FINAL YEAR PROJECT REPORT

ADAPTIVE CRUISE CONTROL FOR VEHICLES

By

A/C ZAID ALI

Pak/20095040, 95(B) EC

ADVISOR

SQUADRON LEADER DR SYED KHURRAM MAHMUD

CO-ADVISOR

SQUADRON LEADER FARRUKH PERVEZ

COLLEGE OF AERONAUTICAL ENGINEERING

PAF Academy, Asghar Khan, Risalpur

March 27, 2024

ADAPTIVE CRUISE CONTROL FOR VEHICLES

By

A/C ZAID ALI

Pak/20095040, 95(B) EC

ADVISOR

SQUADRON LEADER DR KHURRAM MAHMUD

CO-ADVISOR

SQUADRON LEADER FARRUKH PERVEZ

Report submitted in partial fulfillment of the requirements for the degree of Bachelors of Engineering in Avionics, (BE Avionics)

In

COLLEGE OF AERONAUTICAL ENGINEERING

PAF Academy, Asghar Khan, Risalpur

March 27, 2024

Approval

It is certified that the contents and form of the project entitled "Adaptive Cruise Control For Vehicles" submitted by Aviation Cadet Zaid Ali have been found satisfactory for the requirement of the degree.

Advisor: Sqn Ldr Dr.Syed Khurram Mahmud

Signature: ________________

Date: ________________

Co-Advisor: Sqn Ldr Farrukh Pervaiz

Signature: ________________

Date: ________________

Dedication

I want to take this opportunity to express my sincere gratitude to all those who have played a significant role in making this project possible. I would like to extend a special thanks to my loving family and my dedicated advisor, whose unwavering support and guidance have been instrumental in my academic success. Without their encouragement and support, I would not have been able to reach this point. To my parents, in particular, I owe a debt of gratitude for their selfless sacrifices, endless encouragement, and unwavering belief in me. This report is a tribute to them and all those who have contributed to my journey, and I am honored to share this achievement with them.

Acknowledgement

I express my sincere gratitude to Allah Almighty, who bestowed upon me the strength and determination to complete this project to the best of my abilities. My parents, whose unfailing love, unrelenting support, and steadfast prayers have been the compass in my life, deserve the utmost gratitude. Without their support, this accomplishment would not have been possible. I am immensely grateful to my advisor, Sqn Ldr Dr. Khurram Mahmud, for his constant guidance, invaluable feedback, and unwavering support. His encouragement and mentorship have been instrumental in shaping my research skills and intellectual growth. I also extend my heartfelt thanks to my co-advisor, Sqn Ldr Farrukh Pervez, for his valuable input and for sharing his expertise in the field. I want to acknowledge the invaluable assistance of Lab Engineer Saddam, Avn Cdt Asad, and Avn Cdt Sammi, who provided me with their support and help whenever I needed it the most. I am grateful to all my teachers and colleagues who have contributed to my academic and professional growth in various ways. Finally, I would like to acknowledge the support of my friends and family members, who have been there for me throughout my academic journey. Thank you all for your support, encouragement, and motivation.

Abstract

Adaptive Cruise Control (ACC) revolutionizes automotive safety and comfort by automatically adjusting a vehicle's speed to maintain a set distance from the car ahead. This system integrates sensors, radar, and occasionally cameras to monitor real-time traffic conditions, enabling responsive vehicle behavior. What distinguishes ACC is its foundation on a fuzzy inference system (FIS), simulating human decision-making in navigating the complexities of road traffic. Among fuzzy inference techniques, the Mamdani Type-1 model stands out for its effectiveness in automating vehicle speed control, capturing nuanced driving behaviors with simplicity.Deploying a fuzzy inference system in ACC presents challenges. Fine-tuning FIS parameters is crucial for adaptability across diverse driving conditions, translating fuzzy logic into concrete equations demands precision, and integrating ACC with vehicle sensors requires technical sophistication. The evaluation phase, critical for verifying ACC's efficacy in maintaining safe distances, involves simulation-based analyses and real-world tests.ACC's impact extends beyond driver convenience, enhancing road safety by mitigating collision risks. As technology evolves, ACC promises to transform driving experiences and establish new paradigms in vehicle automation. Yet, perfecting ACC technology requires addressing ongoing challenges in FIS tuning, sensor integration, and reliability across diverse scenarios. The relentless pursuit of innovation in ACC development reflects the automotive sector's commitment to safer, more enjoyable driving experiences, heralding a new era of intelligent transportation systems.

Contents

List of Figures

List of Tables

Chapter 1

1 Introduction to the Project

1.1 Project Title

The title of the project is ”Adaptive Cruise Control For Vehicles”.

1.2 Project Description

The initiative to develop an ”Adaptive Cruise Control for Vehicles” system is at the forefront of enhancing driving safety and efficiency on highways. This system aims to automate the process of maintaining a safe following distance, managing the vehicle’s acceleration, engine performance, and braking based on a set time gap from the vehicle in front. The primary goal is to create a system that dynamically adjusts the cruising speed of a vehicle in response to the traffic flow within its lane, with a keen focus on the actions of the vehicles ahead to ensure maximum safety.

This project involves a detailed and comprehensive plan, starting with the creation of a fuzzy inference system (FIS) software. This advanced software acts as the decision-making core, capable of processing diverse sensor inputs to mimic intelligent driving decisions. The project’s scope extends to integrating this software with a variety of sensors, providing the vehicle with an in-depth perception of its surroundings. This integration is crucial for the system to accurately assess traffic conditions and make informed adjustments to the vehicle’s speed and spacing.

Following the development of the software, the next phase involves embedding the FIS within the vehicle’s existing control systems. This integration is essential for achieving a seamless interaction between the vehicle’s software and hardware, ensuring that they operate as a unified system. To evaluate and refine this integration, the project utilizes hardware-in-the-loop simulations. These simulations replicate real-world driving scenarios, offering a platform to test the system’s responses and make necessary refinements to enhance performance.

The culmination of the project focuses on rigorous testing and fine-tuning to ensure the system's readiness to handle a wide range of traffic situations. This stage is critical for verifying that the adaptive cruise control system adheres to the highest standards of safety and efficiency.

Ultimately, this project is not just about improving vehicle control; it's about advancing automotive technology to provide drivers with safer, more responsive driving experiences. Through the use of sophisticated software, comprehensive sensor technology, and thorough testing processes, the project seeks to establish a new benchmark in vehicle automation and safety, significantly reducing the reliance on human intervention in driving and setting the stage for future advancements in autonomous vehicle technologies.

1.3 Scope of the Project

This initiative is centered around the development of an advanced hardware controller that incorporates a Fuzzy Inference System (FIS) to enhance the functionality and efficiency of Adaptive Cruise Control (ACC) systems in modern vehicles. The essence of this technology is to refine how vehicles autonomously adjust their speed and maintain safe distances from other vehicles under varying traffic conditions, emulating the intricate decision-making process of a human driver. By leveraging fuzzy logic, this controller is designed to process sensor data with an allowance for ambiguity, much like human judgment, which enables more fluid transitions and a driving experience that feels more natural.

The integration of a Fuzzy Inference System into the ACC framework marks a significant departure from traditional binary logic systems. Instead of relying on rigid, on-or-off decision-making processes, the FIS introduces a spectrum of 'truths' or possibilities that reflect the real-world complexity of driving scenarios. This approach allows the ACC to make more nuanced decisions, considering a broader range of factors and conditions that drivers face on the roads daily.

By adopting this sophisticated approach, the project aims to equip vehicles with an ACC system that not only responds more adeptly to the dynamic conditions of the driving environment but also offers a more intuitive interface for the user. This means that the system will not just react to the speed and distance of the vehicle ahead; it will also anticipate and adjust to changes in traffic flow and other environmental factors with a level of precision and foresight that closely mirrors human intuition.

The ultimate goal of incorporating an FIS into the ACC system is to significantly elevate the overall safety and comfort of road users. This involves a comprehensive development and testing phase, where the hardware controller undergoes rigorous evaluation under a wide range of driving conditions to ensure its reliability and effectiveness. Through this meticulous process, the project seeks to deliver an ACC system that seamlessly integrates with the vehicle's existing controls, providing a smoother, more responsive driving experience.

This endeavor not only represents a leap forward in automotive safety technology but also sets the stage for further innovations in vehicle automation. By making ACC systems smarter and more adaptable, the project contributes to the broader vision of creating vehicles that can navigate the complexities of the road with minimal human input, thereby enhancing safety and efficiency on our roads. The successful implementation of this technology promises to redefine the standards of automated driving, making it safer and more enjoyable for everyone.

1.4 Summary

The chapter introduces the project titled ”Adaptive Cruise Control For Vehicles,” highlighting its objective of enhancing driving safety and efficiency on highways. It aims to automate maintaining a safe following distance and managing vehicle acceleration, engine performance, and braking based on traffic conditions. The project involves developing a fuzzy inference system (FIS) software and integrating it with various sensors to perceive surroundings accurately. Hardware-in-the-loop simulations are utilized to test system responses and refine integration. The project focuses on rigorous testing to ensure safety and efficiency, aiming to establish a new benchmark in vehicle automation. The scope involves developing an advanced hardware controller with a Fuzzy Inference System (FIS) to enhance Adaptive Cruise Control (ACC) systems in vehicles. The integration of FIS aims to enable more fluid transitions and natural driving experiences by processing sensor data with ambiguity allowance. The project aims to equip vehicles with an ACC system that responds adeptly to dynamic driving conditions and enhances overall safety and comfort. Through comprehensive development and testing, it seeks to deliver an ACC system that seamlessly integrates with existing vehicle controls, making driving safer and more enjoyable. Ultimately, the project contributes to creating vehicles that navigate roads with minimal human input, redefining the standards of automated driving for enhanced safety and efficiency.

Chapter 2

2 Literature Review

2.1 Levels of Autonomy of ACC

The development of Adaptive Cruise Control (ACC) is a key indicator of the progress being made towards fully autonomous vehicles. Initially introduced as a Level 1 technology, ACC provided drivers with a basic form of assistance by maintaining a steady speed without manual input, marking a significant improvement over traditional cruise control systems. This initial stage offered a glimpse into the potential for technology to ease the task of driving, albeit in a limited capacity.

As the technology advanced to Level 2, ACC's capabilities were expanded to include the adjustment of the vehicle's speed in response to the surrounding traffic conditions in real-time. This development was a boon for drivers, particularly during lengthy drives, as it reduced the need for constant manual adjustments to the vehicle's speed, thereby diminishing driver fatigue and enhancing overall driving comfort.

Moving into Level 3 autonomy, ACC underwent further enhancement by acquiring the ability to perform more complex maneuvers, such as executing lane changes. This level of autonomy required the driver to remain alert and ready to take control of the vehicle if necessary, indicating a transition phase where the vehicle and the driver share responsibilities for the task of driving.

At Level 4, the scope of ACC expanded significantly, incorporating it into a system that is capable of performing the vast majority of driving tasks autonomously within specific conditions or environments. Despite this high level of automation, the system still provided the option for manual override, ensuring that drivers could take control of the vehicle when desired or when the situation necessitated it.

The pinnacle of this evolutionary ladder, Level 5, represents the ultimate goal of autonomous driving technology, where ACC matures into a fully autonomous system. At this stage, no human intervention is required at any point during the driving process, allowing the vehicle to handle all aspects of driving across any scenario. This level of autonomy promises a future where vehicles can navigate safely and efficiently without human input, potentially revolutionizing road safety and traffic management.

Each step in the progression from Level 1 to Level 5 autonomy signifies a milestone in the journey towards achieving vehicles that can independently navigate the complexities of the road. With ACC serving as a foundational technology in this evolution, its development not only showcases the incremental advancements in vehicle automation but also underscores the potential for these technologies to transform the driving experience, enhance safety, and optimize traffic flow. As we move closer to realizing the vision of fully autonomous vehicles, ACC continues to play a crucial role in shaping the future of transportation, marking each level of autonomy as a leap forward in our quest for safer, more efficient driving.

2.2 Fuzzy Inference System

Fuzzy Inference Systems (FIS) represent a groundbreaking convergence of quasi-human reasoning capabilities with sophisticated computational procedures, adept at navigating the complexities and indeterminacies that traditional binary decision-making systems often find insurmountable. Anchored in the principles of fuzzy logic, FIS transcends the rigid dichotomies of true or false, instead embracing a gradation of potentialities that mirrors the nuanced decision-making process inherent to human cognition when faced with uncertain or ambiguous situations.

The architecture of an FIS is built upon several key components: a comprehensive set of fuzzy rules that guide its operation, a detailed database that informs these rules, and a dynamic reasoning engine tasked with the application of these rules to derive meaningful outputs. This process initiates with fuzzification, a crucial step where definite inputs are transformed into fuzzy values that embody a range of potential interpretations rather than a singular, fixed point. Following this, these fuzzy values are meticulously processed through an intricate network of rules in a stage known as inference, culminating in the generation of fuzzy outputs. The concluding phase, defuzzification, involves the careful translation of these fuzzy outputs into clear, actionable insights.

The applicability of FIS extends across a diverse array of fields, demonstrating its versatility and effectiveness. In consumer electronics, it enhances the functionality of devices, enabling cameras to focus with greater accuracy and washing machines to tailor their cycles to the specifics of the load. In the volatile arena of financial markets, FIS provides valuable insights, helping analysts forecast stock trends with a higher degree of precision. The healthcare sector benefits from its diagnostic capabilities, where it contributes to more accurate assessments of medical conditions. Moreover, in the realm of smart home technologies, FIS plays a critical role in fine-tuning temperature regulation, ensuring optimal comfort and efficiency. Within the automotive industry, FIS significantly advances safety and performance features, especially in the development of sophisticated systems like Adaptive Cruise Control, which dynamically adjusts a vehicle's speed in response to varying traffic conditions.

The intrinsic value of FIS lies in its ability to approximate human-like reasoning within environments replete with uncertainty and variability. This capacity positions FIS as a pioneering step toward developing a more instinctive form of artificial intelligence, one that empowers machines and systems to interpret and interact with the world in a manner that closely resembles human thought processes and judgments. By leveraging the flexibility and depth of understanding offered by FIS, technology is moving towards a future where artificial intelligence can navigate the complexities of the real world with an unprecedented level of sophistication and nuance, bridging the gap between computational exactitude and the fluidity of human cognition.

2.3 Mamdani Fuzzy Inference System:

The Mamdani Fuzzy Inference System is a cornerstone in the realm of fuzzy logic, celebrated for its intuitive method of emulating the nuances of human decision-making processes. Introduced by Ebrahim Mamdani in the 1970s, it is distinguished by its utilization of straightforward rules that effectively bridge fuzzy inputs to their corresponding outputs. These rules, formulated based on expert insights, are articulated in natural language, enhancing the Mamdani system's accessibility and ease of use for practitioners. At the heart of the Mamdani method is the parallel evaluation of these linguistic rules, which collectively contribute to a comprehensive output representation. This representation, although initially fuzzy, undergoes a defuzzification process, transforming it into a specific, actionable value.

The Mamdani system's appeal lies in its uncomplicated structure and the clarity with which it presents information, making it an ideal choice for application in control systems where the replication of human-like reasoning is paramount. It adeptly manages multiple variables concurrently, sidestepping the complexities often associated with advanced mathematical models. This characteristic makes it highly suited to a variety of practical scenarios, from automation to decision support, where the direct interpretation and application of expert knowledge are crucial.

Despite the existence of alternative fuzzy inference systems that might surpass the Mamdani model in terms of computational speed or efficiency, the Mamdani approach is frequently preferred for its inherent simplicity and the natural manner in which it mirrors human thought processes. Its design is such that it democratizes the use of fuzzy logic, allowing users to implement sophisticated decision-making frameworks without delving into the complexities of the underlying mathematics. This balance between simplicity and effectiveness ensures that the Mamdani system remains a valuable tool for designing solutions that are both practical and closely aligned with the intuitive ways humans approach problem-solving.

Moreover, the Mamdani Fuzzy Inference System's enduring relevance is underscored by its widespread adoption across various sectors, underscoring its versatility and the trust placed in its methodology by professionals seeking to incorporate fuzzy logic into their projects. Its capacity to distill expert knowledge into a set of manageable, interpretable rules without losing the depth of that expertise is a testament to its design philosophy. As such, the Mamdani model not only facilitates the creation of solutions that are sensitive to the nuances of real-world applications but also champions an approach to artificial intelligence that values simplicity, clarity, and the replication of human insight.

2.4 Takagi-Sugeno-Kang (TSK) Fuzzy Inference System

The Takagi-Sugeno-Kang (TSK) Fuzzy Inference System, commonly referred to as the Sugeno model, represents a refined methodology in the realm of fuzzy logic, merging the precision of crisp logic with the nuanced handling of fuzzy inputs. This innovative system was initially conceptualized by Takagi and Sugeno, with significant enhancements later introduced by Kang. It stands as a distinct alternative to the traditional Mamdani model, renowned for its unique approach to generating outputs. Unlike the Mamdani system, which relies on fuzzy sets for its output, the Sugeno model employs a more direct method, producing outputs through specific mathematical functions, often linear combinations of the input variables or fixed constants.

The Sugeno model excels in computational efficiency, making it exceptionally well-suited for applications requiring rapid decision-making in real-time environments. Its streamlined rule structure allows for a succinct representation of logic, and by directly computing the output as a function of input variables, it eliminates the need for the defuzzification step that is essential in Mamdani systems. This omission significantly accelerates the processing speed, enhancing the system's suitability for time-sensitive tasks.

A standout feature of the Sugeno methodology is its inherent adaptability and seamless integration with optimization and adaptive techniques. This flexibility renders it invaluable in the realm of control systems and complex modeling, where the ability to dynamically learn and adjust to changing conditions is crucial. The precision and adaptability of the Sugeno model have garnered it widespread adoption across a diverse array of fields, including automotive engineering, robotics, and even the intricate analysis required in financial and economic forecasting.

The TSK system's robustness is not merely in its computational elegance but also in its practical applicability. It begins with foundational observations and expert knowledge, progressively evolving to refine its inference mechanisms through continuous learning. This capacity for adaptation allows it to fine-tune its analytical models, enhancing its effectiveness and accuracy in interpreting and responding to the complexities of its application environment.

The dynamic and intelligent capabilities of the TSK Fuzzy Inference System position it as a cornerstone in the development of sophisticated, autonomous control systems. It embodies a bridge between theoretical concepts and real-world applications, facilitating advancements in automation and data-driven technologies. By leveraging the TSK system, engineers and researchers can craft solutions that not only navigate but also anticipate the challenges of increasingly complex and automated environments, underscoring the system's pivotal role in pushing the boundaries of what's possible in intelligent system design.

2.5 Tsukamoto Fuzzy Inference System

The Tsukamoto Fuzzy Inference System distinguishes itself within the realm of fuzzy logic through its unique method of generating crisp outputs from fuzzy inputs. What sets the Tsukamoto model apart is its reliance on monotonic membership functions, ensuring that the output is directly influenced by the degree of truth of the inputs. This characteristic makes the Tsukamoto system particularly effective in scenarios where the relationship between variables is well understood and can be represented as either increasing or decreasing functions.

Unlike other fuzzy inference systems that might employ complex defuzzification strategies or linear output functions, the Tsukamoto approach simplifies the process by directly linking the input's fuzzy degree to a precise output value. This direct linkage allows for an intuitive understanding of how inputs affect the system's outputs, facilitating easier adjustments and fine-tuning by system designers.

The Tsukamoto model is especially valued in applications requiring clear, straightforward decision-making processes based on fuzzy logic. Its implementation can be seen in various fields, from control systems that demand precise responses based on fuzzy criteria to decision-making tools where the clarity of the outcome is paramount. The simplicity and directness of the Tsukamoto system offer a balance between the interpretability of fuzzy logic and the necessity for actionable outcomes, embodying a practical approach to handling uncertainty in decision-making processes.

2.6 Neuro-Fuzzy Inference System

The Neuro-Fuzzy Inference System represents a fascinating intersection of neural networks and fuzzy logic, merging the best of both worlds to create a powerful tool for decision-making and prediction. This hybrid approach capitalizes on the adaptive learning capabilities of neural networks and the nuanced reasoning of fuzzy logic. Essentially, it uses a neural network structure to learn the parameters of the fuzzy inference system, enabling it to refine and adjust its rules and membership functions based on incoming data. This learning process allows the system to improve its performance over time, adapting to new patterns or changes in the environment.

