# Safe Halt as Fail-safe Concept for Automated Driving Systems

Vom Fachbereich Maschinenbau an der
Technischen Universität Darmstadt
zur Erlangung des Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte

# Dissertation

vorgelegt von

**Stefan Martin Ackermann M.Sc.**

aus Hagen

| | |
|---|---|
| Berichterstatter: | Prof. Dr. rer. nat. Hermann Winner |
| Mitberichterstatter: | Prof. Dr.-Ing. Markus Maurer |

| | |
|---|---|
| Tag der Einreichung: | 29.11.2022 |
| Tag der mündlichen Prüfung: | 21.02.2023 |

Darmstadt 2023

D 17

Stefan Martin Ackermann M.Sc.: Safe Halt as Fail-safe Concept for Automated Driving Systems
Darmstadt, Technische Universität Darmstadt
Tag der mündlichen Prüfung: 21.02.2023

# Preface

This thesis was written during my activity as a research associate at the Institute of Automotive Engineering (FZD) of the Technical University of Darmstadt (TUDa). This research was conducted within the scope of the UNICAR*agil* project (Förderkennzeichen (FKZ) 16EMO0286). I gratefully acknowledge the financial support for the project from the German Federal Ministry of Education and Research (BMBF). Without this financial support, this thesis would not have been possible. Additionally, I would like to thank Valeo Schalter und Sensoren GmbH for providing ultrasonic sensors, cameras, and Electronic Control Unit (ECU) for the vehicle prototypes.

My special thanks go to my doctoral supervisor Mr. Prof. Dr. rer. nat. Hermann Winner. Your enthusiasm for the research topic, the time you invested in supervising my research activities, and your outstanding remarks in our consultations contributed significantly to the success of this thesis.

Furthermore, I would like to thank Mr. Prof. Dr.-Ing. Markus Maurer from the Institute of Control Engineering at the Technical University of Braunschweig for taking over the role as second examiner.

I would also like to express a big thank you to all former and current employees at FZD. I have always enjoyed our cooperation at FZD. The cohesion of the employees has always been remarkable. The great discussions we had contributed significantly to this thesis. I wish all current and future FZDers the same excellent working atmosphere I was privileged to experience.

In addition, I also want to thank all the students I have supervised during my time at FZD. Our collaboration was very meaningful for me. It was a pleasure to work with you all. In particular, I would like to thank all the research assistants who helped me integrate the Safe Halt components into the UNICAR*agil* vehicles. Your dedication and quality of work have allowed me to allocate time for my research.

My deepest gratitude goes to my family for their encouragement and support throughout the years. You have always supported me, even in difficult phases. I will always be grateful to you for backing me up even during these phases. Your sympathy and trust helped me a lot in writing this thesis.

Finally, I am thankful for all my friends, especially the “Orscheler”, some of whom have accompanied me for many years. I sincerely thank all the dear people in my life for the many beautiful moments and laughs. You mean a lot to me.

Darmstadt, November 2022

# Table of Contents

# List of Symbols and Indices

## Latin formula symbols

| Symbol | Unit | Description |
|---|---|---|
| $a$ | $\mathrm{m/s^2}$ | Acceleration |
| C | - | Spline Parameter |
| $f$ | - | Fault Combinations |
| E | m | ENU East |
| $h$ | m | Height |
| $i$ | - | Increment Variable |
| $k$ | - | Factor |
| $L$ | m | Length |
| $m$ | - | Mission |
| N | m | ENU North |
| $n$ | - | Number of |
| $q$ | - | Quality |
| $s$ | m | Path Length |
| $t$ | s | Time |
| $u$ | - | Spline Parameter |
| $o$ | - | Operability |
| $p$ | - | Performance |
| $v$ | $\frac{\mathrm{m}}{\mathrm{s}}$ | Velocity |
| $X$ | m | Cartesian Coordinate X |
| $Y$ | m | Cartesian Coordinate Y |
| $Z$ | m | Cartesian Coordinate Z |

## Greek formula symbols

| Symbol | Unit | Description |
|---|---|---|
| $\beta$ | rad | Sideslip Angle |
| $\lambda$ | rad | Longitude Angle |
| $\phi$ | rad | Latitude Angle |
| $\psi$ | rad | Yaw Angle |

## Calligraphic symbols and fraktur characters

| Symbol | Unit | Description |
|---|---|---|
| $\mathscr{M}$ | - | Driving Missions |
| $\mathscr{F}$ | - | Independent Faults |
| $\mathscr{P}$ | - | Possible Fault Combinations |

## Indices

| Symbol | Description |
|---|---|
| 1 2 3 | Integer |
| a | Available |
| acc | Acceleration |
| ADS | Automated Driving System |
| AV | Automated Vehicle |
| brake | Brake |
| centr | Centripetal |
| comp | Comparable |
| decel | Deceleration |
| diff | Difference |
| dist | Distance |
| E | ENU East |
| ego | Ego Vehicle |
| ENU | East-North-Up (ENU) |
| geod | Geodesic |
| intersec | Intersection |
| lat | Lateral |
| long | Longitudinal |
| m | Mission |
| max | Maximum |
| min | Minimum |
| N | ENU North |
| nec | Necessary |
| nominal | Nominal |
| obj | Object |
| pathl | Path Length |
| poly | Polygon |
| psi | Psi |
| relev | Relevant |
| safe | Safe |
| set | Setpoint |

## Accents and Operators

| Symbol | Description |
|---|---|
| $\dot{s}$ | Velocity |
| $\ddot{s}$ | Acceleration |
| $\vec{r}$ | Location Vector |

# List of Abbreviations

| | |
|---|---|
| ACC | Adaptive Cruise Control |
| ADAS | Advanced Driver Assistance Systems |
| ADS | Automated Driving System |
| AOSP | Android Open Source Project |
| ASCF | Automatically Commanded Steering Function |
| ASOA | Automotive Service-Oriented Software Architecture |
| AV | Automated Vehicle |
| BMBF | German Federal Ministry of Education and Research |
| BSI | British Standards Institution |
| CAN | Controller Area Network |
| CDF | Cumulative Distribution Function |
| DARPA | Defense Advanced Research Projects Agency |
| DDT | Dynamic Driving Task |
| ECEF | Earth-Centered, Earth-Fixed |
| ECU | Electronic Control Unit |
| ENU | East-North-Up |
| ETRS89 | European Terrestrial Reference System 1989 |
| EU | European Union |
| FKZ | Förderkennzeichen |
| GCS | Geographic Coordinate System |
| GNSS | Global Navigation Satellite System |
| GRS80 | Geodetic Reference System 1980 |
| HD | High Definition |
| HiL | Hardware in the Loop |
| HMI | Human Machine Interface |
| IDEF | Integration Definition |
| IDL | Interface Definition Language |
| IMU | Inertial Measurement Unit |
| ISO | International Organization for Standardization |
| LED | Light Emitting Diode |
| LIN | Local Interconnect Network |
| LVDS | Low Voltage Differential Signaling |
| MEMS | Micro-Electro-Mechanical Systems |

| | |
|---|---|
| MPSoC | Multiprocessor System on a Chip |
| MRC | Minimal Risk Condition |
| MRM | Minimal Risk Maneuver |
| NDA | Non-Disclosure Agreement |
| ODD | Operational Design Domain |
| PAS | Publicly Available Specification |
| ROS | Robot Operating System |
| RTK | Real-Time Kinematic |
| SADT | Structured Analysis and Design Technique |
| SAE | Society of Automotive Engineers |
| SI | International System of Units |
| SiL | Software in the Loop |
| StVG | Straßenverkehrsgesetz |
| TAI | Temps Atomique International (International Atomic Time) |
| TBD | To Be Determined |
| TCP | Transmission Control Protocol |
| TUDa | Technical University of Darmstadt |
| UDP | User Datagram Protocol |
| UN-ECE | United Nations Economic Commission for Europe |
| VaMoRs | Versuchsfahrzeug für autonome Mobilität und Rechnersehen |
| VRU | Vulnerable Road User |
| WGS84 | World Geodetic System 1984 |

# List of Figures

# List of Tables

# Kurzzusammenfassung

Um ein Fahrzeug zum Ziel einer Fahrmission zu bewegen, müssen eine Vielzahl von Aufgaben erfüllt werden. Zu diesen Aufgaben gehören die taktische und strategische Planung der Fahrmission und die Bewegungsregelung des Fahrzeugs in Längs- und Querrichtung. Fahrerassistenzsysteme unterstützen einen menschlichen Fahrzeugführer/eine menschliche Fahrzeugführerin bei der Ausführung dieser Aufgaben. Treten in diesen Systemen Störungen auf, so wird der Fahrzeugführer/die Fahrzeugführerin über die aufgetretenen Systemeinschränkungen informiert und muss die Fahrzeugführung übernehmen. Bei automatisierten Fahrzeugen entfällt dieser Rückgriff auf einen menschlichen Fahrer/eine menschliche Fahrerin. Treten bei diesen Fahrzeugen Systemeinschränkungen auf, so muss eine technische Rückfallebene die Fahrzeugführung übernehmen. Das automatisierte Fahrsystem muss daher bei Ausfällen sicher (fail-safe) sein. Ausfallsicher bedeutet, dass das automatisierte Fahrsystem bei aufgetretenen Fehlern keine Funktion mehr zur Erfüllung einer Fahrmission hat, aber das Fahrzeug in einem sicheren Zustand halten und in einen Zustand minimalen Risikos überführen kann. Dazu wird ein situationsabhängiger risikominimaler Zustand ausgewählt. Er ist gekennzeichnet durch den globalen Zustand minimalen Risikos in Bezug auf die Länge des Manövers zu diesem Zustand und das Restrisiko durch den Zustand minimalen Risikos selbst. Für das Forschungsprojekt UNICAR*agil* wird das Konzept *Sicheres Anhalten* vorgeschlagen. Mit diesem Konzept sollen die oben genannten Anforderungen erfüllt werden. Im Stand der Technik fehlt eine Bewertung dieses Konzepts. Diese Bewertung wird in dieser Arbeit vorgenommen. Das Konzept stützt sich auf vorausgeplante implizite Nottrajektorien, die von einem Planungsmodul erzeugt werden. Ein Alleinstellungsmerkmal des Konzepts ist die Nutzung einer unabhängigen Umfelderfassung zur Absicherung des risikominimalen Manövers bis zum risikominimalen Zustand. Auf Basis der vorausgeplanten impliziten Nottrajektorie und den Daten der unabhängigen Umfelderfassung plant *Sicheres Anhalten* Trajektorien bis zum risikominimalen Zustand. Mit diesem Konzept kann somit auch bei vollständigem Ausfall der Umfelderfassung oder der strategischen und taktischen Planung eines automatisierten Fahrsystems der sichere Zustand aufrecht erhalten werden und das Fahrzeug in einen risikominimalen Zustand überführt werden. Zur Evaluation des Konzepts von *Sicherem Anhalten* wird eine Methodik vorgestellt. Hierzu werden zuerst die Fehlertoleranzregime eines automatisierten Fahrzeugs definiert. Als nächstes wird eine Referenzimplementierung für *Sicheres Anhalten* erstellt. Zu diesem Zweck werden zuerst Anforderungen für ein *Sicheres Anhalten* in einem generischen automatisierten Fahrsystem identifiziert. Ergänzt werden diese durch spezifische Anforderungen, die sich durch die Anwendung im automatisierten Fahrsystem von UNICAR*agil* ergeben. Abschließend werden für ein *Sicheres Anhalten* im automatisierten Fahrsystem von UNICAR*agil* Konzepte und eine synthetisierte Referenzlösung erstellt. Diese Lösung wird mit Testkriterien und Testfällen verifiziert. Eine abschließende Bewertung des Konzepts *Sicheres*

*Anhalten* zeigt eine hohe Effektivität bezüglich der Größe der Teilmenge von Fehlerkombinationen eines automatisierten Fahrsystems, für die *Sicheres Anhalten* eine fail-safe Eigenschaft ermöglicht. Die Anforderungen an *Sicheres Anhalten* sind verifiziert und die Referenzlösung erfüllt die spezifizierten Anforderungen. Das Konzept *Sicheres Anhalten* eignet sich somit zur Anwendung in einem automatisierten Fahrsystem zur Aufrechterhaltung des sicheren Zustands. Es wird eine Validierung des Konzepts im öffentlichen Straßenverkehr empfohlen.

# Abstract

In order to guide a vehicle to the destination of a driving mission, various tasks shall be performed. These tasks include tactical and strategic planning of the driving mission and longitudinal and lateral vehicle motion control. Driver assistance systems support a human vehicle driver in performing these tasks. If faults occur in these systems, the vehicle driver is informed of the system limitations and shall take over the control of the vehicle. This fallback to a human driver is not an option in automated vehicles. If system limitations occur in these vehicles, a automated fallback level shall take over vehicle control. The automated driving system shall therefore be fail-safe. Fail-safe means that when faults occur, the automated driving system no longer has any function to perform a driving mission, but shall maintain the vehicle in a safe state and transition the vehicle into a Minimal Risk Condition (MRC). For this purpose, a situation-dependent MRC is selected. It is characterized by the global MRC concerning the length of the maneuver and the residual risk of the MRC itself. For the research project UNICAR*agil*, the concept *Safe Halt* is proposed. This concept is intended to satisfy the requirements mentioned above. In the state of the art, an evaluation of this concept had not been included. This missing evaluation is performed in this thesis. The concept relies on pre-planned implicit emergency trajectories generated by a planning module. A unique concept feature is an independent environment perception system to ensure the Minimal Risk Maneuver (MRM) up to the MRC. Based on the pre-planned implicit emergency trajectory and the data of the independent environment perception system, *Safe Halt* plans trajectories up to the MRC. Thus, with this concept, even in the presence of failures to the environment perception system and to the strategic and tactical planning of an automated driving system, the safe state can be maintained, and the vehicle can be transitioned to a MRC. A methodology is presented to evaluate the concept of *Safe Halt*. For this purpose, the fault tolerance regimes of an automated vehicle are defined. Next, a reference implementation for *Safe Halt* is provided. For this, requirements for a *Safe Halt* in a generic automated driving system are identified first. These are supplemented by specific requirements from the application in the UNICAR*agil* automated driving system. Finally, concepts and a synthesized reference solution are created for a *Safe Halt* in the UNICAR*agil* Automated Driving System (ADS). The solution is verified with test criteria and test cases. A final evaluation of the *Safe Halt* concept shows a high effectiveness for the size of the subset of fault combinations of an automated driving system for which *Safe Halt* enables a fail-safe property. The requirements for *Safe Halt* are verified, and the specific requirements are met by the reference solution. The concept *Safe Halt* is thus suitable for an automated driving system to maintain a safe state. Validation of the concept in public road traffic is recommended.

# 1 Introduction

This chapter includes the motivation for this thesis in Sec. 1.1. The research questions are formulated subsequently in Sec. 1.2. The chapter concludes with the fundamental methodology for answering the research questions in Sec. 1.3.

## 1.1 Motivation

Automating the Dynamic Driving Task (DDT) has become one of the focal points of technological development in the global automotive industry. Many Advanced Driver Assistance Systems (ADAS) are already available to customers in production vehicles. These assistance systems increase safety and comfort for human drivers performing the DDT. One of those systems is Adaptive Cruise Control (ACC), a cruise control system that also adapts the ego vehicle speed concerning the time gap and relative velocity to a vehicle ahead. If failures occur in these kind of ADAS, the human driver is notified about them, and the assistance system ceases to provide its functionality. Thus, the human driver is the fallback for failures of these assistance systems. The vision for the continuous development of ADAS towards higher levels of automation is the handover of the execution and responsibility for the DDT to another automated system. In an Automated Vehicle (AV), it is not even necessary for a human driver to be able to engage in the DDT. The motivation for this vision is research that shows a potential benefit of the introduction of AVs. First and foremost is the increase of inclusion in mobility. AVs may provide personal transportation for people who cannot operate a vehicle or are otherwise mobility-impaired. The increase in safety is also cited as an argument for introducing AVs. In Germany, around 2,500 people die in road traffic accidents every year[1]. AVs could improve safety, because they cannot be distracted or fatigued. However, in the state of the art, there is no evidence of safety gains from AVs. If functional limitations occur in these AVs or the Automated Driving System (ADS) cannot handle a certain situation, a fallback system shall be activated to take over vehicle control. This automated fallback system has the task of keeping the vehicle in a safe state and transitioning it into a well-defined condition, the Minimal Risk Condition (MRC). This system thus enables the fail-safe property of the ADS. It maintains the safe state even in the event of faults or failures of the ADS. There are also regulatory requirements for an AV to be equipped with automated systems to keep the vehicle in a safe state when its system limits are reached or in the event of faults, and to be able to stop at a MRC. In Germany, this is demanded by the *Gesetz zum autonomen Fahren* (cf. Sec. 2.3.1), but also by the European Union (EU) regulation 2019/2144 (cf.

[1] Statistisches Bundesamt (Destatis): Traffic fatalities in Germany 2021 (2021).

Sec. 2.3.2). In the publicly funded research project UNICAR*agil*[2], the fallback system concept *Safe Halt* is introduced to enable the fail-safe property of an ADS. The project is funded by the German Federal Ministry of Education and Research (BMBF) and explores novel modular hardware and software architectures for AVs. The *Safe Halt* concept has been incorporated into the design of these architectures. An evaluation of the concept in terms of guaranteeing the fail-safe property of an AV is required.

## 1.2 Research Questions and Working Hypothesis

The primary motivation of this thesis is to evaluate whether the *Safe Halt* concept enables the fail-safe property of an ADS, in particular of the UNICAR*agil* ADS. To achieve this objective, the following central research questions are posed.

**RQ. 1:** How can the fallback concept of Safe Halt be evaluated in terms of enabling the fail-safe property of a generic ADS and the UNICARagil ADS?

A *Safe Halt* reference solution shall also be provided to evaluate the concept. To achieve this, the requirements for this reference solution shall be identified. From this follows the research question:

**RQ. 2:** What requirements shall a Safe Halt solution meet?

Once the *Safe Halt* requirements have been identified, a concept and a synthesized reference solution that meet the requirements shall be provided, leading to the research question:

**RQ. 3:** What is a concept and a functional architecture for a Safe Halt solution to satisfy the specified requirements?

For a final evaluation of the concept, the test criteria and test cases for the verification of the *Safe Halt* solution shall be defined. As a result, the final research question is:

**RQ. 4:** How can a Safe Halt solution be verified?

To address these research questions, the following working hypothesis is formulated.

An Integration of Safe Halt into an ADS and in Particular into the UNICAR*agil* ADS Enables its Fail-Safe Property without any Limitations.

[2] RWTH Aachen: UNICARagil Startseite (2022).

In the remaining part of this thesis, the mentioned working hypothesis is attempted to be corroborated. Identified vulnerabilities and limitations of *Safe Halt* are used to update the working hypothesis.

## 1.3 Fundamental Methodology and Structure

The fundamental methodology applied to the research documented by this thesis is shown in Fig. 1-1.

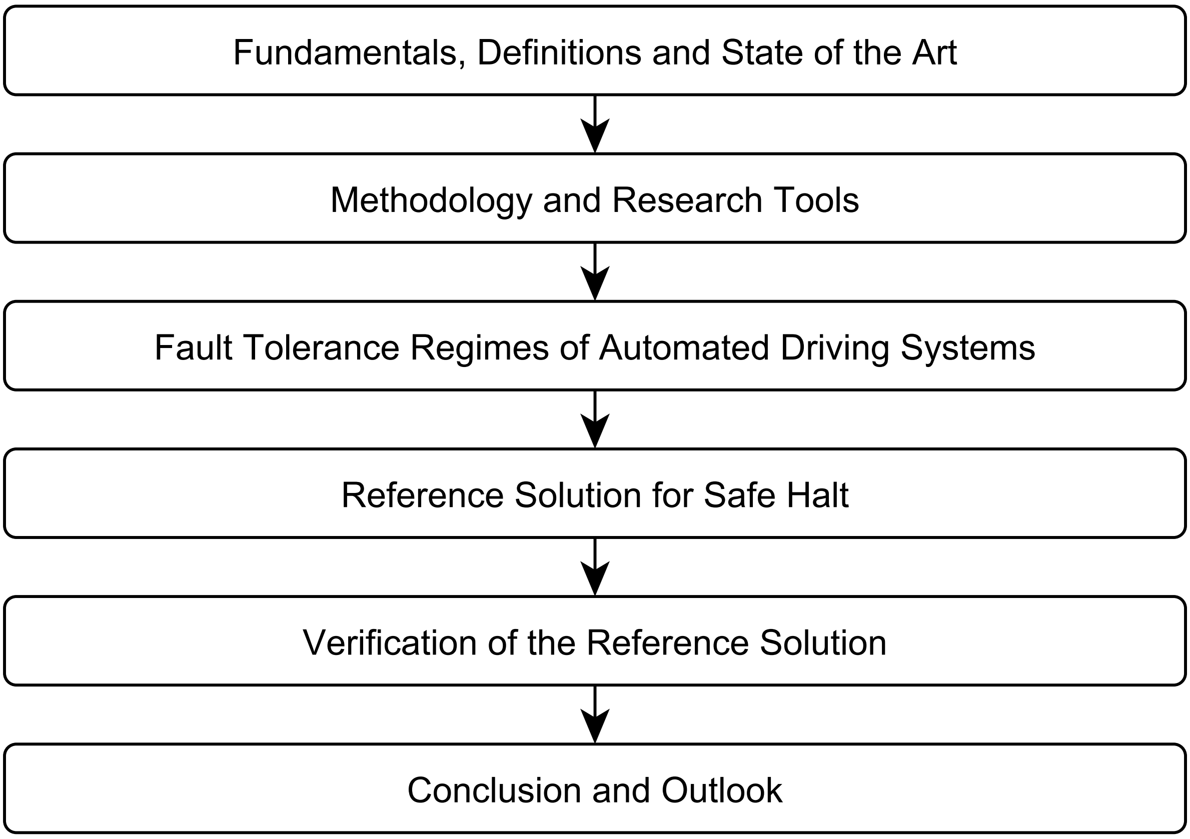

Fundamentals, Definitions and State of the Art
Methodology and Research Tools
Fault Tolerance Regimes of Automated Driving Systems
Reference Solution for Safe Halt
Verification of the Reference Solution
Conclusion and Outlook

Figure 1-1: Fundamental Methodology for this Thesis

The structure of this thesis follows the methodology outlined in Fig. 1-1. The fundamentals, definitions, and state of the art are introduced in Chap. 2. Based on these fundamentals, a methodology for evaluating the *Safe Halt* concept is presented in Chap. 3, which is used to corroborate the working hypothesis. Following the overall methodology, an analysis of the fail-safe property of an ADS is realized in Chap. 4. Subsequently, a reference solution for *Safe Halt* is introduced. For this purpose, the requirements, concepts, and a synthesis of the overall function are provided in Chap. 5, Chap. 6, Chap. 7, Chap. 8, Chap. 9, Chap. 10 and Chap. 11. The findings in these chapters are structured in such a way that a generic ADS is always considered at the beginning of each chapter, followed by an application to the UNICAR*agil* ADS. Next, the verification methods for the *Safe Halt* reference solution are described in Chap. 12. These are applied to the presented reference solution, and the results are evaluated. Finally, the research questions are answered, the working hypothesis is revised, and the *Safe Halt* concept is evaluated in Chap. 13.

# 2 Fundamentals and Definitions

This chapter presents the fundamentals, which provide a basic foundation for this thesis and enable a common understanding of the research task. In most cases, well-known definitions and terms are adopted from existing standards and publications. Some of the definitions are defined diversely in various publications. Relevant to this thesis are only the definitions and terms introduced here. The content of this chapter is ordered according to logical contexts. In the beginning in Sec. 2.1, the fundamental definitions for this thesis are established. In Sec. 2.2, the state of the art of automated fallback systems is reviewed. Regulatory requirements have been introduced for the deployment of Automated Vehicle (AV), which are presented in Sec. 2.3. In Sec. 2.4, the coordinate systems considered in this thesis are specified. The research project UNICAR*agil* is outlined in Sec. 2.5. The chapter ends with introducing the IDEFØ Function Modeling Methodology in Sec. 2.6, which is used throughout this thesis to represent functional architectures.

## 2.1 Definitions

### 2.1.1 Dynamic Driving Task (DDT)

The Dynamic Driving Task (DDT) includes all tasks that a driver shall perform in order to operate a vehicle. The definition used here is directly adopted from Society of Automotive Engineers (SAE) J3016[3a]:

*All of the real-time operational and tactical functions required to operate a vehicle in on-road traffic, excluding the strategic functions such as trip scheduling and selection of destinations and waypoints, and including, without limitation, the following subtasks:*

1. *Lateral vehicle motion control via steering (operational).*
2. *Longitudinal vehicle motion control via acceleration and deceleration (operational).*
3. *Monitoring the driving environment via object and event detection, recognition, classification, and response preparation (operational and tactical).*
4. *Object and event response execution (operational and tactical).*
5. *Maneuver planning (tactical).*
6. *Enhancing conspicuity via lighting, sounding the horn, signaling, gesturing, etc. (tactical).*

[3] SAE International: SAE J3016 (2021). a: p. 9; b: p. 10; c: p.15; d: p. 17; e: p. 6; f: p. 4.

### 2.1.2 Dynamic Driving Task (DDT) Fallback

The Dynamic Driving Task Fallback[3b] is:

*The response by the user to either perform the DDT or achieve a minimal risk condition (1) after occurrence of a DDT performance-relevant system failure(s), or (2) upon operational design domain (ODD) exit, or the response by an ADS to achieve minimal risk condition, given the same circumstances.*

### 2.1.3 Minimal Risk Maneuver (MRM)

The vehicle's transition into the minimum risk condition is called a Minimal Risk Maneuver (MRM). The United Nations Economic Commission for Europe (UN-ECE) working party Automatically Commanded Steering Function (ASCF) considers the Minimal-Risk Manoeuvre[4] to be:

*A procedure aimed at reducing risks in traffic, which is automatically performed by the system when the driver does not respond to a transition demand (e.g. by reducing vehicle speed)*

### 2.1.4 Minimal Risk Condition (MRC)

The Minimal Risk Condition (MRC) [3c] is:

*A stable, stopped condition to which a user or an ADS may bring a vehicle after performing the DDT fallback in order to reduce the risk of a crash when a given trip cannot or should not be continued*

An analysis of this definition can be found in Stolte et al.[5] It is important to acknowledge that the minimal risk condition is not inherently a safe state, i.e., the risk is not inherently acceptable. Koopmann has also created a "User Guide"[6] to SAE J3016. In this guide, he dispels various myths about the standard.

A comprehensive definition of MRC is expected to be given in SAE J3164[7], which has not yet been published.

[4] Experts from OICA/CLEPA: Definition of Minimal Risk Manoeuvre (2015).

[5] Stolte, T. et al.: Taxonomy to Unify Fault Tolerance Regimes for Automotive Systems (2022), p. 7.

[6] Koopman, P.: SAE J3016 User Guide (2023).

[7] SAE International: SAE J3164 (2018).

### 2.1.5 Operational Design Domain (ODD)

A comprehensive definition of Operational Design Domain (ODD) is expected to be given in SAE J3259[8]. At this time, it is unpublished. Already published are definitions from British Standards Institution (BSI) as Publicly Available Specification (PAS) 1883[9] and from Koopmann et al.[10].

For this thesis the definition from SAE J3016[3d] is adopted:

*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.*

### 2.1.6 Automated Driving System (ADS)

According to SAE J3016[3e]:

*The hardware and software that are collectively capable of performing the entire DDT on a sustained basis, regardless of whether it is limited to a specific operational design domain (ODD); this term is used specifically to describe a Level 3, 4, or 5 driving automation system.*

Following this definition, an AV is equipped with an Automated Driving System (ADS). This thesis focuses on levels 4 and 5 ADS. A potential extension to a level 3 ADS is not considered.

### 2.1.7 Levels of Driving Automation

One core definition of SAE J3016[3f] specifies the levels of driving automation. Defined are six levels of automation, ranging from 0 (no driving automation) to 5 (full driving automation):

- *Level 0: No Driving Automation*
- *Level 1: Driver Assistance*
- *Level 2: Partial Driving Automation*
- *Level 3: Conditional Driving Automation*
- *Level 4: High Driving Automation*
- *Level 5: Full Driving Automation*

A critical view of this definition, especially of level 5, is given by Koopmann in his “User Guide”[6].

[8] SAE International: SAE J3259 (2022).

[9] The British Standards Institution: PAS 1883:2020 (2020).

[10] Koopman, P.; Fratrik, F.: How Many ODD, Objects, and Events? (2019).

### 2.1.8 Fault Tolerance Regimes

The journal article by Stolte et al.[11a] (the author of this thesis is also one of the co-authors of that article) shows that fault tolerance regimes for automotive systems are partly defined ambiguously or even entirely different throughout various scientific publications. Therefore, the article's authors propose a taxonomy to unify automotive system fault tolerance regimes alongside related terms' definitions. Those definitions are adopted for this thesis. For compatibility of their established taxonomy, the authors adopted and merged definitions from various publications. The primary references are cited as follows.

#### Safety

Safety[12a] is defined as:

*Absence of unreasonable risk.*

The concept discussed here is functional safety, which can be understood as the absence of unreasonable risks resulting from the faulty behavior of E/E (electrical/electronic) systems. Conversely, in International Organization for Standardization (ISO)/PAS 21448[13] safety is defined as the absence of unreasonable risks arising from functional deficiencies or their implementation. Currently, there is no established framework for determining the metrics that quantify the level of risk or the specific value that represents an unreasonable risk. In addition, it remains unclear which individuals or entities are affected by these unreasonable risks, such as vehicle occupants or the surrounding environment. There is also no consensus on the scope of hazards that should be included in safety assessments, including physical, psychological, and environmental risks. In addition, the understanding of safety varies among the various stakeholders involved in the field. These stakeholders include a range of parties, including companies, the media, and legislators. As a result, the term "safety" and its precise definition are subject to ongoing refinement and lack detailed consensus.

#### Risk

The definition for Risk[12a] is:

*Combination of the probability of the occurrence of harm and the severity of that harm.*

Harm is defined as *physical injury or damage to the health of persons*. Furthermore, it is assumed that the risk increases monotonically as the probability of occurrence of harm or the severity of that harm increases[11b].

[11] Stolte, T. et al.: Taxonomy to Unify Fault Tolerance Regimes for Automotive Systems (2022). a: -; b: p.6; c: p.7; d: p.8.

[12] International Organization for Standardization (ISO): ISO 26262:2018 (2018). a: Part 1 p. 21; b: Part 1 p. 11; c: Part 1 p. 10; d: Part 1 p. 12.

[13] International Organization for Standardization (ISO): ISO/PAS 21448:2019 (2019), p. 7.

### Fault

A fault[12b] is an:

*Abnormal condition that can cause an element or an item to fail.*

Faults are assumed to be discrete and distinguishable[11b].

### Failure

A failure[12c] is the:

*Termination of an intended behaviour of an element or an item due to a fault manifestation.*

### Fault Tolerance

Fault tolerance[12d] is the:

*Ability to deliver a specified functionality in the presence of one or more specified faults.*

### Performance

Performance has not yet been defined for automotive systems[11b]. The authors Stolte et al., therefore, adopt the definition from Walden et al.[14]:

*A quantitative measure characterizing a physical or functional attribute relating to the execution of a process, function, activity, or task; performance attributes include quantity (how many or how much), quality (how well), timeliness (how responsive, how frequent), and readiness (when, under which circumstances)*

### Fault Tolerance Regime

Fault Tolerance Regime[11b] is defined as:

*System property that classifies the systems behavior in the presence of a specific fault combination.*

A *fault combination* is a combination of distinct faults that occur at the same time. Stolte et al. assume that when $\mathscr{F}$ denotes the set of all distinct faults that can occur in a system, $\mathscr{P}(\mathscr{F}) = \{f \mid f \subseteq \mathscr{F}\}$ is the set of possible fault combinations $f$[11b]. Furthermore, it is assumed that a fault tolerance regime is a property valid only for the fault combinations under consideration. It is not a system-wide property[11b].

### Safe State

The safe state[11c] is a

*State in which a system does not pose an unreasonable risk.*

[14] Walden, D. D. et al.: Systems engineering handbook (2015), p. 264.

### Functionality

A functionality[11c] is a

*Behavior of a system expressed in its interaction with its operating environment.*

Using the definitions for safe state and functionality, it is possible to determine for any fault combination $f$ whether a system is in a safe state and fulfills its function, is in a safe state but does not fulfill its function, or is not safe. Therefore the authors introduce the system's operability $o(f)$ as

$$o(f) = \begin{cases} 1, & \text{system is in a safe state while providing} \\ & \text{its specified functionality;} \\ 0, & \text{system is in a safe state while not pro-} \\ & \text{viding its specified functionality;} \\ -1, & \text{otherwise.} \end{cases}$$

### Available performance

The available performance $p_{\mathrm{a}}(f)$[11d] of a system while providing its functionality is defined as:

*Performance that is available for a system to provide its specified functionality.*

### Nominal performance

The nominal performance $p_{\mathrm{nom}}$[11d] is defined as follows:

*Performance with which a fault-free system is expected to be able to provide its specified functionality.*

### Operational

Operational[11d] means

*An operational system has no fault and, thus, can provide its specified functionality with at least nominal performance while maintaining a safe state.*

The authors of Stolte et al. consider fault tolerance regimes. For this reason, they assume that a fault-free system is in a safe state.

### Fail-unsafe

Fail-unsafe[11d] means

*A system is fail-unsafe in the presence of a fault combination if it is not able to maintain a safe state.*

### Fail-safe

Fail-safe[11d] means

*A system is fail-safe in the presence of a fault combination if it ceases its specified functionality and transitions to a well-defined condition to maintain a safe state.*

### Fail-degraded

Fail-degraded[11d] means

*A system is fail-degraded in the presence of a fault combination if it can provide its specified functionality with below nominal performance while maintaining a safe state.*

### Fail-operational

Fail-operational[11d] means

*A system is fail-operational in the presence of a fault combination if it can provide its specified functionality with at least nominal performance while maintaining a safe state.*

The relation between these fault tolerance definitions is summarized in the schema depicted on Fig. 2-1.

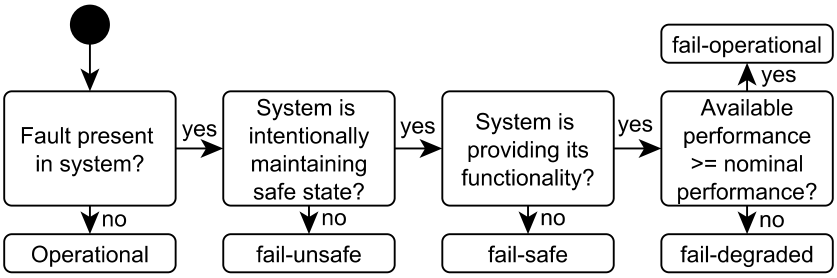

Fault present in system?
yes
System is intentionally maintaining safe state?
yes
System is providing its functionality?
yes
Available performance >= nominal performance?
yes
fail-operational
no
Operational
no
fail-unsafe
no
fail-safe
no
fail-degraded

Figure 2-1: Schema to Distinguish Fault Tolerance Regimes[15]

## 2.1.9 Reference Trajectory

Synonyms are often used in the literature to denote equivalent matters.

The *reference trajectory* is the output of the trajectory planner. The trajectory describes the intended behavior to be realized by a trajectory controller. In literature, there are different terms for the reference trajectory. Synonyms are *target trajectory*[16a] and *trajectory plan*[17a].

[15] Own illustration according to Stolte, T. et al.: Taxonomy to Unify Fault Tolerance Regimes for Automotive Systems (2022), p. 8, Fig. 2

[16] Homolla, T.; Winner, H.: Encapsulated trajectory tracking control (2022). a: p. 4; b: p. 1.

[17] SAE International: SAE J3131 (2022). a: p. 12; b: p. 11.

### 2.1.10 Trajectory Controller

There are also different definitions for a trajectory controller.

A *trajectory controller* generates the actuating commands for the motion actuators of the AV based on the reference trajectory and the current vehicle pose. The actuators include braking, powertrain, and steering. In addition to the term *trajectory controller*, the terms *trajectory tracking controller*[16b] or *Vehicle Motion Control Request Generator*[17a] are used. The output of the *trajectory controller* is the *vehicle motion control request*[17a] or *actuator setpoints*.

### 2.1.11 Trajectory Waypoints

There are also different definitions for a trajectory waypoint. The single state of a trajectory is described as *waypoint*. However, *waypoint* only includes the state for the position. Therefore, the terms *trajectory setpoints* or *trajectory elements* are also used instead.

### 2.1.12 Intended Behavior

The observable vehicle behavior is planned by the ADS based on the observed vehicle environment and the driving mission. The planned vehicle behavior can differ from the observable vehicle behavior since unforeseen disturbances or faults can lead to deviations from the observable vehicle behavior. A planned vehicle behavior can generally be expressed in two different notations. On the one hand, as an explicit trajectory, and on the other hand, as an implicit trajectory.

#### Explicit Trajectory

An explicit trajectory describes the vehicle behavior in temporal-spatial space. In SAE J3131[17b], an explicit trajectory is described as *A timed sequence of locations*. A generalized definition of an explicit trajectory is *Time sequence of state variables*.

The International Electrotechnical Commission, on the other hand, defines a trajectory as *representation of the solution $\vec{x}(t)$ of the state equation as connecting line of the ends of the vector $\vec{x}(t)$ in state space with time as parameter.*[18]

Most relevant to this thesis is the temporal relation between the state vector and time.

#### Implicit Trajectory

An implicit trajectory is characterized by the fact that it contains a spatial reference pose course on the one hand and a location-dependent velocity profile on the other. A time-dependent explicit trajectory can be generated using the pose course and the velocity profile. In this dissertation, the description of an implicit trajectory for the so-called implicit emergency trajectory is used as

[18] International Electrotechnical Commission: Definition for "Trajectory" (2022).

behavioral limits for the function *Safe Halt*. The velocity contained in the implicit trajectory is thereby interpreted as a location-dependent maximum velocity profile. A velocity reduction is allowed, but must not exceed the maximum velocity.

#### Pose Course

A pose course describes the spatial reference behavior of a vehicle. The waypoints of the course include both a position and a yaw angle. Given this definition, a pose course can describe any vehicle behavior in a generalized way. This is particularly relevant if using a vehicle that can use the yaw angle as an independent degree of freedom. For example, identical all-wheel steered vehicles can set the yaw angle independently of the current and previous vehicle positions.

In SAE J3131[17b], a path is defined as *A sequence of locations without respect to time*. However, this definition lacks the inclusion of the yaw angle.

## 2.2 State of the Art

There are various approaches in the state of the art to address the safety of AVs, especially in emergency situations. In this section, the state of art of fallback systems for AVs is reviewed. This section is divided into the topics of emergency stop systems in Sec. 2.2.1, the minimal risk condition in Sec. 2.2.2, and emergency stop systems for electrical/electronic failures in Sec. 2.2.3. The section ends with a conclusion about state of the art in Sec. 2.2.4.

### 2.2.1 Emergency Stop Systems for Automated Vehicles (AV)

Research on automated vehicles began several decades ago. As early as 1979, Tsungawa et al.[19] described an automobile with artificial intelligence. Cameras are used to capture the vehicle environment. The image data is converted into vehicle motion by a problem-solving unit. The vehicle reaches a velocity of 30 km/h in an automated mode. The fallback level of this system is not described in the publication. It is not clear from the publication what happens if vehicle components fail and how the safe state of the vehicle can be maintained.

In Germany, as part of the research project "PROgraMme for a European Traffic of Highest Efficiency and Unprecedented Safety" (PROMETHEUS)[20] [21] [22] the research vehicle Versuchsfahrzeug für autonome Mobilität und Rechnersehen (VaMoRs)[23] was built. The vehicle drove

[19] Tsugawa, S. et al.: An Automobile with Artificial Intelligence (1979).

[20] Zimmer, H.: PROMETHEUS Forschungsprogramm zur Gestaltung des künftigen StraSSenverkehrs (1990).

[21] Mercedes-Benz: 1986: Startschuss zum PROMETHEUS-Forschungsprogramm (2007).

[22] Mercedes-Benz: Das Projekt PROMETHEUS ab 1986: Vorreiter des autonomen Fahrens (2016).

[23] Dickmanns, E.; Zapp, A.: Autonomous High Speed Road Vehicle Guidance (1987).

at a maximum velocity of 96 km/h on a highway section. This vehicle was also equipped with cameras. A visual feedback control system provided automatic vehicle guidance in the structured environment. The publications also lack a description of the fault behavior of the vehicle. It remains unclear how the minimal risk condition can be approached with reasonable risk when elementary components of the vehicle are degraded.

Also in Germany, in the state of Lower Saxony, the research project "Automated Driving" was initiated. As part of this project, a safety concept for automated vehicles was described by Binfet-Kull et al.[24], which can be used to transition the vehicle to a minimal-risk condition in the event of faults of elementary automation components. For this purpose, fault codes for different degrees of degradation severity were defined and risk-minimizing maneuvers were derived from them. Here, the error code "Emergency parking" is described, which leads to the maneuver "Immediate stopping of the vehicle and parking the vehicle by the road side without any direct risk for other road users". The publication does not describe how a fallback level might be designed in order to be able to perform all the risk-minimizing maneuvers described.

In his publication, Ameling[25] describes the functional architecture and implementation of the "Emergency Braking" module. The electric co-pilot of the research project "Automated Driving" is used for this purpose. Potential collision objects are detected by the vehicle's environment perception system. Various parameters are then used to calculate whether emergency braking is necessary. The electronic co-pilot then executes the maneuver. With this implementation, the vehicle can be transitioned to a local minimal risk condition. However, this emergency braking assistant does not involve any degradation of the automation components. Thus, the described system is a pure emergency braking function for inattentive drivers.

A comprehensive overview of the research history of automated vehicles up to the year 2000 can be found in the dissertation by Maurer[26]. The focus of this state of the art is especially on the functional solution of the dynamic driving task using computer vision. No overall concept and no implementation of a fallback solution for the automated driving system is described.

In the two Defense Advanced Research Projects Agency (DARPA) Grand Challenges in 2004 and 2005 and the DARPA Urban Challenge in 2007, teams with automated vehicles competed against each other. Thrun et al.[27] succeeded for the first time in 2005 in completing the specified course the fastest with his team and won the Grand Challenge 2005. The demonstration of the automated vehicles at these Challenges focused on their function in completing the driving task. For safety, the vehicles were equipped with an e-stop system that allowed them to be stopped remotely. With this system, the vehicles could be transitioned to a minimal-risk condition. For this purpose, the

[24] Binfet-Kull, M. et al.: System safety for an autonomous driving vehicle (1998).

[25] Ameling, C.: The electronic copilot for an autonomous vehicle (1999).

[26] Maurer, M.: Flexible Automatisierung von StraSSenfahrzeugen mit Rechnersehen (2000), p.12-25.

[27] Thrun, S. et al.: Stanley: The robot that won the DARPA Grand Challenge (2006).

vehicle was decelerated from the initial condition to a standstill. The minimal-risk condition obtained thus depended on the initial condition.

In 2014, Kwon et al.[28] described an emergency stop system that takes over vehicle control if a human driver can no longer perform the dynamic driving task. The focus here is on the emergency stop system as an assistance function. The system has the task of keeping the vehicle in control in an emergency situation and stopping it without a collision. The vehicle should be guided out of the flow of traffic by changing lanes if possible. The system uses an environment perception system based on radar, camera, and ultrasonic. Fuzzy logic is implemented to plan and finally execute the vehicle's behavior.

Magdici et al.[29] evaluate the most likely future trajectories to predict other road users. For this purpose, the most likely maneuvers of all road users in the vicinity of the AV are predicted first. Based on these, an optimal trajectory that does not result in a collision based on these predictions is calculated for the AV. Subsequently, an emergency trajectory is calculated. For this purpose, all trajectories that the other road users can potentially reach are included in the computation. The emergency trajectory is then planned so that no collision can occur with all predicted trajectories of the other road users. This presents an optimization problem, which must be solved. With this approach, a short-term emergency stop maneuver is available at any time. Due to the lack of environment data updates, extended stop maneuvers to potentially lower-risk conditions are only possible with increased risk during the execution of the maneuver.

Pek et al.[30] take up the idea of planning a fail-safe trajectory and use this approach as motivation to develop an implementation for efficient computation of these emergency trajectories based on convex optimization techniques. As a result of their work, real-time computation of emergency trajectories is possible. In addition, their planner performs the safety verification of the generated trajectories. An additional achievement is that the trajectories are planned in a jerk-minimized way, leading to an increase in vehicle passenger comfort, even in these particular emergency situations. With this approach, the vehicle can be stopped in a specific way. However, the minimal risk condition obtained is not inherently acceptable in terms of safety.

Svensson et al.[31] also describe the generation of emergency trajectories as an optimization problem. In this case, the problem is described as an optimal control problem. To solve this kind of problem, they apply an approach with dynamic programming. However, since this approach does not achieve real-time capability, an additional real-time capable algorithm is presented as the second result of this paper. The cost function of the optimization problem to be solved is weighted such that a final stop located outside the flow of traffic is preferred. Three emergency reactions are considered, stopping in one's lane, safe emergency stop, and emergency stop. The MRM

[28] Kwon, S. et al.: Autonomous emergency stop system (2014).

[29] Magdici, S.; Althoff, M.: Fail-safe motion planning (2016).

[30] Pek, C.; Althoff, M.: Efficient Fail-safe Trajectory Planning (2018).

[31] Svensson, L. et al.: Safe Stop Trajectory Planning (2018).

begins with deceleration in one's lane. It is followed by the emergency stop. The authors assume that the fallback level can access the environment sensor data and that this cannot be influenced by the automation system for the dynamic driving task. However, independent environment perception is not used.

The publication by Krook et al.[32] describes a supervisor for monitoring the vehicle capabilities and the resulting trajectory source selection. The solution allows the trajectory source to be switched to a dedicated emergency trajectory planner if necessary.

Brüdigam et al.[33] adopted a similar approach. In their work, trajectory planning is also considered as an optimal control problem. Specifically, they use a model predictive control approach for generating emergency trajectories. To compute the emergency trajectories, they again use the approach that always assumes the worst-case behavior of other road users. If no solution for the emergency trajectory is found in the immediate moment, a backup, the emergency trajectory from the last time step, is used. This approach also only allows the vehicle to stop immediately based on the latest available environment sensor data.

In Stolte et al.[34], *Safe Halt* is described as a fallback operating mode for the dynamic driving task of an automated construction vehicle. Due to the low speeds of the used prototype vehicle, this operation mode decelerates the vehicle directly to a standstill.

In Xue et al.[35], a fallback system is described, which can be used for vehicle environment sensor failures. For this purpose, the most recently identified road users are created as virtual instances. These virtual instances subsequently perform a worst-case behavior to the AV. Based on this assumption, a model predictive control algorithm is then used to solve the problem that leads to a collision-free standstill of the AV.

A similar approach is also conducted by Emzivat et al.[36]. As an additional feature, the rear space behind the AV is also included in the emergency response. It is assessed up to which distance to the rear the AV can still be perceived by the following traffic in the current situation. This information is introduced as an additional input of the computation responsible for calculating the execution speed of the fallback solution.

A framework for ensuring an executable MRM is presented in the publication by vom Dorff et al.[37]. In their case, two separate trajectory planners with different requirements regarding their fault tolerance are used. According to the paper, the secondary planner shall be fail-operational,

[32] Krook, J. et al.: Safe Stop Supervisor for an Automated Vehicle (2019).

[33] Brüdigam, T. et al.: Stochastic MPC with a Safety Guarantee (2022).

[34] Stolte, T. et al.: Unmanned Protective Vehicle for Highway Hard Shoulder (2015).

[35] Xue, W. et al.: An adaptive model predictive approach in fallback procedure (2019).

[36] Emzivat, Y. et al.: Dynamic driving task fallback for an ads (2017).

[37] Dorff, S. v. et al.: A Fail-safe Architecture for Automated Driving (2020).

while the primary planner shall only be fail-silent. Thus, the primary planner can also be based on the application of neural networks.

In addition to scientific publications, patent applications have also been published on emergency stop systems for AVs. In Kazemi et al.[38], a system for safely stopping an AV is described. If possible, this system stops the vehicle outside of moving traffic. Also, in Le Cornec et al.[39], a system for emergency stopping of an AV is documented. In this publication, the setpoints for an emergency stop maneuver are calculated and executed as required by the actuator system. Finally, Dochow et al.[40] describe a system that aborts the driving mission of an AV and instead approaches a safe stopping location when a human driver in an assisted vehicle does not respond to a takeover request from the vehicle.

### 2.2.2 Minimal-Risk Condition

Hörwick[41] describes the MRC with regard to driver assistance systems. No lane changes are allowed due to the complex nature of the situation, so only stopping in one's own lane remains as the risk-minimal condition. It is essential that the deceleration is not too strong to avoid surprising the following traffic.

In the book *Autonomous Driving*[42], an overview of MRC for production vehicles and research and development vehicles is presented. For assisted systems, the MRC is always assumed to be the human driver. This driver is supposed to control the vehicle in emergency situations. An overview of the state of the art for emergency stop systems and procedures is presented for automated driving.

In their publication, Reschka et al.[43] describe the conditions of the safe state for AVs. For this purpose, the safe states for the automation levels of SAE are discussed. For an AV, it is required that the expected relative speed of the ODD has an influence on the target location, where an emergency maneuver should end. Therefore, for high relative speeds, a MRC is outside of the flow of traffic. Furthermore, the final location of an emergency maneuver must not result in blockage of other road users, especially emergency vehicles. Furthermore, a location is only safe if a stopped vehicle can be seen by other road users at a sufficient distance.

Leonhardt[44] elaborates the definitions of minimal risk conditions from the regulatory point of view. In addition, he points out that not only the risk of the MRC should be considered, but also

[38] Kazemi, M. et al.: Autonomous Vehicle Safe Stop (2019).

[39] Le Cornec, O.: Electronic device for determining an emergency stopping trajectory (2020).

[40] Dochow, G. et al.: Automated driving with a safe stop point (2020).

[41] Hörwick, M. A.: Sicherheitskonzept für hochautomatisierte Fahrerassistenzsysteme (2011), p. 100.

[42] Maurer, M. et al.: Autonomous Driving (2016), p. 475-479.

[43] Reschka, A.; Maurer, M.: Conditions for a safe state of automated road vehicles (2015).

[44] Leonhardt, T.: Minimal Risk Maneuver (2018).

the MRM to this condition. Therefore, he suggests that planning minimal risk maneuvers should depend on the environmental conditions and the remaining vehicle capabilities that are available when the emergency stop is initialized. As potential destinations for an emergency stop maneuver, he suggests stopping in the vehicle's lane or, if possible, outside the flow of traffic.

Patent applications have also been published about the topic of MRC. Laur et al.[45] describes a system that identifies minimal risk conditions as safety zones. For this purpose, a digital map of the vehicle's surroundings is used. An integrated controller guides the vehicle to such a safety zone in an emergency situation. In Chi et al.[46], a system is described that can identify minimal risk conditions for emergency stopping maneuvers. An AV is guided to these locations if it has been evaluated that the the current driving mission cannot be safely continued. A particular feature of this patent application is that a secondary stopping location for another vehicle is also identified in addition to a primary stopping location for the affected vehicle. The purpose of the secondary location is to serve as a safe pick-up spot to take on passengers of the primary vehicle so that they can continue their driving mission with minimal delay.

### 2.2.3 Emergency Stop Systems for Electrical/Electronical Failures

The systems for emergency stopping maneuvers described so far are almost exclusively based on the assumption that the vehicle dynamics control is able to provide a potential MRM and the vehicle actuators can execute it. In the publication by Beyerer et al.[47], actuator failures are included in planning a MRM. With this approach, even failures of the electrics/electronics of the AV can be safely handled. For this purpose, a situation-dependent brake pressure is calculated and held in reserve. This brake pressure is deployed if the vehicle's electrical system fails. A related approach is also pursued by Duerr et al.[48].

### 2.2.4 Conclusion on the State of the Art

Several systems in state of the art can be applied to execute a MRM in an emergency situation of AVs. Different planning approaches are used to evaluate a potential MRM. The most common planning algorithms solve different types of optimization problems. However, approaches based on neural networks are also proposed in some cases. All of the presented systems are different from the concept *Safe Halt* as a fail-safe concept for AVs. The main differentiator of *Safe Halt* from all other systems is the consistent pursuit of a modularized and service-based paradigm in the design of the functional architecture. Another differentiator is the separate, independent environment perception system, which is exclusively responsible for executing the MRM. All

[45] Laur, M. H. et al.: AV SAFE -STOP-ZONE MAPPING SYSTEM (2018).

[46] Chi, E.; Andrade, R. J.: Vehicle System For Determining a Pullover Spot (2020).

[47] Beyerer, J. et al.: Fail-Safe Emergency Stopping for AV (2019).

[48] Duerr, F. et al.: Global optimization of a Fail-Safe Emergency Stop Maneuver (2020).

other systems described in this thesis use the primary environment perception system or act on worst-case predictions. However, the state of the art may not be exhaustive. Currently, common definitions and metrics for evaluating minimal risk conditions are still lacking. It is not yet possible for a developer to derive detailed requirements for a minimal-risk location from the state of the art. Moreover, the focused risk-weighted selection of the final location of a MRM is weak in the state of the art. The same applies to the risk weighting between the minimal risk condition and the minimal risk maneuver to that location. Some approaches can safely handle Electrical/Electronical failures and safely stop the vehicle even in these cases. The *Safe Halt* concept does not have these capabilities. It is a trajectory source and depends on vehicle dynamics control and actuation. Thus, an evaluation of the concept of *Safe Halt* is not part of the state of the art. As such, one of the motivations for this thesis is to provide a proper evaluation of the proposed *Safe Halt* concept.

## 2.3 Legal Regulations

Various legal requirements must be met to grant AVs access to participate in public traffic. These include both national and international regulations. The following section introduces German legislation on approval requirements for AVs. Subsequently, the European regulation is introduced.

### 2.3.1 Gesetz zum autonomen Fahren

On July 28, 2021, the German Straßenverkehrsgesetz (StVG) was amended. Titled *Gesetz zum autonomen Fahren*, the updated law revision describes the requirements for a motor vehicle with autonomous driving function to be allowed to participate in public traffic. Since the legal text is only valid in German, an English translation is omitted.

The new §1d reads as follows:

> Kraftfahrzeuge mit autonomer Fahrfunktion müssen über eine technische Ausrüstung verfügen, die in der Lage ist [..]
> 7. ihre Systemgrenzen zu erkennen und beim Erreichen der Systemgrenze, *beim Auftreten einer technischen Störung, die die Ausübung der autonomen Fahrfunktion beeinträchtigt*, oder beim Erreichen der Grenzen des festgelegten Betriebsbereichs *das Kraftfahrzeug selbstständig in einen risikominimalen Zustand zu versetzen* [..]

According to §1d, the MRM is defined as:

> *Risikominimaler Zustand* im Sinne dieses Gesetzes ist ein Zustand, in dem sich das Kraftfahrzeug mit autonomer Fahrfunktion auf eigene Veranlassung oder auf Veranlassung der Technischen Aufsicht *an einer möglichst sicheren Stelle in den Stillstand versetzt und die Warnblinkanlage aktiviert, um unter angemessener Beachtung der Verkehrssituation die größtmögliche Sicherheit für die Fahrzeuginsassen, andere Verkehrsteilnehmende und Dritte zu gewährleisten*

The law thus requires that an automated vehicle be transitioned to a minimal risk condition when system limits are exceeded or in the event of technical faults. According to the law, minimal risk condition means that it must be "as safe as possible". However, it does not mean that the residual risk of the minimal risk condition is acceptable and therefore safe. In this dissertation, safety is defined as the "absence of unreasonable risk". This definition is thus stricter than the German legal text.

### 2.3.2 Regulation (EU) 2019/2144

In July 2022, the draft for the application of the European Union (EU) Regulation 2019/2144[49] of the European Parliament and the Council as regards uniform procedures and technical specifications for the type-approval of the automated driving system (ADS) of fully AVs was presented. The proposal describes the requirements for an EU type-approval of AVs with regard to their ADS.

Section 3 *DDT at ODD boundaries* of Annex 2 *Performance requirements* requires:

> - 3.1.5 When the ADS reaches the ODD boundaries, it shall perform a Minimum Risk Manoeuvre to reach a MRC and shall warn the operator/remote operator accordingly (if available).

Section 4 *DDT under failure scenarios* requires:

> - 4.1.2.2 The ADS shall execute a safe fall-back response to achieve a MRC in the event of a failure of the ADS and/or other vehicle system that prevents the ADS from performing the DDT.

The draft also defines values for the deceleration that must not be exceeded during the execution of a minimal risk maneuver. These specifications are found in the corresponding section *Minimum Risk Maneuver (MRM) and Minimum Risk Condition (MRC)*:

[49] European Parliament and the Council: EU Regulation - Type-approval of ADS (2022).

- 5.1. During the MRM, the fully automated vehicle with the ADS shall be slowed down, with an aim of achieving a deceleration demand not greater than $4.0\,\mathrm{m\,s^{-2}}$, to a full standstill in the safest possible place taking into account surrounding traffic and road infrastructure. Higher deceleration demand values are permitted in the event of a severe ADS or severe fully automated vehicle failure.
- 5.2. The ADS shall signal its intention to place the fully automated vehicle in an MRC to occupants of the fully automated vehicle as well as to other road users in accordance with traffic rules (e.g., by activating the hazard warning lights).
- 5.3. The fully automated vehicle shall only leave the MRC after confirmation by self-checks of the ADS or/and by the on-board operator (if applicable) or remote intervention operator (if applicable) that the cause(s) of the MRM is no longer present.

In the event of a fault, the European regulation also only requires that the automated vehicle must be transitioned to the "safest possible" location. Analogous to the German legal text, this reached condition is not necessarily "safe". Again, the definition of "safety" used in this dissertation is stricter than the requirement of the EU regulation.

## 2.4 Coordinate Systems

The applied definitions for vehicle dynamics are based, with minor modifications, on the SAE J670[50]. All coordinate systems can be transformed into another coordinate system without loss of information.

### 2.4.1 Earth-Fixed Coordinate System

Earth-fixed coordinate systems are fixed to an earth-fixed reference. An Earth-Centered, Earth-Fixed (ECEF) coordinate system is fixed at the center of mass of a specified reference ellipsoid. A reference ellipsoid is used as a simple mathematical approximation of the earth's shape. The coordinates in this system can be expressed in 3D cartesian or ellipsoid notation. For improved clarity, ellipsoid coordinates are used in this thesis. As for the transformation, the 3D cartesian coordinates are converted to a geodetic latitude $\varphi$, a geodetic longitude $\lambda$, and a height $h$ using a specific reference ellipsoid. Latitude and longitude can be expressed in degrees or radians. In accordance to the International System of Units (SI), radian is used. The ellipsoid height $h$ describes the distance of a point above the reference ellipsoid in m along the ellipsoid normal

[50] SAE International: SAE J670 (2022).

defined by $\phi$ and $\lambda$. For example, if the ODD of an AV is located on the Eurasian plate, the reference system European Terrestrial Reference System 1989 (ETRS89) defined for Europe is used. The system is tied to the Eurasian plate, which moves about 2.5 cm northeast annually. ETRS89 eliminates the time dependence of the coordinates of this movement. The Geodetic Reference System 1980 (GRS80) reference ellipsoid is applied in the ETRS89 coordinate system. Fig. 2-2 shows the relationship between cartesian and ellipsoid coordinates. The position of the point P can be described either by the cartesian coordinates X, Y, Z or by the two angles $\varphi$, $\lambda$ and the height h. Also shown in gray is the shape of the reference ellipsoid.

