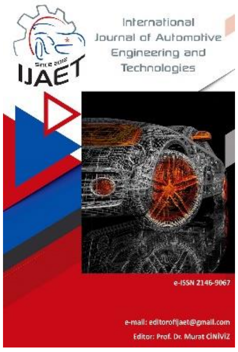
e-ISSN: 2146 - 9067

International Journal of Automotive Engineering and Technologies

journal homepage:

https://dergipark.org.tr/en/pub/ijaet



Original Research Article

A novel system architecture of intelligent adaptive cruise control for safety aspects

hosted by Turkish JournalPark ACADEMIC

Zeynep Musul[1, 3 *], Onur Cihan[2]

[1, 2] Marmara University, Engineering Faculty, Department of Electrical Electronics Engineering, 34854, İstanbul, Türkiye.
[3] ASML Netherlands, DR 5504, Veldhoven, The Netherlands.

ARTICLE INFO

Orcid Numbers

1. 0009-0005-5796-5633
2. 0000-0002-5729-2417

Doi: 10.18245/ijaet.1406829

* Corresponding author
zeynepmusul96@gmail.com

Received: Dec 19, 2023
Accepted: Jul 28, 2024

Published: 30 Sep 2024

Published by Editorial Board Members of IJAET

ABSTRACT

Following industrial and safety standards for autonomous vehicles, Adaptive Cruise Control (ACC) is a widely employed Advanced Driving Assistance System (ADAS) feature in modern vehicles. ACC currently facilitates speed control based on the driver's desired speed value. This study introduces a significant advancement: the Intelligent Adaptive Cruise Control (IACC) feature, accompanied by the development of a control system architecture poised to make noteworthy contributions in scientific, economic, and social dimensions through its integration into autonomous vehicles. The design incorporates crucial elements such as Traffic Sign and Limit Recognition (TSLR), ADAS features, and Global Positioning System (GPS) data, primarily enhancing driver safety through these supportive features. The main focus revolves around designing a system architecture that accommodates these new features to ensure safe driving. The creation of the IACC system architecture is approached using Model-Based System Engineering (MBSE). Through this MBSE methodology, system-level diagrams were crafted, and security considerations were systematically addressed. Several scenarios were devised to evaluate the contributions and were subsequently tested and analyzed. The architecture places particular emphasis on the security aspects of IACC. Leveraging the TSLR feature, the system interprets traffic signs and acquires speed limit data from external sources, preventing the vehicle's speed from exceeding the specified limit. The comparison between the set speed value and the speed limit ensures adherence to safety parameters.
In such scenarios, the system enhances driver support on winding roads by utilizing GPS data to recognize the vehicle in front. This approach significantly elevates the reliability of the IACC feature, particularly in terms of safety sensitivity, when compared to other adaptive cruise control concepts.

Keywords: Model Based System Engineering, Advanced Driving Assistance System, Adaptive Cruise Control

1. Introduction

This literature review encompasses three primary themes within the ACC concept: traffic speed limits, safety implications, and system architecture (such as hardware, control and software systems). This section will highlight certain studies conducted in these areas.

1.1 Background

In previous studies, adaptive speed control systems have been meticulously crafted using a variety of control algorithms [1]. These controllers predominantly include Proportional-Integral-Derivative (PID), Stanley, Pure Pursuit, Linear Quadratic Regulator (LQR), Linear Model Predictive (LMP), and Fuzzy Logic Controller. Model Predictive Control, while proactive in its behaviors, comes with a higher computational burden, but its advantages include a reduced following error compared to other controllers [2]. In one of the previous studies, the primary objective of the ACC is to formulate model equations and, consequently, to develop a suitable objective function based on system logins and states [3]. The Cooperative Adaptive Cruise Control (CACC) concept differs slightly from IACC, with its primary focus on time-gap. A CACC vehicle is recognized as a CACC sequence follower when it meets all three conditions: (1) the automation system is activated, and the CACC vehicle is in CACC operation; (2) the tool is in space editing mode; (3) the tool maintains the intra-sequence spacing as the desired time interval rather than the inter-sequence spacing [4]. Some novel studies focus on the human-machine interface (HMI), including displays, warning devices, operational switches, acceleration, and brake pedals, to improve user acceptance of ACC systems [5]. However, while it focuses on HMI, there is still a need to prepare architectural diagrams to enhance the sensor organization scheme for analyzing the impact of this study.

Low-level system studies in the field of ACC have been discussed. These studies cover aspects such as the control system, the time-gap status of the system, or enhancements to sensor details [6, 7]. It is noticeable that the safety aspect is not extensively addressed. Previous studies have provided various analysis methods for error situations. The aim is to integrate these methods into the system architecture, thus addressing the concept of functional safety [8]. It can be inferred that the system architectural design [9] is not approached using a model-based system engineering method, and adherence to standards is not emphasized. This study introduces an ACC system designed with data sourced from GPS and TSLR, another ADAS feature. The primary objective is to mitigate accidents on curved roads by leveraging GPS data [10]. Even when the driver-entered speed data diverges from the road's speed limit, a control mechanism prevents accidents. This safeguard is realized through an algorithm that compares TSLR feature data with input speed information, ensuring a responsive speed control system for both straight and curved roads.

