A Taxonomy to Unify Fault Tolerance Regimes for Automotive Systems: Defining Fail-Operational, Fail-Degraded, and Fail-Safe

Torben Stolte, Stefan Ackermann, Robert Graubohm, Inga Jatzkowski, Björn Klamann, Hermann Winner, and Markus Maurer

***Abstract*—This paper presents a taxonomy that allows defining the fault tolerance regimes *fail-operational*, *fail-degraded*, and *fail-safe* in the context of automotive systems. Fault tolerance regimes such as these are widely used in recent publications related to automated driving, yet without definitions. This largely holds true for automotive safety standards, too. We show that fault tolerance regimes defined in scientific publications related to the automotive domain are partially ambiguous as well as taxonomically unrelated. The presented taxonomy is based on terminology stemming from ISO 26262 as well as from systems engineering. It uses four criteria to distinguish fault tolerance regimes. In addition to *fail-operational*, *fail-degraded*, and *fail-safe*, the core terminology consists of *operational* and *fail-unsafe*. These terms are supported by definitions of *available performance*, *nominal performance*, *functionality*, and a concise definition of the *safe state*. For verification, we show by means of two examples from the automotive domain that the taxonomy can be applied to hierarchical systems of different complexity.**

***Index Terms*—Safety, fault tolerance, fault tolerance regime, fail-operational, fail-safe, fail-degraded, safe state**

I. INTRODUCTION

In complex safety-critical systems, fault tolerance is a crucial property to ensure operation at an acceptable risk level. The complexity of mechatronic systems such as vehicle systems allows a distinction to be made between different forms of fault tolerance. These are often classified by developers and researchers in regimes such as *fail-operational*, *fail-degraded*, or *fail-safe*. However, technical literature and relevant standards show a lack of uniform definitions for these regimes. This leads to the frequent occurrence of different interpretations and designations of existing concepts; many publications in the context of automated vehicles even use the designations of fault tolerance regimes without stating or referring to any definition [2–23], as presented in Table I.

Yet, a uniform understanding of fault tolerance regimes is essential both for the scientific discussion of fault tolerance in vehicle systems and for the communication between interdisciplinary developers in distributed development projects and other important stakeholders. In particular, the concrete specification of fault tolerance, and therefore a consistent use of terminology becomes even more important when it comes to the development of and safety argumentation for SAE Level 4+ [24] automated vehicles. Vehicle automation impedes a safety argumentation because drivers cannot be included as a fallback layer in the safety concept of the automated driving functionality. Hence, the use of the addressed terms in the literature with a focus on the automotive domain is presented in Section II. In Section III, we propose a taxonomy for fault tolerance regimes that can be applied coherently to automotive systems. The applicability at different architectural levels is demonstrated by two automotive examples in Section IV. Section IV also contains a verification of a set of requirements.

This research is accomplished within the project "UNICAR*agil*" [1] (FKZ 16EMO0285, FKZ 16EMO0286). We acknowledge the financial support for the project by the German Federal Ministry of Education and Research (BMBF). *(Corresponding author: Torben Stolte)*

Torben Stolte, Robert Graubohm, Inga Jatzkowski, and Markus Maurer are with the Institute of Control Engineering at TU Braunschweig, 38106 Braunschweig, Germany {lastname}@ifr.ing.tu-bs.de

Stefan Ackermann, Björn Klamann, and Hermann Winner are with the Institute of Automotive Engineering at TU Darmstadt, 64287 Darmstadt, Germany {firstname(ö=oe).lastname}@tu-darmstadt.de

II. STATE OF THE ART

In order to illustrate the understanding of fault tolerance regimes in literature, we summarize publications that introduce definitions in an automotive context in Subsection II-A and additionally present selected publications from other domains in Subsection II-B. In the following subsections, terms in direct and indirect quotations are used according to the understanding of the cited authors. Terms in our own statements are in accordance with the definitions that we present in Section III.

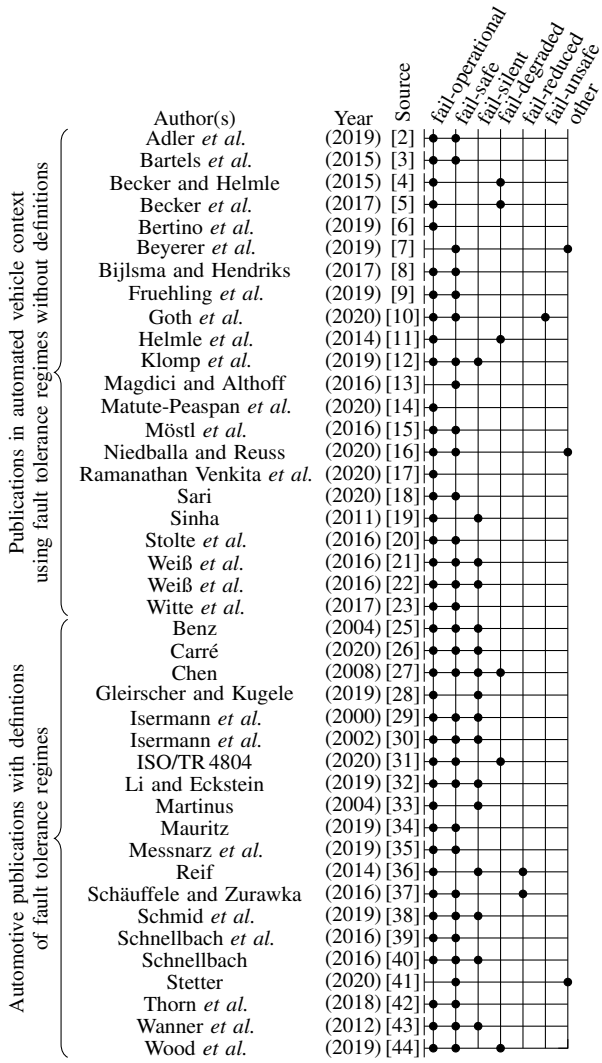
A. Fault Tolerance Regimes in the Automotive Domain

In the automotive domain, standards such as the recent versions of ISO 26262 [45] or ISO/DIS 21448 [46] introduce extensive safety-related terminology. However, the terminology in these two standards does not include definitions of fault tolerance regimes. The same applies to SAE J3016 [24]. An understanding of *fail-safe* at the vehicle level is described by the UNECE in [47] for automated driving applications, yet without distinguishing other fault tolerance regimes.

Aside from standards, several publications related to the automotive domain give definitions for different fault tolerance regimes [25–44]. An overview of the covered literature is presented in Table I. In general, the terms found in literature can be divided into three groups based on the desired system behavior in the presence of a fault. The first group consists of terms that indicate that a system can provide its specified functionality even in the presence of a fault. Terms that target systems that provide impaired functionality after a fault make up the second group. Terms in the third group require systems to fail into a defined state.

1) Upholding functionality: With one exception, every automotive publication with definitions of fault tolerance regimes introduced in Table I includes a regime describing the upholding of functionality [25–40, 42–44]. These publications define terms to address the continued provision of a system's

TABLE I
A NON-EXHAUSTIVE OVERVIEW OF PUBLICATIONS USING FAULT TOLERANCE REGIMES IN AN AUTOMOTIVE CONTEXT.



functionality in the presence of a fault without performance degradation and consistently use the term *fail-operational*. Still, when comparing the definitions, the understanding of the term varies slightly between the publications.

The publications defining *fail-operational* can be divided into three main categories. The first main category includes all publications that expect a *fail-operational* system to strive towards achieving a defined state in the event of a fault and will abort normal operation. The authors of the cited publications refer to the defined state almost exclusively as the safe state. This understanding of the safe state does not match the definition we present in this paper. This first main category can be divided into two subcategories. In the first subcategory [14, 25, 35, 42, 48, 49], a core functionality of the *fail-operational* system is maintained to achieve a defined state. The sole publication of the second subcategory [28] requires the system to maintain full functionality until the defined state is reached. It is not apparent to which subcategory each of the remaining publications [29–31, 37] can be assigned as the provided definitions are not specific enough to make this distinction.

The publications of the second main category [27, 33, 36, 43] describe a *fail-operational* system as a system that maintains its full functionality despite a fault. The authors of the aforementioned publications do not expect a *fail-operational* system to achieve a defined state in case of a fault, but expect it to continue its normal operation. The sole publication that is part of the third main category [44] deviates from the concept of maintaining functionality or achieving a defined state and rather describes a *fail-operational* system as a system that shall not lead to a safety-related situation in the event of a fault.

2) Upholding functionality with reduced performance: Researchers either use the term *fail-degraded* [27, 31, 44] or *fail-reduced* [36, 37] to describe a reduced system performance while maintaining the system's functionality in the presence of a fault. These terms are seldom defined in comparison to those describing either upholding a functionality or switching to a defined state.

According to Wood *et al.* [44] *fail-degraded* means "[...] that the system is still able to operate safely when degraded." In contrast, Chen [27] gives a more concise definition. He understands *fail-degraded* as the property of a system "which has the ability to continue with intended degraded operation at its output interfaces, despite the presence of hardware or software fault, [...]." Thus, Chen emphasizes two requirements for a system to be *fail-degraded*: (1) the system must continue its operation to be *fail-degraded* and (2) the continuation must follow a defined manner. Showing a similar understanding, ISO/TR 4804 defines *fail-degraded* as a property at the vehicle level when automated driving systems "operate with reduced functionality in the presence of a fault" [31].