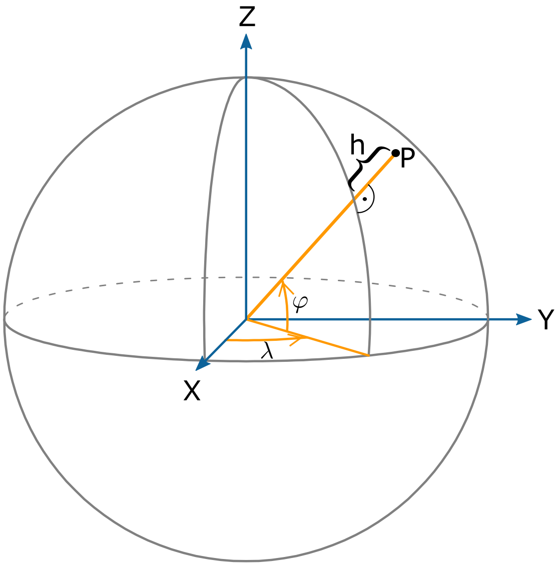


Figure 2-2: Reference Ellipsoid and Global Position P in Cartesian and Ellipsoid Coordinates[51]

The localization functions of the AV acquire the vehicle position utilizing a Global Navigation Satellite System (GNSS). These satellite systems determine the position implementing the reference ellipsoid World Geodetic System 1984 (WGS84). The reference ellipsoid GRS80 is used for the reference position specification. For this thesis, however, the two reference ellipsoids are to be considered identical.

## 2.4.2 Local Navigation System (ENU)

With a map projection, the latitude and longitude of the ellipsoid coordinates are transformed into a metric coordinate system. The axes of this local navigation coordinate system point north, east, and up (ENU). Since vehicles are ground-based, the Up coordinate can be neglected. This reduces the complexity of the required computation to be provided as a 2D system. Fig. 2-3 shows a local navigation coordinate system ENU with its origin described by a global ellipsoidal position.

[51] Adapted from Mike1024: Local tangent plane coordinates (2010)

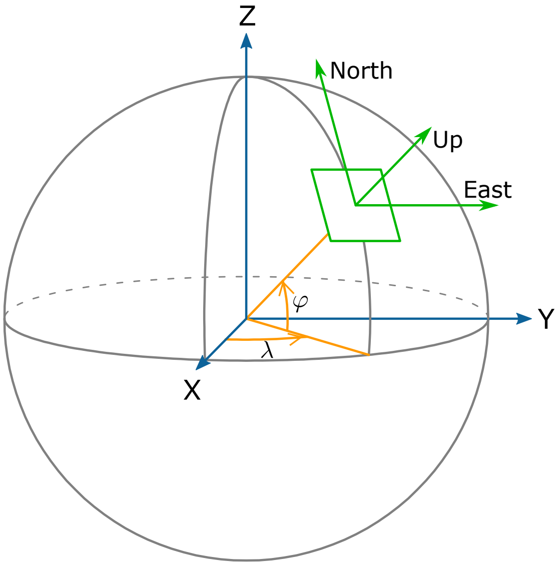


Figure 2-3: Local Navigation Coordinate System ENU with Origin at an Ellipsoid Coordinate[51]

The vehicle's orientation is also relevant for the following considerations and expresses the direction to which the front of the vehicle is aligned. In vehicle terminology, this angle is also referred to as the yaw angle. The yaw angle rotates around the up axis. The angle starts on the east axis at 0 rad. Counterclockwise the angle is positive, and clockwise it is negative. The yaw angle is defined between $[-\pi, \pi]$.

### 2.4.3 Vehicle Fixed Coordinate System

A vehicle-fixed coordinate system is fixed to a reference point of the vehicle. According to SAE J670[52], the origin of the vehicle's fixed coordinate system is at its center of mass. In the context of this thesis, this definition deviates. The reference point is at the intersection of the diagonals of the wheel contact points. The x-axis points to the front of the vehicle, the y-axis points to the left, and the z-axis points contrary to the definition of SAE J670 to the top.

## 2.5 Research Project UNICAR*agil*

The research project UNICAR*agil*[53] [54] is funded by the German Federal Ministry of Education and Research. The project goal is a disruptive modular vehicle architecture for AVs. Prototypes of modular solutions for agile, AV concepts are researched. For this purpose, a disruptive hardware and software architecture is developed. Modular driving platforms with individual

[52] SAE International: SAE J670 (2022).

[53] RWTH Aachen: UNICARagil Startseite (2022).

[54] Woopen, T. et al.: UNICARagil - Disruptive Modular Architectures for AV (2018).

dynamics modules are built. These prototypes are fully automated and driverless. Four differently assembled prototype versions are realized, a vehicle for private use (autoELF), a cab (autoTAXI), a vehicle for transporting packages (autoCARGO), and a minibus providing the capability to transport multiple passengers (autoSHUTTLE). The *Safe Halt* concept is introduced as part of this project, and it has already been considered during the design phase of the functional architecture for the modular vehicle development process. This section illustrates the foundational constraints imposed by the vehicle architectures on the *Safe Halt* functions.

### 2.5.1 Operation Design Domain (ODD)

The ODD comprises the operational environment of the AV. For UNICAR*agil*, an inner-city ODD is anticipated. Complex situations are to be expected during driving missions in inner-cities. Structural separation of lanes can only be expected in some places. A variety of object classes can be expected. This includes other road users and additional subsets of identifiable objects of other types. Speeds are lower in inner-city areas than on freeways. Other road users are expected to travel at speeds up to $20\,\mathrm{m\,s^{-1}}$. The basic concept of the AV of UNICAR*agil* does not limit the time of day and weather conditions. However, within the scope of the research project, the ODD was limited to driving missions at daylight. Additionally, to keep the design complexity minimal during the prototyping phase, the test environment exclude weather conditions of ODD, such as rain, snow, and fog.

### 2.5.2 Agile Automated Vehicles (AV)

The disruptive hardware and software architectures enable a widespread demonstration of agile vehicle behavior. The prototype vehicles have four dynamic modules. Each dynamic module can accelerate and decelerate the vehicle and be steered separately. While driving, each wheel can be steered with a steering angle between $[-60°, 60°]$. These abilities allow the sideslip angle of the vehicle to be considered as an additional degree of freedom. Thus, a vehicle with these abilities is able to follow a path not only spatially but also with an independently defined yaw angle. This characteristic is relevant for describing the vehicle's desired behavior through trajectories. It is insufficient to have only a position reference as part of the trajectory but not the yaw angle. The wheels can also be steered in opposite directions when the vehicle is in standstill. This characteristic allows for a configuration in which all wheels are set at steering angles of 90°. The sideslip angle thus becomes a maximum of 90°. These steering angles allow the vehicle to be driven sideways, for example, for parking operations. The interiors of the vehicle prototypes differ fundamentally from non-AVs. Due to the intended automation of the driving task, a human driver cannot take over the vehicle's control. As a result, there is no steering wheel and no pedals for longitudinal and lateral control in the vehicle interior. However, this is only true for the concept of the UNICAR*agil* vehicles. For the purpose of test operation, two test driver seats are installed into the vehicle, allowing the vehicle to be driven manually.

### 2.5.3 Automotive Service-Oriented Software Architecture (ASOA)

ASOA[55] [56] is a middleware for automotive applications. With this middleware, different vehicle functions can communicate with each other to exchange data. The UNICAR*agil* project implements ASOA in the functions of the ADS. In this software architecture, functions of an ADS are implemented as services. A service has different life cycles. This can be described as different states or properties, such as being paused or active. Implemented interfaces allow for communication between services. Inputs are called requirements, and outputs are called guarantees. The guarantees of one service can be connected to the requirements of one or more other services. Three different types of information are transmitted through the interfaces. The first type contains the user data. The transmission of user data is the primary task of the middleware. Their content includes all data that subsequent functions can process. The second type includes the quality data for the user data. Quality data describes the evaluated quality of the user data, e.g., accuracies of the user data. The third type of information represents the parameters capturing all constants affecting user data generation. The parameters do not change during operation. The middleware allows defining custom interfaces. Complex data types can be transmitted and implemented as custom data structures. The content of the structures can be defined as desired within the definition of the Interface Definition Language (IDL)[57]. The interface definitions between the guarantee of one service and the requirement of another service shall be identical. Another important part of the middleware is the orchestrator. The task of the orchestrator is orchestrating the services within the middleware. For this purpose, the orchestrator can connect service interfaces. Different vehicle operation modes can be realized with the help of this service composition. For this purpose, an operating mode management system[58] is integrated into the ASOA. The operating mode management system serves as a central decision-making instance and takes decisions on which operating mode shall be active at any given moment. In addition, the orchestrator can change the service's current life cycle phase, i.e., to start or pause the services. In order to keep the benefits of modularity, the services cannot be aware of the operating mode they are a part of. Only by orchestrating the services, the required vehicle operation modes are established. Following the guidelines of an ASOA, it shall therefore be guaranteed that the service can execute its functions without knowledge about the current vehicle operation mode.

### 2.5.4 Vehicle Operating Modes

By orchestrating the services, different vehicle operation modes can be realized. The following operating modes are formed for UNICAR*agil*:

[55] Kampmann, A. et al.: Automotive Service-Oriented Software Architecture (2019).

[56] Mokhtarian, A. et al.: Service-oriented Software Architecture for UNICARagil (2020).

[57] Object Management Group: Interface Definition Language (IDL) (2022).

[58] Jatzkowski, I. et al.: Vehicle Operating Mode Management in ASOA (2021).

1. Automated Driving
2. Teleoperation
3. Safe Halt

In the *Automated Driving* mode, the orchestrator connects all the necessary services to fulfill the functions of this operating mode and starts the services. With this operation mode, the DDT are executed by automation. Based on a driving mission, the functions of tactical and strategic planning, trajectory tracking control, and actuation are executed. This mode of operation is the default mode for the AV. In the *Teleoperation* mode, the vehicle control is performed remotely from within a control room. In this case, the orchestrator also connects the services to execute the necessary functions for teleoperation. A human operator in the control room can contact the vehicle via a wireless connection. This connection can also be used to communicate with the vehicle's passengers. The operator observes a digital representation of the environment as detected by the vehicle's environment sensors. The operator has two options for taking over motion control of the vehicle. In the first option, the operator sends trajectories to the vehicle, which are executed by the trajectory tracking control. For this purpose, the control room shall plan trajectories that match the global dynamic state of the vehicle. In the second option, the vehicle is directly commanded by the operator similar to if he would be the actual driver. For this, the control room operator remotely controls the vehicle using a steering wheel and pedals like a non-AV. The third mode is called *Safe Halt* (cf. Sec. 2.5.6). It is named after the concept under evaluation for enabling the fail-safe property of an ADS. The orchestrator switches to this mode when the vehicle's abilities are insufficient for the automated driving mode. Connecting the required services allows the *Safe Halt* functions to be executed.

### 2.5.5 The UNICAR*agil* ADS

The functional architecture of the UNICAR*agil* ADS includes ASOA services to execute of the DDT in the vehicle operating modes *Automated Driving*, *Teleoperation* and *Safe Halt*. The functional architecture of the ADS is implemented as an A-model. This implementation is illustrated in Woopen et al.[59].

### 2.5.6 The Safe Halt Concept

The original concept of *Safe Halt* is described in Woopen et al.[60]. The author of this thesis is also co-author of that paper. The *Safe Halt* concept is a proposed solution for a DDT fallback as described in SAE J3016[61]. *Safe Halt* transitions an AV to a MRC when the ADS cannot continue

[59] Woopen, T. et al.: UNICARagil - Disruptive Modular Architectures for AV (2018), p. 6, fig. 3.

[60] Woopen, T. et al.: UNICARagil - Disruptive Modular Architectures for AV (2018).

[61] SAE International: SAE J3016 (2021), p. 10.

the driving mission within the defined acceptable risk. For this purpose, a higher-level system monitors the capabilities of the ADS on whose information the operating mode is switched from *Automated Driving* to *Safe Halt* mode. For the UNICAR*agil* ADS, this task is performed by a self-perception system, based on Nolte et al.[62]. *Safe Halt* is not permanently integrated into the effect chain for performing the DDT and thus is a cold redundancy. Nevertheless, it shall be ensured that the *Safe Halt* capabilities are permanently monitored so that an activatability of the *Safe Halt* is guaranteed.

The *Safe Halt* concept requires planning an implicit emergency trajectory that originates in the current vehicle state and terminates in the MRC. Therefore, in addition to the trajectory for the *Automated Driving* mode, the planner of the ADS additionally plans an implicit emergency trajectory. The planner of the implicit emergency trajectories shall guarantee the planning of implicit emergency trajectories continuously in the ODD of the AV. The pose course of this implicit emergency trajectory shall be passable. *Safe Halt* does not check the passability. The planner also plans a maximum speed profile for the implicit emergency trajectory. *Safe Halt* shall not plan trajectories whose velocities are above this maximum velocity profile. The planned implicit emergency trajectory is transmitted to *Safe Halt* and cached there. Once activated, *Safe Halt* takes over vehicle control based on the most recent implicit emergency trajectory. It plans trajectories based on the pose course of the most recent implicit emergency trajectory. Deviation from this pose course is excluded. A unique feature of the *Safe Halt* concept is its independent environment perception system, which is used only for monitoring the implicit emergency trajectory for collision objects. If objects are detected on the course of implicit trajectory, *Safe Halt* plans appropriate collision-avoiding trajectories. *Safe Halt* can thus perform a MRM to a MRC independent of the ADS primary environment perception system and strategic and tactical planning.

The maximum speed profile of the implicit emergency trajectory is planned to reduce the vehicle speed quickly to a creep speed at the beginning of the trajectory. The implicit emergency trajectory is then tracked with this creep speed and the vehicle is comfortably stopped at the end of the trajectory. The quick deceleration at the beginning dissipates the vehicle's kinetic energy to reduce risk. The creep speed is as low as possible to reduce the risk in order to lead the vehicle swiftly out of the flow of traffic, but at the same time, as fast as the environment perception system of Safe Halt allows, to not be a surprisingly slow obstacle for the following traffic. If the *Teleoperation* vehicle mode is available, a human operator can remotely take over vehicle control at the end of the MRM and resolve the situation further.

*Safe Halt* can take over vehicle control, if the ADS's capabilities are no longer sufficient to execute the driving mission and *Safe Halt*'s capabilities are more significant than the remaining ADS's capabilities. However, *Safe Halt* is not required to be fail-operational. If the capabilities of

[62] Nolte, M. et al.: Supporting Safe Decision Making (2020).

*Safe Halt* are not sufficient to execute the MRM, the other functions of the ADS used for mission execution shall abort the driving mission and execute a MRM.

## 2.6 IDEFØ Function Modeling Methodology

Various modeling languages are used in systems and software engineering. One family of these modeling languages is called Integration Definition (IDEF)[63]. The content of this section provides a summary about IDEF.

The IDEF family of modeling languages was derived from the Structured Analysis and Design Technique (SADT)[64]. IDEF can be used to model different views of systems. In this modeling language, boxes symbolize functions, and arrows symbolize the flow of information or products. For the purpose of functional modeling of systems IDEFØ is used. This subset of IDEF is tailored to analyze the functional perspectives of a system. Both experts and customers can communicate with each other using the graphical representation of the functional relationships. Its representation supports a developer to identify which functions a system should perform and what is needed to perform them. Fig. 2-4 shows an example system in the IDEFØ representation.

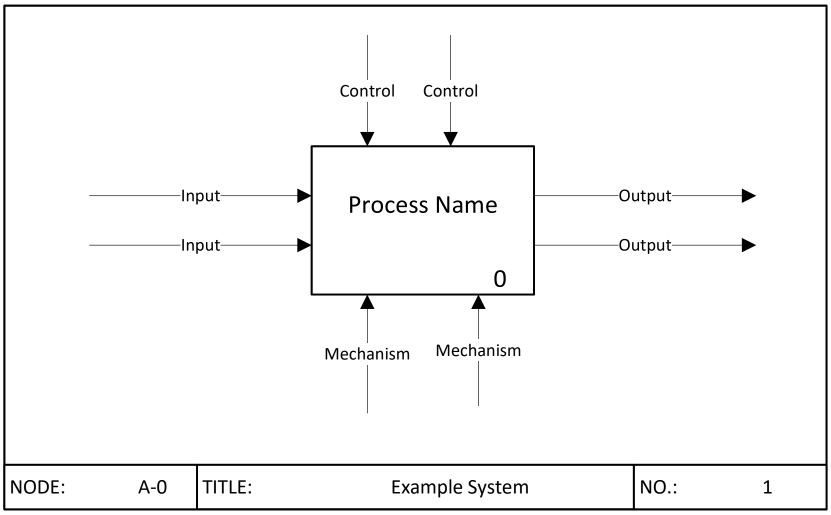


Figure 2-4: IDEFØ Representation of an Example System[65]

Any component of the IDEFØ modeling language is realized as either a box or an arrow. The label inside a box describes the function to be performed. Conceptually, a box does represent a node. Arrows leading into a box from the left are inputs. The function of the box transforms inputs. On the right side of the box are exclusively the outputs. Outputs are the transformed

[63] Knowledge Based Systems, Inc.: IDEF0 Website (2022).

[64] Ross, D.: Structured Analysis (SA) (1977).

[65] Own illustration according to United States Government Army: Systems Engineering Fundamentals (2001) p. 51

inputs. Arrows coming from the top are controls. These are constraints for the performance of the functions. They are different from inputs because controls do not change due to the function. Arrows coming from below describe the mechanisms for performing the function. In particular, these are the resources needed to perform the function. All arrows are labeled according to their context.

The boxes can be connected. In addition, the multi-view representation of the IDEFØ language allows for a detailed node-like overview of the designed system. The system under consideration has a node name followed by a hyphen and the number 0 (e.g., A-0). This system has nodes at the top level, numbered from 1 integer upward. Node 1 has the node name A1 in the first sublevel. Nodes within A1 are again numbered upwards from 1 integer. Node 1 of this second sublevel has the node name A11. This method of node description is continued for all further sub-levels and allows systems to be modeled in hierarchical functional perspectives.

# 3 Methodology and Research Tools

This chapter examines the working hypothesis in more detail based on the fundamentals presented. In the context of this thesis, a corroboration of the working hypothesis is conducted. In Sec. 3.1, the research problem is analyzed. The following Sec. 3.2 then presents the methodology for evaluating *Safe Halt*.

## 3.1 Problem Analysis

In the state of the art, the concept *Safe Halt* is described as a way to autonomously maintain the safe state of an Automated Vehicle (AV) and to perform a Minimal Risk Maneuver (MRM) when required. However, no evaluation of this concept exists to date. In turn, there needs to be more metrics to support any evaluation that would allow the concept to be assessed. After these metrics have been identified, a reference solution for Safe Halt has to be implemented where these metrics can be applied. Finally, a proper evaluation of the concept shall be performed, and a statement shall be made as to whether this concept is able to fulfill all requirements defined in the previous chapters to serve as a legally safe Dynamic Driving Task (DDT) fallback to enable the fail-safe property for AVs. Weaknesses and shortcomings in the concept should be identified and lead to an update of the original concept. The iteration process is concluded if either all requirements are satisfied and all limitations are resolved or if a particular requirement cannot be fulfilled within the scope of this thesis.

## 3.2 Methodology for the Evaluation of Safe Halt

For reusability of the findings in other functional Automated Driving System (ADS) architectures, the methodology distinguishes between a generic ADS and the UNICAR*agil* ADS. The thesis is structured so that at the beginning of each chapter a generic ADS is analyzed. As a specific application of the generic ADS with specific constraints, the UNICAR*agil* ADS is evaluated afterward. Since verification of the findings requires the implementation of a reference solution of the *Safe Halt* concept, the working hypothesis is only applied to the UNICAR*agil* ADS. The methodology for evaluating *Safe Halt* is presented in Fig. 3-1.

In order to assess the property contribution, the fault tolerance regime definitions from Subsec. 2.1.8 are applied to a generic and the UNICAR*agil* ADS. For this purpose, the concepts of function and performance are considered in the context of the ADS. It shall first be defined as

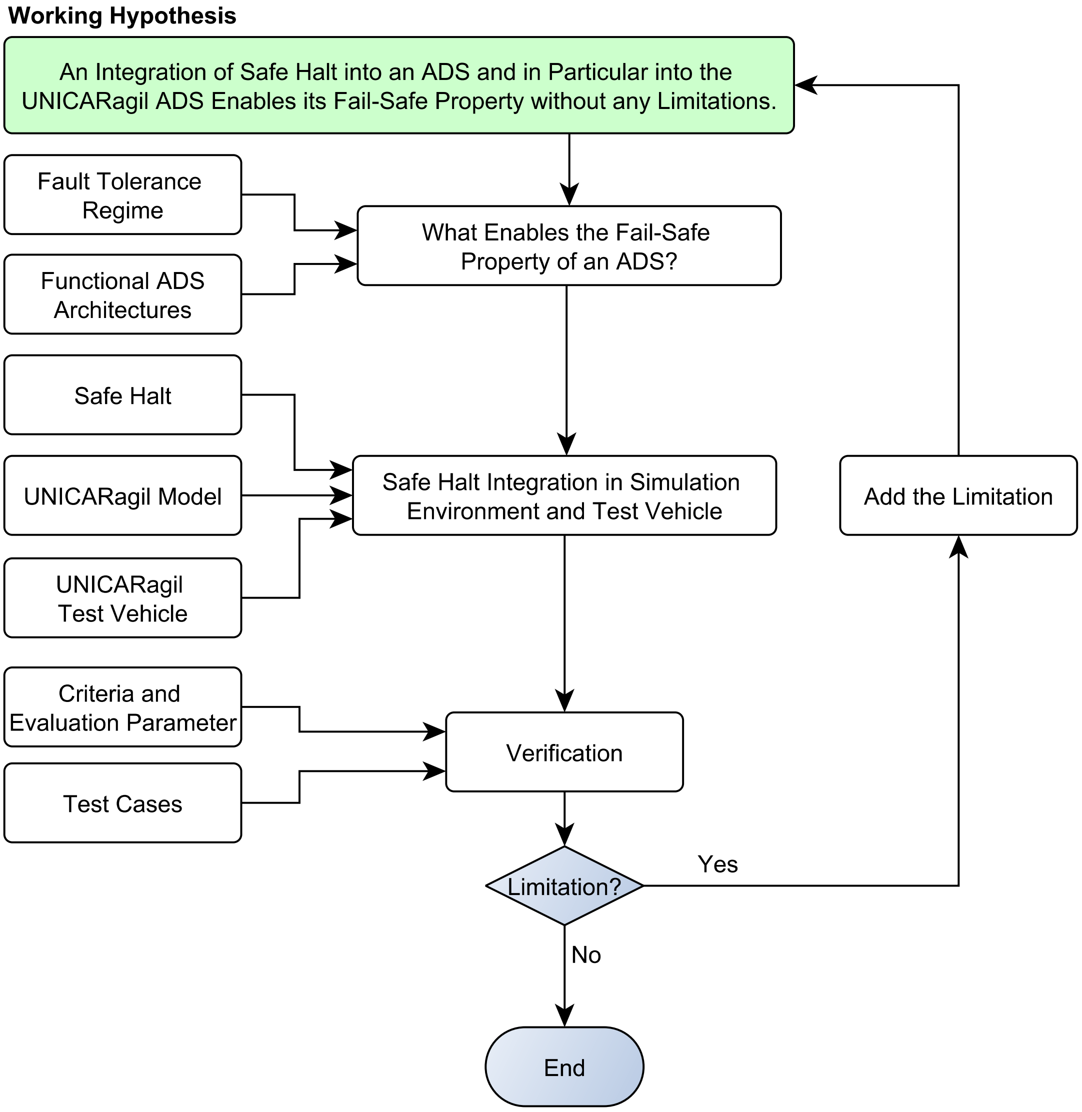

Working Hypothesis
An Integration of Safe Halt into an ADS and in Particular into the UNICARagil ADS Enables its Fail-Safe Property without any Limitations.
Fault Tolerance Regime
Functional ADS Architectures
What Enables the Fail-Safe Property of an ADS?
Safe Halt
UNICARagil Model
UNICARagil Test Vehicle
Safe Halt Integration in Simulation Environment and Test Vehicle
Add the Limitation
Criteria and Evaluation Parameter
Test Cases
Verification
Limitation?
Yes
No
End

Figure 3-1: Methodology for the Evaluation of Safe Halt

how a fail-safe property of a complex system such as an ADS manifests itself. Next, a reference solution for *Safe Halt* is established. For this purpose, the system boundaries are defined first. Subsequently, the requirements for Safe Halt are determined. This determination is conducted for a generic ADS and the UNICAR*agil* ADS. For the application in UNICAR*agil*, the concept and the reference solution of *Safe Halt* is presented. This presentation is followed by integrating the reference solution into Software in the Loop (SiL) and Hardware in the Loop (HiL) test environments and finally into the four prototype test vehicles of the project. Following this is the elaboration of test criteria and test cases to evaluate the reference solution for UNICAR*agil*. With this setup, the limitations of the concept can be identified. Finally, these are used to update the original concept and guide future research needs.

# 4 Fault Tolerance Regimes for Automated Driving System (ADS)

Following the research methodology, this chapter analyzes the fault tolerance regime of an ADS. Within the scope of this thesis, the ADS is one of the most critical components of an Automated Vehicle (AV). As such, this chapter explains it in more detail, mainly focusing on its fail-safe property and its impact on the proposed *Safe Halt*. For this purpose, it is defined how the fail-safe property of an ADS is manifested. The basic principles of fault tolerance regimes have been presented in Sec. 2.1.8 and in this chapter these principles are applied to two different configurations of an ADS. Reference for the following considerations is the publication by Stolte et al.[66] to which the author of this dissertation also contributed. In Sec. 4.1, the fault-tolerance regime is applied to the functional architecture of a generic ADS. The following Sec. 4.2 uses the UNICAR*agil* ADS as a specific functional architecture to apply the fault-tolerance regime to and analyze the differences between it and a generic ADS.

## 4.1 Generic ADS

This section provides an analysis of the fault tolerance of a generic ADS. For this purpose, different generic functional architectures of ADS are presented. Subsequently, the fault tolerance of these ADS is analyzed, particularly concerning the fail-safe property. Then, fault combinations of a generic ADS are introduced, and the common handling approaches of those combinations are described.

### 4.1.1 Purpose of a Generic ADS

The primary task of any ADS is the execution of a driving mission This mission has goals, but it must have at least one goal. In ADS terminology, a goal is a trip's destination. In order to reach the mission goal, an ADS applies tactical and strategic planning procedures. It is of note, that the mission goal is not generated within an ADS and instead always considered as one of the inputs to the ADS. The mission goal is usually provided by a human user.

Fig. 4-1 illustrates a generic ADS in the IDEFØ modeling language. Inputs of the ADS are the driving mission, information about the vehicle environment, and vehicle movement. The output of the system is the vehicle movement in that environment. Behavioral rules (e.g., traffic rules), a time reference, and vehicle parameters (e.g., dimensions of the AV) are added as controls. Energy

[66] Stolte, T. et al.: Taxonomy to Unify Fault Tolerance Regimes for Automotive Systems (2022).

and computing power are provided to the ADS as mechanisms. Four different ADS configurations are introduced and serve as comparable references throughout this thesis. Following the naming conventions presented in Sec. 2.6, each configuration name starts with one of the letters A, B, C, or U. Configuration A represents a generic ADS without the *Safe Halt* functionality. Configuration B is a generically configured ADS with the *Safe Halt* functionality. Configuration U is the functional architecture used in the context of UNICAR*agil*. Finally, configuration C extends the functionality of *Safe Halt* to include a *Safety Monitor*. A detailed evaluation of this last configuration C is outside the scope of this thesis. However, a basic version of it is included in the outlook of this thesis.

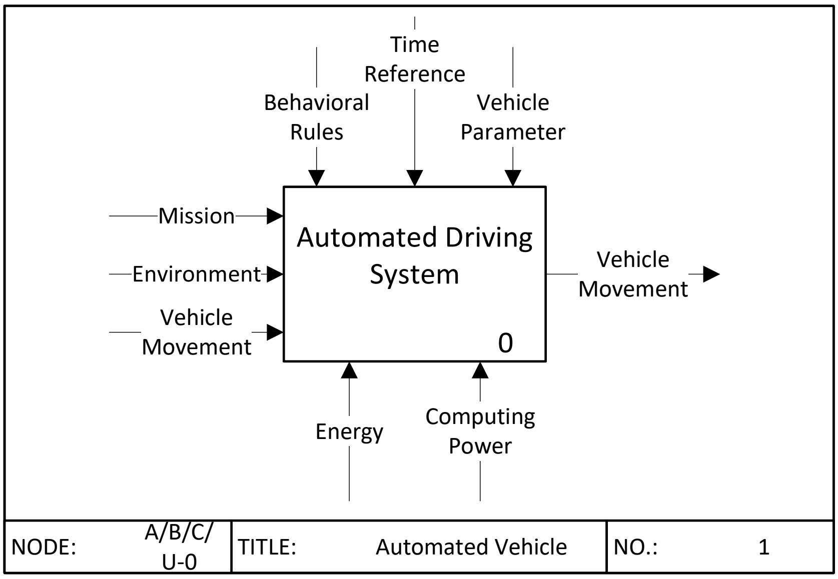


Figure 4-1: A Generic ADS as Part of an AV in the IDEFØ Representation

For the ADS to have a purpose, a set of missions $\mathscr{M}_{\text{nom}}$ within its Operational Design Domain (ODD) shall be selectable by the user. Furthermore, for each mission $m$, a mission quality $q_{\text{nom}}$ is defined so that $\forall\, m \in \mathscr{M}_{\text{nom}}\ \exists\ q_{\text{nom}}(m)$. Depending on its implementation, the mission quality may contain, i.a., the mission execution time, energy consumption, or indicators for driving comfort.

## 4.1.2 Functional ADS Architectures

Various functional architectures have been published to configure ADS. A well-known paradigm is *Sense, Plan, Act*, first used in the Shakey robot by Nilsson[67]. It is shown in Fig. 4-2.

A robot senses its environment, plans its following actions, and then executes them. Afterward, the cycle starts all over again. During each step, the robot plans its next step.

[67] Nilsson, N. J.: A mobile automation (1969).

[68] Own illustration according to Murphy, R.: Introduction to AI robotics (2000), p. 7, fig. I.3.

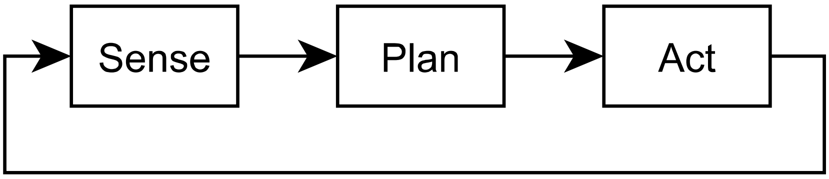


Figure 4-2: Functional Architecture Sense, Plan, Act[68]

Based on this basic paradigm, functional architectures were derived. The publication by Matthaei et al.[69] proposes modular building blocks to describe a functional architecture. It was developed in a top-down approach based on the functional requirements of an ADS. Modular functional blocks are also used in this functional architecture. Different abstraction levels are described for the fulfillment of their functions. Additionally, different time horizons are assigned to the levels. Mission planning and tactical planning take place with comparatively large time horizons. On the other hand, strategic and executive planning takes place in concise time horizons.

A model of a generic functional ADS architecture in the IDEFØ modeling language is provided in Fig. 4-3.

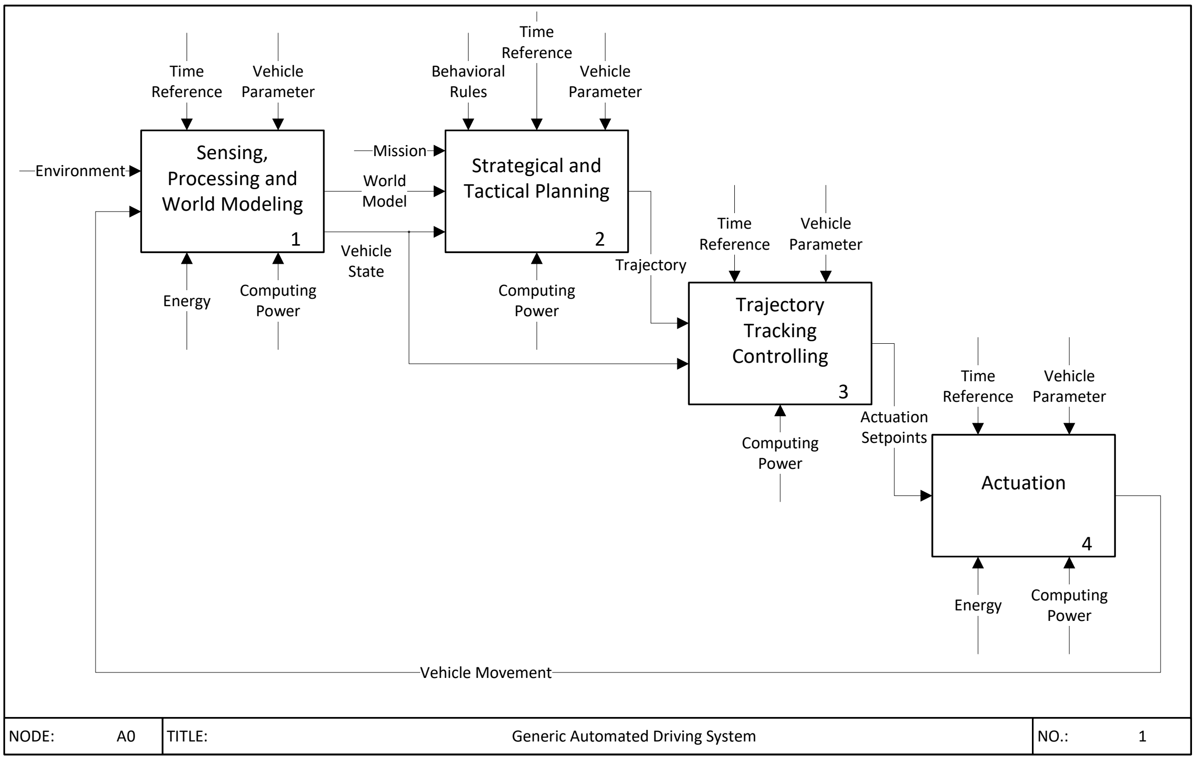


Figure 4-3: The Functional Architecture of a Generic ADS in the IDEFØ Representation

Illustrated are the functional blocks required to perform the functionality of the ADS. Node A1 *Sensing, Processing and World Modeling* performs the functions of sensing, processing, and modeling the environment. A world model is provided as the output of this node. The node also

[69] Matthaei, R.; Maurer, M.: Autonomous driving a top-down-approach (2015), p. 159, fig. 1.

performs the function of determining its vehicle state. This determination includes, for example, the momentary vehicle pose. Node A2 *Strategical and Tactical Planning* obtains the mission goal, the world model, and the vehicle state at its input. This node uses all of its input data to perform strategic and tactical planning to generate trajectories for the mission. Behavioral rules restrict the planning. At the output, a trajectory is provided. Node A3 *Trajectory Tracking Controlling* receives the generated trajectory as its input, and based on it, this node computes the setpoints for the motion actuators. Finally, these setpoints are transformed into the vehicle motion in node A4 *Actuation*. Each node is given the necessary controls and mechanisms that enable the function's performance.

### 4.1.3 Analysis of the Generic ADS Fault Tolerance Regime

Adopted from on the fault tolerance regime definitions[70a] and Sec. 2.1.8, this section analyzes the fault tolerance of a generic ADS. For this purpose, the performance of an ADS is defined first. With $\vec{q}_{\mathrm{m,nom}}$ containing all $q_{\mathrm{nom}}(m)$, $p_{\mathrm{ADS,nom}} = p_{\mathrm{ADS}}(n_{\mathrm{nom}}, \vec{q}_{\mathrm{m,nom}})$ is the nominal performance of the ADS, where $n_{\mathrm{nom}} = |\mathscr{M}_{\mathrm{nom}}|$ denotes the number of nominally selectable missions. Consequently, the available performance $p_{\mathrm{ADS,a}}(f)$ in the presence of a fault combination $f$ can be described as $p_{\mathrm{ADS,a}}(f) = p_{\mathrm{ADS}}(n_{\mathrm{a}}(f), \vec{q}_{\mathrm{m,a}}(f))$, where $\vec{q}_{\mathrm{m,a}}(f)$ contains the achievable mission quality in the presence of the fault combination $f$ $\forall m \in \mathscr{M}_{\mathrm{nom}}$. If the ADS is subject to a fault combination $f_{\mathrm{ADS}}$, it can affect both performance measures, the number of missions available, and the mission quality that can be achieved. If $p_{\mathrm{ADS,a}} = p_{\mathrm{ADS}}(f_{\mathrm{ADS}}) \geq p_{\mathrm{ADS,nom}}$, the automated driving system is fail-operational for the fault combination $f_{\mathrm{ADS}}$. This means that the ADS can perform all specified missions in its ODD with at least nominal mission quality, yielding $n_{\mathrm{a}}(f_{\mathrm{ADS}}) = n_{\mathrm{nom}}$ and $\vec{q}_{\mathrm{m,a}}(f_{\mathrm{ADS}}) \geq \vec{q}_{\mathrm{m,nom}}$. The ADS is fail-degraded when the fault combination $f_{\mathrm{ADS}}$ leads to a particular set of missions being infeasible, yielding $\mathscr{M}_{\mathrm{a}} \subsetneq \mathscr{M}_{\mathrm{nom}}$ with $\mathscr{M}_{\mathrm{a}} \neq \emptyset$, or reduces the achievable quality for at least one available mission $\vec{q}_{\mathrm{m,a}}(f_{\mathrm{ADS}}) < \vec{q}_{\mathrm{m,nom}}$. Then, the ADS is still functional, $o_{\mathrm{ADS}}(f_{\mathrm{ADS}}) = o_{\mathrm{ADS}}(f_{\mathrm{ADS}}) = 1$, yet with decreased available performance $p_{\mathrm{ADS,a}}(f_{\mathrm{ADS}}) < p_{\mathrm{ADS,nom}}$. Since $o_{\mathrm{ADS}}(f_{\mathrm{ADS}}) = 1$, it is expected that the ADS and, thus, the AV maintain a safe state. The ADS has a fail-safe property when, while maintaining the safe state, a fault combination $f_{\mathrm{ADS}}$ results in all missions being infeasible, giving $\mathscr{M}_{\mathrm{a}} = \emptyset$, or the mission quality for the current mission is unacceptable and therefore below a quality threshold $q_{\mathrm{m,min}}$, resulting in $q_{\mathrm{m,a}}(f_{\mathrm{ADS}}) < q_{\mathrm{m,min}}$. This results in $o_{\mathrm{ADS}}(f_{\mathrm{ADS}}) = 0$.

### 4.1.4 Fault Combinations of a Generic ADS

This subsection further discusses the results from the application section of the article by Stolte et al.[70b]. To execute a driving mission, all architectures have functions for sensing the environment, strategic and tactical planning, and actuation (Sense, Plan, Act). In these systems, a fault

[70] Stolte, T. et al.: Taxonomy to Unify Fault Tolerance Regimes for Automotive Systems (2022). a: -; b: p. 9.

combination $f_{\mathrm{ADS}}$ can occur in any of the listed functions. According to Sec. 4.1.3, fault tolerance can only be specified for specific fault combinations $f_{\mathrm{ADS}}$. Therefore, each fault combination must be examined to determine the fault tolerance property of an ADS. In an ADS, $\mathscr{F}$ independent faults can occur. Thus, $\mathscr{P}(\mathscr{F}) = \{f \mid f \subseteq \mathscr{F}\}$ is the set of possible fault combinations $f$. Fault combinations can occur throughout the entire effect chain of the ADS that is shown in Fig. 4-3. Faults manifest not only by degradation or failures of components and systems in the effect chain but also by the fact that an ADS may end up in a situation it cannot handle. In particular, with the increased use of artificial intelligence in perception systems and fusion, situations that were not considered during verification of the ADS can arise. In the International Organization for Standardization (ISO)/Publicly Available Specification (PAS) 21448[71], a distinction is made between known and unknown situations, each of which may be safe or unsafe. Significantly unsafe and unknown situations increase the risk, which is why they are considered faults. Other faults that may occur are due to ambient conditions. Fog and rain, for example, affect lidar sensors. As such, the environment sensors may be limited in their function. Because these limitations can propagate throughout the effect chain of the ADS, they are also considered possible faults. This view on distinguishable faults results in many potential fault combinations.

### 4.1.5 Handling Fault Combinations of a Generic ADS

For the treatment of each fault combination, the effects of the combination on the entire effect chain shall be analyzed. For this purpose, the set containing all known fault combinations must first be generated. Environmental sensors interact with the external system environment. The number of distinguishable faults is, therefore, substantial. A combination of these faults thus leads to an extensive set. Each fault combination shall be handled by the ADS, and therefore for each fault combination, it shall be verified what fault tolerance the ADS has for that. The difficulty is to detect all distinguishable fault combinations effectively. For the detection of these fault combinations, there are self-representation concepts like holistic system-level representations and monitoring by Nolte et al.[72]. Also, there are various concepts for online verification[73] [74] [75] to safeguard the trajectory generated by an ADS. With the help of these approaches, fault combinations in the generation of the intended behavior can be uncovered.

This thesis focuses on providing the fail-safe property of an ADS. This property is set, if the ADS can maintain the safe state for a fault combination and the AV can additionally transition into a Minimal Risk Condition (MRC). Effective fault combination handling is characterized by providing the fail-safe property for many fault combinations. Fig. 4-4 shows two sets of fault

[71] International Organization for Standardization (ISO): ISO/PAS 21448:2019 (2019), p. 7.

[72] Nolte, M. et al.: Supporting Safe Decision Making (2020).

[73] Popp, C. et al.: Approach to Maintain a Safe State of an Automated Vehicle (2022).

[74] Popp, C.: Safety-Check von Trajektorien beim Automatisierten Fahren (2023).

[75] Stahl, Tim Nikolaus: Safeguarding complex and learning-based automated driving functions (2022).

combinations. The more extensive one is the super set containing all known fault combinations, which can occur in the ADS. Depicted as the smaller subset is the collection of known fault combinations for which the ADS has a fail-safe property.

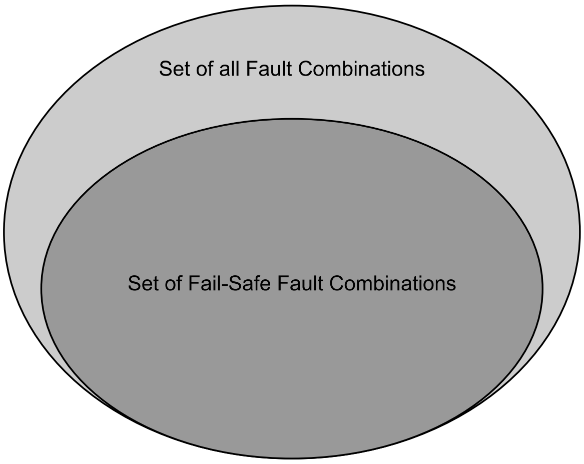

Set of all Fault Combinations
Set of Fail-Safe Fault Combinations

Figure 4-4: The Set of All Fault Combinations and the Fail-Safe Subset

There are two strategies to increase the set of fault combinations for which the ADS is fail-safe. The first strategy is to integrate a redundant ADS into the AV, shown in Fig. 4-1. This strategy can handle all fault combinations of functions and supporting mechanisms. Non-resolvable in redundant systems are systematic fault combinations resulting, e.g., from unknown unsafe situations. To handle these, the substitute system would need other capabilities than the primary ADS. That a substitute system can handle an unknown unsafe situation is not to be expected since, in this case also, the primary system would have received the function for handling the situation.

If the redundant system is successfully activated, the driving mission must be aborted in any case to obtain a MRC. This must be done because the substitute system no longer has another redundant system to fall back to and, consequently, must be considered unsafe. A failure of the substitute system can have fatal consequences.

The second strategy for handling fault combinations is to stop the vehicle in place. For this purpose, the AV is decelerated from the initial state to a standstill. This emergency stop maneuver may bring the vehicle to its dynamics limit. Furthermore, the vehicle may no longer follow the designated lane. In addition, a hazard to the occupants and following traffic arises in case the AV suddenly performs an emergency deceleration. An emergency deceleration in an ODD with low absolute speed is easier to adjust to the following traffic than ODD with high absolute speed, such as on highways. Additionally, the standstill location depends exclusively on the initial state of the AV and the deceleration parameters. As such, depending on the ODD, the probability of coming to a standstill in a safe location can vary.

This thesis explores a third strategy for enabling a fail-safe property for an ADS. The proposed concept has the name *Safe Halt*. In this strategy, an additional node named *Safe Halt* is added to the generic architecture of an ADS, as shown in Fig. 4-5. *Safe Halt* as node B1 receives an

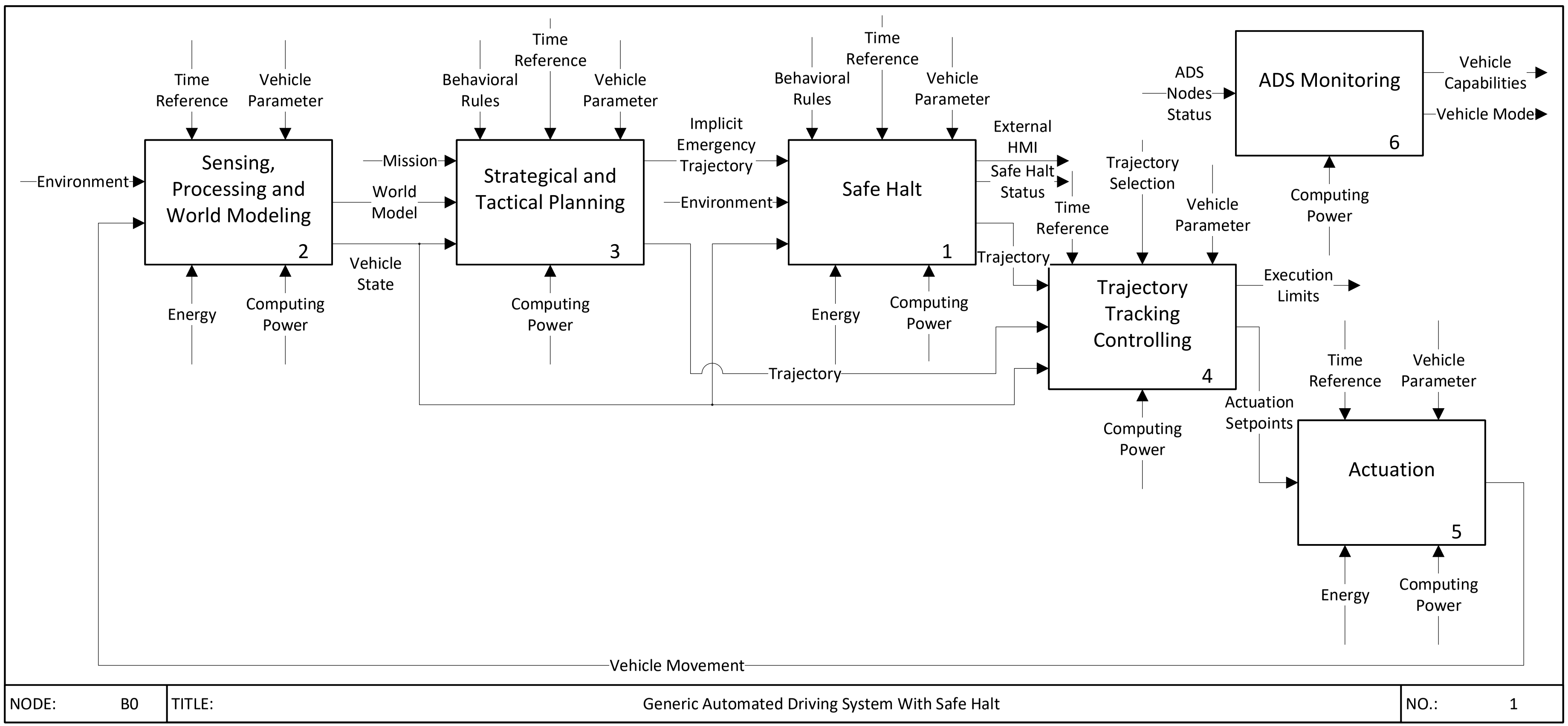


Figure 4-5: The Functional Architecture of a Generic ADS With Safe Halt in the IDEFØ Representation

implicit emergency trajectory with a supplementary maximum speed profile as inputs from node B3 *Strategical and Tactical Planning*. The implicit emergency trajectory and the maximum velocity profile originate in the vehicle state at the planning time. The implicit emergency trajectory describes a pose course from the initial vehicle pose to a minimum risk condition. The maximum velocity profile provides a strong initial deceleration of the vehicle for a rapid velocity reduction of the AV. Subsequently, the remaining low creep velocity is used to approach the MRC. Finally, with a comfortable deceleration, the vehicle is stopped in this MRC. This vehicle transition into the MRC is called a Minimal Risk Maneuver (MRM). To maintain an acceptable risk during the MRM, the spatial course of the implicit emergency trajectory shall be monitored for collision objects. Node B1 *Safe Halt* thus receives information about the *Environment* as input. As a third input, node B1 receives the current vehicle state. Using behavior rules and vehicle parameters as controls, *Safe Halt* outputs its generated trajectories and passes them on to node B4 *Trajectory Tracking Controlling*. In addition, *Safe Halt* monitors its capabilities for executing its functions and provides them to the ADS. Suppose fault combinations occur in the ADS which require the mission to be aborted and a MRC to be reached. In that case, *Safe Halt* will use the last obtained implicit emergency trajectory, maximum velocity profile, and implicit emergency trajectory monitoring to generate trajectories until the MRC is reached. *Safe Halt* can thus maintain the safe state of the ADS despite the complete failure of node B3 and almost complete failure of node B2 and additionally transition the vehicle to a MRC. Node B2 must at least still be able to provide the current vehicle state for *Safe Halt* to perform its function. Not covered are fault combinations of node B4 and node B5. For the *Safe Halt* concept, it is assumed that both nodes can be made fail-operational with reasonable effort.

## 4.2 UNICAR*agil* ADS

This section explores the fault tolerance of the UNICAR*agil* ADS and covers the purpose of this ADS, possible fault combinations, and their fault handling.

### 4.2.1 Purpose of the UNICAR*agil* ADS

The purpose of the UNICAR*agil* ADS is to execute a mission. Four different applications of automated vehicles are demonstrated for the project. Depending on the application, the purpose of the ADS differs. Three applications are meant for transporting people, and one is meant for transporting goods. The ODD is considered to be identical for all applications. The UNICAR*agil* concept aims, in particular, at applying urban and rural transport. Mission planning depends on the vehicle application context. Individual user-specific missions are planned for the private vehicle autoELF and cab autoTAXI. The mission of the autoSHUTTLE is to transport a group of passengers to the desired destination as efficiently as possible. In the case of the goods transporter autoCARGO, mission planning focuses, in particular, on efficient logistics.

### 4.2.2 Functional UNICAR*agil* ADS Architecture

For the project UNICAR*agil*, the A-model[76a] functional architecture is presented. The architecture combines the three classical levels according to Donges[77] with the increased demands to perception and information processing of AV[76b]. The three traditional levels cerebrum, brain stem, and spinal cord are introduced, whose names are based on the human organs with similar functions. The cerebrum comprises the perception of the environment and the tactical and strategic planning of behavior. The brain stem implements the planned behavior through appropriate control. Finally, the actuator system at the spinal cord level commands the actuators for vehicle motion. Supporting external functions is a cloud with a collective traffic memory. A model of the functional UNICAR*agil* ADS architecture in the IDEFØ modeling language is provided on Fig. 4-6.

In contrast to the generic ADS depicted on Fig. 4-5, node U2 *Sensing, Processing and World Modeling* provides the vehicle pose as output. This is the localization function #1 of the UNICAR*agil* ADS. The additional node U6 *Vehicle Dynamics State Estimation* provides the current vehicle dynamics state of the AV and is the localization function #2. Both are global localization functions based on the Geographic Coordinate System (GCS) European Terrestrial Reference System 1989 (ETRS89). Since the earth can be assumed to be flat in the vicinity of the AV (cf. Annex C), node U4 *Trajectory Preprocessing* transforms the planned trajectories from GCS to the local navigation coordinate system East-North-Up (ENU) with origin in the planning position of the trajectory. Node U1 *Safe Halt* receives an implicit emergency trajectory described in the global coordinate system ETRS89 from node U3 *Strategical and Tactical Planning*. Based upon this, *Safe Halt* generates trajectories described in the local navigation coordinate system ETRS89 with origin in the planning position of the implicit emergency trajectory. For these trajectories to be followed, the quality of localization solution #2 must be sufficient according to the ODD and the current vehicle situation. If this is not the case, Node U1 *Safe Halt* cannot be used and the vehicle must be stopped immediately by a brake intervention.

In the UNICAR*agil* ADS, several separate sources of trajectories exist. Node U5 *Trajectory Selection* selects between these trajectory sources. The selection is based on the current vehicle operation mode (cf. Sec. 2.5.4). Subsequent nodes U7 and U8 are functionally identical to nodes B4 and B5 of the generic ADS with *Safe Halt* from Fig. 4-5. Node U9 *Self-Perception* has also been added to the UNICAR*agil* ADS. Its function is to determine evaluation results based on the vehicle capabilities and status of all (other) functions of the ADS.

[76] Woopen, T. et al.: UNICARagil - Disruptive Modular Architectures for AV (2018). a: p. 6; b: p. 5.

[77] Donges, E.: Three Level DDT Structure (1999).

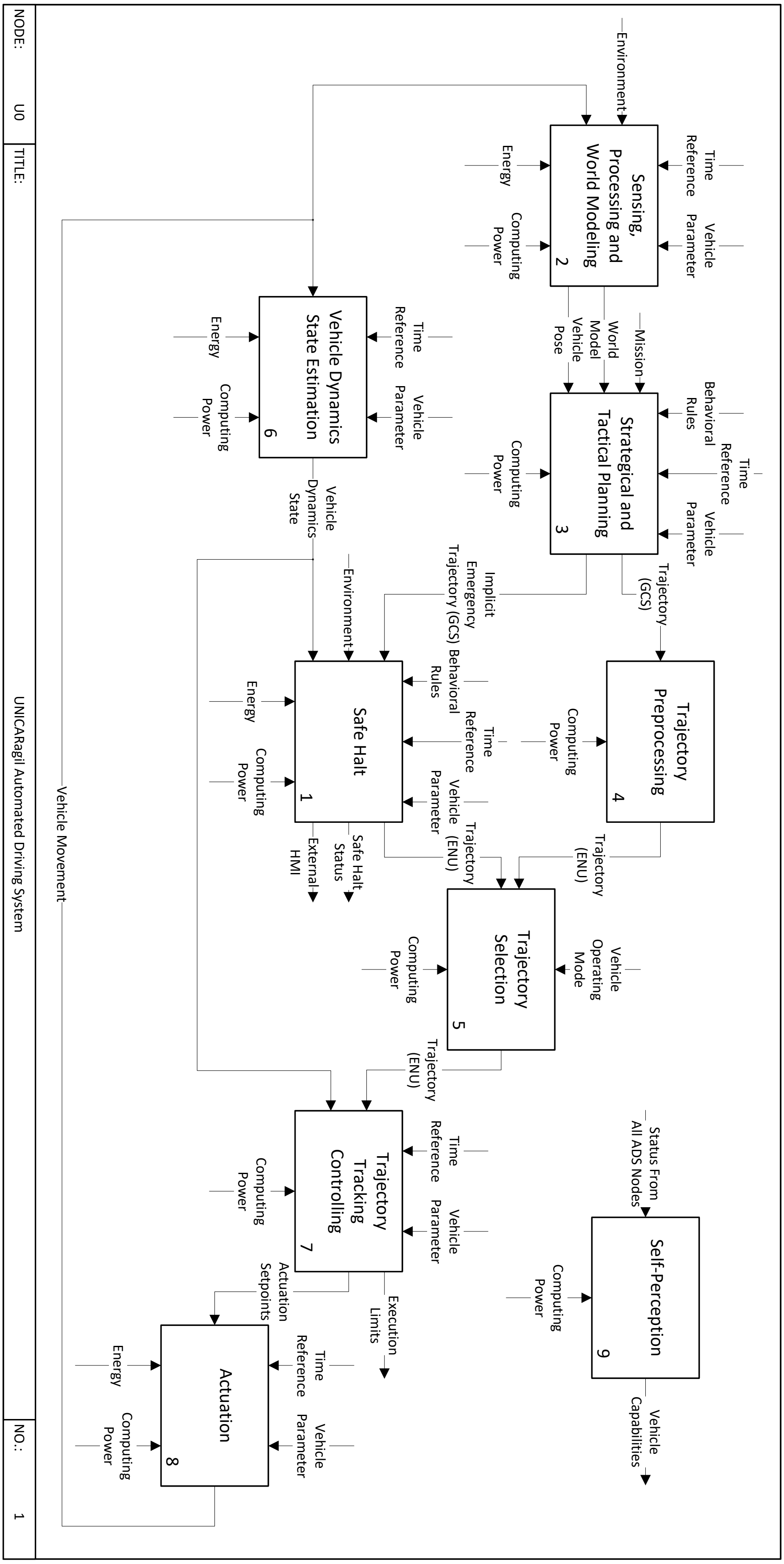


Figure 4-6: The Functional Architecture of the UNICAR*agil* ADS With Safe Halt in the IDEFØ Representation

### 4.2.3 Analysis of the UNICAR*agil* ADS Fault Tolerance Regime

This section describes the results of a fault tolerance regime analysis of the UNICAR*agil* ADS similar to the one described for a generic ADS (cf. Sec. 4.1.3). Also, the UNICAR*agil* ADS with nominal performance offers a set of missions with associated mission quality. Degradation of the UNICAR*agil* ADS can reduce the number of selectable missions and also affect the achievable mission quality. Thus, the results from Sec. 4.1.3 are also valid for the UNICAR*agil* ADS.

### 4.2.4 Fault Combinations of the UNICAR*agil* ADS

Distinguishable faults may occur in all nine functional nodes of Fig. 4-6. These distinguishable faults can then be used to generate the set of fault combinations of the UNICAR*agil* ADS. The same insights regarding the fault combinations of an ADS apply as in Sec. 4.1.4. Fault combinations can influence the capabilities of the UNICARagil ADS. In the UNICAR*agil* ADS the capabilities are monitored by a capability monitor[78a]. It receives information about the vehicles capabilities from all components of the ADS. The aggregation of all individual component capabilities thus generates the overall capability of the ADS. The outcome of this aggregation is compared to the capability requirements for the current driving mission or route segment[78b]. If the capabilities are insufficient for the safe execution of the driving mission, it shall be aborted, and the AV shall initialize the transition into a MRC.

### 4.2.5 Handling Fault Combinations of the UNICAR*agil* ADS

For the UNICAR*agil* ADS, handling fault combinations is done with one of three different strategies. The choice of strategy depends on the fault combination that has occurred and thus on the remaining capabilities of the ADS. Suppose the capabilities of the ADS are no longer sufficient to execute the driving mission safely. In that case, the driving mission shall be aborted, and the AV shall be transitioned to a MRC.