What makes the neuro-fuzzy system particularly appealing is its ability to handle complex, nonlinear problems where traditional analytical methods might falter. By combining the interpretability and simplicity of fuzzy systems with the learning strength of neural networks, it offers a robust framework for modeling and control in uncertain environments. Applications range from financial forecasting and medical diagnosis to advanced control systems in robotics and automotive technology, showcasing its versatility. The neuro-fuzzy system stands out for its user-friendly nature, allowing experts with limited knowledge of neural networks to still harness advanced computational techniques. This blend of accessibility and sophistication makes it a valuable asset in the toolkit of engineers and scientists, enabling more intelligent, adaptive, and effective solutions to a myriad of real-world problems.

2.7 Adaptive Neuro-Fuzzy Inference System

The Adaptive Neuro-Fuzzy Inference System (ANFIS) represents a cutting-edge paradigm that amalgamates the intuitive logic of fuzzy systems with the dynamic learning capabilities inherent in neural networks. This fusion of methodologies positions ANFIS as an exceptional framework capable of approximating nonlinear functions and effectively managing imprecise information, thereby presenting itself as a formidable tool for addressing complex problem-solving scenarios across diverse domains. Fundamentally, ANFIS thrives on its adaptability, fine-tuning the parameters of the fuzzy system based on input-output data to facilitate continual learning and refinement, thereby improving its accuracy over time. This inherent adaptability serves as a cornerstone for ANFIS's success in tasks necessitating nuanced comprehension and adaptability, such as pattern recognition, predictive analytics, and system control.

A distinguishing aspect of ANFIS lies in its adeptness at modeling intricate relationships between input and output variables with exceptional precision, thanks to its structured learning process. This structured learning not only augments the system's performance but also provides invaluable insights into the model's operational mechanisms, furnishing users with a transparent view into the decision-making processes. Whether it pertains to forecasting stock market trends, optimizing industrial processes, or orchestrating the navigation of autonomous vehicles, ANFIS has unequivocally demonstrated its efficacy by furnishing solutions that are both sophisticated and pragmatic. Its widespread application has engendered significant advancements in fields where conventional models may falter in encapsulating complexity or adapting to novel information, thereby underscoring the potency of merging fuzzy logic with neural learning methodologies.

The versatility and robustness of ANFIS further manifest in its capacity to seamlessly integrate with existing frameworks and methodologies, fostering synergy rather than competition. This interoperability extends ANFIS's utility beyond standalone applications, enabling it to enhance the capabilities of existing systems and algorithms across diverse fields. By leveraging its adaptive prowess and the synergistic combination of fuzzy logic and neural networks, ANFIS continues to pave the way for groundbreaking innovations and advancements across a myriad of disciplines.

Furthermore, ANFIS's innate ability to evolve and adapt in response to changing environmental dynamics underscores its relevance and applicability in an ever-evolving landscape. In domains characterized by volatility, uncertainty, complexity, and ambiguity (VUCA), ANFIS serves as a reliable ally, offering insights and solutions that navigate the intricacies of real-world scenarios with finesse and efficacy. Its ability to learn from past experiences, anticipate future trends, and adapt in real-time positions it as a quintessential tool for decision-makers seeking to navigate the complexities of the modern world.

In essence, the Adaptive Neuro-Fuzzy Inference System epitomizes a paradigm shift in computational intelligence, harnessing the synergistic potential of fuzzy logic and neural networks to address complex problems with unparalleled precision and adaptability. As it continues to evolve and mature, ANFIS stands poised to revolutionize diverse fields, offering innovative solutions to the most pressing challenges of our time.

2.8 Summary

The chapter on ”Literature Review” explores the evolution and application of Adaptive Cruise Control (ACC) systems and Fuzzy Inference Systems (FIS) in the automotive industry.Initially introduced as a Level 1 technology, ACC provided basic assistance by maintaining steady speed without manual input. As technology progressed to Level 2, ACC's capabilities expanded to adjust vehicle speed in response to surrounding traffic conditions. Subsequent levels of autonomy, up to Level 5, aim for fully autonomous driving.

FIS, rooted in fuzzy logic, offers a nuanced approach to decision-making, allowing machines to interpret and interact with the world more like humans. The Mamdani FIS uses straightforward rules based on expert insights, while the Sugeno model employs precise mathematical functions for output generation. The Tsukamoto FIS relies on monotonic membership functions for clear decision-making.The Neuro-Fuzzy Inference System combines neural networks and fuzzy logic, offering adaptability and effectiveness in handling complex problems. The Adaptive Neuro-Fuzzy Inference System (ANFIS) further refines this approach by fine-tuning fuzzy system parameters based on input-output data, facilitating continual learning and adaptation.

ANFIS's versatility and adaptability make it a valuable tool for addressing complex problem-solving scenarios across diverse domains, from forecasting stock market trends to optimizing industrial processes. Its ability to evolve and adapt in real-time positions it as a quintessential tool for decision-makers navigating the complexities of the modern world.In summary, the chapter provides a comprehensive overview of ACC and FIS technologies, highlighting their evolution, applications, and potential for revolutionizing various industries.

Chapter 3

3 Preliminaries

3.1 Comparative Analysis

Exploring the diverse universe of fuzzy inference systems (FIS) and the adaptive neuro-fuzzy inference system (ANFIS) offers a fascinating glimpse into how these models tackle the challenges of uncertainty, complexity, and non-linearity in decision-making processes. Each system, from Mamdani and Takagi-Sugeno-Kang (TSK) to Tsukamoto, Neuro-Fuzzy, and ANFIS, brings its unique strengths and suitability to various applications, highlighting the rich tapestry of methodologies in computational intelligence.

The Mamdani FIS, known for its simplicity and intuitive design, mirrors the nuanced way humans make decisions under uncertainty. It's particularly appreciated for its use of linguistic rules, making it accessible and easily interpretable by experts across different fields. However, its reliance on extensive computational resources, especially during the defuzzification stage, poses a challenge for real-time applications, marking a trade-off between user-friendliness and efficiency.

On the other hand, the TSK system stands out for its mathematical precision, producing outputs as functions of inputs, often linear, which streamlines both computation and application in control systems. This makes it a go-to choice for engineers needing fast, accurate responses. Yet, this mathematical rigor can sometimes obscure the intuitive understanding of the decision process, a gap that might not suit every application's needs.

The Tsukamoto model offers an interesting blend, maintaining a direct link between input and output through monotonic membership functions. This ensures a more predictable outcome, albeit still demanding in terms of computation and somewhat limited by the necessity for monotonicity in its output functions, restricting its application scope.

Emerging as a blend of fuzzy logic's human-like reasoning and neural networks' learning prowess, the Neuro-Fuzzy system captures the best of both worlds. It excels in adapting to new data, fine-tuning its rules and functions for enhanced accuracy. However, its sophistication comes with complexity and the need for substantial data for training, which might not always be feasible.

ANFIS takes this integration further, combining the structured approach of TSK FIS with neural networks' adaptability. It shines in environments where relationships between variables are complex and evolving, offering precise modeling capabilities. Nevertheless, the intricate process of setting up and training an ANFIS model, not to mention the computational demands, can be daunting.

Choosing between these systems boils down to balancing interpretability, computational efficiency, and adaptability against the specific needs of the task. For projects where clarity and ease of understanding are paramount, Mamdani's approach is often preferred. When speed and accuracy top the list of requirements, TSK is the likely candidate. Tsukamoto finds its niche in situations where a clear, monotonic relationship can be established. Meanwhile, Neuro-Fuzzy and ANFIS are reserved for contexts demanding high adaptability and learning from complex data sets.

In essence, this comparative journey through fuzzy inference systems underscores the importance of aligning a system's capabilities with the project's goals. As technology evolves, so too does the potential of these systems to address more sophisticated challenges, promising an exciting future for artificial intelligence and machine learning. The balance each system strikes between ease of use, speed, and adaptability makes them invaluable tools in the quest to navigate the uncertainties of the real world.

3.2 Simulink model

The figure [1] presents a comparative analysis of different control systems using Simulink models, a simulation tool for dynamic systems. On the left, we see three separate control configurations: a conventional Proportional-Integral-Derivative (PID) controller, a Type-1 Fuzzy PID controller, and a Type-2 Fuzzy PID controller, each receiving the same reference signal (r) and outputting a control signal (u) to the process (G). The conventional PID controller is the baseline, using fixed gain values to process the error between the reference and the process output (y). The integral absolute error (iae) block measures the performance of the control system. The Type-1 Fuzzy PID introduces fuzzy logic to handle uncertainties and nonlinearities in the process. It modifies the control signal based on fuzzy rules derived from the error and change in error. This should, theoretically, result in a more robust control system compared to the conventional PID, as indicated by the iae block.

Type-2 Fuzzy PID goes further, using a more sophisticated fuzzy system that can handle higher levels of uncertainty. It has the potential to outperform both the conventional and Type-1 Fuzzy PID controllers, especially in systems with significant noise or variability, as suggested by its dedicated iae block.

On the right, detailed blocks for each controller type show how the error (e) and change in error (e) are processed to produce the control signal (u). The presence of the fuzzy inference system block in the Type-1 and Type-2 Fuzzy PID models highlights the additional complexity and adaptability introduced by incorporating fuzzy logic into PID control. This comparative setup is designed to evaluate the efficacy of each control method in terms of precision, adaptability to changing conditions, and overall system performance.

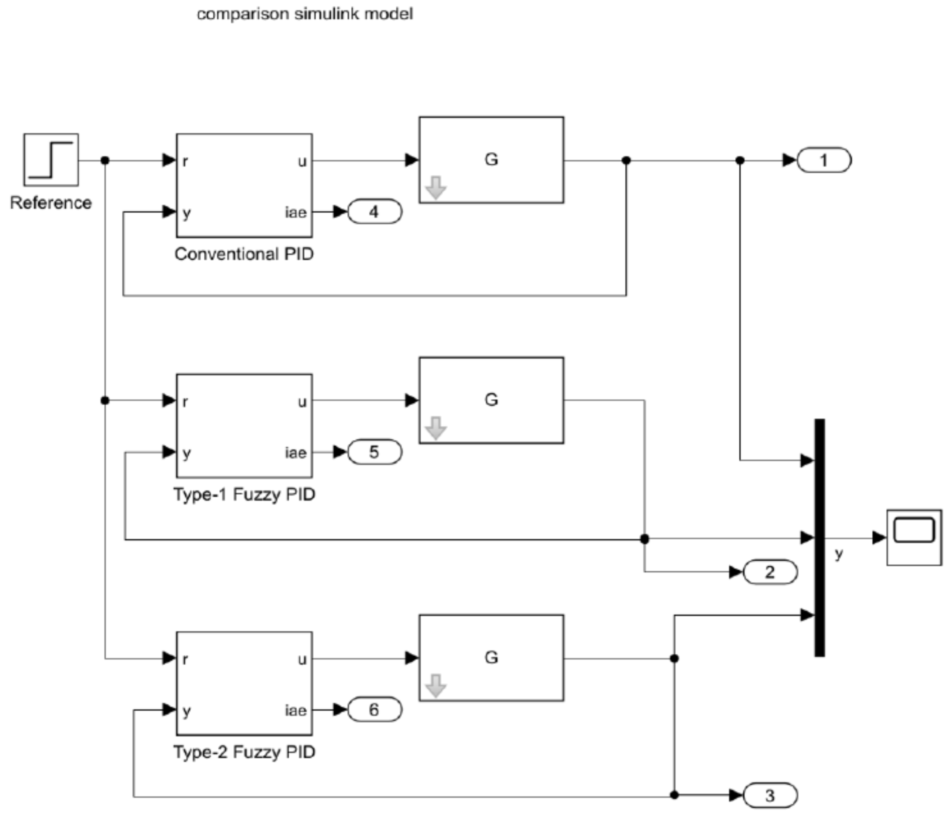


Figure 1: Comparative Simulink Model

3.3 Results

The graph in figure [2] illustrates the performance of three different control systems: a conventional PID, a Type-1 Fuzzy Logic Controller (FLC), and a Type-2 Fuzzy Logic Controller, in response to a given reference signal over time. The output is measured against time (in seconds), with the goal for each system to match the reference signal as closely and as quickly as possible.

The Reference line represents the desired output level. The PID controller's response is the first to reach the vicinity of the reference but shows some initial overshoot, where the output exceeds the desired level before stabilizing. The Type-1 FLC demonstrates a smoother approach towards the reference without significant overshoot, indicating better control of the rise without surpassing the target excessively. Meanwhile, the

Type-2 FLC shows a response similar to the Type-1 but with a slightly higher overshoot, suggesting a marginally more aggressive approach than the Type-1 FLC but less so than the conventional PID.

All systems eventually stabilize close to the reference signal, but the key differences lie in their transient behaviors: how quickly and accurately they reach and maintain the desired output level. The Type-1 and Type-2 FLCs appear to provide a more damped response compared to the conventional PID, which could translate to more stable control in practical applications, albeit with a trade-off in responsiveness as indicated by the overshoot and settling time.

The table provides performance metrics for three control systems: a conventional PID controller and two types of Fuzzy Logic Controllers (FLC), Type-1 and Type-2. The metrics compared are Rise Time, Overshoot, Settling Time, and Absolute Error.

3.4 Rise Time

This is the time taken for the system output to rise from 0 to 100 percent of the reference value for the first time. The PID controller has the fastest rise time of 0.62412 seconds, indicating it reaches the desired output level quicker than both FLCs. The Type-1 FLC has a rise time of 1.4267 seconds, and the Type-2 FLC is slower at 1.8662 seconds.

3.5 Overshoot

This measures how much the output exceeds the reference value before stabilizing. The PID controller shows an overshoot of 11.234 percent, indicating a significant exceedance beyond the target level. In contrast, both FLCs exhibit no overshoot, suggesting a more controlled approach to reaching the target output without surpassing it.

3.6 Settling Time

This metric indicates how long it takes for the output to stabilize within a certain percentage of the reference value after an initial disturbance. The PID controller settles in 4.5583 seconds, while the Type-1 FLC settles slightly faster at 4.1023 seconds. The Type-2 FLC takes the longest to settle at 5.129 seconds.

3.7 Absolute

This is a measure of the overall deviation of the output from the reference value over time. The PID controller has the lowest error at 1.04, suggesting that over time, its output is closest to the reference. The Type-1 and Type-2 FLCs have higher absolute errors of 1.522 and 1.282, respectively, indicating a larger overall deviation from the desired output over time.

These values collectively provide insights into the dynamic performance and accuracy of each controller, with the conventional PID offering a quick response but less stability, and the FLCs providing a more balanced control without overshoot, albeit at the expense of a slower response and slightly less accuracy as reflected in the absolute error.

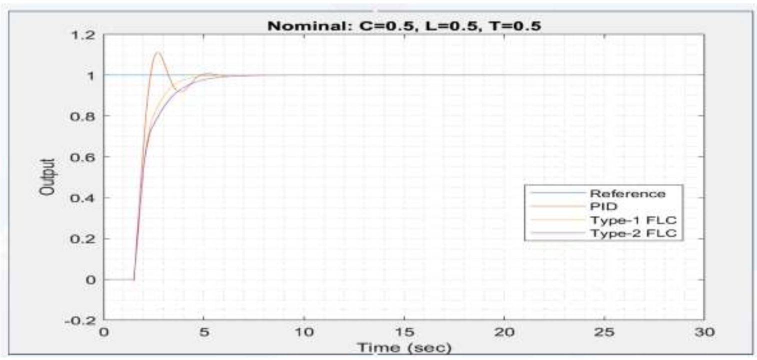


Figure 2: Resulting plot

Table 1: Performance Metrics of Control Systems

Controller	Rise Time	Overshoot	Settling Time	Absolute Error
PID	0.62412	11.234	4.5583	1.04
Type-1 FLC	1.4267	0	4.1023	1.5222
Type-2 FLC	1.8662	0	5.129	1.282

3.8 Summary

The ”Preliminaries” chapter provides a comprehensive comparative analysis of various fuzzy inference systems (FIS) and the adaptive neuro-fuzzy inference system (ANFIS), shedding light on their unique strengths and suitability for different applications.The Mamdani FIS, known for its simplicity and intuitive design, mirrors human decision-making under uncertainty, making it accessible and easily interpretable. However, its computational demands pose challenges for real-time applications.The Takagi-Sugeno-Kang (TSK) system stands out for its mathematical precision, producing outputs as functions of inputs, which streamline computation and application in control systems.

The Tsukamoto model offers a blend of direct input-output linkage through monotonic membership functions, ensuring a more predictable outcome.The Neuro-Fuzzy system combines fuzzy logic's reasoning with neural networks' learning capabilities, excelling in adapting to new data but requiring substantial data for training.ANFIS integrates the structured approach of TSK FIS with neural networks' adaptability, offering precise modeling capabilities but requiring intricate setup and training.The chapter includes a comparative analysis using Simulink models to evaluate different control systems' performance. It presents graphs illustrating the systems' responses to a reference signal over time, along with performance metrics such as rise time, overshoot, settling time, and absolute error.The results highlight the trade-offs between computational efficiency, interpretability, and adaptability in choosing the appropriate control system for specific applications. Ultimately, the chapter underscores the importance of aligning system capabilities with project goals to address complex challenges effectively in artificial intelligence and machine learning.

Chapter 4

4 Methodology

4.1 Project Approach

The methodology depicted in Figure 3 presents a comprehensive and systematic approach to developing a system utilizing a Fuzzy Inference System (FIS), beginning with an extensive ”Literature Review.” This foundational stage involves a thorough examination of existing scholarly work, including theories, existing systems, and various methodologies related to FIS. The literature review is critical for acquiring a deep understanding of the field’s current knowledge landscape, pinpointing established best practices, and identifying any existing gaps or opportunities for innovation that the project might leverage. This step ensures that the project is grounded in solid theoretical underpinnings and is informed by the latest research and developments in the field.

Following the literature review, the project progresses to the ”FIS Software Implementation” phase. Here, the insights and theoretical frameworks gleaned from the literature are translated into a tangible software solution. During this phase, the FIS software is meticulously developed and programmed, with its architecture and functionality being heavily influenced by the findings from the preceding research. This stage is pivotal, as it transforms theoretical concepts into a working software model that embodies the principles of fuzzy logic and inference.

The subsequent phase, ”Sensor Data Acquisition,” involves the systematic collection of data from a variety of sensors. These sensors are tasked with gathering the real-world inputs necessary for the FIS to operate. The accuracy and reliability of this data acquisition step cannot be overstated, given that the efficacy and precision of the FIS heavily depend on the quality of the input data it receives. This phase ensures that the system is furnished with relevant and high-quality data to inform its processing and decision-making capabilities.

Following data collection, the project moves into the ”FIS Hardware Implementation” phase. This crucial stage sees the integration of the developed FIS software with physical hardware components. It involves the practical application and embedding of the FIS into suitable hardware environments, which may range from electronic control units to various other electronic devices. This step is essential for transitioning the FIS from a software-only entity into a fully operational system capable of interacting with the physical world.

The ”Hardware In Loop Simulation” phase follows, wherein the now integrated FIS system undergoes rigorous testing within a simulated environment designed to replicate real-world operational conditions. This simulation is indispensable for assessing the system’s performance, identifying any issues, and making necessary adjustments. It serves as a critical validation step, ensuring the FIS’s functionality and reliability before its actual deployment.

The final phase involves ”Tuning and Adjustments,” a meticulous process of refining and optimizing the system based on feedback and results obtained from the simulation phase. This iterative process is aimed at fine-tuning the system’s parameters to enhance its performance, ensuring it operates efficiently and effectively in a variety of conditions.

This structured project approach is iterative and comprehensive, designed to ensure that each phase of development builds upon the insights and advancements of the preceding stages. By methodically navigating through the stages of theory exploration, software and hardware development, data acquisition, system integration, and exhaustive testing, the methodology aims to develop a robust, efficient, and reliable FIS. This system is expected to perform optimally in its designated application, supported by a foundation of rigorous research, development, and testing protocols.