For Reif [36], a *fail-reduced* system transitions into a state with a reduced functional capability in the presence of a fault. Similarly, Schäuffele and Zurawka [37] define *fail-reduced* as a "continued – albeit restricted – system serviceability" in case of a fault. Still, Reif [36] as well as Schäuffele and Zurawka [37] do not specify further what is meant by "reduced functional capability" or "restricted system serviceability," respectively.

A comparison of the definitions of *fail-degraded* and *fail-reduced* reveals a very similar understanding among the authors. Either definition represents an understanding that a system is able to continue its intended functionality, yet with degraded performance. Still, it becomes obvious that the understanding overlaps with the understanding that researchers have of *fail-operational*.

3) Switching to a defined state: Within the third group of terms, several authors use the terms *fail-safe* [25–27, 29–32, 34, 35, 37–44] and *fail-silent* [25–30, 32, 33, 36, 38, 40, 43], either exclusively or in combination.

The *fail-safe* property of a system is commonly described as the transition into a defined state (usually referred to as "safe state") in the event of failures [25, 27, 30, 35–37, 42–44, 49]. While most definitions describe the safe state as a specific condition of the analyzed (sub-)system [25, 27, 30, 35–37, 43, 49], Thorn *et al.* [42, p. 90] argue that the safe state is a "condition where the vehicle and occupants are safe." This corresponds to Wood *et al.* [44] who describe a fail-safe

system to continue operating "in a safe state in the event of a failure" [44, p. 135]. In [44], it is not completely clear how the authors distinguish between *fail-degraded* and *fail-safe* as the understanding of the term *system* is not further specified, e.g., whether system refers to the overall vehicle system or to different system levels within the vehicle. Luo *et al.* [49, p. 228] append that a system that reverts to a safe state generally no longer provides its required functionality. Isermann *et al.* [30, p. 69] and Wanner *et al.* [43, p. 599] include in their definition that a fail-safe system can also be brought to a safe state externally or passively in the event of failures. Finally, Schäuffele and Zurawka [37, p. 109] and Reif [36, p. 275] stress the fact that, after transitioning to a safe state, a fail-safe system also has to maintain this safe state and exit it only after additional measures were taken (e.g., external reset). The definition of the fail-safe property of an automated driving system in the technical report ISO/TR 4804 [31] specifies the need to achieve a *minimal risk condition* in addition to a safe state in the event of a failure. This extension is largely consistent with the description of a "Failsafe Response" by the UNECE [47].

The *fail-silent* property of a system is commonly described as the guarantee that no system output is provided in the event of failures [25, 27, 28, 30, 33, 36, 43, 48]. Therefore, the systems described as fail-silent usually represent (sub-)components of a larger complex system (e.g., a vehicle). Many authors name a complete shutdown or disabling of the communication of the system under consideration as a potential measure to achieve fail-silent behavior. Hence, the subsystems' functionality is no longer available in a larger system context, unless redundancies exist (cf. II-A1). Reif [36, p. 276] appends that fail-silent systems should not only stop providing output signals, but should also stop reacting on any input signals after a failure occurred. Chen [27, p. 9], Gleirscher and Kugele [28, pp. 5], and Martinus [33, p. 33] argue that fail-silent behavior is a possible manifestation of the fail-safe property of a system [33]. Carré [26] provides conflicting explanations of the fail-silent property of a system. On the one hand, he explains a fail-silent component to be designed to "continue operating properly in the event of the failure into a graceful degraded mode" [26, p. 56]. On the other hand, he characterizes fail-silent behavior of a system by "discontinued operation" [26, p. 66]. Deviating from other related work, Mauritz [34, p. 104] describes the fail-safe property of a component to be characterized by prevention "from further interaction with the remaining system," which corresponds to the common definition of fail-silent behavior. The same can be observed in the work of Stetter [41, p. 52], who bases his explanation of the fail-safe strategy in system design ("enabling a controlled shut-down" in the event of critical faults) on Blanke *et al.* [50].

B. Definitions from Non-Automotive Domains

Similar to the automotive-related publications discussed in Subsection II-A, an inconsistent understanding of fault tolerance regimes can also be observed in other domains.

Outlining concepts in the domain of dependable and secure computing, Avizienis *et al.* [51] use terms like *fail-silent* and *fail-safe* to describe a system's failure behavior. In the context of computing, the authors investigate *service failure modes* that are described as the manifestation of a deviation from correct service [51, p. 3.3.1]. If, by design, a system displays only a specific failure mode, Avizienis *et al.* [51] call it a *fail-controlled system*. The authors differentiate between various failure manifestations: They call a system with halted service failures a *fail-halt system*, a system with stuck service and silent failures a *fail-passive and fail-silent system*, and a system with minor failures (i.e., insignificant consequences) a *fail-safe system*.

Also with a background in dependable computing, Knight describes the *fail-safe* property in [52]. The author points out that even a *fail-safe* system that shuts down on error detection still exhibits a benign type of *continued service* by falling and remaining silent [52, p. 131].

Kopetz [53] describes the use of *fail-safe* and *fail-operational* in real-time systems design. The author classifies real-time systems as *fail-safe* if they are able to detect failures and subsequently identify and quickly reach a safe state [53, p. 15]. Consistent with some automotive publications discussed in the previous section, Kopetz points out that a safe state is a condition of a controlled object and not of the designed computer system itself. In contrast to some publications from the automotive domain, the author classifies applications as *fail-operational* if they "remain operational and provide a minimal level of service even in the case of a failure" [53, p. 15].

In the domain of nuclear safety, *fail-safe* is described by the IAEA [54, p. 11] as the behavior after a component or system failure, leading directly to a safe condition. It is further said that a component or system is only *fail-safe* for a stated kind of failure and situation. A contradicting understanding is used by Möller and Hansson [55] while still referring to [54]. They use the term *safe fail* instead of *fail-safe* to describe a system behaving safely after a component or system failure. The term *fail-safe* is instead used to describe a system that is "designed not to fail." Möller and Hansson [55] also introduce the terms *fail-silence* and *fail-operational*. They define *fail-silence* consistently to the automotive domain as a mechanism that shuts down the system in case of a component failure. They understand *fail-silence* as a sub-category of a *safe fail* mechanism. In contrast, *fail-operational* is understood as a system that continues to work despite a fault occurrence by Möller and Hansson. They also state that a distinction is sometimes made in related literature between the system remaining partially operational, which is then called *fail-active*, and the system remaining fully operational.

NASA [56] uses the term *fail-safe* in a slightly different way by defining it not only as an ability that safely terminates an operation, but potentially control the operation further after failure occurrence.

Finally, in the domain of fault-tolerant control systems, Blanke *et al.* [50, 57] present two definitions each for *fail-operational* and *fail-safe* as system-wide properties. In [50], a *fail-operational* system "is able to operate with no change in objectives or performance despite of any single failure," while a *fail-safe* system is understood as a system that "fails to a state that is considered safe in the particular context." The

definition of *fail-operational* in [57, p. 663] ("The ability to sustain any single failure.") is similar to [50]. In contrast, the definition of *fail-safe* in [57, p. 662] appears inconclusive.

C. Summary

Overall, the literature that uses and defines fault tolerance regimes shows a similar understanding with respect to the different regimes. Still, the understanding is not free of contradictions since slightly varying terms as well as a semantic overlap between the understanding of the terms can be observed in the different publications. Moreover, the literature known to us does not provide taxonomic support for a clear distinction between fault tolerance regimes. Another aspect that is only partially addressed in the available literature is consistency in the use of related terms. An exception is the technical report ISO/TR 4804 [31], which embeds its definitions into the terminology defined in ISO 26262 [45], ISO/DIS 21448 [46], and SAE J3016 [24]. However, ISO/TR 4804 [31] does not provide a taxonomic demarcation within the terminology and limits its definitions to the case of an automated driving system as the design item. In contrast, fault tolerance regimes defined in other publications are applicable to arbitrary systems and system levels.

III. TAXONOMY

The review of publications in Section II reveals a divergent understanding of fault tolerance regimes. Moreover, there are no established normative definitions for fault tolerance regimes available. However, for an interdisciplinary safety argumentation, a harmonized understanding of fault tolerances of complex systems is essential. We therefore present in this section a taxonomy in order to support a unified understanding of fault tolerance regime. Subsection III-A contains the requirements we used as basis for deriving the taxonomy. In Subsection III-B, we describe the related terms on which the definitions of fault tolerance regimes rely before outlining the taxonomy in Subsection III-C. Fig. 1 illustrates the newly defined terms, the related terms, as well as their interconnection.

A. Requirements

The new definitions shall support researchers and developers, who decide which safety functions are needed for specific systems or components. This goal leads to a set of requirements to reach a common understanding of the different fault tolerance regimes.

To reach this common understanding, the new definitions need to be unambiguously understandable for all stakeholders of a domain. Therefore, the fault tolerance regimes have to be clearly separable from each other and have to resolve existing contradictions in current definitions as shown in Section II:

RQ 1: *The fault tolerance regimes must be clearly distinguishable from each other by means of the taxonomy.*

Furthermore, the new definitions need to be widely accepted in the community so that a majority of the stakeholders is aware of the definitions. Thus, the definitions need to be compatible to generally accepted definitions in existing literature as presented in Section II, which leads to the second requirement for our definitions:

RQ 2: *The terms used for the fault tolerance regimes as well as the corresponding definitions should reflect the most common usage in the literature.*

Deviations from this may occur for the case that these definitions contradict a new logical argumentation of a novel definition.

