# Analysis of a safe and reliable automated driving platform


EngD THESIS REPORT

# Analysis of a safe and reliable automated driving platform

ir. Raj Kumar Muniyandi

October/2023

Department of Mathematics & Computer Science

EngD AUTOMOTIVE SYSTEMS DESIGN
Track AUTOMOTIVE SYSTEMS DESIGN

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

# Analysis of a safe and reliable automated driving platform

Raj Kumar Muniyandi

October 2023

Eindhoven University of Technology
Stan Ackermans Institute - Automotive/Mechatronic Systems Design

EngD Report: 2023/097

*Confidentiality Status: Open Access*

**Partners**

Eindhoven University of Technology

**Steering Group**

PROJECT OWNER: Raj Kumar Muniyandi
PROJECT MANAGER: ir. Riske Meijer
PROJECT MENTORS: dr.ir. Tom van der Sande, ir. Jos den Ouden
PROJECT SUPERVISORS: dr.ir. Tom van der Sande, ir. Jos den Ouden

**Date** October 2023

Composition of the Thesis Evaluation Committee:

| | |
|---|---|
| Chair: | dr.ir. Igo Besselink |
| Members: | dr.ir. Tom van der Sande |
| | ir. Jos den Ouden |
| | dr. Mohsen Alirezei |
| | ir. Riske Meijer |

The design that is described in this report has been carried out in accordance with the rules of the TU/e Code of Scientific Conduct.

**Date** October, 2023

Contact address
Eindhoven University of Technology
Department of Mathematics and Computer Science
Automotive Systems Design
MF 5.075
P.O. Box 513
NL-5600 MB
Eindhoven, The Netherlands
+31 402743908

Published by Eindhoven University of Technology

PDEng Report 2023/097

Abstract

This project proposes a mathematical analysis approach for a qualitative SOTIF assessment on cooperative adaptive cruise control. This approach helps in mathematically formulating the SOTIF assessment and deriving safety requirements for ADAS function development. This project also discusses the initial steps of the dSPACE real-time system in the experimental vehicle, Carlab.

Keywords Automated vehicle, ISO 21448, SOTIF, ISO 26262, CACC, dSPACE.

Preferred reference Analysis of a safe and reliable automated driving platform. Eindhoven University of Technology, PDEng Report 2023/097, October 2023.

Partnership This project was supported by Eindhoven University of Technology

# Foreword

Mobility impacts us all in our daily lives and the industry trying to provide mobility to all of us is therefore immense. New technological developments constantly influence this industry and new possibilities to shake up the way we think about mobility arise at an increasingly rapid pace.

With the trend towards further automation, connectivity and cooperation of all sorts of transportation, mobility is further evolving towards Cooperative Connected Automated Mobility. This field focuses amongst others on connecting and cooperative self-driving vehicles.

Key in any automotive design, which is inherently the transportation of people and goods, is vehicle safety. With the recent developments in self-driving vehicles, we have become aware that the interaction between the vehicle and the environment (its Operational Design Domain (ODD)) is still not yet fully understood. Standards such as SOTIF (part of this thesis) focus on describing qualitative procedures in order to facilitate the functional safe design process. Eventually, this all needs to be tested as well as a means of verification. The traditional way of verification in automotive is to test out the functionality before releasing the vehicle in public. Organizations such as the Dutch RDW are responsible for the approval of vehicles for use on public roads. They play a key role in ensuring vehicles adhere to safety standards, preventing any harm to the public. With ODDs not being fully understood, developing safety verification tests that can measure this, is a daunting task.

In this thesis, Raj has worked on this topic, taking the well-known use case of Cooperative Adaptive Cruise Control as an example. TU/e has vast experience with this particular use case, having published numerous publications in the past 15 years as well as organized and participated in many field tests (such as GCDC 2011 and 2016). Raj has been working on the development of a safety framework using standards such as ISO 26262 and SOTIF. The proposed framework uses principles from robust control theory, to verify how the CACC system can stay within safe bounds, taking into account uncertainties. This framework ultimately can lead to a more quantitative way of verifying the safety of self-driving vehicles in the field of Cooperative Connected Automated Mobility and possibly reducing the need for unnecessary testing. Of course, it will never replace testing entirely.

We hope that this thesis will inspire others to further conduct research on this topic and ultimately lead to a more quantitative way of verifying the safety of self-driving vehicles, ultimately leading to safer and more efficient mobility.

*Jos den Ouden, Tom van der Sande*
*October, 2023*

# Acknowledgements

செய்யாமல் செய்த உதவிக்கு வையகமும்
வானகமும் ஆற்றல் அரிது.
-திருவள்ளுவர்

(அதிகாரம்:**செய்ந்நன்றியறிதல்**. குறள் எண்:**101**)

As this thesis marks the end of my two-year journey in the EngD program, I would like to take this opportunity to thank a few people without whom this journey would not have been possible. Since this one-year-long thesis is an integral part of the program, first and foremost I would like to thank my supervisors Tom and Jos, for providing me with the opportunity to work together and guiding me throughout. I would also like to thank Koen for helping me set up dSPACE and Erwin for helping me test with the car. Special thanks to Prabhat for being my lab partner during this thesis and for being the Micheal to my office. Secondly, I am thankful to Riske and Ellen for organizing and managing the EngD program. My gratitude also goes out to all my fellow EngD colleagues for making this journey incredibly fun. I will always cherish our memories of various projects, birthday parties, sports events and road trips. I would also like to thank my friends from Delft and India. To my wife Madhu, who has been a pillar of support and encouragement, I cannot emphasize how grateful I am. You are the best. Last but not least, my heartfelt gratitude to my family, who have always been my biggest supporters whose belief, unconditional love, and support have made me what I am today. I am eternally indebted to them.

*Raj Kumar Muniyandi*
*October, 2023*

# Executive Summary

TU/e has been working together with other research institutes, industries, and government to increase the safety in mobility applications. From the earlier projects to get its self-driving platform (Carlab) on public roads, it is clear that verification and validation according to ISO 26262 do not fulfill the safety requirements of self-driving vehicles. Thus creating the need for the ISO 21448 (SOTIF) standard which focuses on the safety of automated driving. However, the SOTIF assessment is qualitative. To this end, cooperative adaptive cruise control (CACC) is taken as the use case for this work.

This thesis aims to propose a mathematical analysis approach for formulating and analyzing the conventional SOTIF assessment for CACC following the ISO 21448 standard. To implement this, a conventional SOTIF assessment for CACC system is first performed with evidence from the literature. Later, the mathematical formulation and analysis are done using uncertainty modeling. However, using the approach, the worst case upper and lower bounds for the uncertain CACC system could not be found. The hypothesis for this might be due to the presence of internal actuator delay or the lack of a separate weighting function accounting for all the uncertainties. The proposed mathematical analysis is found to help in mathematically formulating the assessment and might help in deriving safety requirements earlier in the development of ADAS functions.

Another objective of this research is to integrate dSPACE MicroAutoBoxIII as a real-time operating system in the existing self-driving platform, Carlab. This is for safe testing and verification of research-developed automated driving systems. The initial steps of dSPACE integration show the communication with Carlab's CAN bus established.

# Contents

# Glossary

| | |
|---|---|
| **FuSa** | Functional Safety |
| **SOTIF** | Safety Of The Intended Functionality |
| **ODD** | Operational Design Domain |
| **HARA** | Hazards and Risk Assessment |
| **STPA** | System Theoretic Process Analysis |
| **FMEA** | Failure Modes and Effect Analysis |
| **ASIL** | Automotive Safety Integrity Level |
| **AD** | Automated Driving |
| **ADS** | Automated Driving System |
| **ADAS** | Automated Driver Assistance Systems |
| **NHTSA** | National Highway Traffic Safety Administration |
| **ISO** | International Standard Organisation |
| **LSAD** | Low Speed Automated Driving |
| **SAE** | Society of Automotive Engineers |
| **ANSI** | American National Standards Institute |
| **UC** | Use Case |
| **V2V** | Vehicle to Vehicle |
| **ACC** | Adaptive Cruise Control |
| **CACC** | Cooperative Adaptive Cruise Control |
| **SSCS** | String Stability Complimentary Sensitivity |
| **GNSS** | Global Navigation Satellite System |
| **IMU** | Inertial Measurement Unit |
| **DBC** | Data Base Container |
| **ECU** | Electronic Control Unit |

# List of Tables

# List of Figures

# 1 Introduction

The National Highway Transportation Safety Administration (NHTSA) reports that 93.5% of all road accidents are caused by some type of "human error" [1] [10]. The remaining 6.5% account for technical failures, environment, and various unknown reasons. With full automation, human error would be eliminated. Therefore it is natural, that the automotive industries and researchers are pushing for automated driving systems. But this also puts forward many other challenges such as "technical failure" and "technical risks" which will increase proportionally when the human driver is out of the loop as shown in Figure 1.1.

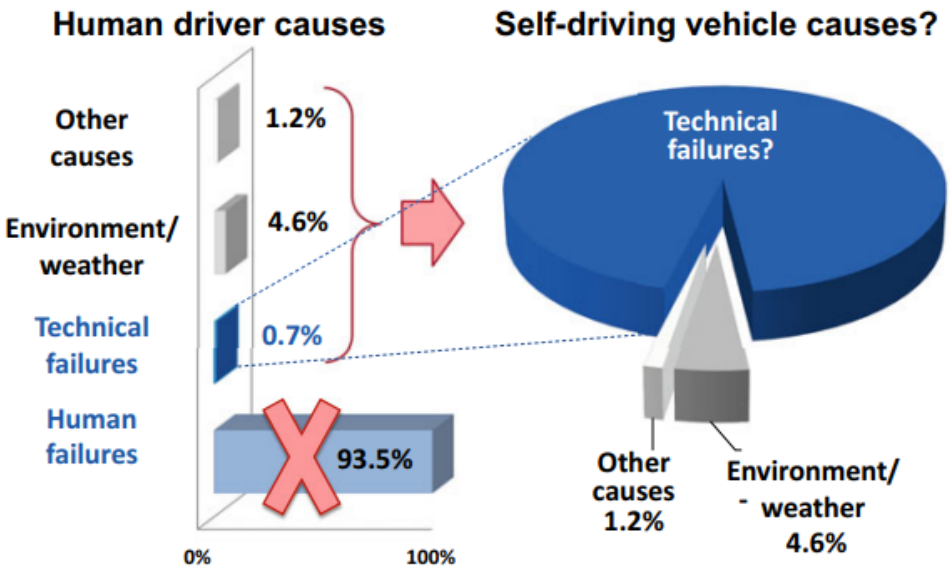



Figure 1.1: NHTSA survey on reasons for road accidents [1].

These technical failures in road vehicles correspond to the problems caused by the hardware and software components. These components can only be considered reliable if their design complies with the necessary safety standards such as ISO 26262 [11], the safety of the intended functionality (SOTIF) [5] standards, and overall vehicle–level safety.

TU/e has been working together with other research institutes, industries, and government to increase the safety in mobility applications. In earlier projects, such as Buurauto-Noom [12], TU/e had already collaborated with Dutch road authorities (RDW) to get its self-driving platform (Carlab), a Toyota Prius, on public roads. One of the key takeaways of that project was that verification and validation

according to ISO 26262 [11] did not fulfill the safety requirements of self-driving vehicles. This is due to ISO 26262 being initially designed as an electronics functional safety design standard and does not specifically address the operational design domain (ODD). Whereas with automated driving the vastness of ODDs makes it very difficult for verification and validation. Thus creating the need for ISO 21448 [5] compliance, which primarily focuses on the safety of the intended functionality under specific ODD.

SOTIF assessment for automated driving functions such as automated emergency braking (AEB)[13], and lane keeping assist (LKA)[14] have been performed in the past. These assessments as per the ISO 21448 standard are qualitative which can be ambiguous representations of safety requirements and constraints. To overcome this a mathematical quantitative analysis of the safety standards is required. The mathematical formulation will provide a precise and unambiguous representation of the assessment and help in capturing the risk of safety-related issues without any discrepancies.

The Automotive laboratory in TU/e is home to various in-house and industry-collaborated mobility projects. The laboratory also enables researchers to develop and test various automated driving systems and control algorithms with the help of experimental vehicles. However, safely testing these algorithms is impossible without a real-time operating system in the vehicle which can promise proper deployment, control, and safety. First steps of developing a safe automated platform have been taken for the Renault Twizys in the lab by modifying the stock vehicle with necessary sensors and a real-time computer for cooperative mobility research[3]. With a real-time system, the potential of Carlab, utilized for environmental perception, can be enhanced to safely test automated driving systems and control algorithms.

Cooperative adaptive cruise control (CACC) is chosen as the use case for this research as it is one of the widely researched ADAS functions with a lot of literature and expertise available on design and experimental validation in TU/e [15] [16][17].

## 1.1 Research objectives

Given the aforementioned challenges, this thesis aims to contribute to safety in automated driving systems by analyzing a safe and reliable automated driving platform. Based on the research objective the following goals are defined:

1. Mathematically analyze the qualitative SOTIF assessment for the cooperative adaptive cruise control function.
2. Integrate dSPACE as the real-time operating platform into the existing framework of Carlab for verification and validation of developed control algorithms.

## 1.2 Report outline

The organization of this thesis can be illustrated with the process flow diagram shown in Figure 1.2. Chapter 2 lays the foundation of the research by delving deep into the concepts of functional safety ISO 26262 and SOTIF ISO 21448. The chapter also introduces the design of the ADAS function cooperative adaptive cruise control which is the use case at hand and the operational design domain (ODD) for this thesis.

Chapter 3 presents the SOTIF framework analysis for the use case of CACC where the hazards and risk associated with the CACC function is discussed in detail. The chapter also talks about the functional modifications in order to make the identified risks safe.

Chapter 4 describes the mathematical formulation of the SOTIF analysis on CACC presented in Chapter 3. The known safe scenarios associated with CACC are also verified through simulation in this chapter.

Chapter 5 talks about the first steps taken to integrate dSPACE as the real-time operating system in the experimental vehicle, Carlab. Finally, Chapter 6 summarizes the main conclusions of this thesis and provides recommendations for further research into this domain.

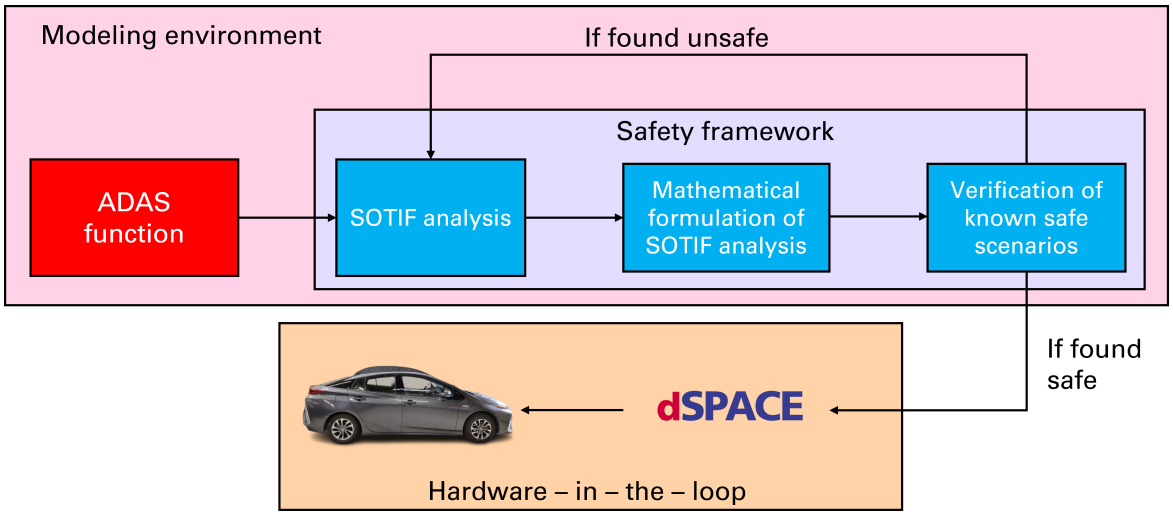



Figure 1.2: Report outline

The project management plan consisting of stakeholders analysis, project plan, risk management, and project deliverables can be found in the Appendix A.