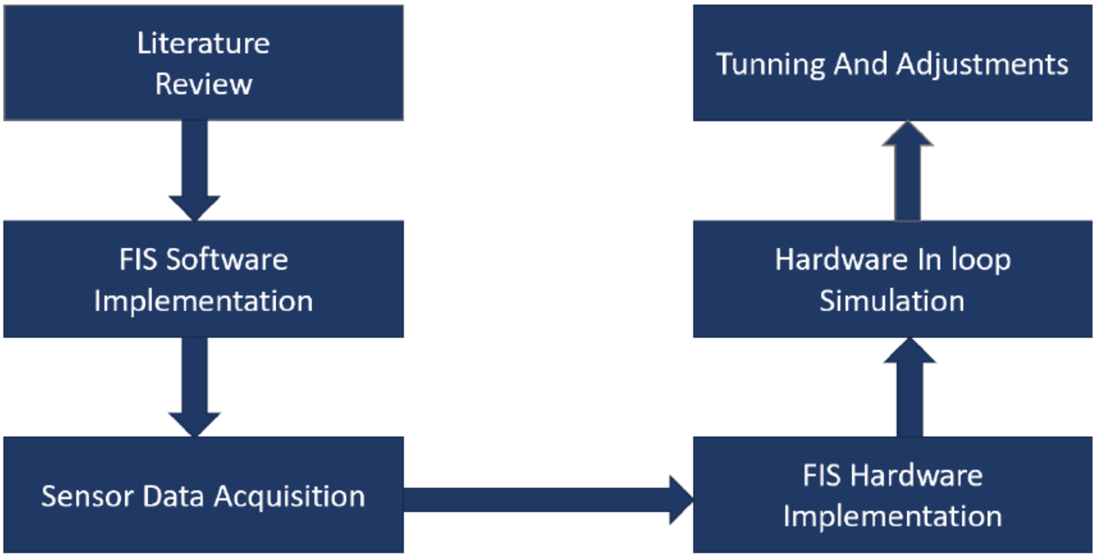


Figure 3: Methodology

4.2 Step 1: FIS Software Implementation

The figure [2] shows fuzzy logic controller interface, from MATLAB's Fuzzy Logic Designer, which is used for designing, testing, and analyzing fuzzy inference systems. The left side of the figure shows the input membership functions for two variables: "Relative speed" and "Headway distance". The right side of the figure presents a 3D surface plot that graphically represents the relationship between the input variables and the output variable "Acceleration".

The process depicted in the figure involves several mathematical steps:

Fuzzification:

The crisp inputs for "Relative speed" and "Headway distance" are converted into fuzzy values by evaluating them against the input membership functions. These membership functions define the degree to which the inputs belong to each of the fuzzy sets. The horizontal lines in the input value graphs likely represent the current input values being evaluated. Rule Evaluation: Fuzzy rules are applied to the fuzzified inputs. These rules are typically in the form of IF-THEN statements that describe the output response for given input conditions. The "AND (min)" notation indicates that the rules use the AND

operator to combine the fuzzy values of the inputs, taking the minimum value of the fuzzy degrees as the result of the rule's antecedent. Aggregation:
The results of all the fuzzy rules are then combined to form a single fuzzy set for the output. This is done by taking the maximum value (union) of all the fuzzy rule outputs.
Defuzzification:
The aggregated fuzzy output is then converted back into a crisp value. This is done using a defuzzification method, such as the centroid method, which calculates the center of area under the curve. The output of this process is the crisp value of "Acceleration" that the system should take given the current inputs. The 3D surface plot visualizes how the "Acceleration" output is expected to change based on the "Relative speed" and "Headway distance". This surface is derived from the fuzzy rules and represents the controller's logic. The plot helps in understanding the relationship between inputs and output and in tuning the fuzzy system for desired performance. For a specific "Relative speed" and "Headway distance", the corresponding "Acceleration" can be read from the surface plot. This acceleration is what the fuzzy logic controller would command, for instance, in an adaptive cruise control system in a vehicle, to maintain a safe and comfortable distance from the car in front, adjusting the vehicle's speed accordingly.
Suppose the membership function equations for the fuzzy set for temprature system:

$$\mu_{\text{Warm}}(T) = \begin{cases} 0 & \text{for } T \leq 20°C \text{ or } T \geq 30°C, \\ \frac{T-20}{25-20} & \text{for } 20°C \leq T \leq 25°C, \\ \frac{30-T}{30-25} & \text{for } 25°C \leq T \leq 30°C. \end{cases}$$

$\mu_{\text{Warm}}(23) = \frac{23-20}{25-20} = \frac{3}{5} = 0.6$

Rule R_i : If x_1 is A_i and x_2 is B_i then $y_i = f_i(x_1, x_2)$

Output y = $\sum_{i=1}^{n} w_i \cdot y_{i \frac{}{\sum_{i=1}^{n} w_i}}$
Here are the rules expressed in table [2]:

Condition 1	Condition 2	Action
2*FrontDistance == VeryClose	delFrontDistance == Closing	Acceleration = Brake
	delFrontDistance == Stable	Acceleration = Zero
1*FrontDistance == Close	delFrontDistance == Stable	Acceleration = MediumBrake
1*FrontDistance == Medium	delFrontDistance == Stable	Acceleration = SlowAccelerate
2*FrontDistance == Far	delFrontDistance == Opening	Acceleration = Accelerate
	delFrontDistance == Stable	Acceleration = Accelerate
3*FrontDistance == VeryClose	CurrentSpeed == Slow	Acceleration = Zero
	CurrentSpeed == Medium	Acceleration = MediumBrake
	CurrentSpeed == Fast	Acceleration = Brake

Table 2: Rules Table

If the front distance is very close and the change in front distance is closing rapidly, then apply the brakes. If the front distance is very close and the change in front distance is stable, then maintain zero acceleration. If the front distance is close and the change in front distance is stable, then apply medium braking. If the front distance is medium and the change in front distance is stable, then accelerate slowly. If the front distance is far and the change in front distance is opening, then accelerate. If the front distance is far and the change in front distance is stable, then maintain acceleration. If the front distance is very close and the current speed is slow, then maintain zero acceleration. If the front distance is very close and the current speed is medium, then apply medium braking. If the front distance is very close and the current speed is fast, then apply the brakes. These rules define the behavior of the acceleration based on the current front distance, the change in front distance, and the current speed of the vehicle.

The figure appears to illustrate the logic and resulting control surface for an Adaptive Cruise Control (ACC) system based on fuzzy logic rules. On the left, we see membership functions for three inputs: ”FrontDistance,” ”delFrontDistance” (change in front distance), and ”CurrentSpeed.” On the right is the 3D control surface showing how the system determines the appropriate ”Acceleration” output based on combinations of input variables.

Here’s how the given fuzzy rules translate into the ACC’s response:
The rules governing the acceleration of a vehicle based on its front distance, change in

front distance, and current speed are designed to ensure safe and efficient driving. When the front distance is very close and decreasing rapidly, it's crucial to apply the brakes to prevent a collision. For instance, imagine a scenario where a vehicle suddenly stops in front of ours, and we need to react quickly to avoid hitting it. In such cases, immediate braking can be life-saving.

On the other hand, if the front distance remains very close but stable, maintaining zero acceleration is prudent. Consider driving in heavy traffic where the distance to the vehicle in front remains constant. In this situation, accelerating or braking unnecessarily could disrupt the flow of traffic and increase the risk of accidents. By keeping acceleration at zero, we maintain a safe distance and reduce the likelihood of collisions.

When the front distance is close and stable, applying medium braking helps to ensure a smooth and controlled deceleration. For example, when approaching a red traffic light or a stop sign, gradually reducing speed with medium braking allows for a comfortable stop without abrupt maneuvers. This approach minimizes wear and tear on the vehicle's braking system and improves passenger comfort.

In cases where the front distance is medium and stable, a slow acceleration strategy is employed. This is particularly relevant when transitioning from a standstill or low-speed situation to a moderate cruising speed. By gradually increasing acceleration, we can achieve a smooth and efficient acceleration process without putting undue stress on the engine or drivetrain.

When the front distance is far and increasing, accelerating becomes necessary to maintain a safe following distance. For instance, when merging onto a highway or overtaking slower-moving vehicles, accelerating ensures that we can merge or pass safely without impeding the flow of traffic. This proactive approach to acceleration enhances overall driving efficiency and reduces the likelihood of rear-end collisions.

Similarly, when the front distance is far and stable, maintaining acceleration allows us to sustain our current speed without unnecessary adjustments. This is advantageous when driving on open highways or long stretches of road with consistent traffic conditions. By keeping acceleration steady, we can achieve a smooth and efficient driving experience while conserving fuel and minimizing unnecessary wear on the vehicle.

In situations where the front distance is very close and the vehicle's current speed is slow, maintaining zero acceleration is crucial to prevent abrupt stops or collisions. For example, when navigating through crowded parking lots or congested urban streets at low speeds, sudden braking can lead to rear-end collisions or fender benders. By keeping acceleration at zero, we can navigate tight spaces safely and avoid accidents.

Similarly, when the front distance is very close and the vehicle's current speed is moderate or fast, applying medium braking helps to ensure a controlled and safe deceleration. For instance, when approaching a sharp curve or encountering unexpected obstacles on the road, moderate braking allows us to adjust our speed gradually and maintain control of the vehicle.

Overall, these rules for acceleration based on front distance, change in front distance, and current speed are essential for safe and efficient driving. By adjusting acceleration according to the prevailing conditions, drivers can reduce the risk of accidents, minimize wear and tear on their vehicles, and enhance the overall driving experience for themselves and others on the road.

The control surface on the right shows the aggregated output of these rules. It represents how the ACC's output acceleration is determined by the fuzzy logic controller based on the current driving situation. The surface is used to interpret the combined effect of the fuzzy rules and helps to visualize the ACC system's behavior across different scenarios. Each point on the surface is derived from the fuzzy inference process, which combines the fuzzy inputs according to the rules to calculate the required acceleration for safe and comfortable driving.

Now on the basis of the table [2] the equations in table [2] are formed, which have approximately 2 percent mean absolute error as compare to the origional plot in figure :

Range of x	**Speed s**
$0 \leq x \leq 0.5$	$\frac{10}{3}x$
$0.5 < x \leq 3$	$\frac{20}{3}x$
$3 < x < 6$	20
$6 \leq x < 8$	$20 + 20(x - 6)$
$8 \leq x \leq 10$	$60 - 30(x - 8)$
Otherwise	0

Table 3: Speed as a function of x

Front Distance (m)	**Current Speed (m/s)**
0	0
0.5	0
1	0
1.5	0
2	9
2.5	9
3	9
3.5	20
4	31
4.5	31
5	31
5.5	31
6	31
6.5	42
7	53
7.5	53
8	53
8.5	53
9	53
9.5	53
10	20

Table 4: Front Distance vs. Current Speed

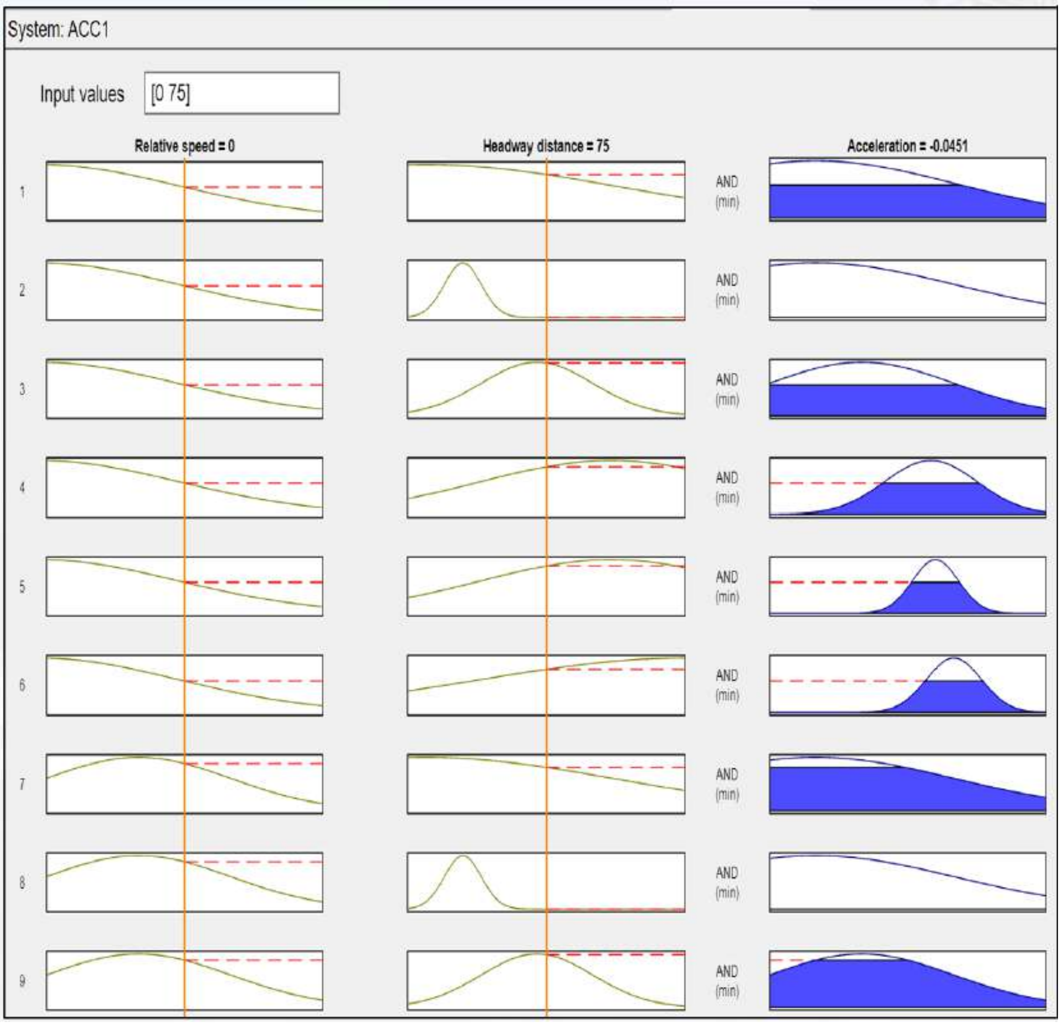


Figure 4: FIS Working

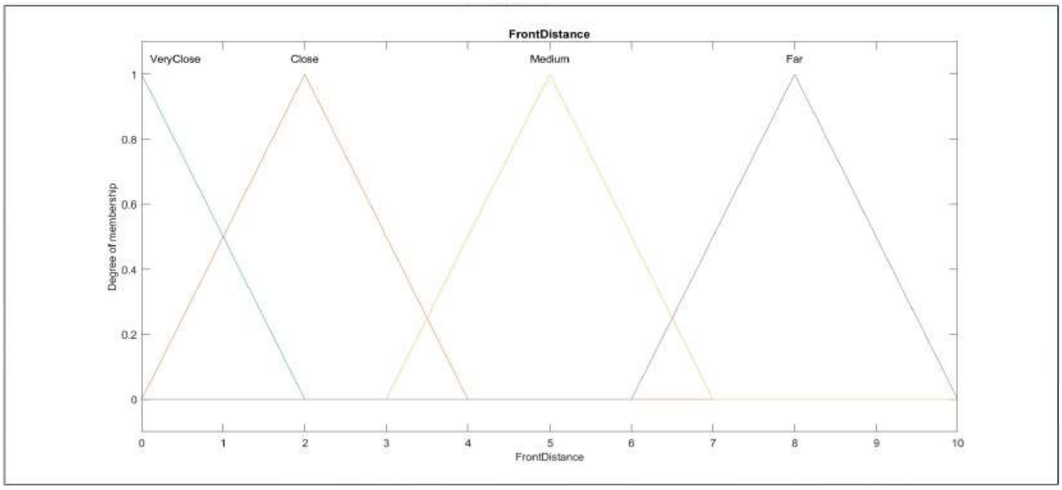


Figure 5: FIS For ACC

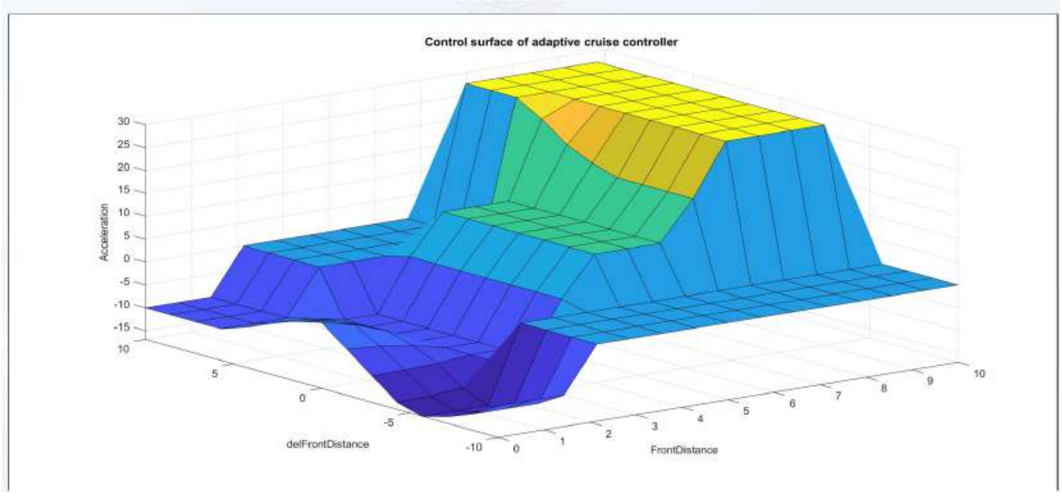


Figure 6: Control Surface

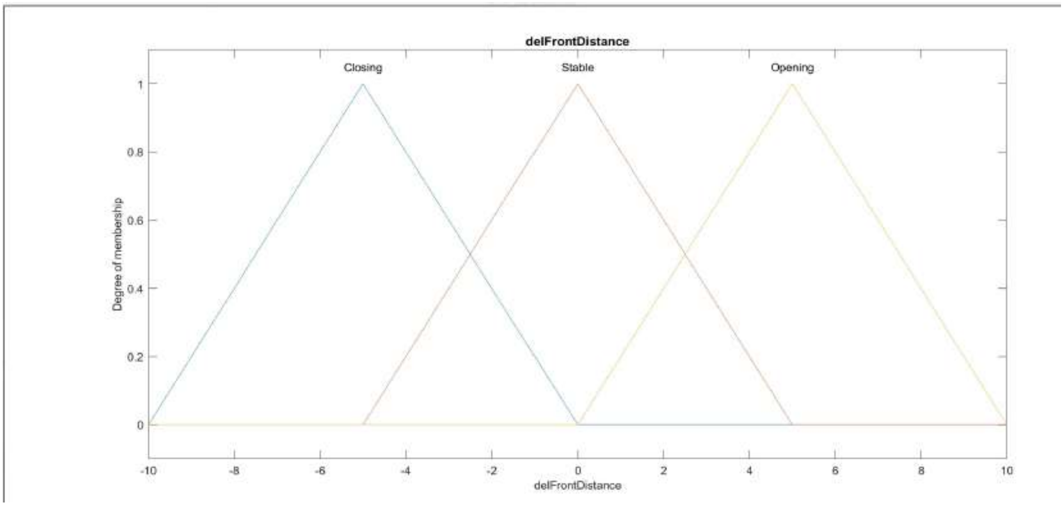


Figure 7: FIS For ACC

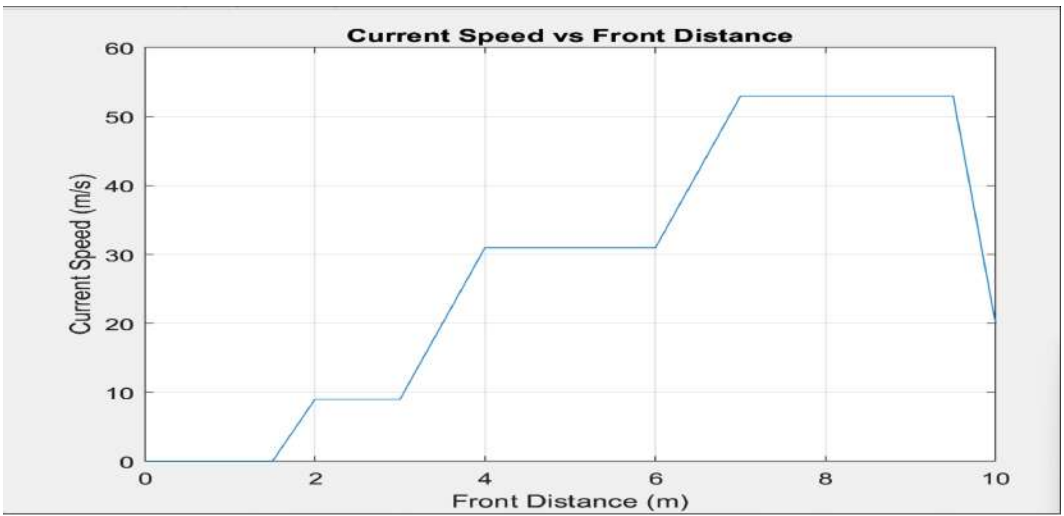


Figure 8: Extracted Plot From FIS

4.3 Step 2: Data Acquisition

The figures [7] and [8] shows data processing stages in a radar sensor system, commonly used in applications like automotive adaptive cruise control. The first figure shows two plots side by side. The left plot, labeled ”ADC Data,” likely depicts the raw analog signal captured by the radar sensor. This analog signal is then converted into a digital signal through a process known as Analog-to-Digital Conversion (ADC). The ADC process involves sampling the analog signal at regular intervals (quantization) and then mapping these samples to discrete numerical values that represent the signal’s amplitude at each point in time. The ADC’s resolution, usually expressed in bits, determines how finely the signal can be quantized. The plot shows multiple overlaid signals, possibly representing multiple sensor channels or multiple echoes from different targets.