The main contributions of this paper can be summarized as follows:

1. **Enhancing safety on curved roads:** The paper will contribute to reducing safety risks on curved roads by leveraging GPS data, thus improving the existing ACC, a vital ADAS feature [11].
2. **Data integration for security:** For security purposes, IACC is expected to strengthen overall security measures by actively sharing and processing traffic speed limit data from another ADAS feature when provided.
3. **Architectural advancements:** This study will contribute to the formation of a novel ACC concept by supporting the previously mentioned features in its architecture. This represents a holistic approach for improving adaptive cruise control functionality.

1.2. The IACC concept

1.2.1. The IACC - speed control

The operation of IACC in cruise control mode is elucidated in the following description. Within the framework of IACC, the speed to be regulated is determined by the value manually set by the driver, representing the targeted speed information.

Irrespective of the presence of a target vehicle, the vehicle adjusts its speed according to the present value. When no target vehicle is detected, the driver uses the preset speed as the speed for the ego vehicle. In Figure 1, v_{ego} represents the current speed of the vehicle. The driver establishes the desired IACC speed value, denoted as v_{set}.

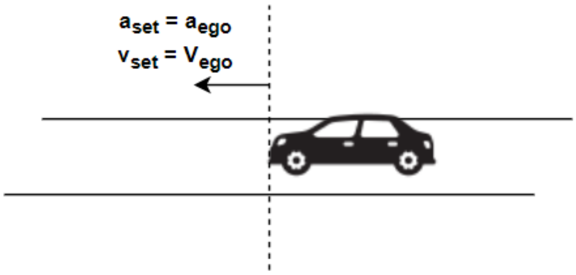


Figure 1: The IACC Feature - Speed Control State

1.2.2 The IACC - following control

In IACC tracking control algorithm, the ego vehicle maneuvers to uphold a specific distance from the vehicle in front. The transition to the subsequent control algorithm hinges on the determination of the target vehicle's presence. Should IACC detect a vehicle within a proximity closer than the preset following clearance (c), the system seamlessly transitions from adaptive cruise control to the following control mode. The velocity of the tracked path is denoted as v_{target}, and its acceleration is represented by a_{target}, as illustrated in Figure 2.

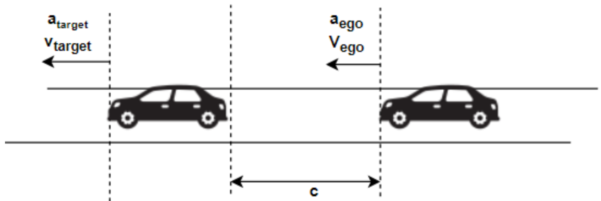


Figure 2: The IACC Feature - Following Control State

1.2 The IACC model-based system architecture

Model-based system architecture is an outcome of MBSE, steadily establishing itself as a mainstream approach in the field of systems engineering. The primary emphasis in systems engineering is directed towards identifying any overlooked communications between subsystems or systems. MBSE brings a systematic and insightful process to the study of systems engineering, manifesting in various methods. In essence, these methods involve creating architectures through models, defining architectural decisions, conducting analyses to support those decisions, and determining novel decisions for subsequent architectural processes [12]. Two fundamental modeling languages are prevalent in this context: SysML (System Modeling Language) and UML (Unified Modeling Language). SysML, tailored for system modeling, offers a reduced set of diagrams compared to UML. These diagrams find particular utility in the creation of requirements. On the other hand, UML, a more expansive modeling language, includes technical details for both software and hardware components, making it suitable for a broader range of applications. At the system level, both languages encompass three distinct diagram groups: requirements, behavior, and structure. Common diagrams shared by both languages include use case, activity, block definition, package, and state machine diagrams.

Earlier research into adaptive speed control systems has utilized a range of control algorithms, including PID, Stanley, Pure Pursuit, LQR, LMP, and Fuzzy Logic Controller. Comparative analyses of their performance have uncovered unique characteristics among these algorithms. Performance comparisons of these algorithms have been conducted, revealing distinctive traits. In summary, the Pure Pursuit controller demonstrates sensitivity in short-distance areas, while its following accuracy is less precise but more stable. The Stanley controller excels in accurate following on smooth paths but exhibits increased errors on curved roads. The Linear Quadratic Regulator control provides a linear solution, and Model Predictive Control exhibits proactive behaviors, albeit with a higher computational burden, leading to fewer following errors than other controllers [13].

These comparative evaluations consider aspects such as vehicle following distance, speed control outputs, stability, and errors. Additionally, error analysis studies have been conducted to enhance collision avoidance for this ADAS feature [14].