Our focus in this paper is the application of the fault tolerance regime taxonomy to the automotive domain (in particular in the context of the automotive safety standards for electrical and electronic (E/E) systems ISO 26262 [45] and ISO/DIS 21448 [46]), bringing forth requirement RQ 3:

RQ 3: *The fault tolerance regime definitions must be applicable to systems of the automotive domain.*

In Section II, we point out that fault tolerance regimes are used at different system levels. Therefore, the definitions should not only be applicable to the vehicle level, but should also cover the fault tolerance behavior of, e.g., subsystems. This means that different behavior at different levels needs to be considered and eventually transformed to the relevant level, e.g., this can also involve the user of a system. This may be necessary because of a reaction of superimposed systems or the environment due to the behavior of the considered system:

RQ 4: *The fault tolerance regime definitions should be applicable at different system levels.*

For a complex system like a vehicle, a high number of possible faults can be expected. Faults may also occur simultaneously and may lead to different behavior of the system depending on its fault tolerance characteristics as stated, e.g., by Isermann *et al.* [29, 30]. This is addressed by requirement RQ 5:

RQ 5: *The fault tolerance regime definitions must work for an arbitrary number of concurrent faults.*

We use these requirements as a basis for the development of our definitions and will verify the definitions by these requirements in Subsection IV-C.

B. Related Terms

In order to define the terms for the fault tolerance regimes in Subsection III-C, it is necessary to provide definitions for related terms. The related terms are either used directly for the definition of the different fault tolerance regimes or in the explanatory text. Most provided related terms are based on ISO 26262 [45]. When we deviate from the definitions provided by ISO 26262, we argue our reasons for this deviation.

Safety: *Absence of unreasonable risk.* [45, Part 1, 3.132]

As a consequence, we assume that there is a risk threshold. Below this threshold, the risk is accepted, while above the threshold, the system is considered unsafe. It is worth noting that we take solely an engineering perspective. A legal perspective would potentially demand zero risk [59].

Risk: *Combination of the probability of the occurrence of harm and the severity of that harm.* [45, Part 1, 3.128]

Harm in the context of ISO 26262 always refers to "physical injury or damage to the health of persons" [45, Part 1, 3.74] and

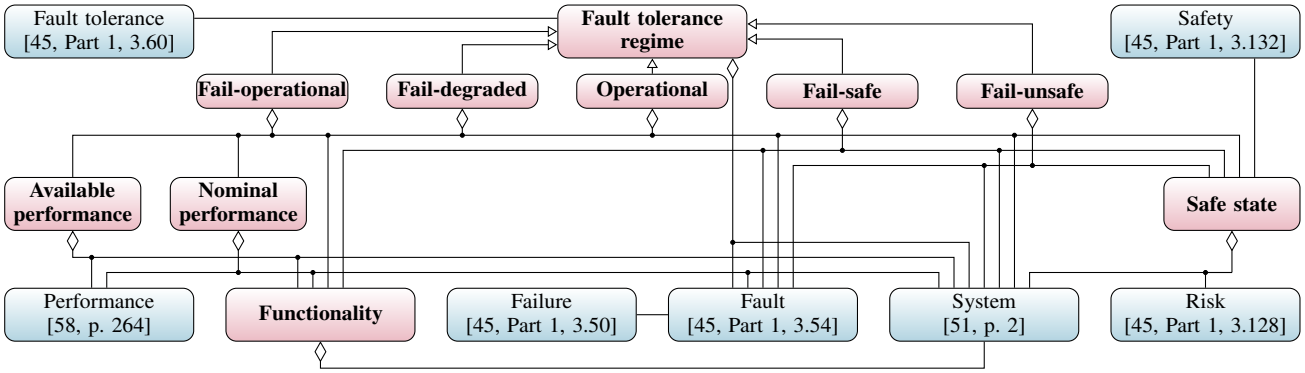


Fig. 1. Terms defined in this paper (in bold) and related terms, where ⊸◇ indicates required terms, ⟶▷ instances, and ── semantically related terms.

excludes damage to, e.g., property or reputation. Furthermore, it is our understanding that, when establishing the combination introduced by the definition of *risk*, an increase in either factor, i.e. the probability of occurrence or the severity, must result in a monotonically increasing risk quantification.

> **System:** *An entity that interacts with other entities, i.e., other systems, including hardware, software, humans, and the physical world with its natural phenomena.* [51, p. 2]

According to Avizienis *et al.* [51, p. 2], the "other systems are the environment of the given system. The system boundary is the common frontier between the system and its environment." ISO 26262 [45, Part 1, 3.163] defines a system as a "set of components or subsystems that relates at least a sensor, a controller and an actuator with one another." We prefer the definition by Avizienis *et al.* over the definition provided by ISO 26262 as it provides a broader perspective on the system concept. By applying the definition of Avizienis *et al.* to the definition of ISO 26262, the controller, the sensor, and the actuator as entities can each be considered as a system by themselves with the remaining components belonging to the environment the entity interacts with. Thus, a narrower system boundary can be drawn compared to ISO 26262 without excluding the specification of the system boundary of ISO 26262.

> **Fault:** *Abnormal condition that can cause an element or an item to fail.* [45, Part 1, 3.54]

This definition is in accordance with Avizienis *et al.* [51], who define a fault as the possible cause of an error, where an error is the system's deviation from its correct external state, i.e., what is perceivable at the system's interfaces to the environment. All external states together make up the system's behavior. While not explicitly stated by ISO 26262 or Avizienis *et al.*, we assume faults to be discrete and distinguishable, as suggested in Part 5 of ISO 26262.

> **Failure:** *Termination of an intended behavior of an element or an item due to a fault manifestation.* [45, Part 1, 3.50]

This definition follows the definition by Avizienis *et al.* [51, p. 3] of a failure as a deviation between the behavior provided by the system and its correct behavior to the point that the system is unable to provide its intended function.

> **Fault tolerance:** *Ability to deliver a specified functionality in the presence of one or more specified faults.* [45, Part 1, 3.60]

This definition is almost identical to the definition given by Avizienis *et al.* for whom fault tolerance "means to avoid [...] failures in the presence of faults" [51, p. 4].

> **Performance:** *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).* [58, p. 264]

Performance has not been defined so far in automotive safety standards. The use of the concept of *performance*, such as in the description of *performance limitations* in the ISO/DIS 21448 standard, is not accompanied by a clear definition of the term *performance*. Consequently, we refer to the definition of *performance* from systems engineering.

C. Definitions

Based on the requirements presented in Subsection III-A together with the related terms in Subsection III-B, we propose the following taxonomy for fault tolerance regimes. The taxonomy consists of five terms: *operational*, *fail-operational*, *fail-degraded*, *fail-safe*, and *fail-unsafe*. They are supported by a concise definition of *safe state* as well as by definitions of *functionality*, *available performance*, and *nominal performance*. However, the term *fault tolerance regime* must first be defined:

> **Fault tolerance regime** (Definition 1)**:** *System property that classifies the system's behavior in the presence of a specific fault combination.*

The term *fault combination* is used as distinct faults can occur at the same time. So, assuming that $\mathcal{F}$ denotes the set of all distinct faults that can occur in a system, $\mathcal{P}(\mathcal{F}) = \{f \mid f \subseteq \mathcal{F}\}$ is the set of possible fault combinations f. The more general assumption of fault combinations used here also covers the "fail behavior" in response to an increasing number of faults, which is used by Isermann *et al.* [29, 30] and Jacobson *et al.* [60].

Unlike other publications, we argue that fault tolerance regimes are not necessarily a system-wide property. Rather,

fault tolerance regimes are a property that must be seen with respect to a set of covered fault combinations. On the one hand, fault tolerance regimes are often connected to an assumption about how many faults can occur at the same time. For instance, a single fault assumption $|f| = 1$ is regularly used in the automotive domain [29]. Thus, all fault combinations $\{f \mid |f| > 1\}$ are not covered by the specific fault tolerance regime. On the other hand, even within the assumed range of $|f|$, a single fault combination that is not covered is enough to invalidate a fault tolerance regime assigned to an entire system. As a consequence, all fault combinations must be known to be able to assign a specific system-wide fault tolerance regime, which is challenging to say the least.

For the following definitions, we use four criteria to distinguish fault tolerance regimes from each other. The *first criterion* is whether a fault is present in the system. Furthermore, fault tolerance regimes are fundamentally associated with the question of what is safe in the context of a system. Thus, the *second criterion* is whether a system allows for a *safe state* in the presence of a fault combination f. We define a *safe state* as follows:

Safe state (Definition 2)**:** *State in which a system does not pose an unreasonable risk.*

With this definition, we generalize inconsistent definitions by ISO 26262, ISO/DIS 21448, and ISO/TR 4804. A *safe state* according to ISO 26262 is an "operating mode, in case of a failure, of an item without an unreasonable level of risk" [45, Part 1, 3.131]. An *operating mode* refers to the "conditions of functional state that arise from the use and application of an item or element" [45, Part 1, 3.102], e.g., "system off", "system active", "degraded operation".