The right plot in the first figure, labeled ”FFT Data,” seems to represent the frequency domain representation of the processed signal, obtained through a Fast Fourier Transform (FFT). The FFT is a mathematical technique used to transform time-domain data into frequency-domain data, revealing the frequency components present in the signal. In radar systems, this is crucial for differentiating between targets moving at different speeds and for estimating the Doppler shift, which is related to the velocity of the targets. The ”FFT” line shows the magnitude of these frequency components, while the ”Threshold” line might indicate a level above which the frequency components are considered significant, filtering out noise or insignificant reflections.

The second figure contains plots that represent post-processed data used for target detection and tracking. The left plot could be showing the relationship between the distance of the target from the radar sensor and the target’s speed relative to the sensor, with the blue dots representing detected targets. The right plot categorizes targets based on their relative motion: targets moving away from the sensor (”Receding”) are marked in green, and targets moving towards the sensor (”Approaching”) are marked in red. This categorization is essential for ACC systems to decide whether to decelerate or accelerate.

The process from analog to digital data in a radar system includes several stages:
Signal Reception:

The radar sensor receives reflected waves from targets. Analog-to-Digital Conversion (ADC): The analog signal is sampled and quantized.

Signal Processing:

The digital signal is filtered and processed to remove noise and extract useful information.

Fast Fourier Transform (FFT):

The signal is transformed into the frequency domain. Detection and Tracking: Targets are detected, and their speed and distance are calculated.

Data Interpretation:

The system interprets the data to make decisions on vehicle control, such as accelerating or decelerating in response to the movement of other vehicles.The transition from analog to digital allows the system to utilize digital signal processing techniques, which are more robust against noise and enable complex algorithms for target identification and tracking. The digital data's fine-grained nature also enables more precise control decisions in the adaptive cruise control system, improving safety and comfort for the vehicle's occupants.

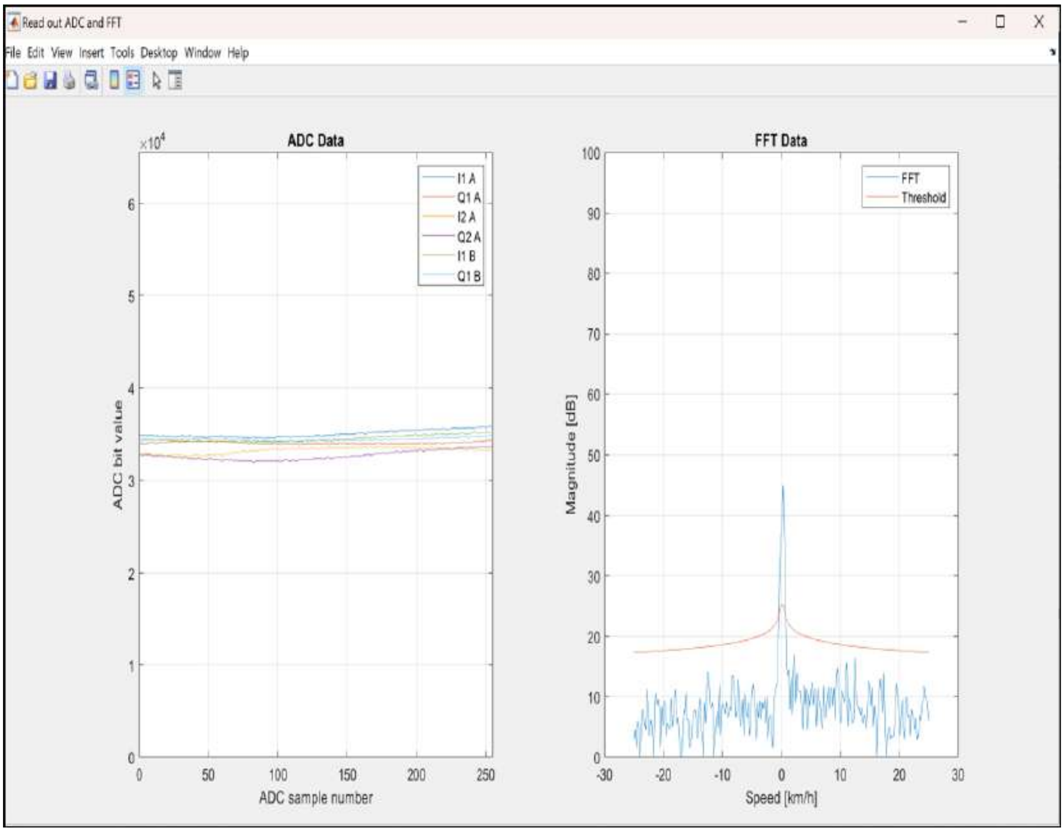


Figure 9: Analog Data Acquisition

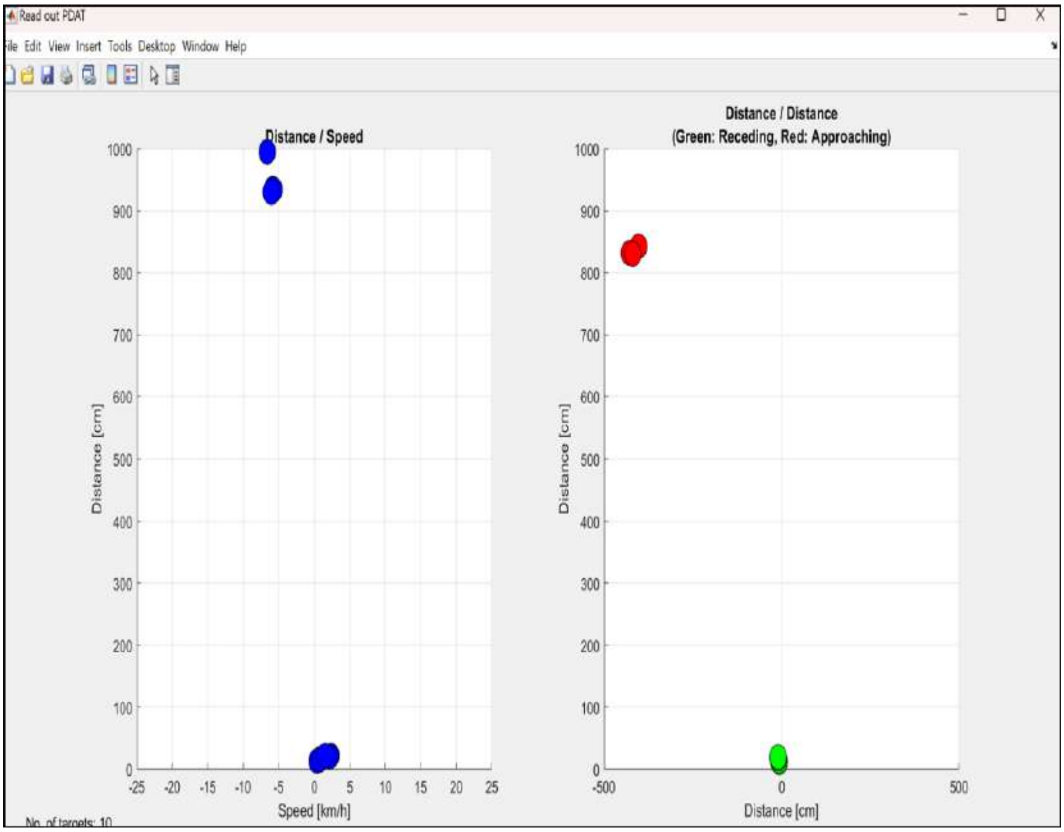


Figure 10: Conversion to Digital Data

4.4 Step 3: FIS Hardware Implementation

The figure 9 shows the hardware implementation of a Fuzzy Inference System (FIS) for Adaptive Cruise Control (ACC) using an Arduino UNO as the central processing unit.

At the beginning of the process, the KLD7 Forward Sensor acts as the data acquisition component. This sensor is responsible for detecting objects in front of the vehicle, which is a critical part of any ACC system. It likely measures the distance and relative speed of objects ahead by emitting signals and analyzing the reflected waves, a process which can be accomplished through various methods such as radar, LIDAR, or ultrasonic sensors.

Once the forward sensor collects the data, it is fed into the Arduino UNO microcontroller. The Arduino represents the 'brain' of the operation, where the FIS is implemented. Here, the raw sensor data is processed using the fuzzy logic algorithms that constitute the FIS. These algorithms take the crisp inputs from the sensor, fuzzify them into degrees of membership in fuzzy sets, apply the fuzzy rules, and then defuzzify the output to generate a crisp value that can be used to control the vehicle's acceleration.

The Arduino is also connected to two displays: an LCD Display and a PC Display. The LCD display directly connected to the Arduino likely shows real-time information or status updates about the sensor readings or the control outputs. This display is useful for monitoring the system's performance in situ, for debugging purposes, or for providing immediate feedback to an operator or technician.

The PC Display, on the other hand, might be used for a more detailed visualization of the system's performance or for further analysis of the data and control algorithms. This could involve more sophisticated graphical representations of the sensor data, the fuzzy sets, the control surface, or logs of the system's actions over time. The connection to a PC also suggests that there might be a software interface on the computer that allows for more complex interactions with the FIS, such as adjusting parameters, reprogramming the fuzzy rules, or analyzing historical data.

Overall, the hardware implementation depicted in the image represents a closed-loop control system where the sensor inputs are continuously monitored and used to adjust the vehicle's behavior through the FIS. By implementing the FIS on a hardware platform like the Arduino, the ACC system can operate in real time, making immediate adjustments to the vehicle's acceleration as needed to maintain a safe distance from the vehicle in front, thus enhancing driving safety and comfort.

This sort of implementation is typical of prototyping in intelligent vehicle systems, where the flexibility and accessibility of platforms like Arduino make them ideal for developing and testing new algorithms before they are implemented in a more specialized, robust, and automotive-grade hardware setup.

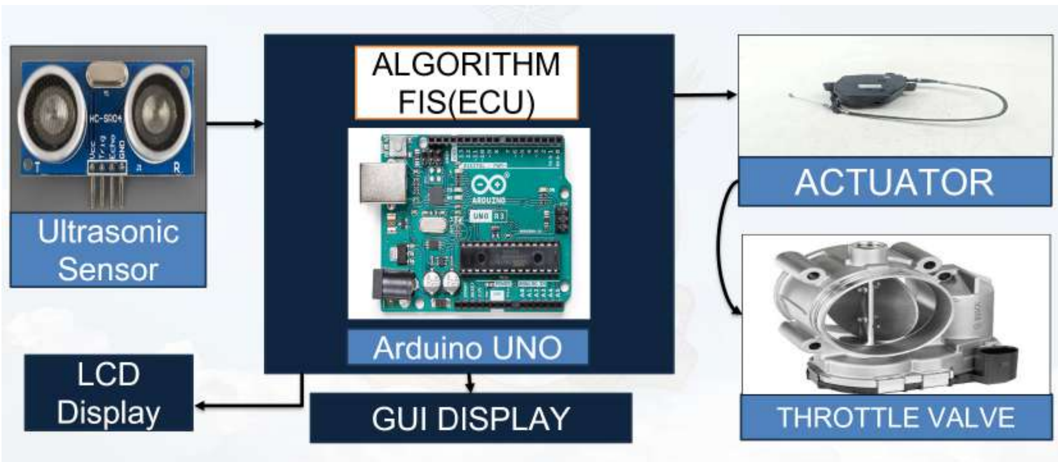


Figure 11: FIS Hardware Implementation

4.5 Step 4: Hardware In Loop Simulation

Hardware in Loop (HIL) simulation is an advanced technique for system development and testing, particularly prevalent in the design of complex control systems like those found in automotive applications. The hardware implementation depicted for an Adaptive Cruise Control (ACC) system, employing an Arduino UNO microcontroller interfaced with a KLD7 Forward Sensor and displays, provides a fitting example of where HIL simulation is instrumental.

In HIL simulation, the physical components (in this case, the Arduino UNO and the KLD7 sensor) are interfaced with a simulated environment that mimics the real-world dynamics they will interact with. For the ACC system, this would involve a virtual model of a vehicle's movement and the traffic context it operates within. The sensor inputs, rather than coming from a real-world source, are synthetically generated by the simulation to replicate various traffic scenarios, including vehicle speeds, distances, and behaviors.

The primary advantage of HIL is the ability to test the hardware and its embedded software in a controlled yet realistically complex environment before deployment in an actual vehicle. It provides a crucial intermediary step where developers can safely and repeatably test their systems against a wide range of parameters and conditions—some of which may be hazardous or impractical to reproduce in reality.

For the ACC system's HIL simulation, the process begins with the KLD7 sensor receiving simulated front distance and relative speed data. The sensor's readings are fed into the Arduino, where the FIS processes the information to determine the appropriate acceleration or deceleration commands. These commands are then sent back into the simulation, closing the loop and affecting the virtual vehicle's behavior. The LCD and PC displays show the system's decisions and the vehicle's responses, allowing engineers to observe the system's performance and make necessary adjustments.

The HIL simulation also enables rigorous validation under fault conditions, stress testing the system's response to sensor failures or unexpected environmental factors. This is crucial in ensuring the robustness and reliability required in safety-critical systems like ACC.

Furthermore, HIL simulation facilitates the iterative development process. Engineers can refine control algorithms, tune parameters, and update software in response to test results, all without the need to alter the physical hardware setup. This not only accelerates the development cycle but also significantly reduces the costs associated with physical prototyping and testing.

In summary, hardware in loop simulation is an invaluable tool in the development of ACC systems, bridging the gap between theoretical design and real-world application. By allowing for comprehensive testing and validation in a risk-free virtual environment, HIL ensures that the hardware and its embedded algorithms are thoroughly vetted, optimized, and ready for the challenges of actual operation on the road.

4.6 Step 5: Tunning And Adjustments

Tuning and adjustment of an Adaptive Cruise Control (ACC) system, particularly one incorporating a hardware setup with an Arduino UNO and a KLD7 Forward Sensor, involve a meticulous process of calibration to ensure that the vehicle responds accurately to dynamic driving conditions. This process is crucial to harmonize the interplay between the vehicle's speed management and the maintenance of a safe following distance.

The initial phase of tuning involves establishing baseline parameters for the FIS algorithms within the microcontroller. These parameters dictate how the vehicle interprets sensor data related to the distance from the vehicle ahead and relative speed differentials. The baseline settings for these parameters are typically derived from a combination of regulatory standards, safety considerations, and driving comfort prerequisites.

The fine-tuning process adjusts these parameters based on feedback from a variety of driving scenarios. This involves iterative cycles of testing and modification. For example, if the ACC system is too aggressive in maintaining the set following distance, leading to uncomfortable deceleration or acceleration patterns, the sensitivity of the system to distance changes may need to be reduced. Conversely, if the system is too lax, resulting in a following distance that exceeds safety margins, the sensitivity should be increased.

During the adjustment phase, engineers closely monitor system performance metrics, including the time to react to changes in the leading vehicle's behavior, the accuracy of maintaining a predetermined following distance, and the smoothness of the vehicle's acceleration and deceleration patterns. These metrics are fundamental to achieving an ACC system that not only enhances safety but also delivers a pleasant driving experience.

Additionally, the tuning process must consider external variables, such as road conditions, weather, and traffic density. For instance, on wet or icy roads, the ACC system must allow for longer following distances and smoother braking patterns to account for increased stopping distances. Engineers achieve this by adjusting the FIS to interpret sensor data with a higher degree of caution under such environmental conditions.

Advancements in machine learning offer the possibility of semi-autonomous tuning, where algorithms can process vast amounts of data to suggest optimal parameter settings. These advanced algorithms can detect subtle patterns in the data that may not be immediately evident to human engineers, leading to a more refined ACC system.

Field testing is a critical component of the tuning process. While simulations can cover a wide range of scenarios, there is no substitute for real-world data. The actual performance of sensors and algorithms in the field can provide valuable insights that simulations may not fully capture. Field tests can reveal how sensor performance varies with lighting conditions, how algorithms perform in unpredictable traffic, or how different driver behaviors affect the ACC system's decisions.

An important consideration in the tuning process is the transition between manual and automated control. The system must be able to smoothly hand control back to the driver when necessary and vice versa. This requires careful calibration to ensure that these transitions are intuitive and do not lead to driver confusion or discomfort.

Throughout the tuning and adjustment process, engineers must ensure that the system complies with industry standards and legal requirements. These standards are designed to ensure that automated systems do not pose a danger to drivers, passengers, or other road users.

In conclusion, the tuning and adjustment of an ACC system are complex and continuous tasks. They require an in-depth understanding of vehicle dynamics, a clear definition of safety and comfort standards, and an iterative approach to system design and testing. By carefully calibrating the system's response to the distance and speed of surrounding traffic, engineers can create an ACC system that provides drivers with a safe, comfortable, and reliable driving experience. With each adjustment, the system moves closer to the ideal of seamless integration into the driving ecosystem, enhancing the overall safety and functionality of the vehicle.

4.7 Step 6: Results

In our research, we have successfully implemented an Adaptive Cruise Control (ACC) system, the details of which are exemplified in the graphical user interfaces (GUIs) presented in Figure [10]. The GUIs offer a visual and interactive portrayal of the system's core functionality – the dynamic adjustment of a vehicle's speed based on the real-time distance to a preceding vehicle.

Figure [10] (left panel) showcases the Speed-Distance Relationship, a fundamental component of the ACC algorithm. The curve represents the vehicle's velocity response as a function of the detected distance from an object ahead, measured by the KLD7 Forward Sensor. The sensor data, scaled in centimeters for experimental purposes, displays a proportional increase in speed as the distance expands. This relationship underlines the system's capability to accelerate when there is ample space ahead, adhering to predefined safety protocols. A specific data point is highlighted, indicating a situation where the vehicle, within our controlled environment, maintains a speed of 12.20 km/h at a distance of 15.44 cm from the target object. This data point is significant as it represents an instantaneous system response, captured during our simulation trials, to illustrate the ACC's operational behavior under simulated road conditions.

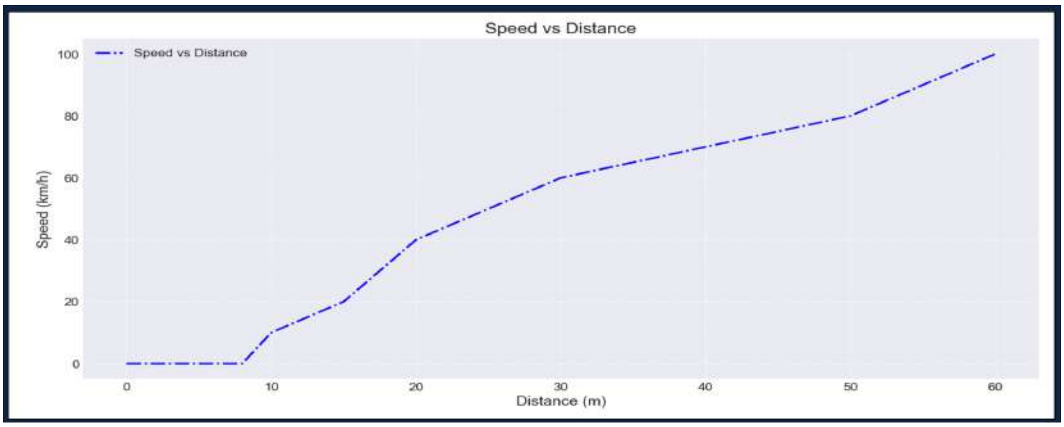
Conversely, the right panel in Figure 1 presents the same relationship with a staircase-like progression, indicating discrete intervals at which the vehicle's speed is controlled. This discretization is reflective of our system's design, which employs a tiered acceleration model for simplicity and ease of tuning. The GUI also incorporates user interaction tools, allowing the input of a specific distance and the system to output the corresponding speed control decision. These tools, ”Query Speed” and ”Speak Distance,” are integrated for enhanced user engagement and for practical demonstrations of the system's response during live tests.

