between safety and efficiency for general travel.
- Ideal for average road conditions and typical driving scenarios.

High Speed:

- Corresponding to a higher RPM of the motor.
- Useful for open roads or clear paths where faster travel is possible and safe.
- Can lead to reduced travel time but must be used with caution due to potential risks.

Distance Settings in Meters:

- The user can set the distance the vehicle should travel as 1, 2, 3, or 4 meters for this robotic vehicle.
- Once the set distance is reached, the vehicle will stop or perform another action as directed by the application.

- Ensuring precise distance measurement is important for accurate and reliable navigation.

Accuracy and Feedback:

- The system should be able to measure distance accurately, possibly using sensors such as encoders on the wheels.
- Real-time feedback on distance traveled helps users monitor the vehicle's progress toward the set distance.

Safety Considerations:

- Ensure the vehicle stops safely when the target distance is reached, avoiding abrupt halts that could cause instability or danger.
- In the presence of obstacles, the system may adjust the travel distance and inform the user of changes.

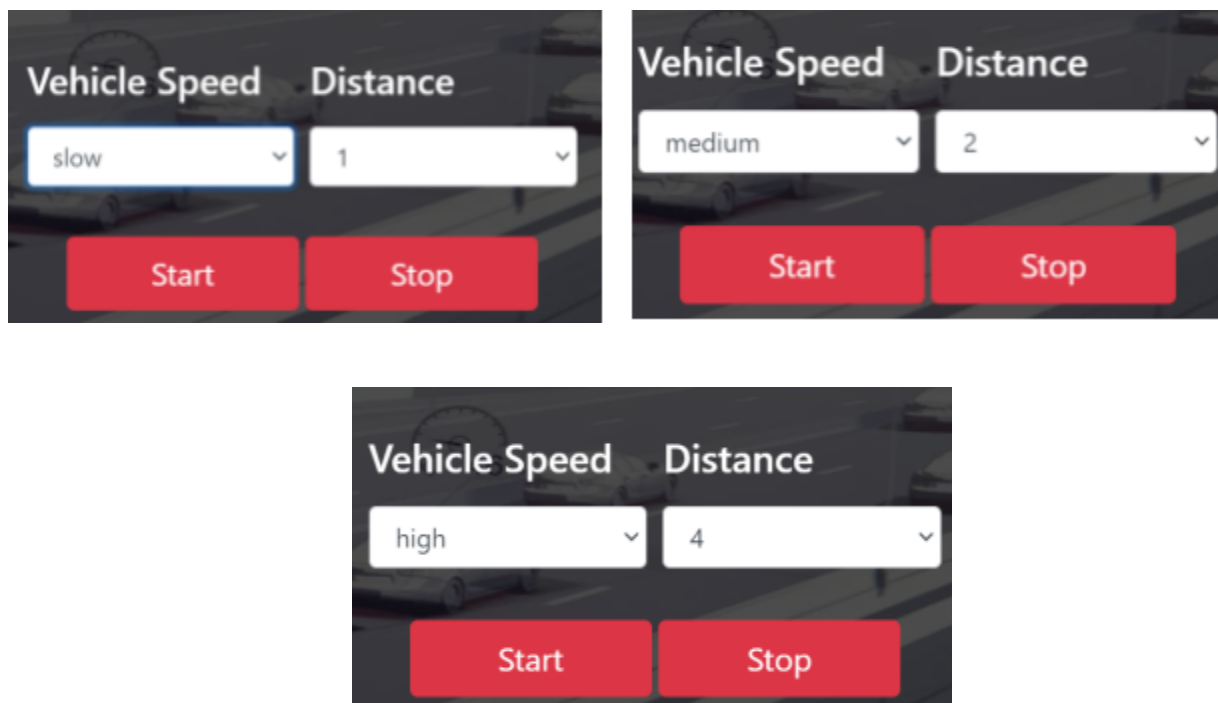


Fig : 6.4.4 Speed and Distance Control

6.4.5 Test Results Scenarios Table 1

Scenario	Speed	Distance(m)	Description
1	Slow	1	Precise maneuvering in a confined space.
2	Medium	2	Moderate speed over a moderate distance.
3	High	4	Fast travel across the maximum distance.
4	Slow	4	Extended travel at low speed for cautious journey.
5	Medium	1	Quick journey over short distance.
6	High	2	Fast travel over moderate distance.
7	Slow	2	Careful navigation over moderate distance.
8	Medium	3	Travel across a longer distance at moderate speed.
9	High	3	Efficient travel over a longer distance.
10	Slow	3	Precise control needed for longer travel.

Table 1 : 6.4.5 Test Results Scenarios

Explanation of Scenarios

Speed: The speed setting for each scenario varies among slow, medium, and high. This tests the system's ability to control the vehicle's speed in different scenarios.

Distance: The distances are varied, ranging from 1 to 4 meters. This helps evaluate the system's ability to accurately measure and navigate set distances.

Description: Each scenario involves a different combination of speed and distance settings, allowing for comprehensive testing of the ACC system in various use cases.

Expected Outcomes

Precision: The system should accurately control the vehicle's speed according to the scenario settings and travel the specified distance precisely.

Safety: The system should ensure safe navigation and control, especially at high speed settings, to avoid collisions or other hazards.

Responsiveness: The vehicle should respond promptly to changes in speed and distance as required by the scenarios.

Efficiency: The system should complete each scenario in an efficient manner, optimizing the travel path and speed to meet the parameters.

6.4.6 Start Time and End Time Control:Evaluate the system's ability to initiate travel at the user-specified start time.Check synchronization of starting the journey with real-time clock functions.

Confirm the application can stop the vehicle or adjust its speed to match the user-specified end time.Test for safe completion of the journey within the designated time frame.

Step-by-Step Controls

Start Time Control :

- At 7:00 AM, the application checks the real-time clock and initiates the journey on time.
- The system starts the vehicle at a slow speed as specified.

This feature allows the user to set the time the vehicle should begin its journey. The system uses a real-time clock to trigger the start of the journey at the specified time.

End Time Control:

- If the journey completes the set distance before the end time of 7:30 AM, the vehicle will stop.
- If the journey takes longer than expected due to obstacles or other delays, the system will adjust the speed to ensure the journey is completed by 7:30 AM.
- The system may provide feedback to the user on progress and any adjustments made.
- End time control was adaptable to user changes and unexpected obstacles, allowing for efficient adjustments as needed.

This feature allows the user to set the time the vehicle should reach its destination or stop its journey. The system uses the end time and distance settings to manage the speed and time required to meet the target.

Speed and Distance Control:

- The vehicle travels at the slow speed specified by the user.
- As the vehicle moves, the system tracks the distance traveled using sensors such as wheel encoders or other measurement devices.
- When the vehicle reaches 1 meter of travel distance, the system prepares to stop the vehicle.

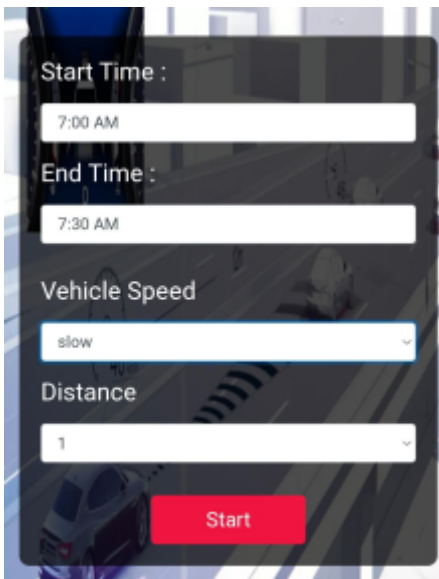
A screenshot of a mobile application interface for vehicle control. The interface is dark-themed with white text and input fields. It features four main settings: 'Start Time' set to '7:00 AM', 'End Time' set to '7:30 AM', 'Vehicle Speed' set to 'slow' (selected from a dropdown), and 'Distance' set to '1' (selected from a dropdown). A red 'Start' button is located at the bottom right of the settings panel. The background shows a blurred image of a vehicle on a road.