# 2 Literature review and preliminaries

This chapter discusses existing safety standards such as ISO 26262 and the need for ISO 21448 for automated driving in detail. It also describes the controller design of a CACC system along with the preliminaries set for this thesis.

## 2.1 Safety in Automated and Autonomous vehicles

Safety in road vehicles is defined as the system's ability to perform the desired function in a safe manner such that the occupants of the vehicle, vulnerable road users, and the vehicle itself is not a safety hazard. The hazard arises from faults, malfunctions, errors, and failures within the hardware and software systems in the vehicle. Thus safety-critical systems should be able to identify these faults as early as possible to prevent unsafe behavior [3]. The main goal of the safety systems is to identify the different levels of risks that are nonacceptable and mitigate them as shown in figure 2.1 to bring the vehicle to an 'acceptable' region of risk.

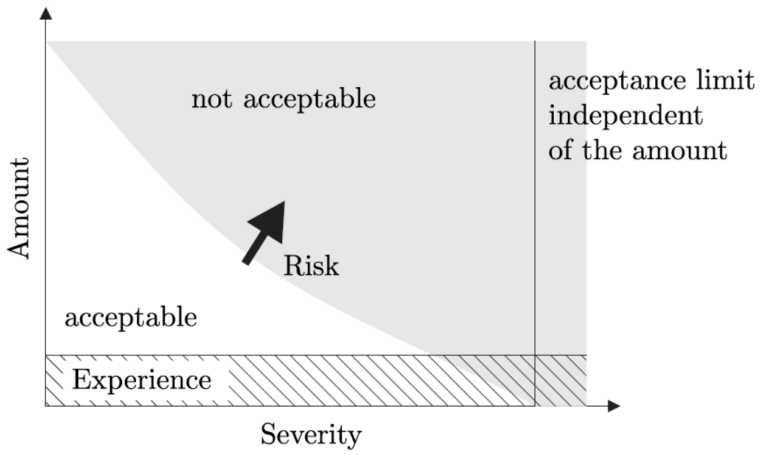



Figure 2.1: Graphical description of acceptable and non-acceptable risk [2].

Even though there are no direct standards for the safety of fully autonomous vehicles, there are various safety standards in place to regulate the design and development of automated vehicles. A list (but not limited to) of the available standards for automated vehicle safety is given in Table 2.1. However, ISO 26262 and ISO 21448 are the widely used safety standards in research and development. The application of standards like ISO 26262 and ISO 21448 in the development of automotive technologies is essential for several reasons such as

| Standard | Name |
|---|---|
| ISO 26262:2018 | Road Vehicles - Functional safety |
| ISO/PAS 21448:2022 | Safety of the intended functionality (SOTIF) |
| ISO 22737:202 1 | Intelligent transport systems - Low speed automated driving (LSAD) |
| ISO/SAE 21434:2021 | Road vehicles — Cybersecurity engineering |
| ANSI/UL4600:2022 | Evaluation of autonomous products |
| ISO 15026:2022 | Systems and software assurance |
| ISO 27001:2022 | Infrastructure security |

Table 2.1: List of available standards for automated vehicle safety.

1. Safety Assurance: The primary goal of these standards is to ensure the safety of vehicles and their occupants. With the increasing complexity of automotive systems, including ADAS and ADS there is a higher risk of functional failures that could lead to accidents. These standards provide a structured approach to identifying, mitigating, and managing these risks.

2. Risk reduction: Applying these standards helps in identifying potential failure modes, assessing their impact on safety, and implementing measures to reduce the associated risks.

3. Regulatory compliance: To make a developed ADAS function or system trustable and roadworthy adhering to these standards will help in getting acceptance from legal authorities like RDW for real-world testing and pilots.

### 2.1.1 ISO 26262 - Functional safety in road vehicles

Functional safety for road vehicles (ISO 26262) covers the random hardware failures and systematic design failures in electrical and/or electronic systems in series production vehicles [11]. The standard helps in defining the requirements and provides guidance to avoid and control random hardware and systematic faults that could violate the safety goals of the system under analysis.

In order to apply the ISO 26262 design process, the functional safety analysis known as the safety life cycle has to be followed as shown in Figure 2.2. The design process is initiated by item definition where the system under scope is described from a nontechnical perspective. The system is defined by its working, interactions, and boundaries. Based on the item definition, the list of potential hazards caused due to malfunction is identified in the vehicle-level hazard analysis. Hazards and Risk Assessment (HARA) and System Theoretic Process Analysis (STPA) techniques are used to define vehicle hazards. In the following step of risk assessment, the identified hazards are categorized based on exposure, severity, and controllability. After categorizing, the hazards are assigned Automotive Safety Integrity Levels (ASIL) ranging from A to D, where Level A represents the lowest degree of vehicle hazard while Level D corresponds to the highest degree [18]. Thus the strongest safety requirements are on ASIL D. Then the safety requirements for the vehicle are defined in the form of safety goals. Every safety goal corresponds to a particular vehicle hazard and inherits its ASIL. Having identified the vehicle-level hazards, the next step is to perform the safety analysis where the underlying mechanisms causing these hazards are identified with the help of the system's working and interaction described in the first step. Two methods are employed here, namely Failure Mode Effect Analysis (FMEA) and STPA. This step is very important as understanding the causes of these hazards can be helpful while proposing the necessary safety mechanisms to reduce the risk due to the

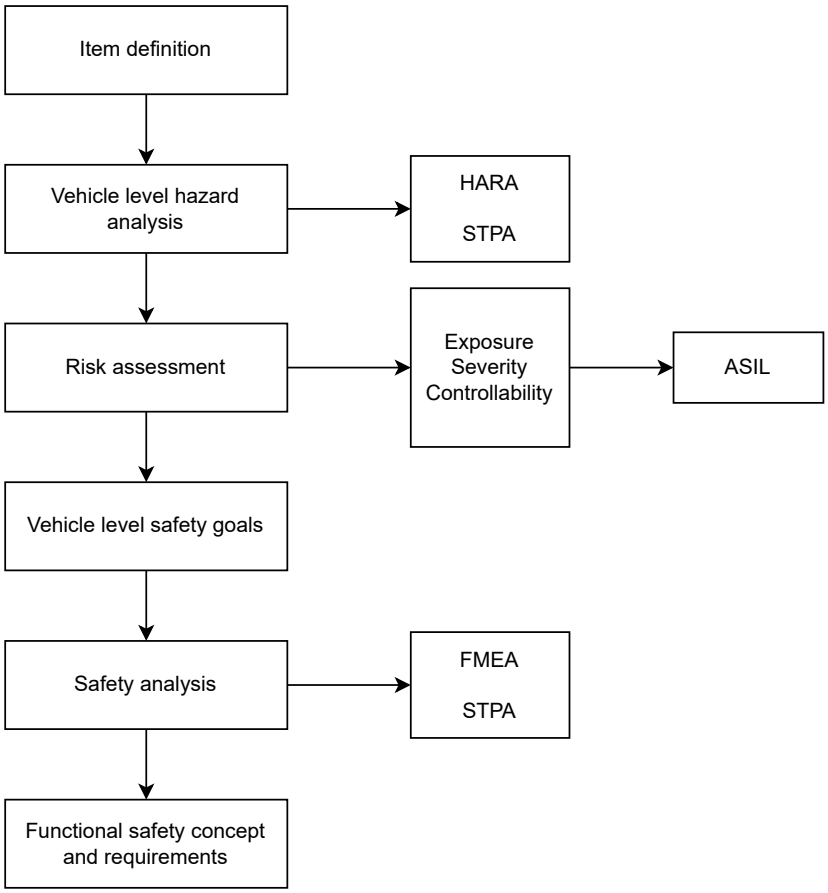



Figure 2.2: ISO 26262 design methodology (adopted from [3] & [4]).

hazard. Finally, with the knowledge of the previous steps, the functional safety concept is proposed along with safety requirements. The functional safety requirements are refined into technical safety requirements for both the hardware and software components in the development phase. At the end of the development phase, these requirements can be traced back using testing and validation.

### 2.1.2 ISO/PAS 21448 - Safety of the intended functionality

Potentially hazardous events can occur even when the system is free of the hardware and software faults covered by the ISO 26262 standard due to functional inefficiencies caused by algorithms, sensors, or human misuse. This might also be the result of using the intended system in a situation for which it was not intended, which leads to inadequate performance of the system. To address these hazardous events and misuse, the Safety Of The Intended Functionality (SOTIF) standard (ISO/PAS 21448) has been defined [19]. *"The absence of unreasonable risk due to hazards resulting from functional insufficiencies of the intended functionality or by reasonably foreseeable misuse by persons is referred to as the safety of the intended functionality"* [5].

This standard is designed to be applied to intended functionality where proper situational awareness is critical to safety and where that situational awareness is derived from complex sensors and processing

algorithms; especially emergency intervention systems (e.g., emergency braking systems) and Advanced Driver Assistance Systems (ADAS) with level 1 and 2 on the SAE standard J3016 automation scale. The functional and system specification includes relevant use cases and those use cases are comprised of several relevant scenarios. These scenarios could contain triggering events that lead to harm.

1. These scenarios can also be caused by reasonably foreseeable misuse, e.g., activating a functionality intended for the highway in an urban setting causes the vehicle to be in a scenario in which it does not detect a red traffic light.
2. Reasonably foreseeable misuse can directly lead to a hazard, e.g. in case of mode confusion where the driver assumes that the system is active even though it is deactivated.
3. The inability to control the hazardous event can also be the result of a reasonably foreseeable misuse, e.g. the driver does not supervise the system as he is supposed to do.

Relevant use cases are therefore broken down into different scenarios which are classified into four areas such as known safe (Area 1), known unsafe (Area 2), unknown unsafe (Area 3), and unknown safe (Area 4) as shown in Figure 2.3.

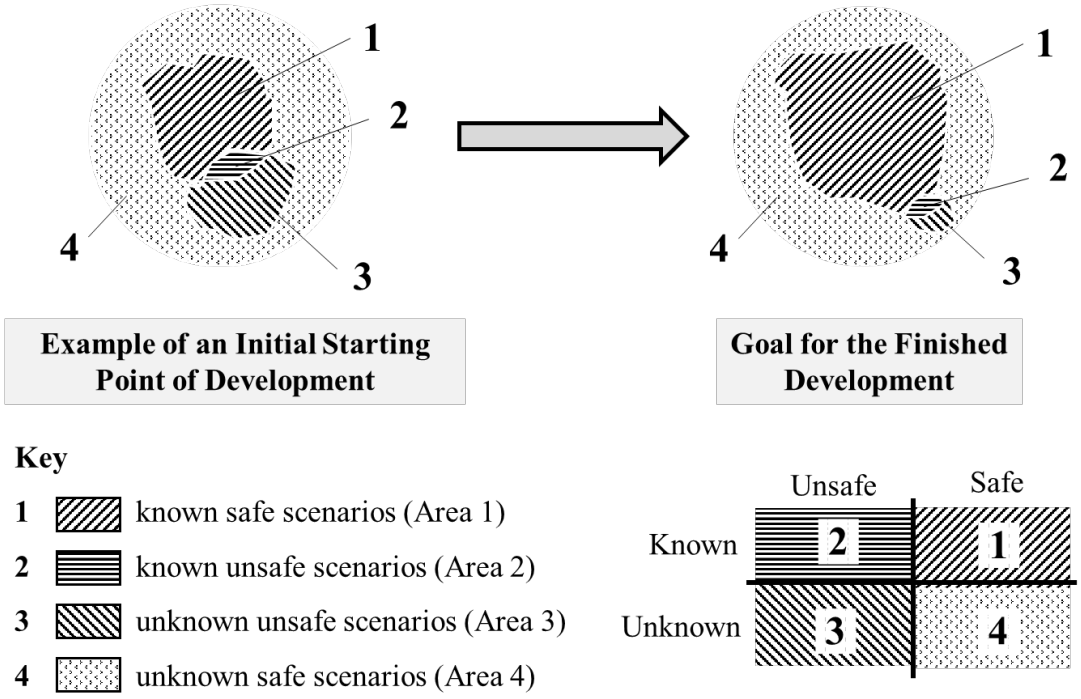



Figure 2.3: Classification of scenarios and evolution resulting from SOTIF[5]

The goal is to minimize Area 2 and Area 3 to make all unsafe (known and unknown) scenarios safe by limiting the Operational Design Domain (ODD) and to provide an argument that these areas are sufficiently small and therefore that the resulting residual risk is acceptable. An ODD can be defined as the intended behavior within the defined environmental condition that the function has been designed

to work on. Thus evolving to the growth of Area 1 during development. Area 4 - unknown safe scenarios are considered as the scenarios for completeness of the system therefore not considered by that standard.

The process flow for applying SOTIF is shown in Figure 2.4. Some of the steps can be performed simultaneously in complementary with the functional safety analysis shown in section 2.1.1. To ensure the SOTIF, choosing a capable overall system architecture becomes crucial, and to guarantee that overall capability, corresponding activities are carried out both early on and throughout the entire functional development lifecycle.