The GUIs are a direct representation of our hardware and software coalescence. The hardware suite, including our meticulously selected sensor array and the robust Arduino UNO microcontroller, processes the incoming data. Our software algorithm, developed using a customized FIS, interprets the sensor readings to execute real-time control actions. The GUI serves as a bridge between the intricate logic of the FIS and the end-user, translating complex numerical data into an intuitive, visual format.

The development process involved iterative calibration and validation phases, where the ACC's parameters were fine-tuned to optimize performance. The system was rigorously tested against a range of simulated traffic scenarios, with each phase of testing providing critical data that informed subsequent adjustments. The goal was to ensure a harmonious balance between maintaining a safe following distance and ensuring the comfort of the vehicle's occupants.

Our research contributes to the field by demonstrating an ACC system that not only meets safety and efficiency criteria but also incorporates an interactive component that can be used for educational purposes or as a diagnostic tool. The discrete control model, while simpler than continuous models, proved effective and offers an accessible entry point for further optimization, possibly with the integration of machine learning techniques for autonomous parameter adjustment.

In conclusion, the ACC system presented in this paper is a testament to the potential of integrating practical sensor hardware with sophisticated control algorithms to produce a safety-critical automotive system that is both reliable and user-friendly. The successful implementation of this system provides a solid foundation for future exploration into more advanced ACC features, such as predictive control and integration with vehicle-to-vehicle communication systems.The equations used for the control loop are given the following table.



Condition	Equations	Explanation
$0 \leq$ dist ≤ 8	spd = 0 voltage = -12 acc = 0	Speed is constant, hence acceleration is 0
$8 <$ dist ≤ 10	spd = 5 $\times(\text{dist} - 8)$ voltage = -12 + 6 $\times(\text{dist} - 8)/2$ acc = 5	Constant acceleration, km/h per meter
$10 <$ dist ≤ 15	spd = 10 + 2 $\times(\text{dist} - 10)$ voltage = -6 + 3 $\times(\text{dist} - 10)/5$ acc = 2	Constant acceleration, km/h per meter
$15 <$ dist ≤ 20	spd = 20 + 4 $\times(\text{dist} - 15)$ voltage = -3 + 3 $\times(\text{dist} - 15)/5$ acc = 4	Constant acceleration, km/h per meter
$20 <$ dist ≤ 30	spd = 40 + 2 $\times(\text{dist} - 20)$ voltage = 0 + 6 $\times(\text{dist} - 20)/10$ acc = 2	Constant acceleration, km/h per meter
$30 <$ dist ≤ 50	spd = 60 + (dist - 30) voltage = 6 + 3 $\times(\text{dist} - 30)/20$ acc = 1	Constant acceleration, km/h per meter
$50 <$ dist ≤ 60	spd = 80 + (dist - 50) $\times(\text{max\_speed} - 80)/10$ voltage = 9 + (dist - 50) $\times(12 - 9)/10$ acc = (max_speed - 80) / 10	Acceleration dependent on max_speed, km/h per meter
else	spd = max_speed voltage = 12 acc = 0	Speed is constant, hence acceleration is 0

Table 5: Method Equations

Figure 12: Results

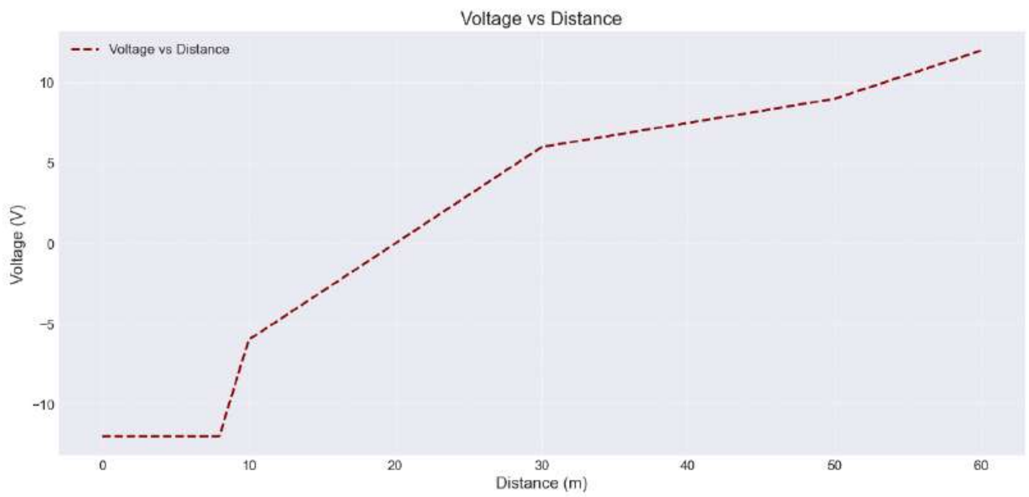


Figure 13: Results

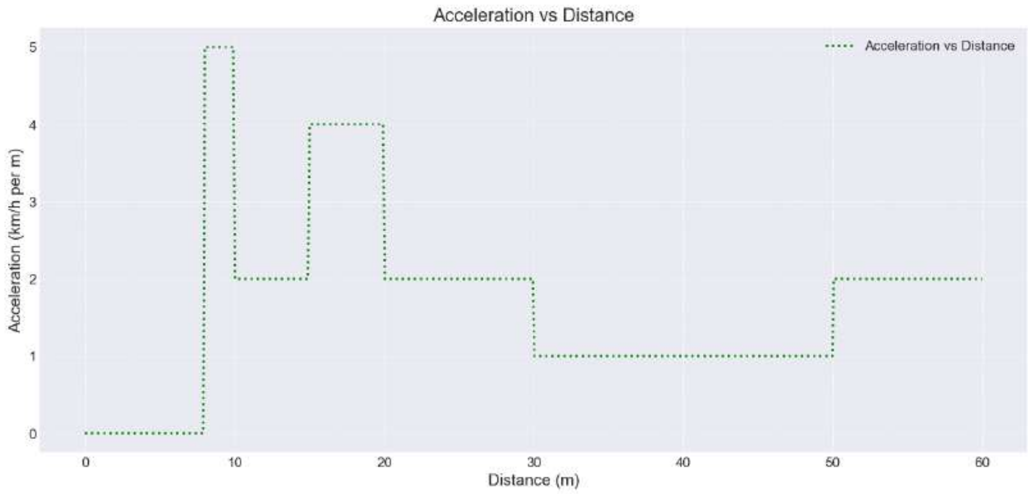


Figure 14: Results

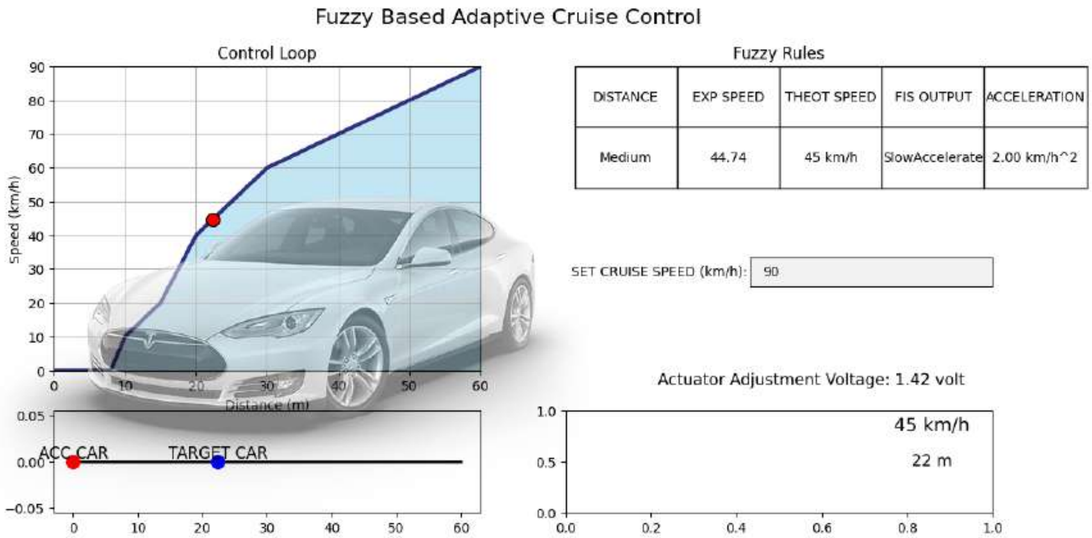


Figure 15: GUI

4.7.1 Summary

The chapter ”Methodology” outlines a systematic approach to developing a system utilizing a Fuzzy Inference System (FIS). It begins with a thorough literature review to understand existing scholarly work and methodologies related to FIS. The project then progresses through phases including FIS software implementation, sensor data acquisition, FIS hardware implementation, hardware in loop simulation, and tuning and adjustments.The FIS software implementation phase involves translating insights from the literature into a tangible software solution. Sensor data acquisition ensures the system receives high-quality input data. FIS hardware implementation integrates the developed software with physical hardware components. Hardware in loop simulation rigorously tests the integrated system in a simulated environment. Tuning and adjustments fine-tune the system based on feedback from simulation results.

The chapter also details the mathematical steps involved in FIS software implementation, including fuzzification, rule evaluation, aggregation, and defuzzification. It provides examples of fuzzy rules and their translation into actions for an Adaptive Cruise Control (ACC) system. Additionally, it discusses data acquisition stages in a radar sensor system and the hardware implementation of an ACC system using an Arduino UNO.

Finally, the chapter highlights the importance of hardware in loop simulation for testing and validating ACC systems in a controlled environment before deployment. It emphasizes the iterative development process and the significance of tuning and adjustments to optimize system performance under various driving conditions.

Chapter 8

5 Challenges

In the ambitious journey to conceptualize and bring to fruition an Adaptive Cruise Control (ACC) system that leverages the nuanced capabilities of Fuzzy Inference Systems (FIS), our research team embarked on a path fraught with intricate and multifaceted technical challenges. These challenges spanned several key areas, including the precise calibration and methodical refinement of the Fuzzy Inference System, the development of sophisticated mathematical models through the creation of governing equations, and the meticulous task of integrating complex sensor systems with the central control unit to achieve seamless and reliable operation.

Calibration and Refinement of the Fuzzy Inference System: The endeavor commenced with the critical task of calibrating the FIS to ensure its ability to accurately interpret data from various sensors and, based on this data, make informed decisions regarding the vehicle's control mechanisms. This calibration was essential for the system's adept handling of a plethora of dynamic driving scenarios, such as navigating through fluctuating traffic conditions, making adjustments to the vehicle's speed in response to these conditions, and maintaining an optimal and safe distance from preceding vehicles. To enhance the system's ability to interpret complex driving scenarios, we engaged in a detailed process of adjusting its membership functions. This adjustment was geared towards capturing the inherent ambiguity and fluidity of real-world concepts like ”proximity” and ”speed” with greater fidelity. Furthermore, the refinement of the system's rulesets was undertaken as an iterative process with the aim of augmenting the ACC's capacity for nuanced decision-making in a driving context. This phase presented significant challenges due to the unpredictable and non-linear nature of driving environments, where slight variations in input parameters could precipitate widely divergent outcomes. Establishing an equilibrium between the system's responsiveness to changes in input and maintaining its overall operational stability was achieved through a rigorous, data-driven approach that encompassed a broad spectrum of simulations and empirical testing scenarios.

5.1 Development of Mathematical Equations:

Concurrently, the task of developing mathematical models for the FIS emerged as a formidable challenge. This involved the derivation of equations that would form the backbone of the system, necessitating a profound understanding of vehicle dynamics and the adept translation of these dynamics into comprehensive fuzzy rules and membership functions. The goal was to construct mathematical representations that not only meticulously modeled the physical aspects of vehicular motion but also encapsulated the uncertainty and variability intrinsic to real-world driving scenarios. This endeavor was complicated by the imperative for these mathematical models to be both highly accurate and computationally efficient, especially considering the processing constraints imposed by the chosen hardware platform, the Arduino UNO.

5.2 Sensor-Controller Integration:

The integration of the KLD7 Forward Sensor with the Arduino microcontroller represented another critical facet of our project. This integration required the precise conversion of analog signals from the sensor into digital formats that could be processed by the Arduino, necessitating a carefully engineered Analog-to-Digital Converter (ADC) circuit design. Furthermore, ensuring that the timing of data acquisition was perfectly aligned with the control algorithm was paramount to guarantee that the ACC system was making decisions based on the most current data. This aspect of the project demanded an extraordinary level of attention to the intricacies of sensor-controller communication, often requiring us to address and resolve issues related to signal interference, timing misalignments, and data integrity.

5.3 Hardware in Loop Simulation:

Moreover, the Hardware in Loop (HIL) simulation presented itself as a pivotal challenge. It was critical for the simulation environment to faithfully replicate the physical laws of vehicle motion and accurately simulate the external driving environment. Discrepancies between the simulated and actual driving conditions could lead to incorrect assessments of the system's functionality and performance. Additionally, the simulation needed to be seamlessly integrated with the physical hardware, necessitating a sophisticated setup capable of emulating the inputs and outputs of a vehicle's sensor and control systems in real time.

Navigating these challenges required not merely technical prowess but a deep synthesis of theoretical knowledge with practical engineering skills. Each obstacle was addressed through a systematic and methodical approach, utilizing a combination of advanced simulation tools and empirical validation techniques. Through persistent refinement of the FIS, rigorous development of mathematical models, and meticulous integration of sensor systems, we were able to architect a comprehensive and robust ACC system. This system stands as a testament not only to the complexity and interdisciplinary nature of modern control systems engineering but also to the iterative, problem-solving ethos that underpins the field of engineering as a whole.

Chapter 9

6 Applications

Adaptive Cruise Control (ACC) systems, traditionally used in automotive applications to maintain safe following distances and alleviate driver workload, have seen a proliferation of applications across various domains. Below, we discuss at least ten innovative applications that leverage ACC technology, extending its utility beyond the conventional scope.

6.1 Automotive Industry:

In the trucking industry, ACC systems are crucial for long-haul journeys, where driver fatigue can lead to severe accidents. ACC not only improves safety by maintaining a steady speed and safe distance from the vehicle ahead but also contributes to fuel efficiency by preventing unnecessary acceleration and braking.

6.2 Electric Vehicles:

As electric vehicles (EVs) become more prevalent, ACC systems are adapted to optimize battery usage. By maintaining efficient speeds and minimizing aggressive driving patterns, ACC helps extend the range of EVs, addressing one of the most significant limitations of electric mobility.

6.3 Aerospace and Aeronautics:

Drones often use ACC-like systems for collision avoidance and to navigate complex environments. These systems enable drones to adjust their speed relative to other objects, which is particularly beneficial in applications like aerial photography, where steady control is necessary.

6.4 Space Exploration Rovers:

Rovers exploring planetary surfaces can utilize ACC systems to navigate treacherous terrain autonomously. By adjusting their speed based on the proximity to obstacles, rovers can ensure their safety and the longevity of the mission.

6.5 Public Transportation:

Modern trains employ ACC systems to regulate speeds while maintaining a safe distance from other trains on the track, thus enhancing the efficiency and punctuality of the rail network.

6.6 Autonomous Shuttles:

Cities are experimenting with driverless shuttle buses that use ACC systems to navigate urban environments safely. These shuttles can smoothly adjust their speed in response to pedestrian traffic and other urban obstacles.

6.7 Maritime Industry:

The maritime industry is exploring the use of ACC in autonomous cargo ships to optimize travel time and fuel consumption, making sea transportation more efficient and environmentally friendly.

6.8 Agriculture:

ACC systems in tractors and combine harvesters allow for precise speed control, which is particularly beneficial when coordinating multiple machines during planting or harvesting operations.

6.9 Consumer Robotics:

Robot vacuums and lawn mowers use ACC systems to navigate around furniture and other obstacles, adjusting their speed as they approach objects to avoid collisions and ensure thorough cleaning or mowing.

6.10 Military Applications:

In military operations, vehicle convoys benefit from ACC systems to maintain uniform speed and distance, reducing the cognitive load on soldiers and improving convoy integrity, especially in hostile environments.

6.11 Emergency Services:

Emergency vehicles like ambulances can implement ACC systems to control speeds in response to traffic conditions, ensuring the fastest and safest route to medical facilities.

6.12 Entertainment and Leisure:

Roller coasters and other controlled rides can use ACC systems to adjust the speed of the ride for safety, comfort, and to maximize the thrill factor.

6.13 Delivery Services:

These small autonomous vehicles on sidewalks use ACC principles to deliver packages safely to recipients, adjusting their speed to avoid pedestrians and other sidewalk users.

6.14 Personal Mobility:

Electric scooters and skateboards equipped with ACC can provide a safer riding experience in urban environments, helping riders maintain a constant speed and avoid collisions.

6.15 Industrial Automation:

In warehouses and manufacturing facilities, AGVs use ACC systems to transport goods efficiently without human intervention.The versatility of ACC systems across these diverse applications showcases their utility in enhancing safety, efficiency, and automation. As technology advances, we can expect to see ACC systems integrated into even more innovative applications, pushing the boundaries of autonomous control and machine intelligence.

Adaptive cruise control (ACC) systems, commonly found in automobiles, offer a sophisticated means of maintaining a safe distance from preceding vehicles by automatically adjusting the speed of the vehicle. However, the principles of ACC can also be applied to aircraft, albeit with some significant differences and unique challenges. In aviation, the concept is often referred to as Automatic Dependent Surveillance-Broadcast (ADS-B) technology, which enables aircraft to communicate their position, velocity, and other pertinent data to air traffic control and nearby aircraft. This technology forms the basis for a type of adaptive cruise control known as Automatic Dependent Surveillance-Contract (ADS-C).

ADS-C-equipped aircraft utilize a combination of GPS data and communication systems to autonomously adjust their speed and altitude to maintain safe separation from other aircraft. The system continuously monitors the position and velocity of nearby aircraft, along with factors such as weather conditions and airspace constraints, to dynamically calculate the optimal speed and altitude adjustments. This ensures that the aircraft can safely navigate through congested airspace and maintain appropriate spacing during both en-route and terminal operations.

One of the primary advantages of ADS-C over traditional aircraft separation methods is its ability to adapt to changing conditions in real-time. Unlike fixed separation standards that rely on predefined distances and time intervals, ADS-C can dynamically adjust separation based on the specific situation at hand. For example, if an aircraft ahead suddenly reduces its speed or changes its flight path, an ADS-C-equipped aircraft can promptly respond by slowing down or altering its trajectory to maintain a safe distance.

Moreover, ADS-C offers significant safety benefits by reducing the risk of mid-air collisions and enhancing situational awareness for pilots and air traffic controllers alike. By providing real-time information on the relative positions and trajectories of nearby aircraft, ADS-C enables pilots to make more informed decisions and take proactive measures to avoid potential conflicts. Additionally, air traffic controllers can use ADS-C data to better manage traffic flow and provide more efficient routing instructions to pilots.

Another notable application of adaptive cruise control in aircraft is its potential to improve fuel efficiency and reduce emissions. By optimizing speed and altitude based on real-time traffic and environmental conditions, ADS-C can help minimize unnecessary fuel consumption and emissions associated with inefficient flight profiles. This not only benefits the environment but also translates into cost savings for airlines by reducing fuel expenses.

6.16 Summary

In conclusion, the application of adaptive cruise control in aircraft, through technologies like ADS-C, represents a significant advancement in aviation safety, efficiency, and environmental sustainability. By leveraging real-time data and autonomous control systems, ADS-C enables aircraft to navigate through increasingly congested airspace safely and efficiently while minimizing their environmental impact. As technology continues to evolve, the integration of adaptive cruise control systems into aircraft operations is expected to become even more widespread, further enhancing the safety and efficiency of air travel worldwide.