2. Materials and Methods

2.1 High-level system diagrams of the IACC using MBSE

2.1.1 The IACC system package diagram

This package is a top-level structure. Creating this structure preserves logical integrity. It makes the model more easily readable and understandable by the customer during the creation phase as given in Figure 3. The package structure can be seen as a kind of containment tree. System packages are categorized into verification, diagrams, requirements, logical diagrams, resources, and behavior analysis, as illustrated in Figure 3. These diagrams are generated during the initial stages of the project's architectural design, providing valuable assistance throughout the design process [15].

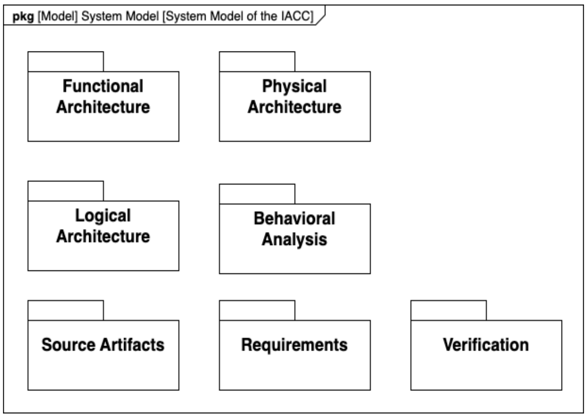


Figure 3: System Package Diagram

2.1.2. The IACC use case diagram

Although use case diagrams primarily comprise actors, systems, and usage patterns, their key objective is to depict the entire scenario simultaneously. Use case scenarios are crafted to delineate situations in which communication transpires between the driver and the vehicle as illustrated in Figure 4. The primary use cases of IACC are explicitly outlined. It is imperative to provide detailed descriptions that delve into a lower level when juxtaposed with block diagrams.

In the IACC use case diagram, components such as include and extend are employed to illustrate the system's operation based on the scenario's situational requirements.

2.1.3. The IACC feature interface diagram

The IACC feature engages with numerous vehicle subsystems and the environment, facilitating input reception and output transmission to relevant subsystems. Interactions are facilitated through interfaces, with a meticulous determination of these interactions and interfaces made during the creation of the system architecture. This diagram showcases the data transmission and reception with the driver, ego vehicle, environment, and traffic speed limit recognition ADAS feature as illustrated in Figure 5.

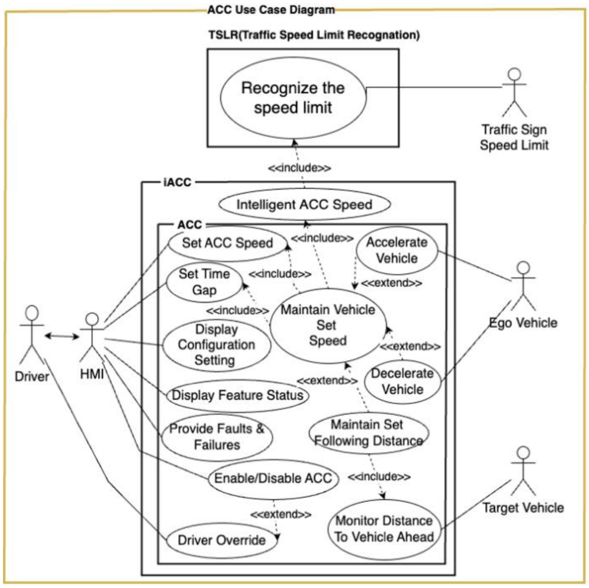


Figure 4: The IACC Use Case Diagram

The IACC interface diagram illustrates the interactions of the IACC feature with other environments, as depicted in Figure 5. The directions of the arrows are assigned based on the unit to which they convey information. In this context, the ego vehicle and the driver are treated as distinct environments, and IACC, the ego vehicle's internal unit, and external factors are represented within the diagram. The primary information flow is delineated through this diagram.

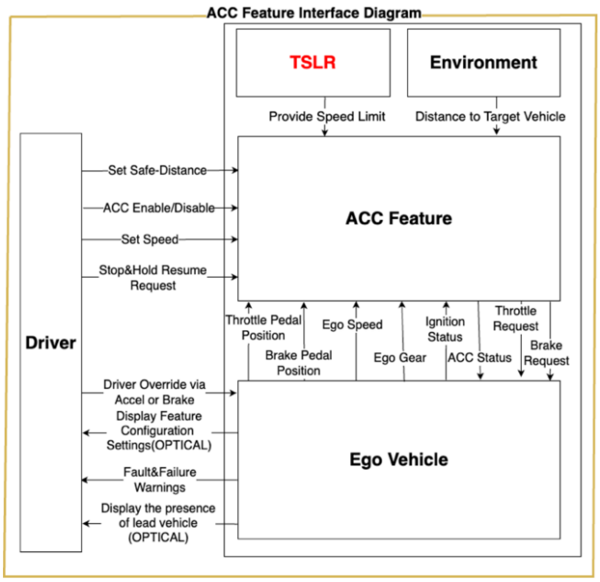


Figure 5: The IACC Feature Interface Diagram

2.1.4. The IACC feature context diagram

A block definition diagram, utilized within the context of the IACC feature, provides a high-level understanding of the IACC's environment. It encompasses the driver's vehicle, the system's environment, and key elements contributing inputs as illustrated in Figure 6. The diagram comprises two distinct sections: the system's environment and the system's features. The high-level actions or modules directly impacting the IACC feature are identified in the system's feature section. The IACC's outputs, represented as throttle or brake properties, are integral to the system. Blocks interacting externally with the system are depicted under the environment.