Fig: 6.4.6 Start Time and End Time Control

6.4.7 Test Results Scenarios Table 2

Scenario	Start Time	End Time	Speed	Distance (m)
1	6:00 AM	6:15 AM	Slow	1
2	7:00 AM	7:30 AM	Medium	4
3	8:00 AM	8:45 AM	High	3
4	9:00 AM	9:30 AM	Slow	2
5	10:00 AM	10:20 AM	Medium	2
6	11:00 AM	11:50 AM	High	4
7	12:00 PM	12:30 PM	Slow	3
8	1:00 PM	1:45 PM	Medium	4
9	2:00 PM	2:15 PM	High	1
10	3:00 PM	3:30 PM	Medium	2

Table 2 : 6.4.7 Test Results Scenarios

Explanation of Scenarios

Start Time: The time at which the journey begins. The start time varies in each scenario, demonstrating the flexibility of the system to begin travel at different times of the day.

End Time: The time at which the journey should end or when the vehicle should reach its destination. The end times are chosen based on the distance and speed parameters.

Speed: The speed setting for the vehicle, ranging from slow to high. This determines how fast the vehicle will travel.

Distance: The distance the vehicle is expected to travel in each scenario, ranging from 1 to 4 meters. This shows how the system can handle varying distances.

CHAPTER 7

CONCLUSION

The comprehensive testing of the advanced adaptive cruise control (AACC) system in the robotic mini car demonstrates its ability to effectively navigate autonomously, adjust speed dynamically, and maintain safe following distances. The system excels in obstacle detection and avoidance, ensuring reliable safety features and collision prevention. Through efficient regulation of speed and smooth driving maneuvers, it optimizes fuel usage and enhances user convenience. The AACC system exhibits robust performance and responsiveness across diverse driving scenarios. While future work may include optimizing control algorithms and exploring integration with emerging technologies, the system already provides a promising glimpse into the future of autonomous transportation. With continued refinement and expansion, the adaptive cruise control system has the potential to revolutionize personal mobility, offering safer and more efficient transportation options.

SUMMARY

The following conclusions can be drawn from the testing results:

1.Autonomous Navigation Capability: The mini car demonstrates robust autonomous navigation capabilities, accurately maneuvering through complex driving scenarios with precise steering and obstacle avoidance. The integration of multiple sensors and advanced control algorithms enables the vehicle to adapt to dynamic environmental conditions and navigate safely in real-world environments.

2.Effectiveness of Adaptive Cruise Control Algorithms: The ACC system effectively regulates vehicle speed, maintaining a safe following distance from preceding vehicles while responding to changes in traffic conditions in a timely manner. The implementation of adaptive control algorithms allows for smooth acceleration, deceleration, and speed adjustments, enhancing both safety and comfort for passengers.

3.Safety and Reliability Features: The system's safety features, including obstacle detection, collision avoidance, and emergency braking, demonstrate high reliability and

effectiveness in mitigating potential hazards. Real-time fault detection mechanisms and recovery actions ensure system robustness and maintain passenger safety even in challenging driving scenarios.

4.Efficiency and Convenience: The ACC system optimizes fuel efficiency by minimizing unnecessary acceleration and deceleration, resulting in reduced emissions and improved traffic flow. Users report a convenient and stress-free driving experience, particularly in congested urban areas, where the system's adaptive capabilities alleviate the burden of manual driving.

5.Future Directions and Optimization: While the ACC system has achieved significant milestones in functionality and performance, there remain opportunities for further optimization and enhancement. Future developments may focus on fine-tuning control algorithms, integrating additional sensor modalities for enhanced perception, and exploring advanced AI techniques for predictive analytics and decision-making.

6.Integration with Emerging Technologies: As the automotive industry continues to evolve, the ACC system can be further integrated with emerging technologies such as vehicle-to-vehicle (V2V) communication, vehicle-to-infrastructure (V2I) communication, and cloud-based services. These integrations will enable enhanced cooperative driving capabilities, real-time traffic management, and seamless connectivity with smart infrastructure.

7.Continued Testing and Validation: Ongoing testing and validation efforts will be essential to validate system performance under diverse conditions and ensure compliance with industry standards and regulatory requirements. Continuous monitoring and feedback from users and stakeholders will drive iterative improvements and refinement of the ACC system.

REFERENCES

- Vahidi, A.; Eskandarian, A. Research advances in intelligent collision avoidance and adaptive cruise control. *IEEE Trans. Intell. Transp. Syst.* 2004, 4, 143–153.
- Yang, L.; Zhao, X.; Wu, G.; Xu, Z.; Barth, M.; Hui, F.; Hao, P.; Han, M.; Zhao, Z.; Fang, S.; et al. Overview of Collaborative Ecological Driving Strategies for Intelligent Connected Vehicles. *J. Traffic Transp. Eng.* 2020, 20, 58–72.
- Hu, Z.; Su, R.; Zhang, K.; Xu, Z.; Ma, R. Resilient Event-Triggered Model Predictive Control for Adaptive Cruise Control Under Sensor Attacks. *IEEE/CAA J. Autom. Sin.* 2023, 10, 807–809.
- Wu, G.; Guo, X.; Zhang, L. Design of Multi objective Robust control Control Algorithm for Vehicle Adaptive Cruise Following. *J. Harbin Inst. Technol.* 2020, 48, 80–86.
- Gao, C.Z. Target Vehicle Selection Algorithm for Adaptive Cruise Control Based on Lane-changing Intention of Preceding Vehicle. *J. Mech. Eng.* 2021, 34, 390–407.
- Liu, Q.; Yang, L.; Gao, B.; Wang, J.; Li, K. Intelligent car following model based on cognitive risk Dynamic equilibrium. *Automot. Eng.* 2022, 44, 1627–1635.
- Jenness, J.W.; Lerner, N.D.; Mazor, S.; Osberg, J.S.; Tefft, B.C. Use of Advanced In-Vehicle Technology by Young and Older Early Adopters; NHTSA: Washington, DC, USA, 2020; pp. 810–828.
- Pan, Y.; Nie, X.; Li, Z.; Gu, S. Data-driven vehicle modeling of longitudinal dynamics based on a multibody model and deep neural networks. *Measurement* 2021, 180, 109541.
- Lien, C.; Shen, C.; Chang, H.; Hou, Y.; Yu, K. Mixed performance for robust fuzzy control of nonlinear autonomous surface vehicle via T-S model approach. *Asian J. Control* 2022, 24, 1059–1073.
- Liu, Y. Research on Longitudinal Dynamic Model and Control Strategy for Active Collision Avoidance. *J. Jiangsu Univ.* 2020, 174–193.

Xie, Y.; Wei, Z.; Zhao, L.; Jia, W.; Wu, C. Base on μ Robust lateral control for human-machine co driving of intelligent vehicles using a comprehensive approach. J. Mech. Eng. 2020, 56, 104–114.

Wang, Y.; Gao, S.; Wang, Y.; Wang, P.; Zhou, Y.; Xu, Y. Robust trajectory tracking control for autonomous vehicle subject to velocity-varying and uncertain lateral disturbance. Arch. Transp. 2021, 57, 7–23. Zhou, B.; Wu, X.; Wei, G. Base on μ Research on Robust control Control of Integrated Vehicle Active suspension. J. Vib. Eng. 2020, 30, 1029–1037.

Guo, N.; Zhang, X.; Zou, Y. Real time predictive control of path following to stabilize autonomous electric vehicles under extreme drive conditions. Automot. Innov. 2022, 5, 453–470. S.A.Bagloee, M. Tavana, M. Asadi and T. Oliver, "Autonomous vehicles: challenges opportunities and future implications for transportation policies", Journal of mode. trans., vol. 24, no. 4, pp. 284-303, 2021.

J. Ploeg, A. F. Serrarens and G. J. Heijenk, "Connect & drive: design and evaluation of cooperative adaptive cruise control for congestion reduction", Journal of Mode. Trans., vol. 19, no. 3, pp. 207-213, 2021.

C. Thompson, "Why driverless cars will be safer than human drivers", Business Insider Nordic, 2020.

T. Louw, R. Madigan, O. Carsten and N. Merat, "Were they in the loop during automated driving? links between visual attention and crash potential", Injury prevention, vol. 23, no. 4, pp. 281-286, 2020.

S. M. Casner, E. L. Hutchins and D. Norman, "The challenges of partially automated driving", Communications of the ACM, vol. 59, no. 5, pp. 70-77, 2020.

R. Duffney, A. Borowsky and H. Bar-Gera, "The effects of adaptive cruise control (acc) headway time on young-experienced drivers' over-taking tendency in a driving simulator", Transportation research part F: traffic psychology and behaviour, vol. 86, pp. 151-160, 2022.