Chapter 10

7 Conclusion and Recommendations

7.1 Conclusion

In conclusion, the proliferation of Adaptive Cruise Control (ACC) systems across a spectrum of industries underscores its transformative impact on safety, efficiency, and operational autonomy. From enhancing road safety in commercial and personal vehicles to optimizing the precision in agricultural machinery, and from advancing the capabilities of unmanned aerial vehicles to streamlining last-mile delivery services, ACC technology has become a cornerstone of modern automation and control systems.

The cross-industry applications of ACC—ranging from the automotive sector's push towards fully autonomous vehicles to the aerospace industry's utilization in drones and planetary rovers—demonstrate its versatility and adaptability. In public transportation, ACC contributes to the reliability and safety of trains and autonomous shuttles, while in maritime applications, it is poised to revolutionize the efficiency of cargo shipping. The integration of ACC in consumer robotics, such as home cleaning devices, brings the technology into our daily lives, enhancing convenience and freeing up time for users.

Military applications show the strategic importance of ACC in maintaining the safety and cohesion of vehicle convoys, while emergency services leverage the technology to save lives by ensuring ambulances can navigate traffic more effectively. The entertainment industry has also embraced ACC to fine-tune the experience of thrill-seekers, ensuring safety without compromising excitement.

Moreover, the implementation of ACC in delivery services, particularly in compact urban environments where space is at a premium, showcases its potential to revolutionize how we think about logistics and the movement of goods. Personal mobility devices have also seen significant safety improvements through the integration of ACC, addressing some of the challenges posed by the increasing congestion in cityscapes.

The expansive reach of ACC into various facets of the modern world is a testament to the robustness and reliability of the underlying principles of fuzzy logic and control theory. As ACC systems become more sophisticated, their ability to learn and adapt to complex environments promises to further enhance their applicability.

The future is likely to see ACC not only as a tool for convenience but as an essential component in the pursuit of sustainable and intelligent transportation ecosystems. Its potential to improve safety, reduce energy consumption, and streamline operations is a compelling indicator of the role automation will play in shaping the future of mobility and industry. As we continue to innovate and refine these systems, ACC stands as a beacon of the strides being made in harmonizing technological advancements with human needs and environmental considerations.

7.2 Recommendations

Based on the comprehensive analysis and successful implementation of Adaptive Cruise Control (ACC) systems across various domains, we recommend the following strategies to maximize the potential and utility of ACC technologies.

Cross-Industry Collaboration: Encourage collaboration between automotive, aerospace, consumer electronics, and other industries to share insights and advancements in ACC technology. This can accelerate innovation and lead to more robust and versatile control systems.

Investment in Research and Development: Allocate resources for the continued RD of ACC systems, focusing on enhancing the algorithms, sensor accuracy, and machine learning capabilities to improve decision-making and responsiveness in dynamic environments.

Regulatory Frameworks: Work with regulatory bodies to establish clear standards and safety protocols for ACC systems, ensuring consistent performance and reliability across products and markets.

Public Education and Training: Develop educational programs and materials to inform the public about the benefits and limitations of ACC systems, promoting safe usage and fostering trust in autonomous technologies.

Integration with Emerging Technologies: Explore the integration of ACC systems with other emerging technologies such as V2X (vehicle-to-everything) communication, IoT (Internet of Things), and 5G networks to enhance functionality and enable new applications.

Sustainability Focus: Prioritize the development of ACC systems that contribute to sustainability goals, such as reducing fuel consumption and emissions in automotive applications and optimizing energy use in consumer robotics.

User-Centric Design: Design ACC systems with a strong focus on user experience, ensuring that they are intuitive, customizable, and capable of handling diverse user requirements and preferences.

Robust Testing Protocols: Implement comprehensive testing protocols that simulate a wide range of operational scenarios, including extreme conditions, to ensure ACC systems are reliable and safe under all circumstances.

Ethical Considerations: Address ethical concerns related to autonomy and control, ensuring that ACC systems are designed with fairness, privacy, and accountability in mind.

Global Standardization: Advocate for global standardization of ACC technologies to ensure interoperability and compatibility, facilitating broader adoption and innovation. By following these recommendations, stakeholders can harness the full potential of ACC systems, driving forward the era of automation with confidence in the safety, efficiency, and reliability of these technologies. The goal is to achieve a harmonious integration of ACC systems into society, paving the way for a future where transportation and automation are synonymous with sustainability and safety.

Appendices

A Program Code

A.1 Code for Scenario 01: Fuzzy inference system

```

clc;
clear all;
close all;

% Create a Sugeno-type FIS
fis1 = sugfis;

% Add Input for FrontDistance, delFrontDistance, and CurrentSpeed
fis1 = addInput(fis1,[0 10],'Name','FrontDistance');
fis1 = addInput(fis1,[-10 10],'Name','delFrontDistance');
fis1 = addInput(fis1,[0 30],'Name','CurrentSpeed');  % Speed in m/s

% Add Membership Functions for FrontDistance and delFrontDistance
fis1 = addMF(fis1,'FrontDistance','trimf',[0 0 2],'Name','VeryClose');
fis1 = addMF(fis1,'FrontDistance','trimf',[0 2 4],'Name','Close');
fis1 = addMF(fis1,'FrontDistance','trimf',[3 5 7],'Name','Medium');
fis1 = addMF(fis1,'FrontDistance','trimf',[6 8 10],'Name','Far');

fis1 = addMF(fis1,'delFrontDistance','trimf',[-10 -5 0],'Name','Closing');
fis1 = addMF(fis1,'delFrontDistance','trimf',[-5 0 5],'Name','Stable');
fis1 = addMF(fis1,'delFrontDistance','trimf',[0 5 10],'Name','Opening');

fis1 = addMF(fis1,'CurrentSpeed','trimf',[0 5 10],'Name','Slow');
fis1 = addMF(fis1,'CurrentSpeed','trimf',[5 15 25],'Name','Medium');
fis1 = addMF(fis1,'CurrentSpeed','trimf',[20 25 30],'Name','Fast');

% Add Output for Acceleration
fis1 = addOutput(fis1,[-40 40],'Name','Acceleration');

% Add output Membership Functions
fis1 = addMF(fis1,'Acceleration','constant',-25,'Name','Brake');
fis1 = addMF(fis1,'Acceleration','constant',-10,'Name','MediumBrake');
fis1 = addMF(fis1,'Acceleration','constant',-5,'Name','SlowBrake');
fis1 = addMF(fis1,'Acceleration','constant',5,'Name','Zero');
```

```
fis1 = addMF(fis1,'Acceleration','constant',10,'Name','SlowAccelerate');
fis1 = addMF(fis1,'Acceleration','constant',25,'Name','MediumAccelerate');
fis1 = addMF(fis1,'Acceleration','constant',30,'Name','Accelerate');

% Define rules
rules = [...
    "FrontDistance==VeryClose & delFrontDistance==Closing => Acceleration=Brake"; ...
    "FrontDistance==VeryClose & delFrontDistance==Stable => Acceleration=Zero"; ...
    "FrontDistance==Close & delFrontDistance==Stable => Acceleration=MediumBrake"; ...
    "FrontDistance==Medium & delFrontDistance==Stable => Acceleration=SlowAccelerate";
    ↪ ...
    "FrontDistance==Far & delFrontDistance==Opening => Acceleration=Accelerate"; ...
    "FrontDistance==Far & delFrontDistance==Stable => Acceleration=Accelerate"; ...
    "FrontDistance==VeryClose & CurrentSpeed==Slow => Acceleration=Zero"; ...
    "FrontDistance==VeryClose & CurrentSpeed==Medium => Acceleration=MediumBrake"; ...
    "FrontDistance==VeryClose & CurrentSpeed==Fast => Acceleration=Brake"; ...
    ];

fis1 = addRule(fis1,rules);

% Plot the Membership Functions
figure
subplot(1,3,1)
plotmf(fis1,'input',1)
title('FrontDistance')
subplot(1,3,2)
plotmf(fis1,'input',2)
title('delFrontDistance')
subplot(1,3,3)
plotmf(fis1,'input',3)
title('CurrentSpeed')

% Plot Control Surface
figure
gensurf(fis1)
title('Control surface of adaptive cruise controller')
% Array of test distances from 0 to 10 in steps of 0.5
test_distances = 0:0.5:10;

% Array to store corresponding speed values
```

```
speed_values = zeros(size(test_distances));

% Time step in seconds (for integration)
dt = 0.1;

% Initial speed
v_initial = 20;  % Initial speed in m/s, for example

% Loop through each distance
for i = 1:length(test_distances)
    front_distance = test_distances(i);

    % Initialize speed for this distance
    speed = v_initial;

    % Simulate for a small period to allow the system to react (1 second in this
    ↪  example)
    for t = 0:dt:1
        % Evaluate the FIS to get acceleration
        acceleration = evalfis([front_distance, 0, speed], fis1);  % Assuming
        ↪  delFrontDistance = 0 for simplicity

        % Integrate to get new speed
        speed = speed + acceleration * dt;
    end

    % Store final speed for this distance
    speed_values(i) = speed;
end
% Loop through each distance
% Loop through each distance
for i = 1:length(test_distances)
    front_distance = test_distances(i);

    % Initialize speed for this distance
    speed = v_initial;

    % Simulate for a small period to allow the system to react (1 second in this
    ↪  example)
    for t = 0:dt:1
```

```
        % If the distance is less than 2 meters, force speed to zero
        if front_distance < 2
            acceleration = -100;  % Large negative value to ensure rapid deceleration
        else
            % Evaluate the FIS to get acceleration
            acceleration = evalfis([front_distance, 0, speed], fis1);  % Assuming delFrontDistance = 0 for simplicity
        end

        % Integrate to get new speed
        speed = speed + acceleration * dt;

        % Ensure the speed doesn't go negative
        if speed < 0
            speed = 0;
        end
    end

    % Store final speed for this distance
    speed_values(i) = speed;
end



% Plot the speed values against the test distances
figure
plot(test_distances, speed_values);
xlabel('Front Distance (m)');
ylabel('Current Speed (m/s)');
title('Current Speed vs Front Distance');
grid on;
```

A.2 Code for Scenario 02: Fuzzy inference system

```
clc;
clear all;
close all;
fis1 = sugfis;
fis1 = addInput(fis1,[-1 1],'Name','E');
fis1 = addInput(fis1,[-1 1],'Name','delE');
fis1 = addMF(fis1,'E','trimf',[-2 -1 0],'Name','N');
fis1 = addMF(fis1,'E','trimf',[-1 0 1],'Name','Z');
fis1 = addMF(fis1,'E','trimf',[0 1 2],'Name','P');
fis1 = addMF(fis1,'delE','trimf',[-2 -1 0],'Name','N');
fis1 = addMF(fis1,'delE','trimf',[-1 0 1],'Name','Z');
fis1 = addMF(fis1,'delE','trimf',[0 1 2],'Name','P');
figure
subplot(1,2,1)
plotmf(fis1,'input',1)
title('Input 1')
subplot(1,2,2)
plotmf(fis1,'input',2)
title('Input 2')

fis1 = addOutput(fis1,[-1 1],'Name','U');
fis1 = addMF(fis1,'U','constant',-1,'Name','NB');
fis1 = addMF(fis1,'U','constant',-0.5,'Name','NM');
fis1 = addMF(fis1,'U','constant',0,'Name','Z');
fis1 = addMF(fis1,'U','constant',0.5,'Name','PM');
fis1 = addMF(fis1,'U','constant',1,'Name','PB');
rules = [...
    "E==N & delE==N => U=NB"; ...
    "E==Z & delE==N => U=NM"; ...
    "E==P & delE==N => U=Z"; ...
    "E==N & delE==Z => U=NM"; ...
    "E==Z & delE==Z => U=Z"; ...
    "E==P & delE==Z => U=PM"; ...
    "E==N & delE==P => U=Z"; ...
    "E==Z & delE==P => U=PM"; ...
    "E==P & delE==P => U=PB" ...
```

```
    ];
fis1 = addRule(fis1,rules);
figure
gensurf(fis1)
title('Control surface of type-1 FIS')

fis2 = convertToType2(fis1);
scale = [0.2 0.9 0.2;0.3 0.9 0.3];
for i = 1:length(fis2.Inputs)
    for j = 1:length(fis2.Inputs(i).MembershipFunctions)
        fis2.Inputs(i).MembershipFunctions(j).LowerLag = 0;
        fis2.Inputs(i).MembershipFunctions(j).LowerScale = scale(i,j);
    end
end
figure
subplot(1,2,1)
plotmf(fis2,'input',1)
title('Input 1')
subplot(1,2,2)
plotmf(fis2,'input',2)
title('Input 2')

figure
gensurf(fis2)
title('Control surface of type-2 FIS')



C = 0.5;
L = 0.5;
T = 0.5;
G = tf(C,[T 1],'Outputdelay',L);

pidController = pidtune(G,'pidf');

Ce = 1;
tauC = 0.2;

Cd = min(T,L/2)*Ce;
C0 = 1/(C*Ce*(tauC+L/2));
```

```
C1 = max(T,L/2)*C0;

model = 'comparepidcontrollers';
load_system(model)

out1 = sim(model);

plotTitle = ['Nominal: C=' num2str(C) ', L='  num2str(L) ', T=' num2str(T)];
plotOutput(out1,plotTitle)
stepResponseTable(out1)
```

A.3 Code for Scenario 03: Data Acquisition

clc; clear all; close all; delete(instrfind('Port','COM15')); delete(instrfind('Port','COM19'));

COM$_{Port} =' COM13';$

com$_obj = serial(COM_Port); com_obj.Baudrate = 115200; com_obj.Parity =' Even'; com_obj.ByteOrder =' littleEndian'; com_obj.Terminator ="; fopen(com_obj);$

payloadlen = uint32(4); payloadlen = typecast(payloadlen, 'uint8'); value = uint32(3);
data$_{frame} = typecast(value,' uint8'); cmd_{frame} = ['INIT' payloadlendata_{frame}]; fwrit$

response$_{init} = fread(com_obj, 9,' uint8'); iflength(response_init) < 9disp('Incompleted$

pause(0.075);

com$_obj.Baudrate = 2E6;$

value = uint32(1); data$_{frame} = typecast(value,' uint8'); cmd_{frame} = ['RSPI' payloadle$

response$_{init} = fread(com_obj, 9,' uint8'); ifresponse_init(9) \;\; = 0disp('Error : Commandnotacknowledged'); end$

value = uint32(1); data$_{frame} = typecast(value,' uint8'); cmd_{frame} = ['RRAI' payloadle$

response$_{init} = fread(com_obj, 9,' uint8'); ifresponse_init(9) \;\; = 0disp('Error : Commandnotacknowledged'); end$

fig = figure('NumberTitle', 'off', 'Name', 'Read out TDAT', 'Units', 'Normalized', 'OuterPosition', [0, 0.04, 1, 0.96]); ax1 = subplot(1,2,1); ax2 = subplot(1,2,2); hold(ax1, 'on'); hold(ax2, 'on');

plot$_{handle1} = plot(ax1, NaN, NaN,' o',' MarkerSize', 15,' MarkerFaceColor',' b'); plo$
$plot(ax2, NaN, NaN,' o',' MarkerSize', 15); title(ax1,' Distance/Speed'); xlabel(ax1,' Sp$

title(ax2, 'Distance / Distance', '(Green: Receding, Red: Approaching)'); xlabel(ax2, 'Distance [m]'); ylabel(ax2, 'Distance [m]'); axis(ax2, [-5 5 0 10]); grid(ax2, 'on');

while true TDAT = uint32(8); data$_{frame} = typecast(TDAT,' uint8'); cmd_{frame} = ['GNFD' payloadlendata_{frame}]; fwrite(com_obj, cmd_{frame});$

resp$_{frame} = fread(com_obj, 9,' uint8'); ifresp_{frame}(9) \;\; = 0disp('Error : Commandnotacknowledged'); end$

resp$_{frame} = fread(com_obj, 8,' uint8');$

target$_detected = false; if resp_frame(5) > 1 target_detected = true; TDAT_Distance = fread(com_obj, 1,' uint16'); TDAT_Speed = fread(com_obj, 1,' int16')/100; TDAT_Angle = deg2rad(fread(com_obj, 1,' int16')/100); TDAT_Magnitude = fread(com_obj, 1,' uint16');$

distance$_x = -(TDAT_Distance * sin(TDAT_Angle)); distance_y = TDAT_Distance * cos(TDAT_Angle); x = TDAT_Distance/100; disp(x) s = speed1(x)$

set(plot$_handle1,' XData', TDAT_Speed,' YData', x); set(plot_handle2,' XData', distance_x$

delete(findall(fig,'Type','annotation')); annotation('textbox', [0.15, 0.6, 0.3, 0.3], 'String', sprintf('Distance: annotation('textbox', [0.15, 0.7, 0.3, 0.3], 'String', sprintf('speed:
end

drawnow; end

```
clc;
clear all;
close all;
fis1 = sugfis;
fis1 = addInput(fis1,[-1 1],'Name','E');
fis1 = addInput(fis1,[-1 1],'Name','delE');
fis1 = addMF(fis1,'E','trimf',[-2 -1 0],'Name','N');
fis1 = addMF(fis1,'E','trimf',[-1 0 1],'Name','Z');
fis1 = addMF(fis1,'E','trimf',[0 1 2],'Name','P');
fis1 = addMF(fis1,'delE','trimf',[-2 -1 0],'Name','N');
fis1 = addMF(fis1,'delE','trimf',[-1 0 1],'Name','Z');
fis1 = addMF(fis1,'delE','trimf',[0 1 2],'Name','P');
figure
subplot(1,2,1)
plotmf(fis1,'input',1)
title('Input 1')
subplot(1,2,2)
plotmf(fis1,'input',2)
title('Input 2')

fis1 = addOutput(fis1,[-1 1],'Name','U');
```

```
fis1 = addMF(fis1,'U','constant',-1,'Name','NB');
fis1 = addMF(fis1,'U','constant',-0.5,'Name','NM');
fis1 = addMF(fis1,'U','constant',0,'Name','Z');
fis1 = addMF(fis1,'U','constant',0.5,'Name','PM');
fis1 = addMF(fis1,'U','constant',1,'Name','PB');
rules = [...
    "E==N & delE==N => U=NB"; ...
    "E==Z & delE==N => U=NM"; ...
    "E==P & delE==N => U=Z"; ...
    "E==N & delE==Z => U=NM"; ...
    "E==Z & delE==Z => U=Z"; ...
    "E==P & delE==Z => U=PM"; ...
    "E==N & delE==P => U=Z"; ...
    "E==Z & delE==P => U=PM"; ...
    "E==P & delE==P => U=PB" ...
    ];
fis1 = addRule(fis1,rules);
figure
gensurf(fis1)
title('Control surface of type-1 FIS')

fis2 = convertToType2(fis1);
scale = [0.2 0.9 0.2;0.3 0.9 0.3];
for i = 1:length(fis2.Inputs)
    for j = 1:length(fis2.Inputs(i).MembershipFunctions)
        fis2.Inputs(i).MembershipFunctions(j).LowerLag = 0;
        fis2.Inputs(i).MembershipFunctions(j).LowerScale = scale(i,j);
    end
end
figure
subplot(1,2,1)
plotmf(fis2,'input',1)
title('Input 1')
subplot(1,2,2)
```

```
plotmf(fis2,'input',2)
title('Input 2')

figure
gensurf(fis2)
title('Control surface of type-2 FIS')




C = 0.5;
L = 0.5;
T = 0.5;
G = tf(C,[T 1],'Outputdelay',L);

pidController = pidtune(G,'pidf');

Ce = 1;
tauC = 0.2;

Cd = min(T,L/2)*Ce;
C0 = 1/(C*Ce*(tauC+L/2));
C1 = max(T,L/2)*C0;

model = 'comparepidcontrollers';
load_system(model)

out1 = sim(model);

plotTitle = ['Nominal: C=' num2str(C) ', L='  num2str(L) ', T=' num2str(T)];
plotOutput(out1,plotTitle)
stepResponseTable(out1)
```