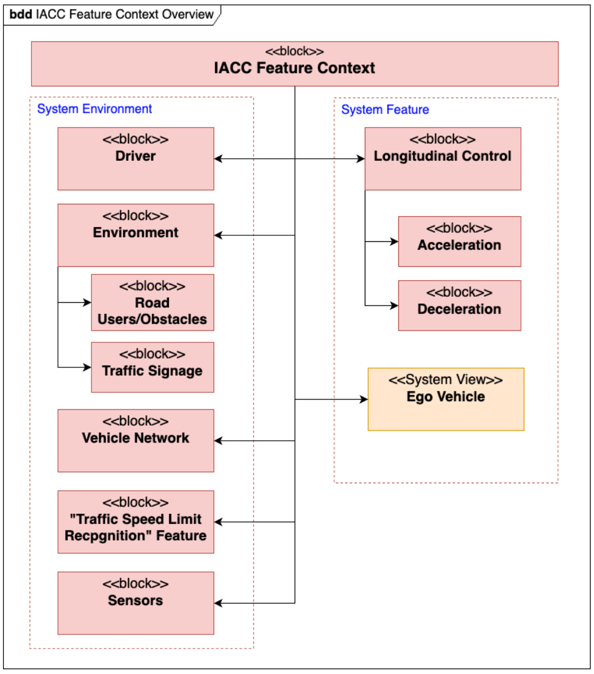


Figure 6: The IACC Feature Context Diagram

2.1.5. The IACC user states

The IACC user states can be found in Table 1.

2.1.6. The IACC block definition diagram

A block definition diagram is a comprehensive system diagram offering a high-level perspective on a system's architecture. Its primary purpose is to model the system without delving into specific requirements. Additionally, this diagram serves as a valuable precursor for constructing an internal block diagram. Positioned at the L0 (High Level) tier, the block definition diagram delineates the logical architecture of the IACC. Preceding the creation of the behavior diagram specifically, the activity diagram defines essential blocks, including environmental perception, the ego vehicle's state, communication with other features, HMI environment, vehicle movement control, error/fault management, and decision making and planning as illustrated in Figure 7.

Table 1: The IACC User State Overview

IACC User State	*Event*
OFF User State	HMI: ACC OFF
ON User State	NA
ACTIVE User State	NA
STANDBY User State	HMI provides ACC status as STANDBY
SPEED CONTROL User State	NA
Intelligent Speed Control User State	HMI: ACC Speed Control ACC maintains road speed limit from TSL Regulate speed if necessary for bends, junctions, intersections, traffic hazard etc.
Set Speed Control User State	HMI: ACC Speed Control ACC maintains the set speed selected by the driver
STOP HOLD User State	HMI: ACC Stop Hold Hold Vehicle Stationary
FOLLOWING CONTROL User State	HMI: ACC Following Control ACC mains the set following distance
SUPPRESS User State	Driver Override HMI: ACC Suppress

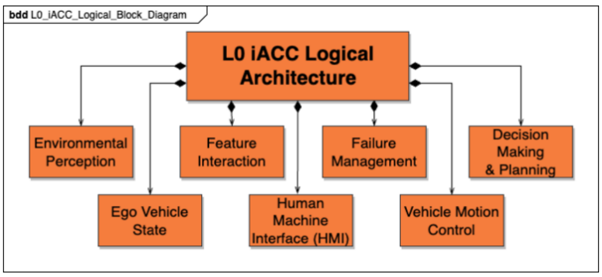


Figure 7: The IACC Logical Block Definition Diagram

2.2 Other adas features supporting the IACC

Numerous ADAS features are strategically designed to facilitate data exchange, enabling seamless control in autonomous vehicles. While this paper primarily focuses on the speed control feature within the IACC, it is essential to note its interaction with fundamental ADAS features. This data flow permits effective communication between control systems when required [16]. The ADAS features collaborating with IACC include Stop&Go, TSLR, Automatic Emergency Braking (AEB), and Lane Keeping Assistance System/Lane Centering Assistance (LKA/LCA).

2.3 Scenarios

This section establishes parameters and preconditions to configure test environment conditions. Eight distinct state scenarios are created for each condition, based on IACC User State Mode, Lead Vehicle Detection Status, Road Type, and Traffic Speed Limit as detailed in Table 2.