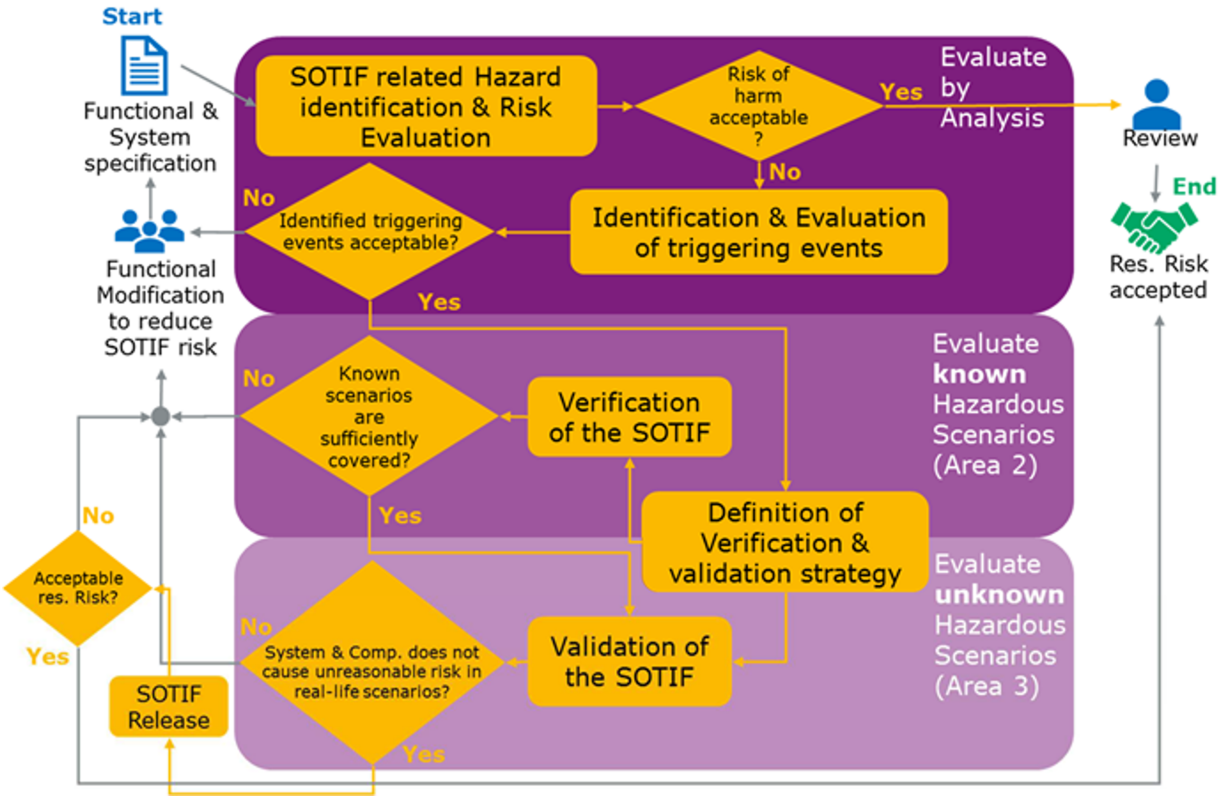



Figure 2.4: Flowchart for the improvement of the intended functionality to ensure its safety [5] [6].

### 2.1.3 Integrated Analysis Approach

ISO 26262 and ISO 21448 are concerned with safety in the automotive industry, they address different aspects of safety. ISO 26262 deals with functional safety across the entire development lifecycle of vehicles, ensuring that systems function correctly and safely under various conditions. On the other hand, ISO 21448 (SOTIF) focuses on the safety of the intended functionality, highlighting the need to consider situations where a system's intended behavior might still lead to safety-critical situations due to limitations in perception or understanding of the environment.

The need for integration of both these standards arises for the development of autonomous vehicles where both fail-safe and fail-operational modes are required [7]. Various state-of-the-art researchers

have integrated the key SOTIF steps into the existing safety analysis process to develop an updated safety analysis process [20] [21].

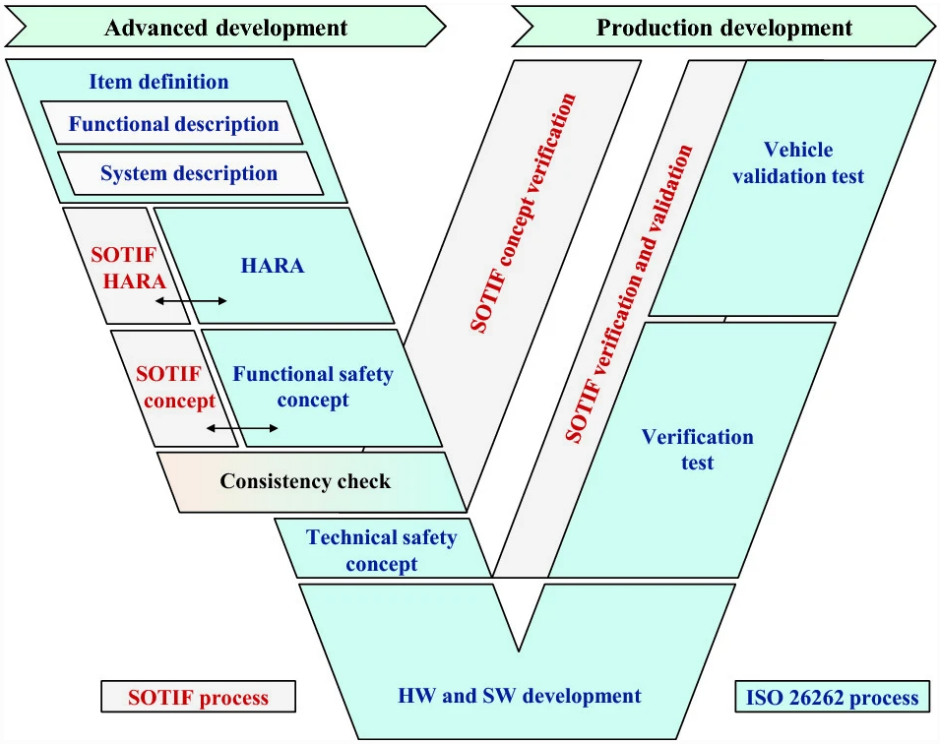



Figure 2.5: Functional safety and SOTIF integrated development approach [7].

The integrated approach in Figure 2.5 shows that the SOTIF process has a strong overlap with ISO 26262. The initial development phase includes item definition, hazard analysis, and risk assessment are in line with both standards. While ISO 26262 outputs a functional safety concept, ISO 21448 produces functional modifications that are measures developed to avoid, reduce, or mitigate SOTIF risks that result from system limitations that lead to safety violations [20]. ISO 26262 and ISO 21448 integrated approach can be very beneficial when dealing with increasing SAE levels of automation [22].

## 2.2 Use Case

This research sets a foundation for functional safety analysis on ADAS functions by selecting Cooperative Adaptive Cruise Control (CACC) as the use case due to its uncertain design parameters and delays. The use case under consideration involves two vehicles of different dynamics with only longitudinal control as the dynamics driving task (DDT). Both vehicles are assumed to be equipped with a CACC system along with necessary sensors.

### 2.2.1 Cooperative adaptive cruise control

Cooperative Adaptive Cruise Control (CACC) is an advanced driver assistance system (ADAS) that builds upon traditional Adaptive Cruise Control (ACC) by enabling vehicles to communicate with each other and cooperate to optimize traffic flow and increase overall safety on the road. CACC uses vehicle-to-vehicle communication for short-distance vehicles following using wireless communication in addition to onboard sensors. The CACC system is subject to performance, safety, and comfort requirements [23]. To meet these requirements, a CACC-equipped vehicle platoon needs to exhibit string-stable behavior, such that the effect of disturbances is attenuated along the vehicle string, thereby avoiding congestion due to so-called ghost traffic jams. The design of the CACC controller and string stability condition followed in this research are discussed in the following sections.

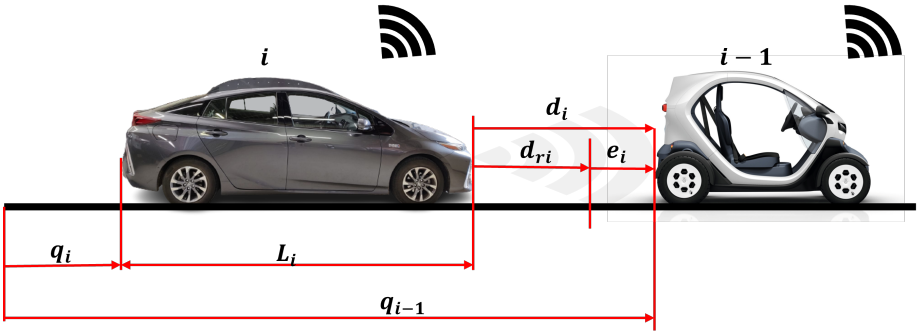



Figure 2.6: CACC-equipped heterogeneous vehicle platoon

### 2.2.2 Controller Design

The controller design presented in this section is proposed for CACC of heterogenous vehicle platoons [8] as an extension to the CACC controller design for homogeneous platoons [17]. Consider a platoon of $m$ vehicles as shown in Figure 2.6, with $d_i$ being the distance between vehicle $i$ and its preceding vehicle $i-1$, and $v_i$ the velocity of vehicle i. The objective of each vehicle is to follow the preceding vehicle at a desired inter vehicle distance $d_{r,i}$. The vehicle dynamics of each vehicle in the platoon are described by,

$$\begin{aligned} \dot{q}_i &= v_i \\ \dot{v}_i &= a_i \\ \dot{a}_i &= -\frac{1}{\tau_i} a_i + \frac{1}{\tau_i} u_i \end{aligned} \qquad i = 1, 2, \ldots, m \tag{2.1}$$

with $q_i, v_i$, and $a_i$ denoting the position of the rear bumper, velocity, and acceleration of vehicle $i$, respectively. The vehicle dynamics model (2.1) is obtained after feedback linearization of a more detailed model [24] and used in multiple works [25][17]. The desired acceleration $u_i$ is considered as the input, and $\tau_i > 0$ denotes a time constant that represents the driveline dynamics of vehicle $i$. The platoon is considered to be a heterogeneous platoon i.e., $\tau_i \neq \tau_j$, which says that the driveline dynamics are different for each vehicle.

The controller's objective is to follow the predecessor vehicle at a desired distance $d_{r,i}$ using a constant time-gap policy [26]:

$$d_{r,i} = r_i + h_i v_i, \quad 2 \leq i \leq m, \tag{2.2}$$

where $h_i > 0$ and $r_i$ denote the minimum headway time and the standstill distance respectively. The spacing error $e_i$ is derived from the Figure 2.6 as

$$e_i = (q_{i-1} - q_i - L_i) - (r_i + h_i v_i), \tag{2.3}$$

where $L_i$ denotes the length of vehicle $i$ as shown in Figure 2.6. The error states are defined as

$$\varepsilon_i = \begin{bmatrix} \varepsilon_{i,1} \\ \varepsilon_{i,2} \\ \varepsilon_{i,3} \end{bmatrix} := \begin{bmatrix} e_i \\ \dot{e}_i \\ \ddot{e}_i \end{bmatrix}, \tag{2.4}$$

the error dynamics are described as

$$\dot{\varepsilon}_i = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau_i} \end{bmatrix} \varepsilon_i + \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{\tau_i} \end{bmatrix} \zeta_i \tag{2.5}$$

where $\zeta_i$ is regarded as the external input of the system 2.5. The external input $\zeta_i$ is defined as

$$\zeta_i := h_i \dot{u}_i + u_i - \left(1 - \frac{\tau_i}{\tau_{i-1}}\right) a_{i-1} - \frac{\tau_i}{\tau_{i-1}} u_{i-1}, \tag{2.6}$$

hence, the control law [8] $\zeta_i = \begin{bmatrix} k_p & k_d & k_{dd} \end{bmatrix}$ or

$$\dot{u}_i = -\frac{1}{h_i} u_i + \frac{1}{h_i} \begin{bmatrix} k_p & k_d & k_{dd} \end{bmatrix} \varepsilon_i + \frac{\tau_{i-1} - \tau_i}{h_i \tau_{i-1}} a_{i-1} + \frac{\tau_i}{h_i \tau_{i-1}} u_{i-1} \tag{2.7}$$

stabilizes the error dynamics, provided that $k_p > 0$, $k_{dd} > -1$, and $k_d > \frac{k_p \tau_i}{1+k_{dd}}$ [8]. $k_p, k_d, k_{dd}$ are the gains of the feedback controller. Note that the above controller requires both $u_{i-1}$ and $a_{i-1}$ as inputs. In the case of a homogeneous platoon, i.e., $\tau_i = \tau_{i-1}$, the controller (2.7) reduces to

$$\dot{u}_i = -\frac{1}{h_i} u_i + \frac{1}{h_i} \begin{bmatrix} k_p & k_d & k_{dd} \end{bmatrix} \varepsilon_i + \frac{1}{h_i} u_{i-1} \tag{2.8}$$

where $u_{i-1}$ is the external input of the system [27]. Since the controller (2.7) requires $a_{i-1}$ also as the input, this needs measurement of longitudinal acceleration, locally in the transmitting vehicle. The measurement of longitudinal acceleration suffers a low signal-to-noise ratio [8], thus (2.8) is adopted in a heterogeneous setting [28] in this thesis.

## 2.3 String stability

String stability is the measure of disturbance attenuation along the vehicle string in the upstream direction. CACC is known to achieve string stability at minimum headway times significantly smaller than 1s [29]. The mathematical definition of string stability is incorporated from Ploeg [17] as follows

*Definition: Consider a string of $m \in \mathbb{N}$ interconnected vehicles. This system is string stable if and only if*

$$\|z_i(t)\|_{\mathcal{L}_p} \leq \|z_{i-1}(t)\|_{\mathcal{L}_p}, \quad \forall t \geq 0, 2 \leq i \leq m \tag{2.9}$$

*where $z_i(t)$ can either be the distance error $e_i(t)$, the velocity $v_i(t)$ or the acceleration $a_i(t)$ of vehicle $i$; $z_1(t) \in \mathcal{L}_p$ is a given input signal, and $z_i(0) = 0$ for $2 \leq i \leq m$. $\|.\|_{\mathcal{L}_p}$ denotes the signal $p$-norm, whereas the vehicles in the string are enumerated $i = 1, ...., m$, with $i = 1$ indicating the lead vehicle. Thus stating $\|z_i(t)\|_{\mathcal{L}_p}$ must decrease in upstream direction.*

### 2.3.1 String stability analysis

The controller described in (2.8) helps in achieving the objective of vehicle following but does not yet guarantee string stability. To analyze string stability, the criteria known as the string stability complementary sensitivity (SSCS) function $\Gamma_i(s)$ presented in [17] is used.

$$\hat{a}_i(s) = \Gamma_i(s)\hat{a}_{i-1}(s), \quad 2 \leq i \leq m, \tag{2.10}$$

where $\hat{a}_i(s)$, denotes the Laplace transform of $a(t)$. The SSCS function can be expressed as

$$\|\Gamma_i(j\omega)\|_{\mathcal{H}_\infty} = \max_{a_{i-1} \neq 0} \frac{\|a_i(t)\|_{\mathcal{L}_2}}{\|a_{i-1}(t)\|_{\mathcal{L}_2}} \tag{2.11}$$

$\|.\|_{\mathcal{H}_\infty}$ denotes the $\mathcal{H}_\infty$ norm, which for scalar transfer functions, equals the supremum of $|\Gamma_i(j\omega)|$ over the frequency $\omega$. Thus, the definition 2.9 becomes:

$$\|\Gamma_i(j\omega)\|_{\mathcal{H}_\infty} \leq 1, \quad 2 \leq i \leq m. \tag{2.12}$$

The above condition (2.12) is used to asses string stability in the frequency domain.