For the first fault handling strategy, the functional blocks for the vehicle operation mode *Automated Driving* abort the driving mission and transition the vehicle to a risk-minimal condition. This strategy is used if the capabilities of the functional blocks for the vehicle operation mode *Automated Driving* are in superior condition compared to the capabilities of the functional block *Safe Halt*. If, on the other hand, the capability evaluation result is reversed, the vehicle operation mode is changed to *Safe Halt*. Thus, the second strategy for fault handling of the UNICAR*agil* ADS is executed. If neither the capabilities of the functional blocks for the vehicle operation mode *Automated Driving* nor the capabilities of the functional block *Safe Halt* are sufficient to transition the AV to a MRC, the third strategy, *emergency stop*, is executed. Performing this strategy, the vehicle is immediately decelerated to a stop in place. As such, the addition of the

[78] Stolte, T. et al.: Towards Safety Concepts for Automated Vehicles (2020). a: p. 1576; b: p. 1578.

functional block *Safe Halt* discussed in this thesis increases robustness of keeping the AV in the safe state and provides the UNICAR*agil* ADS with an additional layer of automated fallback to reach a MRC, even in the presence of failures of functional blocks U2, U3, and U4 in Fig. 4-6. Assuming failures of these blocks, a maximum of fault combinations of these blocks is covered. The capabilities of *Safe Halt* are only selected to ensure maintaining the safe state and to transition the AV into the MRC. *Safe Halt* is not tasked with successfully finishing the original driving mission. Conversely, this means that *Safe Halt* requires a fail-operational property of blocks U5, U6, U7, U8 and U9. Without this fail-operational property, the functions of *Safe Halt* cannot be executed.

# 5 Safe Halt

This chapter establishes the system boundaries, interfaces, and requirements for a *Safe Halt* reference solution. Sec. 5.1 describes the methodology for finding a reference solution. This is followed by Sec. 5.2 with generic requirements for the reference solution. It is characterized by the fact that the requirements are formulated regardless of the specific ADS application. Subsequently, in Sec. 5.3, a generic functional architecture for *Safe Halt* in a generic ADS is presented. It includes all functional blocks to fulfill the functions of the *Safe Halt* node. In Sec. 5.4, the specific requirements for a Safe Halt solution in the UNICAR*agil* ADS are detailed. Finally, the chapter is summarized in Sec. 5.5. As part of the writing of this thesis, requirements were methodically derived and supplemented by requirements identified during the prototypical derivation of a reference solution. Nevertheless, it is not conclusively validated that the requirements are exhaustive.

## 5.1 Solution Determination Methodology

The methodology shown in Fig. 5-1 is followed for determining the reference solution. In the beginning, an analysis of the *Safe Halt* system boundaries is performed. For this purpose, the superordinate and neighboring systems are analyzed. Next, the overall system requirements for *Safe Halt* are determined. This determination is conducted for a generic ADS first and then extended for the UNICAR*agil* ADS. Following this, the requirements for the sub-functions of *Safe Halt* are elaborated. This is also performed for a generic and the specific UNICAR*agil* ADS. Based on these requirements, the concepts for the sub-functions of *Safe Halt* for the UNICAR*agil* ADS are presented. Finally, a synthesis of these concepts to the overall system concept for *Safe Halt* in the UNICAR*agil* context is performed. This concept is then implemented, integrated into the verification environments, and finally verified with the vehicle prototypes.

## 5.2 Requirements of a Safe Halt Reference Solution in a Generic ADS

This section describes the functional requirements for Safe Halt in a generic ADS. These requirements describe which functions a *Safe Halt* shall be able to perform. The generic ADS represented on Fig. 4-5 has the node label B0. *Safe Halt* in turn has node number 1. According to the IDEFØ representation, the node *Safe Halt* has node label B1. The figure shows the function's

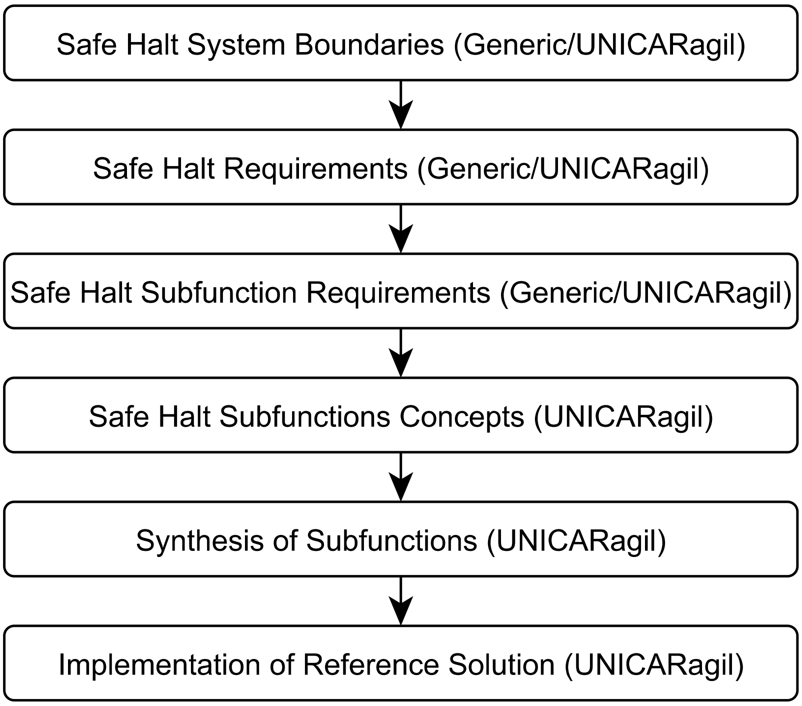

Safe Halt System Boundaries (Generic/UNICARagil)
Safe Halt Requirements (Generic/UNICARagil)
Safe Halt Subfunction Requirements (Generic/UNICARagil)
Safe Halt Subfunctions Concepts (UNICARagil)
Synthesis of Subfunctions (UNICARagil)
Implementation of Reference Solution (UNICARagil)

Figure 5-1: Safe Halt Reference Solution Determination Methodology

inputs, outputs, controls, and resources. This representation clarifies the system boundaries and thus provides the fundamentals for deriving the functional requirements. The labels of the requirements are derived from the node name. Applying the naming convention, they start with the abbreviation **Req.**, followed by the node label **B1** and an incrementing number. First, the requirements for the output of the function are defined. The input requirements needed for the generation of the output follow afterwards.

*Safe Halt* has the task of generating trajectories that can be processed by subsequent trajectory tracking controlling resulting in actuation setpoints for the vehicle actuation. Functionally, this results in the requirement:

**Req. B1-1:** Safe Halt shall generate trajectories

As a second output, *Safe Halt* shall provide its internal status to a neighboring or superordinate system. This information is used to monitor the overall ADS capabilities. To monitor the internal status of the *Safe Halt*, all sub-functions shall determine their status. These individual states are then aggregated and output for the overall state of the function. In particular, all components that initiate a response to the driving environment shall be monitored. What follows is the first requirement:

**Req. B1-2:** Safe Halt shall detect insufficiencies that lead to the loss of guaranteed performance of its functionality

*Safe Halt* receives an implicit emergency trajectory as input from B3 *Strategical and Tactical Planning*. It originates in the vehicle's state at the time of planning and ends in a Minimal Risk Condition (MRC). *Safe Halt* uses this implicit emergency trajectory to generate trajectories. Those shall stay within the spatial course of the pre-planned implicit emergency trajectory. This results in the requirement:

**Req. B1-3:** The trajectories generated by Safe Halt shall track the spatial course of the most recent implicit emergency trajectory. The tolerance is To Be Determined (TBD).

At the implicit emergency trajectory input, a maximum velocity profile is received from B3 *Strategical and Tactical Planning*. This velocity profile originates in the vehicle state and ends in the MRC. Derived from this is the requirement:

**Req. B1-4:** The trajectories generated by Safe Halt shall not exceed the velocities of the maximum velocity profile

*Safe Halt* receives behavioral rules as a control. This control defines constraints for the generation of trajectories. Part of these rules are based on road traffic law, for example, to ensure the legal time gaps between the Automated Vehicle (AV) and other traffic participants when generating emergency trajectories. However, to simplify the representation, the behavioral rules also include the execution constraints of AV, which are output by node B4 *Trajectory Tracking Controlling*. This leads to the requirement:

**Req. B1-5:** The trajectories generated by Safe Halt shall comply with all limits of the behavioral rules (legal, kinematic, and dynamic)

The function B3 *Strategical and Tactical Planning* transmits the pre-planned implicit emergency trajectory to *Safe Halt*. Before transmission, this data is compressed, the implicit emergency trajectories are sampled according to their specific accuracy requirements, and the samples are then transmitted to *Safe Halt*. *Safe Halt*, on the other hand, shall be able to reconstruct this compressed data without losses. This results in the requirement:

**Req. B1-6:** Safe Halt shall reconstruct the compressed input data without losses

To cope with a maximum set of fault combinations of an ADS, the *Safe Halt* function is designed without access to B2 *Sensing, Processing and World Modeling* function. *Safe Halt's* environment perception capabilities are designed to only safeguard the Minimal Risk Maneuver (MRM) while tracking the spatial course of the implicit emergency trajectory. For this reason, *Safe Halt* obtains the environment as input. The generated trajectories shall result in collision-avoiding vehicle behavior. However, the response to the driving environment may only include adjusting the vehicle speed. Spatial deviation from the pre-planned implicit emergency trajectory is not permissible. This leads to the requirement:

**Req. B1-7:** Safe Halt shall respond to the environment to avoid collisions within the planned implicit emergency trajectory

In order to be able to react to changes in the environment, information about it shall be captured by suitable sensors. Since *Safe Halt* is independent of B2 *Sensing, Processing and World Modeling* of the ADS, a separate environment perception system shall be integrated into the ADS. The data provided by the complementary environment perception system shall be processed for collision checking. This results in the requirement:

**Req. B1-8:** Safe Halt shall capture relevant information about the environment with its own environment perception system

The generic ADS with *Safe Halt* is functionally represented on Fig. 4-5. In addition to the functional architecture, other architecture views on the ADS are relevant. Exemplary architecture views are software architectures, data transmission architectures, power supply architectures, and Electronic Control Unit (ECU) architectures. A solution for the *Safe Halt* concept shall satisfy all relevant architectural views and be able to be integrated into them and results in the next requirement:

**Req. B1-9:** Safe Halt shall be integratable into all relevant architectures of the ADS

*Safe Halt* performs a MRM. Neighboring and superordinate systems can monitor the vehicle's state and thus determine whether the vehicle has reached a standstill. However, these systems cannot determine whether the vehicle has reached the MRC or whether *Safe Halt* has stopped the vehicle behind a static object on the implicit emergency trajectory. *Safe Halt* shall inform about the termination of the risk-minimal maneuver. Thus arises the requirement:

**Req. B1-10:** Safe Halt shall inform neighboring and superimposed systems about the termination of the minimal risk maneuver

*Safe Halt* includes the functions *Safe Halt Reconstruction*, *Safe Halt Sensing, Processing and World Modeling*, *Safe Halt Strategical and Tactical Planning* and *Safe Halt Self-Perception*. These functions shall not be executed continuously but can be executed event-based. Before the emergency trajectory can be generated, the strategical and tactical behavior of the vehicle shall be planned. This tactical planning is based on the most recent implicit emergency trajectory and environment data. As soon as either a new implicit emergency trajectory or new environment data is available, a new trajectory planning shall be started. Therefore, the strategical and tactical planning and trajectory generation shall be performed for each input data change. Functions can be started by different triggers. It is sufficient to execute the functions if one of the mentioned triggers applies, meaning all triggers are combined via a logical 'or' operation. This leads to the requirement:

**Req. B1-11:** The Safe Halt functions shall be executed event-based according to the triggers from Tab. 5-1

Table 5-1: Safe Halt Subfunction Triggers

| Function | Trigger 1 | Trigger 2 |
|---|---|---|
| Safe Halt Reconstruction | New Implicit Emergency Trajectory | |
| Safe Halt Sensing, Processing and World Modeling | New Environment Data | |
| Safe Halt Strategical and Tactical Planning | New World Model | New Reconstructed Implicit Emergency Trajectory |
| Safe Halt Self-Perception | Timeout | |

*Safe Halt* is an independent system in the ADS and should therefore be able to be tested independently. For reusability of functions, *Safe Halt* shall be modular in design. It is still part of the safety functions of the ADS. Thus, emphasis is placed on the testability of the solution in order to safeguard its functions. For this, tests shall take place in Software in the Loop (SiL), Hardware in the Loop (HiL) and real test vehicles. A solution for *Safe Halt* shall be testable for each scenario. Furthermore, different approaches to software architectures are available. Robot Operating System (ROS) 2[79] is, besides Automotive Service-Oriented Software Architecture (ASOA), a middleware candidate for the automotive industry. Several vehicle manufacturers are working on their middleware for their automotive software functions. These middlewares are also applied to automated driving functions. Volkswagen is working on the operating system VW.OS[80] based on the Android Open Source Project (AOSP). Mercedes Benz is working on the MB.OS[81]. Tesla has its software stack[82]. Suppliers like Continental, Bosch, and ZF are currently working on middleware. In addition, there are proven standards such as AUTOSAR[83]. In summary, none of the previously mentioned middleware has yet stood out across industries. Therefore, a *Safe Halt* solution shall be independent of any given middleware and operating system. In addition, various hardware is being developed to realize automotive functions. This includes embedded hardware using lean operating systems like Free-RTOS[84] or Petalinux[85] as well as computers running full Linux or Windows operating systems. Their independence from the operational environment is necessary for the reusability of the *Safe Halt* functions. From these follow the additional requirements:

[79] Open Robotics: ROS 2 Documentation (2022).

[80] GoogleWatchBlog-Team: VW.OS (2019).

[81] jesmb.de: MB.OS (2021).

[82] Tesla: Full Self-Driving Computer Installations (2020).

[83] AUTOSAR: AUTOSAR (2022).

[84] Amazon Web Services: FreeRTOS (2022).

[85] Xilinx: PetaLinux Tools (2022).

**Req. B1-12:** Safe Halt shall be an independently deployable and testable module

and

**Req. B1-13:** The Safe Halt solution shall provide generalized input and output interfaces to the operational environments.

A MRM is executed when *Safe Halt* takes control over the vehicle. Other road users in the vehicle's vicinity shall immediately be informed about the emergency situation. Safe Halt shall therefore activate all existing external Human Machine Interface (HMI) elements that raise awareness.

**Req. B1-14:** Safe Halt shall activate all available external HMI to increase awareness and to inform other road users about the emergency situation of the vehicle

## 5.3 Functional Architecture for a Safe Halt Reference Solution in a Generic ADS

Having identified the requirements for executing the sub-functions, this subsection presents a generic functional architecture for *Safe Halt*. This generic functional architecture is the basis for the following more detailed definition of requirements for the sub-functions. It combines all requirements into a generic solution. At the beginning of the concept development, the interfaces of the Safe Halt solution are examined. The implicit emergency trajectories with maximum velocity profile are transmitted to the solution via its input. Req. B1- 11 describes the triggers for the functions of *Safe Halt*. Since the triggers depicted in Tab. 5-1 should only initiate their associated functions, they are implemented as sub-functions of *Safe Halt*. This division into sub-functions ensures that each function can be triggered independently. A generic functional architecture for *Safe Halt* is shown on Fig. 5-2. The figure includes all functions of the generic ADS and shows their dependencies. The controls and mechanisms for functional execution are also visible. The node B11 *Safe Halt Reconstruction* reconstructs the compressed data which contains the implicit emergency trajectory. The function receives computing powers and a time reference for their computations. The reconstructed implicit emergency trajectory and the status of the function are outputs. Node B12 *Safe Halt Sensing, Processing, and World Modeling* receives the vehicle environment as input. As mechanisms, the functions are provided with energy for the environment perception system and computing power for their computations. As controls, the function is supplied with a time reference and vehicle parameters. For example, the vehicle parameters describe the position and orientation of the environment perception system within the AV. As the output of the function, a world model of the vehicle environment and the status of the function is provided. Node B13 *Safe Halt Strategical and Tactical Planning*

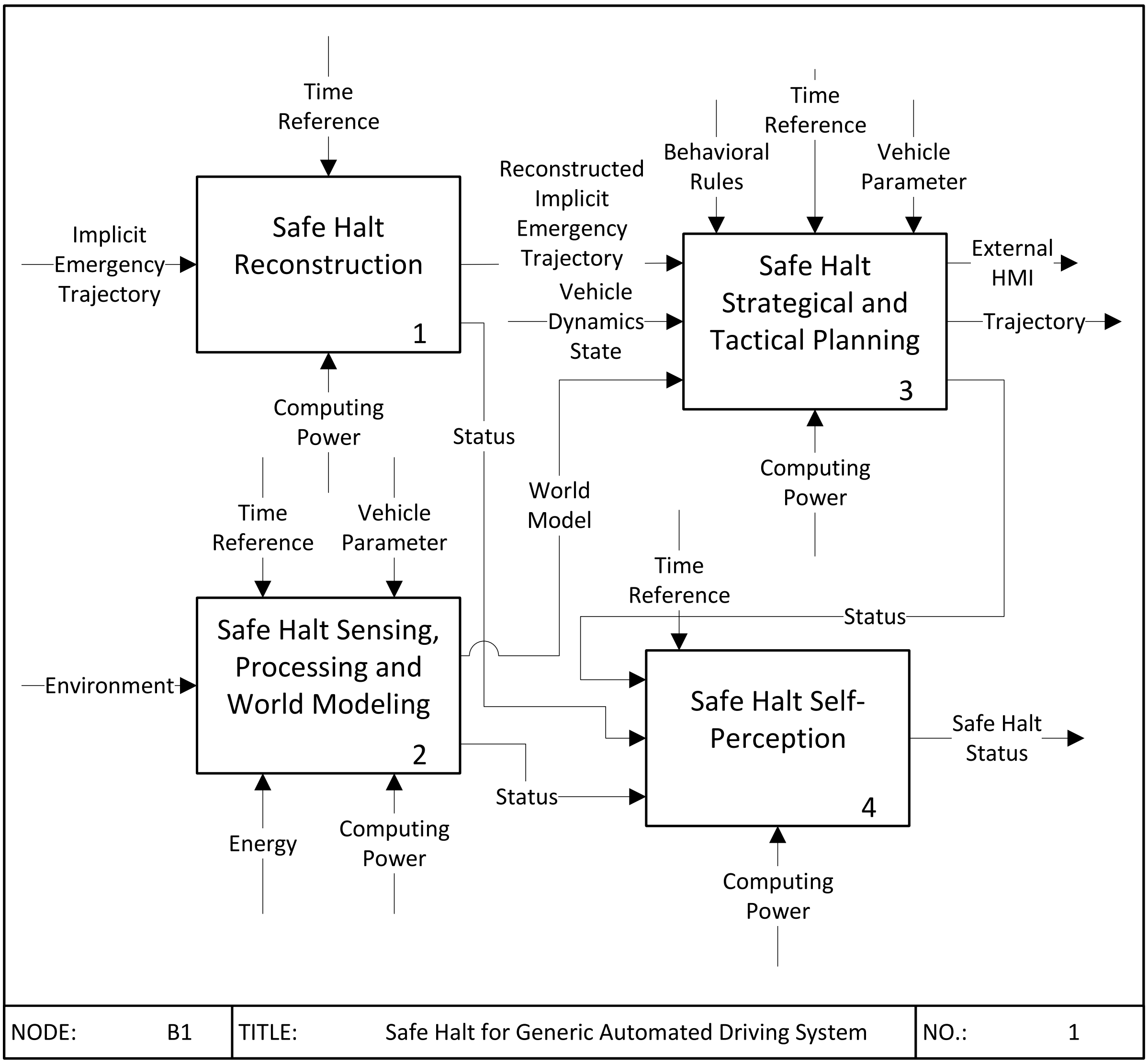


Figure 5-2: Functional Architecture for a Safe Halt Reference Solution in a Generic ADS

handles the planning and the generation of the trajectory. For this purpose, the reconstructed implicit emergency trajectory calculated by node B11, the world model generated by node B12, and the vehicle dynamics state are fed to the function at the input. The output is generated using computing powers as the mechanism and behavioral rules, time references, and vehicle parameters as controls. It consists of the generated trajectory and the status of the function. Node B14 *Safe Halt Self-Perception* finally receives the states of all functional blocks. Consuming computing power and a time reference, the overall status of *Safe Halt* is calculated and output. Since all sub-functions act asynchronously, each function is not triggered by events, but rather scheduled in set time intervals. For a reference solution to satisfy Req. B1- 12, the functions are implemented as tasks (cf. Annex D) and separated into one core and multiple interfaces (cf. Annex E).

## 5.4 Requirements of a Safe Halt Reference Solution in the UNICAR*agil* ADS

This subsection describes the specific requirements for a *Safe Halt* in the UNICAR*agil* ADS. All requirements of a *Safe Halt* for a generic ADS apply for the specific application to the UNICAR*agil* ADS as well. Therefore, a fusion of the generic with the specific requirements listed below shall be performed. The fundamental basis for these requirement derivations is the UNICAR*agil* ADS from Fig. 4-6. It has the node label U0. *Safe Halt* in turn has node number 1. According to the IDEFØ Representation, the node *Safe Halt* therefore has node label U1. The labels of the requirements are derived from this. It starts with the abbreviation **Req.**, followed by the node label **U1** and an incrementing number.

*Safe Halt* in the UNICAR*agil* ADS has the task to output trajectories, which the following functions can use. The UNICAR*agil* ADS relies on the ASOA (cf. Sec. 2.5.3). With this software architecture, different software services can be connected with each other at execution time. *Safe Halt* aims to maintain the safe state in cases of potential failures of nodes U2 *Sensing, Processing and World Modeling* and U3 *Strategic and Tactical Planning* and to be able to execute a MRM. It is assumed that nodes U5, U6, U7, and U8 are fail-operational.

Node U7 *Trajectory Tracking Controller* expects a trajectory in a specified format as one of its input. Node U3 *Strategic and Tactical Planning* plans the trajectory for the vehicle operation mode *Automated Driving* with a frequency of at least 10 Hz. This frequency is also adopted for the trajectory generation of *Safe Halt*. This results in the requirement:

**Req. U1-1:** Safe Halt shall generate trajectories in the required format (cf. Annex B) with frequency at least 10 Hz. The tolerance is TBD

*Safe Halt* provides functions as a service in the service-based software architecture. The UNICAR*agil* ADS includes U9 *Self-Perception* that monitors the vehicle's capabilities. Input to *Self-Perception* is the status information of each service in the software architecture. This includes the *Safe Halt* service. It shall communicate its capabilities. According to modular software architecture, *Safe Halt* shall not know about the current vehicle operation mode. This means that the health status of *Safe Halt* cannot be determined solely based on internal service information. For example, suppose no new implicit emergency trajectories are received over an extended time. In that case, this may be either because the implicit emergency trajectory planner or the communication architecture has failed, or it may be that the current vehicle operation mode does not include sending implicit emergency trajectories to *Safe Halt*. It can, therefore, neither assert that it is not operational nor that it is. *Safe Halt* therefore transmits unbiased information to *Self-Perception*, such as the timestamps for the last successful reception of input or execution of a function. *Self-Perception* can then use the status of all services to determine the overall vehicle capability. As frequency for the status transmission to the *Self-Perception* at least 10 Hz is chosen.

This frequency is arbitrarily chosen and shall be adjusted according to research findings regarding *Self-Perception*. Safe Halt is, therefore subject to the requirement:

**Req. U1-2:** Safe Halt shall provide unbiased status information to the vehicle Self-Perception system at a frequency of at least 10 Hz. The tolerance is TBD

Different architecture views are considered to implement the functional architecture of the UNICAR*agil* ADS. A solution for the *Safe Halt* concept shall satisfy all relevant architectural views and allow for system integration into all of them. The requirements concerning the architectural views are explained below.

### ECU Architecture

The ECU architecture[86] describes the use of the ECUs, which provide computing power for the execution of the vehicle's functions. It shall be possible to integrate a reference solution for *Safe Halt* into the ECU architecture. *Safe Halt* calculations shall be executable on at least one of the ECUs. Safe Halt shall not exceed the computational resources of the hardware used. According to the control unit architecture of UNICAR*agil* with cerebrum, brainstem, and spinal cord, *Safe Halt* is assigned to the brainstem level. This results in the following requirement:

**Req. U1-3:** Safe Halt computations shall be executable on at least one brainstem level ECU

### Communication Architecture

The communication architecture of the UNICAR*agil* ADS is mainly based on Ethernet communication. Four ethernet switches are connected as a ring. Redundant communication between the ethernet switches is realized via this ring architecture[87]. The *Safe Halt* functions shall be able to communicate with relevant components of the ADS via ethernet communication. In particular, the ECU for performing the *Safe Halt* functions shall have an ethernet interface. In addition, the *Safe Halt* shall implement interfaces to enable communication via the ethernet connection. Both User Datagram Protocol (UDP) and Transmission Control Protocol (TCP) are used as ethernet protocols in the communication architecture and should be supported by the *Safe Halt* functions. Therefore the requirement is:

**Req. U1-4:** Safe Halt shall communicate with relevant ADS components via ethernet communication with UDP and TCP

[86] Keilhoff, D. et al.: UNICARagil New architectures vehicle concepts (2019).

[87] Niedballa, D.; Reuss, H.-C.: Concepts of functional safety in E/E-architectures (2020), p. 9.

### Power Supply Architecture

In the UNICAR*agil* vehicles, 48 V power supplies are used for the dynamics modules[88] and 12 V for the other consumers. The ECU and the environment perception system for *Safe Halt* shall be compatible with one of these power supplies. This results in the requirement:

**Req. U1-5:** Safe Halt components shall be compatible with 48 V or 12 V power supply

### ASOA

The *Safe Halt* solution shall be able to interact with ASOA (cf. Sec. 2.5.3) via its inputs and outputs so that *Safe Halt* in ADS can receive the input data and send the output data. The following requirements thus arise for the *Safe Halt* solution as a service in the ASOA:

**Req. U1-6:** Safe Halt shall offer its functions as a service in the ASOA

**Req. U1-7:** Safe Halt shall be able to communicate with relevant other services via requirements and guarantees

This requirement includes user data, quality data, and parameter data.

**Req. U1-8:** Safe Halt shall be able to switch its life cycle between stopped and started

The ASOA can execute functions on shared ECU. Accordingly, each service shall use the ECU resources in a resource-conserving manner. In stopped mode, the service is not needed in the active vehicle operation mode, so the service's resource usage shall be minimal. This leads to the requirement:

**Req. U1-9:** In the stopped state, the resource usage of the Safe Halt service shall be minimal

One of the basic ideas of ASOA is that the vehicle operation modes are only generated by orchestrating the services. The services should not be able to access information about the currently active vehicle operation mode. This creates the requirement:

**Req. U1-10:** The functions of the Safe Halt service shall be provided independently of the knowledge of the current vehicle operating mode

A vehicle operating mode management system is integrated into the ASOA. It monitors the current operating mode and monitors all requirements for potential vehicle operating mode changes. If an operating mode change is necessary, the operating mode management can trigger the mode

[88] Woopen, T. et al.: UNICARagil - Disruptive Modular Architectures for AV (2018), p. 3.

change. *Safe Halt* is an operating mode of the vehicle. After the MRM has been performed, the operating mode management shall be informed about the termination of the maneuver. The operation mode management, in turn, can use this information to change the vehicle operation mode to *Teleoperation* or back to *Automated Driving*. For the *Safe Halt* functions, the following therefore applies:

**Req. U1-11:** Safe Halt shall inform the ASOA vehicle operation mode management when the minimum risk maneuver has been terminated

### Time Reference Architecture

The UNICAR*agil* ADS uses Temps Atomique International (International Atomic Time) (TAI) as a time reference. *Safe Halt* shall therefore have access to a corresponding clock in the time reference architecture:

**Req. U1-12:** Safe Halt shall have access to a TAI clock and shall be able to process this time

In contrast, strictly monotonically increasing clocks are used for time measurements, such as timeouts. Therefore the requirement:

**Req. U1-13:** Safe Halt shall have access to a monotonically increasing clock and shall be able to process this time

The potential applications of *Safe Halt* for the UNICAR*agil* ADS are ASOA, SiL (IPG CarMaker™ [89]), HiL with real ECU and the four UNICAR*agil* prototype vehicles. Hence the specific requirement:

**Req. U1-14:** Safe Halt for the UNICARagil ADS shall be applicable for testability in ASOA, SiL, HiL and the real UNICARagil prototype vehicles

External HMI components have been integrated into the UNICAR*agil* vehicles. This includes direction indicators and Light Emitting Diode (LED) matrices for displaying texts or symbols. These components enhance the vehicle's visibility, especially during a MRM. When the vehicle operation mode *Safe Halt* is activated, the system should also activate this external HMI. Since the HMI components have an ASOA interface, the activation is done via ASOA. This results in the requirement:

**Req. U1-15:** Safe Halt for the UNICARagil ADS shall activate the hazard warning lights and the LED matrices with an ASOA interface

[89] IPG Automotive: CarMaker (2022).

## 5.5 Conclusion

This chapter presents the methodology for establishing the *Safe Halt* reference solution. Following the methodology, requirements for a *Safe Halt* in a generic ADS are first derived, followed by the preparation of a concept and a generic functional architecture. Subsequently, the UNICAR*agil* ADS is used as an application to derive specific requirements for a *Safe Halt* reference solution. As a result of this chapter, the basic generic requirements for a *Safe Halt* are analyzed, and a generic functional architecture is elaborated. This functional architecture is picked up in the following chapters, and the detailed requirements for the sub-functions of *Safe Halt* are determined. This is conducted for a generic ADS, as well as for the UNICAR*agil* ADS. Due to the lack of validity of the derivation of the requirements, it is assumed that the requirements presented are not exhaustive and shall be expanded during a development phase with subsequent validation.

# 6 Neighboring and Superordinate Systems of Safe Halt

After deriving the generic and specific requirements for the *Safe Halt* solution in the previous chapter, this chapter describes the requirements for the neighboring and superordinate systems of *Safe Halt* within the Automated Driving System (ADS). These systems shall meet certain requirements for *Safe Halt* to perform its function. This chapter starts with a description of generic requirements in Sec. 6.1. The following chapter Sec. 6.2 highlights the effects of integrating *Safe Halt* into an ADS. Subsequently, requirements specific to the UNICAR*agil* ADS are added in Sec. 6.3. The concepts for the UNICAR*agil* ADS are presented in Sec. 6.4. Finally, the chapter is summarized in Sec. 6.5.

## 6.1 Requirements for Neighboring and Superordinate Systems in a Generic ADS

*Safe Halt* obtains both the current vehicle state and the implicit emergency trajectory via the inputs. In the generic ADS shown in Fig. 4-5, node B2 *Sensing, Processing and World Modeling* provides the vehicle state. Node B3 *Strategical and Tactical Planning* provides the implicit emergency trajectory. A third function is the node B6 *ADS Monitoring*, which monitors the status of all ADS components and activates *Safe Halt* if necessary. Node B4 *Trajectory Tracking Controlling* provides the execution limits of the vehicle. These are used to consider the kinematic and dynamic constraints of the vehicle for trajectory planning. The requirements for these functions are presented in the remainder of this section.

### 6.1.1 B2: Sensing, Processing and World Modeling

First, the requirements of node B2 *Sensing, Processing, and World Modeling* are described. According to the IDEFØ representation, the node has node label B2. The labels of the requirements are derived from this. It starts with the abbreviation **Req.**, followed by the node label **B2** and an incrementing number.

Safe Halt uses its own environment perception system to capture the vehicle's surroundings. The world model is used to detect collision objects on the course of the implicit emergency trajectory. The course of this implicit emergency trajectory is described in a specific coordinate system. The environment perception sensors are fixed to the vehicle. Thus, their poses with respect to a vehicle-fixed reference point are available. However, the vehicle's pose at the time of the environment perception is unknown. Transforming the vehicle reference point into the same coordinate system as the implicit emergency trajectory is needed. For this reason, the following requirement arises:

**Req. B2-1:** Sensing, Processing, and World Modeling shall provide the vehicle pose

### 6.1.2 B3: Strategical and Tactical Planning

The function *Strategical and Tactical Planning* depicted on Fig. 4-6 has node number 3. According to the IDEFØ Representation, the node has node label B3. The labels of the requirements are derived from this. It starts with the abbreviation **Req.**, followed by the node label **B3** and an incrementing number.

The implicit emergency trajectory is one input of *Safe Halt*. A generic ADS has access to various sources of information, such as its environment perception system or even digital maps. These components' capabilities are designed so that the ADS can perform driving missions. *Safe Halt*, on the other hand, has only the capabilities to monitor the implicit emergency trajectory for collision objects. It cannot plan the Minimal Risk Maneuver (MRM). However, the pose course and the maximal velocity profile can be planned before the start of the MRM. Since the capabilities of the *Safe Halt* should be as minimal as possible to keep complexity low and maintain a certain level of robustness, the planning of the implicit emergency trajectory remains with the *Strategical and Tactical Planning* (cf. Fig. 4-6) of the ADS. *Safe Halt* then only needs to have the capabilities to detect possible collision objects on the implicit emergency trajectory and to adjust the velocity to track this trajectory in a collision-free manner.

The pre-planned implicit emergency trajectory can also be described as an explicit emergency trajectory. An explicit trajectory describes the intended behavior spatiotemporally. The temporal information of a pre-planned explicit emergency trajectory becomes invalid as soon as the *Safe Halt* function has to deviate from this pre-planned velocity profile. One reason for this is, for example, the reaction to an impassable object within the spatial course of the emergency trajectory. All future global timestamps of the explicit emergency trajectory elements thus become invalid. A consequence of this is that time becomes an additional reference and can no longer be considered fixed during the emergency trajectory execution. Spatial conditions that require an adjustment of the vehicle's speed are thus no longer identifiable. For example, a spatial speed adjustment may be required for passing speed bumps or curbs. Therefore, the reference for the intended behavior limit should be a parameter that does not change during the execution of the emergency trajectory. A comparison of an implicit and an explicit emergency trajectory can be found in Tab. 6-1. Due to the outlined advantages, an implicit emergency trajectory is chosen instead of an explicit emergency trajectory. For the function B3 *Strategical and Tactical Planning*, this results in the requirement:

Table 6-1: Comparison Between Implicit and Explicit Emergency Trajectory

| | Explicit Emergency Trajectory | Implicit Emergency Trajectory |
|---|---|---|
| Reference | Time $t$ in s | Trajectory length $s$ in m |
| Advantage | | Location-dependent velocity adaption (curvature, speed bumps). A path with velocity profile is an implicit trajectory |
| Disadvantage | Reference becomes invalid when an object reaction is executed | |

**Req. B3-1:** Strategical and Tactical Planning shall plan implicit emergency trajectories originating in the vehicle state

*Safe Halt* enables the performance of a MRM by executing and monitoring the implicit emergency trajectory. A local Minimal Risk Condition (MRC) can be achieved anytime by stopping the vehicle. Furthermore, a global MRC can be achieved by *Safe Halt*. This state is characterized by the fact that it is a situation-dependent global MRC. The overall risk is composed of the risk due to the condition itself and the MRM to that condition. Planning of the implicit emergency trajectory shall therefore select the global MRC within the uncertainty tolerance considering the capabilities of the *Safe Halt* and plan the implicit emergency trajectory to that condition. State of the art does not yet reflect any statements about criteria of global minimal-risk stopping locations and conditions. However, initial scientific studies have identified criteria[90] for evaluating the safety of MRC. For the planning, the requirement applies:

**Req. B3-2:** Strategical and Tactical Planning shall plan implicit emergency trajectories terminating in the situation-depended global minimal risk condition within the uncertainty tolerance

For reasons of limited communication capacity, the planned implicit emergency trajectories shall be compressed before transmitted to *Safe Halt*. This compression includes the poses and velocity profile of the implicit emergency trajectory. *Safe Halt* at the receiving end shall be able to reconstruct this compressed data without loss of information. For the planning of the implicit emergency trajectory, the following requirement arises:

**Req. B3-3:** Strategical and Tactical Planning shall compress the planned implicit trajectory in a way that Safe Halt can losslessly reconstruct the data

The *Safe Halt* function performs a MRM. The velocity during this maneuver should be as fast as safely possible to avoid surprising the following traffic with a sudden low velocity, which can

[90] Hoppen, F.: Methodik zur Identifikation von sicheren Orten für Nothaltemanöver (2022).

be considered an unreasonable risk. On the other hand, the speed shall be slow enough for *Safe Halt* to monitor the implicit emergency trajectory monitoring for collision objects. During the speed reduction, the own lane must not be left so that the other road users can be warned about the emergency situation. The implications to the planning of the implicit emergency trajectory result in the following requirements:

**Req. B3-4:** Strategical and Tactical Planning shall plan implicit emergency trajectories with a strong initial deceleration (not greater than $4\,\mathrm{m\,s^{-2}}$ [91]) down to a creep speed. With this creep speed, the rest of the emergency trajectory is planned. At the end of the emergency trajectory, the vehicle should be stopped comfortably. The first part of the implicit emergency trajectory shall be within its own lane to give surrounding objects time to adjust to the degraded automated vehicle.

and

**Req. B3-5:** Strategical and Tactical Planning shall plan implicit emergency trajectories with a creep speed to balance the need to perform the minimal risk maneuver quickly and the capabilities of the Safe Halt environment perception to monitor the course of the implicit emergency path. This creep velocity is still To Be Determined (TBD).

To monitor the implicit emergency trajectory for objects, the implicit emergency trajectory shall be within the field of view of the environment perception system of *Safe Halt*. To account for this, the spatial course of the implicit emergency trajectory shall be considered and combined with the yaw angle of the implicit emergency trajectory. The quality of the object states information decreases towards the edge of the sensor field of view. Therefore, the implicit emergency trajectory should be located centrally in the sensor field of view, leading to:

**Req. B3-6:** Strategical and Tactical Planning shall plan implicit emergency trajectories that, by best effort, lie within the field of view of the environment perception of Safe Halt

The vehicle environment perceived at the planning time shall be used to plan the implicit emergency trajectory. The implicit emergency trajectory is planned to avoid short-term collisions with other objects. The states of the relevant objects are included in planning the local maximum speed profile. Thus, the requirement for the planner of an implicit emergency trajectory is as follows:

**Req. B3-7:** Strategical and Tactical Planning shall plan the implicit emergency trajectory with regard to the states and prediction states of all relevant objects at planning time

Finally, the planned implicit emergency trajectories shall be physically feasible. Therefore, the planning shall consider the kinematic limits of the Automated Vehicle (AV). Trajectories shall

[91] European Parliament and the Council: EU Regulation - Type-approval of ADS (2022), p. 10.

only be planned that can be realized due to the kinematic capabilities of the AV. Due to the capabilities of the actuator system, the area for a feasible center of rotation is limited. Furthermore, the planning shall respect the dynamic limits of the AV. These include the acceleration limits, the jerk limits, and the steering angle velocity limits. This leads to the requirement:

**Req. B3-8:** The planning of the implicit emergency trajectory shall comply with the kinematic (limited center of rotation area) and dynamic (limited acceleration, jerk, and steering angle velocity) limits of the AV

### 6.1.3 B6: ADS Monitoring

A vehicle component shall activate *Safe Halt* if the driving mission cannot be continued and the MRC must be reached. *Safe Halt* is permanently active but does not independently intervene in the effect chain of ADS. Therefore, another ADS component shall initiate the activation of *Safe Halt*. This leads for the function of node B6 *ADS Monitoring* to the requirement:

**Req. B6-1:** ADS Monitoring shall monitor the capabilities of the vehicle and activate Safe Halt in the effect chain when needed

### 6.1.4 B4: Trajectory Tracking Controlling

Finally, *Safe Halt* requires constraints to ensure the generation of physically feasible trajectories. Therefore, Req. B3- 7 is also valid for *Safe Halt*. The execution limits include the kinematic and dynamic limits of the AV. These limits shall be determined by the trajectory tracking controlling and transmitted to *Safe Halt*. This leads to the requirement:

**Req. B4-1:** Trajectory Tracking Controlling shall provide the kinematic and dynamic execution limits of the AV for generating the Safe Halt trajectory

## 6.2 Effects of Safe Halt on the ADS

This section examines the impact of a *Safe Halt* solution on the ADS shown in Fig. 4-5. Its integration has an impact on the original ADS. All adverse effects of *Safe Halt* to the original ADS shall be detected and safeguarded. *Safe Halt* is a source of trajectories. In *Automated Driving* vehicle operation mode, trajectories from node B3 *Strategical and Tactical Planning* are continuously sent to node B4 *Trajectory Tracking Controlling*. The source of the trajectory is, therefore, always the same. If B3 *Strategical and Tactical Planning* plans its trajectories based on the last planned trajectories, the trajectories can be assumed to have a constant iterative behavior.

The situation is different if *Safe Halt* is introduced as a second trajectory source. *Safe Halt* plans trajectories independently from the *Strategical and Tactical Planning*. For this reason, it is not ensured that the *Safe Halt* trajectory output matches the current trajectory of B3 *Strategical and Tactical Planning*. Changing the trajectory source thus leads to a risk for the vehicle. For B4 *Trajectory Tracking Controlling*, the change of trajectory sources acts like a step function. It shall be ensured that this trajectory change causes no collisions and that the vehicle does not become unstable. Non-safety critical criteria should also be considered. For example, an unfavorable choice of the initial states of the emergency trajectory can lead to uncomfortable vehicle behavior, which can cause unfavorable reactions of the vehicle occupants or the load.

*Safe Halt* may share hardware, operating system, and communication infrastructure with other functions. *Safe Halt* computations are comparatively resource intensive compared to pure control tasks. *Safe Halt* thus consumes some of the overall available resources. It shall be ensured that the resources are sufficient for all functions sharing the same hardware and operating system. If this is not the case, using Safe Halt can lead to other functions no longer being able to run in real-time, thus creating unreasonable risks. *Safe Halt* relies on pre-planned implicit emergency trajectories. Therefore, a corresponding *Strategical and Tactical Planning* for those implicit emergency trajectories shall be introduced into the ADS. This planning consumes the resources of the vehicle. It shall be ensured that this additional resource requirement of the hardware, operating system, and communication infrastructure can be met. If this demand is not covered, it may lead to the fact that the trajectories for the *Automated Driving* vehicle operation mode can no longer be planned in real-time, which leads to an unreasonable risk. *Safe Halt* increases the complexity of the overall ADS. This increased complexity shall be tested accordingly. *Safe Halt* increases the testing effort by applying the new *Safe Halt* vehicle operating mode, as this change also means a dynamic change of the trajectory source.

After *Safe Halt* has reached the end of the pre-planned implicit emergency trajectory or the vehicle has come to a standstill during the performance of the MRM, an external entity shall take over the vehicle control. This could be a human operator in a remote control room. The operator can connect to the vehicle via a wireless connection, communicate with the occupants, and operate the vehicle via direct or indirect control. *Safe Halt*, therefore, requires the handover of vehicle responsibility to a human operator. There should be sufficient radio communication with the vehicle for the operator to intervene.

## 6.3 Specific Requirements for Neighboring and Superordinate Systems in the UNICAR*agil* ADS

This section discusses the specific requirements for neighboring and superordinate systems of *Safe Halt* in the UNICAR*agil* ADS. These systems are depicted on Fig. 4-6.

### 6.3.1 U6: Vehicle Dynamics State Estimation

There are two different localization functions in UNICAR*agil* ADS. The node U6 *Vehicle Dynamics State Estimation* shown in Fig. 4-6 is the fail-operational localization function and, for this reason, is used by *Safe Halt*. The UNICAR*agil* ADS uses European Terrestrial Reference System 1989 (ETRS89) as the global localization coordinate system. All explicit and implicit trajectories are planned in this coordinate system. Therefore, the functions of *Safe Halt* use this system. The vehicle yaw angle is described in a local navigation coordinate system East-North-Up (ENU). For node U6 *Vehicle Dynamics State Estimation*, there are the requirements:

**Req. U6-1:** Vehicle Dynamics State Estimation shall provide the global vehicle position in the ETRS89 coordinate system

and

**Req. U6-2:** Vehicle Dynamics State Estimation shall provide the yaw angle in a local navigation system ENU

Due to the environment perception setup, the detected objects' velocities are determined relative to the environment perception sensors. For an adaptation of the vehicle speed by *Safe Halt* to objects within the implicit emergency trajectory, this relative speed shall be transformed into an absolute speed. For this purpose, the absolute vehicle speed shall be available. This leads to the requirement:

**Req. U6-3:** Vehicle Dynamics State Estimation shall provide the vehicle velocity

### 6.3.2 U3: Strategical and Tactical Planning

*Safe Halt* generates trajectories that are executed by node U7 *Trajectory Tracking Controller* depicted in Fig. 4-6. Since node U6 *Vehicle Dynamics State Estimation* determines the vehicle position in the ETRS89 coordinate system and the yaw angle in a local ENU system, the trajectories shall also be provided in these coordinate systems. In order to keep the coordinate systems aligned, node U3 *Strategical and Tactical Planning* shall plan the implicit trajectory using these systems. Thus, in addition to the requirements Req. B3- 1 and Req. B3- 2, there is the requirement:

**Req. U3-1:** Strategical and Tactical Planning shall plan implicit emergency trajectories in global coordinate system ETRS89 for positions and local navigation system ENU for yaw angles

As a computation frequency for planning the trajectories, for the UNICAR*agil* ADS 10 Hz are used[92]. The same frequency is also applied for planning the implicit emergency trajectory, leading to the requirement:

**Req. U3-2:** Strategical and Tactical Planning shall plan implicit emergency trajectories with at least 10 Hz

A location-dependent sampling of the implicit emergency trajectory is chosen as the compression for the planned implicit emergency trajectory. The sampling rate depends on the curvature of the trajectory. Tight curves of the trajectory are sampled at a high rate, while curveless regions can be sampled at a lower rate. The sampling rate shall be selected to describe the trajectory in sufficient detail to avoid collision with objects in the environment. This leads to the requirement:

**Req. U3-3:** Strategical and Tactical Planning shall sample the planned implicit emergency trajectory at a curvature-dependent rate to compress it. The actual sampling rate selection is TBD

For a first delimitation of the solution space, the wiki of the free geographic database OpenStreetMap contributes to quality assurance regarding the accuracy[93] of the spatial course of streets in maps:

*To generalize, though, sharp curves (those having a small radius) require many closely-spaced nodes, while broad, long-radius curves can consist of fewer nodes having more distance between them*

The requirements for the content of the implicit emergency trajectory depend on all subsequent trajectory users. *Safe Halt* needs the content for the position, yaw angle, and velocity values. Supplementary data enables *Safe Halt* to perform additional functions. For example, the activation of the direction indicators can also be specified during the planning and transmitted to *Safe Halt*. To validate the integrity of the emergency trajectory, the number of sample values is also included in the trajectory. Further requirements for the content of the emergency trajectories arise from the subsequent node U7 *Trajectory Tracking Controlling*. The UNICAR*agil* ADS uses two independent functions for vehicle localization. The planning of the trajectories is based on one of the localization functions, while the control uses a different source. Both localizations have a systematic offset that shall be resolved by a correction[94]. The data necessary for this correction is therefore preserved in the implicit emergency trajectory. For the planning of the implicit emergency trajectory, the following requirement therefore applies:

[92] Buchholz, M. et al.: Automation of the UNICARagil vehicles (2020), p. 19.

[93] OpenStreetMap contributors: Accuracy OpenStreetMap Wiki (2022).

[94] Homolla, T. et al.: Verfahren zur Korrektur von inkonsistenten Lokalisierungsdaten (2021).

**Req. U3-4:** Strategical and Tactical Planning shall plan implicit emergency trajectories with the content of Annex A

### 6.3.3 U5 and U9: Trajectory Selection and Self-Perception

Node U9 *Self-Perception* depicted on Fig. 4-6 is used to monitor the capabilities of the ADS. If the capabilities of the ADS are no longer sufficient for mission execution, a change of the vehicle operation mode to *Safe Halt* is initiated. For this purpose, node U5 *Trajectory Selection* is integrated into the ADS and performs this mode change. The function switches between the planned trajectory for the vehicle operation mode *Automated Driving* to the trajectory of mode *Safe Halt*. Thus, in addition to the request Req. B6- 1, the request for node U5 *Trajectory Selection* is requirement:

**Req. U5-1:** Trajectory selection should respond to a change of vehicle operating mode to Safe Halt with a change of the selected trajectory source to Safe Halt

### 6.3.4 U7: Trajectory Tracking Controlling

The UNICAR*agil* AV has four independent driving dynamics actuators. This setup allows the sideslip angle to be controlled freely within certain limits. The realizable areas of the center of rotations are described as kinematic limits of the AV. The dynamic limits of the vehicle include, in particular, the selected actuators and their capabilities. A vector of execution limits[95] has been defined for the UNICAR*agil* ADS. This vector shall be determined by U7 *Trajectory Tracking Controlling* and transmitted to *Safe Halt*:

**Req. U7-1:** Trajectory Tracking Controlling shall determine and provide the vector with the kinematic and dynamic execution limits of the UNICARagil AV

## 6.4 Concepts for Neighboring and Superordinate Systems in the UNICAR*agil* ADS

The functional nodes of the UNICAR*agil* ADS are shown in Fig. 4-6.

For planning the implicit emergency trajectory for *Safe Halt*, a dedicated planning algorithm[96] is used. This planning algorithm is integrated into node U3 *Strategical and Tactical Planning*

[95] Homolla, T.; Winner, H.: Encapsulated trajectory tracking control (2022), p. 5.

[96] Wang, L. et al.: Real-Time Safe Stop Trajectory Planning (2020).

depicted on Fig. 4-6 and calculates the implicit emergency trajectory for *Safe Halt*. Its implementation satisfies Req. B3- 1, Req. B3- 3, Req. B3- 4, Req. B3- 5, Req. B3- 7, Req. U3- 1, Req. U3- 2, Req. U3- 3 and Req. U3- 4. In contrast, the requirements Req. B3- 2, Req. B3- 6 and Req. B3- 8 are not yet fully met.

In the UNICAR*agil* ADS, as node U6 *Vehicle Dynamics State Estimation*, a function based on Micro-Electro-Mechanical Systems (MEMS)-Inertial Measurement Unit (IMU) and multi-frequency, multi-constellation Real-Time Kinematic (RTK) Global Navigation Satellite System (GNSS) is used[97] [98]. Of particular note are the capabilities of the function to evaluate its integrity about its dynamics state estimate[99] [100]. Integrating this concept requirements Req. B2- 1, Req. U6- 1 and Req. U6- 2 are satisfied.

The UNICAR*agil* ADS has node U9 *Self-Perception*[101a] for monitoring the capabilities of the ADS. The vehicle capabilities are compared with the requirements by the current route segment[101b]. Based on this concept, Req. B6- 1 can be fulfilled. If the capabilities of the ADS are insufficient for safe mission execution, a vehicle operation mode change to *Safe Halt* is initiated. This is done by changing the trajectory source in node U5 *Trajectory Selection*, thus satisfying Req. U5- 1.

For node U7 *Trajectory Tracking Controlling* an encapsulated controller architecture[102] [103] is used. It takes the vehicle's dynamics state, which is determined by node U6 *Vehicle Dynamics State Estimation*, and the trajectory as the reference state for calculating the actuating set points for the actuators. Thus, Req. B4- 1 and Req. U7- 1 are satisfied.

## 6.5 Conclusion

This chapter elaborates on the generic and specific requirements for neighboring and superordinate systems in a generic and the UNICAR*agil* ADS. These requirements can be adopted by the functional developers of the analyzed systems, thus enabling the integration of *Safe Halt* into the ADS. Additionally, the effects of integrating *Safe Halt* into an ADS are highlighted. For the UNICAR*agil* ADS, the concepts for satisfying the requirements of the neighboring systems are presented.

[97] Buchholz, M. et al.: Automation of the UNICARagil vehicles (2020), p. 1535.

[98] Gottschalg, G.: Data Fusion Architecture with Integrity Monitoring (2022).

[99] Gottschalg, G. et al.: Integrity Concept for Sensor Fusion Algorithms (2020).

[100] Gottschalg, G. et al.: Integrity Based Data Fusion of Redundant Fusion Filters (2021).

[101] Stolte, T. et al.: Towards Safety Concepts for Automated Vehicles (2020). a: p. 1576; b: p. 1578.

[102] Homolla, T.; Winner, H.: Encapsulated trajectory tracking control (2022).

[103] Homolla, T.: Gekapselte Trajektorienfolgeregelung für autonomes Fahren (2023).

Now that the system boundaries and interfaces of *Safe Halt*, as well as the requirements and concepts for neighboring and superordinate systems of *Safe Halt*, have been determined, the requirements and concepts for the sub-functions of *Safe Halt* can be elaborated in the upcoming chapters.

# 7 Safe Halt Implicit Emergency Trajectory Reconstruction

This chapter explains the generic and UNICAR*agil* Automated Driving System (ADS) specific requirements and concepts for the lossless reconstruction function of the implicit emergency trajectory. These functions are provided by the generic function block B11, which is shown in Fig. 5-2.

The function receives the compressed implicit emergency trajectory as its input. The reconstructed implicit emergency trajectory is the output. For time stamping the input and output data, as well as for monitoring its own computation time, time references are available to the function as controls. For the computations of the function, computing power is consumed as the mechanism.

This chapter starts with a description of generic requirements in Sec. 7.1. Subsequently, requirements specific to the UNICAR*agil* ADS are added in Sec. 7.2. The concepts for the UNICAR*agil* ADS are presented in Sec. 7.3. Finally, the chapter is summarized in Sec. 7.4.

## 7.1 Requirements for the Safe Halt Reconstruction in a Generic ADS

The node B3 *Strategical and Tactical Planning* shown in Fig. 4-5 transmits the implicit emergency trajectory as sampled values. The sampled values are the position course, the yaw angle course, and the velocity course. All values shall be reconstructed without loss. The reconstruction must not result in unacceptable deviations between the originally planned and reconstructed trajectories. This leads to the requirement:

**Req. B11-1:** Safe Halt Reconstruction shall reconstruct the sampled values of the implicit emergency trajectory with negligible error

The reconstruction of the pose course may lead to an oscillatory result. These potential oscillations in the pose course depend on the selected reconstruction method. These oscillations shall remain below the perception threshold of translational accelerations of humans. According to Betz[104], the perception threshold lies within the range of $0.17\,\mathrm{m\,s^{-2}}$ to $0.2\,\mathrm{m\,s^{-2}}$. For the *Reconstruction* functionality, therefore, follows the requirement:

[104] Betz, A.: Feasibility analysis and design of WMDS (2015), p. 16.

**Req. B11-2:** Safe Halt Reconstruction shall keep oscillations in the pose course, which arise due to the chosen reconstruction method, below the perception threshold of translational accelerations of humans with $0.15\,\mathrm{m\,s^{-2}}$

The execution times of the function may vary even with identical input because the hardware and software combinations may not be deterministic and thus real-time capable concerning execution time. A new implicit emergency trajectory and, thus, a new reconstruction of an implicit emergency trajectory is expected within an average time frame at the input of node B11 *Safe Halt Reconstruction*. In contrast, the reconstruction of the implicit emergency trajectory takes less than this time frame on average. Therefore, the calculations for the reconstruction take only about a fraction of the time, whereas no calculations shall take place most of the time. During this waiting time, the function shall be able to react to a new implicit emergency trajectory to be reconstructed at any time and shall not consume any computation power. Setting to sleep and waking up of the function is therefore required. During sleep time, the consumption of computation power is minimized but shall start with the computations if a new implicit emergency trajectory is available. This leads to the requirement:

**Req. B11-3:** Safe Halt Reconstruction shall start with its computation when a new implicit emergency trajectory is available. When the computations are done, and there is no new input available, Safe Halt Reconstruction shall minimize computation power consumption

## 7.2 Specific Requirements for the Safe Halt Reconstruction in the UNICAR*agil* ADS

This section describes the requirements for reconstructing the implicit emergency trajectory for the UNICAR*agil* ADS.

For the UNICAR*agil* ADS, the trajectories are described in the global European Terrestrial Reference System 1989 (ETRS89). This coordinate system defines positions with two angles and one height. Since the earth's curvature is negligible in the vehicle near-field (cf. Annex C) and the representation in a cartesian local navigation coordinate system simplifies the calculations, the implicit emergency trajectories are preprocessed and thus transformed into a local navigation coordinate system. The samples of the implicit emergency trajectory, transformed after this preprocessing, are subsequently reconstructed. Since a reconstruction can only be done if the exact compression method is known and in service-oriented software architecture, different sources for implicit emergency trajectories with different compression methods can exist, an exact reconstruction is omitted. Instead, an interpolation method is chosen to fulfill the requirements of *Safe Halt*.

The reconstruction in *Safe Halt* for the UNICAR*agil* ADS consists of two different functions. One of the functions is given the node number U11 *Safe Halt Preprocessing* and the other the node number U12 *Safe Halt Interpolation*. On Fig. 7-1, the functional blocks are shown.

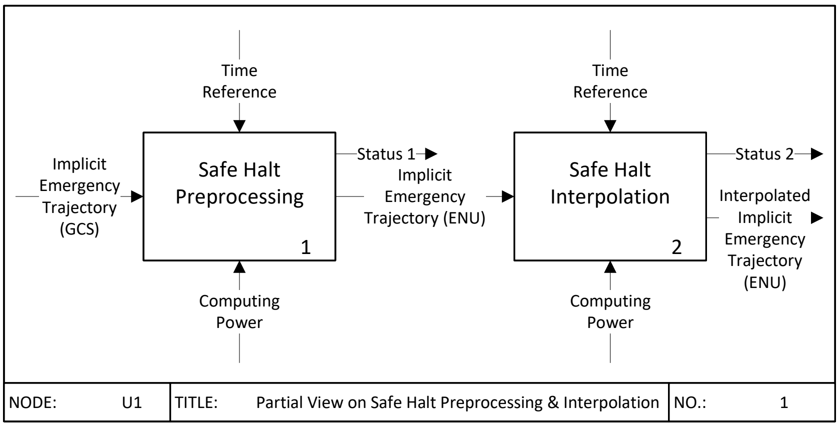

Time Reference
Time Reference
Implicit Emergency Trajectory (GCS)
Safe Halt Preprocessing
1
Status 1
Implicit Emergency Trajectory (ENU)
Safe Halt Interpolation
2
Status 2
Interpolated Implicit Emergency Trajectory (ENU)
Computing Power
Computing Power
NODE: U1
TITLE: Partial View on Safe Halt Preprocessing & Interpolation
NO.: 1

Figure 7-1: U11 Safe Halt Preprocessing and U12 Safe Halt Interpolation

The triggers for the function blocks are listed in Tab. 7-1. Implicit emergency trajectory preprocessing shall start when a new implicit emergency trajectory is received. Interpolation shall start after the preprocessing of the implicit emergency trajectory. Separate requirements are defined for both functions. The labels of the requirements are derived from the node label. It starts with the abbreviation **Req.**, followed by the node label and an incrementing number.

Table 7-1: Trigger for Safe Halt Preprocessing and Safe Halt Interpolation

| Function | Trigger |
|---|---|
| Safe Halt Preprocessing | New Implicit Emergency Trajectory |
| Safe Halt Interpolation | New Preprocessed Implicit Emergency Trajectory |

### 7.2.1 U11: Safe Halt Preprocessing

To correct the systematic offset between two localization functions, the implicit emergency trajectory includes the global position of the planning time (cf. Annex A). The U7 *Trajectory Tracking Controlling* node of the UNICAR*agil* ADS uses this position to correct the offset between both localization functions. It generates a local navigation coordinate system at this planning position to compute the control deviations[105]. For minimizing the number of transformations, also for *Safe Halt*, the local navigation coordinate system originates in the planning position of the implicit emergency trajectory. From this follows the requirement:

[105] Homolla, T.; Winner, H.: Encapsulated trajectory tracking control (2022), p. 5.