Table 2: IACC Simulation Scenarios and Dependencies

No	IACC User State	Lead Vehicle Status	Road Type	Traffic Speed Limit
1	Speed Control User State(v_d)	Not Detected	Straight	NA
2	Speed Control User State(v_d)	Not Detected	Curved	NA
3	Following Control (τ_{gap})	Detected	Straight	NA
4	Following Control (τ_{gap})	Detected	Curved	NA
5	Speed Control User State(v_d)	Not Detected	Straight	TSLR
6	Speed Control User State(v_d)	Not Detected	Curved	TSLR
7	Following Control (τ_{gap})	Detected	Straight	TSLR
8	Following Control (τ_{gap})	Detected	Curved	TSLR

Scenario – III: This scenario represents the situation in which the IACC feature detects a lead vehicle that is traveling on a straight road as illustrated in Figure 8. It has been observed that traffic speed limit information for the vehicle is not transmitted from the relevant ADAS feature. The IACC feature is in the following control system state due to lead vehicle detection: If the driver enters the desired time gap, the entered time gap is set to the current time gap since there is no traffic speed limit value.

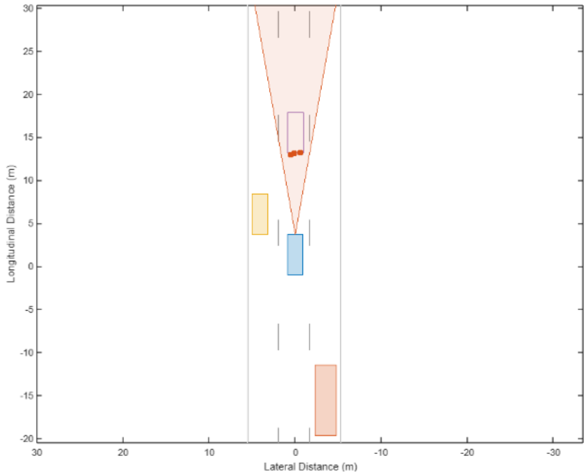


Figure 8: Ego Vehicle (IACC - Blue) Bird's Eye Plot: Straight Road & Detected Lead Vehicle

Test specifications should be given according to scenarios like current velocity, desired velocity, road type, and velocity changes. Elaborate details regarding the scenario can be found in Table 3.

Table 3: Test Specifications: Scenario – III

Parameters	Values
Desired Velocity (kph)	80 kph
Speed Limit (kph)	NA
Lead Vehicle Detection	YES
Lead Vehicle Speed (if applicable)	50 kph
Last Ego Vehicle Velocity(kph)	50 kph

In Scenario III, the behavior of the vehicle on a straight road is elucidated. Another aspect explored in Scenario IV involves examining and comparing the vehicle's response to the curved road type, presenting an additional challenge.

Scenario – IV: This scenario depicts a situation where the IACC feature identifies a lead vehicle and navigates a curved road as illustrated in Figure 9. Notably, it has been observed that the traffic speed limit information for the ego vehicle is not transmitted from the relevant ADAS feature. As the IACC identifies the vehicle in front, the system enters the following control state. If the driver specifies a desired time gap, the time gap is established at the entered value, given the absence of traffic speed limit information. In this scenario, when a lead vehicle is present on the curved road, the IACC processes the detection of this lead vehicle based on the information received from the GPS data.

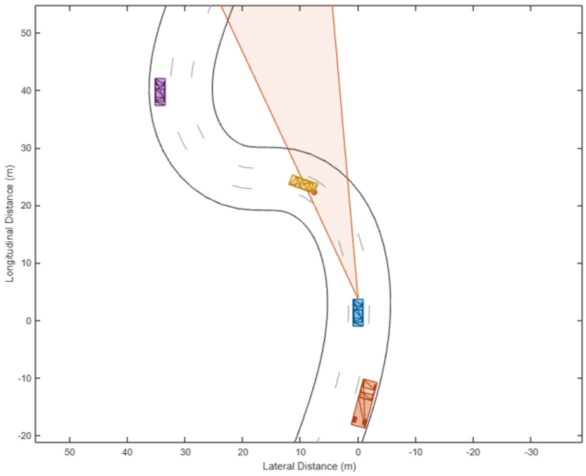


Figure 9: Ego Vehicle (IACC - Blue) Bird's Eye Plot: Curved Road & Detected Lead Vehicle

Test specifications should be defined by parameters such as current velocity, desired velocity, road type, and velocity changes. Elaborate details regarding the scenario can be found in Table 4.

Table 4: Test Specifications: Scenario – IV

Parameters	Values
Desired Velocity (kph)	90 kph
Speed Limit (kph)	NA
Lead Vehicle Detection	YES
Lead Vehicle Speed (if applicable)	50 kph
Last Ego Vehicle Velocity(kph)	50 kph

3. Results and Discussion

3.1 Safety activity discussion

In this section, we discuss the findings explored within the realm of ACC, focusing on safety aspects and the investigated scenarios. The pivotal factors influencing ACC's safety evaluation include:

- Unresponsive Driver Behavior
- Collision Avoidance
- System Failure

In the contemporary automotive landscape, the significance of functionalities offered by electrical and electronic systems in vehicles is steadily growing. Consequently, ensuring the safety of future autonomous and ADAS-equipped vehicles poses a formidable challenge. Designing ADAS devices necessitates a comprehensive understanding of all potential conditions. Road tests play a vital role in this context, creating real driving scenarios and simulating the vehicle's reactions within a controlled environment to facilitate verification.

The ISO 26262 safety lifecycle incorporates "Hazard Analysis and Risk Assessment" (HARA) activities. The primary objective is to determine an Automotive Safety Integrity Level (ASIL) and establish safety targets. Challenges with HARA include ensuring validity, repeatability, and reliability, as the assessment is inherently shaped by human knowledge and experience. Parameters outlined in the ISO 26262 standard, such as severity, controllability, and exposure levels, play a crucial role in the evaluation [17]. The HARA process requires designers to furnish documentation, comprising operational situations and a detailed status document outlining associated hazards and error/failure scenarios. This approach facilitates the identification of potential hazards more efficiently.

ACC primarily aims to safeguard lanes in the longitudinal plane [18]. From a safety perspective, three distinct driving tasks are considered: strategic, tactical, and operational. Strategic safety involves route planning to navigate areas with fewer obstacles or traffic. Tactical safety hinges on decisions related to parameters like ego vehicle speed and the distance between the lead vehicle and the ego vehicle [19]. Operational safety involves the sensing system's ability to perceive the environment and the functioning of ego vehicle actuators, including brake and accelerator pedals, steering, etc.

For IACC, a tactical safety process is essential. The following steps are undertaken by tactical safety:

1. Definition of Function
2. Detection of a Hazardous Situation
3. Objectification of HARA
4. Determination of ASIL
5. Control and decision-making for the determination of ASIL [20]

Verification efforts in this section are intricately linked to the subsequent phase, which involves writing requirements. The specification of functional safety requirements during the requirement writing phase necessitates the application of the aforementioned steps, accompanied by the creation of relevant scenarios.

3.2. Simulation results

This experiment focused on analyzing the safety performance of scenarios created for curved and straight roads, to enhance both driver and vehicle safety. The paper specifically tests a segment of the designed architecture to assess its impact through simulation. The evaluation of the vehicle's performance on safety considered the comparison between the required distance (time gap) and the actual distance between two vehicles. The experiment employed a MATLAB Simulink program, utilizing the Model Predictive Control (MPC) toolbox available in the MATLAB library to test the predetermined scenarios, as outlined in the previous section. There are eight distinct scenarios designed as high-level system test scenarios. Four of these scenarios involve processing TSLR external data as part of an in-device control module, while the remaining scenarios operate without TSLR input, another ADAS feature. IACC driving scenarios vary based on TSLR data, road shape, and lead vehicle status. The high-level system diagrams for IACC were designed and presented in the previous section (Section 2). Additionally, analyses were conducted using the existing MATLAB MPC toolbox with scenario details. However, due to the absence of TSLR data in the toolbox, safety evaluations were performed only when the vehicle in motion acted as the lead vehicle. The relevant scenarios (with a lead vehicle and without TSLR data) include scenario 3 and scenario 4. In terms of safety, the ego vehicle was expected to adjust its speed in all cases and maintain a safety distance equal to or greater than the minimum distance.

Scenario III: For scenario 3, the ego vehicle detected the lead vehicle in front using sensors (ultrasonic, RADAR, LiDAR, GPS). The initial speed of the ego vehicle was set at 100 kph, with the road type defined as straight. In line with the scenario's specifications, the speed of the ego vehicle was adjusted to 80 kph, considering the lead vehicle's speed set at 50 kph. The Vset value was established as 50 kph based on the time value, with the expected speed of the ego vehicle set at 50 kph. The speed change graph is depicted in Figure 10 The simulation time was set at 100 seconds, with a sample time value of 0.05 seconds.

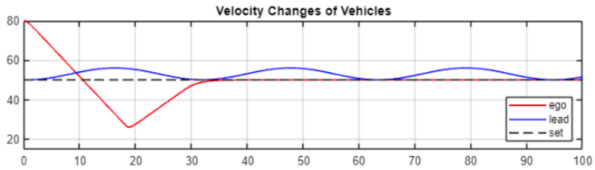


Figure 10: Scenario III – Velocity changes between ego & lead vehicle

Scenario IV: In scenario IV, the speed of the ego vehicle was set to 90 kph, while the lead vehicle's speed was set at 50 kph, considering the distance between them. The Vset value was configured at 50 kph based on the time value, with the expected speed of the ego vehicle also set at 50 kph. The speed change graph is illustrated in Figure 11. The simulation time for this scenario was determined as 400 seconds, with a sample time value of 0.05 seconds.