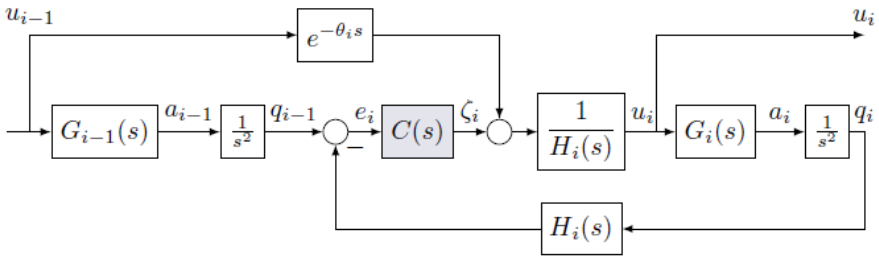



Figure 2.7: Block diagram of CACC controller in a heterogeneous setting [8]

The block diagram in Figure2.7 represents the CACC controller (2.8) in a heterogeneous setting used in this thesis, with controller $C(s)$, the vehicle acceleration transfer function $G_i(s)$, spacing policy

$H_i(s)$ and wireless communication delay $e^{-\theta_i s}$ represented as $D(s)$ in Laplace domain.

$$\begin{aligned} C(s) &= k_p + k_d s + k_{dd} s^2 \\ G_i(s) &= \frac{1}{\tau_i s + 1} e^{-\phi_i s} \\ H_i(s) &= h_i s + 1 \\ D(s) &= e^{-\theta_i s} \end{aligned} \tag{2.13}$$

$\tau_i$, $\phi_i$, and $\theta_i$ represent the driveline lag, internal actuator lag, and communication delay respectively. Using the block diagram of CACC in figure2.7, the SSCS transfer function $\Gamma_i(s)$ for the heterogeneous case [8][28] is defined as:

$$\Gamma_i(s) = \frac{a_i(s)}{a_{i-1}(s)} = \frac{1}{H(s)} \frac{G_i(s)}{G_{i-1}(s)} \frac{D(s)s^2 + G_{i-1}(s)C(s)}{s^2 + G_i(s)C(s)} \tag{2.14}$$

## 2.4 Operational Design Domain

Per SAE J3016[30], the Operational Design Domain (ODD) for a driving automation system is defined as *"Operating conditions under which a given driving automation system, or feature thereof, is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics."*; in short the ODD defines the limits within which the driving automation system is designed to operate, and as such, will only operate when the operating conditions described within the ODD are satisfied.

The taxonomy of ODD for automated vehicles is categorized as physical infrastructure, operational constraints, actors, connectivity, and environmental conditions [31]. Based on this, this thesis considers low-speed urban driving conditions limited to as follows

the parameters intended for this research are limited to as follows

1. Environmental conditions: The vehicle equipped with the ADAS function should be capable of operating in bright light conditions, without any rain, and on dry roads.
2. Physical infrastructure: The vehicle equipped with the ADAS function should be capable of operating on straight flat roads without any intersections and inclinations.
3. Connectivity: The vehicle equipped with the ADAS function should be equipped with V2V communication and a ranging sensor like radar, lidar, or camera.
4. Operating speed: The vehicle's speed limit is set to $50km/hr$ during the deployment of the ADAS function.
5. Actors: The vehicle equipped with the ADAS function is considered to be part of a heterogeneous platoon.

Any deviation from the defined parameters and conditions is considered to be exiting the ODD or outside the ODD.

## 2.5 Summary

This chapter started with introducing the safety standards such as functional safety of road vehicles ISO26262 and safety of the intended functionality for autonomous vehicles ISO21448 and briefly summarized the process flow of SOTIF assessment used in this thesis. Later, the use case of this thesis CACC was described in detail with the controller design and the string stability analysis for heterogeneous platoons. Briefly summarized, this thesis assumes linear longitudinal vehicle dynamics, predecessor follower communication, and a constant time gap distance policy. Additionally, heterogeneity with respect to drivelines $\tau$, internal actuator delay $\phi$, and wireless communication delay $\theta$ is considered throughout this thesis. Finally the ODD considered for this thesis is also defined.

# 3 Conventional SOTIF analysis on CACC function

This chapter delves into the qualitative SOTIF analysis of the ADAS function CACC as proposed by ISO 21448 [5]. The chapter starts by identifying the SOTIF-related hazards followed by the risk assessment. Later, the triggering events for the hazards are analyzed and functional modifications are proposed at the end.

## 3.1 SOTIF analysis of CACC

SOTIF analysis follows a similar framework to the conventional ISO 26262 as shown in Figure 2.5 and can be performed simultaneously during the functional safety analysis. SOTIF focuses on the foreseeable misbehaviors and hazardous situations that can arise from the intended functionality of a system, particularly in the context of autonomous vehicles and ADAS. When applied to a specific ADAS function like CACC, the SOTIF analysis aims to identify and mitigate potential safety issues that might occur even when the function is operating as intended. The SOTIF analysis applied to CACC is explained in the following sections.

### 3.1.1 Functional specification of CACC

Cooperative adaptive cruise control is an ADAS function that uses sensors such as radars, lidars, and cameras to measure the inter-vehicle distance and relative velocity between the ego vehicle and the preceding vehicle. In addition to the measuring capability, the function is also equipped with V2V communication for receiving the target vehicle's desired acceleration.

The system architecture of a vehicle equipped with CACC system is shown in Figure 3.1. The radar is used for measuring the relative distance and speed of the ego vehicle with respect to the preceding vehicle and GNSS/IMU system gives the current vehicle states to the CACC system along with the target vehicle's information such as desired acceleration through wireless V2V communication. The CACC system uses the given information to calculate the control input (desired acceleration) of the ego vehicle which is again converted as actuator setpoints to low-level controllers such as the throttle and brake.

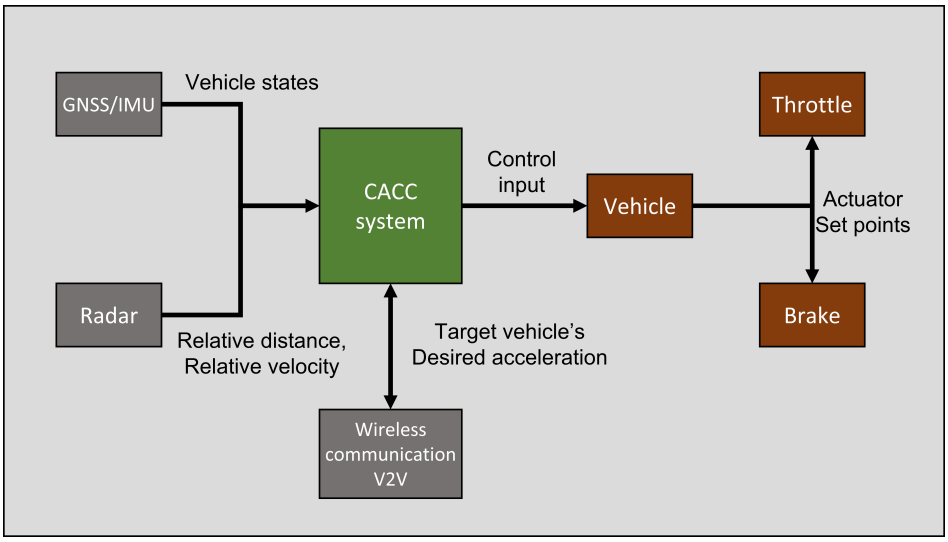



Figure 3.1: System architecture of CACC

### 3.1.2 Hazard Identification

The next step after function specification is to identify the list of function-level hazards caused due to a limitation or malfunction. HARA and STPA techniques are used for defining these hazards by guide words such as *loss of function, more than intended. less than intended, intermittent or wrong timing, incorrect direction, not requested* [4]. The guide word *uncertain* is used to define the hazards in this thesis, as it denotes that the value of the parameter can deviate from the parameter used in the control system.

The hazards are categorized into algorithm-related, sensor/hardware-related, and ODD-related as the standard focuses on functional insufficiency due to complex sensors, processing algorithms, and misuse of the function [5]. For the CACC function, the algorithm-related hazards are due to uncertain communication delay $\theta$, uncertain internal actuator delay $\phi$, and uncertain driveline lag $\tau$. The loss of communication is due to sensor/hardware-related errors and unintended usage corresponds to misuse of the function or ODD-related hazards.Based on the guidelines, a list of function-level hazards due to malfunctions is defined in Table 3.1.

| Type of hazard | Hazard | Description |
|---|---|---|
| Algorithm related | Uncertain communication delay | The desired acceleration is received at a different delay than expected. |
| | Uncertain internal actuator delay | The internal actuator delay of the vehicle is different than expected. |
| | Uncertain driveline lag | The dynamics of the vehicle is slower or faster than intended. |
| Sensor/Hardware related | Loss of communication | Failure of V2V communication from the target vehicle. |
| ODD related | Unintended usage | The function is used at a scenario which it was not intended for. |

Table 3.1: Function level hazards due to malfunctions in the CACC system

### 3.1.3 Risk Assessment

The SOTIF framework shows that the risks associated with vehicle hazards are estimated by defining the severity and controllability levels for the driver or passengers in the driving scenario [5]. The exposure value does not exist for SOTIF and it is delegated to the validation target in the verification and validation strategy [22].

Severity $S$ is the extent of harm caused to the vehicle occupants or the road users in a hazardous situation. The different levels of severity are defined as

- **S0:** No injuries
- **S1:** Light and moderate injuries
- **S2:** Severe and life-threatening injuries (survival probability)
- **S3:** Life-threatening injuries, fatal

Controllability is the ability of vehicle occupants or road users to react on time to reduce the risk of a hazard. The different levels of controllability $C$ are defined as

- **C0:** Controllable in general
- **C1:** Simply controllable
- **C2:** Normally controllable
- **C3:** Difficult to control or uncontrollable

Compared to the risk assessment process described in ISO 26262 [11], SOTIF applies a qualitative or binary risk assessment. If the potentially hazardous event may lead to harm (i.e., severity parameter is non-zero, $S > 0$) and is not simply controllable (i.e., controllability is non-zero, $C > 0$), then the hazardous event requires a SOTIF-related safety modification [20].

According to litertaure[17][32], The main functions of the CACC system are closer-vehicle following and string stable behavior as discussed in the section 2.2.1. The CACC system is known for achieving string stable behavior with closer vehicle following with time gaps $h < 1s$. Therefore, once the vehicle deviates from the modeled system, the time for the driver to take over the longitudinal control and maintain the string stability becomes very narrow resulting in a harmful event. Thus, in this thesis, the time gap $h$ between the ego vehicle and the target vehicle is used for evaluating the risks. A larger time gap $h$ has to be set with the increase in parameters like communication delay $\theta$, internal actuator delay $\phi$, and driveline lag $\tau$ to maintain string stability. Any uncertainty in the parameters mentioned above will require a smaller or larger time gap $h$, which affects the string stable behavior. Hence, the hazards are evaluated with non-zero controllability and severity parameters as shown in Table 3.2, calling for functional modification.

Since string stability is also considered a performance criterion of the CACC function [33], it is chosen as the safety acceptability criterion for the identified hazards in this thesis.

| Hazards | Controllability (C) | Severity (S) |
|---|---|---|
| Uncertain communication delay | C > 0 | S > 0 |
| Uncertain internal actuator delay | C > 0 | S > 0 |
| Uncertain driveline delay | C > 0 | S > 0 |
| Loss of communication | C > 0 | S > 0 |
| Unintended usage | C > 0 | S > 0 |

Table 3.2: Risk Assessment of hazards associated with the malfunctions of CACC system

### 3.1.4 Identification of triggering events

According to SOTIF standard [5], the triggering events that can trigger potentially hazardous behavior are identified and evaluated for their acceptability. The analysis aims to identify and evaluate triggering events, by assessing the known limitations of the system components and foreseeable misuse that could potentially result in vehicle hazards [20]. The system weaknesses include insufficiencies in those of its sensors, algorithms, actuators, and the related scenarios that could lead to identified hazards. The algorithm limitations are considered as the capability of the function to handle possible scenarios. The trigger events for sensors and actuators are usually accuracy, range, response time, and durability to name a few [5](SOTIF standard-Clause 7).

The algorithm-related trigger events in a CACC system might occur due to the model's simplicity by neglecting the uncertainty and dynamics. Setting constant values for $\theta, \phi, \tau$ are also identified as trigger events, as in reality, these values might not be known 100%. Communication loss can be triggered by the failure of the hardware itself or by the accuracy and range limitations of the sensor used. Finally, the unintended usage hazard is triggered by the misuse of the intended functionality in a different environment, location, and weather conditions from the defined ODD in section 2.4. The possible triggering events for the potential hazards in a CACC system are identified as shown in Table 3.3.

| Hazards | Type of triggering event | Description |
|---|---|---|
| Uncertain communication delay | Algorithm and hardware limitation | Simplified model with constant communication delay |
| Uncertain internal actuator delay | Algorithm and hardware limitation | Simplified model with constant internal actuator delay |
| Uncertain driveline lag | Algorithm and hardware limitation | Simplified model with constant driveline lag. |
| Loss of communication | Hardware limitation | Durability of the hardware, accuracy of the sensor, range of the hardware. |
| Unintended usage | Algorithm and hardware limitation | Weather condition and environment. |

Table 3.3: Identification of triggering events for the hazards related to CACC function

### 3.1.5 Functional modification to reduce SOTIF-related risks

Finally, the functional modifications to avoid, reduce or mitigate the SOTIF-related risks are proposed using the ISO 21448 standard [5]. The functional modification is an integral part of the SOTIF analysis as the identified triggering events have the possibility to trigger potentially hazardous behavior. This leads to a hazardous event with incredible harm and cannot be evaluated as acceptable with respect to the safety of the intended functionality [5](SOTIF standard - clause 8).

The identified limitations with respect to the CACC algorithm are tackled using improvements in the algorithm by including all the uncertainties in communication delay $\theta$, internal actuator delay $\phi$, and driveline lag $\phi$. Additionally, when the vehicle is out of the ODD, improvements such as warning the driver can be made. For instance, when the vehicle is operated at a speed $v > 10m/s$, the driver can be warned with alarms for exiting the ODD of $v \leq 10m/s$.

In situations such as when the CACC system loses communication with the preceding vehicle, it can warn the driver. Its functionality can also be reduced to fallback strategies like degraded cooperative cruise control d-CACC, but the transition from CACC to d-CACC changes the system dynamics which still has to be investigated for safe implementation [29]. The sensor and hardware-related hazards are improved by selecting adequate sensor technology, proper mounting, calibration, and redundancy [5]. The functional modifications for SOTIF-related risks of the CACC system are shown in Table 3.4.

| Type of hazard | Safety measures | Derived modification |
|---|---|---|
| Algorithm related | Algorithmic improvements | Modeling a robust CACC function accounting for uncertainties in communication delay, internal actuator delay and driveline lag |
| | Identification of ODD exit with appropriate warning | Warning when the values are out of the ODD range. |
| | Warning or degradation strategy for known unsupported scenarios | Fallback d-CACC or ACC strategy is triggered. Increase $h$ value to maintain safe time gap. ( For both the fallback strategies the transient states have to be investigated) |
| Sensor/Hardware related | Adequate technology, hardware redundancy, location of hardware on the vehicle | Change/replacement of sensors, proper mounting of hardware on the vehicle |
| ODD related | Restriction or reduce functionality when exiting ODD | Triggered warning to the driver and handing over the authority. |

Table 3.4: SOTIF related safety measures for identified hazards

### 3.1.6 Verification of known scenarios

The verification of the modified system should be able to provide the necessary evidence of meeting the acceptance criteria of the system [5]. String stability is stated as the acceptance criteria for this thesis in the previous sections. Thus proving the string stability behavior for the use case will generate

the necessary evidence for a safe CACC system. The string stability analysis of the use case will be described in detail in section 4.6.

## 3.2 Summary

In this chapter, the conventional way of assessing the safety of the intended functionality was carried out using the ISO 21448 standard. Since the intention of the standard is to identify and mitigate the insufficiencies in the sensor and algorithms, the related hazards were identified, and evaluated and the respective functional modifications to meet the acceptability criterion were proposed. In addition, the hazards related to the misuse of the functionality were also addressed, and functional modifications were derived.