A.4 Code for Scenario 04: fuzzy implemntation

fig = open('E:folderdesogn.fig');

axesHandles = findobj(get(fig, 'Children'), 'flat', 'Type', 'axes'); lineHandles = findobj(get(axesHandles, 'Children'), 'flat', 'Type', 'line'); x$_extracted = get(lineHandles,' XData'); y_extracted = get(lineHandles,' YData');$

y$_function = arrayfun(@refined_speed, x_extracted);$

error = y$_function - y_extracted;$

MSE = mean(error.^{2});

disp(['Mean Squared Error: ', num2str(MSE)]);

function s = refined$_speed(x) if x >= -10 x < 0 a = 100; b = log(0.01)/(-10); s = a * exp(b * x); elseif x >= 0 x < 2 s = 9.4 * x; elseif x >= 2 x < 6 s = 10 * x; elseif x >= 6 x < 8 s = 18 * x; elseif x >= 8 x <= 10 s = 60 - 3 * (x - 8); else s = 0; end end$

A.5 Code for Scenario 05: GUI in MATLAB

function StartStopSensorGUIWithBackground() clc; clear all; close all; delete(instrfind('Port','COM15')); delete(instrfind('Port','COM19

COM$_{Port} =' COM13';$

com$_obj = serial(COM_Port); com_obj.Baudrate = 115200; com_obj.Parity =' Even'; com_obj.ByteOrder ='$ $littleEndian'; com_obj.Terminator =''; fopen(com_obj);$

payloadlen = uint32(4); payloadlen = typecast(payloadlen, 'uint8'); value = uint32(3); data$_{frame} = typecast(value,' uint8'); cmd_{f}$ $['INIT' payloadlendata_frame]; fwrite(com_obj, cmd_frame);$

response$_init = fread(com_obj, 9,' uint8'); if length(response_init) < 9 disp('Incompletedatareceived.'); end$

pause(0.075);

com$_obj.Baudrate = 2E6;$

value = uint32(1); data$_{frame} = typecast(value,' uint8'); cmd_frame = ['RSPI' payloadlendata_frame]; fwrite(com_obj, cmd_$

response$_init = fread(com_obj, 9,' uint8'); if response_init(9)\ = 0 disp('Error : Commandnotacknowledged'); end$

value = uint32(1); data$_{frame} = typecast(value,' uint8'); cmd_frame = ['RRAI' payloadlendata_frame]; fwrite(com_obj, cmd_$

response$_init = fread(com_obj, 9,' uint8'); if response_init(9)\ = 0 disp('Error : Commandnotacknowledged'); end$

fig = figure('NumberTitle', 'off', 'Name', 'Read out TDAT', 'Units', 'Normalized', 'OuterPosition', [0, 0.04, 1, 0.96]);

bgAxes = axes('Position', [0 0 1 1], 'Units', 'normalized'); uistack(bgAxes, 'bottom'); backgroundImage = imread('E:folderdesogn4.p imshow(backgroundImage, 'Parent', bgAxes); set(bgAxes, 'HandleVisibility', 'off', 'Visible', 'off');

ax1 = subplot(1,2,1); ax2 = subplot(1,2,2);

hold(ax1, 'on'); hold(ax2, 'on');

plot$_handle1 = plot(ax1, NaN, NaN,' o',' MarkerSize', 15,' MarkerFaceColor',' b'); plot_handle2 = plot(ax2, NaN, NaN,$

axis(ax1, [-25 25 0 10]); set(ax1, 'XColor', 'red', 'YColor', 'red');

grid(ax1, 'on');

title(ax2, 'Distance / Distance', 'Color', 'red'); xlabel(ax2, 'Distance [m]', 'Color', 'red'); ylabel(ax2, 'Distance [m]', 'Color', 'red');

axis(ax2, [-5 5 0 10]); set(ax2, 'XColor', 'red', 'YColor', 'red');

grid(ax2, 'on');

uicontrol('Style', 'pushbutton', 'String', 'Start', 'Position', [10 10 80 30], 'Callback', @startButtonCallback);

uicontrol('Style', 'pushbutton', 'String', 'Stop', 'Position', [100 10 80 30], 'Callback', @stopButtonCallback);

isRunning = false;

function startButtonCallback(,) if isRunning isRunning = true; while isRunning TDAT = uint32(8); data$_{frame} =$ $typecast(TDAT,' uint8'); cmd_frame = ['GNFD' payloadlendata_frame]; fwrite(com_obj, cmd_frame);$

resp$_frame = fread(com_obj, 9,' uint8'); if resp_frame(9)\ = 0 disp('Error : Commandnotacknowledged'); end$

resp$_frame = fread(com_obj, 8,' uint8');$

target$_detected = false; if resp_frame(5) > 1 target_detected = true; TDAT_Distance = fread(com_obj, 1,' uint16'); TDAT_Sp$
$fread(com_obj, 1,' int16')/100; TDAT_Angle = deg2rad(fread(com_obj, 1,' int16')/100); TDAT_Magnitude = fread(com_obj, 1,'$

distance$_x = -(TDAT_Distance * sin(TDAT_Angle)); distance_y = TDAT_Distance * cos(TDAT_Angle); x =$
$TDAT_Distance/100; disp(x)s = speed1(x);$

set(plot$_handle1,' XData', TDAT_Speed,' YData', x); set(plot_handle2,' XData', distance_x/100,' YData', distance_y/100);$

delete(findall(fig,'Type','annotation')); annotation('textbox', [0.15, 0.6, 0.3, 0.3], 'String', sprintf('Distance of Vehicle at front: annotation('textbox', [0.15, 0.7, 0.3, 0.3], 'String', sprintf('Vehicle current speed:

end

drawnow; end end end

function stopButtonCallback(,) if isRunning isRunning = false; end end end

A.6 Code for Scenario 06:Arduino code

include ¡LiquidCrystal.h¿

// Initialize the LCD LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

// Define the pins for the ultrasonic sensor define TRIG$_{P}IN9defineECHO_{P}IN10$

// Define the pins for the LEDs and Buttons define WARNING$_{L}ED_{P}IN13defineAUTO_{M}ODE_{L}ED_{P}IN1//UsePin1fortheM$

// Variables for button states and speed bool manualMode = false; float manualSpeed = 0;

void setup() lcd.begin(16, 2); pinMode(TRIG$_{P}IN, OUTPUT); pinMode(ECHO_{P}IN, INPUT); pinMode(WARNING_{L}ED$

void loop() float distance = measureDistance(); float speed = manualMode ? manualSpeed : refinedSpeed(distance);

// Turn on the mode indicator LED if in automatic mode, off in manual mode digitalWrite(AUTO$_{M}ODE_{L}ED_{P}IN, !manualMode);$

// Check for mode button press if (digitalRead(BUTTON$_{M}ODE_{P}IN) == LOW)//TogglemodemanualMode = !manualMode;$

// In manual mode, check for speed adjustment button presses if (manualMode) if (digitalRead(BUTTON$_{I}NCREASE_{P}IN) ==$

$LOW)manualSpeed+ = 1; //Incrementspeeddelay(200); //Debouncedelayelseif(digitalRead(BUTTON_{D}ECREASE_{P}IN$

$LOW)manualSpeed- = 1; //Decrementspeeddelay(200); //DebouncedelaymanualSpeed = constrain(manualSpeed, 0, 150$

// Update LCD with distance and speed lcd.clear(); lcd.setCursor(0, 0); lcd.print(”Distance: ”); lcd.print(distance); lcd.print(” cm”); lcd.setCursor(0, 1); lcd.print(”Speed: ”); lcd.print(speed); lcd.print(manualMode ? ” man” : ” auto”);

// Blink the warning LED if distance is between 0 and 10 cm blinkWarningLED(distance);

delay(200); // Update rate delay

void blinkWarningLED(float distance) if (distance ¿= 0 distance ¡ 10) digitalWrite(WARNING$_{L}ED_{P}IN, HIGH); delay(250); digi$

float measureDistance() digitalWrite(TRIG$_{P}IN, LOW); delayMicroseconds(2); digitalWrite(TRIG_{P}IN, HIGH); delayMic$

long duration = pulseIn(ECHO$_{P}IN, HIGH); floatdistance = duration*0.034/2; //Speedofsoundwavedividedby2returndis$

float refinedSpeed(float x) float s = 0; if (x ¿= 0 x ¡ 10) s = 0; else if (x ¿= 10 x ¡ 15) // Speed increases from 0 to 10 between distances 10 and 15 s = 2 * (x - 10); else if (x ¿= 15 x ¡ 20) // Speed increases from 10 to 30 between distances 15 and 20 s = 10 + 5 * (x - 15); else if (x ¿= 20 x ¡ 25) // Speed increases from 30 to 50 between distances 20 and 25 s = 30 + 5 * (x - 20); else if (x ¿= 25 x ¡ 30) // Speed increases from 50 to 70 between distances 25 and 30 s = 50 + 5 * (x - 25); else if (x ¿= 30) // Speed is constant 70 when distance is greater than or equal to 30 s = 70; return s;

A.7 Code for Scenario 07: Graphical User Interface with Sound Implementation

import tkinter as tk from tkinter import ttk from tkinter import font as tkFont from matplotlib.backends.backend$_tkaggimportFigureCanv$

Function to calculate speed based on distance def refinedSpeed(x): if x ¿= 0 and x ¡ 10: s = 0 elif x ¿= 10 and x ¡ 15: s = 2 * (x - 10) elif x ¿= 15 and x ¡ 20: s = 10 + 5 * (x - 15) elif x ¿= 20 and x ¡ 25: s = 30 + 5 * (x - 20) elif x ¿= 25 and x ¡ 30: s = 50 + 5 * (x - 25) elif x ¿= 30: s = 70 return s

Enhanced plotting function with styling, now accepts max$_distancedefplot_speed_distance(max_distance = 35) : distances = range(int(max_distance)+1)Dynamicrangebasedonmax_distancespeeds = [refinedSpeed(x)forxindistances]f$ $Figure(figsize = (6, 4), dpi = 100)plot = fig.add_subplot(1, 1, 1)plot.plot(distances, speeds, label = "Speedvs.Distance", colo$ $dodgerblue', marker =' o', linestyle =' -', markersize = 4)plot.set_title("Speed-DistanceRelationship", fontsize = 14, color =' darkblue')plot.set_xlabel('Distance(cm)', fontsize = 10, color =' darkgreen')plot.set_ylabel('Speed(km/h)', font$ $10, color =' darkgreen')plot.grid(True, which =' both', linestyle =' --', linewidth = 0.5)plot.legend()returnfig, plot$

def listen$_for_distance() : recognizer = sr.Recognizer()withsr.Microphone()assource : speed_var.set("Listening...")audio$ $recognizer.listen(source)try : text = recognizer.recognize_google(audio_data)distance = float(text)update_plot_and_speed(dist$ $speed_var.set("CouldnotDistance")exceptsr.RequestErrorase : speed_var.set(f"Couldnotrequestresults; e")exceptValueEr$ $speed_var.set("PleasespeakaDistance.")$

def manual$_input() : try : distance = float(distance_entry.get())update_plot_and_speed(distance)exceptValueError : speed_var.set("PleaseenteraDistance.")$

def update$_plot_and_speed(distance) : globalfig, ax, canvass = refinedSpeed(distance)speed_var.set(f"Speed : skm/h")canvas.get_tk_widget().destroy()Removetheoldwidgetfig, ax = plot_speed_distance(distance)Redrawtheplotwithnewm$ $FigureCanvasTkAgg(fig, master = root)Recreatethecanvascanvas.draw()widget = canvas.get_tk_widget()widget.configur$ $white', relief =' sunken', bd = 2)widget.pack(side = tk.TOP, fill = tk.BOTH, expand = True, padx = 20, pady = 10)ax.scatter([distance], [s], color =' red', s = 50, zorder = 5)Markthequeriedpoint$

Create the main window root = tk.Tk() root.title("Speed-Distance GUI") root.configure(bg='lightgray')

Styling fontStyle = tkFont.Font(family="Lucida Grande", size=12) root.option$_add("*TButton*Font", fontStyle)$

Initial plot setup fig, ax = plot$_speed_distance()canvas = FigureCanvasTkAgg(fig, master = root)canvas.draw()widget = canvas.get_tk_widget()widget.configure(bg =' white', relief =' sunken', bd = 2)widget.pack(side = tk.TOP, fill = tk.BOTH, expand = True, padx = 20, pady = 10)$

Input Frame for better organization input$_frame = tk.Frame(root, bg =' lightgray', pady = 6)input_frame.pack(side = tk.TOP, fill = tk.X, padx = 20)$

Entry for manual distance input tk.Label(input$_frame, text = "EnterDistance : ", bg =' lightgray', fg =' black', font = fontStyle).pack(side = tk.LEFT)distance_entry = tk.Entry(input_frame, font = fontStyle)distance_entry.p$ $tk.LEFT, padx = 10)$

Button for manual distance query query$_button = tk.Button(input_frame, text = "QuerySpeed", command =$

$manual_input, bg =' steelblue', fg =' white', font = fontStyle)query_button.pack(side = tk.LEFT, padx = 10)$

Button to listen for distance listen$_button = tk.Button(input_frame, text = "SpeakDistance", command = listen_for_distance, bg =' steelblue', fg =' white', font = fontStyle)listen_button.pack(side = tk.LEFT, padx = 10)$

Variable and label to display speed speed$_var = tk.StringVar()speed_label = tk.Label(root, textvariable = speed_var, bg =' lightgray', fg =' darkred', font = fontStyle)speed_label.pack(side = tk.TOP, pady = 10)$

root.mainloop()

A.8 Code for Scenario 07: Graphical User Interface-II

import tkinter as tk from tkinter import ttk from tkinter import font as tkFont import matplotlib.pyplot as plt from matplotlib.backends.backend$_t kaggimportFigureCanvasTkAggfrommatplotlib.figureimportFigureimportspeech_r ecognition$

Function to calculate speed based on distance def refinedSpeed(x): if x ¿= 0 and x ¡ 10: s = 0 elif x ¿= 10 and x ¡ 15: s = 2 * (x - 10) elif x ¿= 15 and x ¡ 20: s = 10 + 5 * (x - 15) elif x ¿= 20 and x ¡ 25: s = 30 + 5 * (x - 20) elif x ¿= 25 and x ¡ 30: s = 50 + 5 * (x - 25) elif x ¿= 30: s = 70 return s

Enhanced plotting function with styling def plot$_s peed_d istance() : distances = range(35) speeds = [refinedSpeed(x) forxindis$ $Figure(figsize = (6, 4), dpi = 100) plot = fig.add_s ubplot(1, 1, 1) plot.plot(distances, speeds, label = "Speedvs.Distance", col$ $dodgerblue', marker =' o', linestyle =' -', markersize = 4) plot.set_t itle("Speed-DistanceRelationship", fontsize =$ $14, color =' darkblue') plot.set_x label('Distance', fontsize = 10, color =' darkgreen') plot.set_y label('Speed', fontsize =$ $10, color =' darkgreen') plot.grid(True, which =' both', linestyle =' --', linewidth = 0.5) plot.legend() returnfig$

def listen$_f or_d istance() : recognizer = sr.Recognizer() withsr.Microphone() assource : speed_v ar.set("Listening...") audio$ $recognizer.listen(source) try : text = recognizer.recognize_g oogle(audio_d ata) distance = float(text) query_s peed(distance) ex$ $speed_v ar.set("Couldnotunderstandaudio") exceptsr.RequestErrorase : speed_v ar.set(f"Couldnotrequestresults; e") exceptV$ $speed_v ar.set("Pleasespeakavalidnumber.")$

def query$_s peed(distance) : try : s = refinedSpeed(distance) speed_v ar.set(f"Speed : sunits") ax.clear() Clearpreviousplot$ $plot_s peed_d istance() Redrawtheplotax.scatter([distance], [s], color =' red', s = 50, zorder = 5) Markthequeriedpointcanvas.d$ $speed_v ar.set("Pleaseenteravalidnumber.")$

Create the main window root = tk.Tk() root.title("Speed-Distance GUI") root.configure(bg='lightgray')

Styling fontStyle = tkFont.Font(family="Lucida Grande", size=12) root.option$_a dd("*TButton*Font", fontStyle)$

Create and display the initial plot fig = plot$_s peed_d istance() ax = fig.axes[0] canvas = FigureCanvasTkAgg(fig, master =$ $root) Atk.DrawingArea.canvas.draw() widget = canvas.get_t k_w idget() widget.configure(bg =' white', relief ='$ $sunken', bd = 2) widget.pack(side = tk.TOP, fill = tk.BOTH, expand = True, padx = 20, pady = 10)$

Input Frame for better organization input$_f rame = tk.Frame(root, bg =' lightgray', pady = 6) input_f rame.pack(side =$ $tk.TOP, fill = tk.X, padx = 20)$

Button to listen for distance listen$_b utton = tk.Button(input_f rame, text = "SpeakDistance", command =$ $listen_f or_d istance, bg =' steelblue', fg =' white', font = fontStyle) listen_b utton.pack(side = tk.LEFT, padx =$ $10)$

Variable and label to display speed speed$_v ar = tk.StringVar() speed_l abel = tk.Label(root, textvariable =$ $speed_v ar, bg =' lightgray', fg =' darkred', font = fontStyle) speed_l abel.pack(side = tk.TOP, pady = 10)$

root.mainloop()

A.9 Code for Scenario 07: Graphical User Interface

import serial import time import matplotlib.pyplot as plt import numpy as np from matplotlib.patches import Wedge import matplotlib.patheffects as pe

Set the style for the plot plt.style.use('seaborn-dark-palette')

Function to calculate speed based on distance def calculate$_{speed(distance) : if 0 <= distance < 10 : return 0 elif 10 <= distance < 15 : return 2 * (distance - 10) elif 15 <= distance < 20 : return 10 + 5 * (distance - 15) elif 20 <= distance < 25 : return 30 + 5 * (distance - 20) elif 25 <= distance < 30 : return 50 + 5 * (distance - 25) elif distance >= 30 : return 70}$

Function to plot the theoretical speed profile with enhanced visual appearance def plot$_{t}heoretical_{s}peed_{p}rofile(ax) : distance_{r}ange = np.linspace(0, 30, 300) theoretical_{s}peeds = [calculate_{s}peed(d) for d in distance_{r}ange] ax.plot(distance_{r}ange,$ $Theoretical Speed', color =' midnightblue', linewidth = 3, linestyle =' -', alpha = 0.9, marker = None) ax.fill_{b}etween(dist$ $skyblue', alpha = 0.5) Gradient fill ax.set_{t}itle('Speed-Distance Relationship') ax.set_{x}label('Distance (cm)') ax.set_{y}label('Sp$

Function to draw the speedometer gauge and display distance def draw$_{s}peedometer(ax_{s}peedo, speed, distance, max_{s}peed = 70) : ax_{s}peedo.clear() ax_{s}peedo.add_{p}atch(Wedge((0.5, 0.5), 0.4, -90, -90 + (speed/max_{s}peed * 180), color =' red')) Display speed with shadow effect speed_{t}ext = ax_{s}peedo.text(0.5, 0.5, f' speed : .0f km/h', horizontalalignment =' center', fontsize = 14, color =' black', transform = ax_{s}peedo.transAxes) speed_{t}ext.set_{p}ath_{e}ffects([pe.withStroke(linewid$ $2, foreground = "white")]) Display distance with shadow effect distance_{t}ext = ax_{s}peedo.text(0.5, 0.2, f' distance : .0f cm', ho$ $center', fontsize = 12, color =' black', transform = ax_{s}peedo.transAxes) distance_{t}ext.set_{p}ath_{e}ffects([pe.withStroke(line$ $2, foreground = "white")]) ax_{s}peedo.axis('off')$

Establish a serial connection to the Arduino ser = serial.Serial('COM9', 9600, timeout=1) time.sleep(2)

Prepare the plot for real-time updating plt.ion() fig, ax = plt.subplots(figsize=(12, 6))

Plot the theoretical speed profile plot$_{t}heoretical_{s}peed_{p}rofile(ax)$

Initialize the speedometer axis ax$_{s}peedo = fig.add_{a}xes([0.75, 0.15, 0.2, 0.2], polar = True)$

Initialize the plot for the current measurement point current$_{p}oint, = ax.plot([], [],' ro', markersize = 10, markeredgecolor =' black', markeredgewidth = 1)$

Add text for displaying the current distance and speed info$_{t}ext = ax.text(0.05, 0.95,'', transform = ax.transAxes, fontsize = 12, verticalalignment =' top', color =' black')$

Read data from the serial port and update the plot in real-time print("Collecting data. Close the plot window to stop.") while True: if ser.in$_{w}aiting > 0 : data_{l}ine = ser.readline().decode('utf - 8').rstrip() try : distance = float(data_{l}ine) speed = calculate_{s}peed(distance) current_{p}oint.set_{d}ata(distance, speed) info_{t}ext.set_{t}ext(f' Distance : distance : .2f cm : speed : .2f km/h') draw_{s}peedometer(ax_{s}peedo, speed, distance) plt.pause(0.01) except ValueError : print(f" Received invalid data : data_{l}ine")$

ser.close() plt.ioff() plt.show()

Bibliography

[1] WHO, *World report on road traffic injury prevention*, 2010.