- B. Besselink and K. H. Johansson, "String stability and a delay-based spacing policy for vehicle platoons subject to disturbances", IEEE Transactions on Automatic Control, vol. 62, no. 9, pp. 4376-4391, 2020.
- J. Hu, P. Bhowmick, F. Arvin, A. Lanzon and B. Lennox, "Cooperative control of heterogeneous connected vehicle platoons: An adaptive leader-following approach", IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 977-984, 2020.
- D. L. Luu, C. Lupu, L. S. Ismail and H. Alshareefi, "Spacing control of cooperative adaptive cruise control vehicle platoon", 2020 IEEE International Conference on Automation Quality and Testing Robotics (AQTR), pp. 1-6, 2020.
- K. Santhanakrishnan and R. Rajamani, "On spacing policies for highway vehicle automation", IEEE Transactions on intelligent transportation systems, vol. 4, no. 4, pp. 198-204, 2021.
- D. L. Luu, H. T. Pham, C. Lupu, T. B. Nguyen and S. T. Ha, "Research on cooperative adaptive cruise control system for autonomous vehicles based on distributed model predictive control", International Conference on System Science and Engineering, pp. 13-18, 2021.
- V. K. Vegamoor, S. Darbha and K. R. Rajagopal, "A review of automatic vehicle following systems", Journal of the Indian Institute of Science, vol. 99, no. 4, pp. 567-587, 2021.
- A. G. Ulsoy, H. Peng and M. Çakmakci, Automotive control systems, Cambridge University Press, 2022.
- D. L. Luu, C. Lupu and T. Van Nguyen, "Design and simulation implementation for adaptive cruise control systems of vehicles", 2022 22nd International Conference on Control Systems and Computer Science, pp. 1-6, 2022.

APPENDIX

A. SOURCE CODE

```
#include <Wire.h>

#include <ESP8266WiFi.h>

#include <ESP8266HTTPClient.h>

#include <WiFiClient.h>

#include <ArduinoJson.h>

const char* ssid = "iot";

const char* password = "12345678";

const char* serverName = "http://iotcloud22.in/3173_cruise_control/post_value.php";

WiFiClient client;

HTTPClient http;

int ir, count;

const int IN1 = D1;

const int IN2 = D2;

const int IN3 = D3;

const int IN4 = D4;

void setup() {

  Serial.begin(9600);

  pinMode(IN1, OUTPUT);

  pinMode(IN2, OUTPUT);

  pinMode(IN3, OUTPUT);

  pinMode(IN4, OUTPUT);

  pinMode(D5, INPUT);

  analogWrite(IN1, 0);
```

```

analogWrite(IN2, 0);
analogWrite(IN3, 0);
analogWrite(IN4, 0);
WiFi.begin(ssid, password);
Serial.println("Connecting");
while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
}
Serial.println("");
Serial.print("Connected to WiFi network with IP Address: ");
Serial.println(WiFi.localIP());
delay(1000);
}
void loop() {
  getdata();
}
void getdata() {
  if (WiFi.status() == WL_CONNECTED) {
    //HTTPClient http; //Object of class HTTPClient
    http.begin(client, "http://iotcloud22.in/3173\_cruise\_control/light.json");
    int httpCode = http.GET();
    //Check the returning code
    if (httpCode > 0) {
      // Parsing
    }
  }
}

```

```

}

StaticJsonDocument<256> doc;

DeserializationError error = deserializeJson(doc, http.getString());
Serial.println(http.getString());
if (error) {
  Serial.print(F("deserializeJson() failed: "));
  Serial.println(error.f_str());
  return;
}

//StaticJsonDocument<256> doc;

//  deserializeJson(doc, json);

//  auto error = deserializeJson(doc, json);
if (error) {
  Serial.print(F("deserializeJson() failed with code "));
  Serial.println(error.c_str());
  return;
}

String slow1 = doc["slow1"]; // "on"
String medium1 = doc["medium1"];
String high1 = doc["high1"];
String slow2 = doc["slow2"]; // "on"
String medium2 = doc["medium2"];
String high2 = doc["high2"];

```

```

String slow3 = doc["slow3"]; // "on"
String medium3 = doc["medium3"];
String high3 = doc["high3"];

// Extract values

Serial.print(F("Response:"));

ir = digitalRead(D5);

Serial.println(ir);

// if (ir == 0) {
//   count++;
//   analogWrite(IN1, 0);
//   analogWrite(IN2, 0);
//   analogWrite(IN3, 0);
//   analogWrite(IN4, 0);
// }

if (slow1 == "on1") {
  analogWrite(IN1, 51);
  analogWrite(IN2, 0);
  analogWrite(IN3, 51);
  analogWrite(IN4, 0);
  Serial.println("Forward ");
  if (ir == 0) {
    delay(250);
    analogWrite(IN1, 0);
    analogWrite(IN2, 0);

```

```

analogWrite(IN3, 0);
analogWrite(IN4, 0);
delay(250);
analogWrite(IN1, 71);
analogWrite(IN2, 0);
analogWrite(IN3, 71);
analogWrite(IN4, 0);
Serial.println("slow1 ir ");
delay(250);
}
} else if (slow1 == "off1") {
analogWrite(IN1, 0);
analogWrite(IN2, 0);
analogWrite(IN3, 0);
analogWrite(IN4, 0);
}
if (slow2 == "on2") {
analogWrite(IN1, 81);
analogWrite(IN2, 0);
analogWrite(IN3, 81);
analogWrite(IN4, 0);
Serial.println("Forward ");
if (ir == 0) {
analogWrite(IN1, 100);
analogWrite(IN2, 0);

```

```

analogWrite(IN3, 100);
analogWrite(IN4, 0);
Serial.println("slow2 ir ");
}
} else if (slow2 == "off2") {
analogWrite(IN1, 0);
analogWrite(IN2, 0);
analogWrite(IN3, 0);
analogWrite(IN4, 0);
}
if (slow3 == "on3") {
analogWrite(IN1, 100);
analogWrite(IN2, 0);
analogWrite(IN3, 100);
analogWrite(IN4, 0);
Serial.println("Forward ");
if (ir == 0) {
analogWrite(IN1, 132);
analogWrite(IN2, 0);
analogWrite(IN3, 132);
analogWrite(IN4, 0);
Serial.println("slow3 ir ");
}
} else if (slow3 == "off3") {
analogWrite(IN1, 0);

```

```

analogWrite(IN2, 0);
analogWrite(IN3, 0);
analogWrite(IN4, 0);
}
if (medium3 == "on3") {
analogWrite(IN1, 150);
analogWrite(IN2, 0);
analogWrite(IN3, 150);
analogWrite(IN4, 0);
Serial.println("Forward ");
if (ir == 0) {
analogWrite(IN1, 170);
analogWrite(IN2, 0);
analogWrite(IN3, 170);
analogWrite(IN4, 0);
Serial.println("medium3 ");
}
} else if (medium3 == "off3") {
analogWrite(IN1, 0);
analogWrite(IN2, 0);
analogWrite(IN3, 0);
analogWrite(IN4, 0);
Serial.println("fan_off");
}
if (medium1 == "on1") {

```

```

analogWrite(IN1, 110);
analogWrite(IN2, 0);
analogWrite(IN3, 110);
analogWrite(IN4, 0);
Serial.println("Forward ");
if (ir == 0) {
analogWrite(IN1, 130);
analogWrite(IN2, 0);
analogWrite(IN3, 130);
analogWrite(IN4, 0);
Serial.println("medium1 ");
}
} else if (medium1 == "off1") {

```

```

analogWrite(IN1, 0);
analogWrite(IN2, 0);
analogWrite(IN3, 0);
analogWrite(IN4, 0);
Serial.println("fan_off");
}

```

```

if (medium2 == "on2") {
analogWrite(IN1, 130);
analogWrite(IN2, 0);
analogWrite(IN3, 130);
analogWrite(IN4, 0);
}

```

```

Serial.println("Forward ");
Serial.println("fan_on");
if (ir == 0) {
  analogWrite(IN1, 150);
  analogWrite(IN2, 0);
  analogWrite(IN3, 150);
  analogWrite(IN4, 0);
  Serial.println("medium2 ");
}
} else if (medium2 == "off2") {
  analogWrite(IN1, 0);
  analogWrite(IN2, 0);
  analogWrite(IN3, 0);
  analogWrite(IN4, 0);
  Serial.println("fan_off");
}
if (high1 == "on1") {
  analogWrite(IN1, 200);
  analogWrite(IN2, 0);
  analogWrite(IN3, 200);
  analogWrite(IN4, 0);
  Serial.println("Forward ");
  if (ir == 0) {
    analogWrite(IN1, 220);
    analogWrite(IN2, 0);