# 4 Mathematical analysis of SOTIF assessment

Having performed the conventional SOTIF assessment using qualitative statements and tables in the previous chapter, this chapter proposes the mathematical analysis approach of the ISO 21448 standard for CACC in detail. This chapter only considers algorithm-related hazards for mathematical analysis. The chapter starts by defining the analytical CACC model and goes on to enlist and explain why the uncertainties have to be captured for mitigating the SOTIF-related safety risks. Finally, an uncertain CACC model is analyzed as the modified CACC system.

## 4.1 Function and system specification

The CACC system in the Laplace domain defined in section 2.3.1 is represented with controller $C(s)$, the vehicle acceleration transfer function $G_i(s)$, spacing policy $H_i(s)$ and wireless communication delay $D(s)$.

$$C(s) = k_p + k_d s + k_{dd} s^2 \tag{4.1}$$

$$G_i(s) = \frac{1}{\tau_i s + 1} e^{-\phi_i s} \tag{4.2}$$

$$H_i(s) = h_i s + 1 \tag{4.3}$$

$$D(s) = e^{-\theta_i s} \tag{4.4}$$

Here $\tau_i$, $\phi_i$, and $\theta_i$ represent the driveline lag, internal actuator lag, and communication delay respectively. The CACC system under the considered use case is built around the three given parameters, and tuning parameters which are headway time $h$, and controller gains $k_p, k_d, k_{dd}$. The given parameters reflect the characteristics of the vehicle string while the tuning parameters are used to stabilize error dynamics and achieve string stability.

## 4.2 SOTIF related hazard identification

In real-world situations, the communication delay, internal actuator lag, and driveline lag are not known 100% which results in the CACC system not being certain. When the communication delay $\theta$ is not definite, the desired acceleration from the target vehicle is not received at the intended time for calculating the control set points for the ego vehicle. The uncertain internal lag $\phi$ also causes the vehicle to respond much faster or slower than intended. The uncertain driveline lag $\tau$ corresponds to the dynamics of the vehicle being faster or slower. All the above cases might result in string instability which leads to collision with the other vehicles in the string. The uncertainties can be formulated as:

$$\begin{aligned} \theta_u &= \theta(1 \pm \Delta\theta) \\ \phi_u &= \phi(1 \pm \Delta\phi) \\ \tau_u &= \tau(1 \pm \Delta\tau) \end{aligned} \tag{4.5}$$

where, $\theta_u, \phi_u, \tau_u$ denote the uncertain communication delay, uncertain internal actuator delay, and uncertain driveline lag respectively. $\theta, \phi, \tau$ are the nominal values of the given parameters and $\Delta\theta, \Delta\phi, \Delta\tau$ represents the variation of the given parameters from the nominal or expected value.

## 4.3 Risk assessment

The identified hazards are evaluated in risk assessment by using the controllability and severity values as stated in section 3.1.3. The qualitative risk assessment evaluated the identified hazards with non-zero severity and controllability values based on the time-gap $h$ to maintain string stability. A discrete search for the minimum time gap $h$ is carried out using CACC model defined in 4.1 and the SSCS transfer function mentioned in (2.14) for $\theta, \phi, \tau$ by fixing the controller gains $k_p = 0.2, k_d = 0.7, k_{dd} = 0$. Figure 4.1 shows the relationship of minimum time gap $h$ and communication delay $\theta$, internal actuator delay $\phi$, driveline lag $\tau$ in the subplots.

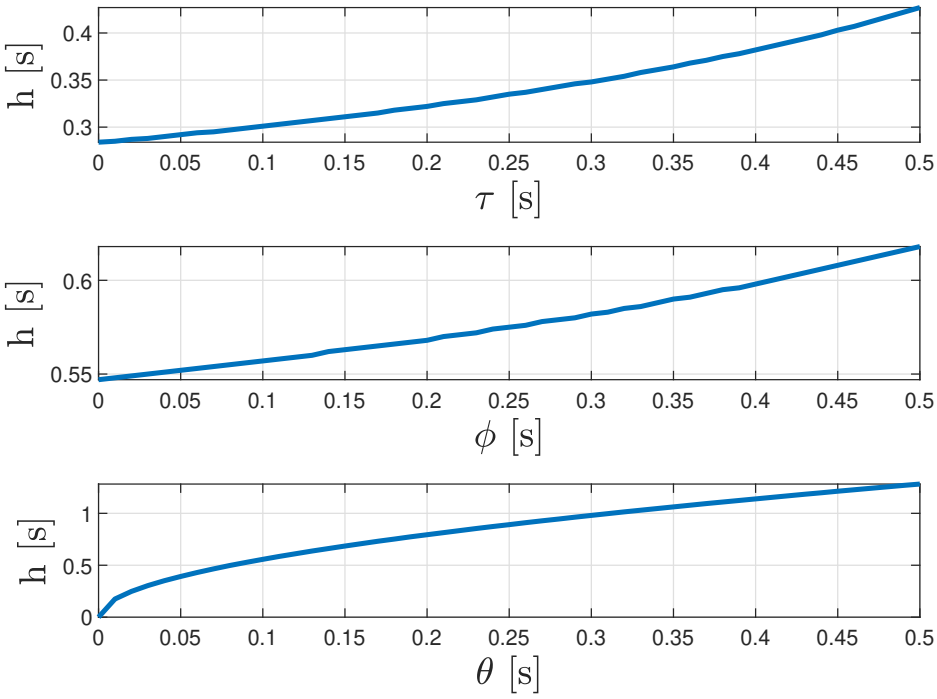



Figure 4.1: Minimum time gap $h$ versus vehicle driveline lag $\tau$ at $\theta = 0.01, \phi = 0.01$, internal actuator lag $\phi$ at $\theta = 0.01, \tau = 0.01$ and communication delay $\theta$ at $\phi = 0.01, \tau = 0.01$

As seen in the plot $\theta$ vs $h$, for a larger communication delay $\theta$ in the system, a larger $h$ value is required to maintain a string stable behavior. The plot $\phi$ vs $h$ also shows that the required time to maintain

string stable behavior increases with the value of internal actuator delay. This is due to the delay in the response of actuators like brake and accelerator. The plot $\tau$ vs $h$ also shows the increase in the time gap when the dynamics of the vehicle become slower. Therefore, in a CACC system with a constant time gap $h$ as mentioned in the use case, it becomes very difficult to maintain string-stable behavior with uncertainty in the above-mentioned parameters, thus $C > 0$ and $S > 0$. The non-zero values for controllability $C$ and severity $S$ denote that the hazards are not safe and have to be mitigated or reduced.

As stated in section 3.1.3, the acceptance criterion for this thesis is chosen as string stability, and the same can be defined as in the section 2.3.1, as

$$\|\Gamma_i(s)\|_{\mathcal{H}_\infty} \leq 1, \quad 2 \leq i \leq m.$$

$\Gamma_i(s)$ is the string stability complimentary sensitivity transfer function mentioned in(2.14). $\|\Gamma_i(s)\|_{\mathcal{H}_\infty}$ is the magnitude of the SSCS transfer function and $m$ is the number of vehicles in the string.

## 4.4 Identification of triggering events

As mentioned in Table 3.3, the triggering events for the algorithm-related hazards were found to neglect the uncertainty and dynamics of the vehicle in the string. This is usually done to reduce the complexity of the system design. A simple vehicle $G_i(s)$ and wireless communication delay $D(s)$ model without any uncertainty in driveline lag $\tau$, internal actuator delay $\phi$ and communication delay $\theta$ is stated in function specification as (4.2) and (4.4). The defined vehicle model does not account for the uncertainties such as $\Delta\tau, \Delta\phi, \Delta\theta$ but rather considers fixed values for $\theta, \phi, \tau$. When the uncertainty in these design parameters is triggered, the system deviates from the defined model dynamics leading to string instability, which can be formulated as

$$\|\Gamma_i(j\omega)\|_{\theta_u,\phi_u,\tau_u} > 1. \tag{4.6}$$

Based on the string instability condition which is critical for a CACC system, the triggering conditions have to be mitigated using functional modifications to the existing CACC system.

## 4.5 Functional modification to reduce SOTIF risk

Based on hazards due to triggering events, functional modification is carried out to the existing CACC system. To capture the uncertainty, a sensitivity analysis is conducted over the given parameters $\theta, \phi$, and $\tau$. The SSCS transfer function mentioned in (2.14) is used for this analysis. Firstly, the communication delay sensitivity is analyzed as it plays a significant role with respect to string stability. The magnitude $\|\Gamma_i(j\omega)\|_{\mathcal{H}_\infty}$ at constant time gap $h = 0.5$s increases with the increase in communication delay $\theta$ implying that a larger communication delay compromises string stability of the controlled system as shown in Figure 4.2. To counteract the string instability caused by the increase in communication delay, a larger minimum headway time $h$ has to be selected.

The internal actuator delay $\phi$ and driveline lag $\tau$ are also seen to exhibit the same characteristics of the communication delay $\theta$. The uncertain $\theta_u, \phi_u$ and $\tau_u$ can be analyzed using minimum and maximum bounds within which they exhibit string-stable behavior, defined as

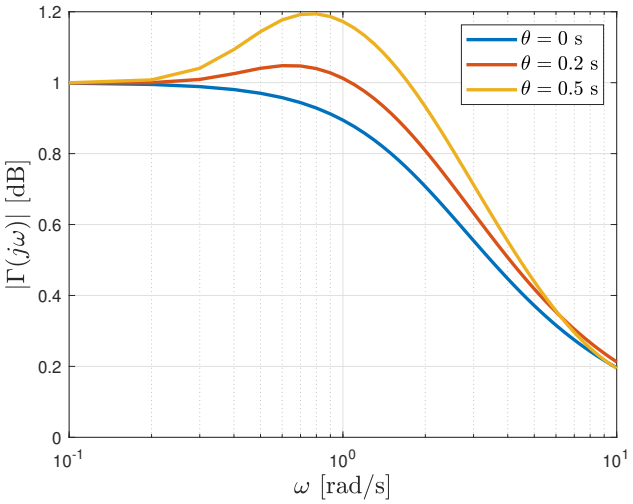



Figure 4.2: (a) Sensitivity analysis for communication delay with $h = 0.5$ s and $k_p = 0.2, k_d = 0.7$, and $k_{dd} = 0$.

$$\theta_u \in [\theta_{min}, \theta_{max}] \tag{4.7}$$

$$\begin{aligned} \phi_u &\in [\phi_{min}, \phi_{max}] \\ \tau_u &\in [\tau_{min}, \tau_{max}] \end{aligned} \tag{4.8}$$

now the uncertain plant or vehicle model and uncertain communication delay in the CACC system are expressed as

$$G_{u,i}(s) = \frac{1}{\tau_{u,i}s + 1} e^{-\phi_{u,i}s} \tag{4.9}$$

$$D_u(s) = e^{-\theta_u s}. \tag{4.10}$$

Further, for analyzing controller stability, all the possible variations of $\tau_u$ and $\phi_u$ (4.8) in the plant model using lumped parameter uncertainty [34] as

$$G_p(s) = G(s)(1 + W_p(s)\Delta_p(s)) \tag{4.11}$$

where $G_p(s)$ denotes the perturbed or uncertain plant which includes the set of all uncertain plants (4.9). $G(s)$ represents the nominal model without any uncertainty as mentioned in (4.2), $W_p(s)$ denotes the weight of lumped uncertainties transfer function and $\Delta_p(s)$ is any stable transfer function which at each frequency is less than or equal to one in magnitude [34]. $W_p(s)$ is expressed as

$$\left| \frac{G_p(j\omega) - G(j\omega)}{G(j\omega)} \right| \leq |W_p(j\omega)|. \tag{4.12}$$

$W_p$ can be calculated using condition (4.12) for for all the variations of $\tau_u$ and $\phi_u$ within the bounds mentioned in (4.8). Using a discrete grid search, Figure 4.3 shows all the possible perturbations of the plant model for $\tau_u$ and $\phi_u$ along with the weighting function $W_p$ represented by the dashed line which

bounds all the perturbations accounting for all the uncertainties in the plant model. The weighting function $W_p(s)$ is represented by a rational transfer function [34] expressed with driveline lag $\tau$ and internal actuator delay $\phi$ uncertainty

$$W_p(s) = \alpha_p \frac{\bar{\tau}s + 1}{\tau_{\min}s + 1} \frac{Ts + 1}{-Ts + 1} - 1, \quad \alpha_p \geq 1 \tag{4.13}$$

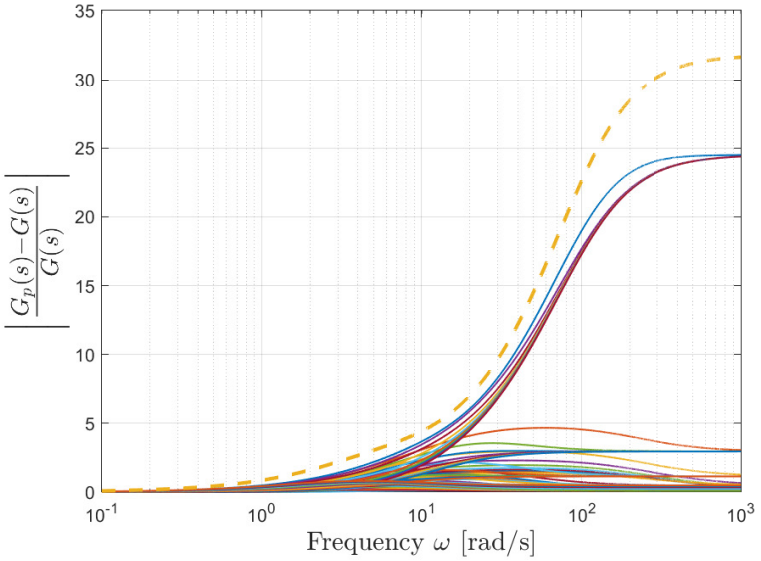



Figure 4.3: Bode magnitude plot of weight function $W_p$ for all plant pertubations.

where $\bar{\tau} = \frac{\tau_{max}+\tau_{min}}{2}$ is the nominal driveline lag and $T = \frac{\phi_{max}-\phi_{min}}{2}$ denotes the internal actuator delay, $\alpha_p$ is the scaling constant. In the same way, all the possible variations of $\theta$ within the bounds (4.7) in the communication delay $D(s)$ (4.10) are found using the delay uncertainty condition [35]

$$D_p(s) = 1 + W_d(s)\Delta_d(s), \tag{4.14}$$

where $D_p$ represents the perturbed delay model which includes all the uncertain delays, $\Delta_d(s)$ is any stable transfer function which at each frequency is less than or equal to one in magnitude [34] and $W_d$ is the weighting function for uncertain communication delays expressed as

$$|W_d(s)| \geq |D_p(s) - 1| \tag{4.15}$$

the weighting function for delay $W_d$ represented by the dashed line in Figure 4.4 bounding all the possible uncertain delays in wireless communication is expressed as a rational transfer function [36]

$$W_d(s) = \alpha_d \frac{2\pi\theta_{max}s}{\pi\theta_{max}s + 1}, \alpha_d \geq 1. \tag{4.16}$$

In the above expression, $\theta_{max}$ is the maximum bound of the communication delay $\theta$ and $\alpha_d$ is a scaling factor. Now, the uncertain plant $G_p$ (4.11) and delay $D_p$ (4.14) are plugged into the SSCS function (2.14), which can be expressed as

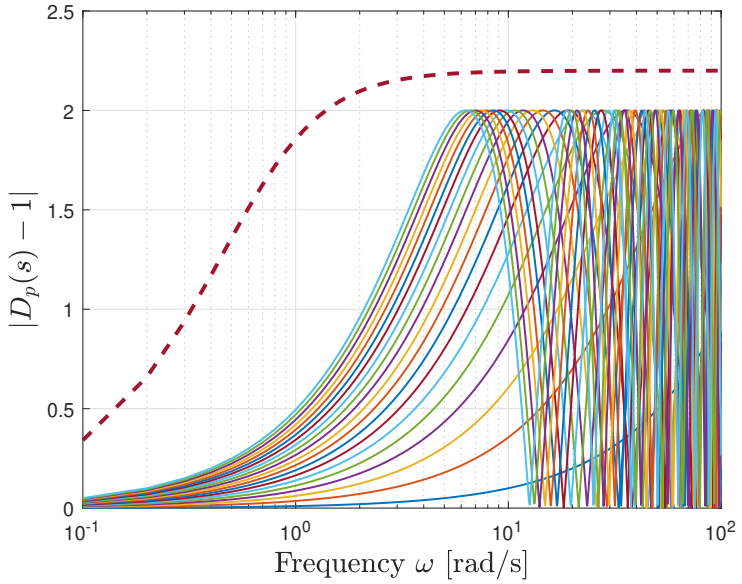



Figure 4.4: Bode magnitude plot of weight function $W_d$ for all delay pertubations.

$$\Gamma_p(s) = \frac{1}{H(s)} \frac{G_{p,i}(s)}{G_{p,i-1}(s)} \frac{D_p(s)s^2 + G_{p,i-1}(s)C(s)}{s^2 + G_{p,i}(s)C(s)}. \tag{4.17}$$

The above equation (4.17), helps in capturing all the uncertainties in communication delay, internal actuator delay, and driveline lag together.