[2] C. Lundquist, *Automotive Sensor Fusion for Situation Awareness*, Thesis for Degree of Licentiate, Linköping University, 2009.

[3] Ibid.

[4] Edmunds, *Traction Control*, Online available at: `http://www.edmunds.com/car-safety/traction-control.html`

[5] Final Rule 49 CFR Parts 571 and 585 - [Docket No. NHTSA–2007-27662].

[6] J.D. Woll, ”VORAD collision warning Radar,” IEEE National Radar Conference - Proc. 1995. IEEE, Piscataway, NJ, USA, 95CH3571-0. pp 369-372.

[7] T.O. Grosch, W. Klimkiewicz, P. Moosbrugger, L. Carpenter, ”24-GHz frequency modulated/continuous wave automotive radar designed for collision warning,” Proc. of SPIE - The International Society for Optical Engineering v 2344 1995- Society of Photo-Optical Instrumentation Engineers, Bellingham, WA, USA. pp 146-158.

[8] D. Esteve, P.A. Rolland, J.J. Simonne, G. Vialaret, ”Prometheus-prochip status of sensor technology applied to automotive collision avoidance,” Proc. of SPIE - The International Society for Optical Engineering, v 2470 1995- Society of Photo-Optical Instrumentation Engineers, Bellingham, WA, USA. pp 386-395.

[9] M. Hischke, ”Collision warning radar interference,” Intelligent Vehicles Symposium, Proc., 1995. IEEE, Piscataway, NJ, USA, 95TH8132. pp 13-18.

[10] S.K. Kenue, ”Selection of range and azimuth angle parameters for a forward looking collision warning radar sensor,” Intelligent Vehicles Symposium, Proceedings 1995. IEEE, Piscataway, NJ, USA, 95TH8132. pp 494-499.

[11] T. Grosch, ”Radar sensors for automotive warning and avoidance,” Proc. of SPIE - the Intern Soc for Optical Engineering, v 2463, 1995, Society of Photo-Optical Instrumentation Engineers, Bellingham, WA, USA, pp 239-247.

[12] J. Li, R.G. Bosisio, K. Wu, ”Collision Avoidance Radar Using Six-Port Phase/Frequency Discriminator (SPFD),” IEEE MTT-S Digest, pp. 1553-1556.

[13] A. K. Patterson, (1998) *Intelligent Cruise Control System Impact Analysis*, Thesis of the Virginia Polytechnic Institute and State University, August 6.

[14] J. Wu, M. McDonald, M. Brackstone, 1998. A fuzzy logic microscopic simulation model for interurban ATT assessment. In: Proceedings of the 10th European Simulation Symposium and Exhibition. Society for Computer Simulation International ISBN 1-56555-154-0, Nottingham, UK.

[15] R. Holve, P. Protzel, J. Bernasch, K. Naab, (1995a) *Adaptive Fuzzy Control for Driver Assistance in Car-Following*, Proceedings of the 3rd European Congress on Intelligent Techniques and Soft Computing - EUFIT '95. Aachen, Germany, Aug., pp. 1149 – 1153.

[16] R. Holve, P. Protzel, K. Naab, (1995b). Generating Fuzzy Rules for the Acceleration Control of an Adaptive Cruise.

[17] V. Milanes, J. Villagra, J. Godoy, and C. Gonzalez, "Comparing fuzzy and intelligent PI controllers in stop-and-go maneuvers," IEEE Trans. Control Syst. Tech., vol. 20, no. 3, pp. 770-778, 2012.

[18] B. Ganji, A. Z. Kouzani, S. Y. Khoo, and S. Z. Mojtaba, "Adaptive cruise control of a HEV using sliding mode control," Expert Syst. with Appli., vol. 41, pp. 607-615, 2014.

[19] W. U. N. Fernando, and S. Kumarawadu, "Discrete-time neuroadaptive control using dynamic state feedback with application to vehicle motion control for intelligent vehicle highway systems," IET Control Theory and Appli., vol. 4, no. 8, pp. 1465-1477, 2010.

[20] Z. Liang, Z. Ren, and X. Shao, "Decoupling trajectory tracking for gliding reentry vehicles," IEEE/CAA J. Automayica Sinica, vol. 2, no. 1, pp. 115-120, 2015.

[21] C. M. Lin, and C. H. Chen, "Car-following control using recurrent cerebellar model articulation controller," IEEE Trans. Veh. Tech., vol. 56, no. 6, pp. 3660-3673, 2007.

[22] M. A. Khalaf, and F. L. Lewis, "Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach," Automatica., vol. 41, no. 1, pp. 779-791, 2005.

[23] J. Zhang, H. Zhang, Y. Luo, and H. Liang, "Nearly optimal Control scheme using adaptive dynamic programming based on generalized fuzzy hyperbolic model," Acta Automatica Sinica., vol. 39, no. 2, pp. 142-148, 2013.

[24] H. Zhang, L. Cui, and Y. Luo, "Near-optimal control for nonzero-sum differential games of continuous-time nonlinear systems using single-network ADP," IEEE Trans. Cybern., vol. 43, no. 1, pp. 206-216, 2013.

[25] V. Bageshwar, W. Garrard, and R. Rajamani, "Model predictive control of transitional maneuvers for adaptive cruise control vehicles," IEEE Trans. Veh. Technol., vol. 53, no. 5, pp. 365-374, 2004.

[26] T. Takagi, and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," IEEE Trans. Syst., Man., Cybern., vol.SMC-15, pp. 116-132, 1985.

[27] K. Mehran, "Takagi-Sugeno fuzzy modeling for process control", Newcastle University, 2008.

[28] T. C. Yang, "Networked control system: A brief survey," in IEE Proc. Control Theory Appl., vol. 153, no. 4, pp. 403-412, 2006.

[29] W. P. M. H. Heemels, and N. van de Wouw, “Stability and stabilization of networked control systems,” in Networked Control Systems, A. Bemporad, W. P. M. H. Heemels, and M. Johansson, Eds. Berlin, Germany: Springer-Verlag, pp. 203-253, 2010.

[30] R. Rajamani, and C. Zhu, “Semi-autonomous adaptive cruise control systems,” IEEE Trans. Veh. Technol., vol. 51, no. 5, pp. 1186-1192, 2002.

[31] M. A. S. Kamal, M. Mukai, J. Murata, and T. Kawabe, “Ecological vehicle control on roads with up-down slopes,” IEEE Trans. Intell. Transp. Syst., vol. 12, no. 3, pp. 783-793, 2011.

[32] Abdullah, R., Hussain, A., Warwick, K., Zayed, A. (2008). Autonomous intelligent cruise control using a novel multiple-controller framework incorporating fuzzy-logic-based switching and tuning. Neurocomputing, 71, 2727-2741. http://dx.doi.org/10.1016/j.neucom.2007.05.016

[33] Cahour, B., Forzy, J. F. (2009). Does projection into use improve trust and exploration? An example with a cruise control system. Safety Science, 47, 1260-1270.

[34] Dassonville, I., Jolly, D., Desodt, A. M. (1996). Trust between man and machine in a teleoperation system. Reliability Engineering and System Safety, 53, 319-325.

[35] Fastrez, P., Haue, J. B. (2008). Designing and evaluating driver support systems with the user in mind. International Journal of Human-Computer Studies, 66, 125-131. http://dx.doi.org/10.1016/j.ijhcs.2008.01.001

[36] Kesting, A., Treiber, M., Schonhof, M., Helbing, D. (2008). Adaptive cruise control design for active congestion avoidance. Transportation Research Part C, 16, 668-683. http://dx.doi.org/10.1016/j.trc.2007.12.004

[37] Lee, J. D., McGehee, D. V., Brown, T. L., Marshall, D. C. (2008). Rear end crash avoidance system (RECAS) algorithm and alerting strategies: effects of adaptive cruise control and alert modality on driver performance. Final report, DOT HS 810 981.

[38] Moon, S., Yi, K. (2008). Human driving data-based design of a vehicle adaptive cruise control algorithm. Vehicle System Dynamics, 8(4), 111-123. http://dx.doi.org/10.1080/00423110701576130

[39] Moon, S., Moon, I., Yi, K. (2009). Design, tuning, and evaluation of a full-range adaptive cruise control system with collision avoidance. Control Engineering Practice, 17, 442-455. http://dx.doi.org/10.1016/j.conengprac.2008.09.006

[40] Rajaonah, B., Tricot, N., Anceaux, F., Millot, P. (2008). The role of intervening variables in driver-ACC cooperation. International Journal of Human-Computer Studies, 66, 185-197. http://dx.doi.org/10.1016/j.ijhcs.2007.09.002

[41] Seiler, P., Song, B., Hedrick, J. K. (1998). Development of a collision avoidance system. SAE Technical Paper, 97-103. http://dx.doi.org/10.4271/980853

[42] Stabach, M. (2009). Factors correlated with traffic accidents as a basis for evaluating advanced drivers assistance systems. Accident Analysis and Prevention, 41, 1025-1033. http://dx.doi.org/10.1016/j.aap.2009.06.014

[43] Zadeh, L. (1992). Fuzzy sets, in Fuzzy models for pattern recognition: methods that search for structures in data. NY: IEEE Press.

[44] Zhang, J., Suto, K., Fujiwara, A. (2009). Effects of in-vehicle warning information on drivers' decelerating and accelerating behaviors near an arch-shaped intersection. Accident Analysis and Prevention, 41, 948-958. http://dx.doi.org/10.1016/j.aap.2009.05.010

[45] Zheng, P., McDonald, M. (2005). Manual vs. adaptive cruise control-can driver's expectation be matched? Transportation Research Part C, 13, 421-431.

[46] Zheng, P., McDonald, M. (2005). Manual vs. adaptive cruise control–Can driver's expectation be matched? Elsevier - Journal of Transportation Research Part C, Vol. 13 No.1, pp.421-431.

[47] Higashimata, A., Adachi, K., Hashizume, T., Tange, S. (2001). Design of a headway distance control system for ACC. Journal of Society of Automotive Engineering, Vol. 22 No. 1, pp. 15–22.

[48] LUO, L.-h., LIU, H., LI, P., WANG, H. (2010). Model predictive control for adaptive cruise control with multi-objectives: comfort, fuel-economy, safety and car-following. Journal of Zhejiang University-SCIENCE A (Applied Physics Engineering), Vol. 11 No.3, pp. 191-201.

[49] Pananurak, W., Thanok, S., Parnichkun, M. (2009). Design of a headway distance control system for ACC. In ROBIO 2009: Proceedinds of the IEEE International Conference on Robotics and Biomimetics, Bangkok, Thailand, pp. 1794-1799.

[50] Ioannou, P. A., Xu, Z., Eckert, S., Clemons, D., Sieja, T. (1993). Intelligent Cruise Control: Theory and Experiment. Proceedings of the 32nd IEEE Conference on Decision and Control, San Antonio, TX, pp.1885-1890.

[51] Osman, K., Rahmat, M. F., Ahmad, M. A. (2009). Modelling and Controller Design for a Cruise Control System. In CSPA 2009: Proceedings of the 5th International Colloquium on Signal Processing Its Applications, Kuala Lumpur, pp. 254-258.

[52] Ko, S.-J., Lee, J.-J. (2007). Fuzzy Logic Based Adaptive Cruise Control with Guaranteed String Stability. In ICCAS 2007: Proceedings of the International Conference on Control, Automation and Systems, COEX, Seoul, Korea, pp. 15-20.

[53] Sathiyan, P., Kumar, S. S., Selvakuar, A. I. (2013). Segmented Fuzzy logic controller for vehicle following with optimised rule base. Journal of Theoretical and Applied Information Technology, Vol. 56 No. 1, pp. 7-15.

[54] Zhang, J. Y., Li, Y. D. (2006). Application of Genetic Algorithm in Optimisation of Fuzzy Control Rules. In ISDA 2006: Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications, Vol. 1 pp. 529- 534.

[55] Kumar, M., Jangra, A., Diwaker, C. (2010). Genetic optimisation of fuzzy rule-base system. International Journal of Information Technology and Knowledge Management, Vol. 2 No. 2, pp. 287-29.

[56] Moon, S., I. Moon, and K. Yi. ”Design, tuning, and evaluation of a full-range adaptive cruise control system with collision avoidance.” Control Engineering Practice, Vol. 17, no. 4, pp. 442–55, Apr. 2009.

[57] Rajaonah, B., N. Tricot, F. Anceaux, and P. Millot. ”The role of intervening variables in driver-ACC cooperation.” Int. J. Hum.-Comput. Stud., Vol. 66, no. 3, pp. 185–97, Mar. 2008.

[58] Muller, R., and G. Nocker. ”Intelligent cruise control with fuzzy logic.” in Proceedings of the Intelligent Vehicles 92 Symposium, Detroit, MI, 1992, pp. 173–8.

[59] Rizvi, R., S. Kalra, C. Gosalia, and S. Rahnamayan. ”Fuzzy adaptive cruise control system with speed sign detection capability.” in Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Beijing, 2014, pp. 968–76.

[60] Astrom, K. J., and T. Hagglund. PID Controllers, 2nd ed. Research Triangle Park, NC: ISA, 1995, pp. 59–80.

[61] Johnson, M., and M. H. Moradi. PID Control. London: Springer-Verlang, 2005, pp. 1–106.

[62] Kaya, I., N. Tan, and D. P. Atherton. ”A refinement procedure for PID controller.” Electr. Eng., Vol. 88, no. 3, pp. 215–21, Mar. 2006.

[63] Kumar, V., and K. P. S. Rana. ”Some investigations on hybrid fuzzy IPD controllers for proportional and derivative kick suppression.” Int. J. Autom. Comput., Vol. 13, no. 5, pp. 516–28, Oct. 2016.

[64] Rajinikanth, V., and K. Latha. ”I-PD controller tuning for unstable system using bacterial foraging algorithm: A study based on various error criterion.” Appl. Comput. Intell. Soft Comput., Vol. 2012, Article ID 329389, pp. 1–10, Jan. 2012.

[65] Mochizuki, S., and H. Ichihara. ”I-PD controller design based on generalized KYP lemma for ball and plate system.” in Proceedings of the European Control Conference, Zurich, 2013, pp. 2855–60.

[66] Visioli, A. Practical PID Control. London: Springer-Verlag, 2006, pp. 19–34.

[67] Wade, H. L. Basic and Advanced Regulatory Control: System Design and Application. Research Triangle Park, NC: ISA, 2004, pp. 39–110.

[68] Selvaperumal, S., and C. Christober Asir Rajan. ”Investigation of fuzzy control based LCL resonant converter in RTOS environment.” J. Intell. Fuzzy Syst., Vol. 26, no. 2, pp. 913–24, Feb. 2014.

[69] Selvaperumal, S., C. Christober Asir Rajan, and S. Muralidharan. ”Stability and performance investigation of a fuzzy-controlled LCL resonant converter in an RTOS environment.” IEEE Trans. Power Electron., Vol. 28, no. 4, pp. 1817–32, Apr. 2013.

[70] Zhou, M., J. Sun, H. Gao, and X. Wang. ”Fuzzy PID control and simulation analysis of cruise control system.” in Information and Automation. Communications in Computer and Information Science, L. Qi, Ed. Berlin, Heidelberg: Springer, Vol. 86, 2011, pp. 289–95.

[71] Mann, G. K. I., B.-G. Hu, and R. G. Gosine. ”Analysis of direct action fuzzy PID controller structures.” IEEE Trans. Syst. Man Cybern. B, Vol. 29, no. 3, pp. 371–88, June 1999.

[72] Edgar, S., and F. Victor. ”Real-time underactuated robot swing-up via fuzzy PI + PD control.” J. Intell. Fuzzy Syst., Vol. 17, no. 1, pp. 1–13, Jan. 2006.

[73] Pan, Y., M. Joo Er, T. Sun, B. Xu, and H. Yu. ”Adaptive fuzzy PD control with stable H tracking guarantee.” Neurocomputing, Vol. 237, pp. 71–8, May 2017.

[74] Pan, Y., H. Yu, and M. Joo Er. ”Adaptive neural PD control with semiglobal asymptotic stabilization guarantee.” IEEE Trans. Neural Netw. Learn. Syst., Vol. 25, no. 12, pp. 2264–74, Dec. 2014.

[75] Pan, Y., H. Yu, and T. Sun. ”Global asymptotic stabilization using adaptive fuzzy PD control.” IEEE Trans. Cybern., Vol. 45, no. 3, pp. 574–82, Mar. 2015.

[76] Kim, B. J., and C. C. Chung. ”Design of fuzzy PD + I controller for tracking control.” in Proceedings of the American Control Conference (IEEE Cat. No.CH37301), Anchorage, AK, Vol. 3, 2002, pp. 2124–9.

[77] Bhushan, B., N. Jha, S. Devra, and S. Pillai. ”Performance analysis of PID and fuzzy PD+I controller on nonlinear systems.” in Proceedings of the IEEE International Advance Computing Conference, Gurgaon, 2014, pp. 1195–200.

[78] Qiu, C. ”A design of automobile cruise control system based on fuzzy PID.” in Proceedings of the International Conference on Information Science, Electronics and Electrical Engineering, Sapporo, 2014, pp. 450–2.

[79] Lee, J.-Y., H.-R. So, Y.-H. Lee, and S.-J. Oh. ”Fuzzy PD plus I controller of a CSTR for temperature control.” J. Korean Soc. Marine Eng., Vol. 39, no. 5, pp. 563–9, 2015.

[80] Prabhakar, G., P. Nedumal Pugazhenthi, and S. Selvaperumal. ”Implementation analysis of state space modelling and control of nonlinear process using PID algorithm in MATLAB and PROTEUS environment.” Appl. Mech. Mater., Vol. 573, pp. 297–303, June 2014.

[81] Dwyer, A. O. ”A summary of PI and PID controller tuning rules for processes with time delay. Part 1: PI controller tuning rules.” in Proceedings of the PID ’00: IFAC Workshop on Digital Control, Terrassa, 2000, pp. 175–80.

[82] Dwyer, A. O. ”A summary of PI and PID controller tuning rules for processes with time delay. Part 2: PID controller tuning rules.” in Proceedings of the PID ’00: IFAC Workshop on Digital Control, Terrassa, 2000, pp. 242–7.

[83] Nath, A., J. Samantaray, and S. Chaudhury. ”Magnetic ball levitation system control using sliding mode control and fuzzy PD+I control: A comparative study.” in Proceedings of the International Conference on Energy, Power and Environment: Towards Sustainable Growth (ICEPE), Shillong, 2015, pp. 1–5.

[84] Sartorius, R., J. de Jesus Moreno, P. Oscar, and R. A. Estela. ”A new approach for adjusting scale factor in fuzzy PD+I controllers with anti-windup.” J. Intell. Fuzzy Syst., Vol. 27, no. 5, pp. 2319–26, May 2014.

[85] Tang, K. S., K. F. Man, G. Chen, and S. Kwong. ”A GA-optimized fuzzy PD+I controller for nonlinear systems.” in Proceedings of the IECONO1: The 27th Annual Conference of the IEEE Industrial Electronics Society, Denver, CO, 2001, pp. 718–23.

[86] Khooban, and M. Hassan. ”Design an intelligent proportional-derivative (PD) feedback linearization control for nonholonomic-wheeled mobile robot.” J. Intell. Fuzzy Syst., Vol. 26, no. 4, pp. 1833–43, Apr. 2014.

[87] Perumal, L., and F. H. Nagi. ”Largest of maximum (LOM) method for switching fuzzy control system.” Australian J. Electr. Electron. Eng., Vol. 4, no. 2, pp. 167–78, Sep. 2015.