When defining the term *safe state*, both ISO/DIS 21448 [46] and ISO/TR 4804 [31] refer to the *minimal risk condition* introduced by SAE J3016 [24] but come to different conclusions. ISO/DIS 21448 equates the *safe state* defined by ISO 26262 with a *minimal risk condition*, which is a vehicle state that is supposed "to reduce the risk of harm, when a given trip cannot be completed" [46, 3.16]. In contrast, ISO/TR 4804 distinguishes between *safe state* and *minimal risk condition* and defines a *safe state* as an "operating mode that is reasonably safe" [31, 3.50], while a *minimal risk condition* is a "condition to which a user or an automated driving system may bring a vehicle after performing the *minimal risk manoeuvre* in order to reduce the risk of a crash when a given trip cannot be completed" [31, 3.29]. Consequently, a *minimal risk condition* is not necessarily safe. Instead, it is the condition a vehicle will reach in response to specific events due to the implemented safety mechanisms. This condition may exceed the accepted risk threshold and would therefore not qualify as a *safe state*. Furthermore, the understanding of *safe state* in ISO/TR 4804 [31, 3.50] reflects the understanding of *safe state* outlined by Reschka and Maurer [61]. Reschka and Maurer argue at the vehicle level that an (automated) vehicle must maintain a *safe state* even without the occurrence of a fault.

Although the definition of the *safe state* provided by ISO/TR 4804 appears reasonably generic, we provide a more general definition for the term *safe state* in Definition 2 for two reasons. Firstly, the definition by ISO/TR 4804 is only intended to be applied at the vehicle level and not at a subsystem level. Secondly, it relies on the term *operating mode*, which however is not further explained in ISO/TR 4804. It presumes knowledge of the corresponding definition in ISO 26262 [45, Part 1, 3.131], though this definition and the associated examples are inconclusive.

Our understanding of the term *safe state* according to Definition 2 is in principle applicable to all kinds of safety considerations: those targeting internal faults as well as those targeting insufficient specification or unconsidered technological limitations. However, the following definitions of fault tolerance regimes presume a sufficient system specification as well as a complete consideration of technological limitations. In the context of E/E automotive systems, our focus is on *functional safety* according to ISO 26262 [45] rather than on the *safety of the intended functionality* according to ISO/DIS 21448 [46]. Still, the definition of *safe state* works for both. Moreover, arguing that a system does not pose an unreasonable risk often requires consideration of neighboring or superimposed systems because a *safe state* preservation could require an adequate reaction of these.

For describing a system, we distinguish between the system's *functionality* and its *performance* while providing the *functionality*. This distinction follows Avizienis *et al.* [51] who propose that a system is specified by a dualism of *functionality* and *performance*. Still, neither term is defined in [51] or in automotive E/E safety standards. Yet, a definition of *performance* is given in systems engineering context [58], cf. Subsection III-B. For *functionality*, we propose the following definition:

Functionality (Definition 3)**:** *Behavior of a system expressed in its interaction with its operating environment.*

This definition integrates the term *behavior*, which is frequently encountered in ISO 26262 and ISO/DIS 21448 to describe what a system does, with a description of functionality by Walden *et al.* [58]. Walden *et al.* state that "the functionality of a system is typically expressed in terms of the interaction of the system with its operating environment [...]" [58, p. 6].

Describing a system by means of *functionality* and *performance*, allows establishing the third and fourth criterion for the distinction of fault tolerance regimes. The *third criterion* is the system's ability to provide its functionality. In general, a *safe state* can be maintained both when a system provides its functionality and when it does not. Therefore, we integrate the availability of a *safe state* with the system's ability to provide its specified functionality into the system's operability $o(f)$ as

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

The *fourth* and *last criterion* used for distinguishing *fault tolerance regimes* is the *available performance* $p_{\mathrm{a}}(f)$ of the system while providing its functionality:

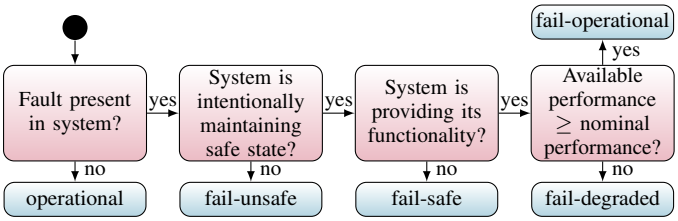


Fig. 2. Scheme to distinguish fault tolerance regimes according to the presented taxonomy.

Available performance (Definition 4)**:** *Performance that is available for a system to provide its specified functionality.*

In order to allow for a distinction of fault tolerance regimes, the *available performance* can be related to the *nominal performance* p_{nom}, which we define as follows:

Nominal performance (Definition 5)**:** *Performance with which a fault-free system is expected to be able to provide its specified functionality.*

As illustrated by the scheme presented in Fig. 2, the evaluation of the four criteria in the order of their appearance in this section allows for a clear distinction of fault tolerance regimes. Thus, it facilitates their definitions, which are outlined in the following paragraphs. As our focus is on fault tolerance, we presume as a starting point that a fault-free system is always able to maintain a *safe state*.

Operational (Definition 6)**:** *An operational system has no fault and, thus, can provide its specified functionality with at least nominal performance while maintaining a safe state.*

Therefore, a system is *operational* for $f = \emptyset \subseteq \mathcal{P}(\mathcal{F})$, from which follows $o(\emptyset) = 1$ and $p_{\mathrm{a}}(\emptyset) \geq p_{\mathrm{nom}}$.

Fail-unsafe (Definition 7)**:** *A system is fail-unsafe in the presence of a fault combination if it is not able to maintain a safe state.*

Consequently, $\{f \mid o(f) = -1\} \subseteq \mathcal{P}(\mathcal{F})$ defines the set of fault combinations that a system cannot handle safely.

Fail-safe (Definition 8)**:** *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.*

Thus, $\{f \mid o(f) = 0\} \subseteq \mathcal{P}(\mathcal{F})$ defines the set of fault combinations that a system can handle in a *fail-safe* manner. It is important to note again that whether a *safe state* is suitable can only be assessed by considering neighboring or superimposed systems. For instance, the *minimal risk condition* described for automated vehicles in ISO/TR 4804 and ISO/DIS 21448 is a potential *safe state*. Other road users, who may be considered neighboring systems, may have to adapt to an automated vehicle that, e.g., pulls over to the side of the road after a fault.

We do not use the term *fail-silent* in this taxonomy for two reasons. Firstly, *fail-safe* and *fail-silent* are not sharply differentiated in literature as shown in Section II. Similar to the *safe state*, we understand a "silent" system output as a specific defined state. Secondly, *fail-silent* as a property that requires systems to have no output at all is hard to imagine. For systems that are required to be continuously active, a suddenly inactive output carries information as well. In contrast, for systems that are only rarely used, a system shutdown indication is common. For example, Electronic Stability Control systems, cf. [29, 30], usually feature an indication to the driver if the systems have encountered a shutdown.

Fail-degraded (Definition 9)**:** *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.*

Thus, $\{f \mid (o(f) = 1) \wedge (p_{\mathrm{a}}(f) < p_{\mathrm{nom}})\} \subseteq \mathcal{P}(\mathcal{F})$ defines the set of fault combinations that a system can handle in a *fail-degraded* manner. In contrast to Definition 6, the available performance $p_{\mathrm{a}}(f)$ is lower than the nominal performance p_{nom} while the overall *safe state* is maintained. *Fail-degraded* is chosen rather than *fail-reduced* because it is more common in the body of literature we have reviewed. An operation with an available performance p_{a} below the nominal performance p_{nom} may require an adaptation to this degradation on superimposed system layers. An example for a *fail-degraded* behavior is the limp home mode of combustion engines, which allows for a continuation of a trip with reduced speed in the presence of a fault [27, 30, 37, 40].

Fail-operational (Definition 10)**:** *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.*

Consequently, $\{f \mid (o(f) = 1) \wedge (p_{\mathrm{a}}(f) \geq p_{\mathrm{nom}})\} \subseteq \mathcal{P}(\mathcal{F})$ defines the set of fault combinations that a system can handle in *fail-operational* manner. Contrary to some researchers' understanding of *fail-operational*, we do not subsume a degraded operation in *fail-operational*. As for Definition 6, the available performance $p_{\mathrm{a}}(f)$ equals at least nominal performance p_{nom} while the overall *safe state* is maintained. Thus, neighboring or superimposed systems can continue their operation as with an *operational* system.

IV. APPLICATION AND VERIFICATION OF THE PROPOSED TAXONOMY

To demonstrate that this taxonomy can be applied to different systems as well as at different system levels, we introduce two examples from the automotive domain. Subsection IV-A illustrates the taxonomy at the example of a steer-by-wire system. As a more complex example, we show in Subsection IV-B that the taxonomy can be applied to an automated driving application as well. Both examples contribute to the verification of the taxonomy against the requirements, which is presented in Subsection IV-C.

A. Steer-by-Wire System

A steer-by-wire system is the first example for evaluating the taxonomy. Steer-by-wire is considered as highly safety-critical in general and is a necessary feature of future automated vehicles. In the steer-by-wire system, the desired steering angle is calculated and controlled purely by an electronic system.

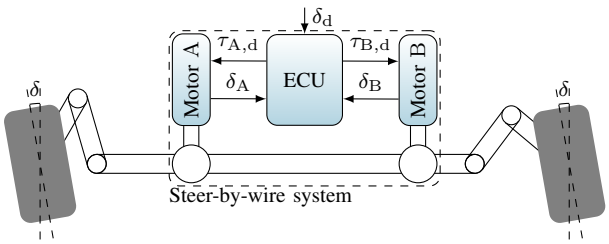


Fig. 3. Sketch of steer-by-wire system. ECU: electronic control unit; δ: steering angle; δ_d: desired steering angle; δ_A: motor angle of motor A; δ_B: motor angle of motor B; $\tau_{\mathrm{A,d}}$: desired torque of motor A; $\tau_{\mathrm{B,d}}$: desired torque of motor B.