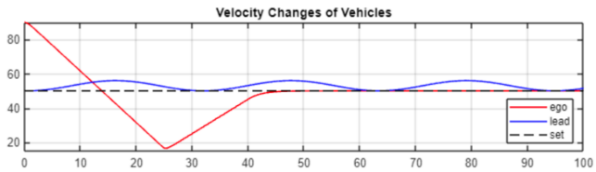


Figure 11: Scenario IV – Velocity changes between ego & lead vehicle

In conclusion, this paper thoroughly analyzed the main goals of IACC – safe distance, and speed control. Employing a MBSE approach, high-level system diagrams were created. The analysis indicated an improvement in driving safety on curved roads with the inclusion of the GPS sensor in the system. Furthermore, the TSLR input information, a key factor in speed control, was integrated into the architecture. The next study will employ a different autonomous vehicle simulation, incorporating additional features, to further analyze the effects of TSLR.

4. Conclusions

In this study, a high-level architectural design approach was employed, utilizing MBSE methods. System architecture diagrams, forming a top-level system structure, were meticulously crafted, incorporating both the existing and newly added features of IACC. These diagrams, detailed in Section 2, guided the design process. The subsequent project task involves the articulation of system requirements, marking the final stage in this phase. The study emphasizes adherence to security standards, underscoring the importance of a robust and standardized approach. The design work included the creation of high-level system diagrams, laying the foundation for the formulation of system requirements. Functional requirements are precisely outlined during the composition of system-level requirements, a crucial component of the project process. The identification of hardware and software components associated with these functional requirements initiates the low-level documentation and the hardware & software design process of the system [21]. Unlike previous studies, this iteration of the ACC feature prioritizes delivering high performance and enhancing driver safety on curved roads. The design incorporates a comprehensive analysis of terrains through GPS and the detection of lead vehicles in blind spots via radar, LiDAR, cameras, and similar sensors. Additionally, the design aims to elevate system awareness by considering traffic speed limits, emphasizing a holistic approach to safety [22]. The ACC concept contemplates scenarios where the driver activates the system without a lead vehicle in the vicinity. In such situations, the system autonomously processes information derived from the TSLR ADAS feature on highways or within city limits, eliminating dependence solely on the driver or lead vehicle. This strategy is aligned to enhanced safe driving support, even in scenarios devoid of a lead vehicle. Future studies are envisioned to incorporate real-time integration with TSLR input, enabling the simulation and performance analysis of data from this additional ADAS feature for architecture design refinement. The study also meticulously evaluated the latest published standards and regulations about ACC. As these standards are pivotal for Original Equipment Manufacturers (OEMs) to ensure ACC compliance with specific safety specifications, due diligence was exercised to ensure that the IACC adheres to these norms. Critical risk factors, such as safety-related collision avoidance, E/E (Electrical/Electronics) system failure, and driver behavior, were scrutinized and incorporated into the creation of system requirements. Anticipated challenges in real-time applications may include exceptions such as irregularities in road shapes, potential connectivity issues with GPS, and blind spots arising from support equipment like sensors. In future work, we plan to concentrate on utilizing appropriate speed limit data in the design and analysis.

CRediT authorship contribution statement

Zeynep Musul: Conceptualization, Investigation, Validation, Methodology, Writing-Original Draft. Onur Cihan: Supervision, Conceptualization, Investigation, Formal analysis.

Declaration of Competing Interest

The authors declare no conflict of interest.

Nomenclature

ACC : Adaptive Cruise Control
ADAS : Advanced Driving Assistance System
ASIL : Automative Safety Integrated Level
C : Clearance
CACC : Cooperative Adaptive Cruise Control
EuroNCAP : The European New Car Assessment Programme
HARA : Hazard Analysis and Risk Assessment
IACC : Intelligent Adaptive Cruise Control
ISO : International Organization for Standardization
KPI : Key Performance Indicators
Kph : Kilometer per hour
LQR : Linear Quadratic Regulator
LMP : Linear Model Predictive
GPS : Global Positioning System
NA : Not Applicable
NHTSA : National Highway Traffic Safety Administration
MBSE : Model-Based System Engineering
OEM : Original Equipment Manufacturer
PID : Proportional-Integral-

Derivative Controller

TSLR : Traffic Speed Limit Recognition

TTC : Time-to-Collision

τ_{gap} : Time Gap

5. References

1. Y. Menshenin, Y. Mordecai, E. F. Crawley and B. G. Cameron, "Model-Based System Architecting and Decision-Making," Handbook of Model-Based Systems Engineering, Cambridge, Moscow, Springer, p. 50, 2020.

2. M. Lu, K. Wevers and R. V. D. Heijden, Technical Feasibility of Advanced Driver Assistance Systems (ADAS) for Road Traffic Safety, Transportation Planning and Technology, vol. 28, no. 3, pp. 167-187, 2005.

3. M. Vajedi and L. N. Azad, Ecological Adaptive Cruise Controller for Plug-In Hybrid Electric Vehicles Using Nonlinear Model Predictive Control, IEEE Transactions on Intelligent Transportation Systems, vol 1, no. 17, pp. 113-122, 2016.

4. L. Xiao, M. Wang, W. Schakel and B. V. Arem, Unravelling effects of cooperative adaptive cruise control deactivation on traffic flow characteristics at merging bottlenecks, Transportation Research Part C: Emerging Technologies, no. 96, pp. 380-397, 2018.

5. L. Xiao and F. Gao, A comprehensive review of the development of adaptive cruise control systems, International Journal of Vehicle Mechanics and Mobility, vol. 48, no. 10, pp. 1167-1192, 2010.

6. E. Kural, T. Hacıbekir and B. Aksun Güvenç, State of Art of Adaptive Cruise Control and Stop & Go Systems, 1st AUTOCOM Workshop on Preventive and Active Safety Systems for Road Vehicles, Istanbul, 2020.

7. Front Radar Sensor, Bosch, 2023. [Online]: https://www.boschmobility.com/en/solutions/sensors/frontradar-sensor/. [Accessed: 18.09.2023].

8. E. NCAP, Assisted Driving Highway Assist Systems Test & Assessment Protocol, EURO NCAP, 2023.

9. J. Sini and M. Violante, A simulation-based methodology for aiding advanced driver assistance systems hazard analysis and risk assessment, Microelectronics Reliability Elsevier, 11 May 2020.

10. Bosch, Ultrasonic Sensor, [online]. Available: https://www.boschmobility.com/en/solutions/sensors/ultrasonicsensor/. [Accessed: 18.09.2023].

11. Z. Liu, Robust Target Recognition and Tracking of Self-Driving Cars with Radar and Camera Information Fusion Under Severe Weather Conditions, IEEE Transactions On Intelligent Transportation Systems, vol. 23, no. 7, pp. 6640 - 6653, 2022.

12. İ. Kılıç, A. Yazıcı, Ö. Yıldız, M. Özçelikors and A. Ondoğan, Intelligent Adaptive Cruise Control System Design and Implementation, 10th Annual System of Systems Engineering Conference, San Antonio, 2015.

13. J. Liu, Z. Yang, Z. Huang, W. Li, S. Dang and H. Li, Simulation Performance Evaluation of Pure Pursuit, Stanley, LQR, MPC Controller for Autonomous Vehicles, in the 2021 IEEE International Conference on Real-time Computing and Robotics, Xining, 2021.

14. C. Madariaga, A. Bashir and C. Swickline, Applying MBSE in Space Based Systems Development, 33rd Annual INCOSE International Symposium Hybrid Event, Honolulu, 2023.

15. M. T. Riaz, S. M. Aaqib, S. Ahmad, S. Amin, H. Ali, S. Husnain and S. Riaz, The Intelligent Transportation Systems with Advanced Technology of Sensor and Network, 2021 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube), Quetta, 2021.

16. D. Kaslow, B. Ayres, P. T. Cahill, L. Hart and R. Yntema, A Model-Based Systems Engineering (MBSE) approach for defining the behaviors of CubeSats, 2017 IEEE Aerospace Conference, Big Sky, MT, 2017.

17. J. Sini, M. Violante, V. Dodde, R. Gnaniah and L. Pecorella, A Novel Simulation-Based Approach for ISO 26262 Hazard Analysis and Risk Assessment, 25th International Symposium on On-Line Testing and Robust System Design (IOLTS 2019), p. 2, 2019.

18. C. Becker, A. Nasser, J. Brewer and J. A. N. T. S. C. Volpe, Hazard and Safety

Analysis of Automated Transit Bus Applications, Federal Transit Administration, Cambridge, 2020.

19. Y. G. Dantas, T. Munaro, C. Carlan, V. Nigam, S. Barner, S. Fan, A. Pretschner, U. Schöpp and S. Tverdyshev, A Model-based System Engineering Plugin for Safety Architecture Pattern Synthesis, 10th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2022), Online Streaming, 2022.

20. S. Khatstgir, H. Sivencrona, S. Birrell, G. Dhadyalla, P. Jennings and P. Billing, Introducing ASIL Inspired Dynamic Tactical Safety Decision Framework for Automated Vehicles, IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, 2017.

21. ISO 15622, Intelligent transport systems Adaptive cruise control systems Performance requirements and test procedures, International Standard ISO 15622, pp. iv-24, 2018.

22. L. Xiao, M. Wang, W. Schakel and B. van Arem, Unravelling effects of cooperative adaptive cruise control deactivation on traffic flow characteristics at merging bottlenecks, Transportation Research Part C: Emerging Technologies vol. 96, pp. 380-397, 2018.