## 4.6 Verification of known scenarios

The functionally modified CACC system is verified based on the acceptance criteria. The acceptance criteria in this thesis is the string stability behavior as explained in the previous sections. The string stability behavior is analyzed using the SSCS with uncertainties $\Gamma_p(s)$ in (4.17). The baseline parameters used for the verification are $\tau = [0.01, 0.5]$, $\phi = [0.01, 0.5]$, $\theta = [0.01, 0.5]$, $h = 0.5$, $k_p = 0.2$ and $k_d = 0.7$. The parameters were chosen based on the sensitivity analysis and widely used values in the literature [17]. In order to capture the known worst case, the possible variations in the $\Gamma_p(s)$ are sampled [1]. However, when the uncertainties in communication delay, internal delay, and driveline lag are plugged into the uncertain SSCS transfer function $\Gamma_p(s)$, it is found that the system becomes unstable, and is not possible to calculate the worst-case upper and lower bounds[2].

The hypothesis for this might be due to the presence of internal actuator lag $\phi$ in the uncertain system. It might also be due to the fact that $G_p$ accounts for internal actuator delay and driveline lag and $D_p$ accounts for the delay in communication, but all three uncertainties are not considered together in a separate weighting function or perturbed model. Further investigation is necessary to bound all the uncertainties using (4.17). From the sensitivity analysis, it can also be found that any uncertainty in

[1] *usample* - Robust control toolbox command in Matlab
[2] *wcgain* - Robust control toolbox command in Matlab

the $\theta, \phi$, and $\tau$ leads to string instability. This is also due to the choice of time gap $h$. Therefore a suitable time gap $h$ value should be chosen to achieve string stability with an uncertain system.

This mathematical analysis approach helps in formulating the qualitative SOTIF assessment. This approach can also be helpful in deriving proper safety requirements during the earlier stages of ADAS development.

## 4.7 Summary

In this chapter, the mathematical analysis of the conventional SOTIF assessment for the CACC system was described in detail. The algorithm-related hazards and their influence on the string stability were captured in mathematical expressions. The shortcomings of neglecting the uncertainties in the model were discussed in the identification of triggering events. Followed by a sensitivity analysis to find the minimum and maximum bounds of $\theta, \phi$ and $\tau$. The bounds were used in the uncertain plant and delay models to capture the uncertainties. Finally, the verification of known scenarios was performed using the uncertain SSCS and it was found that it could not find the worst-case bounds for uncertainty. The hypothesis for this might be due presence of internal actuator lag or the lack of a separate weighting function combining all three uncertainties. Having addressed the first research objective, the second research objective of integration of real-time system in the experimental vehicle will be discussed in the next chapter.

# 5 Automated platform in CarLab

This chapter addresses the second research objective of this thesis, integrating dSPACE as a real-time operating system in Carlab for safely testing automated driving algorithms. This chapter starts by introducing the existing experiment vehicle CarLab along with its existing architecture. The choice of dSPACE as the platform is motivated and the integration of the real-time system is described in detail.

## 5.1 CarLab architecture

The subject experiment vehicle also known as CarLab is an instrumented Toyota Prius Mk4, a well-known hybrid electric vehicle (HEV). The architecture of the vehicle is derived from the Openpilot by comma.ai [37] open-source reverse-engineered DBC file. The current architecture consists of five different electronic control units (ECU) that are identified and one unidentified ECU namely

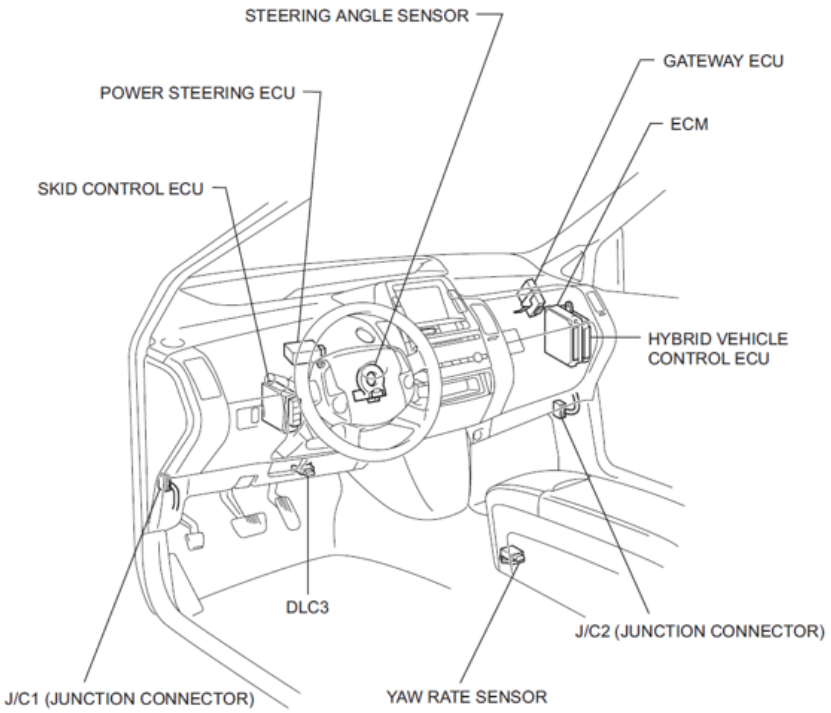



Figure 5.1: ECUs in the Toyota Prius experiment vehicle [9]

1. Driving Support Unit (DSU)
2. Hybrid Control Unit (HCU)
3. Electric Power Steering (EPS)
4. Intelligent Parking Assist System (EPAS)
5. Central Gateway Unit (CGW)
6. Unidentified ECU (XXX).

The location of the ECUs identified can be seen in the Figure 5.1 and 5.3. The unidentified ECU (XXX) consists of various signals that are expected to be grouped with other identified ECUs . These ECUs help in carrying out the various stock ADAS functions such Lane Keeping Assist System (LKAS), IPAS, and ACC. In addition to the stock sensor suite, the vehicle is also equipped with a GNSS + INS device $RT3000$ v3 [38]. This device contains an inertial measurement unit (IMU) which comprises accelerometers and gyroscopes. The $RT3000$ v3 provides data with respect to the different states of the vehicle [39].

## 5.2 Need for dSPACE in Prius

Currently, in order to deploy the ADAS functions and control algorithms under development an external device known as the *"comma two: Panda"* powered by *"open pilot"* [37] is used. The *Panda* is hardware with a universal car interface that helps to access the CAN bus of the vehicle shown in Figure 5.2(a). The *Panda* is plugged into the OBDII port of the vehicle from where the data is transmitted to the PC via the USB. Then specific Python libraries are used to extract the raw CAN data. Since the main idea behind *Panda* is just to enhance the existing vehicles with basic automated driving functions such as ACC and LKA, it lacks the ability of an automated platform that gives access to advanced simulation environments and hardware in the loop testing.

(a) (b)

Figure 5.2: (a) Comma two: Panda (*Open pilot*) (b) MicroAutoBox III (*dSPACE*)

The aim is to replace the *Panda* with a dSPACE MicroAutoBoxIII shown in Figure 5.2(b) system which will act as the real-time operating system in the vehicle for testing the developed control algorithms. The MicroAutoBoxIII is equipped with state of art automotive communication protocols such as Controller Area Network (CAN2.0) and Controller Area Network-Flexible data rate (CAN-FD) which are more standardized for automotive when compared with the USB protocol. dSPACE [40] has its own set of hardware and software suites with comprehensive applications for developing, integrating, testing, data logging, and visualization of ADAS functions and control algorithms. The need for dSPACE in the CarLab is due to the intricate interplay of different sensors, algorithms, and control systems. The real-time operating device sets the foundation for rapid testing and prototyping of control algorithms in the experiment vehicle. This step will also help in overcoming the latency and synchronization issues when testing the algorithms.

dSPACE consists of dedicated software applications like ConfigurationDesk, ControlDesk, and BusManager for handling applications of Rapid Control Prototyping (RCP) developments to hardware in the loop tests, including the implementation of behavior models in Simulink and I/O function code to the vehicle hardware. It also helps in managing the signal paths between the external devices and behavior model interfaces controlling the entire process for the generation of real-time code. In addition to this, dSPACE provides the means for ensuring safety requirements and standards (ISO26262 [11] and ISO21448 [5]) by providing comprehensive testing and validation capabilities.

## 5.3 dSPACE Integration with CarLab

The MicroAutoBoxIII is mounted in the Prius along with the $RT3000$ v3. It is powered by the vehicle's battery through the 12V distribution box on the rear end of the vehicle. The CAN bus of $RT3000$ v3 is connected with the MicroAutoBoxIII using the DB9 to DB9 connector. $RT3000$ v3 is first configured for data transfer through the CAN2.0 and the DBC file is extracted using the proprietary software. This DBC file is fed to real-time application configuration for establishing the communication. The vehicle CAN network is accessed using the OBD II connector at the bottom of the glove box. An OBD II to DB9 cable is used for establishing the connection between the vehicle and MicroAutoBoxIII as shown in 5.4. The real-time application is controlled using the host PC which is connected to MicroAutoBoxIII using ethernet. The new vehicle architecture after the integration of dSPACE is shown in Figure 5.3.