Fig. 3 illustrates the example steer-by-wire system used in this subsection, which consists of three subsystems. There are the two steering motors A and B. The two motors apply torques τ_A and τ_B to the steering rack in order to control the steering angle δ by shifting the steering rack to the left or right. They each can provide motor angles δ_A and δ_B, which directly relate to the actual steering angle δ. The third subsystem is the electronic control unit (ECU), which closes the steering angle control feedback loop.

From a functional perspective, the system is required to steer in terms of *setting the steering angle*. Thus, the actual steering angle δ is the system's output while the desired steering angle δ_d is the input. Internally, the ECU calculates a required steering torque τ_d and allocates it to motors A and B. They are subject to $\tau_\mathrm{d} = \tau_{\mathrm{A,d}} + \tau_{\mathrm{B,d}}$, where $\tau_{\mathrm{A,d}}$ and $\tau_{\mathrm{B,d}}$ are the desired torques for motor A and B. Finally, the resulting steering torque $\tau = \tau_\mathrm{A} + \tau_\mathrm{B}$ alters the actual steering angle δ.

Let's assume that the target application requires a minimum steering angle range $\delta_\mathrm{nom} \in [\underline{\delta}\ \bar{\delta}]$ as well as minimal available steering torque range $\tau_\mathrm{nom} \in [\underline{\tau}\ \bar{\tau}]$ where $\bar{\cdot}$ and $\underline{\cdot}$ indicate the upper and lower bounds of the ranges. Summarized, this yields the steer-by-wire system's nominal performance $p_{\mathrm{SbW,nom}} = p_\mathrm{SbW}(\delta_\mathrm{nom}, \tau_\mathrm{nom})$.

Similarly, the nominal performances of the subsystems can be defined. For motors A and B, these are $p_{\mathrm{A,nom}} = p_\mathrm{A}(\tau_{\mathrm{A,nom}})$ with $\tau_{\mathrm{A,nom}} \in [\underline{\tau}_\mathrm{A}\ \bar{\tau}_\mathrm{A}]$ and $p_{\mathrm{B,nom}} = p_\mathrm{B}(\tau_{\mathrm{B,nom}})$ with $\tau_{\mathrm{B,nom}} \in [\underline{\tau}_\mathrm{B}\ \bar{\tau}_\mathrm{B}]$. For the ECU, a performance measure is, e.g., the number of operations per second. Further explanations regarding the ECU subsystem are omitted as we consider it as *fail-operational* for this example.

Let's assume that motor A is subject to a fault combination f_A and that the motor comes with a mechanism targeting *fail-safe* behavior for f_A. The mechanism forces motor A into a torque-free state ($\tau_{\mathrm{A,fs}} = 0$) and, thus, inhibits its ability to steer while the motor maintains a defined state. Assessing whether motor A with zero steering torque still results in a safe steer-by-wire system requires taking the remaining system components into account, in particular motor B. If $\tau_{\mathrm{B,nom}} \supseteq \tau_\mathrm{nom}$, the steer-by-wire system is *fail-operational* in case of the fault combination f_A. $\tau_{\mathrm{B,nom}} \subset \tau_\mathrm{nom}$ yields a *fail-degraded* system as the steering is still functional, $o_\mathrm{SbW}(f_\mathrm{A}) = 1$, yet with decreased available performance. For the latter, $o_\mathrm{SbW}(f_\mathrm{A}) = 1$ presumes that the resulting available performance $p_{\mathrm{SbW,a}}(f_\mathrm{A}) = p_\mathrm{SbW}(\delta_\mathrm{a}(f_\mathrm{A}), \tau_\mathrm{a}(f_\mathrm{A}))$ suffices to operate the vehicle safely although it is below its nominal performance, $p_{\mathrm{SbW,a}}(f_\mathrm{A}) < p_{\mathrm{SbW,nom}}$. It follows $o_\mathrm{A}(f_\mathrm{A}) = 0$ such that motor A is *fail-safe* for f_A.

Altogether, this example illustrates two different aspects of the taxonomy. Firstly, it demonstrates that the taxonomy can be applied to hierarchical system designs. Secondly, it also shows that the nominal performance is not necessarily congruent to the maximum available performance. Again, it is important to note that the fault tolerance regimes are always related to a specific fault combination.

B. SAE Level 4 Automated Driving Application

As a second example, we apply the taxonomy to an automated vehicle equipped with an SAE level 4 automated driving system (ADS) according to SAE J3016 [24]. This ADS determines the vehicle behavior and is therefore highly safety-critical at vehicle level. Fig. 4 illustrates the example ADS used here, which consists of four subsystems: the normal operation automated driving functionality, the emergency stop system *Safe Halt*, the trajectory selection, and the vehicle motion control and actuation subsystem. From a functional perspective, the ADS is required to realize a desired mission ①, which is the system's input. As output ⑦, the ADS generates the vehicle behavior necessary for mission accomplishment. Internally, the normal operation automated driving functionality, which includes the environment perception and interpretation as well as the behavior planning and trajectory generation, outputs a reference trajectory ②, which is used for normal vehicle operation.

As second output, the subsystem generates an emergency path with associated maximum speed profile ③. This emergency path and speed profile are the inputs of the *Safe Halt* functionality [64], which is developed in the German research project UNICAR*agil* [1]. The emergency path and maximum speed profile are supposed to transition the vehicle from the current vehicle state to a defined state that qualifies as the minimal risk condition for the vehicle according to ISO/DIS 21448 [46]. Thus, the normal operation automated

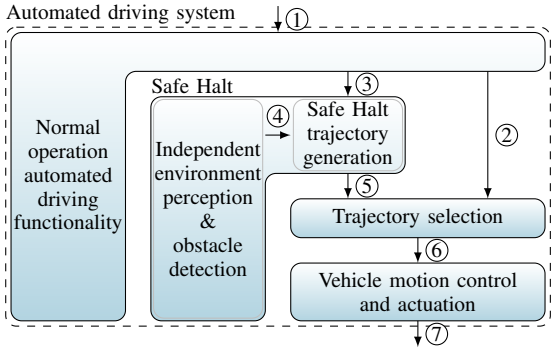


Fig. 4. Functional sketch of an automated driving system with *Safe Halt* emergency stop application. The illustration is based on the generic functional system architecture presented by Matthaei and Maurer [62] and Ulbrich *et al.* [63]. ① Desired mission; ② Normal reference trajectory; ③ Emergency path with maximum speed profile along the path; ④ Obstacle list; ⑤ Emergency reference trajectory; ⑥ Selected reference trajectory; ⑦ Vehicle behavior.

driving functionality determines the minimal risk condition and plans the emergency path in combination with a path-related maximum speed profile to implement a minimal risk maneuver [46] that leads to the minimal risk condition.

The emergency stop system *Safe Halt* determines the emergency reference trajectory ⑤ based on the reference path, the maximum speed profile along the path ③, and an obstacle list ④. The obstacle list is generated by an independent environment perception and obstacle detection system, which monitors the driving corridor along the emergency path to avoid collisions with obstacles. Consequently, the provided obstacle list enables planning a collision-free velocity profile for the minimal risk maneuver. Based on the emergency path and the collision-free velocity profile, the *Safe Halt* trajectory generation provides the emergency reference trajectories ⑤.

The third subsystem is a trajectory selection that selects the reference trajectory ⑥ to be forwarded to the vehicle motion control and actuation. Based on the health status of the ADS subsystems, either the normal reference trajectory or the emergency reference trajectory is selected. Finally, the vehicle motion control and actuation subsystem realizes the vehicle behavior based on the selected reference trajectory.

For describing the performance of the ADS, we assume that the ADS is designed to offer a set of selectable missions $\mathcal{M}_{\mathrm{nom}}$ within its operational design domain (ODD, cf. [24, 65, 66] for explanations). Furthermore, for each mission m, a mission quality q_{nom} shall be required so that $\forall\, m \in \mathcal{M}_{\mathrm{nom}}\ \exists\, q_{\mathrm{nom}}(m)$, which could contain, i.a., the mission execution time as well as measures for driving comfort.

With $\mathbf{q}_{\mathrm{m,nom}}$ containing all $q_{\mathrm{nom}}(m)$, $p_{\mathrm{ADS,nom}} = p_{\mathrm{ADS}}(n_{\mathrm{nom}}, \mathbf{q}_{\mathrm{m,nom}})$ is the nominal performance of the ADS, where $n_{\mathrm{nom}} = |\mathcal{M}_{\mathrm{nom}}|$ 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 $p_{\mathrm{ADS,a}}(f) = p_{\mathrm{ADS}}(n_{\mathrm{a}}(f), \mathbf{q}_{\mathrm{m,a}}(f))$, where $\mathbf{q}_{\mathrm{m,a}}(f)$ contains the achievable mission quality in the presence of the fault combination $f\ \forall m \in \mathcal{M}_{\mathrm{nom}}$.

Let's assume that the normal operation automated driving functionality (NADF) is subject to a fault combination f_{NADF}. This fault combination f_{NADF} can affect both performance measures, the number of available missions as well as the achievable mission quality. If $p_{\mathrm{ADS,a}} = p_{\mathrm{NADF,a}} = p_{\mathrm{ADS}}(f_{\mathrm{NADF}}) \geq p_{\mathrm{NADF,nom}}$, the normal operation automated driving functionality and, thus, the automated driving system is *fail-operational* for the fault combination f_{NADF}. This means that the ADS can perform all specified missions in its ODD with the nominal mission quality, yielding $n_{\mathrm{a}}(f_{\mathrm{NADF}}) = n_{\mathrm{nom}}$ and $\mathbf{q}_{\mathrm{m,a}}(f_{\mathrm{NADF}}) \geq \mathbf{q}_{\mathrm{m,nom}}$.

The ADS is *fail-degraded* when the fault combination f_{NADF} leads to a certain set of missions being infeasible, yielding $\mathcal{M}_{\mathrm{a}} \subsetneq \mathcal{M}_{\mathrm{nom}}$ with $\mathcal{M}_{\mathrm{a}} \neq \emptyset$, or reduces the achievable quality of the available missions $\mathbf{q}_{\mathrm{m,a}}(f_{\mathrm{NADF}}) < \mathbf{q}_{\mathrm{m,nom}}$. Then, the normal operation automated driving functionality is still functional, $o_{\mathrm{ADS}}(f_{\mathrm{NADF}}) = o_{\mathrm{NADF}}(f_{\mathrm{NADF}}) = 1$, yet with decreased available performance, $p_{\mathrm{ADS,a}}(f_{\mathrm{NADF}}) = p_{\mathrm{NADF,a}}(f_{\mathrm{NADF}}) < p_{\mathrm{ADS,nom}}$. For example, faults in a redundantly designed environment perception system can lead to this system property. Since $o_{\mathrm{NADF}}(f_{\mathrm{NADF}}) = 1$, it is expected that the automated driving system and, thus, the automated vehicle maintain a *safe state*[1].

While *fail-operational* and *fail-degraded* behavior can be achieved for the example ADS within the normal operation automated driving functionality, *fail-safe* behavior of the ADS can arise in two ways. For both ways, the fault combination f_{NADF} leads to all missions being infeasible, yielding $\mathcal{M}_{\mathrm{a}} = \emptyset$. For the first option, the *safe state* is maintained by the normal operation automated driving functionality through executing a minimal risk maneuver to transition the vehicle to a minimal risk condition. This results in $o_{\mathrm{ADS}}(f_{\mathrm{NADF}}) = o_{\mathrm{NADF}}(f_{\mathrm{NADF}}) = 0$.

The second way to implement *fail-safe* behavior is the Safe Halt functionality. The Safe Halt functionality can engage if the normal operation automated driving functionality is not able to maintain a *safe state* in the presence of the fault combination f_{NADF} resulting in $o_{\mathrm{NADF}}(f_{\mathrm{NADF}}) = -1$. By providing a minimal risk maneuver that terminates in the preselected minimal risk condition, a *safe state* can be maintained via the Safe Halt functionality. Therefore, $o_{\mathrm{ADS}}(f_{\mathrm{NADF}}) = 0$ results even for $o_{\mathrm{NADF}}(f_{\mathrm{NADF}}) = -1$. The *fail-operational* trajectory selection, which is required for both options, selects the reference trajectory based on the health status of the ADS subsystems.

Overall, this example demonstrates that the taxonomy can also be applied at higher system levels and in more complex contexts. It supports system designers by providing the minimal required performance of a fallback system to cope with faults in the primary system to maintain a *safe state* of the superimposed system.

C. Requirements Verification

We developed our definitions for fault tolerance regimes based on the requirements given in Subsection III-A. These requirements are verified in this subsection.

Requirement RQ 1 demands clear distinguishability of the fault tolerance regimes. To address this requirement, we introduce definitions for functionality as well as available and nominal performance. We include these terms in the definition of fault tolerance regimes together with the scheme provided in Fig. 2. Furthermore, we argue that a system can only be classified with regard to a fault tolerance regime for a specific fault combination. For the classification, it is necessary to assess whether a system is able to provide its specified functionality while also considering whether the system remains in a *safe state*. The nominal performance as well as the available performance due to faults under specified environmental conditions need to be identified during design time to achieve the desired fault tolerance behavior. This is also shown in the two examples in Subsections IV-A and IV-B. Within these examples, the nominal and available performance are described so that the implemented behavior of the system after the occurrence of example faults can be assigned to a fault tolerance regime.

[1]For this example, we presume a proof that the minimal risk condition is a *safe state*. Still, providing such a proof for a given operational design domain is beyond the focus of this papers and requires further research.

In Section II, we show that the state-of-the-art definitions differ or are even inconsistent from each other. Nevertheless, as required by requirement RQ 2, the proposed definitions follow the basic understanding of fault tolerance regimes in the literature. The proposed definitions pick up the three most commonly used terms denoting fault tolerance regimes. Moreover, our definitions differ from, yet are still compatible to the recently published technical report ISO/TR 4804 [31].

Using automotive standards as basis for the development of our definitions paves the way to applicability in the automotive domain as required by requirement RQ 3. As outlined in Subsection III-B, we employ existing definitions from the automotive domain to guide the novel definitions, but also refine the *safe state* to resolve identified contradictions. The examples from the automotive domain, presented in Subsections IV-A and IV-B, provide further evidence for the verification of requirement RQ 3.

Requirement RQ 4 demands applicability at different system levels. For the verification of this requirement, the two presented examples show how lower level entities can be rated by considering the implications of their behavior at higher levels. Additionally, the examples demonstrate that the taxonomy can be applied at system levels other than the vehicle level, which is commonly used in the automotive domain. Thus, the taxonomy fulfills requirement RQ 4.

Finally, to fulfill requirement RQ 5 to address multiple concurrent faults, we provide definitions for fault tolerance regimes covering systems' reactions to *fault combinations*. Fault combinations can consist of a single or multiple faults.

In conclusion, we are able to verify the requirements given in Subsection III-A by complying with the constraints to commonly accepted definitions, adding the consideration of functionality and performance to our definition of fault tolerance regimes, and consequently applying our taxonomy to two different examples.

V. CONCLUSION

In this paper, we present a taxonomy to clearly distinguish fault tolerance regimes in order to overcome a diverging use found in the literature related to the automotive domain. The application of the taxonomy to different automotive systems – two examples are outlined in this paper – indicates its general applicability. To this end, the taxonomy presumes a clear definition of the system's functionality and according performance metrics. The taxonomy and the derived definitions are compatible to the recently published technical report ISO/TR 4804 [31], which defines *fail-operational*, *fail-degraded*, and *fail-safe*. Still, our taxonomy allows for an application at arbitrary system levels while ISO/TR 4804 is limited to automated driving systems at the vehicle level. Moreover, the taxonomy integrates into recent automotive E/E safety standards, where the focus of this paper is on *functional safety* according to ISO 26262 [45]. The defined terms mostly refer to definitions stemming from ISO 26262. However, we present a concise definition of the *safe state* because of inconsistent definitions in ISO 26262, ISO/DIS 21448 [46], and ISO/TR 4804.

In general, *fault tolerance regimes* can be used either ex ante to define system requirements or ex post when evaluating a system's fault tolerance. Furthermore, we recognize the potential of our definitions to also work for domains other than the automotive domain, although we rely primarily on terms from automotive standards as a basis for our taxonomy (cf. Subsection III-B). The automotive standards are in general compatible with more generic technical standards (e.g., IEC 61508 [67]). Additionally, the presented example for a steer-by-wire system may also be applied to any other redundant actuation system of another technical domain, wherein torque and actuated way are relevant factors to fulfill the specified functionality.

Finally, we would like to point out that the taxonomy presented by us is pivotally based on a harmonization of related works and standards and, therefore, has primarily a theoretical foundation. The validation of the applicability of our definitions in industrial series development of a safety-critical system is still pending. In particular, it could be questioned whether the discretization of faults and fault combinations, which is implicitly introduced by the definitions of fault tolerance regimes, is actually practicable in real complex systems.

Our future work targets an extension of the taxonomy towards *safety of the intended functionality* according to ISO/DIS 21448 [46]. Finally, an in-depth discussion towards linking the understanding of safety in the engineering and the legal domain is necessary, in particular for automated driving.

ACKNOWLEDGMENT

The authors would like to thank Moritz Lippert and Tom Michael Gasser for discussing the contents of this paper as well as Sonja Luther and Ibrahim Khan for proofreading.

REFERENCES

[1] T. Woopen *et al.*, "UNICAR*agil* – Disruptive Modular Architectures for Agile, Automated Vehicle Concepts," in *27th Aachen Colloq.*, Aachen, Germany, 2018. DOI: 10.18154/RWTH-2018-229909.

[2] R. Adler *et al.*, "Engineering and Hardening of Functional Fail-Operational Architectures for Highly Automated Driving," in *2019 IEEE Int. Symp. Softw. Rel. Eng. Workshops (ISSREW)*, Berlin, Germany: IEEE, pp. 30–35. DOI: 10.1109/ISSREW.2019.00038.

[3] A. Bartels, U. Eberle, and A. Knapp, "System Classification and Glossary," AdaptIVe Consortium, Wolfsburg, Germany, Deliverable D2.1, 2015.

[4] J. Becker and M. Helmle, "Architecture and System Safety Requirements for Automated Driving," in *Road Veh. Automat. 2*, Cham, Switzerland: Springer Int. Publishing, 2015, pp. 37–48. DOI: 10.1007/978-3-319-19078-5_4.

[5] J. Becker, M. Helmle, and O. Pink, "System Architecture and Safety Requirements for Automated Driving," in *Automated Driving*, D. Watzenig and M. Horn, Eds., Cham, Switzerland: Springer Int. Publishing, 2017, pp. 265–283. DOI: 10.1007/978-3-319-31895-0_11.

[6] F. Bertino, D. de Simone, and L. Piegari, "Design and operation of a fail-operational 5kW 800V-12V DC-DC converter," in *2019 21st Eur. Conf. Power Electron. Appl. (EPE '19 ECCE Europe)*, Genova, Italy: IEEE, pp. 1–10. DOI: 10.23919/EPE.2019.8915502.

[7] J. Beyerer *et al.*, "General Fail-Safe Emergency Stopping for Highly Automated Vehicles," presented at the 9. Tagung Automatisiertes Fahren, Munich, Germany, 2019.

[8] T. Bijlsma and T. Hendriks, "A fail-operational truck platooning architecture," in *2017 IEEE Intell. Veh. Symp. (IV)*, Los Angeles, CA, USA: IEEE, pp. 1819–1826. DOI: 10.1109/IVS.2017.7995970.

[9] T. Fruehling *et al.*, "Architectural Safety Perspectives Considerations Regarding the AI-based AV Domain Controller," in *2019 IEEE Int. Conf. Connected Veh. Expo (ICCVE)*, Graz, Austria: IEEE. DOI: 10.1109/ICCVE45908.2019.8965197.

[10] M. Goth, D. Keilhoff, and H.-C. Reuss, "Fault Tolerant Electric Energy Supply System Design for Automated Electric Shuttle Bus," in *20. Int. Stuttgarter Symp.*, ser. Proc. Vol. 2, Stuttgart, Germany: Springer Vieweg, Wiesbaden, Germany, 2020, pp. 441–455. DOI: 10.1007/978-3-658-30995-4_40.

[11] M. Helmle, F. Hauler, P. Wörz, A. Rühle, and M. Fausten, "Transient system safety and architectural requirement for partly and highly automated driving functions," in *Elect. Electron. Syst. Hybrid Elect. Veh. Elect. Energy Manage.* U. Brill, Ed., Essen, Germany: expert verlag, 2014, pp. 413–423.

[12] M. Klomp, M. Jonasson, L. Laine, L. Henderson, E. Regolin, and S. Schumi, "Trends in vehicle motion control for automated driving on public roads," *Veh. Syst. Dyn.*, vol. 57, no. 7, pp. 1028–1061, 2019. DOI: 10.1080/00423114.2019.1610182.

[13] S. Magdici and M. Althoff, "Fail-safe motion planning of autonomous vehicles," in *2016 IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Rio de Janeiro, Brazil: IEEE, pp. 452–458. DOI: 10.1109/ITSC.2016.7795594.

[14] J. A. Matute-Peaspan, J. Perez, and A. Zubizarreta, "A Fail-Operational Control Architecture Approach and Dead-Reckoning Strategy in Case of Positioning Failures," *Sensors*, vol. 20, no. 2, p. 442, 2020. DOI: 10.3390/s20020442.

[15] M. Möstl, D. Thiele, and R. Ernst, "Towards Fail-Operational Ethernet Based In-Vehicle Networks," in *Proc. 53rd Annu. Des. Automat. Conf. – DAC '16*, Austin, Texas: ACM Press, 2016. DOI: 10.1145/2897937.2905021.

[16] D. Niedballa and H.-C. Reuss, "Concepts of Functional Safety in E/E-Architectures of Highly Automated and Autonomous Vehicles," in *20. Int. Stuttgarter Symp.*, M. Bargende, H.-C. Reuss, and A. Wagner, Eds., ser. Proc. Vol. 2, Stuttgart, Germany: Springer Vieweg, Wiesbaden, 2020, pp. 457–470. DOI: 10.1007/978-3-658-30995-4_41.

[17] S. Ramanathan Venkita, B. Boulkroune, A. Mishra, and E. Van Nunen, "A Fault Tolerant Lateral Control Strategy for an Autonomous Four Wheel Driven Electric Vehicle," in *2020 IEEE Intell. Veh. Symp. (IV)*, Las Vegas, NV, USA: IEEE, pp. 1221–1226. DOI: 10.1109/IV47402.2020.9304704.

[18] B. Sari, "Fail-operational Safety Architecture for ADAS/AD Systems and a Model-driven Approach for Dependent Failure Analysis," Ph.D. dissertation, Univ. Stuttgart, Stuttgart, Germany, 2020.

[19] P. Sinha, "Architectural design and reliability analysis of a fail-operational brake-by-wire system from ISO 26262 perspectives," *Rel. Eng. Syst. Saf.*, vol. 96, no. 10, pp. 1349–1359, 2011. DOI: 10.1016/j.ress.2011.03.013.

[20] T. Stolte, G. Bagschik, and M. Maurer, "Safety Goals and Functional Safety Requirements for Actuation Systems of Automated Vehicles," in *2016 IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Rio de Janeiro, Brazil: IEEE, pp. 2191–2198. DOI: 10.1109/ITSC.2016.7795910.

[21] G. Weiß, P. Schleiß, and C. Drabek, "Fail-operational E/E Architecture for Highly-automated Driving Functions," *ATZelektron. worldw.*, vol. 11, no. 3, pp. 16–21, 2016. DOI: 10.1007/s38314-016-0032-8.

[22] ——, "Towards Flexible and Dependable E/E-Architectures for Future Vehicles," presented at the 4th Int. Workshop Crit. Automot. Appl.: Robustness Saf. (CARS), Gothenburg, Sweden, 2016.

[23] B. Witte *et al.*, "Fail-operational chassis for highly and fully automated trucks," in *7th Int. Munich Chassis Symp. 2016*, P. E. Pfeffer, Ed., ser. Proc. Munich, Germany: Springer Fachmedien Wiesbaden, 2017, pp. 295–305. DOI: 10.1007/978-3-658-14219-3_22.

[24] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, SAE Int. Standard J3016_202104.

[25] S. Benz, "Eine Entwicklungsmethodik für sicherheitsrelevante Elektroniksysteme im Automobil," (in German), Ph.D. dissertation, Univ. Karlsruhe, Karlsruhe, Germany, 2004.

[26] M. Carré, "Autonomic Framework For Safety Management In The Autonomous Vehicle," Ph.D. dissertation, Univ. de Pau et des Pays de l'Adour, Pau, France, 2020.

[27] X. Chen, "Requirements and concepts for future automotive electronic architectures from the view of integrated safety," Ph.D. dissertation, Univ. Karlsruhe, Karlsruhe, Germany, 2008.

[28] M. Gleirscher and S. Kugele, "Assurance of System Safety: A Survey of Design and Argument Patterns," 2019. arXiv: 1902.05537.

[29] R. Isermann, R. Schwarz, and S. Stölzl, "Fault-tolerant Drive-by-Wire Systems: Concepts and Realizations," in *Proc. 4th IFAC Symp. Fault Detection, Supervision Saf. Tech. Processes (SAFEPROCESS)*, ser. IFAC Proc. Volumes, vol. 33, no. 11, Budapest, Hungary: IFAC, 2000, pp. 1–15. DOI: 10.1016/S1474-6670(17)37335-4.

[30] ——, "Fault-tolerant drive-by-wire systems," *IEEE Control Syst. Mag.*, vol. 22, no. 5, pp. 64–81, 2002. DOI: 10.1109/MCS.2002.1035218.

[31] ISO, "Road vehicles – Safety and cybersecurity for automated driving systems – Design, verification and validation," Int. Org. Standardization, Geneva, Switzerland, Tech. Rep. ISO/TR 4804:2020.

[32] M. Li and L. Eckstein, "Fail-Operational Steer-By-Wire System for Autonomous Vehicles," in *2019 IEEE Int. Conf. Veh. Electron. Saf. (ICVES)*, Cairo, Egypt: IEEE. DOI: 10.1109/ICVES.2019.8906395.

[33] M. A. Martinus, "Funktionale Sicherheit von mechatronischen Systemen bei mobilen Arbeitsmaschinen," (in German), Ph.D. dissertation, Tech. Univ. München, Munich, Germany, 2004.

[34] M. Mauritz, "Engineering of Safe Autonomous Vehicles through Seamless Integration of System Development and System Operation," Ph.D. dissertation, Tech. Univ. Clausthal, Clausthal, Germany, 2019.

[35] R. Messnarz, G. Macher, J. Stolfa, and S. Stolfa, "Highly Autonomous Vehicle (System) Design Patterns – Achieving Fail Operational and High Level of Safety and Security," in *Syst., Softw. Services Process Improvement*, ser. Commun. Comput. Inf. Sci. A. Walker, R. V. O'Connor, and R. Messnarz, Eds., vol. 1060, Cham: Springer Int. Publishing, 2019, pp. 465–477. DOI: 10.1007/978-3-030-28005-5_36.

[36] K. Reif, *Automobilelektronik: Eine Einführung für Ingenieure*, (in German). Wiesbaden, Germany: Springer Fachmedien Wiesbaden, 2014. DOI: 10.1007/978-3-658-05048-1.

[37] J. Schäuffele and T. Zurawka, Eds., *Automotive Software Engineering*, 2nd ed., trans. by R. Carey. Warrendale, PA, USA: SAE Int., 2016.

[38] T. Schmid, S. Schraufstetter, S. Wagner, and D. Hellhake, "A Safety Argumentation for Fail-Operational Automotive Systems in Compliance with ISO 26262," in *2019 4th Int. Conf. Syst. Rel. Saf. (ICSRS)*, Rome, Italy: IEEE, pp. 484–493. DOI: 10.1109/ICSRS48664.2019.8987656.

[39] A. Schnellbach, M. Hirz, and J. Fabian, "Comparison of fail-operational software architectures from the viewpoint of an automotive application," *Elektrotech. Inf. Tech.*, vol. 133, no. 6, pp. 283–293, 2016. DOI: 10.1007/s00502-016-0420-z.

[40] A. Schnellbach, "Fail-operational automotive systems," Ph.D. dissertation, Tech. Univ. Graz, Graz, Austria, 2016.

[41] R. Stetter, *Fault-Tolerant Design and Control of Automated Vehicles and Processes: Insights for the Synthesis of Intelligent Systems*, red. by J. Kacprzyk, ser. Stud. Syst., Decis. Control. Cham, Switzerland: Springer Int. Publishing, 2020, vol. 201. DOI: 10.1007/978-3-030-12846-3.

[42] E. Thorn, S. Kimmel, and M. Chaka, "A Framework for Automated Driving System Testable Cases and Scenarios," Nat. Highway Traffic Saf. Admin., Washington, DC, USA, Final Rep. DOT HS 812 623, 2018, p. 180.

[43] D. Wanner, A. Trigell, L. Drugge, and J. Jerrelind, "Survey on Fault-Tolerant Vehicle Design," *World Elect. Veh. J.*, vol. 5, no. 2, pp. 598–609, 2012. DOI: 10.3390/wevj5020598.

[44] M. Wood *et al.*, "Safety First for Automated Driving," Aptiv, Audi, Baidu, BMW, Continental, Daimler, Fiat Chrysler Automob., Here, Infineon, Intel, Volkswagen, White Paper, 2019.

[45] *Road Vehicles – Functional Safety*, Int. Org. Standardization Standard ISO 26262:2018.

[46] *Road Vehicles – Safety of the Intended Functionality*, Int. Org. Standardization Standard ISO/DIS 21448:2021.

[47] Economic Commission for Europe, "Revised Framework document on automated/autonomous vehicles," United Nations Econ. Social Council, Geneva, Switzerland, ECE/TRANS/WP.29/2019/34/Rev.2, 2020.

[48] M. Gleirscher, "Behavioural Safety of Technical Systems," Ph.D. dissertation, Tech. Univ. München, Munich, Germany, 2014.

[49] Y. Luo, A. K. Saberi, and M. van Brand, "Safety-Driven Development and ISO 26262," in *Automot. Syst. Softw. Eng.* Y. Dajsuren and M. van den Brand, Eds., Cham, Switzerland: Springer Int. Publishing, 2019, pp. 225–254. DOI: 10.1007/978-3-030-12157-0_10.

[50] M. Blanke, W. Christian Frei, F. Kraus, J. Ron Patton, and M. Staroswiecki, "What is Fault-Tolerant Control?" In *4th IFAC Symp. Fault Detection, Supervision Saf. Tech. Processes (SAFEPROCESS)*, ser. IFAC Proc. Volumes, vol. 33, no. 11, Budapest, Hungary: IFAC, 2000, pp. 41–52. DOI: 10.1016/S1474-6670(17)37338-X.

[51] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Trans. Dependable Secure Comput.*, vol. 1, no. 1, pp. 11–33, 2004. DOI: 10.1109/TDSC.2004.2.

[52] J. Knight, *Fundamentals of Dependable Computing for Software Engineers*, ser. Chapman Hall/CRC Innov. Softw. Eng. Softw. Develop. Boca Raton, FL, USA: Chapman Hall/CRC, 2012. DOI: 10.1201/b11667.

[53] H. Kopetz, *Real-Time Systems – Design Principles for Distributed Embedded Applications*, 2nd ed., red. by J. A. Stankovic, ser. Real-

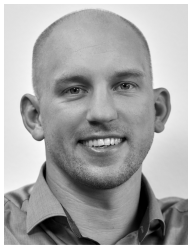
Time Syst. Ser. Boston, MA: Springer US, 2011. DOI: 10.1007/978-1-4419-8237-7.
[54] IAEA, "Safety related terms for advanced nuclear plants," Int. At. Energy Agency, Vienna, Austria, IAEA-TECDOC-626, 1991.
[55] N. Möller and S. O. Hansson, "Principles of engineering safety: Risk and uncertainty reduction," *Rel. Eng. Syst. Saf.*, vol. 93, no. 6, pp. 798–805, 2008. DOI: 10.1016/j.ress.2007.03.031.
[56] NASA, "ISS Safety Requirements Document," Nat. Aeronaut. Space Admin., Houston, TX, USA, Baseline SSP 51721, 2019.
[57] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*, 3rd ed. Berlin, Heidelberg, Germany: Springer, Berlin, Heidelberg, 2016. DOI: 10.1007/978-3-662-47943-8.
[58] D. D. Walden, G. J. Roedler, K. Forsberg, R. D. Hamelin, and T. M. Shortell, Eds., *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, 4th ed. Hoboken, NJ, USA: Wiley, 2015.
[59] T. M. Gasser, *Discussion Safety/Safe State/Risk Minimal Condition*, (in German), E-mail, 2021.
[60] J. Jacobson, L.-Å. Johansson, and M. Lundin, "Safety of Distributed Machine Control Systems," Swedish Nat. Testing Res. Inst., Borås, Sweden, SP REPORT 1996:23.
[61] A. Reschka and M. Maurer, "Conditions for a safe state of automated road vehicles," *it – Inf. Technol.*, vol. 57, no. 4, pp. 215–222, 2015.
[62] R. Matthaei and M. Maurer, "Autonomous driving – a top-down-approach," *Automatisierungstechnik*, vol. 63, no. 3, pp. 155–167, 2015. DOI: 10.1515/auto-2014-1136.
[63] S. Ulbrich *et al.*, "Towards a Functional System Architecture for Automated Vehicles," 2017. arXiv: 1703.08557.
[64] S. Ackermann and H. Winner, "Systemarchitektur und Fahrmanöver zum sicheren Anhalten modularer automatisierter Fahrzeuge," (in German), in *Workshop Fahrerassistenz und automatisiertes Fahren*, Walting, Germany: Uni-DAS e.V., 2020.
[65] P. Koopman and F. Fratrik, "How Many Operational Design Domains, Objects, and Events?" In *Proc. AAAI Workshop Artif. Intell. Saf. 2019*, H. Espinoza, S. Ó hÉigeartaigh, X. Huang, J. Hernández-Orallo, and M. Castillo-Effen, Eds., ser. CEUR Workshop Proc. Vol. 2301, Honolulu, HI, USA: CEUR-WS, pp. 45–48.
[66] *Operational Design Domain (ODD) Taxonomy for an Automated Driving System (ADS) – Specification*, Brit. Standards Inst. Publicaly Available Specification PAS 1883:2020.
[67] *Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems*, Int. Electrotechnical Commission Standard IEC 61508:2010.

Torben Stolte studied Automation Technologies at Universität Lüneburg (Diplom (FH) 2008) and Electrical Engineering at Technische Universität Braunschweig (M. Sc. 2011). Since 2011 he is a research assistant at the Institute of Control Engineering of Technische Universität Braunschweig. Parallely, he worked in a collaboration with Porsche Engineering as functional safety engineer from 2011 to 2014. His research interest is safety of automated vehicles. He investigates the potential of fault-tolerant vehicle motion control towards safety of steering, brake, and drive actuators.

Stefan Ackermann finished his Bachelor of Science and Master of Science Degrees in Mechatronics in the Field of Automotive Mechatronics at Technical University of Darmstadt. Since 2017 he is a research assistant at the Institute of Automotive Engineering at Technical University of Darmstadt. His research focuses on safety applications for automated vehicles. While pursuing his PhD he develops fallback systems for the dynamic driving task of automated vehicles.

Robert Graubohm is a research assistant with the Institute of Control Engineering at the Technische Universität Braunschweig. Before that, he finished a Master of Science Degree in Industrial Engineering in the Field Mechanical Engineering at TU Braunschweig and a Master of Business Administration at the University of Rhode Island. His main research interests are development processes of automated driving functions and the safety conception in an early design stage.

Inga Jatzkowski works as a research assistant at the Institute of Control Engineering at TU Braunschweig since 2016 and is currently pursuing her PhD. She holds a Master of Science Degree in Navigation and Field Robotics from Leibniz University Hannover. Her main research topics are self-awareness and the development of self-perception for automated vehicles.



Björn Klamann finished his Master of Science Degree in Mechanical and Process Engineering at Technical University of Darmstadt. Since 2018 he is a research assistant at the Institute of Automotive Engineering at Technical University of Darmstadt. In his main research topic, the safety of automated vehicles, he investigates the approach of a modular safety approval.

Hermann Winner began working at Robert Bosch GmbH in 1987, after receiving his PhD in physics, focusing on the predevelopment of "by-wire" technology and Adaptive Cruise Control (ACC). Beginning in 1995, he led the series development of ACC up to the start of production. Since 2002 he has been pursuing the research of systems engineering topics for driver assistance systems and automated driving as Professor of Automotive Engineering at the Technische Universität Darmstadt. He discovered the "approval trap" of autonomous driving, the still unsolved challenge to validate safety of autonomous driving before market introduction.

Markus Maurer received the Diploma degree in electrical engineering from the Technische Universität München, in 1993, and the Ph.D. degree in automated driving from the Group of Prof. E. D. Dickmanns, Universität der Bundeswehr München, in 2000. From 1999 to 2007, he was a Project Manager and the Head of the Development Department of Driver Assistance Systems, Audi. Since 2007, he has been a Full Professor of automotive electronics systems with the Institute of Control Engineering, Technische Universität Braunschweig. His research interests include both functional and systemic aspects of automated road vehicles. (Picture: Jessen Oestergaard / Daimler und Benz Stiftung)