**Req. U11-1:** Safe Halt Preprocessing shall generate a local navigation coordinate system at the position of planning for each implicit trajectory and transform all position samples of the trajectory into this coordinate system

In the content of the implicit emergency path, additional data is required for subsequent functions but not required by U11 *Safe Halt Preprocessing* and U12 *Safe Halt Interpolation*. This data could either be transmitted bypassing the preprocessing function or appended to the preprocessed position data. However, a subsequent synchronization of the additional data with the preprocessed position data is more complex. For this reason, the additional data is not passed by and is passed to the preprocessing function. From this follows the requirement:

**Req. U11-2:** Safe Halt Preprocessing shall return the unused content of the implicit trajectory unchanged, together with the preprocessed position data

In order to be able to check the age of the output data, preprocessing shall provide the output data with a Temps Atomique International (International Atomic Time) (TAI) timestamps. From this follows the requirement:

**Req. U11-3:** Safe Halt Preprocessing shall provide all output data with TAI timestamps

For U9 *Self-Perception*, *Safe Halt* shall also communicate its status. Since, due to the service-oriented software architecture, the function cannot know whether missing input data represents a fault condition or not. The TAI timestamps of the last validly received implicit emergency trajectory and the last successfully transformed implicit emergency trajectory shall be communicated to U9 *Self-Perception* of the UNICAR*agil* ADS. From this follows the requirement:

**Req. U11-4:** Safe Halt Preprocessing shall check all input and output data for plausibility. The time stamps of the last validly checked input and output data shall be provided to Self-Perception

Furthermore, the function monitors its execution time. The execution time will increase if the Electronic Control Unit (ECU) on which the function is executed is busy due to an overload. For this reason, the execution time is transmitted to the *Self-Perception*. With the help of a statistical evaluation of the execution time, *Self-Perception* can detect an overload of the ECU. From this follows the requirement:

**Req. U11-5:** Safe Halt Preprocessing shall determine its execution time to perform its function and communicate it to Self-Perception

### 7.2.2 U12: Safe Halt Interpolation

Since poses and a velocity profile describe the implicit emergency trajectory, the path length $s$ can be used as a reference parameter. Poses and velocities can be described as a function of this parameter. The trajectory to be generated by *Safe Halt* shall be time-sampled and include the poses and their first two derivatives. For this reason, the transition from the spatial to the temporal course of the emergency trajectory poses shall be performed based on the velocity profile. For this calculation, the kinematics of the point mass are used[106]. The vehicle is assumed to be a point mass on the implicit emergency trajectory. The implicit emergency trajectory velocity follows from the derivation of the time to

$$v(s) = \frac{\mathrm{d}s}{\mathrm{d}t} = \dot{s}. \tag{7-1}$$

Separating the variables gives the solution

$$\mathrm{d}t = \frac{\mathrm{d}s}{v(s)}. \tag{7-2}$$

A mutual integration results in

$$\int_{t_0}^{t} \mathrm{d}t = \int_{s_0}^{s} \frac{\mathrm{d}s}{v(s)}. \tag{7-3}$$

Considering the vehicle as a point mass, its motion in the local navigation coordinate system can be described by

$$\vec{r} = \vec{r}(s) = \begin{pmatrix} E(s) \\ N(s) \end{pmatrix}, \tag{7-4}$$

with $E(s)$ being the east value and $N(s)$ being the north value in the cartesian local navigation system. The fixed reference point is at the origin of the local navigation coordinate system. Since the vehicle is floor-bound, the third axis, the up-axis, is neglected.

If this equation is derived, the velocity vector is described by

[106] O'Reilly, O. M.: Engineering Dynamics (2019), p. 4.

$$\vec{v} = \dot{\vec{r}} = \frac{\mathrm{d}\vec{r}}{\mathrm{d}t} = \frac{\mathrm{d}\vec{r}}{\mathrm{d}s}\frac{\mathrm{d}s}{\mathrm{d}t} = \frac{\mathrm{d}\vec{r}}{\mathrm{d}s} \cdot \dot{s}. \tag{7-5}$$

If this equation is derived a second time, the acceleration vector is obtained as

$$\vec{a} = \dot{\vec{v}} = \frac{\mathrm{d}\vec{v}}{\mathrm{d}t} = \frac{\mathrm{d}^2\vec{r}}{\mathrm{d}s^2} \cdot \dot{s}^2 + \frac{\mathrm{d}\vec{r}}{\mathrm{d}s} \cdot \ddot{s}. \tag{7-6}$$

An implicit emergency trajectory includes $n$ sampled poses and maximum velocities. Accordingly, $n-1$ segments exist between these samples. Thus, for each segment $i$, the relation of Equ. 7-5 and Equ. 7-6 exist.

### Implicit Emergency Trajectory Positions

To apply Equ. 7-5 and Equ. 7-6, the components of the formulas shall be calculable. For the calculation of the *Safe Halt* output follows, therefore the requirement for the position interpolation:

**Req. U12-1:** Safe Halt Interpolation shall be able to calculate all entries of the table Tab. 7-2 for position interpolation

Table 7-2: Required Information for the Generation of Emergency Trajectories

| Identification | Description | Symbol |
|---|---|---|
| U12-1-1 | Implicit Emergency Trajectory Velocity | $\dot{s}$ |
| U12-1-2 | Implicit Emergency Trajectory Acceleration | $\ddot{s}$ |
| U12-1-3 | Spatial Gradient of the Implicit Emergency Trajectory | $\frac{\mathrm{d}\vec{r}}{\mathrm{d}s}$ |
| U12-1-4 | Spatial Curvature of the Implicit Emergency Trajectory | $\frac{\mathrm{d}^2\vec{r}}{\mathrm{d}s^2}$ |

The interpolation for positions may result in inaccuracies compared to a perfectly reconstructed implicit emergency trajectory. An estimate of this inaccuracy is therefore necessary. An inaccuracy that is too large shall cause the implicit emergency trajectory to become invalid. To estimate the spatial inaccuracy, the linear distance between successive implicit emergency trajectory positions is calculated as a reference. This value is compared to the length of the interpolated implicit emergency trajectory. If both lengths deviate unacceptably far from each other, for example because the passable space at the side of the implicit emergency trajectory is narrow, this interpolated implicit emergency trajectory must not be used. In this case, this implicit emergency trajectory is discarded, and *Safe Halt* continues to use the trajectory processed in the previous execution. From this follows the requirement

**Req. U12-2:** Safe Halt Interpolation shall compare the inaccuracy resulting from the interpolation of positions with a linear reference. Too large (To Be Determined (TBD)) deviations shall lead to a discard of the implicit emergency trajectory

### Implicit Emergency Trajectory Yaw Angles

A similar interpolation procedure is required for the yaw angle. Again, the yaw angle is described as a function of the path and trajectory length $s$. This leads to

$$\psi = \psi(s). \tag{7-7}$$

Thus the yaw rate follows with

$$\dot{\psi} = \frac{\mathrm{d}\psi}{\mathrm{d}t} = \frac{\mathrm{d}\psi}{\mathrm{d}s}\frac{\mathrm{d}s}{\mathrm{d}t} = \frac{\mathrm{d}\psi}{\mathrm{d}s} \cdot \dot{s}. \tag{7-8}$$

A second derivation results in the yaw acceleration with

$$\ddot{\psi} = \frac{\mathrm{d}\dot{\psi}}{\mathrm{d}t} = \frac{\mathrm{d}^2\psi}{\mathrm{d}s^2} \cdot \dot{s}^2 + \frac{\mathrm{d}\psi}{\mathrm{d}s} \cdot \ddot{s}. \tag{7-9}$$

Thus, the requirement for the interpolation of the yaw angle is as follows:

**Req. U12-3:** Safe Halt Interpolation shall be able to calculate all entries of the table Tab. 7-3 for yaw angle interpolation

Table 7-3: Required Input Information for the Calculation of Emergency Trajectory Yaw Angles

| Identification | Description | Symbol |
|---|---|---|
| U12-3-1 | Implicit Emergency Trajectory Velocity | $\dot{s}$ |
| U12-3-2 | Implicit Emergency Trajectory Acceleration | $\ddot{s}$ |
| U12-3-3 | Yaw Angle Gradient of the Implicit Emergency Trajectory | $\frac{\mathrm{d}\psi}{\mathrm{d}s}$ |
| U12-3-4 | Yaw Angle Curvature of the Implicit Emergency Trajectory | $\frac{\mathrm{d}^2\psi}{\mathrm{d}s^2}$ |

### Implicit Emergency Trajectory Velocity Profile

The last data to be interpolated is the velocity profile of the implicit emergency trajectory. This velocity profile $\dot{s}$ is only available at discrete implicit emergency trajectory elements and shall, therefore, also be interpolated for the intermediate sections. Tab. 7-2 and Tab. 7-3 show that the path and trajectory length velocities $\dot{s}$ and accelerations $\ddot{s}$ are needed for the calculations of velocities and accelerations. Thus, an interpolation method for the velocity profile shall be selected from which $\dot{s}$ and $\ddot{s}$ can be calculated as a function of $s$. This leads to the requirement:

**Req. U12-4:** Safe Halt Interpolation of the velocity profile shall allow the calculation of $\dot{s}$ and $\ddot{s}$

In addition, an interpolation of the velocity profile should be jerk-limited. This increases the comfort for the vehicle occupants. The requirement therefore follows:

**Req. U12-5:** Safe Halt Interpolation of the velocity profile shall limit the vehicle jerk

Similar to the *Safe Halt Preprocessing* functions Req. U11-2, Req. U11-3, Req. U11-4 and Req. U11-5, *Safe Halt Interpolation* shall also include basic functionalities. This leads to the requirements:

**Req. U12-6:** Safe Halt Interpolation shall return the unused content of the implicit trajectory unchanged, together with the interpolated data

and

**Req. U12-7:** Safe Halt Interpolation shall provide all output data with TAI timestamps

and

**Req. U12-8:** Safe Halt Interpolation shall check all input and output data for plausibility. The time stamps of the last validly checked input and output data shall be provided to Self-Perception

and finally

**Req. U12-9:** Safe Halt Interpolation shall determine its execution time to perform its function and communicate it to Self-Perception

## 7.3 Concepts for Safe Halt Implicit Emergency Trajectory Preprocessing and Reconstruction in the UNICAR*agil* ADS

This section presents the concept and implementation of the reference solution for the UNICAR*agil* ADS.

### 7.3.1 U11: Safe Halt Preprocessing

For the concept and implementation of U11 *Safe Halt Preprocessing*, a reference solution is designed that satisfies the generic requirement Req. B11- 3 and the UNICAR*agil* requirements Req. U11- 1, Req. U11- 2, Req. U11- 3, Req. U11- 4 and Req. U11- 5. A task-based software architecture (cf. Annex D) is used to find a suitable solution. The software library GeographicLib[107] for geodetic calculations is applied for the preprocessing of the implicit emergency trajectory positions.

### 7.3.2 U12: Safe Halt Interpolation

#### Implicit Emergency Trajectory Positions

Due to the non-equidistant selection of the implicit emergency trajectory elements in time or space, the Nyquist-Shannon sampling theorem cannot be applied. Instead, an interpolation method is chosen that satisfies the requirements for the emergency trajectory generation of *Safe Halt*. The trajectory generation generates not only the emergency trajectory poses but also their first two derivatives (cf. Annex A). The local navigation coordinate system is used for specifying the velocity and acceleration vectors. Therefore, an interpolation method is sought that can be used to interpolate the implicit emergency trajectory in such a way that the emergency trajectories can be generated from it. Seeking the interpolation method, the implicit emergency trajectory positions are considered in more detail. By definition, a trajectory can be described as a changing positional vector. The distance covered is the path or trajectory length $s$. The path velocity $\dot{s}$ is given by

$$|\vec{v}| = v = \frac{\mathrm{d}s}{\mathrm{d}t} = \dot{s}. \tag{7-10}$$

This path velocity $\dot{s}$ is included as the velocity profile in the implicit emergency trajectory (cf. Annex A, Tab. A-2, EP_E6).

[107] Charles Karney: GeographicLib documentation (2022).

Now, formula Equ. 7-4, Equ. 7-5 and Equ. 7-6 are applied. The first path length derivative is denoted by $'$. The second derivative correspondingly with $''$. This results in

$$\frac{\mathrm{d}\vec{r}}{\mathrm{d}s} = \begin{pmatrix} E'(s) \\ N'(s) \end{pmatrix}, \tag{7-11}$$

and

$$\frac{\mathrm{d}^2\vec{r}}{\mathrm{d}s^2} = \begin{pmatrix} E''(s) \\ N''(s) \end{pmatrix}. \tag{7-12}$$

The calculation of the acceleration requires that a spatial interpolation of the implicit emergency trajectory positions can be derived at least twice continuously according to the path length $s$. To determine the reference solution for *Safe Halt*, a linear or quadratic interpolation of the implicit emergency trajectory positions is eliminated. Due to the potentially high number of implicit emergency trajectory samples, higher-order polynomial interpolation is also excluded. Especially with many implicit emergency trajectory samples, a higher degree polynomial interpolation tends to an undesired oscillation behavior. Instead, a piecewise interpolation, also called spline interpolation, is used. A piecewise cubic spline interpolation guarantees twofold differentiability. For this, the spline between two consecutive implicit emergency trajectory positions is described by cubic equations.

The implicit emergency trajectory positions are two-dimensional. The north and east coordinates of the spline shall depend on a shared parameter $u$. For this reason, a parametric spline is chosen to interpolate the implicit emergency trajectory positions. A natural cubic spline is characterized by minimizing the curvature of the spline. An elastic strip, freely rotatable at its ends and spatially fixed but rotatable at its midpoints, takes the course of a natural cubic spline. Therefore, for a minimum curvature of the spline course, the natural parametric spline course is selected. Akima splines are also twofold differentiable, but these are not continuous in the second derivative at the spline's breakpoints and thus lead to an unlimited jerk at these points.

The parameter $u$ is used to describe the interpolated two-dimensional position course. The value $u \in [0,1]$ is selected as the parameter range for interpolation. $u$ is in range of $u_0$ to $u_n$ where $n$ is the number of cubic spline segments. Each implicit emergency trajectory position receives a specific $u_i$. $u = 0$ is the beginning of the implicit emergency trajectory, and $u = 1$ is the end of the implicit emergency trajectory. The partitioning of the interval between $u = 0$ and $u = 1$ can be done using the following methods:

1. Uniformly Spaced

2. Chord Length
3. Centripetal
4. Path Length

Uniformly spaced assumes that the distances between implicit emergency trajectory positions are identical. Chord length calculates the linear distance between two consecutive positions and uses this distance to parameterize the spline. Centripetal is an extension of the chord length method. Here the parameterization is chosen so that a resulting centripetal force is proportional to the curvature of the spline segment. The path length $L$ is used as a parameter in the path length method. The parameterization is defined by $s \in [0, L]$, where $L$ is the total length of the implicit emergency trajectory. The velocity can be specified using the tangent vector in unit length using this representation.

A parameterization in path length allows the calculations of $\dot{s}$ and $\ddot{s}$. Both are needed for the calculations in Equ. 7-5 and Equ. 7-6. Therefore, the path length parameterization is selected. The path length parameterization cannot be calculated directly since the course of the cubic spline must first be known to determine the path segment lengths $s_i$ of each implicit emergency trajectory position segment. Its interpolation is performed in three steps. In the first step, the parametric cubic spline interpolation is performed using centripetal parameterization. As a result, the two-dimensional cubic spline interpolation of the implicit emergency trajectory positions with centripetal parametrization is obtained. In the next step, the implicit emergency trajectory length is calculated. To calculate the total implicit emergency trajectory length, the $i$ segment lengths $s_i$ are added. For a segment $i$ applies

$$s_i(u_i) = \int_{u_{i,0}}^{u_i} \sqrt{\mathrm{E}_i'(u_i)^2 + \mathrm{N}_i'(u_i)^2} \mathrm{d}u_i, \tag{7-13}$$

where the cubic splines with centripetal parametrization are

$$E(u_i) = \mathrm{C}_{3,\mathrm{E,centr},i}(u_i - u_{i,0})^3 + \mathrm{C}_{2,\mathrm{E,centr},i}(u_i - u_{i,0})^2 + \mathrm{C}_{1,\mathrm{E,centr},i}(u_i - u_{i,0}) + \mathrm{C}_{0,\mathrm{E,centr},i} \tag{7-14}$$

and

$$N(u_i) = \mathrm{C}_{3,\mathrm{N,centr},i}(u_i - u_{i,0})^3 + \mathrm{C}_{2,\mathrm{N,centr},i}(u_i - u_{i,0})^2 + \mathrm{C}_{1,\mathrm{N,centr},i}(u_i - u_{i,0}) + \mathrm{C}_{0,\mathrm{N,centr},i} \tag{7-15}$$

Due to the two-dimensional nonlinear relationship, integral Equ. 7-13 must be calculated numerically. For each implicit emergency trajectory position, the path length $s_i$ from the beginning of the implicit emergency trajectory is known after this step. In the third step, these distances are used as a second interpolation parameter. Each implicit emergency trajectory position is assigned the distance $s_i$ as new breakpoints. With this parameterization, the interpolation is performed again. The result is a parametric two times continuously differentiable formulation of the implicit emergency trajectory positions expressed for each spline segment $i$ with

$$E(s_i) = \mathrm{C}_{3,\mathrm{E,pathl},i}(s-s_i)^3 + \mathrm{C}_{2,\mathrm{E,pathl},i}(s-s_i)^2 + \mathrm{C}_{1,\mathrm{E,pathl},i}(s-s_i) + \mathrm{C}_{0,\mathrm{E,pathl},i} \tag{7-16}$$

and

$$N(s_i) = \mathrm{C}_{3,\mathrm{N,pathl},i}(s-s_i)^3 + \mathrm{C}_{2,\mathrm{N,pathl},i}(s-s_i)^2 + \mathrm{C}_{1,\mathrm{N,pathl},i}(s-s_i) + \mathrm{C}_{0,\mathrm{N,pathl},i} \tag{7-17}$$

The trajectory length $s$ since its beginning of the implicit emergency trajectory is used as the spline parameter. Each spline element $i$ has its own parameterization $C_0$, $C_1$, $C_2$ and $C_3$. With this interpolation method requirement Req. U12- 1 is satisfied. After the interpolated path lengths between each implicit emergency trajectory position segment have been calculated, the segment lengths can be compared with the linear distance of the implicit emergency trajectory positions. By comparing both lengths in this way, it is possible to identify whether the interpolation procedure has led to oscillation. This satisfies Req. U12- 2. Due to the selected interpolation method, potential oscillation behavior cannot be excluded.

### Implicit Emergency Trajectory Yaw Angles

Additionally, the yaw angle samples shall be interpolated. Since the yaw angle is a scalar, no parametric interpolation is needed. According to Req. U12- 3 the interpolation shall be continuously differentiable twice. Like the approach before, Req. U12- 3 is satisfied by selecting a natural cubic spline interpolation. The segment lengths $s_i$ between the implicit emergency trajectory positions have already been calculated for the position interpolation and are used again as breakpoints. The interpolation result gives for each spline segment $i$

$$\psi(s_i) = \mathrm{C}_{3,\mathrm{psi,pathl},i}(s-s_i)^3 + \mathrm{C}_{2,\mathrm{psi,pathl},i}(s-s_i)^2 + \mathrm{C}_{1,\mathrm{psi,pathl},i}(s-s_i) + \mathrm{C}_{0,\mathrm{psi,pathl},i} \tag{7-18}$$

### Implicit Emergency Trajectory Velocity Profile

The last data to be interpolated is the velocity profile of the implicit emergency trajectory. Tab. 7-2 and Tab. 7-3 show that the path length velocities $\dot{s}$ and accelerations $\ddot{s}$ are needed for the calculation of velocities and accelerations in the local navigation coordinate system. Thus, an interpolation method is selected from which $\dot{s}$ and $\ddot{s}$ can be calculated as a function of $s$.

For a path length $s$ depended velocity applies

$$a = \frac{\mathrm{d}v}{\mathrm{d}t} = \frac{\mathrm{d}v}{\mathrm{d}s}\frac{\mathrm{d}s}{\mathrm{d}v} = \frac{\mathrm{d}v}{\mathrm{d}s} \cdot v, \tag{7-19}$$

and thus

$$a(s) = \frac{\mathrm{d}v}{\mathrm{d}s}(s) \cdot v(s) = \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}s}(v^2(s)). \tag{7-20}$$

Consequently, the path length acceleration $\ddot{s}(s)$ can be derived from the slope of the squared velocity profile $\dot{s}^2(s)$. A linear interpolation method results in a constant slope of velocity spline segments and, thus, a constant acceleration.

The output of *Safe Halt* is a time-equidistant sampled trajectory. Therefore, the path length parameter $s$ shall be transformed to a function of time $t$. The formula Equ. 7-3 is used to acquire such a function. Since the path acceleration $a(s)$ can be derived from the slope of a $v^2(s)$, this relation is inserted into the formula to

$$\int_{t_0}^{t} \mathrm{d}t = \int_{s_0}^{s} \frac{\mathrm{d}s}{\sqrt{v^2(s)}}. \tag{7-21}$$

As a result, the method used to interpolate the squared velocity profile affects the calculation of the right-hand side of Equ. 7-21. In case a linear approach is selected for the interpolation of the velocity profile, the formula is

$$v^2(s) = C_1 s + C_0. \tag{7-22}$$

$C_1$ and $C_0$ are the parameters of a linear equation of one velocity segment, $s$ is the path length parameter of the implicit emergency trajectory. The integral therefore is

$$\int_{t_0}^{t} \mathrm{d}t = \int_{s_0}^{s} \frac{\mathrm{d}s}{\sqrt{C_1 s + C_0}} = \frac{2\sqrt{C_1 s + C_0}}{C_1} + C, \tag{7-23}$$

and thus

$$t(s) = t_0 + \frac{2\sqrt{C_1 s + C_0}}{C_1} + C. \tag{7-24}$$

Rearranging the equation and setting the initial conditions of $s_0 = 0$ and $t_0 = 0$ leads to

$$s(t) = \frac{C_1 \cdot t^2}{4} + \sqrt{C_0} \cdot t. \tag{7-25}$$

Equ. 7-25 can be solved analytically. The linear interpolation of the location-dependent quadratic velocity profile leads to a constant acceleration within the implicit emergency trajectory segments. At the breakpoints between the velocity segments, the slope abruptly changes. This leads to a step in acceleration and, thus, to an unlimited jerk. Therefore, this does not satisfy Req. U12- 5. A higher polynomial interpolation method shall be selected to increase the driving comfort by limiting the jerk. For a jerk-limited motion, a continuously differentiable interpolation is needed. For this, it is necessary that the interpolation is continuously differentiable at least once. For this reason, the first possible option is a quadratic interpolation of the squared velocity. The quadratic interpolation of the velocity, however, has a weak point. The initial slope and curvature can only be freely selected to a limited extent. On the contrary, it is advantageous for driving comfort if the current acceleration of the vehicle can be included in a renewed velocity interpolation. A cubic spline interpolation allows the specification of the initial slope and, thus, the initial acceleration. With the help of an even higher polynomial interpolation, a further degree of freedom is achieved in the design of the interpolation course. With cubic interpolation, it is possible to freely select the initial slope at the starting point of the velocity profile. As a result, the equation for the cubic velocity profile is

$$v^2(s) = C_3 s^3 + C_2 s^2 + C_1 s + C_0. \tag{7-26}$$

This means that for the time dependency

$$\int_{t_0}^{t} \mathrm{d}t = \int_{s_0}^{s} \frac{\mathrm{d}s}{\sqrt{C_3 s^3 + C_2 s^2 + C_1 s + C_0}}, \tag{7-27}$$

and thus

$$t(s) = t_0 + \int_{s_0}^{s} \frac{\mathrm{d}s}{\sqrt{C_3 s^3 + C_2 s^2 + C_1 s + C_0}}. \tag{7-28}$$

For the interpolation of the location-dependent squared velocity, the methods summarized in Tab. 7-4 can be used. With a cubic interpolation method, the initial slope of the interpolation can be arbitrarily chosen. Thus the current vehicle acceleration can be included in the interpolation of the updated velocity profile. As such, the cubic interpolation is selected to satisfy Req. U12- 5.

Table 7-4: Interpolation Methods for the Location-Dependent Squared Velocity Profile

| Interpolation Method | Resulting Driving Comfort |
|---|---|
| Linear | Constant acceleration on velocity segments, jerk at breakpoints between segments |
| Quadratic | Jerk-limited acceleration during the entire execution of the implicit emergency trajectory |
| Cubic | Jerk-limited acceleration during the entire execution of the implicit emergency trajectory. Inclusion of the current acceleration in the interpolation. |

The lengths $s_i$ of the implicit emergency trajectory positions are used for the cubic interpolation of the squared velocity profile. The squared velocity is a scalar. The interpolation result gives for each spline segment $i$

$$v^2(s_i) = \mathrm{C}_{3,\mathrm{v,pathl},i}(s - s_i)^3 + \mathrm{C}_{2,\mathrm{v,pathl},i}(s - s_i)^2 + \mathrm{C}_{1,\mathrm{v,pathl},i}(s - s_i) + \mathrm{C}_{0,\mathrm{v,pathl},i}. \tag{7-29}$$

With these interpolation concepts, Req. U12- 1, Req. U12- 2, Req. U12- 3 and Req. U12- 4 are satisfied. Overall, these concepts are intended to satisfy Req. B11- 1, Req. B11- 2 and Req. B11- 3. For the concepts and implementations of the interpolation methods, a reference solution is designed that additionally satisfies the UNICAR*agil* requirements Req. U12- 6, Req. U12- 7, Req. U12- 8 and Req. U12- 9. As a consequence of this analysis, a task-based software architecture (cf. Annex D) is used for U12 *Safe Halt Interpolation*.

## 7.4 Conclusion

This chapter identifies the generic and specific requirements for reconstructing the contents of the implicit emergency trajectory. For the UNICAR*agil* ADS, the reconstruction function is split into the two sub-functions *Safe Halt Preprocessing* and *Safe Halt Interpolation* based on the specific requirements. An analysis of the specific requirements concerning the interpolation reveals that position, yaw angle, and velocity profile should be interpolated using natural cubic splines to generate jerk-limited emergency trajectories.

# 8 Safe Halt Sensing, Processing, and World Modeling

In this chapter, the generic and UNICAR*agil* Automated Driving System (ADS) specific requirements and concepts for the environment perception of *Safe Halt* are elaborated. The generic functional block B12 is presented in Fig. 5-2.

The function receives the vehicle environment as input. A world model of this vehicle environment and the status of the function are the output. The vehicle parameters, such as the poses of the environment perception sensors and a time reference, are provided as controls for the function. The environment perception sensors require energy. Also needed is computation power to execute the functions. They are both provided as mechanisms.

This chapter starts with a description of generic requirements in Sec. 8.1. Subsequently, requirements specific to the UNICAR*agil* ADS are added in Sec. 8.2. The concepts for the UNICAR*agil* ADS are presented in Sec. 8.3. Finally, the chapter is summarized in Sec. 8.4.

## 8.1 Requirements for the Safe Halt Sensing, Processing and World Modeling in a Generic ADS

Node B12 *Safe Halt Sensing, Processing and World Modeling* monitors the vehicle environment for collision detection on the implicit emergency trajectory. Relevant are all objects which can occur in the Operational Design Domain (ODD) and which would lead to a collision with the Automated Vehicle (AV) or violate traffic laws. Traffic law violations can result, for example, from an impermissible low time gap to a leading vehicle. This leads to the first generic requirement:

**Req. B12-1:** Safe Halt Sensing, Processing, and World Modeling shall monitor the implicit emergency trajectory for impassable objects

As an output of this monitoring, there should be a world model that can be used by the subsequent node B13 *Safe Halt Strategical and Tactical Planning* to check for collision objects. This leads to the requirement:

**Req. B12-2:** Safe Halt Sensing, Processing, and World Modeling shall output a world model that can be used for collision checking

The required range of the environment sensors depends on the ODD of the AV. The faster the vehicle can move in its ODD, the greater the range of environment perception shall be. For a Minimal Risk Maneuver (MRM), a maximum deceleration of $4\,\mathrm{m\,s^{-2}}$ is allowed[108]. Thus Equ. 8-1 gives an underestimate for the braking distance.

$$s_{\text{brake}} = \frac{v_{\text{max}}^2}{2 \cdot 4\,\mathrm{m\,s^{-2}}} \tag{8-1}$$

and thus the requirement:

**Req. B12-3:** The range of the environment perception of Safe Halt Sensing, Processing, and World Modeling shall be greater than the result of the Equ. 8-1

The environment perception system shall be able to determine the relevant object states. Relevant object states are all states that are required for spatial-temporal collision checking. Therefore, only the existence of an object is required, but a classification of objects is optional. For this purpose, the position of the object in a shared vehicle fixed coordinate system shall be available, and additionally, the absolute velocity of the object. For a collision check, the objects' extent, i.e., the length and width, shall also be determined. Tab. 8-1 shows the relevant object states.

Table 8-1: Mandatory Object States for Collision Check

| Object State | Unit |
|---|---|
| Longitudinal Distance | m |
| Lateral Distance | m |
| Velocity in Longitudinal Direction | $\mathrm{m\,s^{-1}}$ |
| Velocity in Lateral Direction. | $\mathrm{m\,s^{-1}}$ |
| Length | m |
| Width | m |

Thus, for an environment perception system of *Safe Halt* applies

**Req. B12-4:** Safe Halt Sensing, Processing, and World Modeling shall provide object states mentioned in Tab. 8-1

This requirement thus also describes the requirement for the output world model. It reads

**Req. B12-5:** Safe Halt Sensing, Processing, and World Modeling shall provide a world model with the states of Tab. 8-1 in a vehicle fixed coordinate system

An acceptable rate of false-negative object detection shall be achieved. The required false negative rate depends on the frequency and duration of use of the *Safe Halt* vehicle operating mode and

[108] European Parliament and the Council: EU Regulation - Type-approval of ADS (2022), p. 9.

the acceptable rate of false-negatives in these cases. Since false-negative detections can lead to collisions, these must be minimized. False-positive object detection is less relevant than false-negative object detection because *Safe Halt* is a safety function. False-negatives result in slower driving on the implicit emergency trajectory until the Minimal Risk Condition (MRC) is reached. Permanent false-positive detections, however, may prevent reaching the end of the implicit emergency trajectory. Both unnecessarily slow driving on the implicit emergency trajectory and preventing the achievement of the MRC increase the risk of the AV. Both the false-negative and the false-positive rate shall therefore be kept at an acceptable level, yielding:

**Req. B12-6:** Safe Halt Sensing, Processing, and World Modeling shall have an acceptable false-negative and false-positive object detection rate (both To Be Determined (TBD))

The implicit emergency trajectory may lead the AV at its destination to externally reserved[109] areas. These areas are characterized by the fact that they are not reserved for the AV. Other road users can be expected in these areas. The potential encounter with Vulnerable Road User (VRU) is particularly critical. The MRC is approached with creep speed. Thus, a larger field of view is required for the near field, and also a second sensor principle is required since all sensor principles have advantages and disadvantages. It follows:

**Req. B12-7:** The sensor field of view for low speeds of Safe Halt Sensing, Processing, and World Modeling shall be without gaps and reduce the false negative rate by using diversitary environment sensor principles for low velocities

In addition, *Safe Halt* shall receive and process the sensor data to perceive the relevant environment using the environment sensors. The environment sensors work asynchronously and do not transmit data at deterministic time stamps. Therefore, asynchronous reception of sensor data shall be ensured, leading to the requirement:

**Req. B12-8:** Safe Halt Sensing, Processing, and World Modeling shall receive and deserialize the environment sensor data asynchronously

If the field of view of the environment sensors overlap, a fusion of these sensor data shall be considered. Each sensor principle has strengths and weaknesses. By fusing the sensor data, the overall quality of the environment perception can be increased. The ranges of the sensor fields of view may differ significantly. Redundant coverage of the sensor field of view may only be provided at short range. Since *Safe Halt* is a safety function, the false-negative rate in the detection of objects shall be minimized. A problem of fusion is the case if the two sensor data to be fused contradict each other. In case one sensor principle detects an object, but the other object does not confirm it, a contradiction arises that is difficult to resolve. For the *Safe Halt* solution, this problem is circumvented by not resolving this problem. This is done by a logical

[109] Glatzki, F. et al.: Behavioral Attributes for (BSSD) (2021), p. 670.

OR combination of the sensor data of different environment sensors. Once an environment sensor has detected an object, this detection is assumed to be true. This approach minimizes the false negative rate but, on the other hand, maximizes the false positive rate:

**Req. B12-9:** Safe Halt Sensing, Processing, and World Modeling shall fuse the data of overlapping environment sensor fields of view to minimize the false-negative rate for object detection

The typical execution time to execute the function is less than the detection frequency of the environment sensors. While waiting for new sensor data, computing power consumption shall be minimized. For this purpose, the function shall have a procedure to check for new input data and otherwise minimize its resource consumption. This leads to the requirement

**Req. B12-10:** Safe Halt Sensing, Processing, and World Modeling shall start with its computation when new sensor data is available. When the computations are done, and there is no new input available, Safe Halt Sensing, Processing, and World Modeling shall minimize computation power usage

## 8.2 Specific Requirements for the Safe Halt Sensing, Processing, and World Modeling in the UNICAR*agil* ADS

In this section, the requirements for the environment perception system of the *Safe Halt* for the UNICAR*agil* ADS are described. The UNICAR*agil* ADS has with node U2 *Sensing, Processing and World Modeling* depicted on Fig. 4-6 a 360° environment perception[110]. However, this environment perception is not fail-operational. To maximize the number of fault combinations for which the ADS has a fail-safe property, the usage of this environment perception for *Safe Halt* is omitted. Instead, a separate environment perception system is introduced into the ADS. The UNICAR*agil* vehicles are agile and can be operated at low speeds with a sideslip angle of 90°. In addition, the vehicle is designed to be bidirectional. Thus, the requirement for detecting relevant objects is also a 360° perception of the vehicle environment. The motion behavior, in particular the expected vehicle speed, depends on the direction of motion. Since the vehicle is operated in an inner-city ODD (cf. Sec. 2.5.1), it is assumed that the vehicle is mainly operated with small sideslip angles. Large velocities are thus expected in the forward and reverse vehicle directions. Driving with 90° sideslip angle is limited to parking operations. In these cases, low speeds are expected. For the UNICAR*agil* ODD, a maximum speed of $20\,\mathrm{m\,s^{-1}}$ is assumed. According to Equ. 8-1, the sensor range shall thus be in the forward and reverse directions of

[110] Buchholz, M. et al.: Automation of the UNICARagil vehicles (2020), p. 1537.

$$s_{\mathrm{brake,long}} = \frac{(20\,\mathrm{m\,s^{-1}})^2}{2 \cdot 4\,\mathrm{m\,s^{-2}}} = 50\,\mathrm{m}. \tag{8-2}$$

For the lateral direction during parking maneuvers, a velocity of 4 m s$^{-1}$ is assumed. According to Equ. 8-1, the sensor range shall thus be in the sideways directions of

$$s_{\mathrm{brake,lat}} = \frac{(4\,\mathrm{m\,s^{-1}})^2}{2 \cdot 4\,\mathrm{m\,s^{-2}}} = 2\,\mathrm{m}. \tag{8-3}$$

The environment perception is thus divided into several environment sensors. Since these environment sensors operate asynchronously, the sensor data processing is also divided according to Req. B12- 8. To satisfy Req. B12- 5 and Req. B12- 9, a function for transforming the individual sensor data into a vehicle fixed coordinate system and fusion is also introduced. Thus, the environment perception of *Safe Halt* consists of node U13 *Safe Halt Long Range Sensor Front Processing* for high velocity forward, U14 *Safe Halt Long Range Sensor Back Processing* for high velocity backward, U15 *Safe Halt* 360° *Short Range Sensor Processing* for low speeds in all directions, and U16 *Safe Halt Object List Transformation and Fusion* for the transformation of the individual sensor data into a shared vehicle fixed coordinate system and the fusion of the sensor data. On Fig. 8-1, the functional blocks are shown.

The triggers for the described function blocks are shown in Tab. 8-2. The sensor data processing shall start when new sensor data is received. Object list transformation and fusion shall begin when new object lists have been received from all three sensors. Alternatively, a timeout of 100 ms is set so that a failure of one of the environment sensors does not result in a deadlock.

Table 8-2: Trigger for Safe Halt Long and Short Range Environment Perception and Fusion

| Function | Trigger 1 | Trigger 2 |
|---|---|---|
| Safe Halt Long Range Sensor Front Processing | New Long Range Sensor Front Data | |
| Safe Halt Long Range Sensor Back Processing | New Long Range Sensor Back Data | |
| Safe Halt 360° Short Range Sensor Processing | New Short Range Sensor Data | |
| Safe Halt Object List Transformation and Fusion | New Processed Data From all Three Sensors | 100 ms |

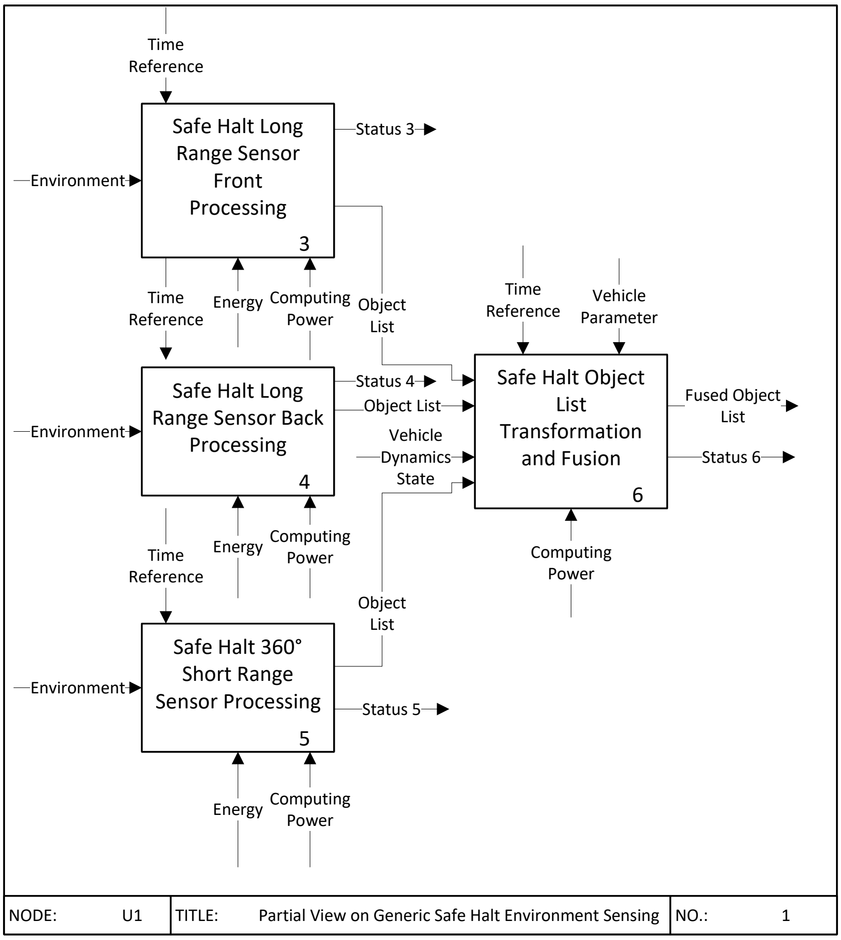


Figure 8-1: Functional Architecture of Safe Halt Sensing, Processing, and World Modeling in the UNICAR*agil* ADS

### 8.2.1 U13 & U14: Safe Halt Long Range Sensor Processing

For long-range environment perception, Equ. 8-2 requires a range of 50 m. This leads to the requirement:

**Req. U13-U14-1:** Safe Halt Long Range Sensor Processing for front and back shall have a minimal range of 50 m

The main directions of motion are forward and backward. The implicit emergency trajectories may lead the AV both to the right and left. Therefore follows the requirement:

**Req. U13-U14-2:** Safe Halt Long Range Sensor Processing for front and back shall be mounted centrally on the front and rear of the vehicle if the sensors' field of view is symmetrical in the sensor's field of view direction

Since the trajectories shall be planned with 10 Hz, the cycle time of the long-range sensors shall also be at least 100 ms. This is the only way to ensure that each new trajectory includes the updated environment. If this requirement is not met, prediction of object states based on the previous sensor cycles can be used. It follows the requirement:

**Req. U13-U14-3:** Safe Halt Long Range Sensor Processing for front and back shall perceive the environment with at least 10 Hz

In order to be able to check the age of the output data, the environment data output shall include a Temps Atomique International (International Atomic Time) (TAI) timestamp. From this follows the requirement:

**Req. U13-U14-4:** Safe Halt Long Range Sensor Processing shall provide all output data with TAI timestamps

For the *Safe Halt* status determination, the status of the long-range sensor perception shall be available. From this follows the requirement

**Req. U13-U14-5:** Safe Halt Long Range Sensor Processing shall check the perceived environment data for plausibility. The time stamps of the last validly checked output data shall be provided to Self-Perception

Furthermore, the function monitors its execution time. From this follows the requirement:

**Req. U13-U14-6:** Safe Halt Long Range Sensor Processing shall determine its execution time to perform its function and communicate it to Self-Perception

### 8.2.2 U15: Safe Halt $360°$ Short Range Sensor Processing

For short-range environment perception, Equ. 8-3 requires a range of 2 m. This leads to the requirement:

**Req. U15-1:** Safe Halt Short Range Sensor Processing for sideways direction shall have a minimal range of 2 m

According to Req. B12- 7, the environment perception for low-speed shall be without gaps and reduce the false negative rate by using diversitary environment sensor principles. The sensor range required for this depends on the creep velocity during MRM and is TBD. Therefore follows the requirement:

**Req. U15-2:** Safe Halt Short Range Sensor Processing for front and back direction shall have a minimal range of TBD

Req. U13-U14- 3, Req. U13-U14- 4, Req. U13-U14- 5 and Req. U13-U14- 6 are also valid for the Safe Halt Short Range Sensor Processing.

### 8.2.3 U16: Safe Halt Object List Transformation and Fusion

The environment perception for *Safe Halt* in UNICAR*agil* ADS consists of three separate environment sensor components. All environment sensor components detect objects in their sensor field of view and output them in their sensor coordinate system. Due to the overlap of the sensor's field of view in the front and rear of the vehicle, fusion of these sensor data is required to minimize false-negative object detection, resulting in the following requirement:

**Req. U16-1:** Safe Halt Object List Transformation and Fusion shall fuse the sensor data from the Safe Halt Long Range Sensor Processing with the Safe Halt 360° Short Range Sensor Processing. The sensor data shall be logically combined in an OR fashion

For this sensor data fusion and the later evaluation of the world model, the environment sensor data shall be transformed into a shared coordinate system fixed to the vehicle. The output world model then contains objects that are referenced to this coordinate system. From this follows the requirement:

**Req. U16-2:** Safe Halt Object List Transformation and Fusion shall transform the sensor data of all environment sensors in a shared vehicle fixed coordinate system. For UNICARagil, this coordinate system is located at the intersection of the diagonals through the wheel contact points

## 8.3 Concepts for the Safe Halt Sensing, Processing, and World Modeling in the UNICAR*agil* ADS

For automotive applications, lidar sensors, radar sensors, cameras, and ultrasonic sensors are used to monitor the vehicle environment. Cameras are often used to classify objects. Camera image processing is also resource intensive. Since an embedded hardware implementation for the *Safe Halt* functions is aimed for and this hardware is resource-reduced, the use of a camera sensor system is not considered. Ultrasonic sensors are particularly relevant for the close range. The selected positioning of the ultrasonic sensors allows a wide field of view of objects to be monitored, although the range is too short for higher vehicle speeds. Radar sensors and lidar sensors can be used for higher speeds. Lidar sensors have a high resolution and a good accuracy in determining the position of objects. The disadvantage of lidar sensors is their susceptibility to external conditions such as weather. Radar sensors are more robust against weather conditions. The disadvantage of radar sensors is the lower angular accuracy and resolution of position measurements.

### 8.3.1 U13, U14 & U15: Safe Halt Sensor Processing

For the application of *Safe Halt* in the project UNICAR*agil*, the perception sensor setup for Safe Halt uses the radar and ultrasonic sensor principles. Two Continental ARS-408-21 radar sensors[111] (cf. Annex F) are installed in the vehicle. One radar sensor is in the center of the vehicle's front, and one is in the center of the vehicle's rear. According to the technical documentation of the radar sensors, new sensor data is transmitted cyclically every 70 ms to 80 ms. The transmission time depends on the number of objects on the object list. The sensor data is transmitted via an Controller Area Network (CAN)-Bus. The data rate is up to 500 $\mathrm{kbit\,s^{-1}}$. The data is transformed to Ethernet with the CAN-Ethernet-Gateway Ixxat FRC-EP170[112]. This gateway is used since the Electronic Control Unit (ECU) for the *Safe Halt* functionalities does not provide enough CAN interfaces to connect the radar sensors directly. The FRC-EP170 model provides interfaces for four separate CAN networks and one Ethernet network. The bidirectional communication between the CAN network and the Ethernet network is IP-based using the Transmission Control Protocol (TCP) protocol.

Twenty-four ultrasonic sensors supplement this environment sensor system. These are distributed around the entire contour of the vehicle to provide 360° coverage of the vehicle's surroundings with a range of up to 4 m. In state of the art for parking assistance systems, up to 6 sensors each are used in the rear and front of a vehicle. A maximum of 12 ultrasonic sensors can thus be connected to one ultrasonic sensor ECU. With this number of sensors, the vehicle's near field can be monitored without gaps. Since for the UNICAR*agil* AV, it is also possible to drive the vehicle sideways, twelve additional sensors are integrated into the vehicle sides, and these are connected to a second ECU. Simulations by Valeo Schalter und Sensoren GmbH demonstrate that this setup meets the requirements of 360° coverage. Due to the Non-Disclosure Agreement (NDA), these simulation results cannot be presented here. In addition, four cameras with fisheye lenses are installed in the vehicle. The camera images are fused with the data from the ultrasonic sensors directly on the ECU of the ultrasonic sensors. This means that *Safe Halt* does not evaluate the camera images directly. The sensors also have a defined cycle time, but here too, the transmission time depends on the number of detected objects. The sensor data is transmitted via BroadR-Reach. The ultrasonic sensor, the cameras with fisheye lenses, and the ECU both are provided by the Valeo Schalter und Sensoren GmbH[113]. Due to the availability of the sensor components, two different ultrasonic sensor models are used. The ultrasonic sensors in the front and rear of the vehicle and the cameras with the fisheye lenses make up the first model. There are no publicly available data sheets for this model. The ultrasonic sensors in the vehicle sides form the second model. The data sheet for these sensors can be found in Annex H. Twelve ultrasonic sensors are connected to one ECU for each sensor model. There is a primary

[111] CONTINENTAL ENGINEERING SERVICES: ARS 408-21 Website (2022).

[112] HMS Industrial Networks: Ixxat FRC-EP170 Website (2022).

[113] VALEO SERVICE: Valeo Ultrasonic Sensor Website (2022).

and a secondary ECU. The four fisheye cameras are connected to the primary ultrasonic sensor ECU. The controller provides the supply voltage for the ultrasonic sensors and cameras and communicates with the ultrasonic sensors via Local Interconnect Network (LIN) connections and with the cameras via Low Voltage Differential Signaling (LVDS) connections. The primary ultrasonic ECU is a gateway for the secondary ECU. It receives all data from the secondary ECU and sends it to the *Safe Halt* 360° *Short Range Sensor Processing* function. In the UNICAR*agil* vehicles, a communication ring is set up with four Ethernet switches. The two ECUs are each connected to one of the switches. Since the switches do not offer BroadR-Reach communication, BroadR-Reach-to-Ethernet converters are used as a bidirectional gateway between BroadR-Reach and Ethernet.

Fig. 8-2 shows the UNICAR*agil* sensor setup for *Safe Halt*. Shown are the outlines of the UNICAR*agil* prototype vehicle. The two radar fields of view for the near and far ranges are schematically illustrated. Also shown is the 360° environment perception with ultrasonic sensors and cameras with fisheye lenses.

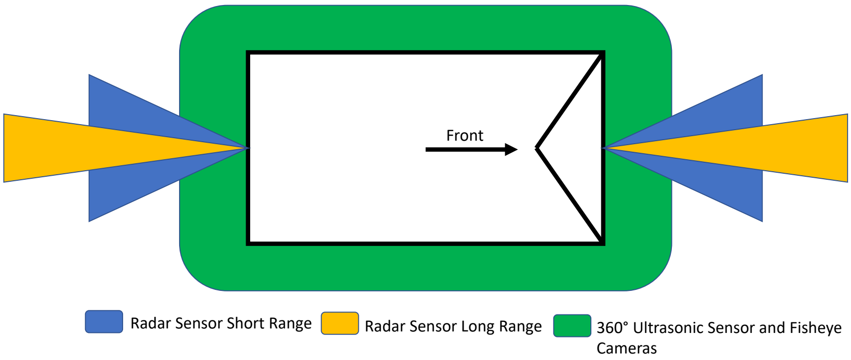


Figure 8-2: Schematic Sensor Setup for Safe Halt Sensing, Processing, and World Modeling in the UNICAR*agil* ADS

### 8.3.2 U16: Safe Halt Object List Transformation and Fusion

The environment sensors output their data in their sensor fixed coordinate systems. For further processing of the sensor data, the sensor data shall be transformed into a shared vehicle fixed coordinate system. First, the shared vehicle-fixed reference point shall be selected. For UNICAR*agil*, a vehicle-fixed reference point is selected, which can be used equally by all functions of the ADS effect chain. The vehicle fixed reference point is located at the intersection of the diagonals through the wheel contact points. The vehicle dynamics state estimation calculates accelerations and yaw rates in its coordinate system. This information is then transformed into the vehicle reference point. The lever arm between the vehicle fixed reference point and vehicle dynamics state estimation shall be as short as possible to keep the accuracy of the transformation as high as possible. For this reason, the vehicle fixed reference point is at the identical height above the ground as the reference point of the vehicle dynamics state estimation. The height of the vehicle's fixed reference point is optional for *Safe Halt* since this axis can be neglected for the movement of a floor-bound vehicle.

Besides transforming the sensor data into a shared coordinate system, sensor data fusion is performed. For *Safe Halt*, a sensor setup consisting of radar sensors in the front and back and 360° coverage with ultrasonic sensors and cameras is used. The ranges of the sensor's field of view differ significantly. The radar sensors used can detect objects in a limited field of view up to 200 m, with the ultrasonic sensors achieving a range up to 4 m. To minimize the false positive rate, the overlapping sensor fields of view of radar sensors and ultrasonic sensors are logically or combined. The representation of the world model can be described by free space or object lists. Since the environment sensors have been parameterized in such a way that they generate object lists, and the following functions of *Safe Halt* can also use object lists, object lists are selected as the world model for *Safe Halt*. All states of the objects are transformed into the vehicle fixed coordinate system.

### 8.3.3 Synthesis of the Safe Halt Sensing, Processing, and World Modeling

The final functional blocks U13, U14, U15, and U16 are presented on Fig. 8-3.

The environment sensors are installed in the four prototype UNICAR*agil* vehicles. On Fig. 8-4, the vehicle autoSHUTTLE is shown with highlighted sensors for *Safe Halt*. The environment sensors not visible in the image are distributed on the vehicle in mirror symmetry to the longitudinal axis.

With this concept the requirements Req. B12- 1, Req. B12- 2, Req. B12- 3, Req. B12- 4, Req. B12- 5, Req. B12- 7, Req. B12- 8, Req. B12- 9, Req. B12- 10 are fulfilled. Due to a lack of comprehensive test case generation in real road traffic, it cannot be verified that Req. B12- 6 is met. The specific UNICAR*agil* *Safe Halt* requirements Req. U13-U14- 1, Req. U13-U14- 2, Req. U13-U14- 3, Req. U13-U14- 4, Req. U13-U14- 5, Req. U13-U14- 6, Req. U15- 1, Req. U15- 2, Req. U16- 1 and Req. U16- 2 are satisfied.

## 8.4 Conclusion

The requirements for Safe Halt Sensing, Processing, and World Modeling are derived in this chapter. This is done first for a generic ADS, followed by the application in the UNICAR*agil* ADS. The high degrees in the agility of the UNICAR*agil* AV pose particular challenges to the sensor fields of view. The sensor setup for *Safe Halt* thus consists of a 360° environment perception. Due to the bi-directionality of the AV and the assumption that the vehicles are primarily operated with a small sideslip angle, the range of the environment detection in the front and rear direction shall be dimensioned according to the maximum velocity of the vehicle. On the other hand, with large sideslip angles, the vehicle is only operated at low speeds so that a small sensor range can

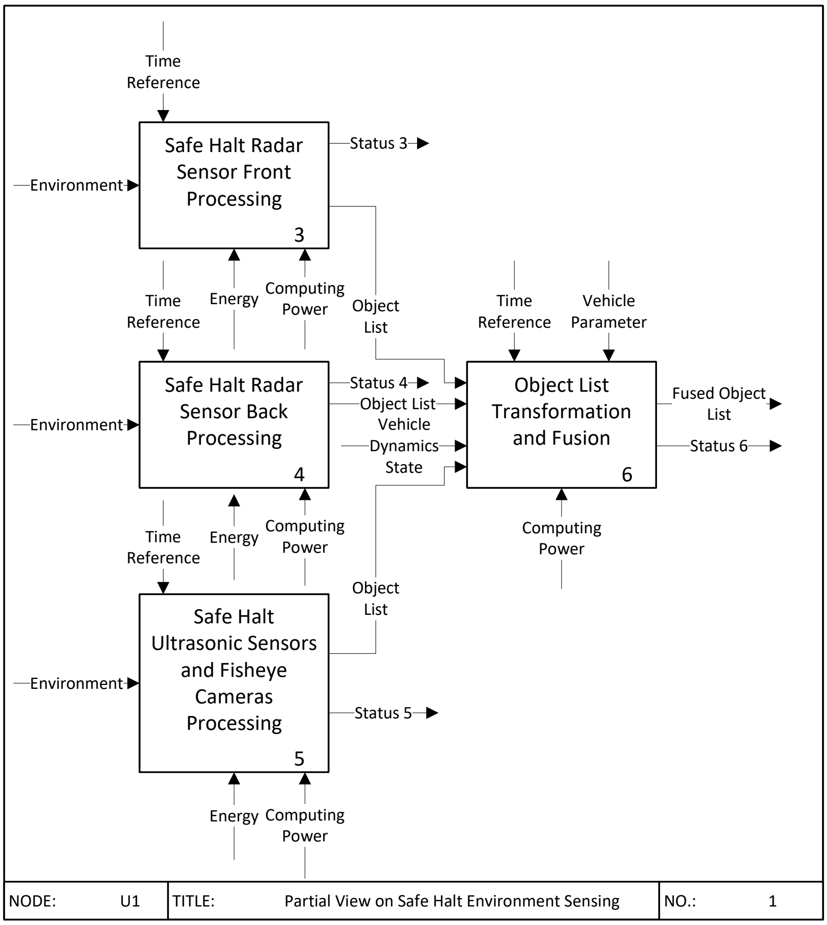
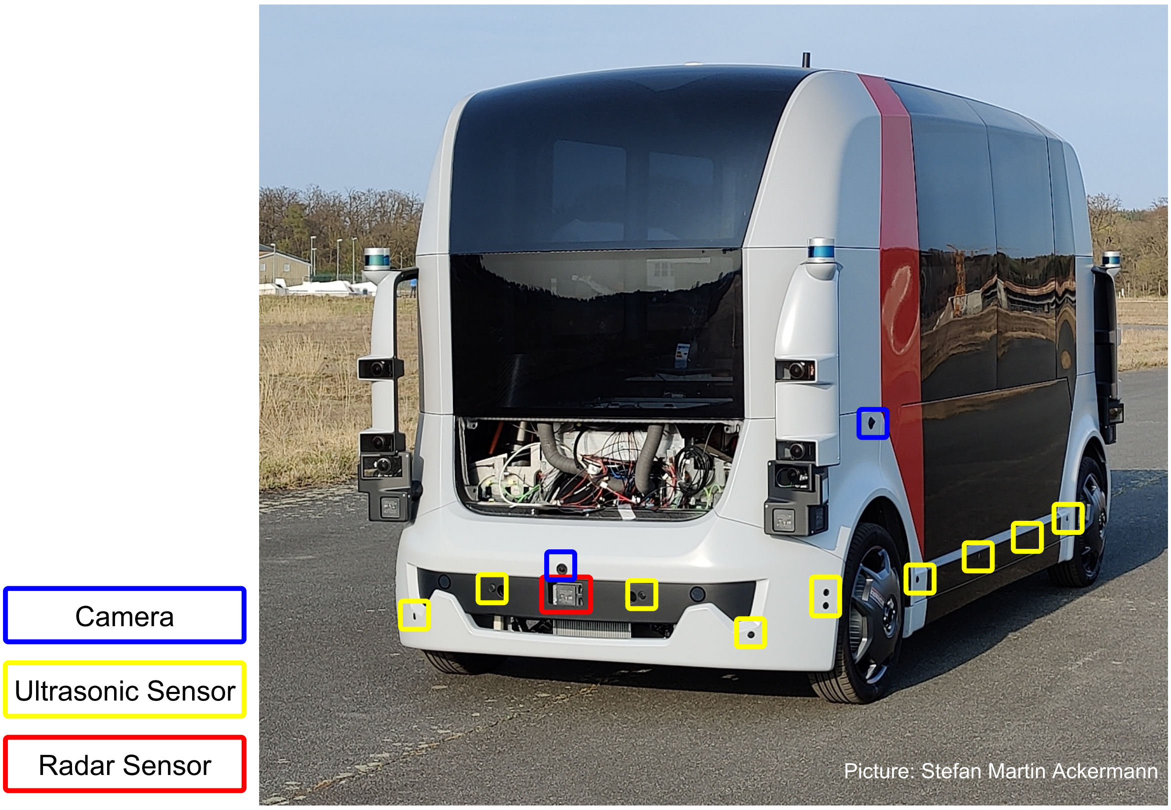


Figure 8-3: Final Functional Architecture of Safe Halt Sensing, Processing, and World Modeling in the UNICAR*agil* ADS

satisfy the requirements for the lateral direction. For the UNICAR*agil* AV, one radar sensor in the front, one in the rear, and a 360° coverage with ultrasonic sensors and four cameras with fisheye lenses are used.



Picture: Stefan Martin Ackermann

Figure 8-4: Safe Halt Environment Sensor Setup of autoSHUTTLE

# 9 Safe Halt Strategical and Tactical Planning

In this chapter, the generic and UNICAR*agil* Automated Driving System (ADS) specific requirements and concepts for the strategical and tactical planning of *Safe Halt* are explained. The generic functional block B13 is presented on Fig. 5-2.

As input, the function receives the reconstructed implicit emergency trajectory. For strategical and tactical planning, the vehicle dynamics state is also provided as input to the function. Finally, the generated world model for collision detection and adaption of vehicle velocity to objects within the implicit emergency trajectory is provided as input. *Safe Halt Strategical and Tactical Planning* has the task of generating trajectories that are both collision-free and comply with traffic rules. For this purpose, behavioral rules are assigned to the function as control. In addition to a time reference, vehicle parameters are also required by the function. These vehicle parameters are needed for the collision check. The functions shall be executed on an Electronic Control Unit (ECU). This is ensured by providing computing power as a mechanism. The function's output is a trajectory that can be used by subsequent function node B4 *Trajectory Tracking Controller* depicted on Fig. 4-5. In addition, this function outputs signals to activate the hazard warning lights and external warning Human Machine Interface (HMI) components of the Automated Vehicle (AV) and the status of the function.