```

```

analogWrite(IN3, 220);
analogWrite(IN4, 0);
Serial.println("high1");
}
} else if (high1 == "off1") {
analogWrite(IN1, 0);
analogWrite(IN2, 0);
analogWrite(IN3, 0);
analogWrite(IN4, 0);
Serial.println("motor off");
}
if (high2 == "on2") {
analogWrite(IN1, 220);
analogWrite(IN2, 0);
analogWrite(IN3, 220);
analogWrite(IN4, 0);
Serial.println("Forward ");
if (ir == 0) {
analogWrite(IN1, 240);
analogWrite(IN2, 0);
analogWrite(IN3, 240);
analogWrite(IN4, 0);
Serial.println("high2");
}
} else if (high2 == "off2") {

```

```

analogWrite(IN1, 0);
analogWrite(IN2, 0);
analogWrite(IN3, 0);
analogWrite(IN4, 0);
Serial.println("motor off");
}
if (high3 == "on3") {
analogWrite(IN1, 255);
analogWrite(IN2, 0);
analogWrite(IN3, 255);
analogWrite(IN4, 0);
Serial.println("Forward ");
Serial.println("motor on");
} else if (high3 == "off3") {
analogWrite(IN1, 0);
analogWrite(IN2, 0);
analogWrite(IN3, 0);
analogWrite(IN4, 0);
Serial.println("motor off");
}
http.end(); //Close connection
//Serial.println(flag);
delay(100);}

```

Application Source Code:

```
<?php
session_start();

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta http-equiv="refresh" content="30">

<title>Adaptive Cruise Control System</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css">

<script src="https://cdn.jsdelivr.net/npm/jquery@3.6.4/dist/jquery.slim.min.js"></script>

<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js"></script>

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Abril+Fatface&display=swap"
rel="stylesheet">

<style>

body

{

height:100vh;
```

```

background-repeat:no-repeat;
background-size:100% 100%;
background-attachment:fixed;
color:black;
background-image:url("image/one.webp");
}

```

```

</style>

```

```

</head>

```

```

<body>

```

```

<h2 class="text-center text-white py-2" style="background-color:rgba(0,0,0,0.7);">Adaptive
Cruise Control System</h2>

```

```

<div class="container-fluid">

```

```

<div class="row">

```

```

<div class="col-md-6 offset-md-2 mt-5 mb-3">

```

```

<div class="card mt-4" style="background-color:rgba(0,0,0,0.7);border-radius:10px;">

```

```

<div class="card-body">

```

```

<form action="" method="post">

```

```

<div class="form-row">

```

```

<div class="form-group col-md-6">

```

```

<label for="email" class="text-white"><h4>Start Time :

```

```

<?=@$_SESSION["start_time"]?></h4></label>

```

```

<input type="time" class="form-control" name="start_time" id="email" required>

```

```

</div>

```

```

<div class="form-group col-md-6">

```

```

<label for="email" class="text-white"><h4>End Time :
<?=@$_SESSION["end_time"]?></h4></label>

<input type="time" class="form-control" id="email" name="end_time" required>

</div>

</div>

<div class="form-row mt-3">

<div class="form-group col-md-6">

<label for="email" class="text-white"><h4>Vehicle Speed</h4></label>

<select class="form-control" name="speed" required>

<option value="">Choose vehicle speed</option>

<option value="slow">slow</option>

<option value="medium">medium</option>

<option value="high">high</option>

</select>

</div>

<div class="form-group col-md-6">

<label for="email" class="text-white"><h4>Distance</h4></label>

<select class="form-control" name="distance" required>

<option value="1">1</option>

<option value="2">2</option>

<option value="3">3</option>

<option value="4">4</option>

</select>

</div>

</div>

```

```

<center>

  <input type="submit" class="btn btn-danger btn-lg mt-3" style="width:150px;" value="Start"
  name="start">

</center>

</form>

</div>

</div>

</div>

<div class="col-md-4 mt-5 mb-3">

  <div class="card mt-4" style="background-color:rgba(0,0,0,0.7);border-radius:10px;">

    <div class="card-body">

      <form action="" method="post">

        <div class="form-row mt-3">

          <div class="form-group col-md-6">

            <label for="email" class="text-white"><h4>Vehicle Speed</h4></label>

            <select class="form-control" name="speed1" required>

              <option value="">Choose vehicle speed</option>

              <option value="slow">slow</option>

              <option value="medium">medium</option>

              <option value="high">high</option>

            </select>

          </div>

          div class="form-group col-md-6">

            <label for="email" class="text-white"><h4>Distance</h4></label>

```

```

<select class="form-control" name="distance1" required>
<option value="1">1</option>
<option value="2">2</option>
<option value="3">3</option>
<option value="4">4</option>
</select>

</div>

</div>

<center>

<input type="submit" class="btn btn-danger btn-lg mt-3" style="width:150px;" value="Start"
name="start1">

<input type="submit" class="btn btn-danger btn-lg mt-3" style="width:150px;" value="Stop"
name="stop">

</center>

</form>

</div>

</div>

</div>

</div>

</div>

<?php
if(isset($_POST["start"]))
{
$start_time=$_POST["start_time"];
$end_time=$_POST["end_time"];

```

```

$_SESSION["start_time"]=$start_time;
$_SESSION["end_time"]=$end_time;
$speed = $_POST["speed"];
$_SESSION["speed"]=$speed;
$distance = $_POST["distance"];
$_SESSION["distance"]=$distance;
}
if(isset($_POST["stop"]))
{
$speed2 = $_POST["speed1"];
$distance2 = $_POST["distance1"];
unset($_SESSION["start_time"]);
    $v=json_encode(array($speed2.$distance2=>"off".$distance2));
file_put_contents("light.json",$v);
}
if(isset($_POST["start1"]))
{
$speed3 = $_POST["speed1"];
$distance3 = $_POST["distance1"];
unset($_SESSION["start_time"]);
    $p=json_encode(array($speed3.$distance3=>"on".$distance3));
file_put_contents("light.json",$p);
}
if(isset($_SESSION["start_time"]))
{

```

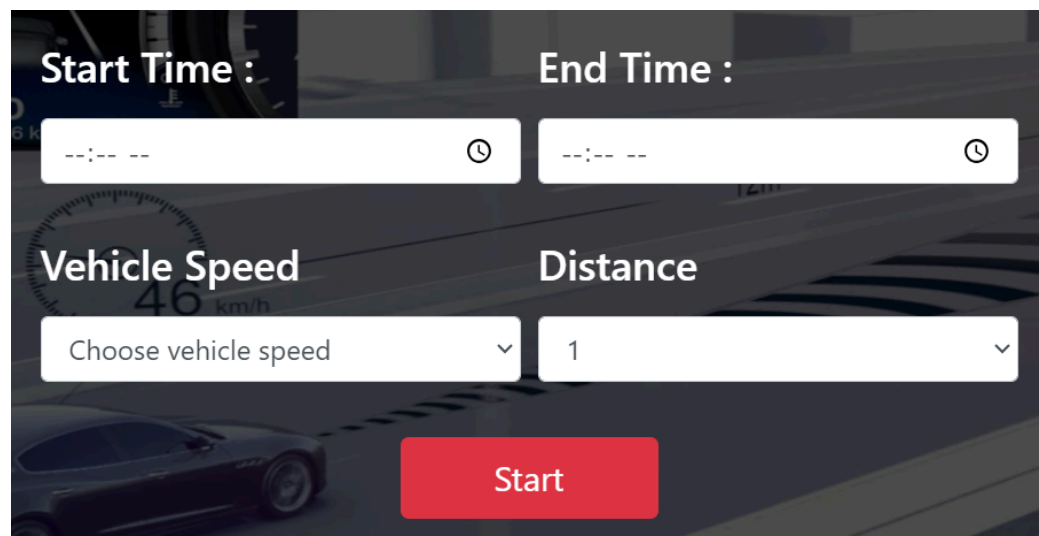
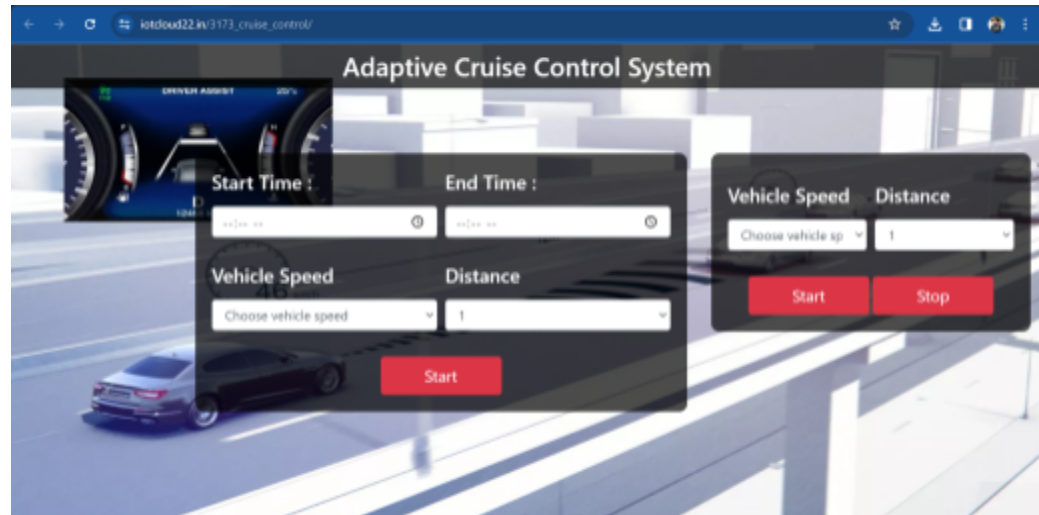
```

$startTime = $_SESSION["start_time"];
$endTime = $_SESSION["end_time"];
$speed1 = $_SESSION["speed"];
$distance1 = $_SESSION["distance"];

// Get the current time
date_default_timezone_set('Asia/Kolkata');
$currentDateTime = new DateTime();
$currentTime = $currentDateTime->format('H:i:s');
file_put_contents("light.json",$a);
} else {
//echo "time is in between";
$a=json_encode(array($speed1.$distance1=>"on".$distance1));
file_put_contents("light.json",$a);
}
}
</body>
</html>