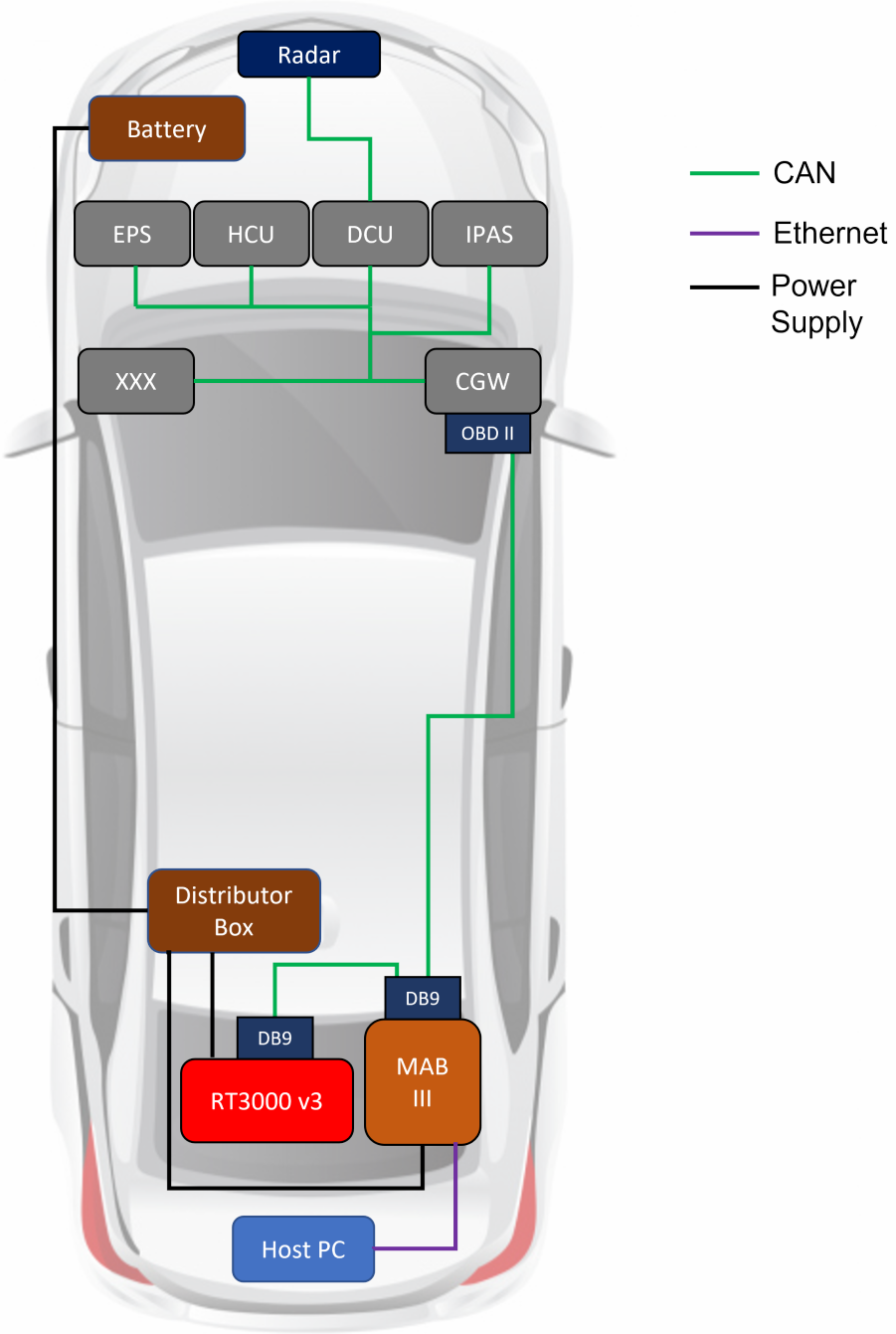



Figure 5.3: New Carlab architecture

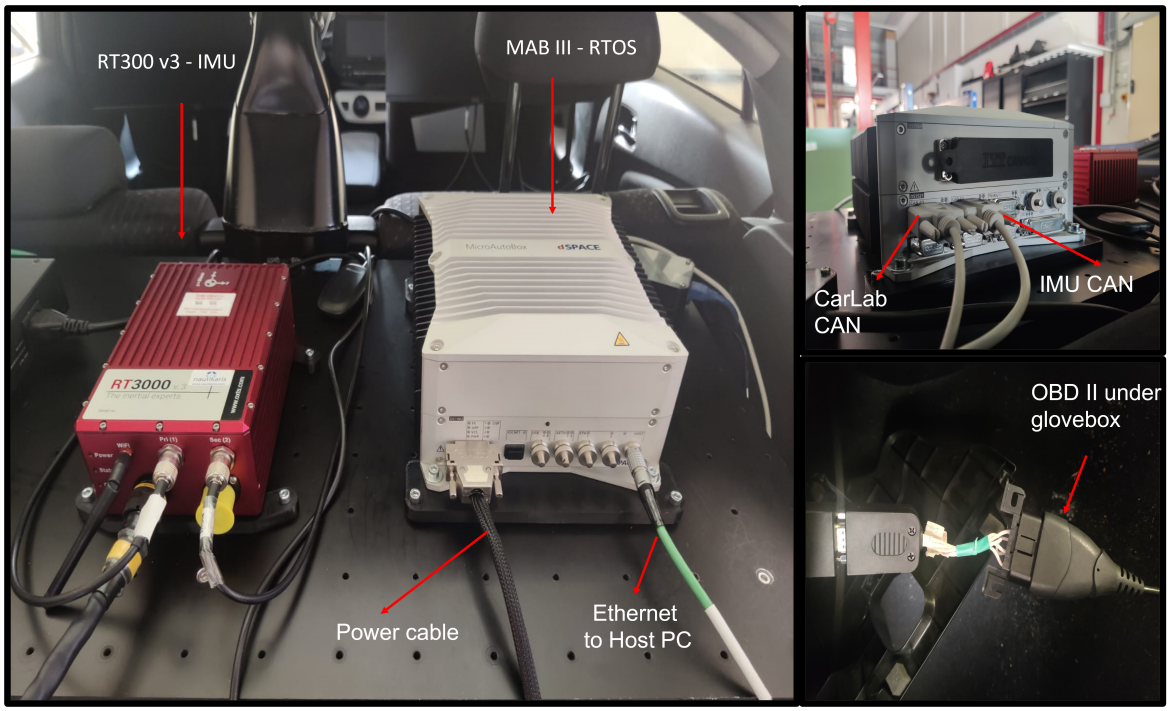



Figure 5.4: Left image: RT3000 v3 and MicroAutoBoxIII mounted on CarLab, Right top image: Backside of MicroAutoBoxIII with two CAN channels for CarLab and IMU, Right bottom image: OBD II connection under the glovebox for connecting with the CarLab bus network.

## 5.4 Pipeline for testing and validation

A new pipeline for testing the developed control algorithms is illustrated in Figure 5.5. The pipeline outlines the steps involved in developing and deploying an ADAS function using dSPACE tools.

1. **Configuring DBC file:** The first step involves configuring the DBC file of the Toyota Prius. The DBC file contains information about the messages and signals in a CAN bus. It defines how data is exchanged between various ECUs on the vehicle's CAN bus. It includes details about the messages, signal names, data types, scaling factors, and more. The signals required for the test and the related ECUs are identified using the signal identifiers.

2. **Behaviour model for ADAS function:** The ADAS function is modeled using a Simulink environment. This model is known as the behavior model as it captures the functions that have to be carried out along with responses to the design parameters. The behavior model is usually a physics-based mathematical model replicating the actual system.

3. **Real-time application in ConfigurationDesk:** The behavior model is imported into ConfigurationDesk where the input/output functions of the model are identified. The connection between the I/O functionality in ConfigurationDesk and Simulink models is realized via model interfaces. Model interfaces are model port blocks that act as the channel between the ConfigurationDesk model and the Simulink model. Any changes in the Simulink model or ConfigurationDesk model can be synchronized using the model analysis function. It also creates a signal

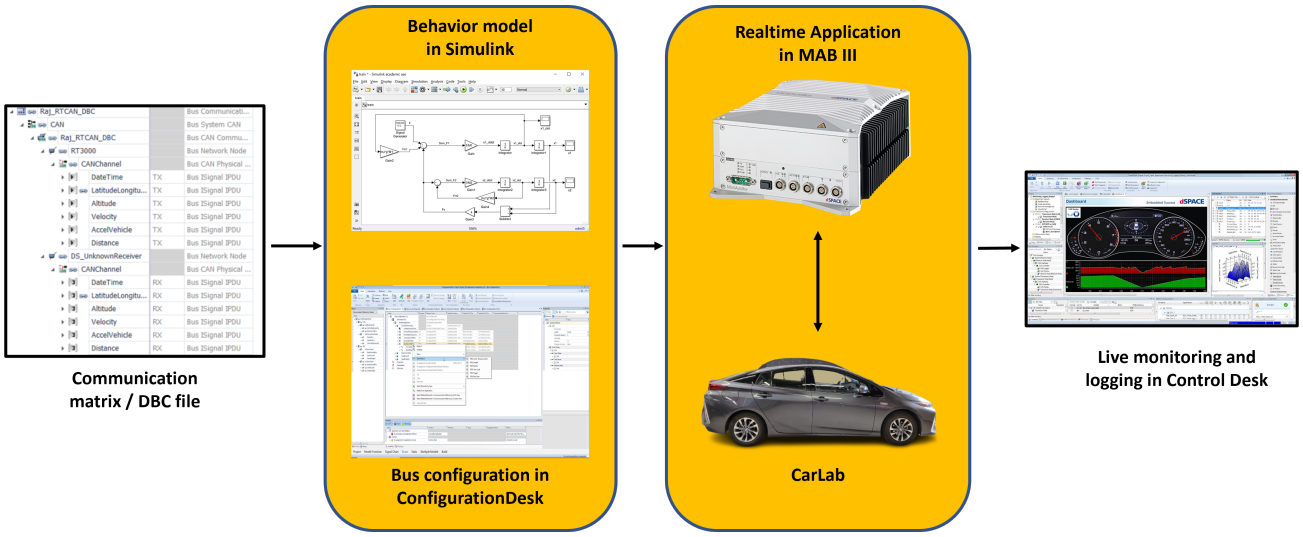



Figure 5.5: Pipeline for rapid control prototyping in CarLab using dSPACE

chain that helps the user visualize and control the interactions of the Simulink model with the MicroAutoBoxIII and the vehicle hardware.

The bus network is configured by importing the intended DBC file and the necessary ECUs containing the relevant signals are selected for the bus simulation. The model access is enabled through which the function can controlled in the experimenting software ControlDesk. Then the suitable communication protocol CAN2.0 is selected and the right hardware channel is also chosen. MicroAutoBoxIII has six CAN2.0/CAN-FD channels. The baud rate selected for the Prius and $RT3000$ v3 is $500kbit/s$. Finally, the bus access is configured to enable communication between the MicroAutoBoxIII and the vehicle.

After the bus network configuration, the application is checked for any errors or warnings in ConfigurationDesk and finally sent for building the real-time application of the ADAS function. The build operation yields two files with $application.rta$ and $application.sdf$ format, the former is used for deploying the real-time application on MicroAutoBoxIII while the latter is used for experimenting with the real-time application in ControlDesk.

4. **Deploying of real-time application to MicroAutoBoxIII in CarLab:** Once the build operation is finished, the $application.rta$ file is automatically mounted to the real-time hardware MicroAutoBoxIII. The real-time application on the MicroAutoBoxIII can be controlled using a web interface or a host computer.

5. **Live monitoring and data logging in ControlDesk:** Another way of testing the developed and built real-time application is to import it into ControlDesk. ControlDesk is a dSPACE tool for real-time experiment control and visualization. It provides a platform for the developer to monitor and interact with the running ADAS function and other components in real time. Finally, it also helps to visualize, log, and replay data from the experiment. It even allows manual interventions if necessary, this way the required functionalities of the ADAS function can be inspected thoroughly.

The pipeline is set in a way that it covers the stages of modeling the ADAS function to testing. The

detailed description of configuring the model in ConfigurationDesk and bus network configurations are explained in [41].

## 5.5 Testing communication with RT3000 v3 and CarLab CAN bus

After the hardware integration, using the new pipeline, the communication with RT3000 v3 and CarLab is established. As discussed earlier, both the DBC files are configured [41] and the real-time application was built using the ConfigurationDesk. Then the real-time application is loaded in ControlDesk, where the required signals such as steer angle, acceleration, engine rpm from the CarLab network, and acceleration signals from the $RT3000$ v3 are selected to be visualized from the sake of the experiment as shown in Figure 5.6. When the bus is simulated from ControlDesk, the tool mounts the real-time application on MicroAutoBoxIII and starts the communication. This is indicated by the LEDs on the CAN channels on the MicroAutoBoxIII. A monitor tool is used in the ControlDesk application for monitoring the selected signals. One major advantage of the dSPACE interface is that it can directly output physical signal values along with the CAN signals in decimal or hexadecimal values. Specific signals can also be filtered for inspection purposes. Finally, the data from the experiment can be saved into a buffer which can exported as $.mat$ files for postprocessing or analysis. Having access to the CarLab's CAN offers numerous advantages for development, diagnostics, and customization.

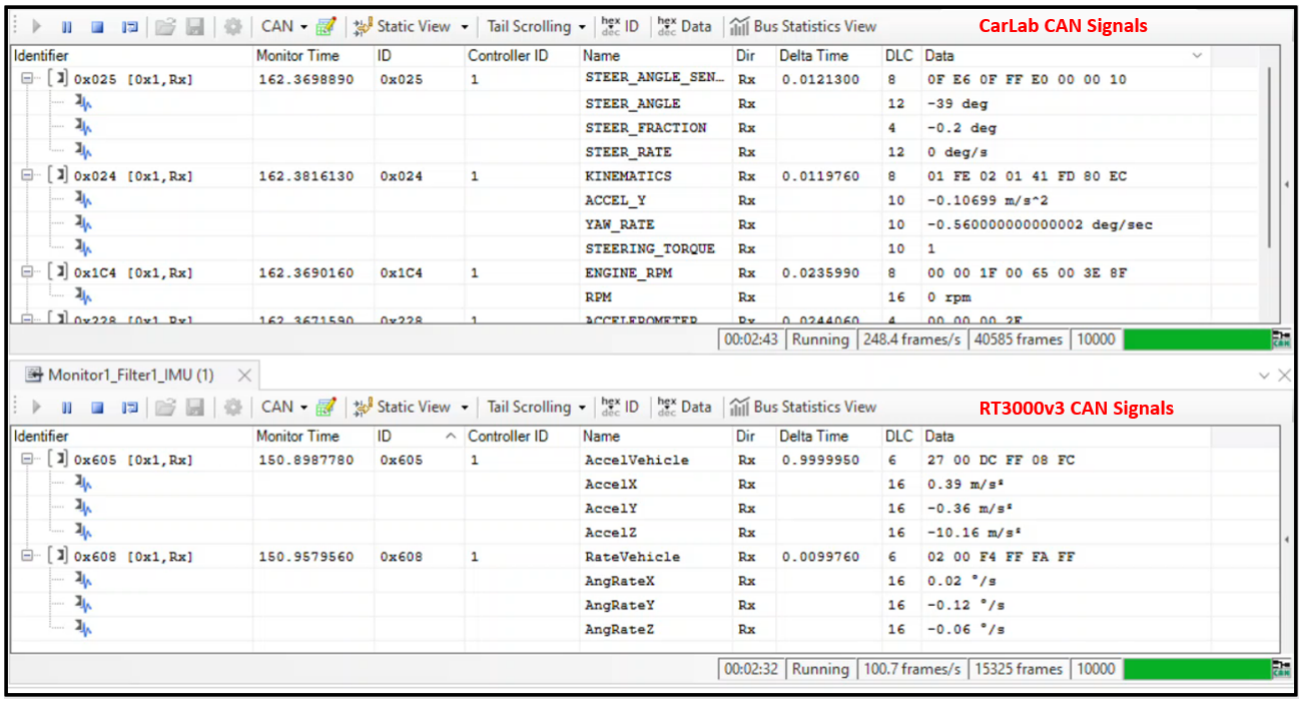



Figure 5.6: CAN signals from CarLab and RT3000v3 bus network visualized in ControlDesk

To this end, access to the vehicle and $RT3000$ v3 CAN bus has been established. All the signals from the identified ECUs can be received, visualized, and exported using the dSPACE ControlDesk. Transmitting signals to the CAN bus has to be studied and investigated further in order to successfully test automated driving systems.

## 5.6 Summary

This chapter shows the first steps towards integrating dSPACE MicroAutoBoxIII as the real-time system in the Carlab. Firstly, the existing architecture of the vehicle was studied using the available DBC file and existing hardware. Later, the MicroAutoBoxIII was connected and configured with the vehicle and IMU CAN bus for establishing communication.

# 6 Conclusions and Recommendations

This chapter discusses the conclusions drawn based on the work done in this thesis and recommendations for future work.

## 6.1 Conclusions

One of the objectives of the project is to propose a mathematical analysis approach for SOTIF assessment for CACC to overcome the conventional qualitative method of SOTIF assessment in words. To that end, the conventional SOTIF analysis is first conducted in Chapter 3, and later in Chapter 4, the mathematical analysis is proposed.

First, through a brief literature study the ISO 21448 standards and the use case CACC is introduced. A heterogeneous CACC system with uncertainties in communication delay $\theta$, internal actuator delay $\phi$, and driveline lag $\tau$ is considered throughout the research. The conventional SOTIF assessment is done based on the ISO 21448 standard. The CACC SOTIF-related hazards related to algorithm limitation, sensor/hardware limitation, and misuse of functionality are identified. String stability is chosen as the acceptability criterion for the identified hazards. The risk due to the hazards is evaluated using the influence of time gap $h$ with non-zero controllability and severity. The usage of simplified model dynamics with fixed values of $\theta, \phi$, $\tau$ and without any uncertainty is found as the triggering event for the algorithm-related hazards. Finally, the SOTIF-related hazards for CACC are considered to be mitigated by accounting for all the uncertainties in system modeling and also triggering a warning to the fallback driver to take over authority. However, this conventional way of assessment is only backed up with evidence from the literature. To overcome this, the conventional SOTIF assessment is mathematically analyzed using uncertainty modeling for algorithm-related hazards. The hazards, risks, and triggering events related to algorithm limitations are mathematically formulated. An uncertain CACC model is used as the functionally modified system for achieving SOTIF. The uncertainties such as communication delay $\theta$, internal actuator delay $\phi$, and driveline lag $\tau$ are captured using the weighting function $W_p$ and $W_d$. These weighting functions are used to formulate a new uncertain SSCS function which accounts for all the uncertainties in plant and delay. However, when the uncertainties in communication delay, internal delay, and driveline lag are plugged into the uncertain SSCS transfer function, it is found that the system becomes unstable, and is not possible to calculate the worst-case upper and lower bounds. This is hypothesized due to the presence of internal actuator delay $\phi$ in the uncertain system and also due to the lack of a separate weighting function or perturbed model which accounts for all the uncertainties in plant and delay together.

Therefore this approach of mathematically analyzing the SOTIF assessment can help in formulating the qualitative SOTIF assessment and also in deriving safety requirements in early stages of ADAS development.

The second objective of this project is to integrate dSPACE MicroAutoBox III as the real-time operating system in the existing experimental vehicle, Carlab. The need for the real-time operating system is to safely test the developed automated driving systems and control algorithms. dSPACE is selected as the real-time system for its ability in rapid control prototyping and standard automotive communication protocols like CAN2.0/CAN-FD. CAN 2.0 protocol is used to establish communication between the dSPACE MAB III and Carlab. However, within the project tenure, only the hardware integration and access to the signals from the identified ECUs in the vehicle CAN bus are done. A first attempt has been made to analyze the vehicle CAN-bus structure to this end, where we were able to receive all the identified signals from the vehicle.