This chapter starts with a description of generic requirements in Sec. 9.1. Subsequently, requirements specific to the UNICAR*agil* ADS are added in Sec. 9.2. The concepts for the UNICAR*agil* ADS are presented in Sec. 9.3. Finally, the chapter is summarized in Sec. 9.4.

## 9.1 Requirements for the Safe Halt Strategical and Tactical Planning in a Generic ADS

Node B13 *Safe Halt Strategical and Tactical Planning* has the function of planning and trajectory generating based on the most recent reconstructed implicit emergency trajectory, the vehicle dynamics state, and the latest world model. Therefore, the function has to execute three sub-functions. The first subfunction is the collision check. For the collision check, the two-dimensional extensions of the AV on the implicit emergency trajectory are checked against the two-dimensional extensions of detected objects in the world model. If these extensions overlap in space and time, a collision occurs. The first requirement is, therefore:

**Req. B13-1:** Safe Halt Strategical and Tactical Planning shall check the implicit emergency trajectory for collision objects in the world model in space and time

If such a collision is detected, a second subfunction shall adapt the velocity profile of the implicit emergency trajectory to avoid this collision. In addition, the function shall ensure that traffic rules, such as time gaps to the object on the implicit emergency trajectory, are respected. This leads to the requirement:

**Req. B13-2:** Safe Halt Strategical and Tactical Planning shall adapt the implicit emergency trajectory velocity profile to avoid collisions and to respect traffic rules

Finally, the last subfunction shall generate and output trajectories based on this updated speed profile:

**Req. B13-3:** Safe Halt Strategical and Tactical Planning shall generate trajectories based on the implicit emergency trajectory poses and the adapted velocity profile

The poses and the world model shall be in the same coordinate system to perform the collision check. For this reason, a transformation to a shared coordinate system shall take place. This leads to the requirement:

**Req. B13-4:** Safe Halt Strategical and Tactical Planning shall transform the world model and the implicit emergency trajectory into the same coordinate system for the collision check

Checking for collisions consumes a lot of computing power. For this reason, the number of objects to be checked in detail shall be minimized:

**Req. B13-5:** Safe Halt Strategical and Tactical Planning shall establish filter criteria that minimize the number of detailed collision checks

For updating the velocity profile, the collision object's location and absolute velocity shall be known for each potential collision. This leads to the requirement:

**Req. B13-6:** Collision Check of Safe Halt Strategical and Tactical Planning shall determine the object position $s_{\mathrm{obj}}$ and the object velocity $v_{\mathrm{obj}}(s)$ on the most recent implicit emergency trajectory

Once the position and absolute velocity of an object on the implicit emergency trajectory have been determined, they shall lead to an update of the velocity profile. To reduce the risk, the objective is to move the AV on the implicit emergency trajectory as fast as possible while not violating the original velocity profile. This leads to the requirement:

**Req. B13-7:** Velocity profile adaption of Safe Halt Strategical and Tactical Planning shall adapt the velocity profile with the maximum possible velocity at any time, while not violating the original velocity profile

To avoid deceleration with a large distance in front of an object, a reaction to an object on the implicit emergency trajectory shall only occur once a minimum deceleration is required. This leads to the requirement:

**Req. B13-8:** The reaction to an object on the implicit emergency trajectory shall not be performed until a minimum deceleration (To Be Determined (TBD)) is necessary

If the distance and velocity of the object on the implicit emergency trajectory allow its own AV to accelerate, this acceleration shall be limited, leading to:

**Req. B13-9:** If the states of the collision object allow acceleration of the AV, the acceleration shall be limited to TBD

*Safe Halt Strategical and Tactical Planning* is part of the safety function of *Safe Halt*. The behavior of the function shall therefore be deterministic, and the results of the calculations shall always be available within an estimable period. This leads to the requirement:

**Req. B13-10:** Safe Halt Strategical and Tactical Planning shall be deterministic, and the results of the calculations shall always be available within an estimable time

The function shall determine and output its status for *Safe Halt Self-Perception*, yielding:

**Req. B13-11:** Safe Halt Strategical and Tactical Planning shall determine and output its status

After an object has been detected on the implicit emergency trajectory, it shall be checked whether the environment perception provides quality information about the detected object. This quality information shall be passed to the *Self-Perception* as part of the status of the function:

**Req. B13-12:** If the environment perception provides quality information about an detected object on the implicit emergency trajectory, this information shall be transmitted to the Self-Perception

The typical execution time to generate trajectories is less than the targeted frequency for trajectory generation. While waiting for the next trigger to generate a new emergency trajectory, the computing power consumption shall be minimized. For this purpose, the function shall check if a new trajectory shall be generated and otherwise minimize its resource consumption. This leads to the requirement:

**Req. B13-13:** Safe Halt Strategical and Tactical Planning shall start with its computation when a new trajectory generation is due. When the computations are done, and there is no new trigger for trajectory generation, it shall minimize computation power usage

The function generates trajectories for executing a Minimal Risk Maneuver (MRM). Road users in the vicinity of the AV shall be informed about the execution of this MRM. In addition, the awareness of other road users shall be increased so that they are warned about the upcoming MRM. Therefore, the function shall activate the hazard warning lights and all warning external HMI elements, resulting in the requirement:

**Req. B13-14:** Safe Halt Strategical and Tactical Planning shall output signals to activate the warning hazard lights and all warning external HMI elements to notify other road users about the MRM

## 9.2 Specific Requirements for the Safe Halt Behavior and Trajectory Planning in the UNICAR*agil* ADS

In this section, the requirements for node U17 *Safe Halt Behavior and Trajectory Planning* for the UNICAR*agil* ADS is described. On Fig. 9-1 the functional block is shown.

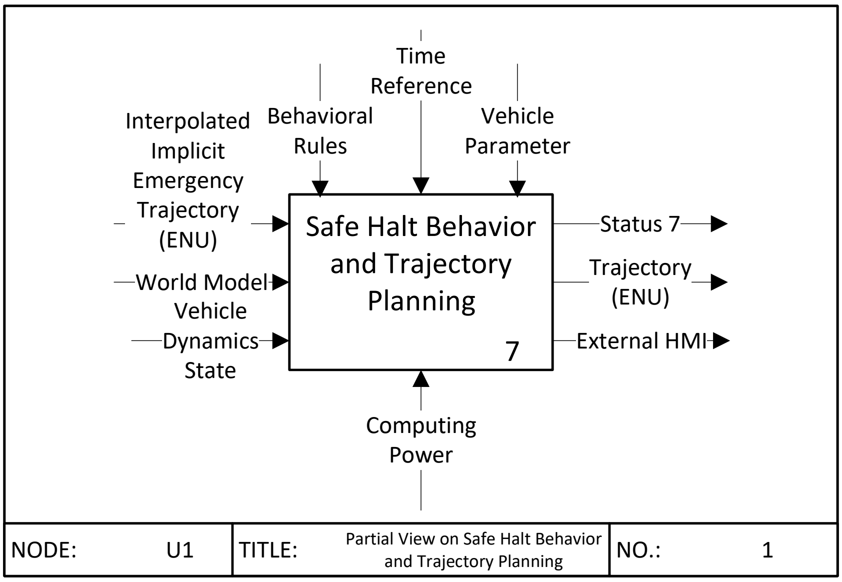

Time Reference
Behavioral Rules
Vehicle Parameter
Interpolated Implicit Emergency Trajectory (ENU)
World Model
Vehicle Dynamics State
Safe Halt Behavior and Trajectory Planning
7
Status 7
Trajectory (ENU)
External HMI
Computing Power
NODE: U1
TITLE: Partial View on Safe Halt Behavior and Trajectory Planning
NO.: 1

Figure 9-1: Safe Halt Behavior and Trajectory Planning in the UNICAR*agil* ADS

The function receives the preprocessed and interpolated implicit emergency trajectory, the vehicle's dynamics state and the world model related to the vehicle-fixed reference point as input. As output the function status is transmitted to the Self-Perception as well as the generated trajectory in the local navigation coordinate system. This trajectory is also the output of the entire *Safe Halt* and is transmitted to node U5 *Trajectory Selection* depicted on Fig. 4-6. In addition, the function outputs a signal to activate the hazard warning lights and the warned external HMI components.

For the UNICAR*agil* ADS, *Safe Halt* transforms the positions of the implicit emergency trajectory into a local navigation coordinate system East-North-Up (ENU). The object list as the world model uses the vehicle fixed coordinate system. From this follows the requirement:

**Req. U17-1:** Safe Halt Behavior and Trajectory Planning shall transform the local navigation system of the implicit emergency trajectory and the vehicle fixed world model in the same coordinate system

UNICAR*agil* vehicles can be operated with a sideslip angle of 90°. Therefore, the collision check shall meet the requirement:

**Req. U17-2:** Safe Halt Behavior and Trajectory Planning shall check for collisions even with sideslip angles of 90°

## 9.3 Concepts for the Safe Halt Behavior and Trajectory Planning in the UNICAR*agil* ADS

This section presents the concept for node U17 *Behavior and Trajectory Planning*. This section is divided into three parts, each of which describes one of the following sub-functions *Collision Check*, *Velocity Profile Adaptation*, and the *Trajectory Generation*.

### 9.3.1 Collision Check

Firstly, all objects in the world model are filtered to minimize the number of objects before performing the detailed collision check. Different filter stages are applied. The computation effort increases monotonously for each applied filter stage.

The entire world model is scanned for the first stage of object filtering. Only the reference points of the objects in the world model are compared with the vehicle-fixed reference point. The relevant filter distance depends on the current velocity $v_{\mathrm{ego}}$ of the AV. The vehicle velocity is measured by node U6 *Vehicle Dynamics State Estimation*. It outputs the vehicle velocity in a local navigation coordinate system. The velocity in the north direction is $v_{\mathrm{ENU,N}}$, in the east direction is $v_{\mathrm{ENU,E}}$ and the up direction in the north direction is $v_{\mathrm{ENU,U}}$ The formula for the vehicle velocity is, therefore

$$v_{\mathrm{ego}} = \sqrt{v^2_{\mathrm{ENU,N}} + v^2_{\mathrm{ENU,E}} + v^2_{\mathrm{ENU,U}}}. \tag{9-1}$$

This velocity is used to calculate the braking distance with Equ. 9-2.

$$d_{\text{relev}} = \frac{v_{\text{ego}}^2}{2 \cdot 4\,\text{m}\,\text{s}^{-2}} \tag{9-2}$$

The relevant distance additionally compensates for inaccuracies caused by neglecting the extension of the collision object and of the AV. For this reason, the relevant distance is multiplied by a safety factor $k_{\text{safe}}$.

$$d_{\text{relev,safe}} = d_{\text{relev}} \cdot k_{\text{safe}} \tag{9-3}$$

The squared distance between the object's reference point and the vehicle's reference point is calculated for each object in the world model. The squared distance is used because the numerical calculation of a root consumes a lot of computing power. The exact distance does not need to be calculated since only a comparison between the squared distance, and the squared relevant distance has to be performed. This squared distance measure is called comparable distance. The first filter criterion is, therefore

$$s_{\text{dist,comp}}^2 < s_{\text{relev}}^2. \tag{9-4}$$

Once the inequality is satisfied, the corresponding object is relevant for the next filter stage.

On Fig. 9-2, the reference points of two objects are shown with green crosses. In addition, the relevant distance $s_{\text{relev}}$ is shown as a circle around the vehicle reference point. Object 1 distance $s_1$ is within the relevant distance and remains relevant, and object 2 with distance $s_2$ is filtered out.

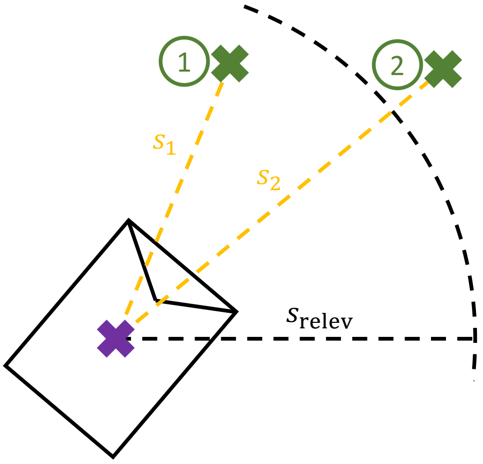


Figure 9-2: Filter Stage 1: Relevant Distance Check

Before the next filter stage, the remaining object list and the implicit emergency trajectory are transformed into a shared coordinate system. The vehicle-fixed coordinate system and the local navigation coordinate system of the implicit emergency trajectory can be used as the shared coordinate system. The result of the collision check is the object state $s_{\mathrm{obj}}$ and $v_{\mathrm{obj}}(s_{\mathrm{obj}})$ on the implicit emergency trajectory. This object state is only valid for the local navigation coordinate system of the emergency trajectory. For this reason, each object that is relevant according to the first filter stage is transformed into the local navigation coordinate system of the current implicit emergency trajectory. For this transformation, the global pose of the vehicle during the environment perception is included in the world model. It can be used to transform the vehicle-fixed object list into the local navigation coordinate system. On Fig. 9-3, the vehicle (black rectangle) can be seen in the local navigation coordinate system of the implicit emergency trajectory (orange). In the world model are two objects (green). Both are available in the vehicle fixed coordinate system (red). Also known is the global position of the vehicle and, thus, the position of the vehicle reference point (purple) within the local navigation coordinate system. The two objects are now transformed into the local navigation coordinate system (blue). Thus, both the implicit emergency trajectory and the objects are in the same coordinate system and can be used for further collision checks.

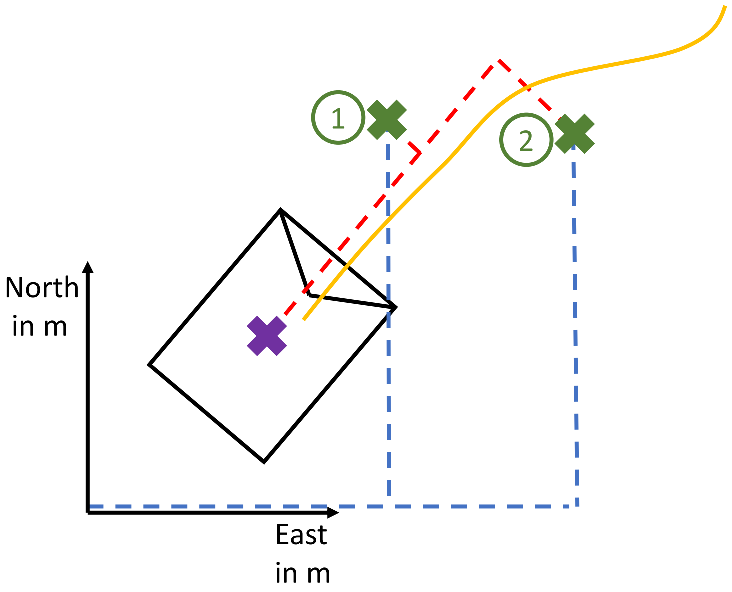


Figure 9-3: Object State Transformation to Local Navigation Coordinate System

For filter stage two, the dynamic states of all objects are examined. If the object is static, the shortest squared distance between the reference point of the object and the entire implicit emergency trajectory is calculated. In this case, the implicit emergency trajectory is considered only as a one-dimensional path. Suppose even the shortest squared distance between the static object and the implicit emergency trajectory is greater than a distance $d_{\mathrm{stat}}$. In that case, the corresponding object is irrelevant and is thus filtered out. Two objects classified as static can be seen on Fig. 9-4. The minimum distance to the one-dimensional implicit emergency trajectory for each static object is calculated. The object is still relevant if this distance is less than $d_{\text{stat}}$. If the distance is greater, the object is filtered out. In this example, Object 1 is filtered out, and Object 2 remains relevant.

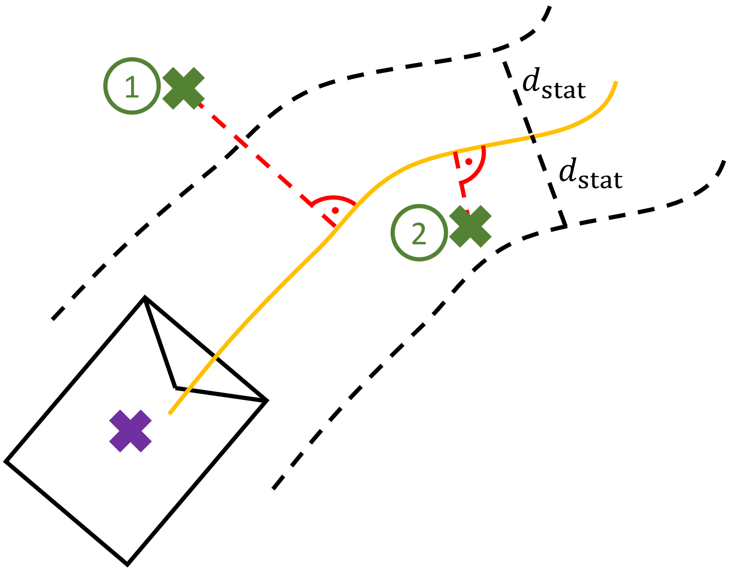


Figure 9-4: Filter Stage 2: Distance to Static Objects

In the third filter stage, the remaining static objects and all dynamic objects are checked in detail for collisions. For this purpose, the extent of the objects and the implicit emergency trajectory are used. The implicit emergency trajectory and the objects on the object list are considered in two dimensions. The width of the driving corridor depends on various influences. Since the vehicle has the sideslip angle as an additional degree of freedom, the first influence is the sideslip angle at the vehicle position $s$ on the implicit emergency trajectory. To calculate the sideslip angle, the velocity vector is used as a tangent to the position course of the trajectory. In addition, the vehicle orientation is calculated at this position. The difference between the angles of these two vectors is the sideslip angle $\beta$. The distances of the vehicle corners from the vehicle reference point are known due to the vehicle parameter control input. Using the sideslip angle, a rotation of these vehicle corners is performed. On Fig. 9-5, the vehicle without a sideslip angle is shown dashed in pink. The yaw angle at the vehicle position rotates the vehicle (black rectangle). This rotation causes the diagonal vehicle corners to diverge from the implicit emergency trajectory. The result is a widened driving corridor occupied by the vehicle. The sideslip angle $\beta$ and the driving corridor width thus depend on the position $s$. The UNICAR*agil* vehicle can be operated with a sideslip angle of 90°. With this, the driving corridor thus has the vehicle length as the driving corridor width.

The second influence on the width of the driving corridor is vehicle speed. The higher the speed, the greater the speed-dependent width addition to the driving corridor. The third influence is a safety factor. The resulting driving corridor width of the first two influences is multiplied by this factor. The extent of the objects is determined by the data about the length and width of

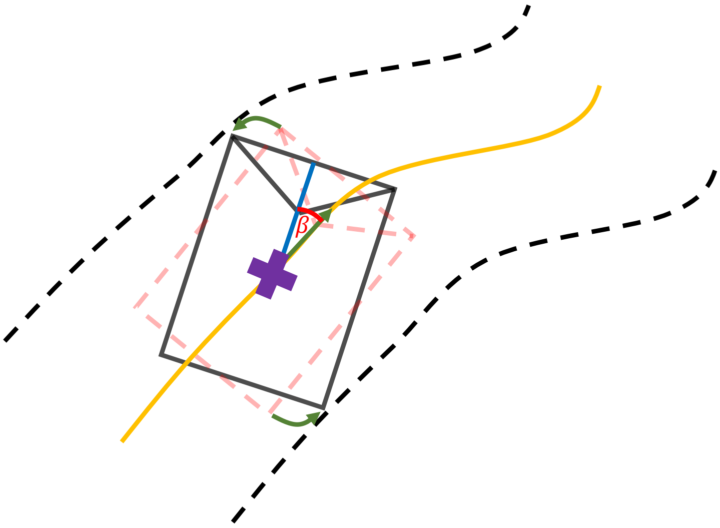


Figure 9-5: Sideslip Angle Influence on Driving Corridor Width

the objects from the world model, as well as their orientation. The standard deviation for the object positions additionally allows to enlarge the object polygons to account for this inaccuracy. The technical documentation of the radar sensors does not indicate where the reference point is located on the extended object rectangle. Therefore, it is assumed that the reference point is located at the center of the shortest side closest to the AV. It is the back center for departing traffic, and for oncoming traffic, its the front center of the extended object.

Finally, each object is checked for overlap with the generated driving corridor. Objects that do not overlap the driving corridor are filtered out. The overlap polygon is calculated if objects overlap with the implicit emergency trajectory. Fig. 9-6 shows the vehicle, the implicit emergency trajectory, the driving corridor, and two objects. Object 1 does not overlap the driving corridor of the implicit emergency trajectory, whereas Object 2 does. The resulting overlap polygon is shown in purple.

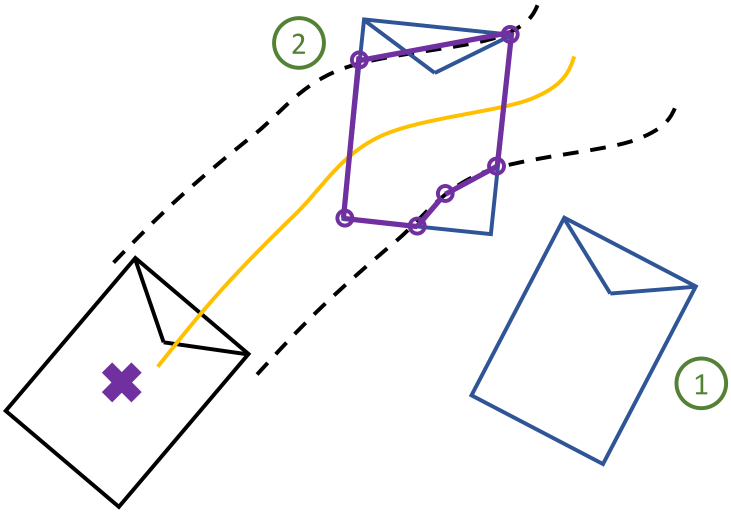


Figure 9-6: Collision Check with Overlap Polygon

As a result of the collision check, there shall not only be the information that a collision with an object will occur but also at which path position $s_{\text{obj}}$ this collision will occur and which speed $v_{\text{obj}}$ the object has. Fig. 9-7 shows a detailed view of the collision check. An object overlaps the driving corridor of the implicit emergency trajectory. An overlap polygon in purple is formed. For an overlap polygon with $n$ corner points, $n$ path position $s_{\text{poly,intersec},i}$ are thus calculated. The location with the least distance on the one-dimensional implicit emergency trajectory is searched from each corner point of the overlapping polygon. For each corner polygon, this gives an $s$ on the implicit emergency trajectory. Relevant for collision avoidance is the point with the smallest $s_{\text{poly,intersec}}$. In the figure, this is drawn as a green square. For the velocity calculation, the determined absolute velocity of the object is used. It is shown as a gray vector. Subsequently, the implicit emergency trajectory slope at the position $s_{\text{poly,intersec}}$ is calculated. Together with the absolute velocity, the velocity tangent $v_{\text{obj}}$ of the object at position $s_{\text{poly,intersec}}$ is calculated. A collision occurs when the vehicle reaches this object $s_{\text{poly,intersec,min}}$. For this reason, a time gap-dependent distance is added.

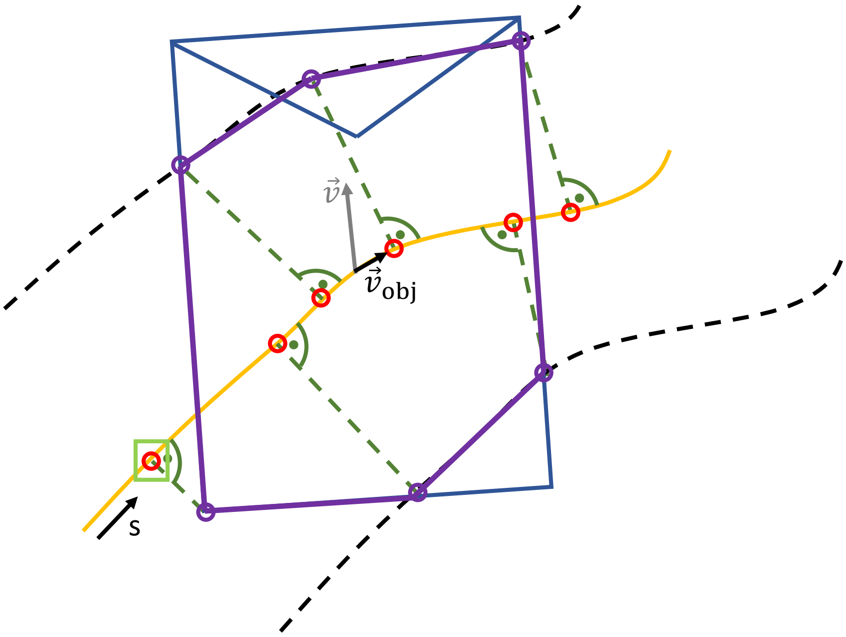


Figure 9-7: Collision Check with Overlap Polygon in Detail

With this concept, the requirements Req. B13- 1, Req. B13- 4, Req. B13- 5, Req. B13- 6, Req. B13- 10, Req. B13- 13, Req. U17- 1 and Req. U17- 2 are satisfied.

### 9.3.2 Velocity Profile Update

After the state of the collision object on the implicit emergency path has been calculated, the velocity profile of the trajectory is updated. *Safe Halt* is a safety function. The highest priority is therefore given to a deterministic behavior of all functions. This also applies for updating the velocity profile.

In Fig. 9-8, the original velocity profile is plotted with black crosses. For this purpose, the velocity samples were squared. Also shown is the interpolation result of the original velocity profiles. The black curve represents this result. Depicted as a green cross is the current position $s$ and the current velocity $v^2(s)$ of the AV. The position $s_{\mathrm{obj}}$ and the tangential squared velocity $v_{\mathrm{obj}}$ of the object identified in the previous step on the implicit emergency path is indicated as a red cross. The original velocity profile will now be adapted to avoid a collision with the object. The necessary deceleration $\ddot{s}_{\mathrm{nec}}$ in $\mathrm{m\,s^{-2}}$ of the AV for collision avoidance can be calculated with

$$\ddot{s}_{\mathrm{nec}} = \frac{1}{2} \cdot \frac{v_{\mathrm{ego}}^2 - v_{\mathrm{obj}}^2}{s_{\mathrm{ego}} - s_{\mathrm{obj}}}. \tag{9-5}$$

Using the slope triangle illustrated in the figure, the slope of the line connecting the state of the AV and the object on the implicit emergency trajectory is calculated. This slope corresponds to twice the necessary deceleration $\ddot{s}_{\mathrm{nec}}$.

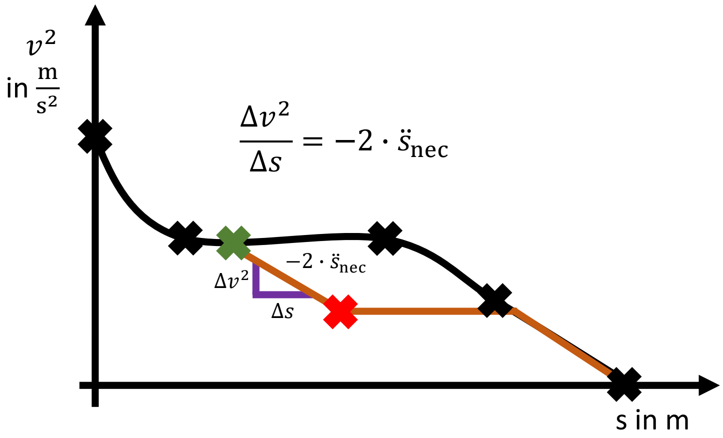


Figure 9-8: Velocity Profile with Object State

The velocity profile adaption is performed for each new world model available. At the beginning of the adaption, the state of the AV is determined. For this purpose, the velocity profile of the previous implicit emergency trajectory and the relative time since the last velocity profile generation is used.

A decision tree is developed to update the velocity profile. This tree is illustrated in Fig. 9-9. Example velocity profiles for the different cases of the decision tree can be found in Annex I. This decision tree is used to determine the squared velocity samples to be included in an adapted velocity profile. The first squared velocity sample is always the current state of the AV. As a result of this step, all velocity samples up to the next vehicle standstill are known. These samples are then interpolated to obtain a reconstructed location-dependent velocity profile.

The first decision is to check whether an object on the implicit emergency trajectory has been identified or not. If an object has been identified, different cases have to be considered. The cases

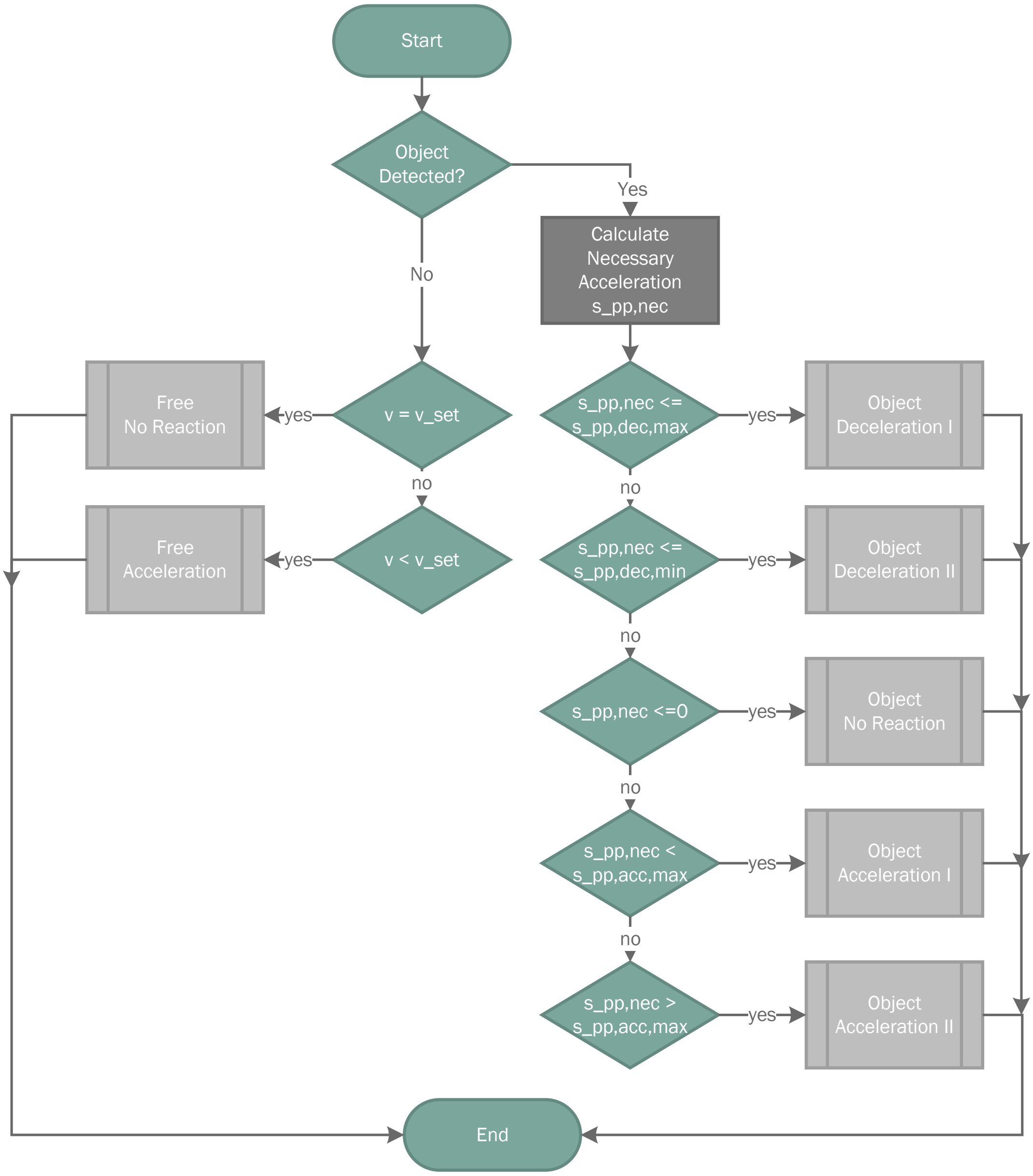


Figure 9-9: Decision Tree for Adapting the Velocity Profile

depend on the acceleration necessary to avoid collision with the object. A collision is avoided if the vehicle maintains the velocity profile adapted to the identified object. Therefore, Equ. 9-5 is used to calculate the necessary acceleration. The obtained acceleration can be both positive and negative. Following this calculation, various cases are considered.

### Object Deceleration I

In the first case, the necessary deceleration is smaller than the parameterizable maximum deceleration allows:

$$\ddot{s}_{\text{nec}} < \ddot{s}_{\text{decel,max}} \tag{9-6}$$

This parameter is used to limit the maximum deceleration in order to avoid the occurrence of phantom braking, which can lead to unstable vehicle behavior or even vehicle rollover. So instead of adapting the velocity profile to the necessary deceleration, a new entry in the velocity profile is calculated using the maximum deceleration. This entry leads to a collision in the case of a correct-positive object state.

### Object Deceleration II

In the second case, the necessary deceleration is within the minimum and maximum parameterizable deceleration range.

$$\ddot{s}_{\text{nec}} < \ddot{s}_{\text{decel,min}} \tag{9-7}$$

The minimum deceleration is the necessary deceleration at which a reaction to the object on the implicit emergency trajectory has to begin. This setting prevents the vehicle from reacting to an object at a big distance but with a minimum deceleration. If the vehicle is very slow, the distance between the vehicle and the object for a deceleration with minimum deceleration is only short. The vehicle thus approaches the object very slowly until the necessary deceleration is reached and then abruptly sets the necessary deceleration. This leads to uncomfortable vehicle behavior. In addition to the necessary deceleration, a distance parameter is therefore introduced.

$$\Delta s <= \Delta s_{\text{react}} \tag{9-8}$$

The reaction to the object thus begins either when the minimum necessary deceleration is reached or a minimum reaction distance is reached.

### Object No Reaction

In the third case, the acceleration is negative or non-existent, and the vehicle is not within the reaction distance.

$$\ddot{s}_{\text{nec}} <= 0\,\text{m}\,\text{s}^{-2} \tag{9-9}$$

or

$$\Delta s > \Delta s_{\text{react}} \tag{9-10}$$

In this case, the object is not further included in the speed profile update but is ignored instead.

### Object Acceleration I

The necessary acceleration is smaller than the maximum parameterized acceleration in the following case.

$$\ddot{s}_{\text{nec}} < \ddot{s}_{\text{acc,max}} \tag{9-11}$$

In addition, the velocity of the vehicle in the original velocity profile is calculated at the location of the object. If this original velocity is smaller than the object's velocity, the original velocity is inserted into the updated speed profile at the object's location. If the opposite is the case, the object state is inserted.

### Object Acceleration II

In the last case with an identified object, the necessary acceleration is greater than the parameterized maximum acceleration.

$$\ddot{s}_{\text{nec}} > \ddot{s}_{\text{acc,max}} \tag{9-12}$$

The maximum acceleration prevents the vehicle from following a fast-accelerating vehicle with the same acceleration. In this case, substitute quantities are calculated. For this purpose, the position is searched where the vehicle reaches the object's speed with the parameterized maximum acceleration. These substitute variables are then inserted into the updated velocity profile.

### Free No Reaction

If it was recognized at the beginning of the decision tree that no object had been detected on the implicit emergency trajectory, two cases are considered. In the first case, the vehicle has the speed specified by the original speed profile.

$$v_{\mathrm{AV}}(s) \approx v_{\mathrm{set}}(s) \tag{9-13}$$

In this case, the speed profile is updated so that all remaining implicit emergency trajectory entries are used for the updated speed profile.

### Free Acceleration

In the second case, the vehicle is slower than it should be according to the original speed profile.

$$v_{\mathrm{AV}}(s) < v_{\mathrm{set}}(s) \tag{9-14}$$

This situation can occur, for example, because an object was present and caused the vehicle to react but left the driving corridor again. In this case, the original speed profile is used to return to the original speed using the maximum acceleration. This new element is then inserted into the updated speed profile.

With this concept, the requirements Req. B13- 2, Req. B13- 7, Req. B13- 8, Req. B13- 9, Req. B13- 10 and Req. B13- 13 are met.

## 9.3.3 Trajectory Generation

Up to this step, the interpolated poses of the implicit emergency trajectory and the adapted velocity profile are available. The trajectory to be generated by *Safe Halt* (cf. Annex B) is an explicit trajectory. It contains the reference states sampled in time. This temporal sampling of the trajectory based on the interpolated poses and the adapted velocity profile is done by transforming it with Equ. 7-3. The velocity samples are interpolated cubically by Equ. 7-29.

This means for the time dependency

$$\int_{t_0}^{t} \mathrm{d}t = \int_{s_0}^{s} \frac{\mathrm{d}s}{\sqrt{\mathrm{C}_{3,\mathrm{v},\mathrm{pathl},i}(s-s_i)^3 + \mathrm{C}_{2,\mathrm{v},\mathrm{pathl},i}(s-s_i)^2 + \mathrm{C}_{1,\mathrm{v},\mathrm{pathl},i}(s-s_i) + \mathrm{C}_{0,\mathrm{v},\mathrm{pathl},i}}}, \tag{9-15}$$

and thus

$$t(s) = t_0 + \int_{s_0}^{s} \frac{\mathrm{d}s}{\sqrt{\mathrm{C}_{3,\mathrm{v,pathl},i}(s-s_i)^3 + \mathrm{C}_{2,\mathrm{v,pathl},i}(s-s_i)^2 + \mathrm{C}_{1,\mathrm{v,pathl},i}(s-s_i) + \mathrm{C}_{0,\mathrm{v,pathl},i}}}. \tag{9-16}$$

To solve the equation, $t_0 = 0$ is set. Therefore, for a time $t$, the position $s$ is identified that satisfies the equation

$$0 = \int_{s_0}^{s} \frac{\mathrm{d}s}{\sqrt{\mathrm{C}_{3,\mathrm{v,pathl},i}(s-s_i)^3 + \mathrm{C}_{2,\mathrm{v,pathl},i}(s-s_i)^2 + \mathrm{C}_{1,\mathrm{v,pathl},i}(s-s_i) + \mathrm{C}_{0,\mathrm{v,pathl},i}}} - t. \tag{9-17}$$

The solution to this problem is not necessarily supposed to be on the same segment. It can also lie in a subsequent segment. Therefore, an additional check has to be made to see if the solution is already found in the same segment. If not, the time to the end of the current segment is subtracted from $t$, and the solution search is continued in the next segment. Further iterations are performed until the solution is found.

To solve this equation, a minimization problem is formulated. Therefore, the minimization problem

$$0 = \left\| \int_{s_0}^{s} \frac{\mathrm{d}s}{\sqrt{\mathrm{C}_{3,\mathrm{v,pathl},i}(s-s_i)^3 + \mathrm{C}_{2,\mathrm{v,pathl},i}(s-s_i)^2 + \mathrm{C}_{1,\mathrm{v,pathl},i}(s-s_i) + \mathrm{C}_{0,\mathrm{v,pathl},i}}} - t \right\|_2^2, \tag{9-18}$$

arises. This equation corresponds to a root finding. The Jacobian matrix of this equation can be calculated analytically. In this case, it is

$$J_f(s) = \frac{1}{\sqrt{\mathrm{C}_{3,\mathrm{v,pathl},i}(s-s_i)^3 + \mathrm{C}_{2,\mathrm{v,pathl},i}(s-s_i)^2 + \mathrm{C}_{1,\mathrm{v,pathl},i}(s-s_i) + \mathrm{C}_{0,\mathrm{v,pathl},i}}}. \tag{9-19}$$

Various approaches exist for solving minimization problems. One approach used in stochastic for nonlinear regression is the least squares method. This balancing calculation is used to calculate unknown parameters in balancing formulas. Possible algorithms for solving this optimization problem are the Gauss-Newton and the Levenberg-Marquardt. The Levenberg-Marquardt is more robust and is therefore selected to solve the minimization problem. This algorithm can be used to solve minimization problems of the type

$$\min[f_1^2 + f_2^2 + \cdots + f_m^2] \tag{9-20}$$

with $f_i = f_i(x_1, \ldots, x_n)$. Applied to this case, the problem $\min[f_1^2]$ is thus to be solved with $f_1 = f_1(s)$. In each algorithm step $k$, the Jacobian matrix is computed at the current value for $s_k$. It is calculated analytically by Equ. 9-19. This avoids the numerical calculation of the Jacobian matrix and thus increases the efficiency of the calculations. Applied to this case, the formula is thus

$$f_1 = \int_{s_0}^{s} \frac{\mathrm{d}s}{\sqrt{\mathrm{C}_{3,\mathrm{v,pathl},i}(s-s_i)^3 + \mathrm{C}_{2,\mathrm{v,pathl},i}(s-s_i)^2 + \mathrm{C}_{1,\mathrm{v,pathl},i}(s-s_i) + \mathrm{C}_{0,\mathrm{v,pathl},i}}} - t. \tag{9-21}$$

The algorithm also allows setting upper and lower limits for the solution space. Since the algorithm searches for slopes, the lower limit cannot be set to $s_0 = 0$. However, if the vehicle is at a standstill, there is a singularity at $s_0 = 0$, so the calculations for the integral fail. Therefore, the solution space is set as $s \in [0.001, +\infty]$. By setting this space, the solution is searched only within these values. The integral cannot be determined analytically. For this reason, the integral

$$\int_{s_0}^{s} \frac{\mathrm{d}s}{\sqrt{\mathrm{C}_{3,\mathrm{v,pathl},i}(s-s_i)^3 + \mathrm{C}_{2,\mathrm{v,pathl},i}(s-s_i)^2 + \mathrm{C}_{1,\mathrm{v,pathl},i}(s-s_i) + \mathrm{C}_{0,\mathrm{v,pathl},i}}}, \tag{9-22}$$

at each $s_k$ during the execution of the minimization algorithm shall be calculated numerically. Since the function is in the denominator of a fraction, the occurrence of singularities shall be expected. One method for numerical integration is the Gauss-Kronrod quadrature formula. Gauss-Kronrod quadrature is a general-purpose routine and can handle singularities arising from the inverse functions. Combining the Levenberg-Marquardt algorithm with the Gauss-Kronrod quadrature formula, the corresponding position $s$ can be calculated for each time $t$.

For the trajectory, the time-sampled trajectory elements are calculated. Starting at $t = 0$, the temporally equidistant emergency trajectory time stamps are determined. The path position $s$ is calculated using the above procedure for each of these timestamps. For the UNICAR*agil* application $\Delta t = 0.1\,\mathrm{s}$ and the emergency trajectory thus has 50 elements. After this step, all 50 positions $s$ of the time sampled trajectory are available.

Those position values $s$ are now used to calculate $\dot{s}(s)$ and $\ddot{s}(s)$, since both are needed according to Tab. 7-2 and Tab. 7-3

For $v(s)$ it is

$$v(s) = \dot{s}(s) = \sqrt{v^2(s)}. \tag{9-23}$$

The acceleration $a(s)$ is calculated with

$$a(s) = \ddot{s}(s) = \frac{\mathrm{d}v}{\mathrm{d}t} = \frac{\mathrm{d}v}{\mathrm{d}s}\frac{\mathrm{d}s}{\mathrm{d}t} = v\frac{\mathrm{d}v}{\mathrm{d}s} = \frac{1}{2}\frac{\mathrm{d}v^2}{\mathrm{d}s}. \tag{9-24}$$

Here $\frac{\mathrm{d}v^2}{\mathrm{d}s}$ is the slope of $v^2(s)$. Since $v^2(s) = \mathrm{C}_{3,\mathrm{v,pathl},i}(s-s_i)^3 + \mathrm{C}_{2,\mathrm{v,pathl},i}(s-s_i)^2 + \mathrm{C}_{1,\mathrm{v,pathl},i}(s-s_i) + \mathrm{C}_{0,\mathrm{v,pathl},i}$, the following applies

$$a(s) = \ddot{s} = \frac{1}{2}(3\cdot\mathrm{C}_{3,\mathrm{v,pathl},i}(s-s_i)^2 + 2\cdot\mathrm{C}_{2,\mathrm{v,pathl},i}(s-s_i) + \mathrm{C}_{1,\mathrm{v,pathl},i}). \tag{9-25}$$

Using $\dot{s}$ and $\ddot{s}$, Equ. 7-5 and Equ. 7-6 can be used to calculate the velocity and acceleration vectors in the local navigation coordinate system. Both data are required in the form of polar coordinates. For this reason, the magnitude and angle of the velocity and acceleration are calculated for each emergency trajectory element. For the calculation of the yaw angle, yaw rates, and yaw acceleration, $\dot{s}$ and $\ddot{s}$ are substituted into the Equ. 7-8 and Equ. 7-9. Finally, the trajectory is prepared and output according to U7 *Trajectory Tracking Control* interface. This concept satisfies Req. B13- 3, Req. B13- 10, Req. B13- 11 and Req. B13- 13. An extension to satisfy Req. B13- 12 is conceivable but is not pursued here.

Other outputs are the signals for activating the external HMI and the hazard warning lights. Since *Safe Halt* has no information about the current vehicle operation mode, *Safe Halt* sends these signals permanently. Analogous to U5 *Trajectory Selection*, a new block *HMI Activation Selection* is integrated into the ADS. Depending on the current vehicle operation mode, this function selects the signals for the hazard warning lights and the external HMI. Through this concept, Req. B13- 14 is fulfilled.

## 9.4 Conclusion

This chapter presents the generic and specific requirements for the *Safe Halt* function *Strategical and Tactical Planning*. The three sub-functions collision check, velocity profile update, and trajectory generation, are identified based on these requirements. For each subfunction, concepts for meeting the requirements are described. Different filtering stages are processed for the collision check to keep the calculation effort as low as possible. Using different filter criteria, irrelevant objects are filtered out, and a detailed collision check is performed only for the remaining objects. If a collision is detected, the velocity profile of the implicit emergency trajectory is updated based on the identified collision object. For this purpose, a deterministic adaptation algorithm is executed using a decision tree. Finally, the updated velocity profile is used to generate the emergency trajectories for the output of the function of *Safe Halt*. This is done by solving a minimization problem using numerical computation.

# 10 Safe Halt Self-Perception

In this chapter, the generic and UNICAR*agil* Automated Driving System (ADS) specific requirements and concepts for the *Safe Halt Self-Perception* are introduced. The generic functional block B14 is presented on Fig. 5-2.

The function receives the status of all functional nodes of *Safe Halt* as input. The function combines all those statuses to the overall status of *Safe Halt* and outputs it. A time reference is provided as control, and computation time as a mechanism for the function.

This chapter starts with a description of generic requirements in Sec. 10.1. Subsequently, requirements specific to the UNICAR*agil* ADS are added in Sec. 10.2. The concepts for the UNICAR*agil* ADS are presented in Sec. 10.3. Finally, the chapter is summarized in Sec. 10.4.

## 10.1 Requirements for the Safe Halt Self-Perception in a Generic ADS

*Safe Halt* shall communicate its overall status to B6 *ADS Monitoring*. According to modular software architecture, *Safe Halt* shall not know the current vehicle operation mode. This means that the overall status of *Safe Halt* cannot be determined solely based on internal information. *Safe Halt* can, therefore, neither assert that it is not operational nor that it is operational. According to the requirements, the neutral status data for each functional block of *Safe Halt* is transmitted to node B16 *Self-Perception*. The function shall synchronize this status information. This leads to the first requirement:

**Req. B14-1:** Safe Halt Self-Perception shall synchronize the status information of all Safe Halt functions

After synchronization, the gathered status information shall be transmitted to B6 *ADS Monitoring* node in Fig. 4-5. This requests the requirement:

**Req. B14-2:** Safe Halt Self-Perception shall transmit the synchronized status information of Safe Halt to ADS Monitoring

Depending on the design of the *ADS monitoring*, it shall be determined where monitoring of the determined status takes place. If a function's execution time is considered as an example, it will fluctuate due to the lack of deterministic timing behavior. However, it is to be expected that a statistical evaluation can estimate the runtimes. If a runtime deviates massively from this estimation, this can indicate a resource problem of the function. A statistical evaluation of the runtime can be done either directly by the *Safe Halt Self-Perception* or by the *ADS Monitoring*. This statistical evaluation can also be extended, for example, to the transmission times via the interfaces. This evaluation can reveal deviations in the transmission time of implicit emergency trajectories and thus indicate a problem with the communication architecture of the ADS. If the statistical evaluation already takes place in the *Safe Halt Self-Perception*, different quality classes (e.g., failed, bad, default, good) can be transmitted to the *ADS Monitoring* instead of the timestamps as the result of the evaluation. Therefore follows the requirement:

**Req. B14-3:** For Safe Halt Self-Perception, it shall be defined whether a statistical evaluation of the status data takes place in the function or the superordinate system

The typical execution time to execute the function is less than the targeted cycle time. While waiting for a new cycle, computing power consumption shall be minimized. For this purpose, the function shall have a function to check for a new cycle and otherwise minimize its resource consumption. This leads to the requirement:

**Req. B14-4:** Safe Halt Self-Perception shall start with its computation when a new cycle is due. When the computations are done, and there is no new cycle due, it shall minimize computation power usage

## 10.2 Specific Requirements for the Safe Halt Self-Perception in the UNICAR*agil* ADS

This section describes the requirements for *Safe Halt Self-Perception* for the UNICAR*agil* ADS. The functional block U18 is illustrated on Fig. 10-1.

The UNICAR*agil* ADS has with node U9 *Self-Perception* depicted on Fig. 4-6 an implementation of the generic node B6 *ADS Monitoring*. The status information from node U18 *Safe Halt Self-Perception* shall be provided to this node and updated periodically. The frequency for this update shall be sufficiently high so that the ADS does not face critical statuses. The frequency of transmission of the status information is To Be Determined (TBD), for the *Safe Halt* at hand a frequency of at least 10 Hz is chosen, which leads to the requirement:

**Req. U17-1:** Safe Halt Self-Perception shall synchronize and transmit the Safe Halt status information to U9 Self-Perception with a frequency of at least 10 Hz

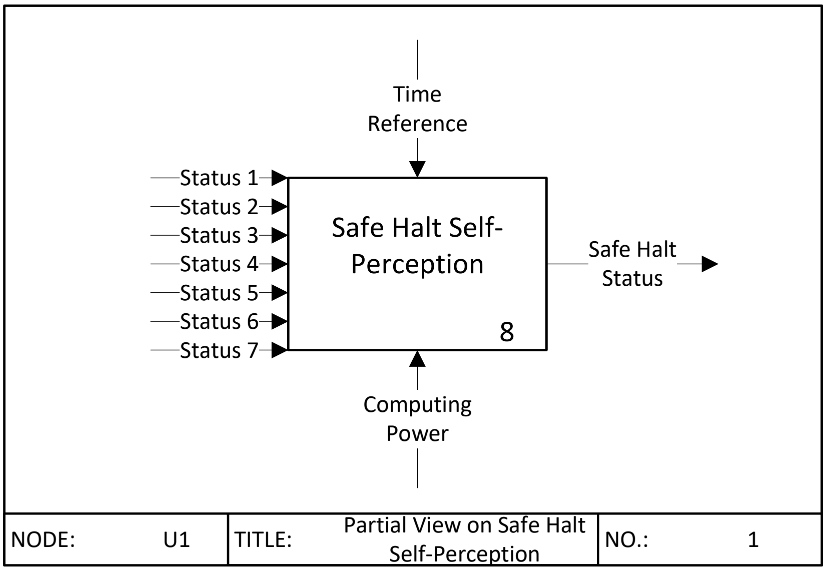


Figure 10-1: Safe Halt Self-Perception in the UNICAR*agil* ADS

## 10.3 Concept for the Safe Halt Self-Perception in the UNICAR*agil* ADS

As a concept for U18 *Safe Halt Self-Perception* in the UNICAR*agil* ADS, the raw status information of all *Safe Halt* functional blocks are retrieved and processed with at least 10 Hz. This processing is executed by generating quality classes. For this purpose, the raw status information is compared with the class requirements. If the raw status information is within the values of a class, this class is assigned. Subsequently, all determined classes are output to U9 *Self-Perception* of the UNICAR*agil* ADS. This concept satisfies Req. B14- 1, Req. B14- 2, Req. B14- 4 and Req. U17- 1.

## 10.4 Conclusion

This chapter identifies the generic and specific requirements for *Safe Halt Self-Perception*. Due to the service-based software architecture, the function cannot evaluate whether it is faulty or intentional for each status. Therefore, the function should transmit neutral information to a superordinate system, function B6 *ADS Monitoring* or U9 *Self-Perception*. Only these functions can evaluate the status of the Automated Vehicle (AV) in the context of the vehicle operation mode and the status of the other functional nodes of the ADS. For the UNICAR*agil* ADS, the *self-perception* of the UNICAR*agil* ADS is sent the status information of *Safe Halt* with at least 10 Hz.

# 11 Synthesis of Safe Halt Sub-functions and Implementation for the UNICAR*agil* ADS

In this chapter, the synthesis of the sub-functions of the *Safe Halt* to the reference solution for the UNICAR*agil* ADS is described in Sec. 11.1. Subsequently in Sec. 11.2, the implementation of the reference solution is addressed. Finally, the chapter is summarized in Sec. 11.3.

## 11.1 Synthesis of the Safe Halt Reference Architecture

After describing the sub-functions of *Safe Halt* in the previous chapters, this section presents the result of the functional synthesis of these subfunctions. It is essential to avoid mutual interference of the subfunction capabilities by the synthesis. As a reference for the solution of *Safe Halt* in UNICAR*agil* ADS, the architecture depicted on Fig. 11-1 is proposed.

## 11.2 Implementation of the Safe Halt Reference Solution

The hardware used for the calculations is selected at the beginning of implementing the Safe Halt solution. Within the project UNICAR*agil*, an Electronic Control Unit (ECU) architecture based on three levels is used. The first and most powerful level is the so-called cerebrum level. On this level, Linux-based standard computers with graphic card support are used. As a second level, the brainstem is integrated. The brainstem is embedded hardware with an embedded operating system. An AMD Xilinx Multiprocessor System on a Chip (MPSoC) is used in the project context. The Zynq UltraScale+ EG[114] model offers both a non-real-time capable processor core and a real-time capable processor.

The non-real-time capable processor is a quad-core arm Cortex-A53, while the real-time capable chip is a dual-core arm Cortex-R5F. The arm Cortex-A53 is a 64-bit multicore processor from 2012 that implements the Armv8 instruction set. The *Safe Halt* solution generates trajectories and provides them to *Trajectory Tracking Controlling*. *Safe Halt* is not required to operate deterministically in time. All sub-functions of *Safe Halt* are implemented as threads. The parallel computation of the functions in threads can thus be distributed to different processor cores.

[114] Advanced Micro Devices, Inc: Zynq UltraScale+ MPSoC (2022).

[115] AMD Xilinx: Zynq UltraScale+ EG Figure (2022)

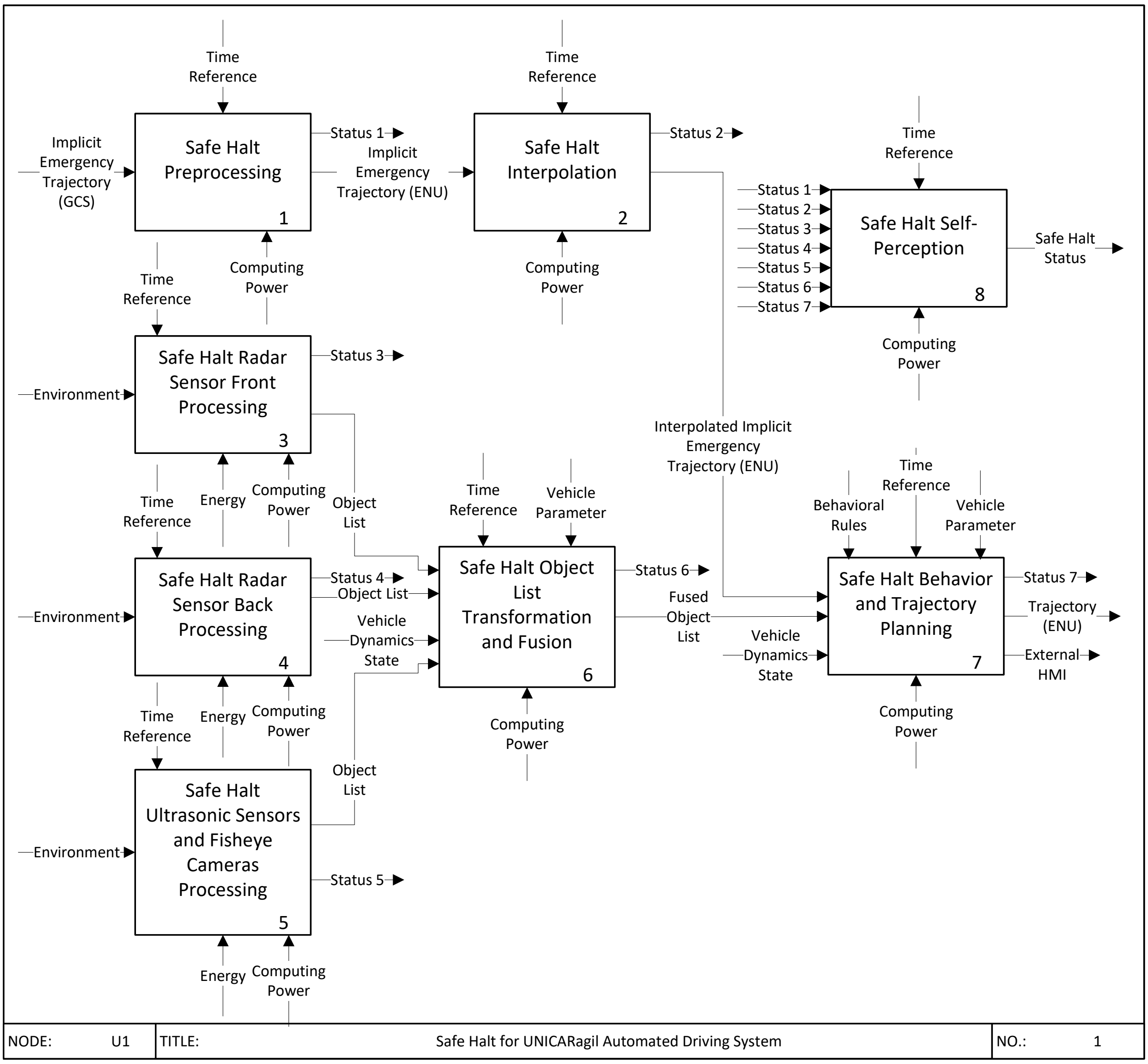


Figure 11-1: Safe Halt Reference Solution for the UNICAR*agil* ADS

The arm Cortex-A53 is a quad-core processor and is, therefore, suitable for the thread-based implementation of the *Safe Halt* functions.

Xilinx offers the Petalinux Tools[116] for the Zynq UltraScale+ EG. These tools can configure, build and install specific embedded Linux solutions. Petalinux is an embedded Linux with a command line interface without a graphical user interface. The middleware Robot Operating System (ROS) is used in applications for automated driving. However, ROS only supports the operating systems Ubuntu and Debian. Windows 10 is supported experimentally. ROS 2 is in development. This operating system version offers micro-ros, an implementation also for embedded hardware. However, the Xilinx hardware is not yet officially supported[117]. The *Safe Halt* reference solution is implemented in the programming languages C++ and parts also in C.

[116] Xilinx: PetaLinux Tools (2022).

[117] micro-ROS: micro-ROS (2022).

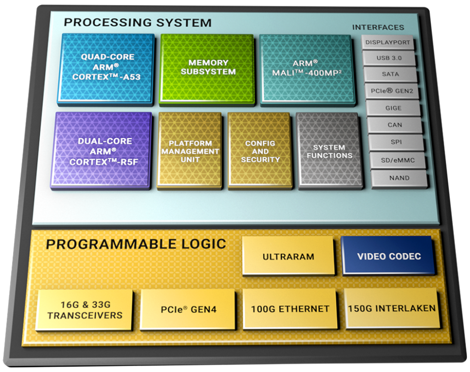


Figure 11-2: AMD Xilinx Zynq UltraScale+ EG Platform[115]

An Automotive Service-Oriented Software Architecture (ASOA)-based reference solution of the Safe Halt functions is implemented for the arm Cortex-A53 processor.

### 11.2.1 Software in the Loop (SiL) Test Environment

For the Software in the Loop (SiL) test environment, *Safe Halt* is integrated into an IPG CarMaker™[118] simulation environment. IPG CarMaker™ is a simulation software that can equip virtual vehicle prototypes with automated driving functions. Using this simulation, these prototypes can be used to establish a test environment for the development phase of the function. The *Safe Halt* functions are integrated into the CarMaker™ subsystem *Vehicle Control*. Two different options for testing the *Safe Halt* functions are implemented. The first option includes the two vehicle operating modes *Automated Driving* and *Safe Halt*. Preplanned trajectories and implicit emergency trajectories can be transmitted into the simulation environment. Vehicle operating mode switching can then be initiated during the simulation using trigger criteria. For the second option, the vehicle is permanently in *Safe Halt* mode. This option is used when only the *Safe Halt* functions are tested, but not the switching of vehicle operating mode. The virtual vehicle prototypes are equipped with simulated environment sensors to perceive the simulated vehicle environment. The environment sensors include two radar sensors, 24 ultrasonic sensors, and four cameras. These sensors are mounted to the virtual prototype using the same vehicle-fixed poses as the sensors in the real UNICAR*agil* vehicle. For the two radar sensors, IPG CarMaker™ radar sensor models are used. The 24 ultrasonic sensors and the four cameras are integrated as IPG CarMaker™ object sensors. All environment sensors are configured according to their data sheets. This includes the aperture angle of the sensor field of view, the accuracy, the resolution, and the positioning and orientation in the vehicle.

[118] IPG Automotive: CarMaker (2022).

### 11.2.2 Hardware in the Loop (HiL) Test Environment

The Hardware in the Loop (HiL) test environment consists of a simulation computer running the simulation environment IPG CarMaker™ and the brainstem hardware AMD Xilinx Zynq UltraScale+ EG. The Xilinx ECU executes the *Safe Halt* functions as an ASOA service. The CarMaker™ simulation environment is extended by an ASOA interface. A proxy ASOA service is implemented to generate implicit emergency trajectories. This service generates these test trajectories based on preplanned poses. In addition, the current vehicle pose is obtained from the simulation environment. Using this current pose, the service generates an implicit emergency trajectory originating in this pose. *Safe Halt* can be tested without the original implicit emergency trajectory planner with this approach. The real-time capable processor executes the Trajectory Tracking Controlling. This controlling generates actuation setpoints for the vehicle's dynamics actuators based on the input trajectory. These actuation setpoints are transmitted back to the simulation environment via an ASOA interface, where they control the virtual vehicle dynamics actuators and thus move the vehicle through the simulation environment. This setup can be used to perform execution time tests of the *Safe Halt* functions. Since *Safe Halt* uses sensor data, a real sensor setup connected to the HiL would have to perceive an environment as in the virtual environment. In this thesis, this approach is not pursued further. Therefore, only the functionality of *Safe Halt* without environment perception is tested in HiL. A possible alternative is the addition of a new ASOA interface for transmitting virtual sensor data to *Safe Halt* in HiL. This interface would only be needed for the HiL, but not for the real vehicle prototype. This additional interface may have unexpected influences on the functions of *Safe Halt*. Therefore it is not implemented. This HiL is therefore limited to tests without the use of the environment sensors.

### 11.2.3 UNICAR*agil* Vehicle Prototype Test Environment

The test vehicles from the UNICAR*agil* project are used for the final implementation. To do this, the *Safe Halt* software is installed on the Xilinx ECU. All implementations of the hardware and software components of nodes U5, U6, U7, and U8 depicted on Fig. 4-6 are required for *Safe Halt* operation. In addition, all Control (Behavioral Rules, Time Reference, and Vehicle Parameter) and Mechanism (Energy and Computing Power) have to be available for the *Safe Halt* function. The environment sensors for *Safe Halt* are integrated into these test vehicles and connected both mechanically and electrically to the vehicle prototype.

## 11.3 Conclusion

This chapter synthesizes the subfunction of the reference solution of *Safe Halt* to the final solution. The synthesis shall not impact the individual capabilities of the sub-functions. The implementation of the reference solution uses embedded hardware as the mechanism for the *Safe*

*Halt* functions. This reference solution is tested in various test environments. These include SiL, a HiL using the embedded hardware platform, and as an application in the four prototype vehicles of the UNICAR*agil* project. The reference solution is successfully integrated into all test environments.

# 12 Verification of the Safe Halt Reference Solution

The *Safe Halt* solution presented in the previous chapter is verified in this chapter. In Sec. 12.1, test cases for the verification are described. This is followed by applying and evaluating the test cases in Sec. 12.2. The chapter ends with a conclusion in Sec. 12.3.

The requirements from Chap. 5 are used for this purpose. At least one test criterion is specified for each requirement.

The solution for *Safe Halt* is an emergency trajectory source. Thus, *Safe Halt* generates the desired behavior of the vehicle. The actual vehicle behavior may deviate from this desired behavior. Faults in vehicle dynamics control, vehicle dynamics state estimation, and vehicle dynamics actuation, as well as in supporting functions such as communication and power supply, can cause actual vehicle behavior to deviate from *Safe Halt*'s planned desired behavior.

Another challenge is the change of vehicle operating mode. In default operation, the vehicle is in *Automated Driving* vehicle operation mode. A mode change to *Safe Halt* is conducted if the driving mission cannot be safely continued. Therefore, it shall be checked whether a change of the trajectory sources is permissible in the current vehicle state. This check shall be performed permanently during *Automated Driving* vehicle operating mode. If a change to *Safe Halt* mode is impossible, the driving mission shall be aborted and the vehicle transitioned to the Minimal Risk Condition (MRC) in *Automated Driving* mode. According to the service orientation, the *Safe Halt* functions do not know the current vehicle operating mode. Also unknown to *Safe Halt* is the existence and number of other trajectory sources in the shared ADS. Thus, *Safe Halt* cannot check whether its planned emergency trajectory matches the currently used trajectory.

This finding allows limiting test responsibility to trajectory generation according to requirements and collision avoiding response to objects on the implicit emergency trajectory.

## 12.1 Safe Halt Test Cases

Test cases are used to test the *Safe Halt* functions. These can be used to check the performance of the *Safe Halt* system. The focus is on verifying the requirements of Chap. 5. For the requirements of the sub-functions of Safe Halt, additional subfunction-dependent verification methods are elaborated, depending on the implementation concept. A detailed presentation of these results is omitted within the scope of this thesis.

The tests take place both in simulation and in actual test drives. Since the tests all take place only on a simulated and real test site, the test results are limited only to verification of the requirements.

A downstream validation study shall include test cases also in real road traffic to ensure that the *Safe Halt* requirements list is complete.

The tests start without considering a vehicle operating mode change from *Automated Driving* to *Safe Halt*. The test vehicle is, therefore, in *Safe Halt* mode in the entire test case. The initial state of the vehicle is the standstill. This initial state is selected because dynamic states would have to be reached manually, and then a dynamic changeover would have to take place. Usually, the activation of *Safe Halt* takes place from an initial speed. Therefore, as soon as the vehicle operation mode *Automated Driving* is available, the tests shall be repeated with a driving mode change. *Safe Halt* receives a single test implicit emergency trajectory from a substitute implicit emergency trajectory planner. The spatial course of the implicit emergency trajectory can be designed arbitrarily. Only the initial velocity of the velocity profile is always a standstill. For the tests, an implicit emergency trajectory is assumed to move the vehicle to the right towards the right edge of the lane. The implicit emergency trajectory's speed profile is selected to accelerate the vehicle from the standstill to a test speed. This test speed corresponds to the activation speed of *Safe Halt*. Then the vehicle speed is quickly reduced, and the vehicle moves towards the MRC. In front of the MRC, the vehicle is then decelerated comfortably to a standstill. The courses of the poses and maximum velocity profiles are varied.

Fig. 12-1 shows an exemplary spatial implicit emergency trajectory of a real test drive. The vehicle is at position (0|0) at the start of the test drive. Illustrated as circles are the waypoints of the implicit emergency trajectory.

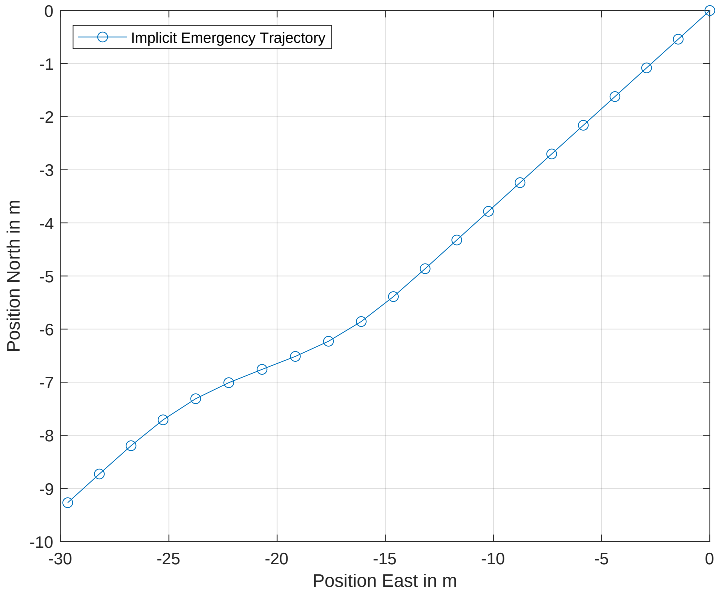


Figure 12-1: Test Implicit Emergency Trajectory Positions

Fig. 12-2 shows the yaw angle when moving on the implicit emergency trajectory. In the beginning, the vehicle is at the start of the implicit emergency trajectory. Again, circles denote the waypoints of the implicit emergency trajectory.

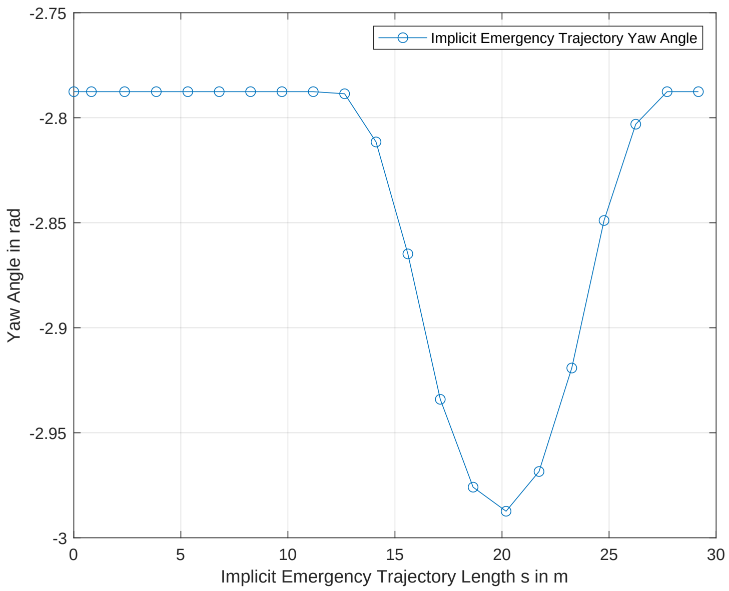


Figure 12-2: Test Implicit Emergency Trajectory Yaw Angle

Fig. 12-3 finally shows the associated maximum speed profile. Initially, the vehicle is accelerated from a standstill to the test speed. This is followed by a sharp reduction in speed. The implicit emergency trajectory is then continued at this creep speed, and the vehicle is stopped comfortably at the minimum risk location. The circle again shows the waypoints.

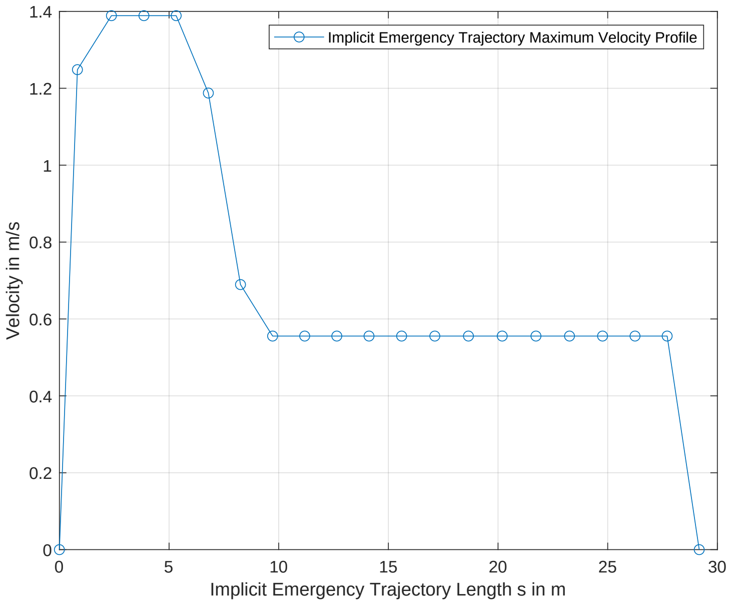


Figure 12-3: Test Implicit Emergency Trajectory Maximum Velocity Profile

*Safe Halt*'s response to emergency path-blocking static and dynamic objects is tested in the second part of the tests. Only the two radar sensors in the front and rear of the vehicle are active in these test cases. Objects outside the driving corridor of the implicit emergency trajectory should not lead to any reaction of *Safe Halt*. On the other hand, objects inside the driving corridor should lead to an object reaction of *Safe Halt*. The emergency trajectory should adapt the speed of the test vehicle to the speed of the object blocking the driving corridor to avoid a collision. *Safe Halt* is tested with dynamic objects at a standstill in the driving corridor. Various object classes were tested, including a soft target in the shape of a vehicle rear end, a BMW i3, and a UNICAR*agil* vehicle. Fig. 12-4 shows the soft target in the shape of a vehicle rear end and a BMW i3.

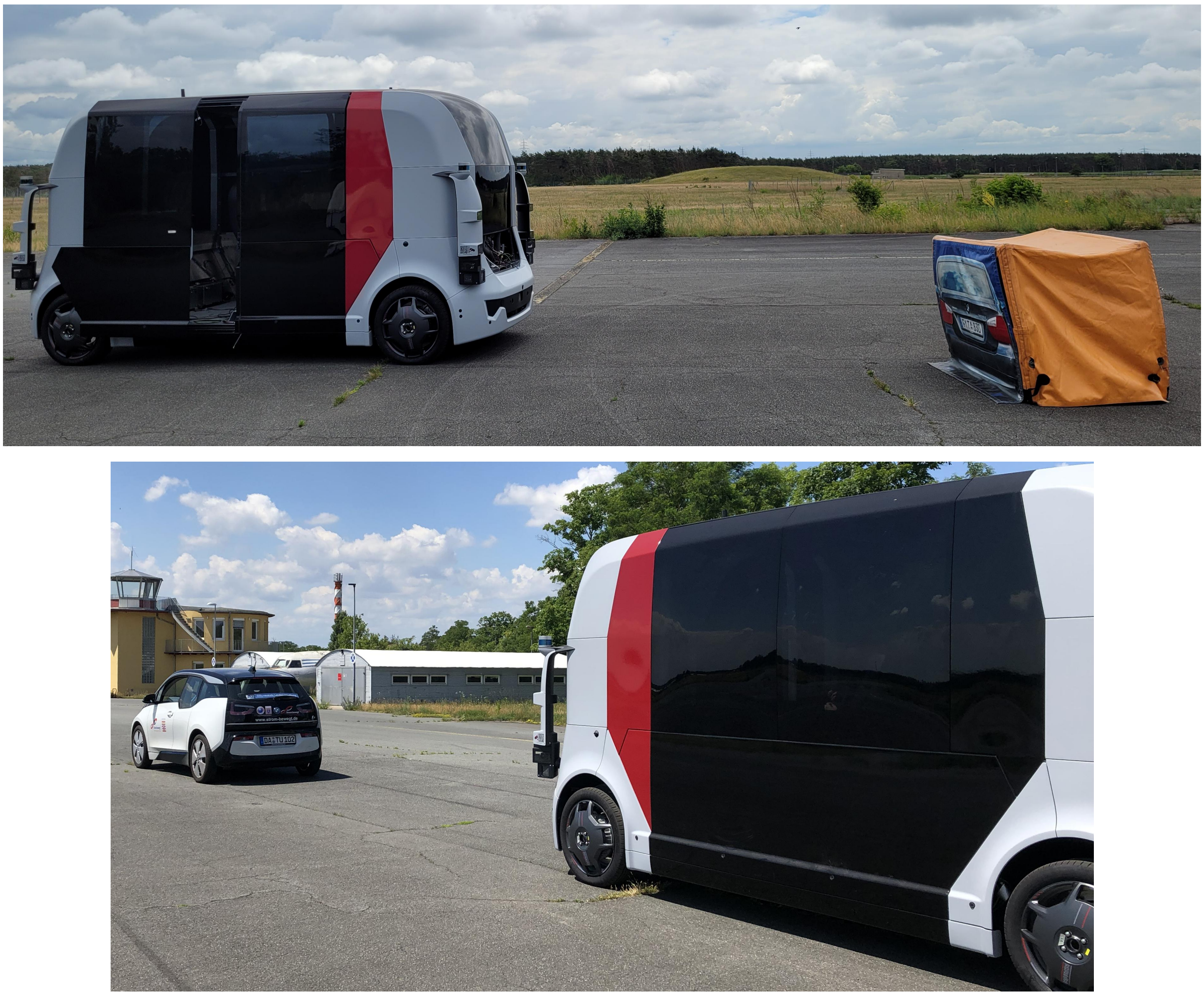


Figure 12-4: Test Collision Objects

## 12.2 Evaluation of the Safe Halt Test Cases

For the test evaluation, actual test runs are performed, and the generated output data is recorded for later evaluation. First, the *Safe Halt* solution is tested with regard to its requirements. If possible, the actual hardware and the real UNICAR*agil* test vehicle are used in the test environment. The majority of the requirements can be tested during actual tests. Basic tests regarding the response to implicit emergency trajectory-blocking objects can also be performed in actual test runs. On the other hand, more critical situations with dynamic maneuvers only take place simulatively in the IPG CarMaker™ simulation environment.

First, the requirements from Chap. 5 are considered, test criteria are derived, and the test environment is defined. Tab. 12-1 and Tab. 12-2 show the requirements, their test criteria, and the test results. The test results are discussed in more detail below.

Req. B1- 1 and Req. U1- 1 require that the *Safe Halt* solution generates new trajectories with at least 10 Hz. Therefore, the frequency of emergency trajectory generation is monitored as a test criterion, and a frequency >10 Hz is expected as a pass criterion. For this purpose, the send timestamp differences of successive emergency trajectories of Safe Halt are observed. On Fig. 12-5, a Cumulative Distribution Function (CDF) plot of the trajectory timestamp differences of one test run is presented. Most of the trajectories follow within 100 ms of a previous trajectory. Due to the asynchronous reception of object lists from the environment sensors, new emergency trajectories are computed even after a more negligible time difference. The execution time of the trajectory generation takes about 20 ms. This can be seen as the first slope in the figure. Directly after the generation of a trajectory, another new trajectory is generated. This is because, during the runtime of trajectory generation, either a new implicit emergency trajectory or a new object list arrived. The cycle time of both radar sensors is around 80 ms. This can be seen as the next slope in the figure. Finally, there is the 100 ms timeout when one of the sensors does not send new sensor data within this timeout. A few trajectory time differences are larger than the targeted 100 ms. This is to be expected since *Safe Halt* does not have real-time functionality and is not run on any real-time hardware. However, the test proves that the specification regarding the frequency of emergency trajectory generation is met. Furthermore, this demonstrates that *Safe Halt* responds event-based to input data. Thus Req. B1- 11 is also satisfied. Integrated output data checks also check the content of the trajectory for plausibility. Only checked trajectories are output.

Req. B1- 2 and Req. U1- 2 state that *Safe Halt* shall monitor its abilities and transmit it to the *Self-Perception* of the UNICAR*agil* ADS. Therefore, sending the *Safe Halt* status information is used as a criterion. Again, a frequency of 100 ms is expected for passing. In addition, the content of the status information shall be complete and include the status of the sub-functions. For this, the send timestamps of the status information to the *Self-Perception* are tested. A CDF plot is shown on Fig. 12-6. The status information is sent strictly after timeouts. There are no other triggering events. Therefore, the CDF plot shows that the time differences are almost always the targeted 100 ms. Here, integrated output data checks also check the content of the status date for plausibility. Only checked status data are output.

Table 12-1: Results of the Safe Halt Reference Solution Verification 1

| Requirement | Test Environment | Criteria | Fail/Pass |
|---|---|---|---|
| Req. B1- 1 and Req. U1- 1 | Test Vehicle | Valid Trajectories with Frequency >= 10 Hz. The tolerance is TBD | Pass |
| Req. B1- 2 and Req. U1- 2 | Test Vehicle | Valid Status Information with Frequency >= 10 Hz. The tolerance is TBD | Pass |
| Req. B1- 3 | Test Vehicle | Generated Trajectories Track Implicit Emergency Trajectory. The tolerance is TBD | Pass |
| Req. B1- 4 | Test Vehicle | Velocity in Generated Trajectories Does not Exceed Maximum Velocity Profile | Pass |
| Req. B1- 5 | Test Vehicle | Generated Trajectories comply with Behavioral Rules | Pass |
| Req. B1- 6 | Test Vehicle | Safe Halt receives and Reconstructs the Most Recent Implicit Emergency Trajectory | Pass |
| Req. B1- 7 | Test Vehicle | Generates Trajectories that Avoid Collisions with Objects on the Implicit Emergency Path | Pass |
| Req. B1- 8 | Test Vehicle | Perceives the Environment and Detects Objects on the Implicit Emergency Trajectory | Pass |

Table 12-2: Results of the Safe Halt Reference Solution Verification 2

| Requirement | Test Environment | Criteria | Fail/Pass |
|---|---|---|---|
| Req. B1- 9,<br>Req. U1- 3,<br>Req. U1- 4 and<br>Req. U1- 5 | Test Vehicle | Reference Solution complies to ECU, Communication and Power Supply architecture | Pass |
| Req. B1- 9,<br>Req. U1- 6,<br>Req. U1- 7,<br>Req. U1- 8,<br>Req. U1- 9 and<br>Req. U1- 10 | Test Vehicle | Reference Solution provides Functionality as ASOA service | Pass |
| Req. B1- 10 and<br>Req. U1- 11 | Test Vehicle | Termination Signal at End of MRM | *Not Implemented* |
| Req. U1- 12 and<br>Req. U1- 13 | Test Vehicle | Safe Halt receives and processes TAI time reference | Pass |
| Req. B1- 11 | Test Vehicle | Safe Halt Functions are Triggered Eventbased | Pass |
| Req. B1- 12,<br>Req. B1- 13 and<br>Req. U1- 14 | IPG CarMaker™, HiL, Test Vehicle | Safe Halt Functionality is Integratable in Different Application Environments | Pass |
| Req. B1- 14 and<br>Req. U1- 15 | Test Vehicle | All external HMI Elements to Raise Awareness are Activated | *Not Implemented* |

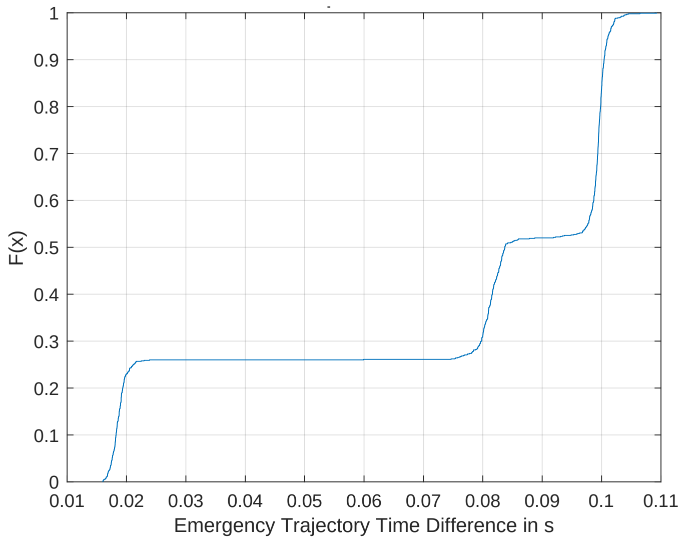


Figure 12-5: CDF of Trajectory Generation Time Difference

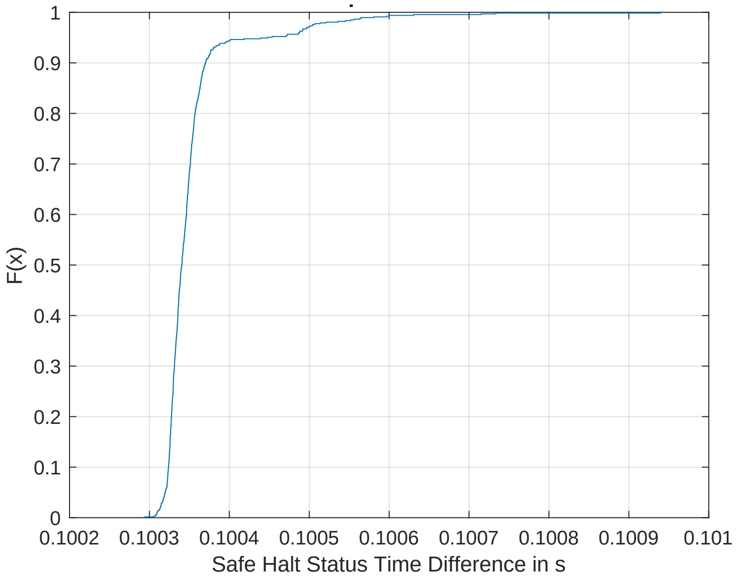


Figure 12-6: CDF of Safe Halt Status Time Difference

Req. B1- 3 and Req. B1- 4 require *Safe Halt* to comply with the most recent implicit emergency trajectory. The generated trajectories must not deviate from the implicit emergency trajectory contents. This concerns the spatial reference positions, the reference yaw angles, and the maximum velocity profile. Therefore, for verification, it is checked whether the emergency trajectory respects the implicit emergency trajectory. For this purpose, as a reference, a linear interpolation of the contents of the implicit emergency trajectory is compared with the reconstruction. The required tolerance in tracking the implicit emergency trajectory is TBD and depends on the Operational Design Domain (ODD) of the Automated Vehicle (AV). This test also verifies that the reconstruction of the implicit trajectory is successful, thus satisfying Req. B1- 6. Fig. 12-8 shows a recorded trajectory during the execution of the test case implicit emergency trajectory.

The entire implicit emergency trajectory can be seen in the top part of the figure. The trajectory is seen in more detail in the middle and bottom parts of the figure. In particular, in the lowest image, a deviation of the emergency trajectory elements from the linearly interpolated implicit emergency trajectory position course can be seen. The cubic interpolation of the implicit emergency trajectory positions causes this deviation. As a metric to evaluate the interpolation results and to check for oscillation of the position course, a linear interpolation of the position course is calculated as a reference. This step calculates the segment lengths of the implicit emergency trajectory. In addition, the segment lengths of the cubically interpolated position course are calculated. For the evaluation, both results are put into relation segment by segment, and a corresponding factor per segment is calculated. With the help of this factor, the maximum spatial deviation of the cubic interpolation can be determined. For the test run considered here, a CDF plot of the factors of all test implicit emergency trajectory segments is shown on Fig. 12-7. The plot shows no significant lengthening factors. This check is performed for each new interpolated implicit emergency trajectory. A segment factor that is too large will cause the corresponding emergency path to be discarded.

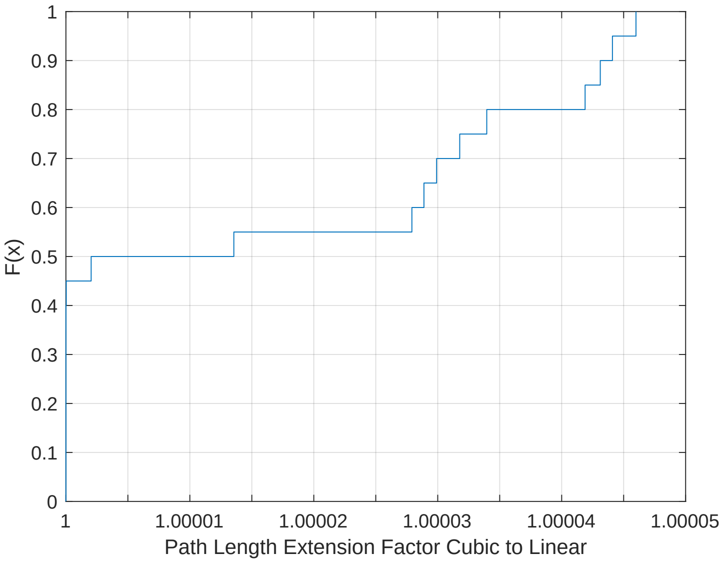


Figure 12-7: CDF of Path Length Extension Factor Cubic to Linear

On Fig. 12-9 the velocity profile of the first emergency trajectory is shown. Due to the cubic interpolation of the squared velocity profile, a deviation from the linear interpolation can be seen here as well. However, it is clearly recognizable that the cubic interpolated course runs through the implicit emergency trajectory maximum velocity profile elements. Because of the interpolation, the velocity is higher, especially at the beginning, than it would be allowed if linearly interpolated. By interpolating the velocity profile of the implicit emergency trajectory, the velocity is described as a function of the trajectory length $s$. However, the trajectory generated by *Safe Halt* depends on time $t$. Verifying that the trajectories generated by *Safe Halt* do not exceed the velocity profile should be done within the function *Safe Halt* since only the reference time $t$ is included in the output trajectory.

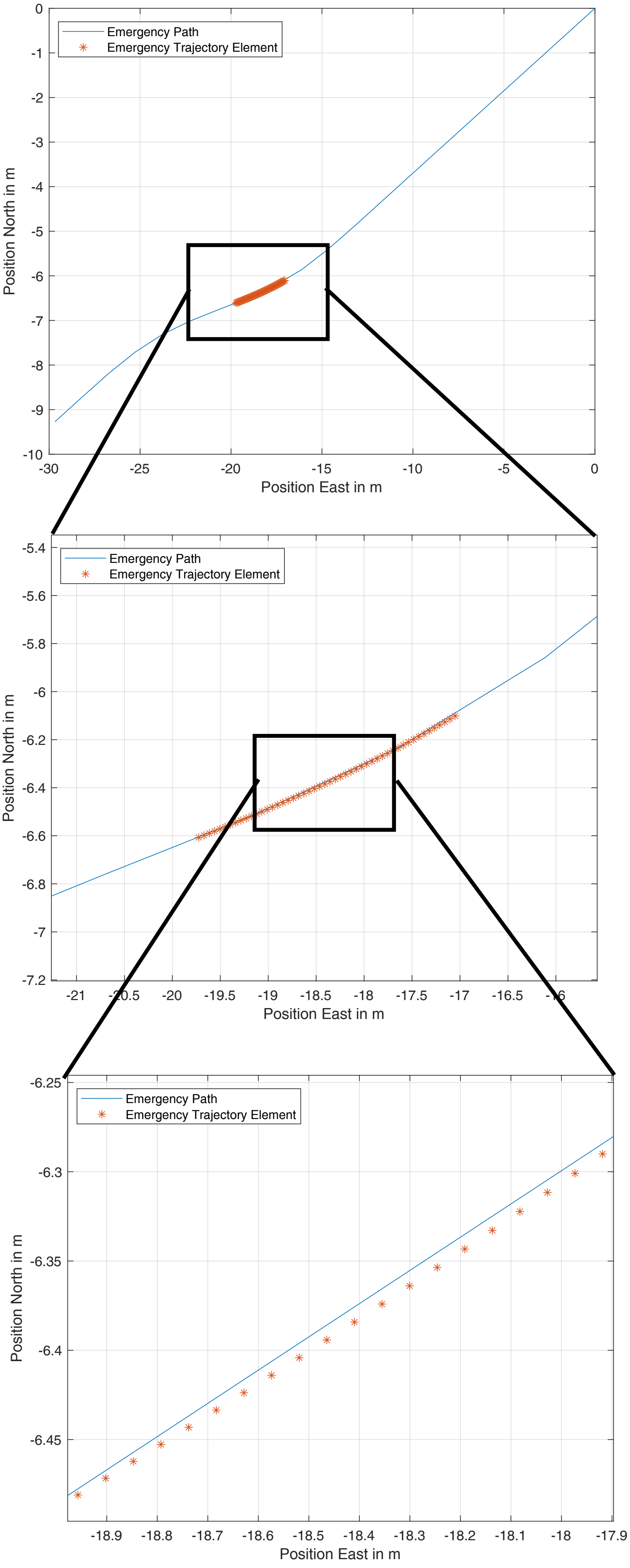


Figure 12-8: Visualization of the Positional Deviation for a Test Implicit Emergency Trajectory

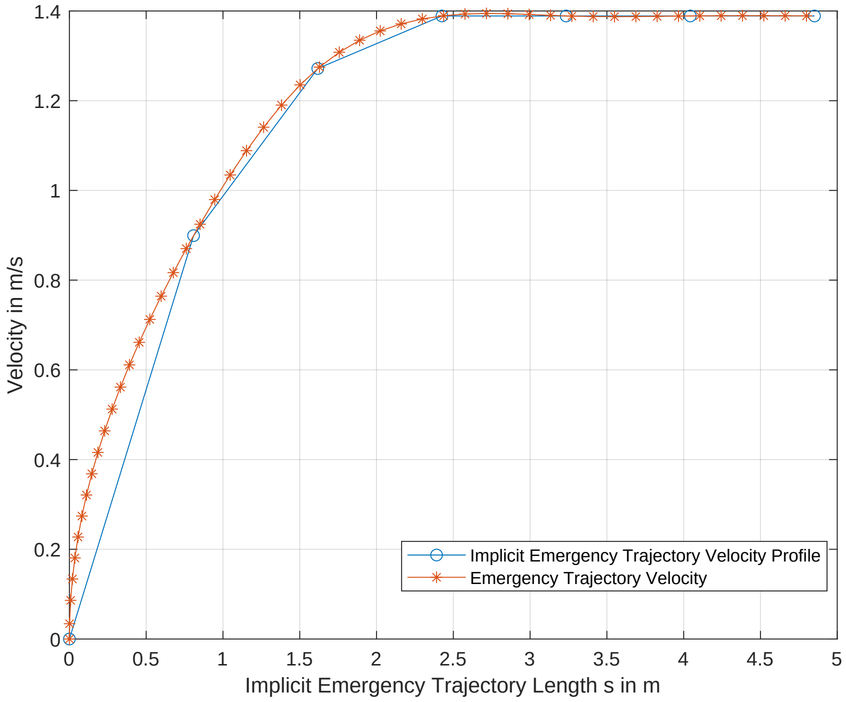


Figure 12-9: Test Implicit Emergency Trajectory and a Generated Emergency Trajectory to Visualize a Velocity Deviation

Fig. 12-10 shows the yaw angle progression for an emergency trajectory. It can be seen that the first trajectory tracks the yaw angle perfectly. Analogous to the verification for the velocity profile, in the implicit trajectory, the yaw angle is described as a function of the trajectory length $s$. However, in the output of *Safe Halt*, the yaw angle is described as a function of time $t$. Verification that the yaw angle profile is maintained should therefore be done within *Safe Halt*.

Req. B1- 7 and Req. B1- 8 require *Safe Halt* to respond to objects on the implicit emergency trajectory. For this purpose, collision with another object is used as a criterion. If an object is placed in the implicit emergency trajectory so that a collision would occur without a reaction of *Safe Halt*, and *Safe Halt* prevents a collision by adapting the velocity, this criterion is passed. Furthermore, it can be shown with this evaluation that *Safe Halt* complies with all behavior rules, in this case, the time gap to the detected object, and thus Req. B1- 5 is fulfilled. Fig. 12-11 shows the beginning of a real test drive. The ego vehicle is drawn as a black rectangle. The position of the ego vehicle is determined by the localization function of the ADS and mapped into the local navigation coordinate system of the current implicit emergency trajectory. Due to the low vehicle speed, almost all objects are filtered out. The reference position of one remaining object is marked with a blue cross. In this state, the vehicle starts accelerating to the test speed. The higher vehicle speed increases the relevant object area that shall be checked for collision objects. Many relevant objects are shown on Fig. 12-12. Objects in front of the AV as well as behind the vehicle are identified. This verifies that both radar sensors are active and that the transformation to the shared vehicle reference point was successful. The objects are checked in detail for collisions in the driving corridor. The collision check results in one object located in the driving corridor.

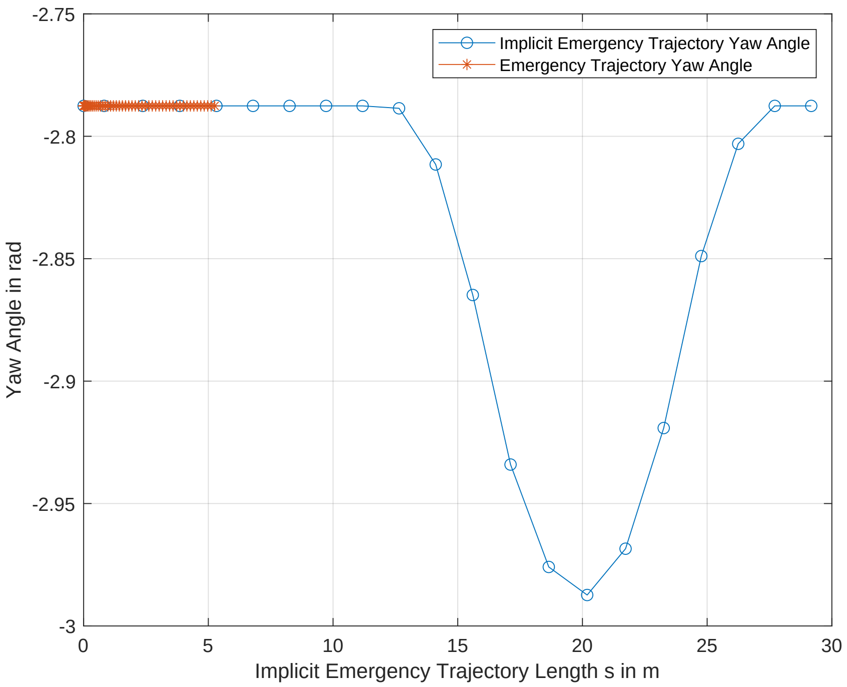


Figure 12-10: Test Implicit Emergency Trajectory and a Generated Emergency Trajectory to Visualize a Yaw Angle Deviation

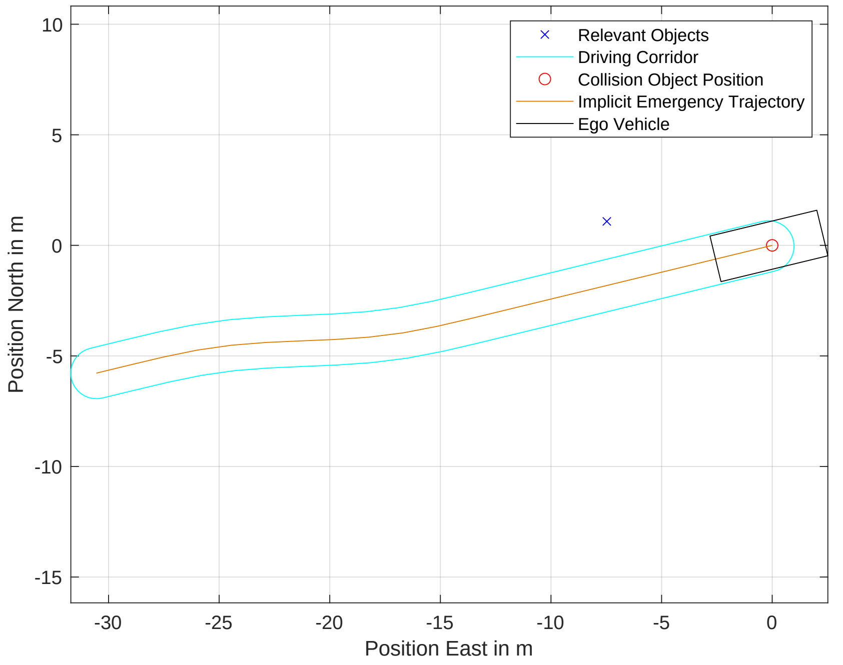


Figure 12-11: Start of Testrun with Collision Objects in the Driving Corridor

The overlapping polygon of the implicit emergency trajectory and the object is pink. The object stands still. Based on the identified object, the collision object position is determined. This is drawn as a red circle. The speed profile is planned so that the vehicle reaches the object's speed at this position, which is the standstill. Due to the real test drives with the prototype vehicles, a significant safety distance to the collision object is maintained to reduce the risk. Therefore, the red circle is located at a considerable distance from the object compared to usual vehicle behavior. Due to the velocity-dependent width of the driving corridor in Equ. 9-5, the driving corridor is wider than at the beginning of the test drive at a lower speed.

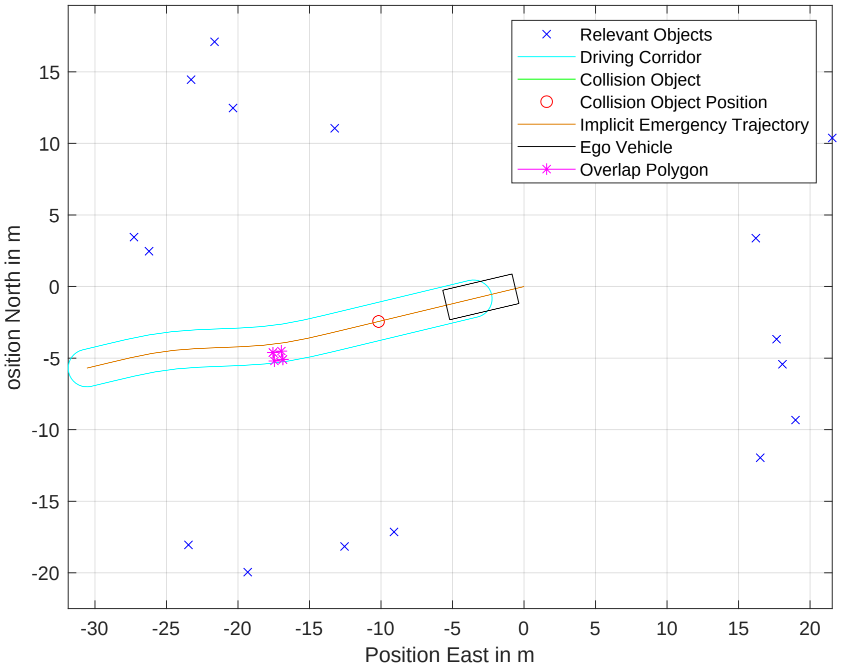


Figure 12-12: Testrun with Collision Objects in the Driving Corridor At Test Speed

The AV is brought to a standstill. This is shown in Fig. 12-13. Due to the standstill, again, only a few relevant objects are shown. In addition, the width of the driving corridor is again minimal.

For this test drive, the soft target, which can be seen on Fig. 12-4, was used as the target object. It is pulled out of the driving corridor on Fig. 12-14. The overlap polygon is reduced, but the object still overlaps the implicit emergency trajectory.

In figure Fig. 12-15, the object was pulled entirely out of the driving corridor. According to the specifications, the vehicle is accelerated again.

Finally, in figures Fig. 12-16 and Fig. 12-17, the AV approaches the MRC.

Further test cases are performed with other objects. Fig. 12-18 shows an excerpt from a test case with a BMW i3.

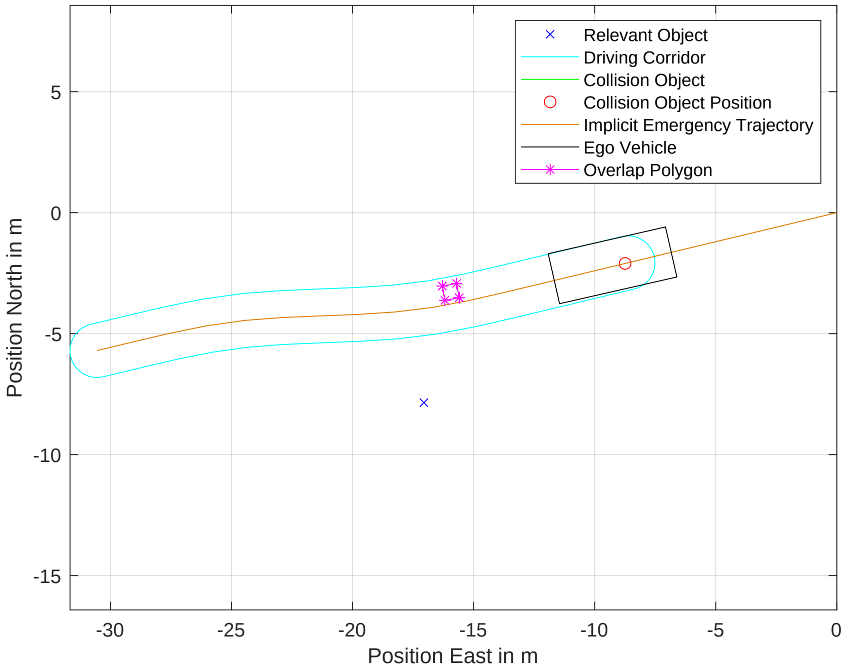


Figure 12-13: Testrun with Collision Objects in the Driving Corridor at Standstill

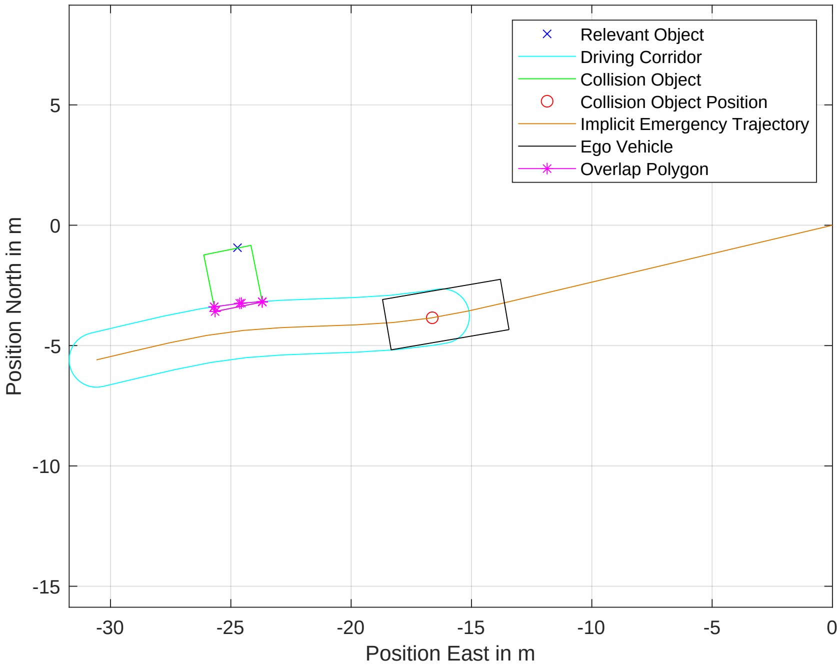


Figure 12-14: Testrun with Collision Objects in the Driving Corridor at Standstill with Object Removal

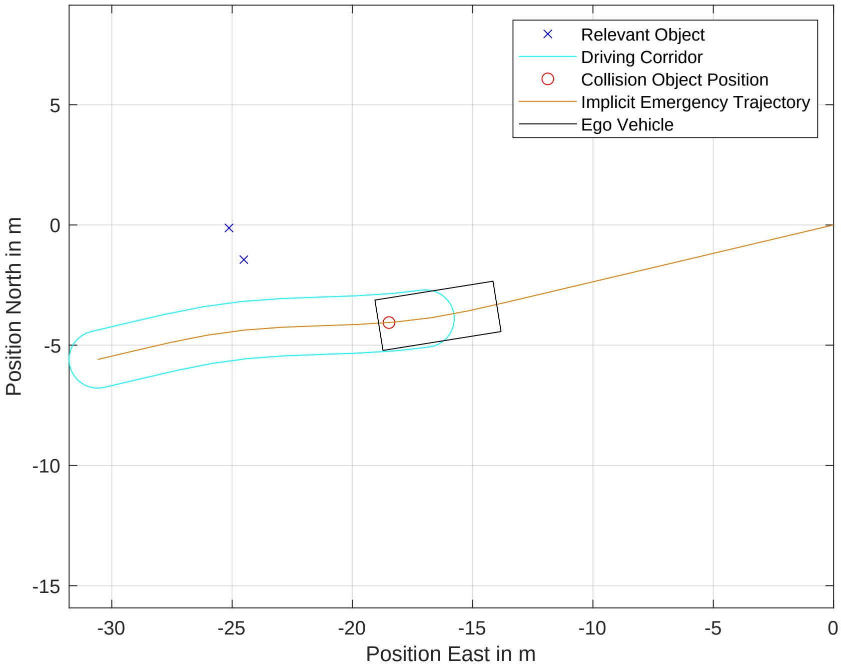


Figure 12-15: Testrun with Collision Objects in the Driving Corridor at Standstill with Removed Object

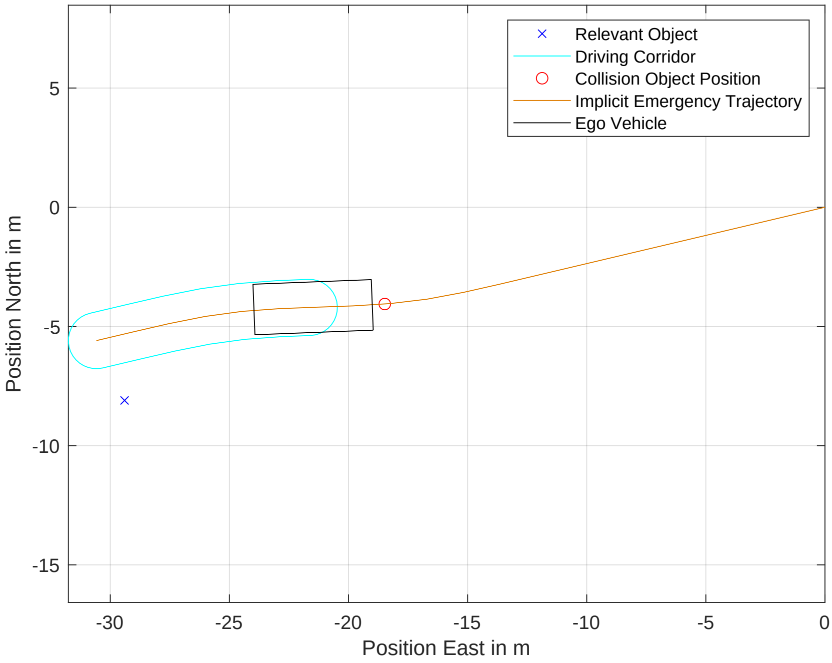


Figure 12-16: Testrun with Collision Objects in the Driving Corridor Close to the MRC

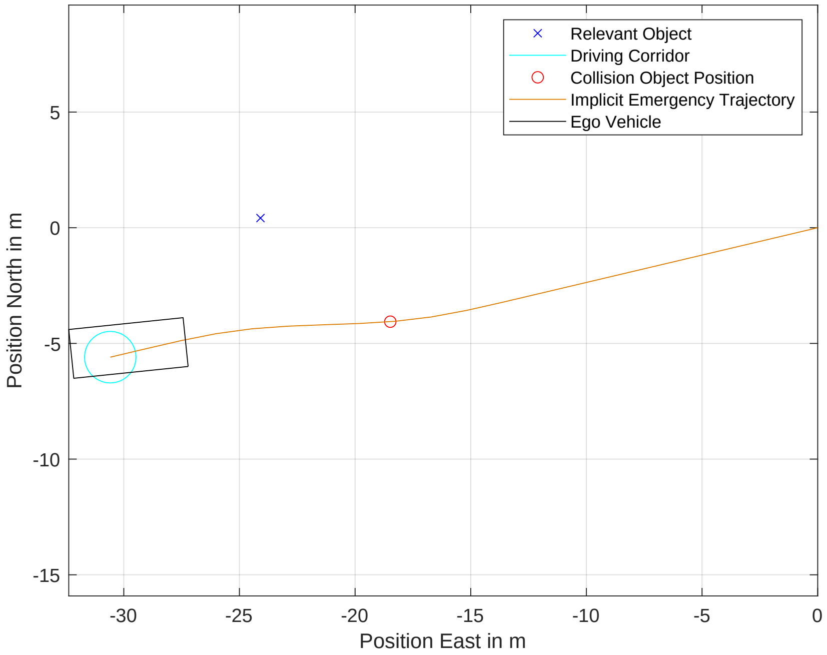


Figure 12-17: Testrun with Collision Objects in the Driving Corridor at the MRC

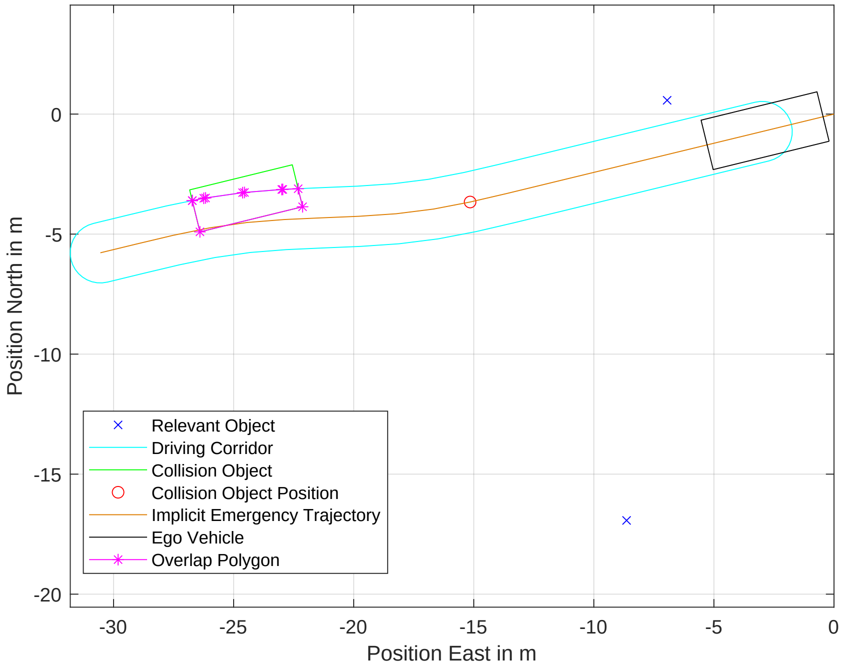


Figure 12-18: Testrun with Collision Object BMW i3

To verify Req. B1- 9, Req. U1- 3, Req. U1- 4 and Req. U1- 5, it is checked whether *Safe Halt* is available and can communicate with the relevant components in ADS. The functionality was available in the actual test drives, and the vehicle followed the generated trajectories. These pass criteria are therefore met.

Req. B1- 9, Req. U1- 6, Req. U1- 7, Req. U1- 8, Req. U1- 9 and Req. U1- 10 are tested by checking whether *Safe Halt* can be discovered as an Automotive Service-Oriented Software Architecture (ASOA) service by ASOA Orchestrator. It also checks whether the service can be started and connected to other ASOA services.

To fulfill the requirements Req. B1- 10 and Req. U1- 11, an interface and a communication partner shall be defined. No communication partner was identified during the solution implementation for UNICAR*agil*. Therefore, no concept and implementation for these requirements were implemented. Due to this lack of implementation, these requirements are not met.

Req. B1- 12 and Req. B1- 13 are satisfied if *Safe Halt* generates trajectories with correct Temps Atomique International (International Atomic Time) (TAI) time stamps. The *Trajectory Tracking Controlling* compares the TAI time reference of the trajectory with the TAI time reference of the *Vehicle Dynamics State Estimation*. Only if both times are nearly identical the input data is used to calculate the setpoints of the actuators. Since these checks were performed successfully and the vehicle was operated, both requirements are fulfilled.

That the requirements Req. B1- 12, Req. B1- 13 and Req. B1- 14 are met is verified by the fact that the functions could be used equally in Software in the Loop (SiL), Hardware in the Loop (HiL) and the actual test vehicles. Each of these test environments uses the same generic interfaces. This ensures that Safe Halt's functions can be integrated into other environmental conditions or middlewares.

Req. B1- 14 is fulfilled if the hazard warning lights are activated after Safe Halt activation and corresponding warning messages are issued on the external HMI for the surrounding road users. The interface to the external HMI is defined, but the activation of the external HMI has not been tested thoroughly. Thus, the verification of this requirement is pending.

## 12.3 Conclusion

This chapter presents test cases and test criteria for verifying a *Safe Halt* reference solution. The actual prototype vehicles of the research project UNICAR*agil* are primarily suitable as test environments. SiL and HiL test environments have the disadvantage that the actual vehicle is not operated in these environments. However, these environments are suitable for a rapid prototyping development of *Safe Halt*. Various test implicit emergency trajectories are generated for test cases, which can be used to verify the requirements and functions of *Safe Halt*. For test independence from the ADS, test cases are selected where the AV is in a standstill and the vehicle operation mode *Safe Halt*. The test implicit emergency trajectories accelerate the vehicle to a test speed, after which the actual functionality of *Safe Halt* can be tested. For all requirements, test criteria are presented that a reference solution of *Safe Halt* shall meet. These test cases and criteria are then applied to the reference implementation of *Safe Halt* in UNICAR*agil* ADS. The results of the verification can be found accordingly in the chapter.

# 13 Conclusion and Outlook

In this section, the research questions and the working hypothesis are revisited in Sec. 13.1. In Sec. 13.2, the conclusion of this work is presented and summarized. Finally in Sec. 13.3, the outlook and future work is given.

## 13.1 Review of the Research Questions and Working Hypothesis

This thesis aims to evaluate the *Safe Halt* concept to enable a fail-safe property of an Automated Driving System (ADS).

The first research questions was

**RQ. 1:** How can the fallback concept of Safe Halt be evaluated in terms of enabling the fail-safe property of a generic ADS and the UNICAR*agil* ADS?

The question is answered by defining the fault tolerance regimes for ADS in Chap. 4. If *Safe Halt* helps an ADS to have a fail-safe property, the evaluation is positive. One metric for evaluating *Safe Halt* is the set of fault combinations for which an ADS with *Safe Halt* has a fail-safe property. The larger this set is with respect to the set of possible fault combinations, the greater the effectiveness of such a system.

A reference solution of the *Safe Halt* concept had to be implemented to evaluate it. As a second research question, the following question was therefore asked regarding the requirements:

**RQ. 2:** What requirements shall a Safe Halt solution meet?

To answer this question, the requirements for a *Safe Halt* reference solution were determined in Chap. 5, Chap. 7, Chap. 8, Chap. 9, and Chap. 10. The requirements were defined in two stages. First, a generic ADS was considered, and the requirements for this generic ADS were described. Subsequently, the UNICAR*agil* ADS is used as a possible manifestation of an ADS, and the detailed requirements are derived. Using this methodology, the generic requirements can also be applied to other ADS.

The third research question was:

**RQ. 3:** What is a concept and a functional architecture for a Safe Halt solution to satisfy the specified requirements?

To answer this question, concepts for meeting the requirements for a *Safe Halt* reference solution for the UNICAR*agil* ADS are presented in Chap. 5, Chap. 7, Chap. 8, Chap. 9, Chap. 10 and Chap. 11. Since generic ADS can have diverse manifestations, only a generic functional concept for a *Safe Halt* solution is presented for them. However, the application to UNICAR*agil* opens potentially relevant perspectives that may be transferable to other ADS expressions as well.

The final research question was:

**RQ. 4:** How can a Safe Halt solution be verified?

To answer this question, the requirements for a *Safe Halt* reference solution in Chap. 5 were used to define test criteria and test cases for each requirement. Using test cases representing different configurations of an implicit emergency trajectory, the test criteria can be applied. Pass and fail criteria are defined for each test criterion, ant then applied to the implemented reference solution of *Safe Halt*.

The original working hypothesis can be evaluated using these answers to the research questions. It was:

> An Integration of Safe Halt into an ADS and in Particular into the UNICAR*agil* ADS Enables its Fail-Safe Property without any Limitations.

In this thesis, limitations of the *Safe Halt* concept were uncovered.

The first limitation concerns the reconstruction of the pre-planned implicit emergency trajectory. This planning of the implicit emergency trajectory compresses it prior to the transmission to *Safe Halt*. The compressed data includes the poses and maximum velocity profile at each sampled element of the implicit trajectory. Not included so far is the course angle of the vehicle, i.e., the velocity direction at each sampled element. This missing information can cause a reconstruction of the compressed data to result in undesirable oscillatory behavior of the position trajectory. Therefore, the course angle should also be integrated into the implicit emergency trajectory to address this limitation.

The second limitation concerns the neighboring and superordinate systems in the ADS. *Safe Halt* can only fulfill its function if all subsequent systems are fail-operational. For UNICAR*agil*, these are nodes U5 *Trajectory Selection*, U6 *Vehicle Dynamics State Estimation*, U7 *Trajectory Tracking Controlling*, and U8 *Actuation* depicted on Fig. 4-6. In addition, *Safe Halt* cannot activate itself. For this reason, node U9 *Self-Perception* shall monitor the capabilities of the ADS and switch to *Safe Halt* when necessary. This limitation does not concern the concept *Safe Halt* itself, but the requirements for neighboring systems. If these requirements are not met, the functionality of *Safe Halt* is unavailable.

The third limitation includes the concept *Safe Halt* only marginally. The function acts as an independent source of emergency trajectories. Other vehicle operation modes may also include trajectory sources. Before switching between trajectory sources, it shall be checked whether the trajectories of both sources match each other. If not, switching the trajectory source can cause dangerous situations if the trajectories are far apart.

The fourth limitation to the capabilities of *Safe Halt* is the lack of traffic signal phase detection. With the current environment perception and infrastructure usage, *Safe Halt* cannot detect traffic signal phases. Thus, when planning the implicit emergency trajectory, it shall be ensured that no traffic signals are crossed. Furthermore, since *Safe Halt* has no information about traffic rules, it cannot use right before left and similar traffic rules. Therefore, compliance with traffic rules should be observed via the planning of the implicit emergency trajectory.

With these identified limitations of the working hypothesis, the working hypothesis can now be updated.

Functionally, no further limitations of *Safe Halt* were identified. The reference solution meets the known specifications of the *Safe Halt* concept. However, a comprehensive validation of the concept is missing in this thesis. During a validation study, further limitations may be uncovered that were not revealed during verification.

## 13.2 Conclusion

The concept for *Safe Halt* described at the beginning of this thesis has been proven to work by defining the fail-safe properties of an ADS and by providing a functional implementation of a reference solution. The concept provides an automated Dynamic Driving Task (DDT) fallback solution for an ADS. Assuming that most fault combinations of an ADS occur in nodes B2 *Sensing, Processing and World Modeling* and B3 *Strategical and Tactical Planning* depicted on Fig. 4-5 and all other nodes can be made fail-operational with reasonable effort, *Safe Halt* is located in the optimal position in the ADS effect chain. Using the strategies introduced in Sec. 4.2.5 to handle fault combinations of an ADS with *Safe Halt*, it has been demonstrated that *Safe Halt* should only be fail-safe. If it fails, a Minimal Risk Maneuver (MRM) shall be executed in *Automated Driving* vehicle operating mode. Thus, it is unnecessary to configure the environment perception of *Safe Halt* to be fail-operational. Due to the pre-planned implicit emergency trajectory of node B3 *Strategical and Tactical Planning*, *Safe Halt* shall not include capabilities to independently plan a MRM. The environment sensor system shall only have the capabilities to ensure the execution of the MRM at an average low speed. Thus, the required capabilities for *Safe Halt* are also minimal. This includes both the complexity of the system and its testing effort. *Safe Halt* is thus a candidate to be included as a standard in future functional architectures of ADS. Due to the flexibility enabled by the planning of the implicit emergency trajectory, from today's point of view, there is also no argument against the introduction of specific requirements for MRM, such as those to be maintained in SAE J3164[119].

Koopman requires in his presentation *Autonomous Vehicle Standards & Open Challenges*[120] that Automated Vehicle (AV) shall be more than fail-safe to terminate a driving mission gracefully. However, according to the fault tolerance regime selected in this thesis, a fail-safe property of an ADS is sufficient to fulfill this function. Furthermore, he mentions safety architectures as a challenge for AV. Explicitly, he writes that redundancy is not necessarily sufficient since it does not cover common mode failures. This challenge is addressed by the ADS architecture with *Safe Halt*. This DDT fallback is not a redundancy and thus avoids common mode failures. The second challenge with two conflicting computations is addressed in *Safe Halt* by always reacting as soon as one computation detects an object. Since *Safe Halt* is a deterministic safety function and thus the false-negative rate shall be minimized, this approach is plausible. Finally, *Safe Halt* provides an architectural and functional solution to the challenge of the fail-safe property of an AV. However, the challenge remains to detect the faults that have occurred. Nevertheless, again, *Safe Halt* provides a fail-safe property for various fault combinations. Thus, fault detection shall not detect the exact fault combination. It is sufficient only to detect that *something is wrong* and to activate *Safe Halt* on this basis.

## 13.3 Outlook and Future Work

This section highlights indications of future research potential concerning a *Safe Halt* function.

### 13.3.1 Applicability to Other ADS

The basic applicability of the findings of this thesis has already been ensured by structuring the chapters of this thesis. In the future, however, an extension of the functionality of *Safe Halt* is conceivable. In the functional architecture of the SAE J3131[121], the functional block *Safety Monitor* is provided. This function has the task of identifying safety-critical anomalies in the planned intended behavior of the vehicle and preventing violations of safety criteria by monitoring them. Solutions have already been presented for the functional fulfillment of a *Safety Monitor*. For example, Stahl[122] presents an online verification for an automated racing vehicle, and Popp et al.[123] (the author of this thesis is also one of the co-authors of that article) and Popp[124] present

[119] SAE International: SAE J3164 (2018).

[120] Koopman, Philip: AV: Standards & Open Challenges (2022), slide 5.

[121] SAE International: SAE J3131 (2022), p. 3.

[122] Stahl, Tim Nikolaus: Safeguarding complex and learning-based automated driving functions (2022).

[123] Popp, C. et al.: Approach to Maintain a Safe State of an Automated Vehicle (2022).

[124] Popp, C.: Safety-Check von Trajektorien beim Automatisierten Fahren (2023).

different approaches for maintaining the safe state of an automated vehicle. These approaches use the same environment perception system for safety monitoring and planning the intended behavior. *Safe Halt* introduces its own environment perception system and sensor data processing to the AV. *Safe Halt*'s reference solution can be extended to provide safety monitoring of the trajectory for *Automated Driving* vehicle operating mode. For this purpose, all functions of the *Safety Monitor* can be integrated into *Safe Halt*. In addition, the environment perception system of *Safe Halt* is available as an additional source of environment information. Fig. 13-2 shows *Safe Halt* extended by the functional blocks for the realization of a *Safety Monitor*. For this purpose, the functional ADS architecture C0 is introduced. It is shown in Fig. 13-1.

Integration of a *Safety Monitor* in *Safe Halt* for the UNICAR*agil* ADS is not conceivable in this way since this functionality would bypass the *Self-Perception* of the ADS. However, it would be possible to transmit only the result of the trajectory check to *Self-Perception* without *Safe Halt* modifying the trajectory. A trajectory classified as unsafe would then cause *Self-Perception* to change the vehicle operating mode to *Safe Halt*.

### 13.3.2 High Definition (HD) Maps

*Safe Halt*'s knowledge about the current scene is limited to the poses of the implicit emergency trajectory. *Safe Halt* has no further information about the vehicle environment. However, for predicting the states of detected objects, it may be necessary to know the street courses and, thus, the scenery. So far, it has been assumed that the average vehicle speed during a *Safe Halt* MRM is low enough to dispense with an accurate prediction of the surrounding objects. However, if validation reveals that more precise prediction of objects is necessary, the use of digital HD maps in *Safe Halt* can be explored.

### 13.3.3 Evaluation of the Situation-Dependent Global Minimal Risk Condition

With a solution of *Safe Halt*, a global risk minimization of a MRM can be performed. A local Minimal Risk Condition (MRC) is always aimed for in state of the art. With the introduction of *Safe Halt*, the risk of a MRC can be combined with the risk of the MRM, and thus the situation-dependent global MRC can be obtained. However, the metrics for assessing the risk of global MRC are still lacking for this purpose. The same is true for the risk assessment of a MRM. Future work should aim to evaluate these risks. These findings should then be incorporated into the planning of implicit emergency trajectories. The selection of the situation-dependent MRC shall be made dynamically. Before selecting the MRC, B3 *Strategic and Tactical Planning* shall check whether the condition is reachable and thus unoccupied. The own environment perception can check this, but it is also possible to get this information via infrastructure sensors or cloud functions.

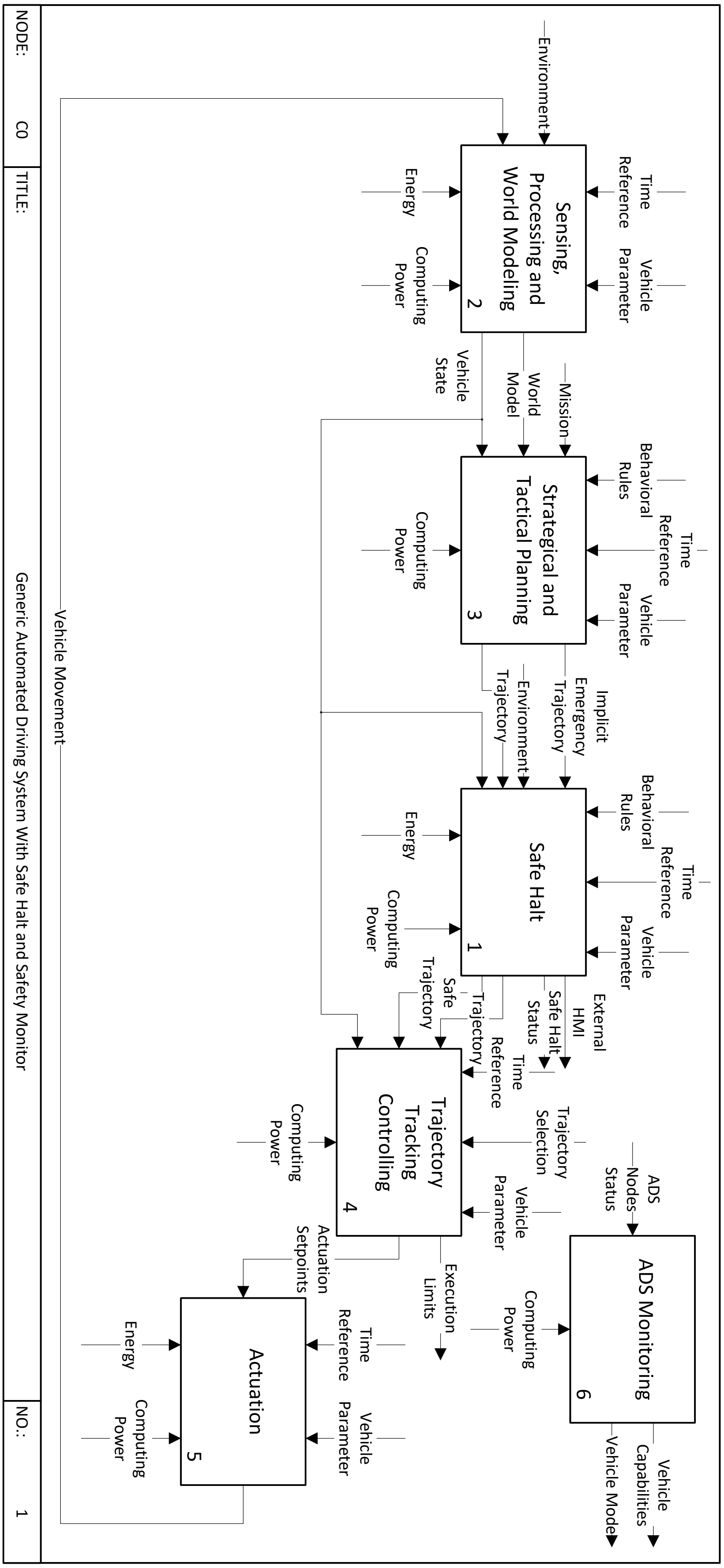


Figure 13-1: The Functional Architecture of a Generic ADS with Safe Halt and Safety Monitor in the IDEFØ Representation

### 13.3.4 Reduction of Accident Consequences

*Safe Halt* is a safety function for AV. The task of this system is thus to prevent collisions with other road users during the MRM. The scope of the function could be extended to reduce the severity of damage caused by a collision in the case of collisions that cannot be prevented. For this purpose, the accident consequence mitigation approaches of Heck[125] could be integrated into the velocity profile adaption functionality of *Safe Halt*. For accident consequence mitigation, *Safe Halt* shall detect the passenger compartment of the collision object. This is not possible with the current environment perception system. Appropriate planning approaches also exist for planning accident consequence-minimizing trajectories using an optimal control problem. Parseh et al.[126] present their findings and verify them in a simulation environment.

### 13.3.5 Ergonomic Design of the MRM

In this thesis, the focus is on the fail-safe property of an ADS. *Safe Halt* has the task to maintain the safe state even in case of failures of parts of the ADS and to execute a MRM to the situation-dependent global MRC. The MRM is planned to avoid collisions with other road users. A MRM is an emergency situation and will, therefore, only have to be executed in very few exceptional cases. Human vehicle occupants will not experience such a situation frequently and will be suddenly surprised when it is executed. Since it is an AV, the human passenger cannot communicate with the vehicle. Thus, executing a MRM can become an overwhelming event for a human passenger. Therefore, in addition to risk minimization, it is imperative to consider the ergonomic side of the MRM. The occupants of the vehicle should be informed of the situation that has occurred. Furthermore, the MRM should be planned in such a way that they address the needs of human passengers. To this end, research by Karakaya et al.[127] [128] published, and should be integrated in the planning of the implicit emergency trajectory and the trajectory generation functionality of *Safe Halt*.

### 13.3.6 Validation

This thesis identifies the requirements for a generic *Safe Halt*. Together with specific requirements for the UNICAR*agil* ADS, concepts and implementations of a reference solution for *Safe Halt* in the UNICAR*agil* ADS are provided. However, this reference solution was only tested on closed-off test sites with a small number and class of objects and scenarios. Validation within the scope of this thesis was impossible because the prototype vehicles of UNICAR*agil* have no approval for public road traffic, and validation requires comprehensive planning and execution

[125] Heck, J.: Unfallfolgenminderung im Querverkehr (2015).

[126] Parseh, M. et al.: Minimizing Collision Severity for Highly Automated Vehicles (2021).

[127] Karakaya, B.; Bengler, K.: Driver Behavior During Minimal Risk Maneuvers (2021), p. 691.

[128] Karakaya, B. et al.: A Video Survey on MRM and MRC (2020).

of the test cases. For this reason, validation does not occur as part of this thesis. However, no findings were made that prevented validation. Future work should therefore address the validation of *Safe Halt*.

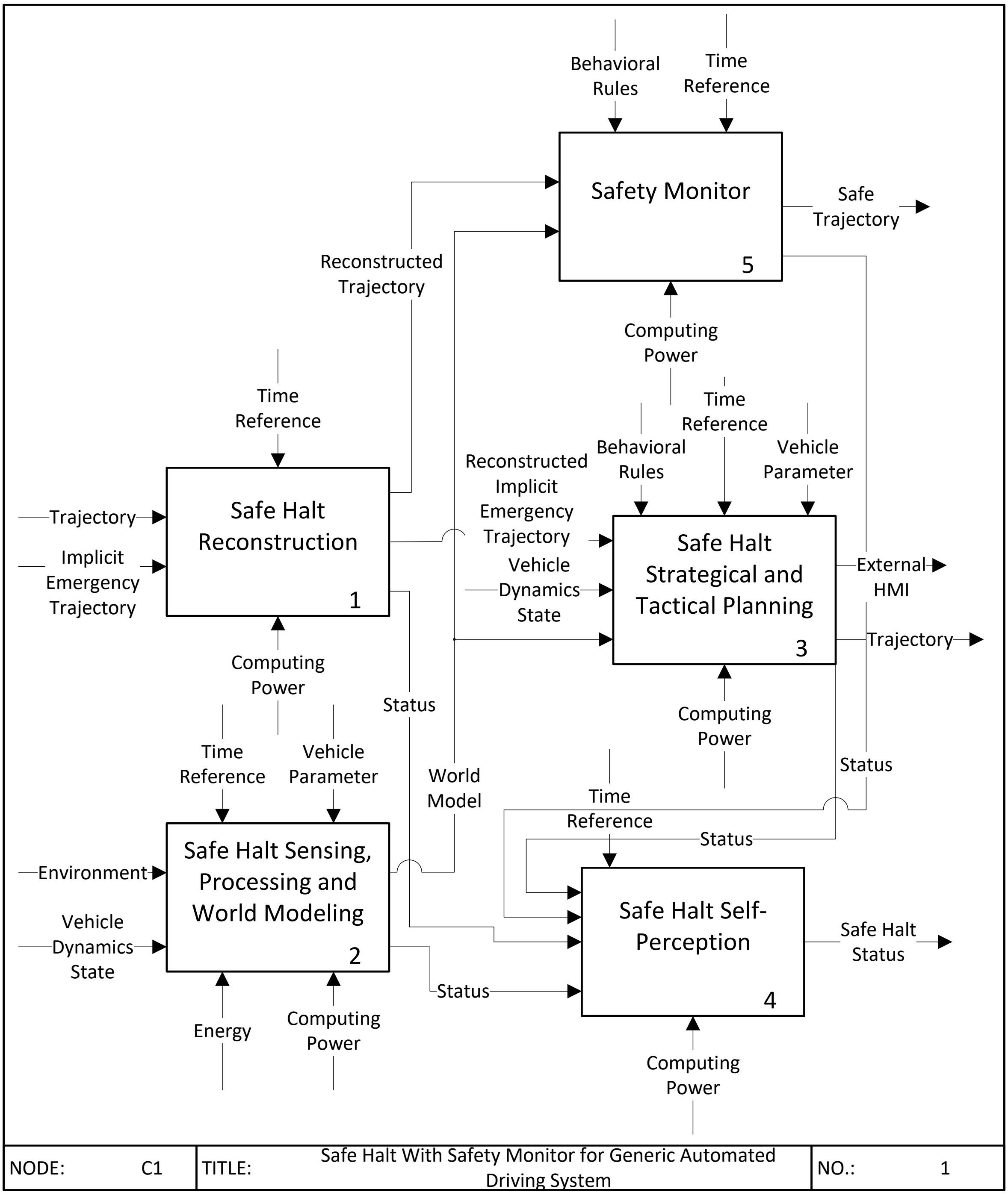


Figure 13-2: Functional Architecture for a Safe Halt Reference Solution with Safety Monitor in a Generic ADS

# A Implicit Emergency Trajectory of the UNICAR*agil* Automated Driving System (ADS)

The implicit emergency trajectory used by the UNICAR*agil* ADS has the following content. The emergency trajectory defined for the UNICAR*agil* ADS consists of two parts. The first part includes information independent of the number of emergency path elements. Tab. A-1 represents this first part of the emergency trajectory. The second part includes the content of each emergency trajectory element. The spatial distance between two emergency trajectory elements depends on the curvature of the emergency trajectory. The number of implicit emergency trajectories is limited to 100 elements for the UNICAR*agil*. However, this number can be increased if necessary. In Tab. A-2, the implicit emergency trajectory elements of this second part are provided. Due to the transformations of the implicit trajectories into local navigation coordinate systems, it shall be guaranteed that the spatial length of the implicit emergency trajectory does not lead to an unacceptable inaccuracy (cf. Annex C).

Table A-1: Header for the UNICAR*agil* ADS Implicit Emergency Trajectory Interface

| Identifier | Description | Unit | Value |
|---|---|---|---|
| EP_C1 | Number of implicit emergency trajectory elements | | |
| EP_C2 | Output timestamp of implicit emergency trajectory in TAI time | s | |
| EP_C3 | Output timestamp of implicit emergency trajectory in TAI time | ns | |
| EP_C4 | implicit emergency trajectory was planned with current vehicle pose (0) or using previous implicit emergency trajectory (1) | | 0 or 1 |
| EP_C5 | implicit emergency trajectory was planned with localization function 1 (0) or 2 (1) | | 0 or 1 |
| EP_C6 | Vehicle pose the implicit emergency trajectory was planned with (ETRS89 Latitude) | rad | |
| EP_C7 | Vehicle pose the implicit emergency trajectory was planned with (ETRS89 Longitude) | rad | |
| EP_C8 | Vehicle pose the implicit emergency trajectory was planned with (ETRS89 Height) | m | |
| EP_C9 | Vehicle pose the implicit emergency trajectory was planned with (ENU Yaw) | rad | |
| EP_C10 | Localization timestamp of the planning pose TAI time | s | |
| EP_C11 | Localization timestamp of the planning pose TAI time | ns | |

Table A-2: Content for the UNICAR*agil* ADS Implicit Emergency Trajectory Interface

| Identifier | Description | Unit | Value |
|---|---|---|---|
| EP_E1 | Position (ETRS89 Latitude) | rad | |
| EP_E2 | Position (ETRS89 Longitude) | rad | |
| EP_E3 | Position (ETRS89 Height) | m | |
| EP_E4 | Longitudinal slope | | |
| EP_E5 | Lateral slope | | |
| EP_E6 | Velocity in implicit emergency trajectory course direction | $\mathrm{m\,s^{-1}}$ | |
| EP_E7 | Yaw angle (ENU) | rad | |
| EP_E8 | Direction indicator activator. Off (0), Left (1), Right (2) | | 0 or 1 or 2 |

# B Explicit Trajectory of the UNICAR*agil* ADS

The explicit trajectory used by the UNICAR*agil* ADS has the following content. The trajectory defined for the UNICAR*agil* ADS consists of two parts. The first part includes information independent of the number of trajectory elements. Tab. B-1 represents this first part of the trajectory. The second part includes the content of each trajectory element. The time difference between two trajectory elements is 100 ms, and the total length of the trajectory is 5 s. Thus, the trajectory has 50 elements. In Tab. B-2, the trajectory elements of this second part are provided.

Table B-1: Header for the UNICAR*agil* ADS Explicit Trajectory Interface

| Identifier | Description | Unit | Value |
|---|---|---|---|
| T_C1 | Number of reference trajectory elements | | 50 |
| T_C2 | Output timestamp of reference trajectory in TAI time seconds | s | |
| T_C3 | Output timestamp of reference trajectory in TAI time nanoseconds fraction | ns | |
| T_C4 | Reference trajectory was planned with current vehicle pose (0) or with previous reference trajectory (1) | | 0 or 1 |
| T_C5 | Reference trajectory was planned with localization solution 0 (0) or 1 (1) | | 0 or 1 |
| T_C6 | Vehicle pose the reference trajectory was planned with (ETRS89 Latitude) | rad | |
| T_C7 | Vehicle pose the reference trajectory was planned with (ETRS89 Longitude) | rad | |
| T_C8 | Vehicle pose the reference trajectory was planned with (ETRS89 Height) | m | |
| T_C9 | Vehicle pose the reference trajectory was planned with (ENU Yaw) | rad | |
| T_C10 | Localization timestamp of the planning pose TAI time | s | |
| T_C11 | Localization timestamp of the planning pose TAI time | ns | |

Table B-2: Content for the UNICAR*agil* ADS Explicit Trajectory Interface

| Identifier | Description | Unit | Value |
|---|---|---|---|
| T_E1 | Timestamp in TAI time | s | |
| T_E2 | Timestamp in TAI time | ns | |
| T_E3 | Standstill indicator. No standstill (0), standstill (1) | | 0 or 1 |
| T_E4 | Sideways parking indicator. No sideways parking (0), sideways parking (1) | | 0 or 1 |
| T_E5 | Position (ENU East) | m | |
| T_E6 | Position (ENU North) | m | |
| T_E7 | Longitudinal slope | | |
| T_E8 | Lateral slope | | |
| T_E9 | Velocity magnitude | $\mathrm{m\,s^{-1}}$ | |
| T_E10 | Velocity angle | rad | |
| T_E11 | Acceleration magnitude | $\mathrm{m\,s^{-2}}$ | |
| T_E12 | Acceleration angle | rad | |
| T_E13 | Yaw angle | rad | |
| T_E14 | Yaw velocity | $\mathrm{rad\,s^{-1}}$ | |
| T_E15 | Yaw acceleration | $\mathrm{rad\,s^{-2}}$ | |

# C Analysis of the Influence of the Curvature of the Earth on Local Navigation Coordinate Systems

The earth can be assumed to be flat in close range around the vehicle. Global localization functions output vehicle poses in Geographic Coordinate System (GCS) (cf. Sec. 2.4.1). The localization functions of the UNICAR*agil* ADS use the coordinate system ETRS89. Position data in this coordinate system are three-dimensional and contain the two angles for longitude and latitude and the height above the reference ellipsoid. Instead of using a global coordinate system, a local ENU coordinate system can be generated (cf. Sec. 2.4.2). This system neglects the earth's curvature and outputs the vehicle poses in cartesian coordinates. For simplification, the 'up'-axis of this coordinate system is neglected since the vehicle is floor-bound. If the global positions of the localization functions are mapped into this local navigation coordinate system with neglected earth curvature, inaccuracies will occur. These inaccuracies create a systematic deviation between the mapped vehicle position in the local navigation coordinate system and the actual ETRS89 vehicle position. Below, this inaccuracy is analyzed.

Due to the ellipsoid shape of the earth, the resulting inaccuracy depends on the vehicle's yaw angle. The yaw angle with the most significant influence on the inaccuracy is determined initially. For this purpose, the reference ellipsoid World Geodetic System 1984 (WGS84) is used. The Geodetic Reference System 1980 (GRS80) ellipsoid for the ETRS89 coordinate frame differs only with some ţm compared to the WGS84. First, a starting position is chosen. It is located in the northern hemisphere at the airfield in Darmstadt-Griesheim, Germany, at latitude $lat = 49.855\,213\,999\,999\,7^\circ$, longitude $lon = 8.593\,434\,000\,000\,02^\circ$ and height $h = 0\,\mathrm{m}$. Starting from this position, the yaw angle is divided into 30° sections. 0° is in the direction to the east. The yaw angle rotates counterclockwise up to 360°. First, the geodesic distance $d_{\mathrm{geod}}$ between two ellipsoidal coordinates is calculated. A geodesic distance is the shortest path between two points on an ellipsoid[129a]. This distance is calculated using a root finding algorithm[129b]. Second, the distance of two ellipsoidal coordinates in a local navigation coordinate system is calculated. For this purpose, a local navigation coordinate system is created at the ellipsoidal position of the origin. The second ellipsoidal coordinate is mapped into the same local navigation coordinate system.

The up-axis of the local navigation coordinate system is neglected for the *Safe Halt* and the trajectory tracking controller. Thus the distance $d_{\mathrm{ENU}}$ between the origin of the local navigation system and the mapped second ellipsoid coordinate is calculated by

[129] Karney, C. F. F.: Algorithms for geodesics (2013). a: p.1; b: -.

$$d_{\mathrm{ENU}} = \sqrt{d_{\mathrm{ENU,N}}^2 + d_{\mathrm{ENU,E}}^2}. \tag{C-1}$$

With this, the difference of geodesic distance with the distance $d_{\mathrm{ENU}}$ can be determined using

$$d_{\mathrm{diff}} = |d_{\mathrm{geod}} - d_{\mathrm{ENU}}|. \tag{C-2}$$

This distance difference $d_{\mathrm{diff}}$ is thus the distance between the geodesic distance between two ellipsoid coordinates on the reference ellipsoid and the distance resulting from mapping both ellipsoid coordinates into the same local navigation coordinate system.

A reference distance difference is defined to calculate the yaw angle's influence. For each yaw angle, the resulting geodesic distance for the reference distance difference is determined. A root-finding method is used for this purpose. The reference distance difference is set to 1 m. Tab. C-1 shows that the yaw angles in the north (90°) and south (270°) directions allow the shortest geodesic distance for a reference distance difference of 1 m. In the column "Normalized", the values are normalized to the value for the yaw angle with the most significant influence on the resulting distance difference. It is evident that the yaw angle influences the resulting inaccuracy, but it is smaller than 0.2 % at most. The worst case is a yaw angle in the southern direction at 270°. Since the earth's ellipsoid is flattened at the poles, it is plausible that the most significant curvature and, thus, the greatest distance difference is expected at the equator.

Table C-1: Geodesic Distances for 1 m Distance Difference

| Yaw Angle $\psi$ in ° | Geodesic Distance $d_{\mathrm{geod}}$ in m | Normalized |
|---|---|---|
| 0 | 62576 | 1.00189 |
| 30 | 62525 | 1.00107 |
| 60 | 62475 | 1.00027 |
| 90 | 62461 | 1.00005 |
| 120 | 62475 | 1.00027 |
| 150 | 62525 | 1.00107 |
| 180 | 62576 | 1.00189 |
| 210 | 62524 | 1.00106 |
| 240 | 62473 | 1.00023 |
| 270 | 62458 | 1 |
| 300 | 62473 | 1.00023 |
| 330 | 62524 | 1.00106 |

The worst-case yaw angle of 270° is assumed for further calculations. For this yaw angle, the geodesic distances are calculated for different reference distance differences. Since the geodesic distance calculation is only possible using a root search procedure, the geodesic distance is only given for selected distance differences in Tab. C-2.

Table C-2: Geodesic Distances With 270° Yaw Angle for Given Distance Differences

| Distances Difference $d_{\text{diff}}$ in m | Geodesic Distance $d_{\text{geod}}$ in m |
|---|---|
| 0.0001 | 2899 |
| 0.001 | 6246 |
| 0.01 | 13456 |
| 0.05 | 23010 |
| 0.1 | 28997 |
| 0.2 | 36526 |
| 0.5 | 49573 |
| 1 | 62458 |
| 2 | 78692 |
| 3 | 90080 |
| 4 | 99146 |
| 5 | 106827 |
| 6 | 113493 |
| 7 | 119505 |
| 8 | 125049 |
| 9 | 129916 |
| 10 | 134560 |

If a localization solution is used that outputs global poses in the ETRS89 coordinate system, and the curvature of the earth is neglected for planning trajectories, a systematic inaccuracy of 1 cm occurs after around 13.5 km. After 29 km the inaccuracy is already 10 cm. The relation between the geodesic distance and the distance difference can be seen as a double logarithmic plot in Fig. C-1.

A linear relationship can be observed. A linear relationship in a double logarithmic plot can be formulated as

$$d_{\text{diff}} = a \cdot d_{\text{geod}}^{k}. \tag{C-3}$$

The slope $k$ of the straight line is calculated with

$$k = \frac{\log\left(d_{\text{diff},1}/d_{\text{diff},0}\right)}{\log\left(d_{\text{geod},1}/d_{\text{geod},0}\right)}. \tag{C-4}$$

If the determined values are inserted, the slope $k = 3$ is obtained. The constant $a$ is calculated with

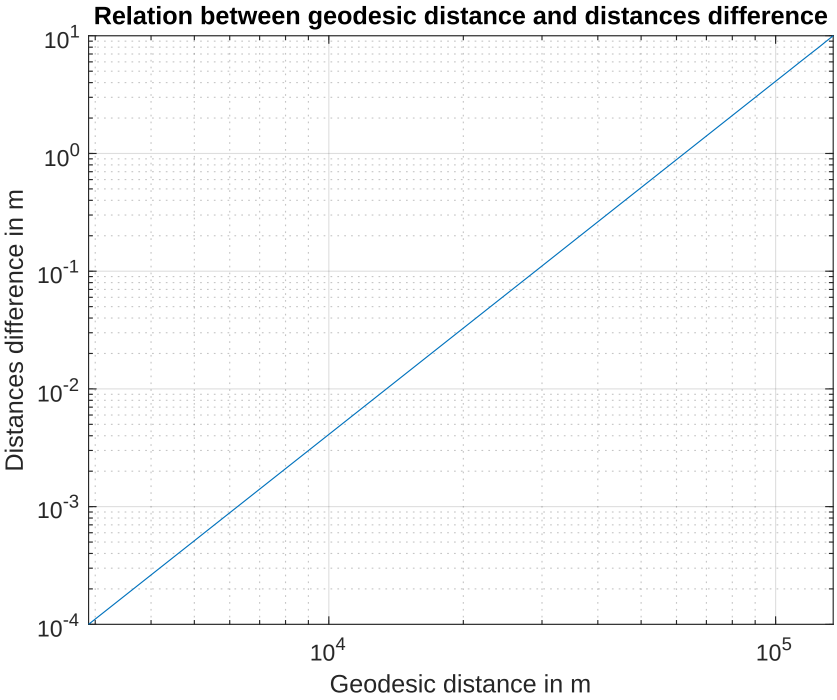


Figure C-1: Relation Between Geodesic Distance and Distance Difference

$$a = \frac{d_{\text{diff},0}}{d^{k}_{\text{geod},0}}, \tag{C-5}$$

to $a = 4.1043 \times 10^{-15}$ m.

The distance difference thus depends cubically on the geodesic distance. The analysis proves that for an origin point in the northern hemisphere, a yaw angle of 270ř has the most significant influence on the resulting inaccuracy due to distance differences. The study further shows the cubic relationship between the geodesic distance and the resulting distance difference. After a geodesic distance of 60 km, a systematic inaccuracy of 1 m arises from mapping the ellipsoidal position of the localization function into a remote local navigation coordinate system. Therefore, neglecting the curvature of the earth is only allowed locally.

# D Task-based Software Architecture of the Safe Halt Reference Solution

To meet the requirements of executing *Safe Halt*'s functions in a resource-efficient manner while maintaining minimum latency response to function input data, each function of *Safe Halt* is implemented as a task[130]. A task has the structure in Fig. D-1.

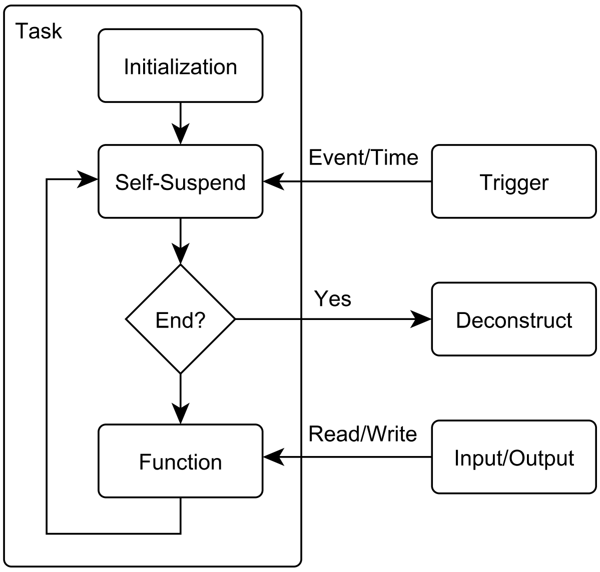


Figure D-1: Functional Architecture of a Task

During the initialization, the corresponding task is initialized. Initialization includes all calculations independent of the task's computation and shall be performed only once. With this strategy, the number of necessary calculations is minimized. If a calculation does not have to be updated over the entire lifetime of the task, it is performed during initialization. After a task is initialized, it enters an idle state. In this state, resource usage is minimized. Tasks can be started by external triggers or by timeouts. After a task has been started, it is checked whether it has been started to terminate itself or whether a calculation has to be performed. If a calculation is to be carried out, the input data is read in the function first. Then these input data are checked. This includes a check for completeness and plausibility. The criteria for the check depend on the contents of the input data. If the examination of the input data turns out positive, then the actual computation starts. After completion of the calculation, the calculated output data are also checked. Only if the output data are plausible and complete are output via the output interface. After the computations are terminated, the task goes into the idle state again, provided that, in the meantime, no new trigger was received. To terminate the task, an appropriate flag is set, and the task is then triggered. This task model is applied to all encapsulated *Safe Halt* functions in this generic form.

[130] Kampmann, A. et al.: Optimization-based Resource Allocation for ASOA (2022), p. 5.

# E Task-based Core and Interface Software Architecture for the Safe Halt Reference Solution

To use the *Safe Halt* functions in different operational environments, the functions are divided into a core and interfaces. The core of a function can be used identically in each operational environment. Specific interfaces are made available to the operational environment via interfaces. With the help of this interfaces, the individual tasks of a function can be started and stopped by the operating environment. In addition, the interfaces can be used to fill the inputs of the individual tasks in order to test the *Safe Halt* functions. For example, the entire core of the functions can be inserted directly for Software in the Loop (SiL) integration. Furthermore, the interfaces can be used to be directly integrated into different middleware. For the project UNICAR*agil* the middleware Automotive Service-Oriented Software Architecture (ASOA) is implemented. Using this approach, however, the function can also be made available in Robot Operating System (ROS) or similar middleware. Fig. E-1 shows the functional core of the task and its external interfaces.

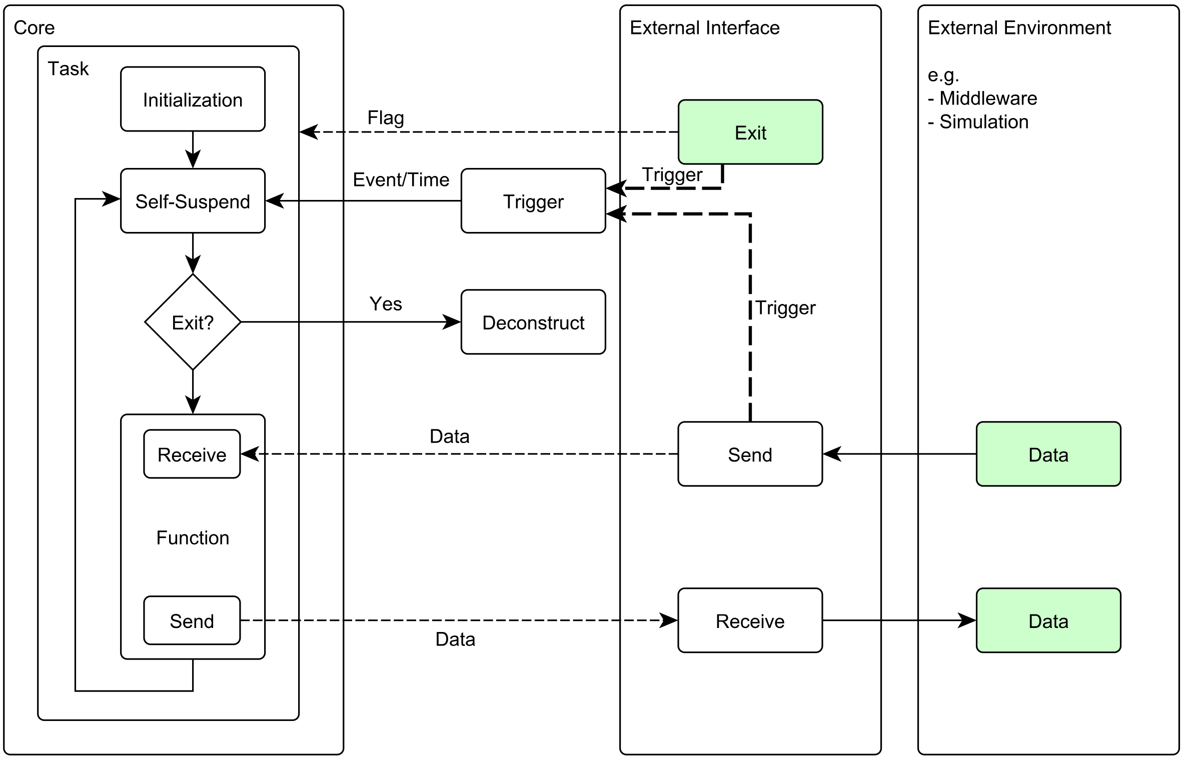


Figure E-1: Functional Architecture of a Task with External Interfaces and Environments

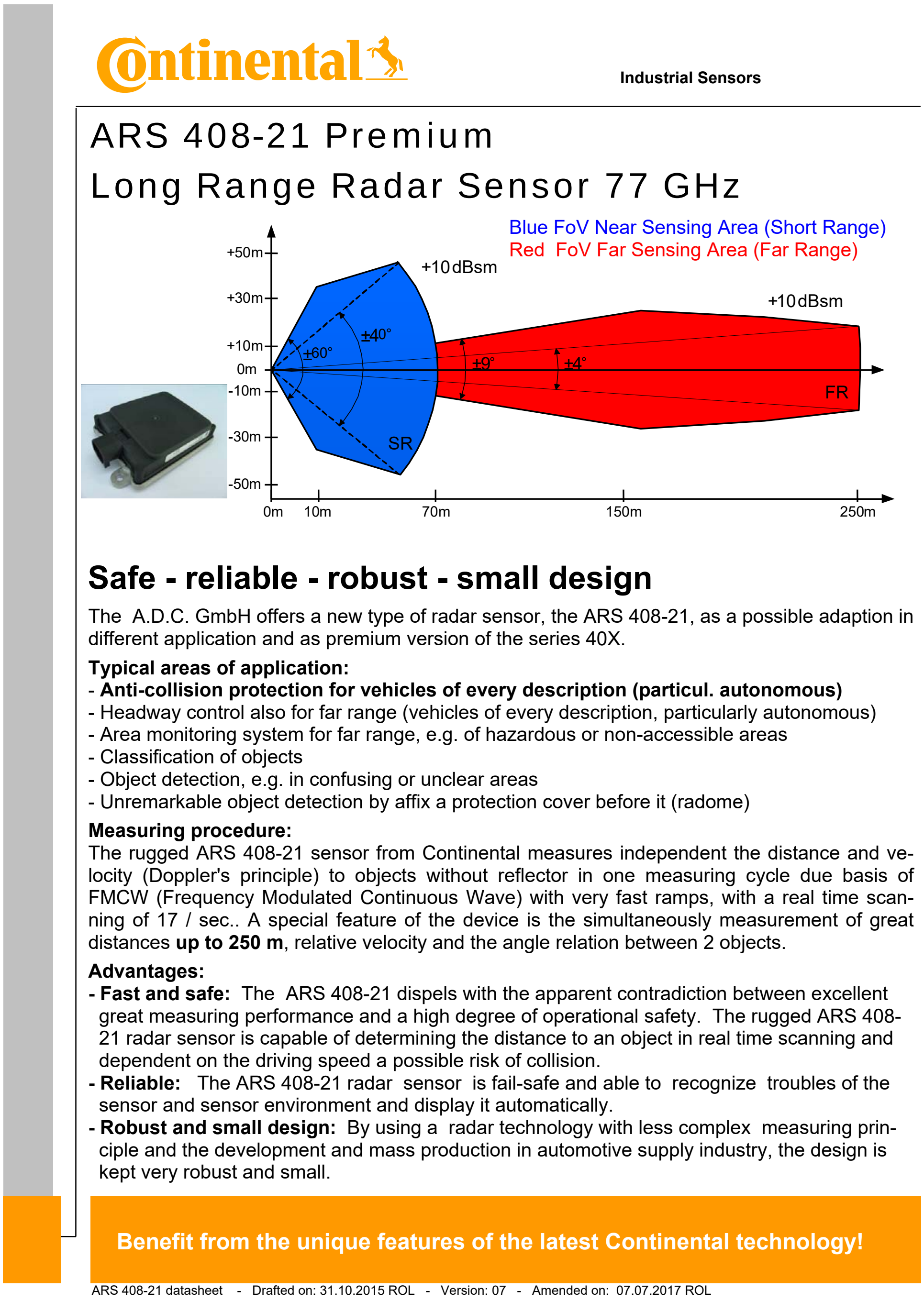
The core of the function can be started via the environment-independent interfaces and also be terminated again using the *Exit* variable. The environment communicates with the functionality core via the environment-independent functions *Send* and *Receive*. This extension of the basic task model from Annex D maximizes the coverage of the code that can be reused in different operational environments, e.g. in SiL. A change of middleware thus does not require testing of the entire code core, but shall only ensure that the interfaces are correctly addressed by the operational environment. The same applies to the operating system used. The core and the interfaces are compilable for Linux[131] as well as for Windows[132]. This is relevant, because the simulation environment IPG CarMaker™ [133] offers a more accessible support of the debugging functionality under the Windows operating system. The SiL environment with the integrated *Safe Halt* functions can thus be tested independently of the operating system.

[131] Linux Kernel Organization, Inc.: The Linux Kernel (2022).

[132] Microsoft: Microsoft Windows (2022).

[133] IPG Automotive: CarMaker (2022).

# F Continental ARS408-21 Radar Sensor



Continental
+50m
+30m
+10m
0m
-10m
-30m
-50m
0m
10m
70m
150m
250m
+10dBsm
+10dBsm
±60°
±40°
±9°
±4°
SR
FR

Industrial Sensors

## ARS 408-21 Premium
## Long Range Radar Sensor 77 GHz

Blue FoV Near Sensing Area (Short Range)
Red FoV Far Sensing Area (Far Range)

### Safe - reliable - robust - small design

The A.D.C. GmbH offers a new type of radar sensor, the ARS 408-21, as a possible adaption in different application and as premium version of the series 40X.

**Typical areas of application:**
- **Anti-collision protection for vehicles of every description (particul. autonomous)**
- Headway control also for far range (vehicles of every description, particularly autonomous)
- Area monitoring system for far range, e.g. of hazardous or non-accessible areas
- Classification of objects
- Object detection, e.g. in confusing or unclear areas
- Unremarkable object detection by affix a protection cover before it (radome)

**Measuring procedure:**
The rugged ARS 408-21 sensor from Continental measures independent the distance and velocity (Doppler's principle) to objects without reflector in one measuring cycle due basis of FMCW (Frequency Modulated Continuous Wave) with very fast ramps, with a real time scanning of 17 / sec.. A special feature of the device is the simultaneously measurement of great distances **up to 250 m**, relative velocity and the angle relation between 2 objects.

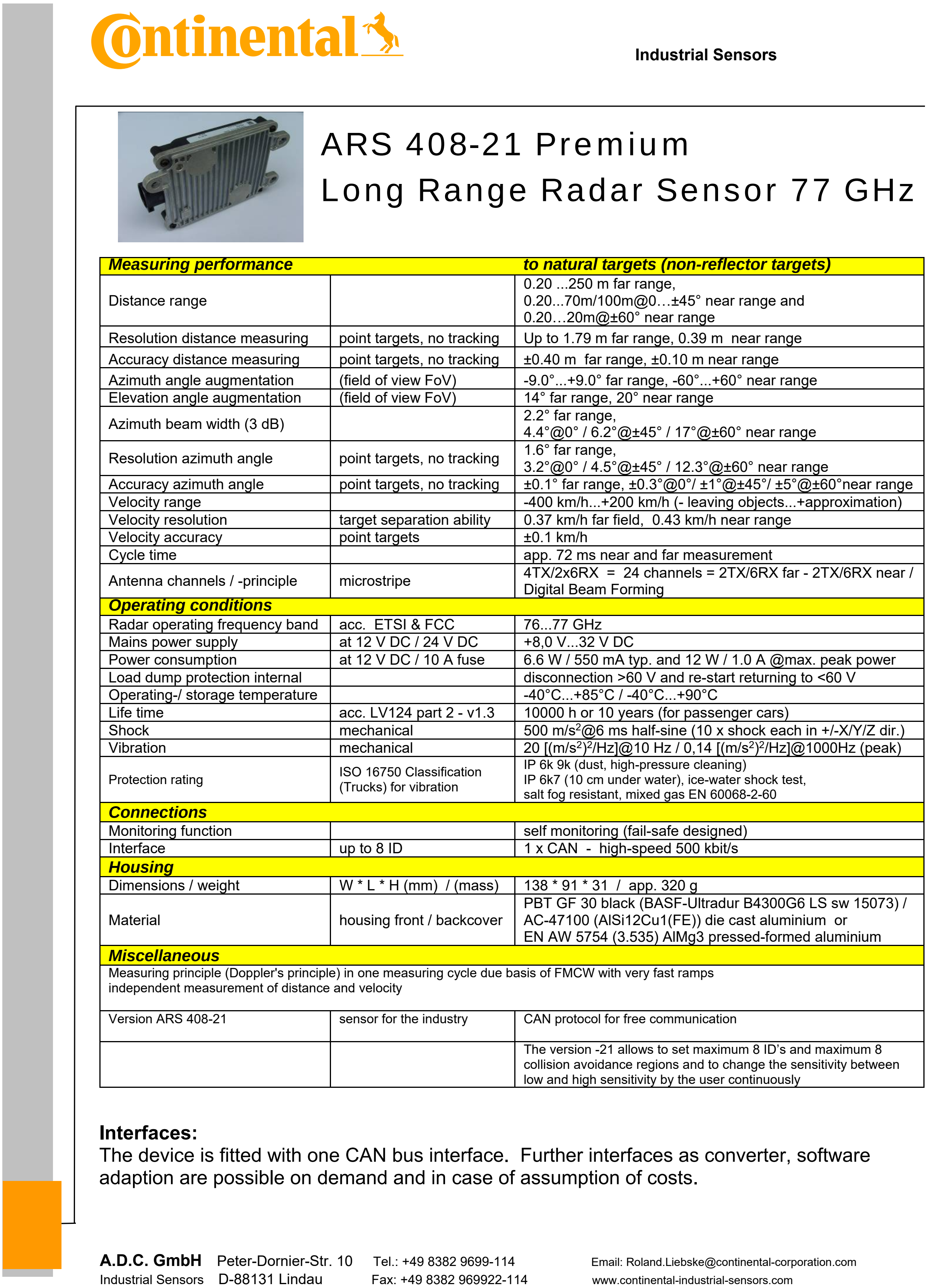
**Advantages:**
- **Fast and safe:** The ARS 408-21 dispels with the apparent contradiction between excellent great measuring performance and a high degree of operational safety. The rugged ARS 408-21 radar sensor is capable of determining the distance to an object in real time scanning and dependent on the driving speed a possible risk of collision.
- **Reliable:** The ARS 408-21 radar sensor is fail-safe and able to recognize troubles of the sensor and sensor environment and display it automatically.
- **Robust and small design:** By using a radar technology with less complex measuring principle and the development and mass production in automotive supply industry, the design is kept very robust and small.

**Benefit from the unique features of the latest Continental technology!**

ARS 408-21 datasheet - Drafted on: 31.10.2015 ROL - Version: 07 - Amended on: 07.07.2017 ROL

Figure F-1: Continental ARS408-21 Radar Sensor Data Sheet 1[134]

[134] A.D.C. GmbH: Continental ARS408-21 Data Sheet (2022)



Continental

Industrial Sensors

ARS 408-21 Premium
Long Range Radar Sensor 77 GHz

| ***Measuring performance*** | | ***to natural targets (non-reflector targets)*** |
|---|---|---|
| Distance range | | 0.20 ...250 m far range,<br>0.20...70m/100m@0…±45° near range and<br>0.20…20m@±60° near range |
| Resolution distance measuring | point targets, no tracking | Up to 1.79 m far range, 0.39 m near range |
| Accuracy distance measuring | point targets, no tracking | ±0.40 m far range, ±0.10 m near range |
| Azimuth angle augmentation | (field of view FoV) | -9.0°...+9.0° far range, -60°...+60° near range |
| Elevation angle augmentation | (field of view FoV) | 14° far range, 20° near range |
| Azimuth beam width (3 dB) | | 2.2° far range,<br>4.4°@0° / 6.2°@±45° / 17°@±60° near range |
| Resolution azimuth angle | point targets, no tracking | 1.6° far range,<br>3.2°@0° / 4.5°@±45° / 12.3°@±60° near range |
| Accuracy azimuth angle | point targets, no tracking | ±0.1° far range, ±0.3°@0°/ ±1°@±45°/ ±5°@±60°near range |
| Velocity range | | -400 km/h...+200 km/h (- leaving objects...+approximation) |
| Velocity resolution | target separation ability | 0.37 km/h far field, 0.43 km/h near range |
| Velocity accuracy | point targets | ±0.1 km/h |
| Cycle time | | app. 72 ms near and far measurement |
| Antenna channels / -principle | microstripe | 4TX/2x6RX = 24 channels = 2TX/6RX far - 2TX/6RX near / Digital Beam Forming |
| ***Operating conditions*** | | |
| Radar operating frequency band | acc. ETSI & FCC | 76...77 GHz |
| Mains power supply | at 12 V DC / 24 V DC | +8,0 V...32 V DC |
| Power consumption | at 12 V DC / 10 A fuse | 6.6 W / 550 mA typ. and 12 W / 1.0 A @max. peak power |
| Load dump protection internal | | disconnection >60 V and re-start returning to <60 V |
| Operating-/ storage temperature | | -40°C...+85°C / -40°C...+90°C |
| Life time | acc. LV124 part 2 - v1.3 | 10000 h or 10 years (for passenger cars) |
| Shock | mechanical | 500 m/s$^2$@6 ms half-sine (10 x shock each in +/-X/Y/Z dir.) |
| Vibration | mechanical | 20 [(m/s$^2$)$^2$/Hz]@10 Hz / 0,14 [(m/s$^2$)$^2$/Hz]@1000Hz (peak) |
| Protection rating | ISO 16750 Classification (Trucks) for vibration | IP 6k 9k (dust, high-pressure cleaning)<br>IP 6k7 (10 cm under water), ice-water shock test,<br>salt fog resistant, mixed gas EN 60068-2-60 |
| ***Connections*** | | |
| Monitoring function | | self monitoring (fail-safe designed) |
| Interface | up to 8 ID | 1 x CAN - high-speed 500 kbit/s |
| ***Housing*** | | |
| Dimensions / weight | W * L * H (mm) / (mass) | 138 * 91 * 31 / app. 320 g |
| Material | housing front / backcover | PBT GF 30 black (BASF-Ultradur B4300G6 LS sw 15073) /<br>AC-47100 (AlSi12Cu1(FE)) die cast aluminium or<br>EN AW 5754 (3.535) AlMg3 pressed-formed aluminium |
| ***Miscellaneous*** | | |
| Measuring principle (Doppler's principle) in one measuring cycle due basis of FMCW with very fast ramps<br>independent measurement of distance and velocity | | |
| Version ARS 408-21 | sensor for the industry | CAN protocol for free communication |
| | | The version -21 allows to set maximum 8 ID's and maximum 8 collision avoidance regions and to change the sensitivity between low and high sensitivity by the user continuously |

**Interfaces:**
The device is fitted with one CAN bus interface. Further interfaces as converter, software adaption are possible on demand and in case of assumption of costs.

**A.D.C. GmbH** Peter-Dornier-Str. 10 Tel.: +49 8382 9699-114 Email: Roland.Liebske@continental-corporation.com
Industrial Sensors D-88131 Lindau Fax: +49 8382 969922-114 www.continental-industrial-sensors.com

Figure F-2: Continental ARS408-21 Radar Sensor Data Sheet 2[134]

# G Continental ARS408-21 Radar Sensor Object List

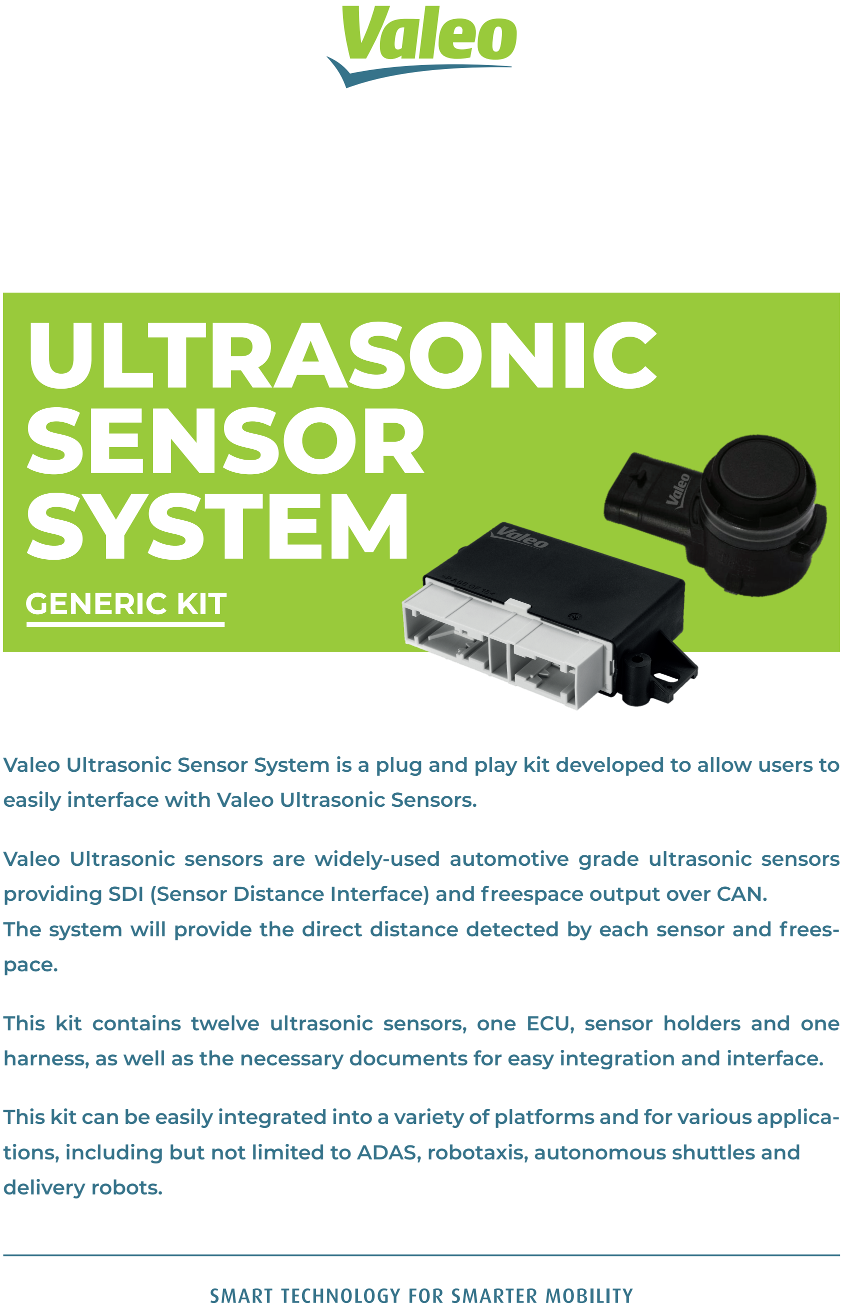
In Tab. G-1 the relevant used object states of the Continental ARS 408-21 radar sensors are presented.

Table G-1: Continental ARS408-21 Radar Sensor Object List Content

| Object State | Unit | Description |
|---|---|---|
| DistLong | m | Longitudinal distance |
| DistLat | m | Lateral distance |
| VrelLong | $\mathrm{m\,s^{-1}}$ | Relative velocity in longitudinal direction |
| VrelLat | $\mathrm{m\,s^{-1}}$ | Relative velocity in lateral direction. 0 $\mathrm{m\,s^{-1}}$ for moving objects |
| ArelLong | $\mathrm{m\,s^{-2}}$ | Relative acceleration in longitudinal direction |
| ArelLat | $\mathrm{m\,s^{-2}}$ | Relative acceleration in lateral direction. 0 $\mathrm{m\,s^{-1}}$ for moving objects |
| DynProp | | Dynamic property of the object |
| DistLong_rms | m | Standard deviation of Longitudinal distance |
| DistLat_rms | m | Standard deviation of Lateral distance |
| VrelLong_rms | $\mathrm{m\,s^{-1}}$ | Standard deviation of Relative velocity in longitudinal direction |
| VrelLat_rms | $\mathrm{m\,s^{-1}}$ | Standard deviation of Relative velocity in lateral direction |
| ArelLong_rms | $\mathrm{m\,s^{-2}}$ | Standard deviation of Relative acceleration in longitudinal direction |
| ArelLat_rms | $\mathrm{m\,s^{-2}}$ | Standard deviation of Relative acceleration in lateral direction |
| Length | m | Length of the tracked object |
| Width | m | Width of the tracked object |

The objects on the radar sensor list have a spatial extent in length and width. If the length and width cannot be measured by radar measurements, the radar sensor sets both values to the default value 1 m.

# H Valeo Sideways Ultrasonic Sensors



ULTRASONIC SENSOR SYSTEM

GENERIC KIT

Valeo Ultrasonic Sensor System is a plug and play kit developed to allow users to easily interface with Valeo Ultrasonic Sensors.

Valeo Ultrasonic sensors are widely-used automotive grade ultrasonic sensors providing SDI (Sensor Distance Interface) and freespace output over CAN.
The system will provide the direct distance detected by each sensor and freespace.

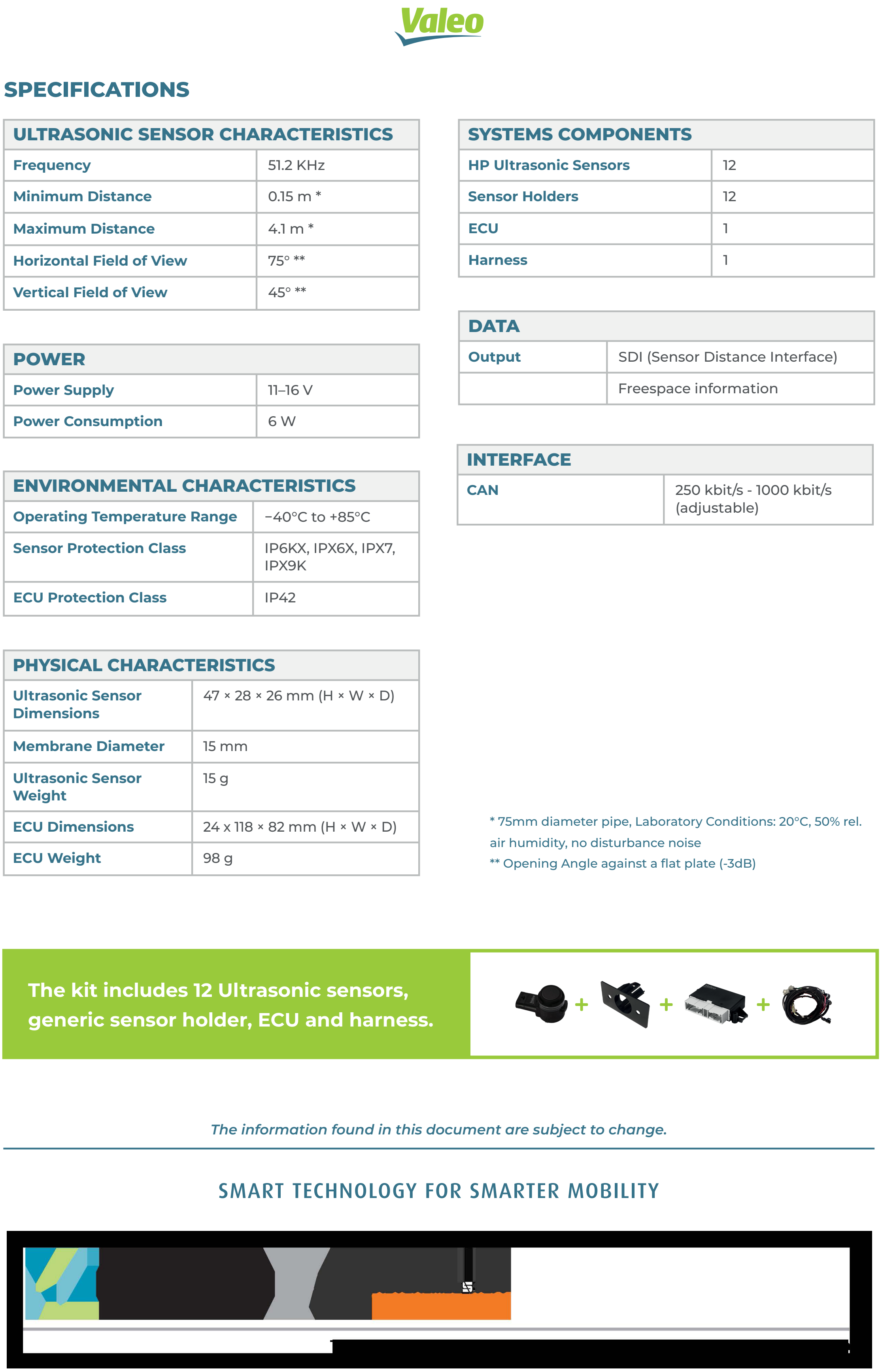
This kit contains twelve ultrasonic sensors, one ECU, sensor holders and one harness, as well as the necessary documents for easy integration and interface.

This kit can be easily integrated into a variety of platforms and for various applications, including but not limited to ADAS, robotaxis, autonomous shuttles and delivery robots.

SMART TECHNOLOGY FOR SMARTER MOBILITY

Figure H-1: Valeo Sideways Ultrasonic Sensor Data Sheet 1[135]

[135] www.AutonomousStuff.com: Valeo Ultrasonic Sensor Data Sheet (2022)



Valeo

## SPECIFICATIONS

| ULTRASONIC SENSOR CHARACTERISTICS | |
|---|---|
| Frequency | 51.2 KHz |
| Minimum Distance | 0.15 m * |
| Maximum Distance | 4.1 m * |
| Horizontal Field of View | 75° ** |
| Vertical Field of View | 45° ** |

| POWER | |
|---|---|
| Power Supply | 11–16 V |
| Power Consumption | 6 W |

| ENVIRONMENTAL CHARACTERISTICS | |
|---|---|
| Operating Temperature Range | −40°C to +85°C |
| Sensor Protection Class | IP6KX, IPX6X, IPX7, IPX9K |
| ECU Protection Class | IP42 |

| PHYSICAL CHARACTERISTICS | |
|---|---|
| Ultrasonic Sensor Dimensions | 47 × 28 × 26 mm (H × W × D) |
| Membrane Diameter | 15 mm |
| Ultrasonic Sensor Weight | 15 g |
| ECU Dimensions | 24 x 118 × 82 mm (H × W × D) |
| ECU Weight | 98 g |

| SYSTEMS COMPONENTS | |
|---|---|
| HP Ultrasonic Sensors | 12 |
| Sensor Holders | 12 |
| ECU | 1 |
| Harness | 1 |

| DATA | |
|---|---|
| Output | SDI (Sensor Distance Interface) |
| | Freespace information |

| INTERFACE | |
|---|---|
| CAN | 250 kbit/s - 1000 kbit/s (adjustable) |

* 75mm diameter pipe, Laboratory Conditions: 20°C, 50% rel. air humidity, no disturbance noise

** Opening Angle against a flat plate (-3dB)

**The kit includes 12 Ultrasonic sensors, generic sensor holder, ECU and harness.**

*The information found in this document are subject to change.*

SMART TECHNOLOGY FOR SMARTER MOBILITY

Figure H-2: Valeo Sideways Ultrasonic Sensor Data Sheet 2[135]

# I Velocity Profiles for the Cases of the Decision Tree for Velocity Adaptation

## Object Deceleration I

In Fig. I-1 such a situation is illustrated. The Automated Vehicle (AV) is in the green state. An object was detected in the red state. The required deceleration is greater than the maximum deceleration allows. For this reason, a substitute state is calculated to adjust the vehicle speed. It is located on a parallel line through the speed of the object. The remaining part of the velocity profile assumes a constant velocity of the object and eventually follows the original velocity profile to the end of the implicit emergency trajectory. If the object is not a false positive detection, this velocity profile will result in a collision.

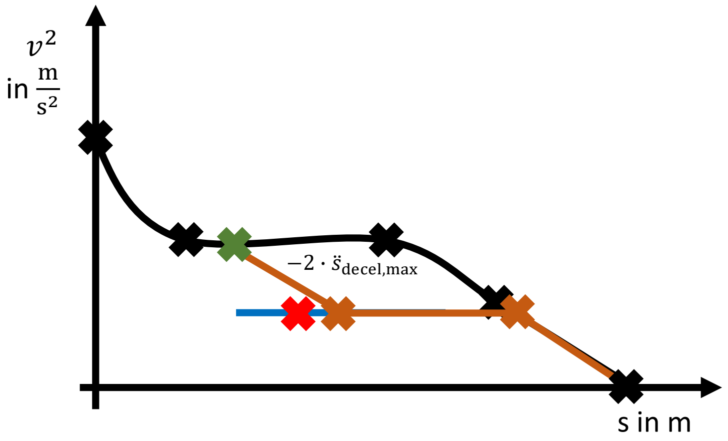


Figure I-1: Velocity Profile with High Necessary Deceleration

## Object Deceleration II

Fig. I-2 demonstrates this situation. Again, the AV is in the green state. An object was detected in the red state. In this situation, either the required deceleration is greater than the minimum allowed deceleration and less than the maximum allowed deceleration, or the vehicle is within a reaction distance $\Delta s_{\mathrm{obj}}$ from the object. Again, it is assumed that the object is moving with constant speed on the implicit emergency trajectory.

## Object No Reaction

Fig. I-3 demonstrates this situation. Again, the AV is in the green state. An object was detected in the red state. In this situation, the necessary deceleration is smaller than the minimum allowed deceleration and the distance to the object is sufficiently large. In this case, the object is ignored for updating the velocity profile until the object becomes relevant for deceleration.

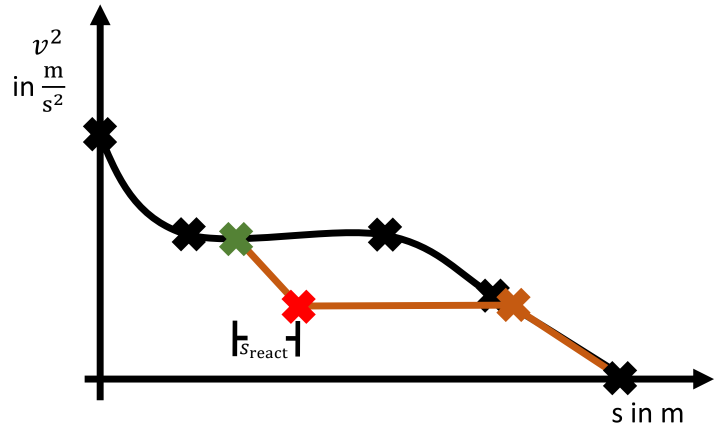


Figure I-2: Velocity Profile with Default Necessary Deceleration

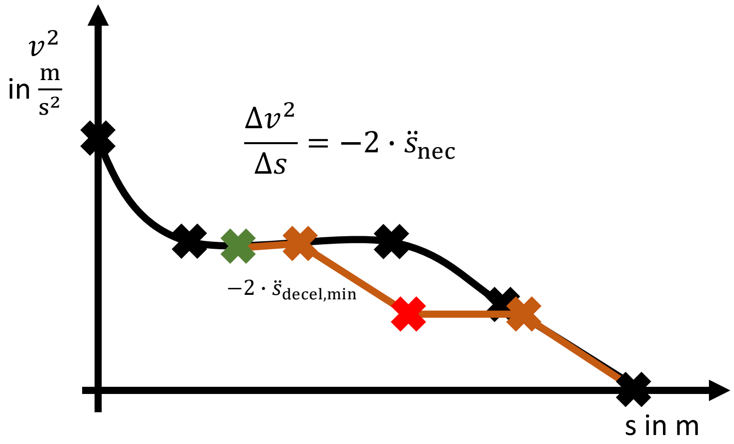

$v^2$ in $\frac{\mathrm{m}}{\mathrm{s}^2}$
$\frac{\Delta v^2}{\Delta s} = -2 \cdot \ddot{s}_{\mathrm{nec}}$
$-2 \cdot \ddot{s}_{\mathrm{decel,min}}$
s in m

Figure I-3: Velocity Profile with Low Necessary Deceleration

### Object Acceleration I

On Fig. I-4 this situation is depicted. Again, the AV is in the green state. An object was detected in the red state. From the original implicit emergency trajectory velocity profile, the maximum velocity at location $s_{\mathrm{obj}}$ of the object is also calculated. This is drawn in blue. Since this speed is greater than the object allows in this case, the object speed is used. For the update of the velocity profile, a constant object velocity is again assumed.

### Object Acceleration II

Fig. I-5 demonstrates this situation. Again, the AV is in the green state. An object was detected in the red state. The necessary acceleration is greater than the maximum permissible acceleration. For this reason, a substitute state is calculated to adjust the vehicle speed. It is located on a parallel line through the speed of the object. Again, this substitute state is compared with the maximum speed profile of the implicit emergency trajectory. In this case, the maximum speed profile is higher than the substitute state. For this reason, the substitute state is integrated into the updated speed profile.

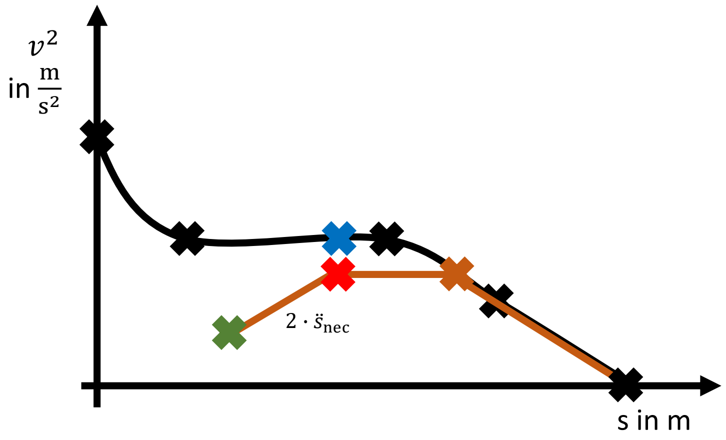

$v^2$ in $\frac{\mathrm{m}}{\mathrm{s}^2}$
$2 \cdot \ddot{s}_{\mathrm{nec}}$
s in m

Figure I-4: Velocity Profile with Default Necessary Acceleration

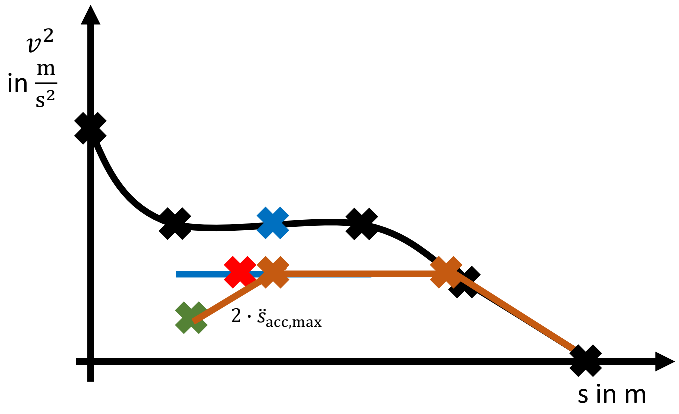

$v^2$ in $\frac{\mathrm{m}}{\mathrm{s}^2}$
$2 \cdot \ddot{s}_{\mathrm{acc,max}}$
s in m

Figure I-5: Velocity Profile with High Necessary Acceleration

Finally, Fig. I-6 shows an acceleration situation where the object has a higher velocity than the velocity profile of the implicit emergency trajectory allows. So instead of integrating the object state into the updated velocity profile, the maximum velocity of the original velocity profile valid at the location $s_{\mathrm{obj}}$ of the object is selected, shown here in blue.

### Free No Reaction

In this case, all subsequent velocity samples of the original velocity profile are used.

### Free Acceleration

The situations are similar to the accelerations in Fig. I-5 and Fig. I-6.

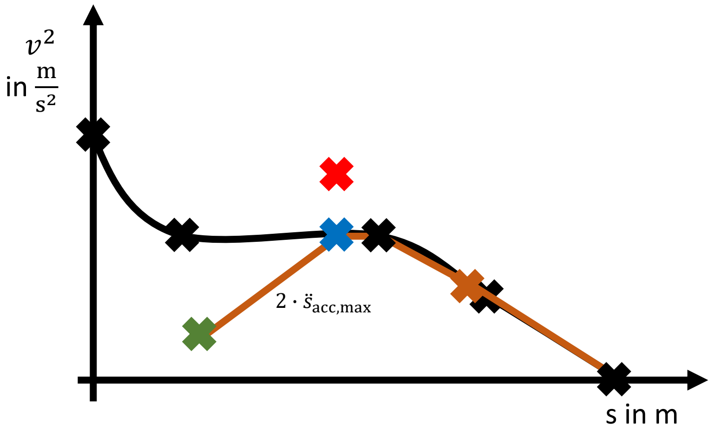


Figure I-6: Velocity Profile with Default Necessary Acceleration and Maximum Velocity Profile

# Bibliography

**A.D.C. GmbH: Continental ARS408-21 Data Sheet (2022)**
A.D.C. GmbH: ARS 408-21 Premium Long Range Radar Sensor 77 GHz, URL: https://www.continental-automotive.com/getattachment/5430d956-1ed7-464b-afa3-cd9cdc98ad63/ARS408-21_datasheet_en_170707_V07.pdf.pdf, 2022, visited on 11/29/2022

**Advanced Micro Devices, Inc: Zynq UltraScale+ MPSoC (2022)**
Advanced Micro Devices, Inc: Zynq UltraScale+ MPSoC, in: Xilinx, URL: https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html, 2022, visited on 11/04/2022

**Amazon Web Services: FreeRTOS (2022)**
Amazon Web Services: FreeRTOS - Market leading RTOS (Real Time Operating System) for embedded systems with Internet of Things extensions, in: FreeRTOS, Amazon Web Services, URL: https://www.freertos.org/index.html, 2022, visited on 09/14/2022

**AMD Xilinx: Zynq UltraScale+ EG Figure (2022)**
AMD Xilinx: Zynq UltraScale+ EG, AMD Xilinx, URL: https://www.xilinx.com/content/xilinx/en/products/silicon-devices/soc/zynq-ultrascale-mpsoc/_jcr_content/root/imageTabParsys/childParsys-productAdvantages/xilinxcolumns/childParsys-1/xilinximage.img.png/1615828221284.png, 2022, visited on 09/09/2022

**Ameling, C.: The electronic copilot for an autonomous vehicle (1999)**
Ameling, C.: The electronic copilot for an autonomous vehicle: design and first results, in: Proceedings 199 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems (Cat. No.99TH8383), pp. 521–526, 1999

**AUTOSAR: AUTOSAR (2022)**
AUTOSAR: AUTOSAR, AUTOSAR, URL: https://www.autosar.org/, 2022, visited on 09/14/2022

**Betz, A.: Feasibility analysis and design of WMDS (2015)**
Betz, Alexander: Feasibility analysis and design of wheeled mobile driving simulators for urban traffic simulation, Fortschritt-Berichte VDI Reihe 12, Verkehrstechnik, Fahrzeugtechnik, Als Ms. gedr. Edition, VDI-Verlag, 2015

**Beyerer, J. et al.: Fail-Safe Emergency Stopping for AV (2019)**
Beyerer, J.; Doll, J.; Duerr, F.; Flad, M.; Frey, M.; Gauterin, F.; Hohmann, S.; Knoch, E.; Kohlhaas, R.; Lauber, A.; Pistorius, F.; Roschani, M.; Ruf, M.; Sax, E.; Strasser, S.; Ziehn, J.: General Fail-Safe Emergency Stopping for Highly Automated Vehicles, in: (Hrsg.): Tagung Automatisiertes Fahren 2019, 2019

**Binfet-Kull, M. et al.: System safety for an autonomous driving vehicle (1998)**
Binfet-Kull, Maria; Heitmann, Peter; Ameling, Christian: System safety for an autonomous driving vehicle, in: IEEE International Conference on Intelligent Vehicles. Proceedings of the 1998 IEEE International Conference on Intelligent Vehicles, vol. 2, 1998

**Brüdigam, T. et al.: Stochastic MPC with a Safety Guarantee (2022)**
Brüdigam, Tim; Olbrich, Michael; Wollherr, Dirk; Leibold, Marion: Stochastic Model Predictive Control with a Safety Guarantee for Automated Driving, in: IEEE Transactions on Intelligent Vehicles, 2022

**Buchholz, M. et al.: Automation of the UNICARagil vehicles (2020)**
Buchholz, Michael; Gies, Fabian; Danzer, Andreas; Henning, Matti; Hermann, Charlotte; Herzog, Manuel; Horn, Markus; Schön, Markus; Rexin, Nils; Dietmayer, Klaus; Fernandez, Carlos; Janosovits, Johannes; Kamran, Danial; Kinzig, Christian; Lauer, Martin; Molinos, Eduardo; Stiller, Christoph; Ackermann, Stefan; Homolla, Tobias; Winner, Hermann; Gottschalg, Grischa; Leinen, Stefan; Becker, Matthias; Feiler, Johannes; Hoffmann, Simon; Diermeyer, Frank; Lampe, Bastian; Beemelmanns, Till; Van Kempen, Raphael; Woopen, Timo; Eckstein, Lutz; Voget, Nicolai; Moormann, Dieter; Jatzkowski, Inga; Stolte, Torben; Maurer, Markus; Graf, Jürgen; Hinüber, Edgar Leuer Von; Siepenkötter, Norbert: Automation of the UNICARagil vehicles, in: 29th Aachen Colloquium Sustainable Mobility 2020, 2020

**Charles Karney: GeographicLib documentation (2022)**
Charles Karney: GeographicLib GeographicLib documentation, URL: https://geographiclib.sourceforge.io/, 2022, visited on 10/27/2022

**Chi, E. et al.: Vehicle System For Determining a Pullover Spot (2020)**
Chi, Emily; Andrade, Ryan Joseph: Autonomous Vehicle System For Determining a Pullover Spot In Response To Detected Local Failure, 2020

**CONTINENTAL ENGINEERING SERVICES: ARS 408-21 Website (2022)**
CONTINENTAL ENGINEERING SERVICES: ARS 408-21, in: Continental Engineering Services, URL: https://conti-engineering.com/components/ars-408/, 2022, visited on 10/28/2022

**Dickmanns, E. et al.: Autonomous High Speed Road Vehicle Guidance (1987)**
Dickmanns, E.D.; Zapp, A.: Autonomous High Speed Road Vehicle Guidance by Computer Vision 1, in: IFAC Proceedings Volumes, Vol. 20, pp. 221–226, 1987

**Dochow, G. et al.: Automated driving with a safe stop point (2020)**
Dochow, Gerhard; Bieger, Stefan; GmbH, Continental Automotive: Method and device for automated driving with a safe stop point, URL: https://patents.google.com/patent/US20200012276A1/en, 2020

**Donges, E.: Three Level DDT Structure (1999)**
Donges, Edmund: A Conceptual Framework for Active Safety in Road Traffic, in: Vehicle System Dynamics, 1999

**Dorff, S. v. et al.: A Fail-safe Architecture for Automated Driving (2020)**
Dorff, Sebastian vom; Boddeker, Bert; Kneissl, Maximilian; Franzle, Martin: A Fail-safe Architecture for Automated Driving, in: 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 828–833, 2020

**Duerr, F. et al.: Global optimization of a Fail-Safe Emergency Stop Maneuver (2020)**
Duerr, F.; Ziehn, J.; Kohlhaas, R.; Roschani, M.; Ruf, M.; Beyerer, J.: Realtime Global optimization of a Fail-Safe Emergency Stop Maneuver for Arbitrary Electrical/ Electronical Failures in Automated Driving, in: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), pp. 1–8, 2020

**Emzivat, Y. et al.: Dynamic driving task fallback for an ads (2017)**
Emzivat, Yrvann; Ibanez-Guzman, Javier; Martinet, Philippe; Roux, Olivier H.: Dynamic driving task fallback for an automated driving system whose ability to monitor the driving environment has been compromised, in: 2017 IEEE Intelligent Vehicles Symposium (IV), pp. 1841–1847, 2017

**European Parliament and the Council: EU Regulation - Type-approval of ADS (2022)**
European Parliament and the Council: Regulation (EU) 2019/2144: Type-approval of the automated driving system (ADS) of fully automated vehicles, URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=PI_COM:Ares(2022)2667391, 2022, visited on 08/01/2022

**Experts from OICA/CLEPA: Definition of Minimal Risk Manoeuvre (2015)**
Experts from OICA/CLEPA: Industry proposal about Minimal risk manoeuvre, Automatically Commanded Steering Function (ACSF) - Experts from OICA/CLEPA, URL: https://wiki.unece.org/download/attachments/26902713/ACSF-03-15%20-%20%28OICA-CLEPA%29%20Minimal%20risk%20manoeuvre.pdf?api=v2, 2015, visited on 08/02/2022

**Glatzki, F. et al.: Behavioral Attributes for (BSSD) (2021)**
Glatzki, Felix; Lippert, Moritz; Winner, Hermann: Behavioral Attributes for a Behavior-Semantic Scenery Description (BSSD) for the Development of Automated Driving Functions, in: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), pp. 667–672, 2021

**GoogleWatchBlog-Team: VW.OS (2019)**
GoogleWatchBlog-Team: Android Auto: Volkswagen entwickelt Google-freies 'VW OS' auf Android-Basis - Google fordert zu viele Daten - GWB, GoogleWatchBlog-Team, URL: https://www.googlewatchblog.de/2019/09/android-auto-google-volkswagen/, 2019, visited on 09/14/2022

**Gottschalg, G.: Data Fusion Architecture with Integrity Monitoring (2022)**
Gottschalg, Grischa: Data Fusion Architecture with Integrity Monitoring for State Estimation in Automated Driving, phdthesis, TU Darmstadt, 2022

**Gottschalg, G. et al.: Integrity Based Data Fusion of Redundant Fusion Filters (2021)**
Gottschalg, Grischa; Becker, Matthias; Leinen, Stefan: Integrity Based Data Fusion of Redundant Fusion Filters for Vehicle Dynamic State Estimation in Automated Driving, in: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), pp. 3759–3765, 2021

**Gottschalg, G. et al.: Integrity Concept for Sensor Fusion Algorithms (2020)**
Gottschalg, Grischa; Becker, Matthias; Leinen, Stefan: Integrity Concept for Sensor Fusion Algorithms used in a Prototype Vehicle for Automated Driving, in: 2020 European Navigation Conference (ENC), pp. 1–10, 2020

**Heck, J.: Unfallfolgenminderung im Querverkehr (2015)**
Heck, Julian: Grundlagen für ein bordautonomes Handlungskonzept zur Unfallfolgenminderung im Querverkehr, phdthesis, Universitätsbibliothek Braunschweig, 2015

**HMS Industrial Networks: Ixxat FRC-EP170 Website (2022)**
HMS Industrial Networks: Ixxat FRC-EP170, URL: https://www.ixxat.com/de/produkte/automotive-loesungen/uebersicht/embedded-plattform/frc-ep-170?ordercode=1.01.0142.00000, 2022, visited on 10/28/2022

**Homolla, T.: Gekapselte Trajektorienfolgeregelung für autonomes Fahren (2023)**
Homolla, Tobias: Gekapselte Trajektorienfolgeregelung für autonomes Fahren, phdthesis, TU Darmstadt, 2023

**Homolla, T. et al.: Verfahren zur Korrektur von inkonsistenten Lokalisierungsdaten (2021)**
Homolla, Tobias; Gottschalg, Grischa; Winner, Hermann: Verfahren zur Korrektur von inkonsistenten Lokalisierungsdaten in modularen technischen Systemen, in: Uni-DAS 13. Workshop Fahrerassistenz und automatisiertes Fahren. FAS 2020, 2021

**Homolla, T. et al.: Encapsulated trajectory tracking control (2022)**
Homolla, Tobias; Winner, Hermann: Encapsulated trajectory tracking control for autonomous vehicles, in: Automotive and Engine Technology, 2022

**Hoppen, F.: Methodik zur Identifikation von sicheren Orten für Nothaltemanöver (2022)**
Hoppen, Fabian: Entwicklung einer Methodik zur Identifikation von sicheren Orten für Nothaltemanöver fahrerloser Fahrzeuge, 2022

**Hörwick, M. A.: Sicherheitskonzept für hochautomatisierte Fahrerassistenzsysteme (2011)**
Hörwick, Markus A.: Sicherheitskonzept für hochautomatisierte Fahrerassistenzsysteme, phdthesis, Technische Universität München, 2011

**International Electrotechnical Commission: Definition for "Trajectory" (2022)**
International Electrotechnical Commission: IEC 60050 - International Electrotechnical Vocabulary - Details for IEV number 351-41-10: "trajectory", URL: https://www.electropedia.org/iev/iev.nsf/display?openform&ievref=351-41-10, 2022, visited on 11/06/2022

**International Organization for Standardization (ISO): ISO 26262:2018 (2018)**
International Organization for Standardization (ISO): ISO 26262:2018: Road Vehicles - Functional Safety, 2018

**International Organization for Standardization (ISO): ISO/PAS 21448:2019 (2019)**
International Organization for Standardization (ISO): ISO/PAS 21448:2019: Road vehicles Safety of the intended functionality, 2019

**IPG Automotive: CarMaker (2022)**
IPG Automotive: CarMaker | IPG Automotive, CarMaker, URL: https://ipg-automotive.com/de/produkte-loesungen/software/carmaker/, 2022, visited on 09/14/2022

**Jatzkowski, I. et al.: Vehicle Operating Mode Management in ASOA (2021)**
Jatzkowski, Inga; Stolte, Torben; Graubohm, Robert; Maurer, Markus; Kampmann, Alexandru; Alrifaee, Bassam; Kowalewski, Stefan; Buchholz, Michael; Dietmayer, Klaus: Integration of a Vehicle Operating Mode Management into UNICARagils Automotive Service-oriented Software Architecture, 2021

**jesmb.de: MB.OS (2021)**
jesmb.de: MB.OS, jesmb.de, URL: https://jesmb.de/7905/, 2021, visited on 09/14/2022

**Kampmann, A. et al.: Automotive Service-Oriented Software Architecture (2019)**
Kampmann, Alexandru; Alrifaee, Bassam; Kohout, Markus; Wustenberg, Andreas; Woopen, Timo; Nolte, Marcus; Eckstein, Lutz; Kowalewski, Stefan: A Dynamic Service-Oriented Software Architecture for Highly Automated Vehicles, in: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pp. 2101–2108, 2019

**Kampmann, A. et al.: Optimization-based Resource Allocation for ASOA (2022)**
Kampmann, Alexandru; Luer, Maximilian; Kowalewski, Stefan; Alrifaee, Bassam: Optimization-based Resource Allocation for an Automotive Service-oriented Software Architecture, in: 2022 IEEE Intelligent Vehicles Symposium (IV), pp. 678–687, 2022

**Karakaya, B. et al.: Driver Behavior During Minimal Risk Maneuvers (2021)**
Karakaya, Burak; Bengler, Klaus: Investigation of Driver Behavior During Minimal Risk Maneuvers of Automated Vehicles, in: Black, Nancy L.; Neumann, W. Patrick; Noy, Ian (Hrsg.): Proceedings of the 21st Congress of the International Ergonomics Association (IEA 2021), Springer International Publishing, 2021

**Karakaya, B. et al.: A Video Survey on MRM and MRC (2020)**
Karakaya, Burak; Kalb, Luis; Bengler, Klaus: A Video Survey on Minimal Risk Maneuvers and Conditions, in: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, Vol. 64, pp. 1708–1712, 2020

**Karney, C. F. F.: Algorithms for geodesics (2013)**
Karney, Charles F. F.: Algorithms for geodesics, in: Journal of Geodesy, 2013

**Kazemi, M. et al.: Autonomous Vehicle Safe Stop (2019)**
Kazemi, Moslem; Bardapurkar, Sameer; LLC, Uatc: Autonomous Vehicle Safe Stop, URL: https://patents.google.com/patent/US20190235499A1/en, 2019

**Keilhoff, D. et al.: UNICARagil New architectures vehicle concepts (2019)**
Keilhoff, Dan; Niedballa, Dennis; Reuss, Hans-Christian; Buchholz, Michael; Gies, Fabian; Dietmayer, Klaus; Lauer, Martin; Stiller, Christoph; Ackermann, Stefan; Winner, Hermann; Kampmann, Alexandru; Alrifaee, Bassam; Kowalewski, Stefan; Klein, Fabian; Struth, Michael; Woopen, Timo; Eckstein, Lutz: UNICARagil New architectures for disruptive vehicle concepts, in: Bargende, Michael; Reuss, Hans-Christian; Wagner, Andreas; Wiedemann, Jochen (Hrsg.): 19. Internationales Stuttgarter Symposium, Springer Fachmedien Wiesbaden, 2019

**Knowledge Based Systems, Inc.: IDEF0 Website (2022)**
Knowledge Based Systems, Inc.: IDEF Integrated DEFinition Methods (IDEF), Knowledge Based Systems, Inc. URL: https://www.idef.com/, 2022, visited on 10/03/2022

**Koopman, P.: SAE J3016 User Guide (2023)**
Koopman, Phil: SAE J3016 User Guide, URL: https://users.ece.cmu.edu/~koopman/j3016/index.html, 2023, visited on 06/15/2023

**Koopman, P. et al.: How Many ODD, Objects, and Events? (2019)**
Koopman, Philip; Fratrik, Frank: How Many Operational Design Domains, Objects, and Events?, 2019

**Koopman, Philip: AV: Standards & Open Challenges (2022)**
Koopman, Philip: Autonomous Vehicle: Standards & Open Challenges, URL: https://users.ece.cmu.edu/~koopman/lectures/L130_AV_Standards_Challenges.pdf, 2022, visited on 11/11/2022

**Krook, J. et al.: Safe Stop Supervisor for an Automated Vehicle (2019)**
Krook, Jonas; Svensson, Lars; Li, Yuchao; Feng, Lei; Fabian, Martin: Design and Formal Verification of a Safe Stop Supervisor for an Automated Vehicle, in: 2019 International Conference on Robotics and Automation (ICRA), pp. 5607–5613, 2019

**Kwon, S. et al.: Autonomous emergency stop system (2014)**
Kwon, Surim; Jung, Changyoung; Choi, Taesung; Oh, Youngchul; You, Byungyong: Autonomous emergency stop system, in: 2014 IEEE Intelligent Vehicles Symposium Proceedings, pp. 444–449, 2014

**Laur, M. H. et al.: AV SAFE -STOP-ZONE MAPPING SYSTEM (2018)**
Laur, Michael H.; Szabo, Ronald J.; Hilnbrand, Brian R.; DELPHITECHNOLOGIES, I. N.C.: AUTOMATED-VEHICLE SAFE -STOP-ZONE MAPPING SYSTEM, 2018

**Le Cornec, O.: Electronic device for determining an emergency stopping trajectory (2020)**
Le Cornec, Olivier: Electronic device for determining an emergency stopping trajectory of an autonomous vehicle, related vehicle and method, 2020

**Leonhardt, T.: Minimal Risk Maneuver (2018)**
Leonhardt, Thorsten: Minimal Risk Maneuver, in: ko-HAF Abschlussveranstaltung, URL: https://www.ko-haf.de/fileadmin/user_upload/media/abschlusspraesentation/12_Ko-HAF_Minimal-Risk-Maneuver.pdf, 2018, visited on 11/05/2022

**Linux Kernel Organization, Inc.: The Linux Kernel (2022)**
Linux Kernel Organization, Inc.: The Linux Kernel Archives, URL: https://www.kernel.org/, 2022, visited on 10/27/2022

**Magdici, S. et al.: Fail-safe motion planning (2016)**
Magdici, Silvia; Althoff, Matthias: Fail-safe motion planning of autonomous vehicles, in: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), pp. 452–458, 2016

**Matthaei, R. et al.: Autonomous driving  a top-down-approach (2015)**
Matthaei, Richard; Maurer, Markus: Autonomous driving  a top-down-approach, in: at - Automatisierungstechnik, pp. 155–167, 2015

**Maurer, M.: Flexible Automatisierung von StraSSenfahrzeugen mit Rechnersehen (2000)**
Maurer, Markus: Flexible Automatisierung von StraSSenfahrzeugen mit Rechnersehen, phdthesis, Universität der Bundeswehr München, 2000

**Maurer, M.; Gerdes, J. C.; Lenz, B.; Winner, H.: Autonomous Driving (2016)**
Maurer, Markus; Gerdes, J. Christian; Lenz, Barbara; Winner, Hermann: Autonomous Driving, Springer Berlin Heidelberg, 2016

**Mercedes-Benz: 1986: Startschuss zum PROMETHEUS-Forschungsprogramm (2007)**
Mercedes-Benz: 1986: Startschuss zum PROMETHEUS-Forschungsprogramm, URL: https://media.mercedes-benz.com/article/e7142515-d8a3-44d9-be3d-5b1cb09f05f4, 2007, visited on 06/19/2023

**Mercedes-Benz: Das Projekt PROMETHEUS ab 1986: Vorreiter des autonomen Fahrens (2016)**
Mercedes-Benz: Das Projekt PROMETHEUS ab 1986: Vorreiter des autonomen Fahrens, URL: https://media.mercedes-benz.com/article/b7bf109a-260a-4031-91b4-9d32830dfd49, 2016, visited on 06/19/2023

**micro-ROS: micro-ROS (2022)**
micro-ROS: Supported Hardware, in: micro-ROS Supported Hardware, URL: https://micro.ros.org/docs/overview/hardware/, 2022, visited on 11/04/2022

**Microsoft: Microsoft Windows (2022)**
Microsoft: Microsoft Windows - Operating System, in: Windows, URL: https://www.microsoft.com/de-de/windows, 2022, visited on 10/27/2022

**Mike1024: Local tangent plane coordinates (2010)**
Mike1024: Local tangent plane coordinates, URL: https://en.wikipedia.org/wiki/Local_tangent_plane_coordinates#/media/File:ECEF_ENU_Longitude_Latitude_relationships.svg, 2010, visited on 08/17/2022

**Mokhtarian, A. et al.: Service-oriented Software Architecture for UNICARagil (2020)**
Mokhtarian, Armin; Alrifaee, Bassam; Kampmann, Alexandru: The Dynamic Service-oriented Software Architecture for UNICARagil, in: 29th Aachen Colloquium Sustainable Mobility 2020, pp. 275–284, 2020

**Murphy, R.: Introduction to AI robotics (2000)**
Murphy, Robin: Introduction to AI robotics, MIT Press, 2000

**Niedballa, D. et al.: Concepts of functional safety in E/E-architectures (2020)**
Niedballa, Dennis; Reuss, H.-C.: Concepts of functional safety in E/E-architectures of highly automated and autonomous vehicles, in: Bargende, Michael; Reuss, Hans-Christian; Wagner, Andreas (Hrsg.): 20. Internationales Stuttgarter Symposium, Springer Fachmedien Wiesbaden, 2020

**Nilsson, N. J.: A mobile automation (1969)**
Nilsson, Nils J.: A mobile automaton: An application of artificial intelligence techniques, 1969

**Nolte, M. et al.: Supporting Safe Decision Making (2020)**
Nolte, Marcus; Jatzkowski, Inga; Ernst, Susanne; Maurer, Markus: Supporting Safe Decision Making Through Holistic System-Level Representations & Monitoring – A Summary and Taxonomy of Self-Representation Concepts for Automated Vehicles, arXiv, URL: http://arxiv.org/abs/2007.13807, 2020, visited on 09/09/2022

**O'Reilly, O. M.: Engineering Dynamics (2019)**
O'Reilly, Oliver M.: Engineering Dynamics: A Primer, 3rd ed. 2019. Edition, Springer International Publishing : Imprint: Springer, 2019

**Object Management Group: Interface Definition Language (IDL) (2022)**
Object Management Group: Interface Definition Language (IDL), Object Management Group, URL: www.omg.org/spec/IDL/About-IDL/, 2022, visited on 09/08/2022

**Open Robotics: ROS 2 Documentation (2022)**
Open Robotics: ROS 2 Documentation  ROS 2 Documentation: Humble documentation, Open Robotics, URL: https://docs.ros.org/en/humble/index.html, 2022, visited on 09/14/2022

**OpenStreetMap contributors: Accuracy  OpenStreetMap Wiki (2022)**
OpenStreetMap contributors: Accuracy  OpenStreetMap Wiki, OpenStreetMap contributors, URL: https://wiki.openstreetmap.org/wiki/Accuracy#Topology, 2022, visited on 08/21/2022

**Parseh, M. et al.: Minimizing Collision Severity for Highly Automated Vehicles (2021)**
Parseh, Masoumeh; Asplund, Fredrik; Svensson, Lars; Sinz, Wolfgang; Tomasch, Ernst; Torngren, Martin: A Data-Driven Method Towards Minimizing Collision Severity for Highly Automated Vehicles, in: IEEE Transactions on Intelligent Vehicles, Vol. 6, pp. 723–735, 2021

**Pek, C. et al.: Efficient Fail-safe Trajectory Planning (2018)**
Pek, Christian; Althoff, Matthias: Computationally Efficient Fail-safe Trajectory Planning for Self-driving Vehicles Using Convex Optimization, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 1447–1454, 2018

**Popp, C.: Safety-Check von Trajektorien beim Automatisierten Fahren (2023)**
Popp, Christoph: Simultaner Safety-Check von Trajektorien beim Automatisierten Fahren im Urbanen Verkehr, phdthesis, TU Darmstadt, 2023

**Popp, C. et al.: Approach to Maintain a Safe State of an Automated Vehicle (2022)**
Popp, Christoph; Ackermann, Stefan; Winner, Hermann: Approach to Maintain a Safe State of an Automated Vehicle in Case of Unsafe Desired Behavior, in: 14. Uni-DAS e.V. Workshop Fahrerassistenz und automatisiertes Fahren: 09. 11.05.2022, 2022

**Reschka, A. et al.: Conditions for a safe state of automated road vehicles (2015)**
Reschka, Andreas; Maurer, Markus: Conditions for a safe state of automated road vehicles, in: it - Information Technology, Vol. 57, pp. 215–222, 2015

**Ross, D.: Structured Analysis (SA) (1977)**
Ross, D.T.: Structured Analysis (SA): A Language for Communicating Ideas, in: IEEE Transactions on Software Engineering, pp. 16–34, 1977

**RWTH Aachen: UNICARagil Startseite (2022)**
RWTH Aachen: UNICARagil Startseite, RWTH Aachen, URL: www.unicaragil.de, 2022, visited on 09/08/2022

**SAE International: SAE J3016 (2021)**
SAE International: J3016 Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, 2021

**SAE International: SAE J3131 (2022)**
SAE International: J3131 Definitions for Terms Related to Automated Driving Systems Reference Architecture, 2022

**SAE International: SAE J3164 (2018)**
SAE International: J3164 (WIP) Taxonomy and Definitions for Terms Related to Automated Driving System Behaviors and Maneuvers for On-Road Motor Vehicles - SAE International, 2018

**SAE International: SAE J3259 (2022)**
SAE International: J3259 (WIP) Taxonomy & Definitions for Operational Design Domain (ODD) for Driving Automation Systems, 2022

**SAE International: SAE J670 (2022)**
SAE International: J670 Vehicle Dynamics Terminology, 2022

**Stahl, Tim Nikolaus: Safeguarding complex and learning-based automated driving functions (2022)**
Stahl, Tim Nikolaus: Safeguarding complex and learning-based automated driving functions via online verification, phdthesis, TUM School of Engineering and Design, 2022

**Statistisches Bundesamt (Destatis): Traffic fatalities in Germany 2021 (2021)**
Statistisches Bundesamt (Destatis): Verkehrstote in Deutschland 2021, in: Statistisches Bundesamt, URL: https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Verkehrsunfaelle/Tabellen/verkehrstote-nach-alter.html, 2021, visited on 11/06/2022

**Stolte, T. et al.: Taxonomy to Unify Fault Tolerance Regimes for Automotive Systems (2022)**
Stolte, Torben; Ackermann, Stefan; Graubohm, Robert; Jatzkowski, Inga; Klamann, Björn; Winner, Hermann; Maurer, Markus: Taxonomy to Unify Fault Tolerance Regimes for Automotive Systems: Defining Fail-Operational, Fail-Degraded, and Fail-Safe, in: IEEE Transactions on Intelligent Vehicles, Vol. 7, pp. 251–262, 2022

**Stolte, T. et al.: Towards Safety Concepts for Automated Vehicles (2020)**
Stolte, Torben; Graubohm, Robert; Jatzkowski, Inga; Maurer, Markus; Ackermann, Stefan Martin; Klamann, Björn; Lippert, Moritz; Winner, Hermann: Towards Safety Concepts for Automated Vehicles by the Example of the Project UNICARagil, 2020

**Stolte, T. et al.: Unmanned Protective Vehicle for Highway Hard Shoulder (2015)**
Stolte, Torben; Reschka, Andreas; Bagschik, Gerrit; Maurer, Markus: Towards Automated Driving: Unmanned Protective Vehicle for Highway Hard Shoulder Road Works, in: 2015 IEEE 18th International Conference on Intelligent Transportation Systems, pp. 672–677, 2015

**Svensson, L. et al.: Safe Stop Trajectory Planning (2018)**
Svensson, Lars; Masson, Lola; Mohan, Naveen; Ward, Erik; Brenden, Anna Pernestal; Feng, Lei; Torngren, Martin: Safe Stop Trajectory Planning for Highly Automated Vehicles: An Optimal Control Problem Formulation, in: 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 517–522, 2018

**Tesla: Full Self-Driving Computer Installations (2020)**
Tesla: Full Self-Driving Computer Installations, Tesla, URL: https://www.tesla.com/support/full-self-driving-computer, 2020, visited on 09/14/2022

**The British Standards Institution: PAS 1883:2020 (2020)**
The British Standards Institution: PAS 1883:2020: Operational design domain (ODD) taxonomy for an automated driving system (ADS) - Specification, 2020

**Thrun, S. et al.: Stanley: The robot that won the DARPA Grand Challenge (2006)**
Thrun, Sebastian; Montemerlo, Mike; Dahlkamp, Hendrik; Stavens, David; Aron, Andrei; Diebel, James; Fong, Philip; Gale, John; Halpenny, Morgan; Hoffmann, Gabriel; Lau, Kenny; Oakley, Celia; Palatucci, Mark; Pratt, Vaughan; Stang, Pascal; Strohband, Sven; Dupont, Cedric; Jendrossek, Lars-Erik; Koelen, Christian; Markey, Charles; Rummel, Carlo; Niekerk, Joe van; Jensen, Eric; Alessandrini, Philippe; Bradski, Gary; Davies, Bob; Ettinger, Scott; Kaehler, Adrian; Nefian, Ara; Mahoney, Pamela: Stanley: The robot that won the DARPA Grand Challenge, in: Journal of Field Robotics, Vol. 23, pp. 661–692, 2006

**Tsugawa, S. et al.: An Automobile with Artificial Intelligence (1979)**
Tsugawa, Sadayuki; Yatabe, Teruo; Hirose, Takeshi; Matsumoto: An Automobile with Artificial Intelligence, in: Proceedings of the Sixth International Joint Conference on Artificial Intelligence: Tokyo, August 20 - 23, 1979, pp. 893–895, 1979

**United States Government Army: Systems Engineering Fundamentals (2001)**
United States Government Army: Systems Engineering Fundamentals, 2001

**VALEO SERVICE: Valeo Ultrasonic Sensor Website (2022)**
VALEO SERVICE: Parking sensor ultrasonic technology for car, URL: https://www.valeoservice.com/en-com/passenger-car/car-sensor-and-switches/parking-sensor-ultrasonic-technology-car, 2022, visited on 10/28/2022

**Walden, D. D.; Roedler, G. J.; Forsberg, K.; Hamelin, R. D.; Shortell, T. M.: Systems engineering handbook (2015)**
Walden, David D.; Roedler, Garry J.; Forsberg, Kevin; Hamelin, R. Douglas; Shortell, Thomas M.: Systems engineering handbook: a guide for system life cycle processes and activities, 4th edition. Edition, Wiley, 2015

**Wang, L. et al.: Real-Time Safe Stop Trajectory Planning (2020)**
Wang, Lingguang; Wu, Zhenkang; Li, Jiakang; Stiller, Christoph: Real-Time Safe Stop Trajectory Planning via Multidimensional Hybrid A-Algorithm, in: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), pp. 1–7, 2020

**Woopen, T. et al.: UNICARagil - Disruptive Modular Architectures for AV (2018)**
Woopen, Timo; Eckstein, Lutz; Kowalewski, Stefan; Moormann, Dieter; Maurer, Markus; Ernst, Rolf; Winner, Hermann; Katzenbeisser, Stefan; Becker, Matthias; Stiller, Christoph; Furmans, Kai; Bengler, Klaus; Lienkamp, Markus; Reuss, Hans-Christian; Dietmayer, Klaus; Lategahn, Henning; Siepenkötter, Norbert; Elbs, Martin; V. Hinüber, Edgar; Dupuis, Marius; Hecker, Christian; Lampe, Bastian; Böddeker, Torben; Kampmann, Alexandru; Alrifaee, Bassam; Stolte, Torben; Jatzkowski, Inga; Möstl, Mischa; Ackermann, Stefan; Amersbach, Christian; Püllen, Dominik; Leinen, Stefan; Diermeyer, Frank; Keilhoff, Dan; Buchholz, Michael: UNICARagil - Disruptive Modular Architectures for Agile, Automated Vehicle Concepts, in: Volume 1 / 2018 27th Aachen Colloquium Automobile and Engine Technology 2018, pages 663–694, 2018

**www.AutonomousStuff.com: Valeo Ultrasonic Sensor Data Sheet (2022)**
www.AutonomousStuff.com: Valeo Sideways Ultrasonic Sensor Data Sheet, URL: https://autonomoustuff.com/-/media/Images/Hexagon/Hexagon%20Core/autonomousstuff/pdf/valeo-uss-can-datasheet-whitelabel.ashx, 2022, visited on 11/29/2022

**Xilinx: PetaLinux Tools (2022)**
Xilinx: PetaLinux Tools, in: PetaLinux Tools, PetaLinux Tools, URL: https://www.xilinx.com/products/design-tools/embedded-software/petalinux-sdk.html, 2022, visited on 09/14/2022

**Xue, W. et al.: An adaptive model predictive approach in fallback procedure (2019)**
Xue, Wei; Zheng, Rencheng; Yang, Bo; Wang, Zheng; Kaizuka, Tsutomu; Nakano, Kimihiko: An adaptive model predictive approach for automated vehicle control in fallback procedure based on virtual vehicle scheme, in: Journal of Intelligent and Connected Vehicles, Vol. 2, pp. 67–77, 2019

**Zimmer, H.: PROMETHEUS Forschungsprogramm zur Gestaltung des künftigen StraSSenverkehrs (1990)**
Zimmer, H.: PROMETHEUS Ein europäisches Forschungsprogramm zur Gestaltung des künftigen StraSSenverkehrs, in: Forschungsgesellschaft für StraSSen-und Verkehrswesen: StraSSenverkehrstechnik. Bd, Vol. 34, 1990

# Own Publications

Woopen, Timo; Lampe, Bastian; Böddeker, Torben; Eckstein, Lutz; Kampmann, Alexandru; Alrifaee, Bassam; Kowalewski, Stefan; Moormann, Dieter; Stolte, Torben; Jatzkowski, Inga; Maurer, Markus; Möstl, Mischa; Ernst, Rolf; **Ackermann, Stefan**; Amersbach, Christian; Leinen, Stefan; Winner, Hermann; Püllen, Dominik; Katzenbeisser, Stefan; Becker, Matthias; Stiller, Christoph; Furmans, Kai; Bengler, Klaus; Diermeyer, Frank; Lienkamp, Markus; Keilhoff, Dan; Reuss, Hans-Christian; Buchholz, Michael; Dietmayer, Klaus; Lategahn, Henning; Siepenkötter, Norbert; Elbs, Martin; Hinüber, Edgar von; Dupuis, Marius; Hecker, Christian: UNICARagil - Disruptive Modular Architectures for Agile, Automated Vehicle Concepts, in: 27th Aachen Colloquium Automobile and Engine Technology 2018, Aachen, 2018.

Keilhoff, Dan; Niedballa, Dennis; Reuss, Hans-Christian; Buchholz, Michael; Gies, Fabian; Dietmayer, Klaus; Lauer, Martin; Stiller, Christoph; **Ackermann, Stefan**; Winner, Hermann; Kampmann, Alexandru; Alrifaee, Bassam; Kowalewski, Stefan; Klein, Fabian; Struth, Michael; Woopen, Timo; Eckstein, Lutz: UNICARagil - New Architectures for Disruptive Vehicle Concepts, in: 19th Stuttgart International Symposium - Automotive and Engine Technology, 19.-20. MArch 2019, Stuttgart, 2019.

Buchholz, Michael; Gies, Fabian; Danzer, Andreas; Henning, Matti; Hermann, Charlotte; Herzog, Manuel; Horn, Markus; Schön, Markus; Rexin, Nils; Dietmayer, Klaus; Fernandez, Carlos; Janosovits, Johannes; Kamran, Danial; Kinzig, Christian; Lauer, Martin; Molinos, Eduardo; Stiller, Christoph; Wang, Lingguang; **Ackermann, Stefan**; Homolla, Tobias; Winner, Hermann; Gottschalg, Grischa; Leinen, Stefan; Becker, Mathias; Feiler, Johannes; Hoffmann, Simon; Diermeyer, Frank; Lampe, Bastian; Beemelmanns, Till; Kempen, Raphael van; Woopen, Timo; Eckstein, Lutz; Voget, Nicolai; Moormann, Dieter; Jatzkowski, Inga; Stolte, Torben; Maurer, Markus; Graf, Jürgen; Hinüber, Edgar von; Siepenköter, Norbert: Automation of the UNICARagil Vehicles, in: 29th Aachen Colloquium Sustainable Mobility 2020, Aachen, 2020.

**Ackermann, Stefan**; Winner, Hermann: Systemarchitektur und Fahrmanöver zum sicheren Anhalten modularer automatisierter Fahrzeuge, in: 13. Workshop Fahrerassistenzsysteme und automatisiertes Fahren, Walting, 2020.

Stolte, Torben; Graubohm, Robert; Jatzkowski, Inga; Maurer, Markus; **Ackermann, Stefan**; Klamann, Björn; Lippert, Moritz; Winner, Hermann: Towards Safety Concepts for Automated Vehicles by the Example of the Project UNICARagil, 29th Aachen Colloquium Sustainable Mobility 2020, Aachen, 2020.

Stolte, Torben; **Ackermann, Stefan**; Graubohm, Robert; Jatzkowski, Inga; Klamann, Bjorn; Winner, Hermann; Maurer, Markus: A Taxonomy to Unify Fault Tolerance Regimes for Automotive Systems: Defining Fail-Operational, Fail-Degraded, and Fail-Safe, in: IEEE Transactions on Intelligent Vehicles, 2021.

Popp, Christoph; **Ackermann, Stefan**; Winner, Hermann: Approach to Maintain a Safe State of an Automated Vehicle in Case of Unsafe Desired Behavior, in: 14. Workshop Fahrerassistenzsysteme und automatisiertes Fahren, Berkheim, 2022.

# Own Patents

**Ackermann, Stefan**; Winner, Hermann; Buchholz, Michael. 2021. Modul und Verfahren zur Absicherung von Solltrajektorien für automatisiertes Fahren (DE102019125401A1). Deutsches Patent- und Markenamt. https://register.dpma.de/DPMAregister/pat/register?AKZ=1020191254019

# Supervised Theses

**Klamann, Björn:** Requirement Analysis for a Safety Assessment of Modular Systems
Master-Thesis Nr. 673/17

**Lippert, Moritz:** Development of a Methodology for the Categorization of Road Sections
Master-Thesis Nr. 679/18

**Homolla, Tobias:** Development of a Concept for a 3 DoF Vehicle Dynamics Control
Master-Thesis Nr. 680/18

**Hoppen, Fabian:** Design of a Method for the Identification of Safe Spots for Emergency Halts of Driverless Vehicles
Master-Thesis Nr. 771/20

**Poudel, Pramod:** Development and Implementation of a Function for Identifying Safe Stopping Locations for Driverless Vehicles
Master-Thesis Nr. 842/21

**Lyu, Chunghyun:** Development of a methodology to evaluate the effects of mode change to "Safe Halt" mode in the UNICAR*agil* project
Master-Thesis Nr. 849/22

**Leinberger, Moritz:** Development and Implementation of an Assistance System for Multi-Directional Sideloaders for the Detection of the Entry Funnel of Narrow Aisles in Narrow Aisle Warehouses
Master-Thesis Nr. 850/22

**Kötter, Marco:** Development of a Trajectory Generation for Automated Vehicles
Bachelor-Thesis Nr. 1346/19

**Lemcke, Mathias:** Development and Implementation of a Sensor Selection Logic for Automated Vehicles
Bachelor-Thesis Nr. 1367/20

**Glaser-Gallion, Michael Wilfried:** Design and Implementation of a Human-Machine-Interface for an Automated Vehicle in the UNICAR*agil* Project
Bachelor-Thesis Nr. 1371/20

**Lu, Viet Khanh:** Concept and Design of a Human-Machine Interface for an Automated Vehicle in the UNICAR*agil* Project
Bachelor-Thesis Nr. 1374/20

**Muth, Alexander:** Development and Implementation of a Test Environment for the Emergency Stop Function Safe Halt in the Project UNICAR*agil*
Bachelor-Thesis Nr. 1396/21

**Hugo, Thomas:** Development and Implementation of an Ultrasonic Sensor Feature Extraction Method for the Emergency Stop Function Safe Halt in the Project UNICAR*agil*
Bachelor-Thesis Nr. 1398/22

**Aghadavoodi, Erfan:** Development and Implementation of a Function for Identifying Safe Stopping Locations for Driverless Vehicles
Bachelor-Thesis Nr. 1399/22