```

B. SCREENSHOTS



Vehicle Speed **Distance**

slow 1

Start Stop

Vehicle Speed **Distance**

Choose vehicle sp 1

Start Stop

Vehicle Speed **Distance**

medium 2

Start Stop

Vehicle Speed **Distance**

high 4

Start Stop

Start Time :
8:30 AM

End Time :
9:00 AM

Vehicle Speed
high

Distance
4

Start

Start Time :
8:30 AM

End Time :
9:00 AM

Vehicle Speed
medium

Distance
4

Start

Start Time :
7:00 AM

End Time :
7:30 AM

Vehicle Speed
slow

Distance
1

Start

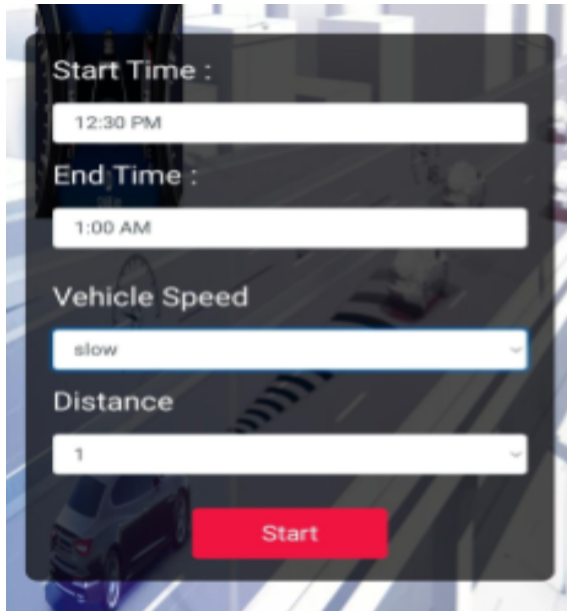
Start Time :
8:30 AM

End Time :
9:00 AM

Vehicle Speed
high

Distance
4

Start



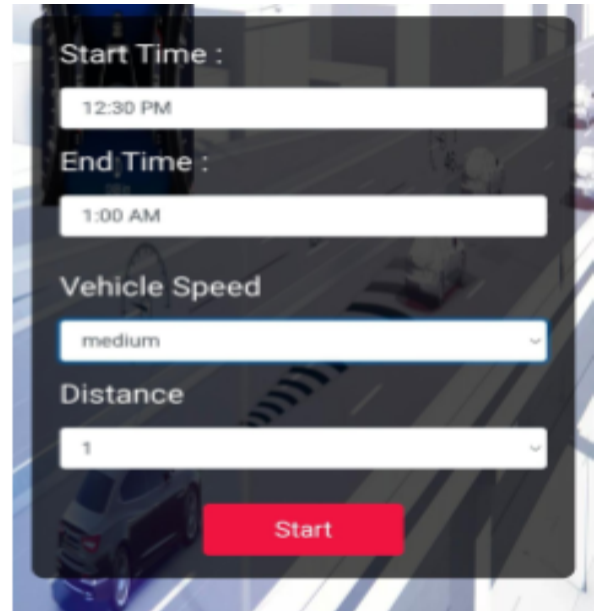
Start Time :
12:30 PM

End Time :
1:00 AM

Vehicle Speed
slow

Distance
1

Start



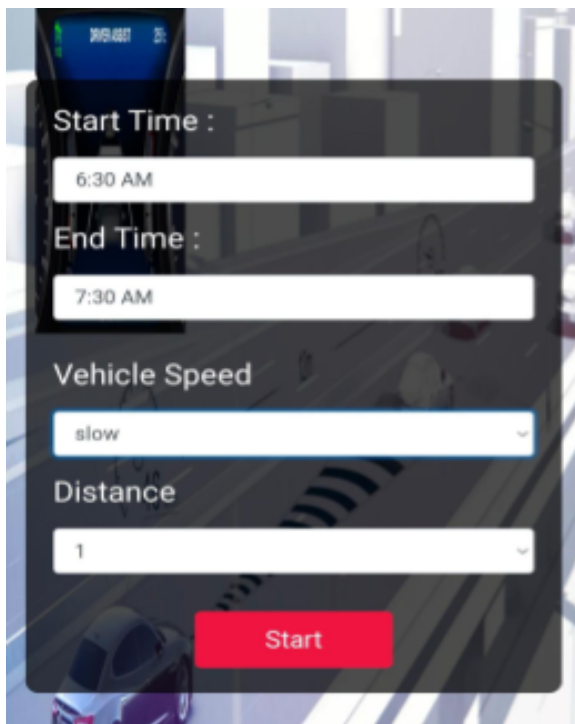
Start Time :
12:30 PM

End Time :
1:00 AM

Vehicle Speed
medium

Distance
1

Start



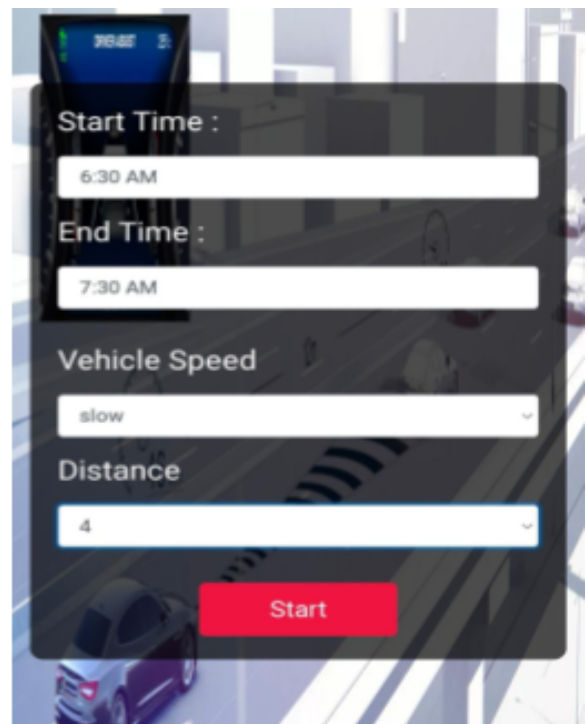
Start Time :
6:30 AM

End Time :
7:30 AM

Vehicle Speed
slow

Distance
1

Start



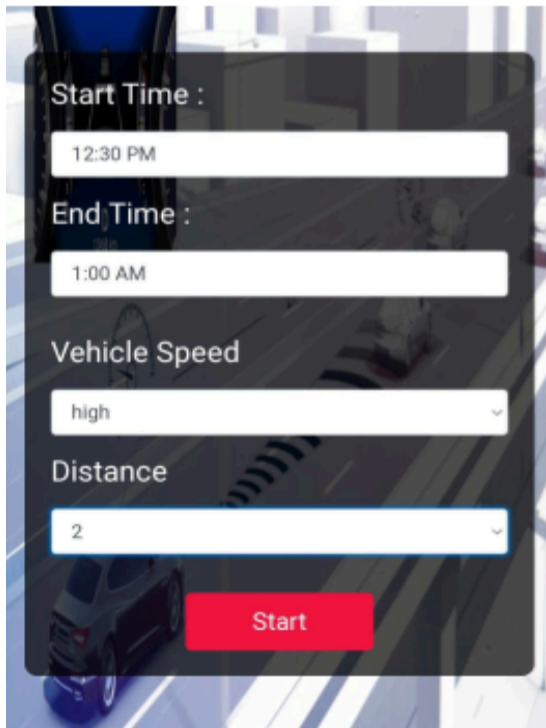
Start Time :
6:30 AM

End Time :
7:30 AM

Vehicle Speed
slow

Distance
4

Start



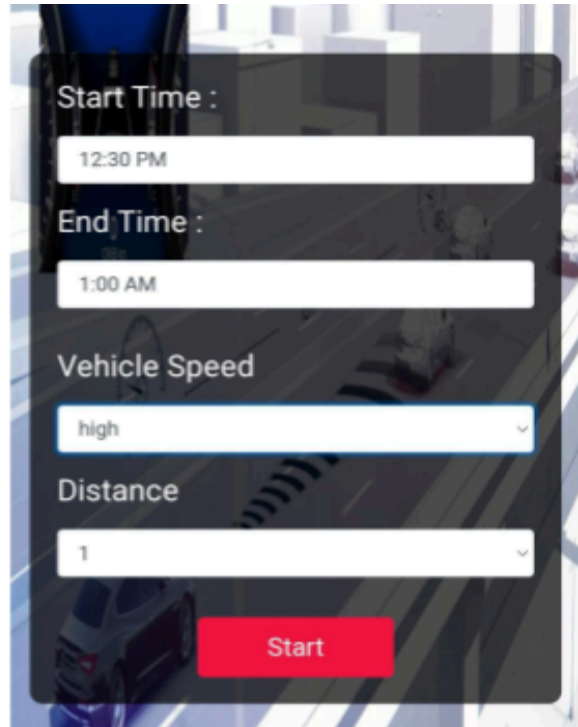
Start Time :
12:30 PM

End Time :
1:00 AM

Vehicle Speed
high

Distance
2

Start



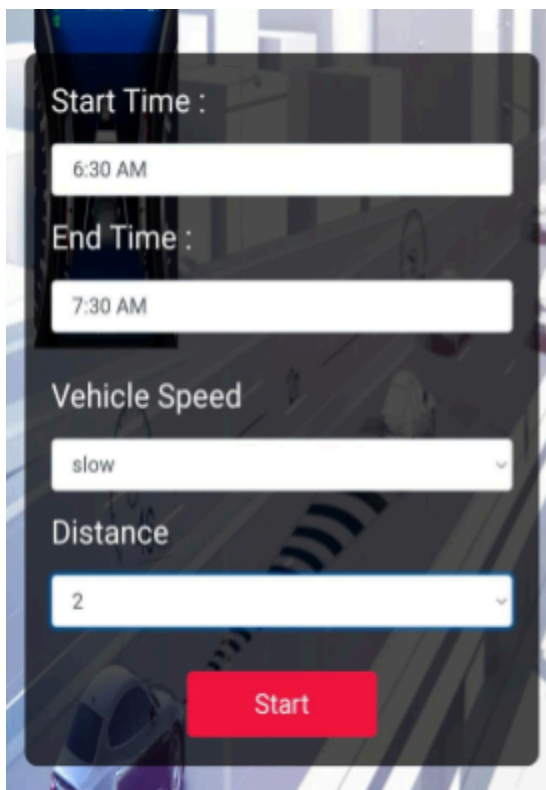
Start Time :
12:30 PM

End Time :
1:00 AM

Vehicle Speed
high

Distance
1

Start



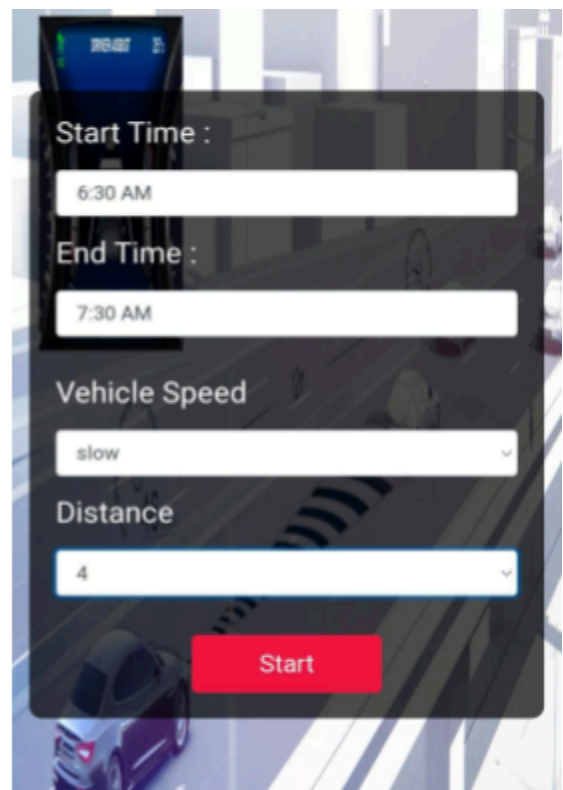
Start Time :
6:30 AM

End Time :
7:30 AM

Vehicle Speed
slow

Distance
2

Start



Start Time :
6:30 AM

End Time :
7:30 AM

Vehicle Speed
slow

Distance
4

Start

C. RESEARCH PAPER

ADVANCED ADAPTIVE CRUISE CONTROL SYSTEM

P Meiyazhagan*, K. Ashok kumar

Department of Computer Science & Engineering
Sathyabama Institute of Science and Technology

Chennai-600119, Tamil Nadu, India

¹meiyazhaganmailbox@gmail.com ²ashokkumar.cse@sathyabama.ac.in

Abstract— This paper presents the development and implementation of an advanced adaptive cruise control (ACC) system in an autonomous vehicle. The proposed system aims to achieve efficient travel by managing speed, distance, start time, and end time based on user input and real-time environmental conditions. This technology aims to increase the autonomy and usefulness of cars through intelligent automation. The brains behind the seamless data interchange between various components of the Internet of Things are the IoT microcontrollers. Ultrasonic sensors supply critical data that is necessary for real-time environmental awareness, enabling the system to modify its direction and speed in reaction to its environment.

The motor driver and DC motor that make up the propulsion system allow for exact control over the vehicle's motions. The addition of a robot module improves the system's flexibility by enabling it to navigate over complex terrain and adapt to changing conditions. A unique battery management system extends the life of the vehicle by guaranteeing optimal power consumption.

In addition to promising a better economy, this state-of-the-art cruise control system has an adaptive function that enables it to respond appropriately to changes in the surrounding environment. The Adaptive Cruise Control System is a significant development in intelligent transportation systems, paving the way for safer, more efficient, and versatile automobiles in the era of smart mobility. It can traverse challenging terrain, avoid obstructions, and adjust speed.

Keywords— Adaptive Cruise Control System, intelligent transportation systems, automobiles.

I. INTRODUCTION

The IoT-powered Adaptive Cruise Control System aims to revolutionize the automobile industry by combining state-of-the-art technology to enhance vehicle autonomy and usefulness. The technology seeks to enhance real-time environmental awareness by utilizing ultrasonic sensors and Internet of Things capabilities, enabling the vehicle to adapt to changing road conditions and obstacles. The precise control over vehicle motions provided by the propulsion system, which consists of motor drivers and DC motors, ensures responsive and efficient driving experiences. Cruise control systems have long been integrated into vehicles to offer drivers a convenient means of maintaining a consistent speed on highways. However, conventional cruise control systems lack the flexibility to adapt to changing road conditions and traffic patterns. In response to this limitation, our research introduces an innovative approach by integrating distance and time-based traveling parameters into the cruise control system. This advancement aims to provide drivers with greater control over their driving experience while ensuring safety and efficiency.

II. RELATED WORKS

The basis of any mentoring course is Performance Analysis Systems. With the guidance and assistance of statistical data, an individual will surely progress, which also helps the business that employs him, will succeed. A background inquiry is done in order to look at comparable existing techniques that are utilised to analyse student performance. Before starting our literature review, we learn about three existing systems that are comparable to the one that is being proposed.

Umapath et al (2018), The field of study on adaptive cruise control is expanding since it helps drivers in a variety of traffic situations and lightens their workload. As cruise control measures the gap between our car and the one in front of us, ACC disperses itself from it, maintaining our car's speed in line with the one in front of us. The actuators, control devices, and sensors that make up ACC are diverse. The goal of this technology is to lower the number of traffic accidents while also making driving more comfortable.

Zhu Yunhai, (2023), This study provides a robust controller for the automotive adaptive cruise control system, which consists of two layers: the upper and lower structures, based on the μ control theory, to increase the anti-interference and resilience of the system in car-following mode. The higher controller computes the predicted acceleration of the car using an optimal control approach based on the safety distance and two vehicle speeds, and it computes the theoretical safety distance between two autos based on the present operating circumstances. Conversely, the lower μ integrated controller of a car longitudinal dynamics system is constructed in this research using the adaptive cruise control system's performance criteria, and it is solved in Matlab. Subsequently, simulation and computation show that the dual-layer LQR- μ controller that was designed has good robustness and robust performance, which can greatly enhance the vehicle's

anti-interference capability and driving safety during the subsequent cruise process.

Nishant Doshia et al (2022), The buzzword of the present era of technology, the Internet of Things (IoT), gives commonplace items intelligence. The idea of the Internet of Things goes beyond computers and smartphones to create intelligent homes, cars, and communities. The frequency of traffic accidents is increasing, which is the most urgent problem in connection to expanding car use. The problem with contemporary cars is that they don't provide us with safety, which might result in deadly collisions. The idea of Internet of Things (IoT) in cars was created to guarantee road conditions' efficiency and safety. It is a mobile communication system that is dynamic and works with humans and automobiles to provide real-time road traffic information, ensure passenger comfort, and boost transportation efficiency. The study examines how smart vehicles have changed over the course of the six automation phases and contrasts different IoT-based approaches.

Wang Zengcai (2022), This work proposes a unique adaptive cruise control (ACC) method that is based on active disturbance rejection control (ADRC) and model predictive control (MPC). The higher controller of the ACC system in this article employs an MPC algorithm. The top controller will provide the necessary acceleration to the lower controller by taking all relevant factors into account. Additionally, an MPC assisted by predictive estimation of acceleration is presented in order to overcome oscillations in the vehicle's acceleration and improve the accuracy of the predictive model in the MPC controller. For the lower-level controller to regulate the throttle and brake actuators, it is challenging to create an accurate vehicle dynamic model because of the uncertainties associated with vehicle data and the road environment. Thus, to improve control accuracy and reduce the impact of internal or external disturbance, feed-forward control based on a vehicle dynamic model (VDM) and compensating

control based on ADRC are employed. Ultimately, traffic tests confirmed the suggested ideal ACC system configuration. The findings demonstrate that, in comparison to the conventional ACC controller, ACC with APE can precisely manage the host vehicle's tracking with reduced acceleration fluctuation. Furthermore, the ACC-APE-ADRC controller may still precisely and swiftly track the intended acceleration even in the event that the vehicle's mass and the road's slope are altered.

M.Ramaswamy et al (2022), In order to develop a multimodal speed control mechanism for a double cab off-road electric vehicle working in open port settings, the article presents an Internet of Things platform. It makes an effort to incorporate an embedded system that monitors the vehicle and its payload using an ARM core-based STM32 Microcontroller. Based on this information, it suggests adjusting the speed in response to real-time fluctuations. The system design takes into account the impact of the following factors: battery SoC (State of Charge), battery SoH (State of Health), vehicle speed, engine speed, passenger occupancy status, and real-time navigation utilising a variety of sensors and GPS-based Internet of Things maps. It creates a closed-loop monitoring plan to adhere to a set of guidelines that permits the car to run safely for an extended period of time. The benefits and vehicle performance in connection to the algorithm's efficiency are established by the results of the modular test simulation process, which were achieved using the embedded-C firmware. These results cover a large dynamic range of varied cargo and passenger combinations.

III . MATERIALS & METHODS

Existing System

The majority of automobiles now use cruise control systems that rely on outdated mechanical or electrical parts that are not particularly adaptable to

changing circumstances. Conventional cruise control systems only maintain the driver's speed; they are unable to react to environmental changes on their own. These systems are less flexible to changing road conditions and obstacles in real time since they often lack IoT capabilities and sophisticated sensors. Furthermore, the existing configuration lacks sophisticated motor control systems and variable robot modules, which limits the system's adaptability to different terrains and circumstances. Given how quickly technology is developing, it is obvious that a more advanced and responsive cruise control system is needed. One that uses IoT, ultrasonic sensors, and improved motor control.

Proposed System

The proposed Adaptive Cruise Control System offers a significant advance over the present versions by integrating state-of-the-art technology, resulting in a more intelligent and flexible solution. The primary processing unit of this system is an IoT microcontroller, which facilitates efficient coordination and communication between various components. Because of the carefully placed ultrasonic sensors that provide real-time environmental data, the gadget may dynamically change its speed and direction in reaction to its environment.

A strong motor driver, DC motor, and flexible robot module are added to the system to increase its capabilities and enable it to move swiftly over a range of surfaces. The proposed system is intended to have adaptive capabilities that go beyond simple cruise control; it will be intelligent enough to respond to obstacles and changes in its surroundings. The integration of a state-of-the-art battery management system ensures efficient power use, hence increasing the vehicle's total longevity and sustainability.

With this proposed system, we see a more sophisticated, safe, and adaptable cruise control system that not only meets but exceeds the evolving demands of modern transportation. By using cutting-edge

technology, the Adaptive Cruise Control System represents a paradigm shift towards intelligent transportation solutions for the future.

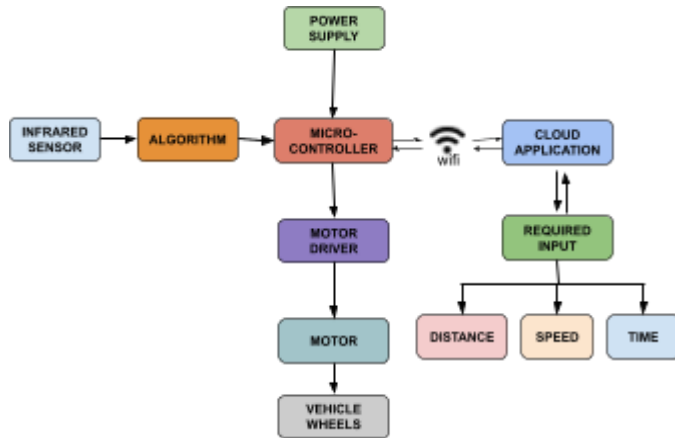


Figure 1. Proposed Block Diagram

Methodology

The creation of the Adaptive Cruise management System utilizing IoT follows a rigorous process to include cutting-edge technologies for enhanced vehicle management and flexibility. The first step in the procedure is to choose and integrate IoT microcontrollers to serve as the central processing unit, allowing for seamless data flow between components. The system may dynamically alter its direction and speed in reaction to its surroundings because ultrasonic sensors enable the real-time collecting of environmental data.

The vehicle's motion may be precisely controlled by the propulsion system, which is made up of motor drivers and DC motors. This enhances the vehicle's responsiveness and efficiency when driving. Evaluate the system's ability to initiate travel at the user-specified start time. Check synchronization of starting the journey with real-time clock functions.

Confirm the application can stop the vehicle or adjust its speed to match the user-specified end time. Test for safe completion of the journey within the designated time frame. Verify that the application can accurately set

and adjust the vehicle's speed according to user input. Assess the responsiveness of speed changes in different traffic and environmental conditions. Test the application's ability to set and monitor the specified travel distance. Ensure the vehicle stops or alters course once the target distance is reached.

Implementation

To implement the Adaptive Cruise Control System via the Internet of Things, a number of hardware components need to be combined, including a microcontroller, an ultrasonic sensor, a DC motor, a motor driver, a robot module, and a WiFi module. Together, these components enable the system to travel over a range of terrains with effectiveness, modify route and speed, and gather ambient data in real time.

Coordinating communication between the many hardware components is a critical function of the microcontroller, which acts as the system's central processing unit. In order for the system to make wise decisions and adapt to its environment, it needs the ambient data that the ultrasonic sensor provides.

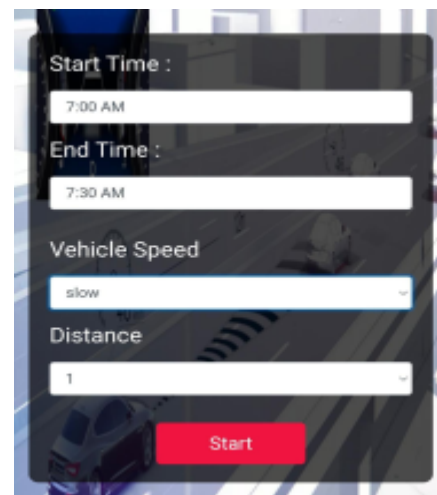


Figure 2. Cloud Application Interface

The robot module enhances the system's ability to adjust to changing circumstances and travel across

difficult terrain, while the propulsion system, which comprises of a DC motor and motor driver, provides the driver with precise control over the vehicle's motions.

A state-of-the-art battery management system is also fitted to ensure maximum power efficiency and prolong the vehicle's useful life. By using cutting-edge technology and IoT capabilities, the Adaptive Cruise Control System offers an increasingly intelligent, secure, and adaptive solution for modern transportation demands.

In general, the deployment of the IoT-based Adaptive Cruise Control System seeks to transform intelligent transportation systems by offering cutting-edge functionality and flexibility to satisfy the ever-evolving needs of modern mobility.

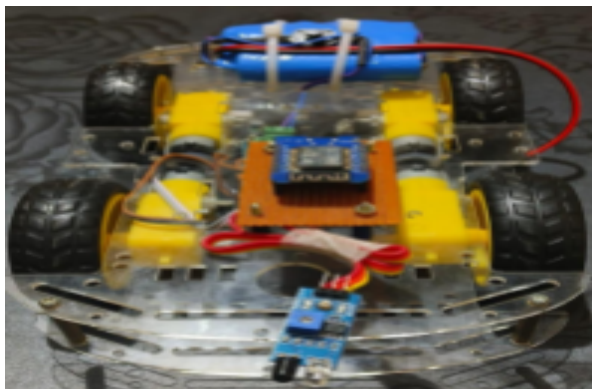


Figure 3.Front View

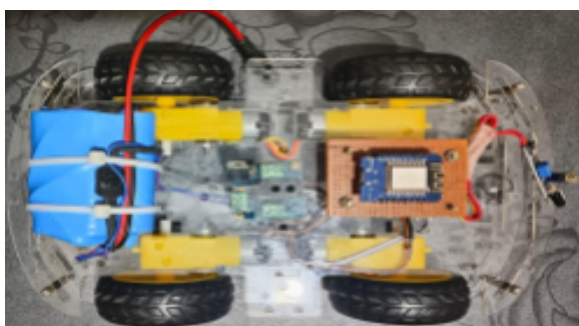


Figure 4 .Front View

Experimental Results

It is projected that the Adaptive Cruise Control System with IoT will give users a more intuitive, adaptable, and cost-effective driving experience. By integrating cutting-edge technologies including IoT, ultrasonic sensors, motor drivers, DC motors, robot modules, and effective battery management, the system aims to increase vehicle autonomy and usefulness.

Users may expect the system to dynamically adjust speed and direction in response to changing road conditions and obstacles, giving them a safer and more responsive driving experience overall. The system's capacity to adjust to various environments and settings, as well as to respond sensibly to impediments and environmental changes, guarantees a more versatile and secure cruise control solution.

Scenario	Speed	Distance(m)
1	Slow	1
2	Medium	2
3	High	4
4	Slow	4
5	Medium	1
6	High	2
7	Slow	2
8	Medium	3
9	High	3
10	Slow	3

Table 1. Test Results Scenarios

Additionally, the incorporation of advanced battery management ensures effective power usage, prolonging the vehicle's life and promoting sustainability.

Scenario	Start Time	End Time	Speed	Distance (m)
1	6:00 AM	6:15 AM	Slow	1
2	7:00 AM	7:30 AM	Medium	4
3	8:00 AM	8:45 AM	High	3
4	9:00 AM	9:30 AM	Slow	2
5	10:00 AM	10:20 AM	Medium	2
6	11:00 AM	11:50 AM	High	4
7	12:00 PM	12:30 PM	Slow	3
8	1:00 PM	1:45 PM	Medium	4
9	2:00 PM	2:15 PM	High	1
10	3:00 PM	3:30 PM	Medium	2

Table 2. Test Results Scenarios

To sum up, the expected outcomes of the Internet of Things-driven Adaptive Cruise Control System signify a revolutionary change in intelligent transportation solutions, offering a more advanced and user-focused driving experience that meets the evolving needs of contemporary mobility.

IV. CONCLUSIONS

To summaries, The comprehensive testing of the advanced adaptive cruise control (AACC) system in the robotic mini car demonstrates its ability to effectively navigate autonomously, adjust speed dynamically, and maintain safe following distances. The system excels in obstacle detection and avoidance, ensuring reliable safety features and collision prevention. Through efficient regulation of speed and smooth driving maneuvers, it optimizes fuel usage and enhances user convenience. The AACC system exhibits robust performance and responsiveness across diverse driving scenarios. While future work may include optimizing control algorithms and exploring integration with emerging technologies, the system already provides a promising glimpse into the future of autonomous transportation. With continued refinement and expansion, the adaptive cruise control

system has the potential to revolutionize personal mobility, offering safer and more efficient transportation options.

V. REFERENCES

1. Vahidi, A.; Eskandarian, A. Research developments in adaptive cruise control and intelligent collision avoidance. 2021, 4, 143–153 in IEEE Trans. Intell. Transp. Syst.
2. Barth, M.; Hui, F.; Hao, P.; Han, M.; Zhao, Z.; Fang, S.; Yang, L.; Zhao, X.; Wu, G.; Xu, Z.; et al. An overview of cooperative ecological driving techniques for smart, networked cars. 2020; 20 J. Traffic Transp. Eng., 58–72.
3. Ioannou, P.A.; Zhang, J. Environmental and fuel economic aspects of large truck longitudinal control in mixed traffic. 2020, 7, 92–104; IEEE Trans. Intell. Transp. Syst.
4. Zhang, K.; Xu, Z.; Ma, R.; Su, R.; Hu, Z. Robust Event-Triggered Model Predictive Control for Adaptive Cruise Control in the Face of Sensor Attacks. Autom. Sin. IEEE/CAA J. 2023, 10, 807–809.
5. Design of Multi-objective Robust Control Algorithm for Vehicle Adaptive Cruise Following by Wu, Guo, and Zhang. Harbin Institute of Technology, J. (2019) 48, 80–86.
6. Target Vehicle Selection Algorithm for Adaptive Cruise Control Based on the Preceding Vehicle's Intention to Change Lanes, Gao, C.Z. 2021; J. Mech. Eng. 34: 390–407.
7. Yang, L.; Liu, Q.; Wang, J.; Gao, B.; Li, K. An intelligent vehicle that adheres to a cognitive risk dynamic equilibrium model. 2020; Automot. Eng. 44: 1627–1635.
8. Use of Advanced In-Vehicle Technology by Young and Older Early Adopters; NHTSA: Washington, DC, USA, 2018; pp. 810–828; Jenness, J.W.; Lerner, N.D.; Mazon, S.; Osberg, J.S.; Tefft, B.C.
9. Gu, S.; Li, Z.; Nie, X.; Pan, Y. longitudinal dynamics of vehicles modelled by data using deep neural networks and a multibody model. Measurements: 180, 109541, 2021.
10. Yu, K.; Chang, H.; Shen, C.; Lin, C.; and Hou, Y. Mixed results for nonlinear autonomous surface vehicle resilient fuzzy control using the T-S model technique. Asian J. Control 24, 1059–1073. 2022.
11. Liu, Y. Investigation into Control Strategy and Longitudinal Dynamic Model for Active Collision Avoidance. (2020) J. Jiangsu Univ., 174–193.