## 6.2 Recommendations

Further research and investigation for bounding all the uncertainties with the CACC system using worst-case upper and lower bounds would be beneficial. The mathematical analysis of SOTIF assessment on CACC in this thesis is only applied to algorithm-related hazards. The same analysis can used for sensor/hardware-related hazards by accounting for the calibration errors, and detection errors like false positives. The CACC system with uncertainty is found to exhibit string instability due to having a constant time gap. This can be solved by choosing a larger time gap $h$ which accounts for all the uncertainties or having a varying time gap. However, for implementing varying time gaps, the transient states from one-time gap to another have to be investigated in detail.

For the integration of dSPACE in Carlab, transmitting the signals through MAB III to the vehicle can be achieved by more research on the architecture of ECU nodes in the vehicle. Identifying the right CAN messages to be sent to the ECUs might help in transmitting the right signals. Further, the correlation of the signals received between RT3000 v3 and Carlab can also be studied for better accuracy and setting up ground truth. The work can also be further extended with the design of functional architecture with proper abstraction between hardware, perception, planning, control, data logging, and fault diagnosis components.

# Bibliography

[1] Thomas Winkle. Product development within artificial intelligence, ethics and legal risk: Exemplary for safe autonomous vehicles. 2021.

[2] Stephan Reinhofer, Markus Ernst, Jürgen Fabian, and Adam Schnellbach. Analysis and development of fail operational automotive mechatronic systems. pages 36–42, 2015.

[3] Franciscus Nicolaas Hoogeboom. Safety of automated vehicles: design, implementation, and analysis. 2020.

[4] Abhishek Balakrishna Bhat. Assessment and preliminary vehicle-level architecture design of an sae level 4 rated vehicle platform. October 2022. EngD thesis. - Confidential.

[5] ISO. Road vehicles – safety of the intended functionality. 2019.

[6] Adam Schnellbach and Gerhard Griessnig. Development of the iso 21448. pages 585–593, 2019.

[7] Ovidiu Vermesan, Roy Bahr, Reiner John, Marco Ottella, Robin Gjølstad, Ole Buckholm, and Hans-Erik Sand. Advancing the design of fail-operational architectures, communication modules, electronic components, and systems for future autonomous/automated vehicles. pages 53–71, 2020.

[8] Erjen Lefeber, Jeroen Ploeg, and Henk Nijmeijer. Cooperative adaptive cruise control of heterogeneous vehicle platoons. *IFAC-PapersOnLine*, 53(2):15217–15222, 2020.

[9] Jérôme Maye and Mario Krucker. Communication with a toyota prius.

[10] Santokh Singh. Critical reasons for crashes investigated in the national motor vehicle crash causation survey. 2015.

[11] ISO. Road vehicles – Functional safety. (ISO 26262), 2011.

[12] Mahmoud Abdelhady and Alireza Moayyedi. Design and implementation of an infrastructure-independent automated valet parking system into a toyota prius. October 2022. EngD thesis.

[13] Hua Zhou, Xiaoyan Li, Xia He, Pingfei Li, Lingyun Xiao, and Daowen Zhang. Research on safety of the intended functionality of automobile aeb perception system in typical dangerous scenarios of two-wheelers. *Accident Analysis & Prevention*, 173:106709, 2022.

[14] Mingyue Yan, Wuwei Chen, Qidong Wang, Linfeng Zhao, Xiutian Liang, and Bixin Cai. Human–machine cooperative control of intelligent vehicles for lane keeping—considering safety of the intended functionality. In *Actuators*, volume 10, page 210. MDPI, 2021.

[15] Victor Dolk, Jos den Ouden, Sander Steeghs, Jason Gideon Devanesan, Irfan Badshah, Adityen Sudhakaran, Koos Elferink, and Debayan Chakraborty. Cooperative automated driving for various traffic scenarios: Experimental validation in the gcdc 2016. *IEEE Transactions on Intelligent Transportation Systems*, 19(4):1308–1321, 2018.

[16] Tom van der Sande and Henk Nijmeijer. From cooperative to autonomous vehicles. *Sensing and Control for Autonomous Vehicles: Applications to Land, Water and Air Vehicles*, pages 435–452, 2017.

[17] Jeroen Ploeg, Bart T. M. Scheepers, Ellen van Nunen, Nathan van de Wouw, and Henk Nijmeijer. Design and experimental evaluation of cooperative adaptive cruise control. pages 260–265, 2011.

[18] ISO. Road vehicles — functional safety — part 9: Automotive safety integrity level (asil)-oriented and safety-oriented analyses. 2019.

[19] Arash Khabbaz Saberi. Functional safety: A new architectural perspective: Model-based safety engineering for automated driving systems. September 2020. Proefschrift.

[20] Christopher Becker, John C Brewer, Larry Yount, et al. Safety of the intended functionality of lane-centering and lane-changing maneuvers of a generic level 3 highway chauffeur system. 2020.

[21] D ENSEMBLE. 2.10 iterative process document and item definition. 2018.

[22] Dietmar Kinalzyk. Sotif process and methods in combination with functional safety. pages 612–623, 2021.

[23] Jeroen Ploeg, Dipan P. Shukla, Nathan van de Wouw, and Henk Nijmeijer. Controller synthesis for string stability of vehicle platoons. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):854–865, 2014.

[24] Shahab Sheikholeslam and Charles A Desoer. Longitudinal control of a platoon of vehicles with no communication of lead vehicle information: A system level study. *IEEE Transactions on vehicular technology*, 42(4):546–554, 1993.

[25] Elaine Shaw and J Karl Hedrick. String stability analysis for heterogeneous vehicle strings. pages 3118–3125, 2007.

[26] Rajesh Rajamani and Chunyu Zhu. Semi-autonomous adaptive cruise control systems. *IEEE Transactions on Vehicular Technology*, 51(5):1186–1192, 2002.

[27] Jeroen Ploeg. Analysis and design of controllers for cooperative and automated driving. 2014.

[28] Max Bolderman, AAJ Erjen Lefeber, Jeroen Ploeg, and Henk Nijmeijer. Observer-based control for string stable cacc within heterogeneous vehicle platoons. 2020.

[29] Jeroen Ploeg, Elham Semsar-Kazerooni, Guido Lijster, Nathan van de Wouw, and Henk Nijmeijer. Graceful degradation of cooperative adaptive cruise control. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):488–497, 2015.

[30] On-Road Automated Driving (ORAD) Committee. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, apr 2021.

[31] Automated Vehicle Safety Consortium et al. Avsc best practice for describing an operational design domain: Conceptual framework and lexicon. *SAE Industry Technologies Consortia*, 2020.

[32] Cong Wang and Henk Nijmeijer. String stable heterogeneous vehicle platoon using cooperative adaptive cruise control. pages 1977–1982, 2015.

[33] Jeroen Ploeg, Nathan Van De Wouw, and Henk Nijmeijer. Lp string stability of cascaded systems: Application to vehicle platooning. *IEEE Transactions on Control Systems Technology*, 22(2):786–793, 2013.

[34] Sigurd Skogestad and Ian Postlethwaite. Multivariable feedback control: analysis and design. 2005.

[35] Haitao Xing, Jeroen Ploeg, and Henk Nijmeijer. Compensation of communication delays in a cooperative acc system. *IEEE Transactions on Vehicular Technology*, 69(2):1177–1189, 2019.

[36] Munther A. Dahleh Mohammed Dahleh and George Verghese. George verghese lectures on dynamic systems and control.

[37] Commaai. Comma.ai - openpilot, an open source driver assistance system. *GitHub*.

[38] OxTs. Rt3000 v3 - the industry standard gnss/ins for adas and autonomous vehicle testing.

[39] SS Shetty. Development of a digital twin of a toyota prius mk4. 2022.

[40] dSPACE. Autonomous driving solutions.

[41] TU/e Automotive Laboratory. Carlab. `https://gitlab.tue.nl/20215152/prius-carlab`, 2023.

# A Project management

This chapter describes the project management plan to identify measures for monitoring, controlling, and executing the project.

## A.1 Stakeholder analysis

Mendelow's matrix analysis is used to identify the different direct and indirect stakeholders associated with this project. The identified stakeholders are placed in respective quadrants based on their level of interest and influence in the project decisions and outcomes. This analysis helps in managing the stakeholders and prioritizing their needs. The stakeholder's analysis used in this project is shown in Figure A.1.

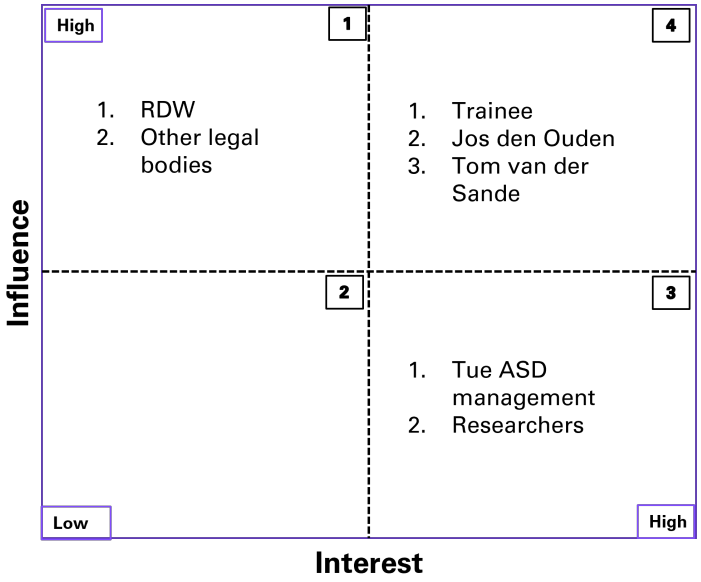



Figure A.1: Stakeholders analysis

## A.2 Project plan

In the project's initial phase, several discussions were conducted to understand the goals of this project. The goals are transformed into milestones. Regular bi-weekly meetings with the supervisors were scheduled with supervisors to identify possible risks and gain useful insights into the project. Finally, project steering committee meetings were scheduled for every 5-6 weeks with the supervisors and program manager of the EngD program to review and evaluate the progress of the project. Figure A.2 shows the overall project plan with the respective milestones identified and realized along different stages of the project

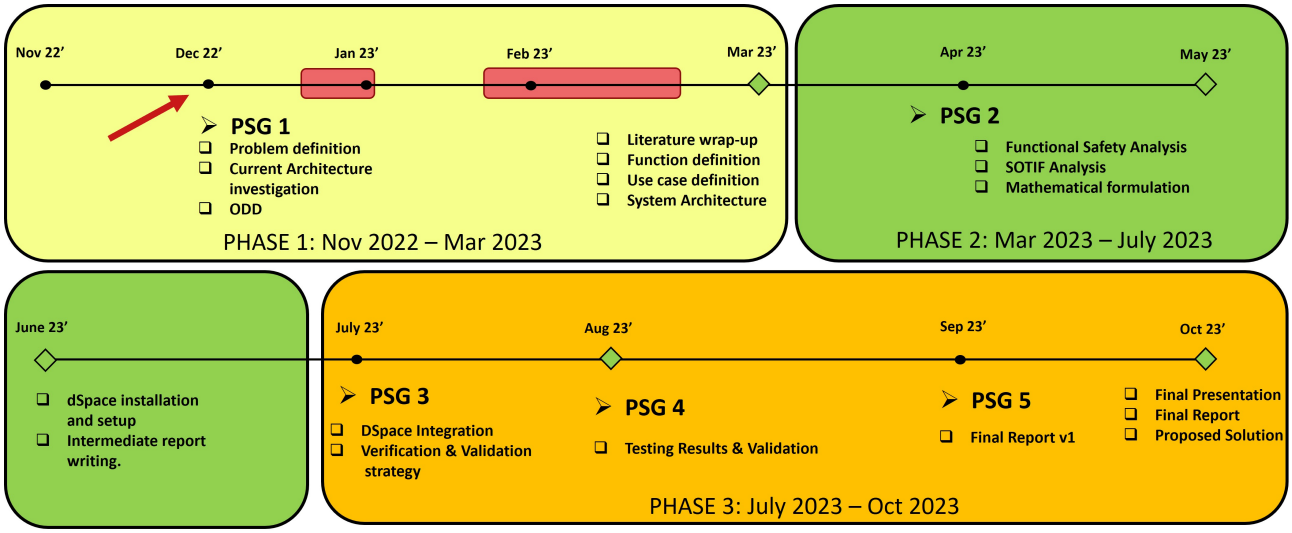



Figure A.2: Project plan

## A.3 Risk management plan

Identification of risks in the project plays a vital role in timely completion and taking necessary measures against any uncertainties associated with it. The risks were brainstormed during the biweekly progress meetings and steering committee meetings. The risk management plan followed for this project is shown is Figure A.3

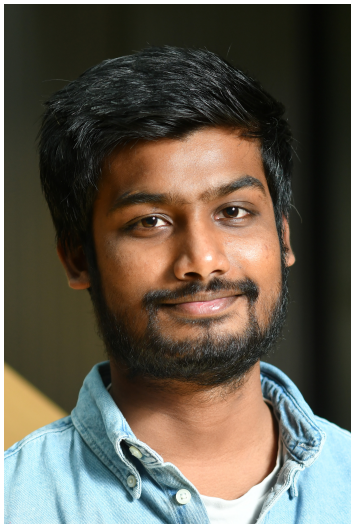
| S.No | Risk | Category | Mitigation/Contingency plan |
|---|---|---|---|
| 1 | Design of high-fidelity models | Design | Started with very simple model and adding complexity step by step. |
| 2 | Time and scope for dSpace integration | Verification and validation | Simultaneous dSpace setup and validation using dSpace. |
| 3 | Uncertainties with dSPACE software and hardware | Verification and validation | dSPACE support & help forums. Hardware support from experts. |
| 4 | Real vehicle testing | Verification and validation | Validation through dSpace or simulation. |

Figure A.3: Risk management plan

## A.4 Project Deliverables

It is agreed with the supervisors that all the project-related code, simulation, and testing data will be stored in a Git repository [41]. The repository shall also contain the manual for the dSPACE setup and configuration. The repository will be shared with the supervisors after the completion of the project.

# About the author



Raj Kumar Muniyandi received his B.E in Automobile Engineering (2016) at Anna University, India, and MSc in Vehicle Engineering (2020) at Delft University of Technology, The Netherlands. During his studies, he specialized in dynamics and controls and finished his graduation, Electric vehicle platooning at signalized intersections in collaboration with the Research Institute of Highways in China, and Rijkswaterstaat, Rijswijk. Currently, he is pursuing the Engineering Doctorate degree in Automotive Systems Design at TU/e where he carries out his individual project at the Automotive lab in TU/e. The project concerns contributing to the safety of automated vehicles and integrating dSPACE as the real-time system in the existing experimental vehicle. The project is titled Analysis of a safe and reliable automated driving platform. His career interest lies in the development of automated driver assistance systems (ADAS), model-based design (MBD), and functional safety of road vehicles.

PO Box 513
5600 MB Eindhoven
The Netherlands
tue.nl

**EngD AUTOMOTIVE SYSTEMS DESIGN**
**Track AUTOMOTIVE SYSTEMS DESIGN**

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY