Analysis of Safety of The Intended Use (SOTIF)

Edward Schwalb
MSC Software

***Abstract*—The ISO/PAS 21448 Safety Of The Intended Function (SOTIF) standard recommends the continuous improvement of autonomous drivers to maximize the portion of known safe scenarios. Whereas control on individual outcomes is limited, engineering processes must exert control over the continuous improvement of overall performance statistics. The required quantitative analysis is performed by combining residual risk analysis, hazard modeling and Bayesian probabilistic reasoning. To account for the scope increase associated with automation above level 2, we extend mainstream standards to encompass multiple vehicles. We clarify how sneak conditions contribute to unsafe outcomes without failure, and explain how traditional tools need to be extended. We detail methods for modeling a driving loop in which perception algorithms are trained to detect hazards and the decision logic actively avoids accidents. The resulting unified hazard models rely on quantitative phenomenological characteristics that can be calibrated against ground truth obtained on the test track. We propose a scenario requirement specification process, and describe an incremental process for assembling scenario libraries which can be used for continuous integration. We show how to apply machine learning paradigm to enable development against unknown scenarios and enable interpretability of driver logic. Finally, we show how to quantitatively compare autonomous drivers and discriminate incremental improvements from regression using posterior probabilities.**

***Index Terms*—Autonomous Vehicles, Testing, Simulation, Scenarios, Hazards, Quantifying Performance, Reliability, Safety**

I. Introduction

A. Appropriate Statistics

There are two main factors that contribute to vehicle fatalities: the number of crashes per vehicle, and the number of passengers or pedestrians involved in each crash. Whereas the number of crashes per vehicle depends on the driver and vehicle technology, the number of passengers or pedestrians involved in a crash does not. Further, as opposed to human drivers, an autonomous driver will be bound to the vehicle. Thus, the statistics of about 1.16 fatalities per 100 million miles in the US [11] is misleading. Instead, the appropriate statistics is the ratio of the number of fatality crashes to the number of Vehicle Miles Traveled (VMT)[1]. Given that there are about ≈30 driver deaths per million vehicle-years for all vehicles [29], that less than 2% of crashes are fatal (see **Figure 7**) and about 12,000 miles driven per vehicle annually [31], we observe approximately >3 million VMT between incidents. Consequently, achieving a confidence factor requires testing Autonomous Vehicles (AV) for the equivalent of 50 million *naturalistic miles*, which can be completed within a single year using a fleet of 1000 vehicles each driving on public roads ≈300 miles daily.

[1]Each crash involves a driver, and may involve passengers or pedestrians, with various degrees of injuries.

There is a gap between human performance and the best-reported self-driving car performance of about 10^4 miles per disengagement [14]. After considering that disengagement are far more frequent than incidents, and that less than 2% of incidents are fatal (see **Figure 7**), the gap is of the order of 10x-100x *for the limited Operational Design Domains (ODD) deployed*. During 2017 and 2018 AVs were involved in >100 accidents. A comparison of the AV accidents to the SHRP naturalistic driving study data [12] reveals that autonomous drivers were involved in more injuries and property damage than a comparable general population of drivers [38]. Most of the accidents occurred at very low speed, typically at <15mph, mostly human drivers rear-ending the AV and rarely the fault of the AV. Nevertheless, further analysis of accident descriptions reveals that AVs are significantly worse than the general population of human drivers at avoiding preventable crashes [38]. Such avoidance should leverage *leading metrics* (as opposed to *lagging metrics*) [26].

Key Point 1: Safe AV are practical but there is a need for industry wide consensus on what are the metrics and the goals.

B. Data Collection Requirements

It is critical that the traffic data collected supports comparing performance of an AV to that of a human driver. According to NHTSA's NMVCCS database, analyzing the 94% of crashes attributed to human error reveals that 41% of human errors were attributed to perception challenges, 33% were decision errors, 11% are performance errors (>half attributed to non-performance) [10]. Indeed, an independent Insurance Institute Highway Safety (IIHS) Highway Loss Data Institute (HLDI) evaluation of existing prototype testing resulted in the following conclusion: "We urge NHTSA to consider wherever possible making its granting of exemptions contingent upon [the manufacturer] sharing test data ... demonstrating that the [AV] is designed in ways that improve upon the [recognition] and decisions of human drivers" [39].

Measurement of safety can be performed in the context of a scenario. As an example, one can measure the probability of a crash for a scenario such as following a vehicle making a maneuver. Using the Bayesian approach [20], this measure is denoted by $P(crash \mid scenario)$. Data can be collected to enable comparing such a measure for humans vs autonomous drivers, across all scenarios reported by [15]. Unfortunately, the data collected traditionally is insufficient. Specifically, the NHTSA data reported in [15] represents the distribution of pre-crash scenarios, denoted by $P(scenario \mid crash)$. To estimate the target metric of $P(crash \mid scenario)$ there is a need to leverage the following Bayesian formulation:

$$P(crash \mid scenario) = P(scenario \mid crash)\frac{P(crash)}{P(scenario)}$$

The advantage of the Bayesian approach is the *quantitative specification of smart miles*: Using simulation to skew the prior $P(scenario)$, to increase the rate of rare events by a factor of, e.g. α=1000x, results in increasing the effectiveness (of simulation over naturalistic miles) for estimating $P(scenario \mid crash)$ by the same α factor. Given the pre-crash scenario data available, to estimate $P(crash \mid scenario)$ requires knowledge of both priors $P(crash)$ and $P(scenario)$. Whereas the prior $P(crash)$ can be estimated from general crash data, the prior $P(scenario)$ can be measured by classifications of scenarios extracted from vehicle instrumentation and data collected by telematics companies [44] typically targeting the Usage Base Insurance (UBI) market.

Key Point 2: Leverage telematics to collect reliable scenario frequency data and enable context specific comparison of safety between autonomous vehicles and humans.

C. Scope of The Challenge

A review of the pre-crash scenarios reported by the US Department of Transportation National Highway Traffic Safety Administration (NHTSA), combining the data in Table 6 depicted in **Figure 2**, and Tables 10, 11 and 12, reveals that only a small fraction, less than 1% of the crashes, are attributed to vehicle failure [15]. Thus, AV safety is a multi-agent problem [33]: Whereas about 30% of the crashes are associated with a single vehicle, about 63% are associated with 2 vehicles and about 6.3% are associated with 3 or more vehicles, as depicted in **Figure 1**. Similarly, 22 of 37 crash scenarios are associated with multiple vehicles.

Mainstream safety engineering methods, such as Fault Tree Analysis (FTA) IEC 61025 [4], Failure Mode Effect Analysis (FMEA) [5], [6], and standards such as the US Federal Motor Vehicle Safety Standards (FMVSS) 1-series [8], and Functional Safety Standard ISO-26262 [2], are focused on single vehicle failures. This is appropriate for vehicles driven by humans.

In contrast, for AVs having a driving responsibility similar to that of humans, such a focus accounts for less than 1% of the crashes, and thus less than 1% of the risk. Consequently, the scope of the FTA, FMEA models and technical standards such as FMVSS and ISO-26262 need to account for the multi-agent aspects of driver behavior and other traffic participants.

Key Point 3: Improve safety engineering methods and technical standards to account for multiple agents.

D. Example Scenario #1: Backing Up Into Another Vehicle

To illustrate an important 2-vehicle NHTSA scenario, consider the autonomous shuttle grazing accident which occurred in November 2018. The shuttle encountered a semi-trailer truck which was backing up to drop a delivery. The truck was slowly backing up, with the appropriate audio alert, and as the passengers watched, it eventually grazed the shuttle, as depicted in **Figure 3**; no passengers were injured.

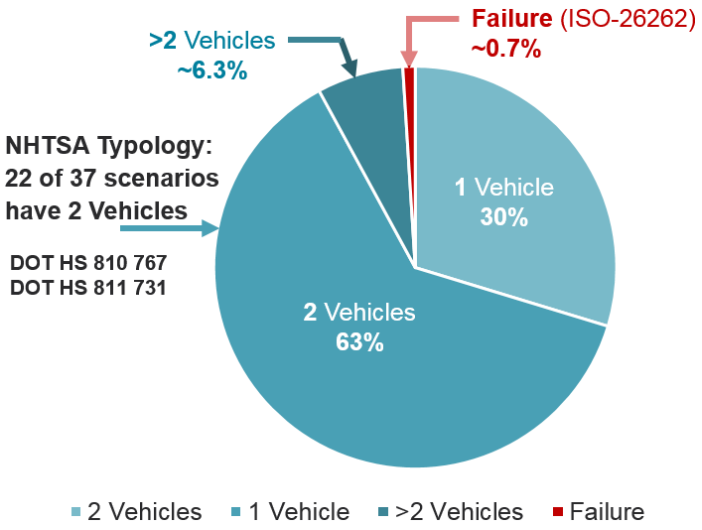


Fig. 1: Distribution of Crashes per NHTSA [15].

	Type	Number of Crashes	Number of Persons	Number of Injured Persons	Persons per Crash	Injured Persons per Crash
1-Vehicle Crash	All	1,879,000	2,657,000	709,000	1.41	0.38
	Light	1,673,000	2,398,000	637,000	1.43	0.38
	Light/All	89.0%	90.3%	89.8%	1.01	1.01
2-Vehicle Crash	All	3,890,000	10,885,000	1,722,000	2.80	0.44
	Light	3,869,000	10,829,000	1,712,000	2.80	0.44
	Light/All	99.5%	99.5%	99.4%	1.00	1.00
Greater Than 2-Vehicle Crash	All	401,000	1,801,000	388,000	4.49	0.97
	Light	401,000	1,800,000	388,000	4.49	0.97
	Light/All	99.9%	99.9%	100.0%	1.00	1.00

Fig. 2: Severity by Number of Vehicles Involved [15].

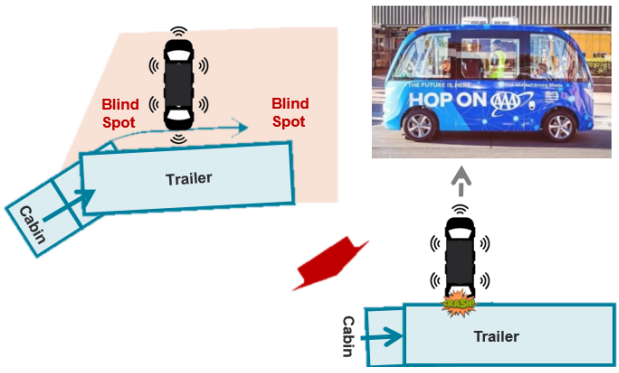


Fig. 3: Shuttle Grazing Example

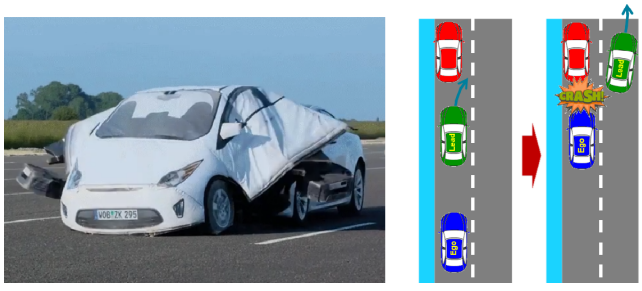


Fig. 4: ADAS Cruise Control Crash Example

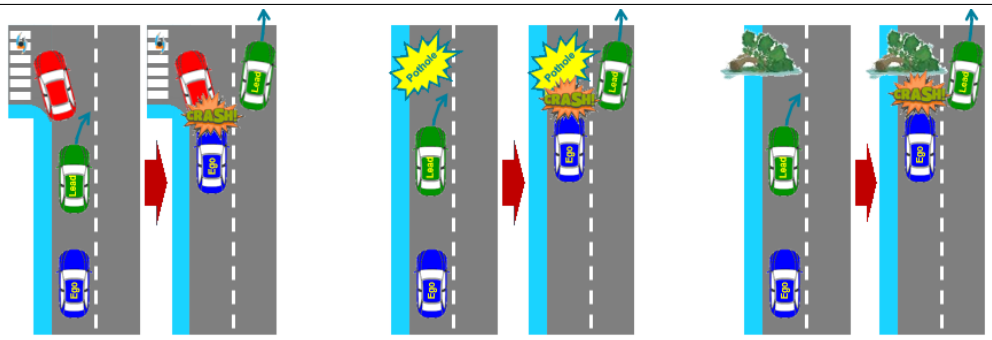


Fig. 5: Scenarios Equivalent to the ADAS Cruise Control Crash Example

City officials indicated that: "The shuttle did what it was supposed to do, in that its sensors registered the truck and the shuttle stopped". Such a response was deemed inappropriate, because, e.g. if the truck was a meter further it could have toppled the shuttle with passengers inside. The passengers indicated that "We had about 20 feet of empty street behind us, and most human drivers would have thrown the car into reverse and used some of that space to get away from the truck. Or at least leaned on the horn and made our presence harder to miss."

Clearly, the requirements for this 2-agent scenario must include avoiding an accident by moving in reverse. According to the NHTSA statistics, *backing into another vehicle accounts for $>$ 3% of the crashes*. Failing to include such a requirement in the development phase, and further failing to validate it prior to product release, is inappropriate.

Key Point 4: Require explicit specification of procedures to actively avoid crashes due to other traffic participants.

E. Example Scenario #2: Following Vehicle Making Maneuver

To illustrate an important 3-vehicle NHTSA scenario, consider the safety of commonly deployed Advanced Driver Assistance (ADAS) cruise control. In a test conducted in June 2018, and depicted in **Figure 4** performed the following three steps:

- **Step 1:** Engage ADAS cruise control to follow the lead vehicle in front.
- **Step 2:** The lead vehicle moves to the right lane to avoid a static obstacle in front.
- **Step 3:** The Ego vehicle faces the static obstacle, unable to brake, and crashes into that obstacle.

This example illustrates a problem which is common to millions of vehicles on the road as of 2019: The ADAS sensors are only capable of analyzing the state of the lead vehicle immediately in front, but unable to analyze the state of the vehicle in front of it. All recent vehicle model ADAS systems deployed before 2019 exhibit this problem. The test depicted in **Figure 4** was conducted on Jun 12, 2018, with Tesla vehicles, two years after the release of the radar designed to detect such situations [41]. It turns out that humans are challenged by this scenario as well: According to NHTSA statistics, *following a vehicle making maneuver accounts for >1% of the crashes*.

This example represents an entire class of scenarios, depicted in **Figure 5**. The obstacle may be a vehicle stopped to allow a pedestrian to cross safely. Alternatively, the obstacle may be a pothole or a flooded section of the road which does not appear in the map. Similarly, the obstacle may be a tree, a pedestrian or other objects which fell onto the road and are *not mapped*. Clearly, there are countless possibilities represented by "following vehicle making maneuver", for which an autonomous driver must perform better than humans.

Key Point 5: Require testing 3 vehicle scenarios, including the scenario of "following a vehicle making a maneuver".

F. Regulatory Approach

The regulatory approach, as described in [7], is to evaluate six possible validation methods:

Test As Is: Using an AV which controls itself conducts the testing directly against the off-the-shelf product. The challenge with this approach is the prohibitive cost of setting up countless complicated multi-agent situations.

Library of Compliance Tests: The manufacturer can provide a library of pre-programmed compliance tests. This method is also hampered by the prohibitive cost of achieving sufficient coverage of countless complicated physical multi-agent tests. Further apriory knowledge of the compliance tests enables engineering processes to be tailored to passing those tests (="overfit"), reducing their validity.

External Controller: An AV remote control mechanism extends the above library approach by enabling the execution of tests not disclosed to the manufacturer. This option relies on undeveloped technology, and may be associated with security risks. Further, this option is also hampered by the prohibitive cost of achieving sufficient coverage.

Simulation Testing: A simulated environment controlled by the testing authority is cost effective and overcomes the barriers for achieving coverage. The most significant challenge is to ensure validity and repeatability of simulation results. State-of-the-art validation methods are not mature yet.

Technical Documentation: Review of technical design documentation and testing results has been effective for non-autonomous system; static source code reviews are commonly used for development of tests and assessment of correctness. The challenge is that documenting the design of e.g. a neural network, is not informative about its functionality and reliability; for interpretability, see [49], [50], [51], [52]. It is not clear what would be required and sufficient documentation.

Fig. 6: Accidents 100 years ago [13].

Surrogate Vehicles: Use of surrogate vehicles with human drivers to evaluate performance of an AV. This option is only applicable to level 3, whereas the scope of validation must include ability to test levels 4,5 without a driver.

Key Point 6: The regulatory approach needs to acknowledge that, as depicted in **Figure 1**, rather than extend current methods, we must account for the full scope of multi-agent scenarios.

The situation in 2019 is similar to the situation during the development of cars in the early 20th century. At that time, traffic indicators and rules were not understood. Driving test regulation was not enacted until 1950s, two generations later! Without the foundation of traffic safety in place, the early years of driving were chaotic, stressful, and dangerous [13]. A photo of an accident is depicted in **Figure 6**. "Screaming pedestrians were scattered like ninepins some were bowled over or tossed against store fronts." Detroit Free Press, Jan. 20, 1919

II. Risk and Hazards

Identification and classification of hazards is at the core of safety engineering and validation. Hazards can be classified as one of three distinct categories:

- Hazards originating from within the autonomous vehicle (AV) system, due to system limitations or failures. The scope of ISO-26262 covers these hazards.
- Hazards originating due to misuse by an operator. Intentional misuse is regarded as a security topic, and is not in scope for safety standards. Unintentional misuse is in scope, however the nature of that misuse determines which safety standard is applicable.
- Hazards originating from environment objects, which are either static or dynamic. Dynamic object include other traffic participants, e.g., vehicles, pedestrians, etc. Static objects include mapped and unmapped objects. Mapped objects include, for example, road geometry, road markings and indicators, speed bumps, rumble strips, etc. Unmappped static objects include trees which fell on the road and potholes, temporary road indicators, e.g. workzones, etc.

According to the SOTIF standard, system failures and unintentional misuse are regarded as internal hazards, whereas

Crash Severity	Crash Incidents	% of Injuries	K Person Injuries	A Person Injuries	B Person Injuries	C Person Injuries	O Prop Damage
K = Fatality	4,502	1.8%	4,901	1,353	1,284	1,027	2,357
A = Sever Injury	33,247	13.0%	0	39,711	7,696	7,976	26,668
B = Major Injury	62,474	24.5%	0	0	74,811	12,247	66,451
C = Minor Injury	39,222	15.4%	0	0	0	55,767	52,985
O = No Injury / PDO (Property Damage Only)	115,451	45.3%	0	0	0	0	261,971
Total	**254,896**	**100%**	**4,901**	**41,064**	**83,791**	**77,017**	**410,432**

Fig. 7: Relative Frequencies of Injury Types [16].

intentional misuse and environment objects are regarded as external hazards. However, unintentional misuse can occur due to external factors as well. As an example, consider, unintentional misuse due to inability to comply with temporary geographic restrictions issued by external factors, e.g. authorities, due to traffic conditions or emergency situations. Also consider unintentional misuse due to unexpected weather conditions or unexpected GPS signal degradation. These examples motivate a distinct misuse category. Consequently, it may be appropriate to provide a separate classification for misuse to account for both internal and external factors.

A. Estimating Risk

Our approach is to extend the well established risk estimation approach used by ISO-26262 practitioners to encompass multiple vehicles. According to the ISO-26262 methodology, the risk is assessed by estimating the Severity S, Exposure E and Controllablability C of a hazard [2]. In practice, the values for S can be limited to either *property damage* or *injury* or *fatality*; see **Figure 7** for relative frequency of injury severities. The values for E could be $0 < E < 1$ representing the probability that a hazard would occur. The values for C could be $0 < C < 1$ whereby $C = 1$ represents *inability* of the vehicle to avoid the materialization of hazard (e.g., crash is guaranteed), and $C = 0$ represents *always avoiding the materialization* (e.g., crash is always avoided). With this approach, hazards are grouped according to their severity. Denoting the hazards in each group as $\mathcal{H}$, the corresponding risk R_H for each hazard $H \in \mathcal{H}$ is always $R_H = E_H \times C_H$ where:

$$\begin{cases} H \in \mathcal{H}_{pdo} & if \;\; S_H = "property \;\; damage \;\; only" \\ H \in \mathcal{H}_{injury} & if \;\; S_H = "injury" \\ H \in \mathcal{H}_{fatality} & if \;\; S_H = "fatality" \end{cases} \tag{1}$$

In contrast to ISO-26262 for which hazards represent failure of a single vehicle, estimating the residual risk associated with autonomous driving requires considering hazards comprising of multiple vehicles in the context of specific scenarios.

Example 1: To illustrate the various categories, consider the data obtained from Table 30 of [16] based on the 2010 MAIS and KABCO data, depicted in **Figure 7**. We observe that less than 2% of the crash incidents results in a fatal injury, i.e. hazards in $\mathcal{H}_{fatality}$. Almost half result in property damage only, i.e. hazards in $\mathcal{H}_{pdo}$. The remaining half is associated with multiple injury severitries, i.e. hazards in $\mathcal{H}_{injury}$.

For a collection of *disjoint* hazards $\mathcal{H}$, one can estimate the overall associated residual risk as the sum of the individual residual risks. When hazards can compound, however, there is a need to assess the *compounded* severity, exposure and controllability. For simplicity, it is reasonable to assume that the severity of the compound hazard equals the maximum severity of each of its components. Denoting $H = \{h_1, ..., h_n\} \subset \mathcal{H}$ as the set of hazards which may compound, the overall residual risk *for each severity type* ('property damage' or 'injury' or 'fatality') is:

$$R_{\mathcal{H}} = \sum_{\forall H \subset \mathcal{H}} E_H \times C_H \tag{2}$$

Given that there are 2^n subsets to consider for a set $\mathcal{H}$ of n hazards, the exponential complexity renders explicit computation impractical. There is a need to provide a good estimate without explicit consideration of all possible subsets.

B. Bounding Risk

Given that $R_H = E_H \times C_H$ and both E, C are between 0 and 1, it is clear that $R_H \leq E_H$ and $R_H \leq C_H$. Thus, an obvious method to provide an upper bound to R_H is to obtain an upper bound for E_H or C_H. We further simplify by assuming inability to avoid materialization of hazards, using $C = 1$. This bounds the compound risk by the joint probability of all hazards in $H \subset \mathcal{H}$ compounding, namely

$$R_H = R_{h_1,...,h_n} \leq P(h_1, ..., h_n) \tag{3}$$

Such simplification enables bounding the risk by estimating the probability density associated with materialization of any subset of the hazards in $\mathcal{H}$. More formally, the total risk R_h for every individual hazard $h \in \mathcal{H}$ is bounded by the sum of the probability that it would materialize in isolation, plus the probability that it would compound with any other subset of $\mathcal{H}$, namely:

$$R_h \leq \sum_{\forall H \subset \mathcal{H}} P(h \cup H) \tag{4}$$

To avoid considering hazard combination which are not likely to compound, $\mathcal{H}$ can be partitioned into k clusters $C_1, ..., C_k$ such that the probability of compounding hazards in different clusters is very low, namely

$$h_i \in C_i, \;\; h_j \in C_j \;\; \rightarrow \;\; P(h_i, h_j) << min(P(h_i), P(h_j)) \tag{5}$$

With those clusters, the complexity of computing the residual risk is linear in the number of clusters, but the quality of the approximation degrades exponentially with their size:

$$R_{\mathcal{H}} \leq \sum_{i=1}^{k} R_{C_i} \leq \sum_{i=1}^{k} 2^{|C_i|} \max_{\forall h \in C_i} P(h) \tag{6}$$

The bound can be made tighter by modeling the conditional probabilities and causality of the compounding hazards, e.g. using Probabilistic Causal Models [21]. A polynomial complexity bound for the full Bayesian inference can be controlled by mapping clusters to buckets and applying elimination [22].

C. Dynamic View of Hazards

The dynamic view of hazards is intended to formulate avoidance of preventable hazards by maximizing their Time To Materialization (TTM). According to this dynamic view, an autonomous driver needs to continuously monitor multiple potential hazards at all times, estimate their TTM, and act to defer (or avoid by deferring indefinitely) the materialization of the most imminent severe hazards.

Definition 1: The Time To Materialization (TTM) of a hazard as the estimated time delta at which a hazard will materialize.

Example 2: The metric of Time To Collision (TTC) represents the duration of time until the materialization of a crash, assuming no changes in velocity and direction. The TTC has proven to be an effective measure for discriminating critical from normal behaviour and identifying risky situations [9]. A situation is regarded as unsafe when TTC is lower than a minimum threshold. Autonomous drivers should continuously estimate TTC and select actions maximizing it.

Safe driving with dynamic hazards, as specified by **Algorithm 1**, continuously scans the environments, estimates the TTM for foreseable hazards, and applies the action which increases the TTM for the most imminent severe hazard.

Input : A collection of hazards $\mathcal{H} = \{H_1, ..., H_n\}$
Input : Hazard severities $\mathcal{R} = \{R_1, ..., R_n\}$
Input : Possible actions $\mathcal{A} = \{A_1, ..., A_k\}$
Dependency: Function $Sense()$ reading sensor data.
Dependency: Function $Perceive(data)$ estimating the current situation $\mathcal{S}$.
Dependency: Function $Monitor(H, \mathcal{S})$ estimating the TTM for a hazard H in a given situation $\mathcal{S}$.
Dependency: Function $Simulate(A, \mathcal{S})$ simulate consequence of applying action A to situation $\mathcal{S}$.
Dependency: Function $Act(A)$ applying action A.
Dependency: Function $Eval(\mathcal{T}, \mathcal{R})$ determining cost of TTMs $\mathcal{T}$ for severities $\mathcal{R}$.
Loop
  $\mathcal{S} \leftarrow Perceive(Sense())$;
  for *every hazard* $H_i \in \mathcal{H}$ **do**
    $T_i^0 \leftarrow Monitor(H_i, \mathcal{S})$
  end
  for *every actions* $A_i \in \mathcal{A}$ **do**
    $\mathcal{S}^{new} \leftarrow Simulate(A_i, \mathcal{S})$;
    for *every hazard* $H_j \in \mathcal{H}$ **do**
      $T_j \leftarrow Monitor(H_j, \mathcal{S}^{new})$
    end
    $C_i \leftarrow Eval(\{T_1^0, ..., T_n^0\}, \{T_1, ..., T_n\}, \mathcal{R})$;
  end
  $Act(A_i)$ for i minimizing C_i;
end

Algorithm 1: Driving Loop for Dynamic Hazards

Example 3: Consider the shuttle grazing example above. According to **Algorithm 1**, the shuttle's autonomous driver should maximize the TTM of the imminent crash by selecting A_i="reverse".

The systematic validation and safety monitoring of **Algorithm 1** is possible by observing the situation $\mathcal{S}$, the TTM estimates T_i^0, T_i, and A_i for every iteration of the control loop.

Key Point 7: Avoid explicit coding of rules while enabling validation of hazard avoidance using the *dynamic view of hazards*, the implied continuous TTM estimation and action selection.

D. Hazard Modeling

The key aspect of our approach is to decouple the measurement of TTM estimation error from the simulation based analysis of their implications, as depicted in **Figure 8**. We train the perception and fusion models to identify hazards, essentially extending the basic Object Event Detection Response (OEDR) framework [30] depicted in **Figure 9a** with hazard identification. As such, object tracking and fusion across multiple sensors are intermediate steps in a dynamic hazard identification and TTM estimation pipeline as depicted in **Figure 9b**, which could be implemented, e.g. using a Deep Neural Network (DNN). Similarly, the SOTIF temporal view (**Figure 3** of [1]) can be extended by adding the hazard analysis steps to the scene interpretation logic, as depicted in **Figure 10**.

Calibration against test track data can be performed by comparing TTM estimates against ground truth, and estimating the distribution parameters using regression, as depicted in **Figure 9c**. Subsequently, the simulation of **Algorithm 1** samples that distribution when injecting errors, and provides a uniform streamlined impact analysis for all hazard types.

1) **Modeling Limitations:** We extend well established methods for modeling system failure to modeling of TTM errors for dynamic hazards. Some well established approaches model the dependencies and data paths along the driving loop pipeline, and formalize the input specification for each component. Subsequently, analyzing deviations from those specification can be performed using Failure Mode Effects and Criticality Analysis (FMECA) per Automotive Industry Action Group (AIAG) FMEA-4 [6], fault tree analysis (FTA), and others. We propose to extend these methods by defining the system as *comprising of all traffic participants within a scenario*, and perform FMECA and FTA by *modeling all traffic participants within a scenario*. This includes modeling potential behaviors of all traffic participants.

Key Point 8: FMECA and FTA analysis should model all traffic participants for the applicable scenario.

Example 4: As an example, consider the ADAS cruise control use-case whereby the system is tasked with maintaining a safe distance from a lead vehicle. The system being modeled comprises of the Ego vehicle plus the lead vehicle being followed. Cruise control can be very effective using measurement of the TTC, which can be achieved by measurement of the relative distance and velocity between the

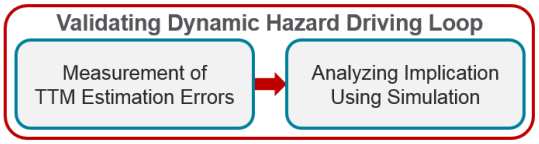


Fig. 8: Validating Dynamic Environment Hazard Driving.

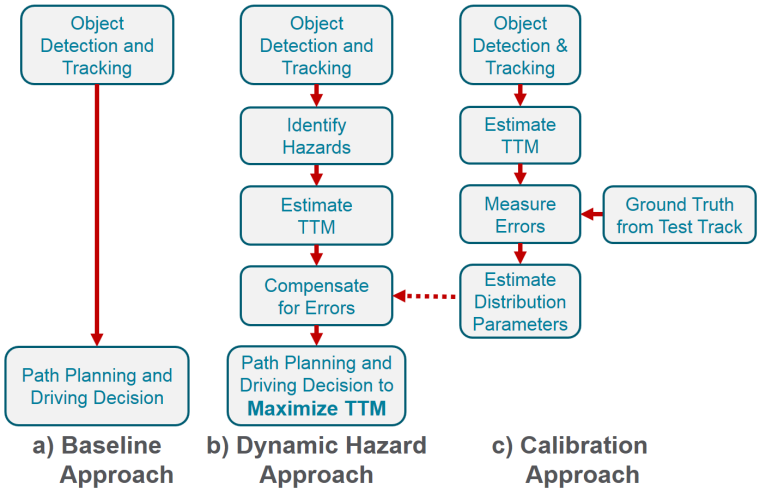


Fig. 9: Extending OEDR with Hazard Analysis.

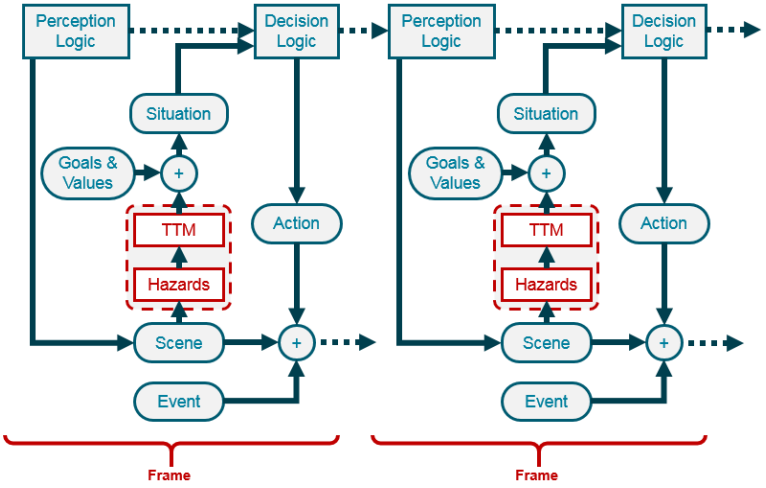


Fig. 10: Extending SOTIF Temporal View with Hazards.

Ego and the lead vehicle. To understand the capabilities and limitations of the sensor, one can model the *measured* relative distance $\Delta X_{measured}$ as the sum of the ground truth ΔX and an error ϵ_x, as depicted in **Figure 11**. Similarly, the *measured* relative velocity is the sum of the ground truth ΔV and an error ϵ_v. The *measured* TTC is:

$$TTC_{measured} = \frac{\Delta X + \varepsilon_x}{\Delta V + \varepsilon_v} = TTC_{ground\_truth} + \varepsilon_{TTC} \quad (7)$$

The advantage of this model is that the error ϵ, and its distribution, can be *measured against the ground truth* obtained from physical test readings, as well as using simulated synthetic scene images. We conducted such an experiment by training a Deep Neural Network (DNN) estimator to measure the distance to the lead vehicle using simulation images. We trained the estimator (i.e. regressor) on sunny day images,

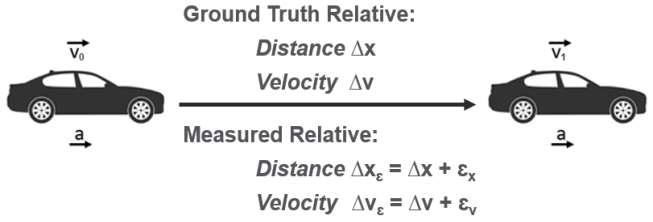


Fig. 11: Simple ADAS TTC Error Model.

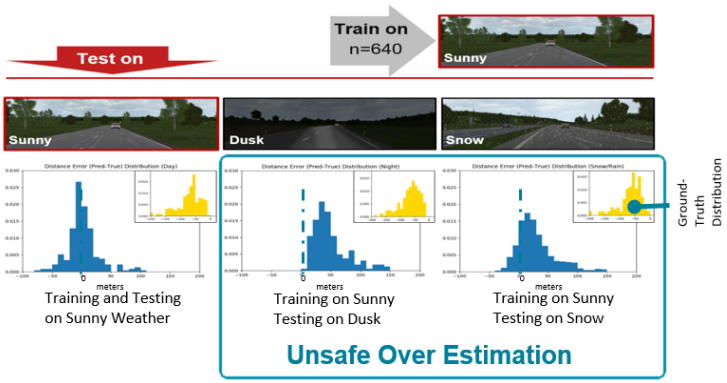


Fig. 12: Simple ADAS Error Measurement.

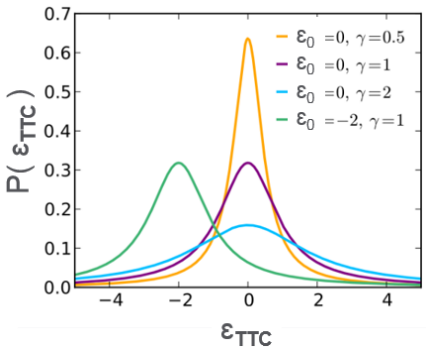


Fig. 13: The Cauchy TTC Error Distribution [48].

and measured the error on sunny day images, as well as on dusk and snow images. The results, depicted in **Figure 12**, show that *the DNN estimator error is normally distributed*. The consequence is that the TTC error is distributed according to the ratio of two normal distribution, giving rise to the Cauchy distribution, parameterized by γ, as depicted in **Figure 13**:

$$P(\varepsilon_{TTC}) = \frac{1}{\pi\gamma\left[1 + \frac{(\varepsilon_{TTC} - \varepsilon_0)^2}{\gamma}\right]} \tag{8}$$

Most importantly, we observe that the estimation error ε_x for weather dusk and snowy images is biased, with $\varepsilon_0 > 5m$. This implies *unsafe overestimation* because a TTC-based crash-avoidance braking will be triggered later than would otherwise be triggered without such errors.

Key Point 9: Validation of the dynamic hazard driving loop should focus on measurement of TTM estimation error distribution against ground truth, and the subsequent evaluation of the impact of such error distribution using simulation, as depicted in **Figure 8**.

2) **Modeling Misuse:** Only unintentional misuse is in scope for safety standards; intentional misuse is regarded as a security issue. The unintentional misuse can be classified into two main categories:

- Usage outside the Operational Design Domain (ODD) of the vehicle. Avoiding such misuse requires automatically detecting situations outside the ODD, and communicating such a detection to the operator.
- Inappropriate response to temporary restrictions, e.g., per authorities, oil spills due to accidents or other unforeseen events. Avoiding such misuse requires modeling unplanned restrictions and communicating to the operator.

In all cases, communication of an imminent misuse must allow the operator reasonable time and options to respond and avoid further misuse. Often this requires priming drivers for some time, e.g. 20 seconds [46]. Once appropriate communication takes place, any subsequent misuse becomes intentional and out of scope for safety standards. Note that analysis of sneak labeling and sneak instructions described in Section III below is applicable in this context.

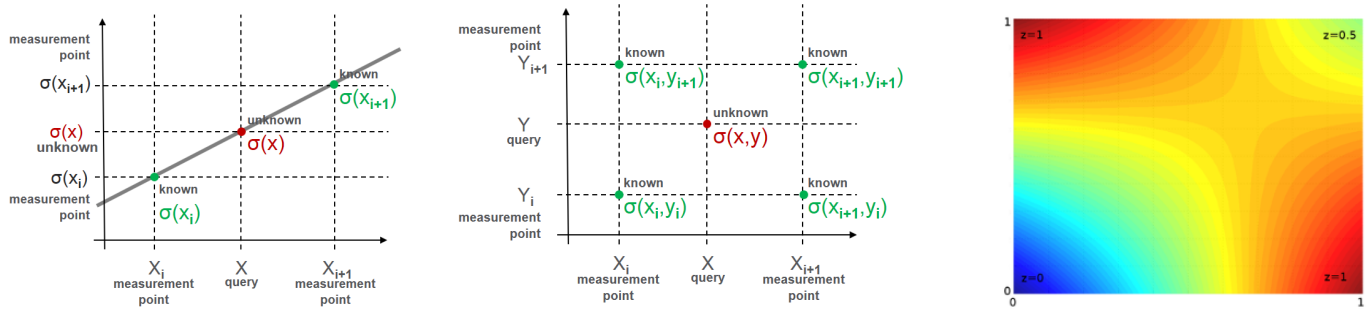
Key Point 10: Formulate the requirement for ODD detection, handling temporary restrictions and the implied communications to prevent unintentional misuse.

The misuse hazard is associated with TTM which can be estimated within the dynamic hazard driving loop. As an example, a TTM of 20 seconds is associated with the first handover notification to the human driver [46]. Much as with estimation of TTC, the error in such estimation can be measured by comparing to a ground truth. Such a measurement should quantify the distribution of errors $P(\varepsilon_{TTM})$, enabling the impact evaluation in simulation to obtain relative frequency and confidence for various outcomes.

Key Point 11: Validation of the risk of misuse should focus on measurement of TTM estimation error against ground truth, and the subsequent evaluation of the impact of such error using simulation, as depicted in **Figure 8**.

3) **Environment Hazard Modeling:** To simplify and unify the modeling of the wide range of environment hazards, we measure the TTM error for various location on the road and propagate the implications using simulations. This approach decouples the challenge of measuring TTM error distribution from the validation task, as depicted in **Figure 8**. For modeling error and limitations due to road conditions, the measurement is performed by placing the sensors at various locations on the road, recording its relative distance and velocity estimates, and applying parametric regression to estimate the mean and variance of the TTM error distribution.

Example 5: Consider, for example, modeling ε_{TTC} for a two-way road upon entering a tunnel. A depiction of the TTC scale model for a tunnel is provided in **Figure 15**. The light orange circles represent the confidence intervals of TTC estimation as expected to occur at the center of the circle. Prior to entering the tunnel, the TTC estimation is very



(a) Single variable linear interpolation. (b) Grid view for bi-linear interpolation. (c) Heat map view of bi-linear interpolation.

Fig. 14: Linear interpolation.

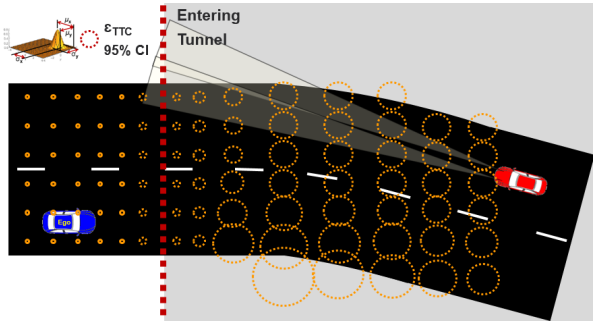


Fig. 15: Simple ADAS TTC Tunnel Error Model.

accurate, depicted by small radius circles. Subsequently, upon entry, the lighting condition change, resulting in an increased TTC estimation error rate. The most significant error rates occur when the trajectory of the two vehicles are at their largest angle. The loss of GPS signal within the tunnel results in compounded hazards, modeled by increased circle size at the Ego moves deeper into the tunnel.

Key Point 12: There is a need to train perception algorithms to estimate TTM. Subsequently, there is a need to measure the *location-dependent* distribution of estimation errors against ground truth obtain from physical test track testing.

E. Modeling Bias and Variance

The TTM error estimation approach requires measurement of both mean (=bias) and confidence interval (=scale) of the estimation error for various locations on the road (across all hazard types). Whereas it is not practical to measure and specify the confidence intervals for all possible locations on the road, it is reasonable to take measurements for specific grid points. The visualization of the TTM error distribution in **Figure 15** uses circle centers to depict the point of measurement, arrows originating at those centers to visualize error mean (=bias) and the radius to depict $2\times$ the error standard deviation (half of confidence interval) measured. Subsequently, during simulation, estimating the TTM confidence interval for a specific location x,y is achieve by linear interpolation against the nearest grid points specified.

How Interpolation Works: For a single dimension, require that measurements of the standard deviation are available at points $(x_1, \sigma(x_1)), ..., (x_n, \sigma(x_n))$, as depicted in **Figure 14a**. Estimating the value of $\sigma(x)$ for an arbitrary point x, requires two steps:

- **Step 1:** Find i such that $x_i < x < x_{i+1}$.
- **Step 2:** Compute $\sigma(x)$ according to:

$$\frac{\sigma(x) - \sigma(x_i)}{x - x_i} = \frac{\sigma(x_{i+1}) - \sigma(x_i)}{x_{i+1} - x_i} \tag{9}$$

Solving for $\sigma(x)$ we get

$$\sigma(x) = \sigma(x_i)(1-m) + \sigma(x_{i+1})m \quad where \quad m = \frac{x - x_i}{x_{i+1} - x_i} \tag{10}$$

The same pattern applies for estimating the average $\mu(x)$. Generalizing this approach to >2-dimensions gives rise to the bi-linear and tri-linear interpolation [47], which enables estimating the mean and confidence interval at the unknown point (x, y), given measured data at grid points $(x_1, y_1, \sigma(x_1, y_1)), ..., (x_n, y_n, \sigma(x_n, y_n))$, as depicted in **Figure 14b**. A full bi-linear interpolation for points between the measured grid points is depicted in **Figure 14c**.

Key Point 13: Use simple parametric distribution model of TTM estimation error between measured locations.

Using a simplistic parametric model of mean and variance of error distribution, enables the development of a surrogate model which quantifies the acceptable error rates.

F. Compound Hazard Modeling

Seldom does a single hazard cause an accident or a functional failure. Consequently, it is critical to model the impact when multiple compound hazards are materialized. The compound effect will materialize once all compounding hazards materialize, thus the TTM for the compound effect is the maximum TTM of all the compounding hazards.

Example 6: Consider the two hazards of (1) GPS loss in a tunnel and (2) a slippery road section. The TTM of materialization of the compound hazard is the largest TTM of each of GPS loss and slippery road individually. Authorities can place alerts for slippery road sections within a tunnel.

Key Point 14: Require modeling of compounding hazards.

III. Sneak Glitches Without Failure

Sneak Events are unplanned "emergent" latent behaviors, which were originally discovered through the Apollo program in 1967, after the Apollo 1 capsule fire. According to the *Apollo Sneak Circuit Analysis Handbook* [32], "Sneak circuits are commonly known as system glitches or anomalies which are not contingent on component failures. It has been found that such sneak circuits have distinct classifiable characteristics which make engineering analysis feasible."

The lessons learned from the capsule fire triggered a systematic study of sneak conditions which had actually occurred. The studies included a missile accidentally launched from a B-52 bomber while parked on the ground, bombs accidentally released from a B-52, a nuclear bomb inadvertently armed, a business jet which lost electrical power and drained its batteries while in flight without any indication to the crew, and an electric utility lineman electrocuted while working on a power line to which the power was "off".

All of these events occurred without the failure of any components. As an example, consider the case of mysterious activation of fire alarms triggered by activation of the sprinkler system through water pressure sensors. The sneak event is caused by flushing a sufficient number of toilets which would cause the same pressure drop required to trigger the alarm.

During the 1990s, Martyn Thomas famously reported an automotive sneak event [34] (Chaprter 2, Proposed Approach, Explicit Dependability Claims): "While the radio was switched on, when the brake pedal was depressed at the same time as a rear window was being operated, the air bags deployed."

More recently, in April 2016, a near miss event occurred with the *driverless* Bombardier Sao Paulo monorail, when a train departed with its doors open [35]. Operational staff intervened and stopped the train with the emergency stop button. The Sneak Timing Analysis discovered that the Automatic Train Control (ATC) system sent an errant door open command in an arbitrary manner which caused doors to open.

Today we know that sneak events are universal [36]. The AIAA S-102.2.5-2019 Standard [37] requires tracing all paths that affect an outcome, *assuming that all components are operating properly*. A sneak event is characterized as follows:

- It is caused by introducing an unintended path e.g. wiring, tubing, software interfaces, operator actions, instrumentation, mechanical interlocks, or some combination thereof, or condition e.g. timing incompatibility.
- It leads to unintended system effects ranging from an intermittent nuisance to complete system loss or a fatality.

The AIAA Standard covers four types of *sneak conditions*:

- **Sneak Paths:** Unexpected paths along which current, energy, or logical sequence flows in an unintended direction.
- **Sneak Timing:** Events occurring in an unexpected or conflicting sequence.
- **Sneak Indications:** Ambiguous or false displays of system operating conditions that may cause the system or operator to take an undesired action.
- **Sneak Labels:** Incorrect labeling of system functions (e.g., system inputs, controls, displays, and buses) that may cause an operator error.

The overall maturity levels specified by the AIAA Sneak Circuit Standard are provided in **Table I**. To achieve high dependability, the highest maturity levels require quantifying the probability of sneak circuit condition occurrence with confidence bounds. Regardless of whether the analysis is performed using manual, semi-automated, or automated methods, a prerequisite is the collection, processing, and evaluation of detailed system design information. The analysis should be applied on the entire system rather than on a pre-selected subset of components. Such pre-selection usually partitions the system along the same boundaries that hide sneak conditions from the system designers. Excluded components may have critical sneak effects.

Maturity	Requirement
Level 1	Identify Hardware Sneak Paths and Timing
Level 2	Identify Software Sneak Paths and Timing, Specify Severity Classification Disposition for each: Critical for Safety, Mission, Reliability, Maintenance, Monitoring
Level 3	Specify Key Indicators, then Identify Specific Condition, then Apply **FMECA/Hazards Analysis** for each condition.
Level 4	Quantify **Probability of Condition Occurrence** with Confidence Bounds.

TABLE I: SCA AIAA S-102.2.5-2019 National Standard

Key Point 15: Include AIAA Sneak Circuit Standard [37] as part of SOTIF recommended practices.

Traditional sneak condition analysis decomposes the system using functional operators and assembles a list of topographs to be matched against a list of "clue topographs". For each match, an exhaustive list of combinations is tested in an attempt to identify an undesired outcome.

Such analysis is not applicable to systems leveraging AI components for which topographs cannot be extracted. These methods must be extended to replace topographs clues with skewing of $P(scenario)$ to achieve hazard and residual risk distribution mode discovery for $P(crash \mid scenario)$. Specifically, there is need to determine how these modes are changing or shifting across evolution of autonomous driver versions.

Key Point 16: Extend Sneak Condition Analysis to support systems with AI components.

IV. Scenarios

The key engineering demands addressed by scenarios are:

- **Transfer Learning from Simulation to Physical:** Bridging the "reality gap" that separates simulated robotics from experiments on hardware could be achieved using Domain Randomization (DR), which is applied to AV engineering as depicted in **Figure 16**. Models that transfer to reality are trained on simulated images by randomizing rendering in the simulator. With sufficient variability in the simulator, the real world may appear to the model as just another variation. The approach was applied to deliver the first successful transfer to the real world of a control DNN trained only on simulated RGB images, without pre-training on real images [53]. Subsequently, using only data from a simulator with non-realistic random textures, without pre-training on real images, DR has successfully trained a real-world object detector that is accurate to 1.5cm and robust to distractors and partial occlusions [54].
- **Interpretation of Autonomous Driver:** Components based on DNNs are opaque black-boxes having an incomprehensible decision logic. Interpretability can be made practical by analyzing a single *logical* scenario at a time. The general interpretability paradigm uses a comprehensible surrogate model, such as a Generalized Additive Model (GAM) or a Decision Tree, to approximate the behavior of the opaque

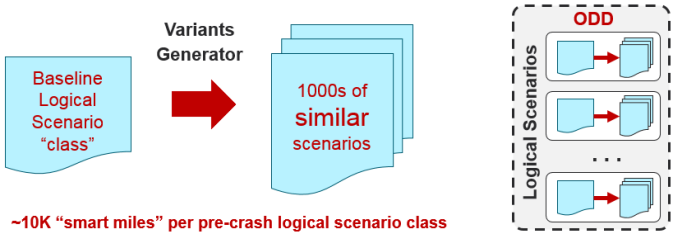


Fig. 16: ODD representation using population of scenarios.

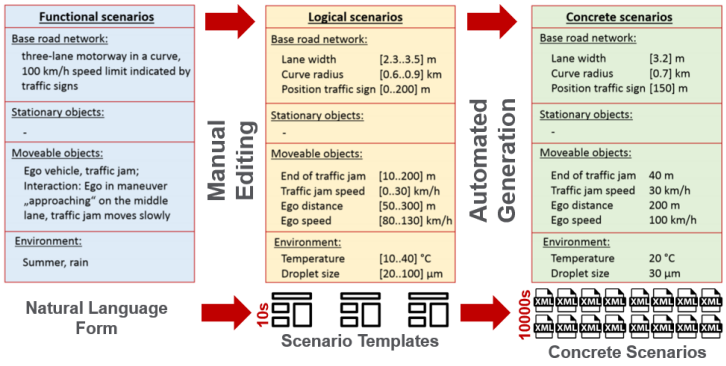


Fig. 17: Scenario Specifications [18].

DNN using carefully constructed variations of specific examples [49], [50], [51], [52]. This requires sampling from a purpose built population of test examples. The approach is applied to simulation models by sampling from a purpose-built population of scenarios, as depicted in **Figure 16**.

- **Decomposition of the Safety Problem:** Achieving *safe in use* requires that the vehicle engineering extends beyond explicit specifications to cover *foreseeable use*. The *functional safety discipline* addresses this need with *functional performance analysis*, which is a process starting from unknown limitations and systematic failures, and culminating with their detailed quantification. The *functional performance analysis* is made practical by operating on a single *logical scenario* at a time. To achieve statistical confidence, each baseline scenario is expanded into a population of similar scenarios, as depicted in **Figure 16**.
- **Instilling Trust:** In order to effectively interact with or supervise a robot, humans need to have an accurate mental model of its capabilities and how it acts. For most tasks, the essence of a control logic can be captured in a few critical states in which action selection, e.g. maximizing TTM, is critical. User studies show that analysis of such critical states increases trust [55]. The critical states can be uncovered by sampling scenarios from a purpose built distribution [55], as depicted in **Figure 16**.

Key Point 17: Represent the ODD using population of scenarios purpose built for the engineering task at hand, including training, testing, interpretation and safety validation.

Indeed, all major AV developers leverage scenarios, and all safety standards require establishment of a requirement specification process, in which scenarios are first class citizens. As such, the Pegasus approach [18] defines the following levels of scenario specifications depicted in **Figure 17**:

- **Functional Scenarios:** This is an informal specification of the scenario, using ambiguous natural language. This could be, for example, "Ego backing up into another vehicles", or "Another vehicle backing up into Ego", or "Following vehicle making maneuver". The informal specification may include an informal specification of the ODD, such as "on the freeway" or "within an intersection". This specification is regarded as informal.
- **Logical Scenarios:** Once an ambiguous functional scenario specification is available, there is a need to identify the specific variations which are included. There is a need to specify whether the road is curved and the curve radius. For the pedestrian scenarios, there is a need to further describe the *range* of relative distances of pedestrians from each of the two vehicles. Other parameters may include velocity ranges for the two vehicles, pedestrian walking direction, visibility conditions and road traction, etc. A scenario with a fully specified list of parameter *ranges* is regarded as a *logical scenario*. This is regarded the *semi-formal* specification of the scenario. It is *not* formal because it does not specify all detailed requirements, such as the shape of the parking structure and road markings.
- **Concrete Scenarios:** Once a logical scenario is specified, the scenario specification needs to be made formal. This includes specification of, e.g. the shape of parking structures, road elevation maps and profile of speed bumps, exact geometries of e.g., lanes and parking spots, occluding objects such as trash cans and walls, road markings such as crosswalks and manually placed indicators by authorities, and numerous other details not included in the logical semi-formal specification. Each concrete scenario specification must include all the information making it directly executable in a simulator. This is regarded as the *formal* specification of the scenario.

Example 7: As an example, the NHTSA specifies about 100 distinct pre-crash functional scenarios [15]. With the Pegasus approach, each such functional scenario may be associated with thousands of logical scenarios. Each logical scenario may be associated with thousands of concrete scenarios. In total, there would be about 10^8 concrete scenarios representing all of the funcitonal NHTSA pre-crash scenarios.

The list of NHTSA pre-crash scenarios is not representing all possible scenarios for which a an autonomous driver is to be tested. As an example, the NHTSA pre-crash scenario list excludes parking scenarios, work zones, and side-of-cliff driving, and many other important scenarios.

Key Point 18: The scenario requirements process must start with informal functional scenarios, proceed with specification of the population of scenarios representing the ODD, and commence with a library of concrete formal scenarios which can execute on a simulator and with which specific tests are conducted.

A. Hazard Materialization

Given a library of scenarios $\mathcal{L}$, we can bound the residual risk associated with a hazard using

$$R_h \leq P(h) = \sum_{\forall s \in \mathcal{L}} P(h|s)P(s) \tag{11}$$

This approach is attractive because it decomposes the task of quantifying the distribution $P(h)$ by performing millions of independent tasks, one for each *concrete* scenario, and combining the results using a post-processing aggregation step.

B. Scenario Specification Process

Building effective scenario libraries is important for development, but *critical* for validation. The effectiveness of the library stems from its coverage: the goal should be to *develop the smallest library achieving adequate coverage*. As discussed above, estimating $P(crash \mid scenario)$ and the coverage can be achieved determining the prior $P(scenario)$ through extraction of scenario characteristics from vehicle instrumentation and data collected by telematics companies.

The general approach for assembling a library with quantifiable coverage is depicted in **Figure 18**. The starting point is a baseline scenario. The 3D environment, e.g., road geometry and "furniture", can be based on specific real world environments scanned by HD mapping and surveying technologies [43]. For a given 3D environment, the traffic participants can be added to implement various logical scenarios, e.g., with a single vehicle in front, or multiple vehicles both in front, in other lanes, in opposite directions, occupying intersections and other possible situations.

Once the baseline scenario is completed, it can be parameterized to represent the ODD, as depicted in the 2nd steps of **Figure 18**. Such parameterization enables sampling from the parameter distribution provided in logical scenarios to specify, e.g. weather and visibility conditions, position of static "road furniture" such as occluding bus stops and trees, traffic lights, indicators and road marks, initial positions and velocities of traffic participants, etc. Each parameter is associated with a distribution which can be measured using telematics data.

Subsequently, once a template is available, a large number of variations can be generated to represent a population of similar but sufficiently different scenarios which enables analysis of boundaries between safe and unsafe driver behavior. Each of those variants is executed, e.g., using a simulator such as Virtual Test Drive (VTD) [42]. The resulting logs can be stored in a data format which enables schema discovery, e.g. dense HDF5 format, to facilitate extraction and subsequent ingestion into large databases.

Finally, data analytics is used for determining the coverage of the ODD, as well as the effectiveness of safety boundary coverage. Additional scenarios may need to be added where boundaries are not identified.

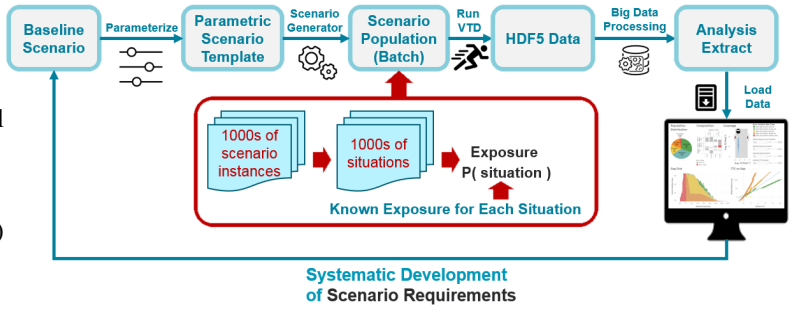


Fig. 18: Scenario Specification Process.

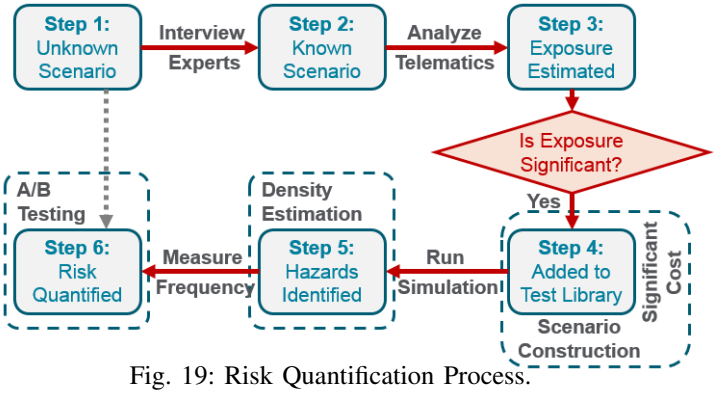


Fig. 19: Risk Quantification Process.

C. Risk Quantification Process

The contribution of a scenario s to the residual risk for a hazard h is smaller than the prior probability $P(s)$ of that scenario. Excluding from the test library a collection of scenarios S such that $\varepsilon > \sum_{\forall s \in S} P(S)$ results in an error for estimating R_h of less than ε, namely

$$R_h + \varepsilon \geq \sum_{\forall s \in (\mathcal{L}-S)} P(h|s)P(s) \tag{12}$$

The implied risk quantification process is depicted in **Figure 19**. Initially, functional scenarios are unknown. As scenarios are identified, in step 2, they are added to a list of known functional scenarios. The exposure of each known scenario is subsequently analyzed, in step 3, e.g. using *telematics data*. If the exposure identified is significant, then step 4 is executed, the scenario is built and added to the test library; *the cost of this step is significant*. Subsequently, the probability density of unsafe situation is estimated in step 5. Finally, the newly added scenario, together with the other scenarios, are used for A/B comparison of driver versions.

Key Point 19: Best practices are needed to define a process for identifying scenarios and assembling a library supporting A/B comparison.

D. Constrained ODD Representation

The ODD is a description of the specific operating conditions in which the automated driving system is designed to properly operate, including roadway types, speed range, environmental conditions (e.g. weather, daytime/nighttime), prevailing traffic laws, and other domain constraints [45].

The logical scenario specification defines populations of concrete scenarios. Define $V_1, ..., V_n$ as the parameters controlling that population, and denote $v_1, ..., v_n$ as an assignment

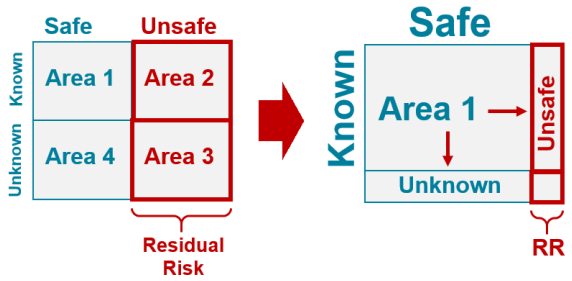


Fig. 20: SOTIF Areas [1].

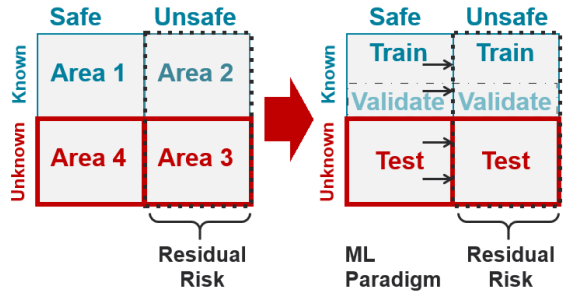


Fig. 21: SOTIF ML partitioning

of values to all these variables. We define the ODD by adding a set of constraints $C_1, ..., C_q$ mapping a variable value assignment to a true/false, namely $C_i : (v_1, ..., v_n) \mapsto \{0, 1\}$. We *exclude* from the ODD scenarios which do not satisfy these constraints, namely those for which $C_i : (v_1, ..., v_n) \mapsto 0$.

Example 8: Consider the scenario "following vehicle making maneuver" with the initial velocities of Ego and the lead vehicle are V_1, V_2, so that the ODD population corresponds to a bi-variate distribution $P(V_1, V_2)$. Define, for example:

$$\begin{aligned} C_1 &\equiv V_1 \in [3, 35] & C_2 &\equiv V_1 - V2 \in [0, 5] \\ C_3 &\equiv V_1 \in [3, 70] & C_4 &\equiv V_1 - V2 \in [0, 20] \end{aligned} \tag{13}$$

Initially, the development can focus on satisfying all four constraints C_1, C_2, C_3, C_4, namely restrict Ego velocities to be above 3mph but under 35mph, and relative velocity of Ego approaching the lead vehicle at less than 5mph. Subsequently, the development can proceed with removing C_1 allowing initial velocity of up to 70mph. Subsequently, proceed with removing of C_2 increasing the scope to include situations that Ego approaches the lead vehicle at a rate of up to 20mph.

Key Point 20: Representing ODDs using constrained scenario population enables incremental expansion by gradual constraint removal.

E. Working with Unknown Scenarios

The well understood machine learning paradigm can be applied to enable modern engineering to effectively develop, and continuously improve, against unknown scenario. Starting from the SOTIF area decomposition depicted in **Figure 20**, the ODD population is sampled to obtain a library of development scenarios. This sample, regarded as the set of known scenarios, can be partitioned into training and validation subsets, as depicted in **Figure 21**. Such partitioning enables estimating the statistical performance of the AV driver against unknown scenarios, much as the performance of machine learning models is estimated during training using validation

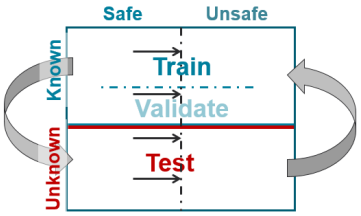


Fig. 22: Applying ML Paradigm to SOTIF.

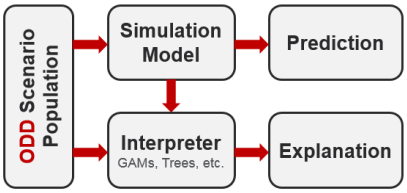


Fig. 23: Applying ML Interpretablity Paradigm

data [56]. The implied development process, depicted in **Figure 22**, trains models using a subset of the ODD scenarios, and estimates their performance using a disjoint subset.

Subsequently, the AV behavior can be interpreted using the architecture depicted in **Figure 23**. An interpreter model, e.g. GAM or Tree, can be trained using simulation results obtained from additional scenarios sampled from the ODD population. Interpretations are obtained using the partial dependency analysis methods described in [49], [50], [51].

F. Impact of Edge Cases

In contrast to corner cases, which are typically referred to as *pathological input outside the intended scope*, edge cases are *input within scope*. For example, when processing a stream of video frames, it is common to see the object detection bounding boxes "flicker" due to changes of input across frames. Frames for which the bounding boxes disappear (only to appear back in subsequent frames) are failures *within* scope, and thus regarded as *edge* cases. The "flickering" is addressed by object tracking algorithms, which fuse data from multiple sensors, aggregate and interpolate detection across frames. As such, frames for which object detection disappears can be regarded as an example of *failure for input in scope*, and tracking can be viewed as *a fault tolerance algorithm*.

This perspective acknowledges that, whereas *frame-level failure of individual components is very common*, the overall system must be robust to such failures. Considering that a component may have a failure rate of $> 10^{-2}$, the overall system failure rate for naturalistic driving miles must be $< 10^{-12}$ frames (=10^8 miles $\times$ >1000 frames per mile across numerous sensors).

Key Point 21: There is a need to quantify the rate at which fusion reduces failure rates of individual components.

The simplified failure model is depicted in **Figure 24a** describes hazardous situations occurring when the overall system fails due to sufficient number of failures of individuals sensors at specific frames, but *only when the fusion (fault tolerance) algorithm fails as well*. When such hazardous situations occur, they are transient, depicted as sparse temporal bars in **Figure 24b**. The residual risk increases with the duration of hazardous situation persistence.

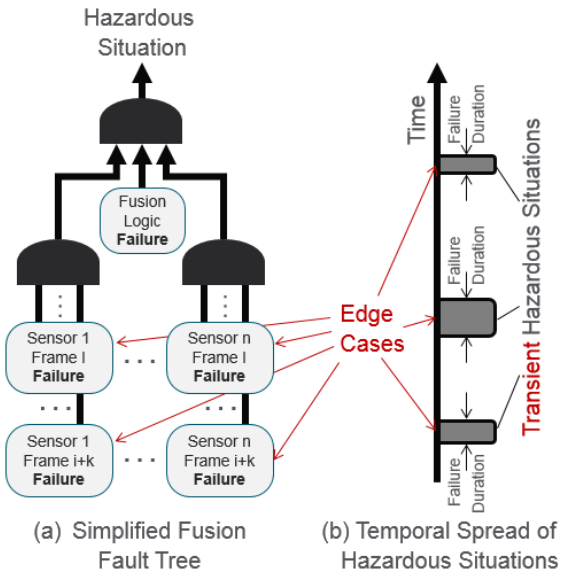


Fig. 24: Impact of Edge Cases.

Key Point 22: There is a need to quantify the persistence duration of hazardous situations to control residual risk.

G. SOTIF Area Analysis

The SOTIF specification divides scenarios into safe and unsafe, known and unknown, as depicted in **Figure 20**. Whereas there is limited control on outcomes of individual scenarios, engineering processes must exert control over the continuous improvement of performance statistics. As such, compliance to SOTIF requires continuously increasing Area 1 at the expense of Areas 2,3,4, also depicted in **Figure 20**.

SOTIF Areas 1,2: All scenarios for which tests are run are regarded as known. Scenarios for which the probability of $P(h)$ is too high are regarded as unsafe; all other scenarios are regarded as safe. The *known-safe-ratio*, which is size of Area 1 as compared to Area 2, is the ratio between the safe and unsafe scenarios, namely:

$$R_{known\_safe\_unsafe} = \frac{Area1}{Area2} \tag{14}$$

Note that, using Area 1,2 = $P(\neg h), P(h)$ respectively, the ratio $Area1/Area2$ represents $P(\neg h)/P(h) = p(\neg h|s)/p(h|s)$ because the prior $P(s)$ cancels out.

SOTIF Areas 3,4: A key challenge with compliant implementation of SOTIF is the appropriate analysis of unknown scenarios. To address this challenge, using the aforementioned machine learning paradigm, we assume that the distribution of hazard materialization for the unknown scenarios is similar to the corresponding distribution for the known scenarios. With this assumption:

$$R_{unknown\_safe\_unsafe} = \frac{Area4}{Area3} \approx \frac{Area1}{Area2} \tag{15}$$

H. Residual Risk

In the context of SOTIF, the residual risk is defined as the total prior probability of the unsafe scenarios, namely

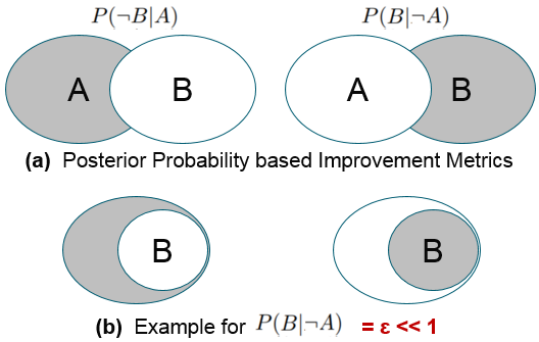


Fig. 25: Posterior Probability Based A/B Comparison.

$$\text{Residual Risk} = \frac{Area2 + Area3}{Area1 + Area2 + Area3 + Area4} \tag{16}$$

Modeling that all areas sum to 1, to obtain the formulation for residual risk, the above can be combined into the following two equations:

$$\begin{aligned} A_3 + A_4 &= 1 - A_1 - A_2 \\ \frac{A_4}{A_3} &= \frac{A_1}{A_2} \end{aligned} \tag{17}$$

This leads to the final simplified residual risk formulation:

$$\begin{aligned} \text{Residual Risk} = A_2 + A_3 &= \frac{A_1A_2 + A_2^2 + A_2 - A_1A_2 - A_2^2}{A_1 + A_2} \\ &= \frac{A_2}{A_1 + A_2} = \frac{\sum P(h|s)}{\sum P(s)} \end{aligned} \tag{18}$$

where $A_1 + A_2 = \sum(P(h|s) + P(\neg h|s)) = \sum P(s)$ is the coverage. As the safety increases, A_1 approaches 1 and A_2 approaches zero, the change of residual risk, e.g. due to engineering changes, is dominated by the derivative of A_2.

Key Point 23: It is sufficient to quantify the residual risk using $\sum P(s)$ from *telematics* and $\sum P(h|s)$ from *simulation*.

I. Measuring Progress

Measuring progress is at the heart of SOTIF. Specifically, there is a need to perform A/B comparison of distributions $\sum P_A(h|s)$ and $\sum P_B(h|s)$ for revisions A,B respectively. *The comparison must be conditional in nature and cannot rely on the priors of* $P(A)$ *and* $P(B)$.

Denote $P(\neg B|A)$ as the probability distribution that revision B did *not* experience a materialized hazard h given that revision A *did*. Denote $P(B|\neg A)$ as the probability distribution that revision B *did* experience materialization of h given that revision A did *not*. Then, as depicted in **Fig 25**:

- $P(\neg B|A) =$ improvement of B relative A
- $P(B|\neg A) =$ regression of B relative to A

Example 9: When $P(B|\neg A)$ approaches 0 then regression is almost non-existent. At that point we obtain that:

- $P(B) = P(B|A) + P(B|\neg A) \approx P(B|A)$.
- $P(A) \geq P(B)$ implying an improvement.

Key Point 24: Comparing autonomous drivers to a human or other autonomous drivers should be done using hazard materialization posterior probabilities.

V. Conclusion

We presented a probabilistic framework for incrementally bounding the residual risk associated with autonomous drivers and enables quantifying progress. The approach extends mainstream safety standards based on hazard analysis to encompass multiple vehicles. It introduces a dynamic hazard perspective according to which the autonomous driver continuously monitors for imminent hazards and selects actions which maximizes the Time To Materialization (TTM) of these hazards. We describe hazard modeling approaches which trains machine learning models to identify hazards, estimate their TTM, and enables their calibration against physical test track data. We provide a simple incremental process for acquiring scenario libraries which serve as the backbone of a continuous integration. We explain how mainstream machine learning methjods can be applied to develop against unknown scenarios and interpret the resulting driving logic. We show that the approach enables implementing the continuous expansion of SOTIF Area 1 through measurement of improvements vs regressions using conditionals. This approach can measure the relative safety of autonomous drivers as compared to human drivers.

References

[1] Road vehicles – Safety of the intended functionality. ISO/PAS 21448:2019.

[2] Road vehicles – Functional safety. ISO 26262:2018.

[3] IEC Functional Safety Standards, IEC 61508 Edition 2.0

[4] IEC 61025 ED. 2.0 (2006): FAULT TREE ANALYSIS (FTA)

[5] Procedures for performing a failure mode effect and critical analysis. Department of Defense (US). MIL-P-1629.

[6] Potential Failure Mode Effects Analysis. Automotive Industry Action Group (AIAG).

[7] Removing Regulatory Barriers for Vehicles with Automated Driving Systems. Advanced Notice of Proposed Rule Making (ANPRM). NHTSA-2019-0036 RIN 2127-AM00.

[8] NHTSA Federal Motor Vehicle Safety Standards (FMVSS)

[9] US DOT Federal Highway Administration. Surrogate Safety Assessment Model (SSAM) Final Report. FHWA-HRT-08-051

[10] National Motor Vehicle Crash Causation Survey (NMVCCS).

[11] IIHS HLDI Fatality Facts 2017 State by state. https://www.iihs.org/topics/fatality-statistics/detail/state-by-state

[12] K. Campbell. The SHRP 2 Naturalistic Driving Study: Addressing Driver Performance and Behavior in Traffic Safety, TR NEWS 282 SEPTEMBEROCTOBER 2012

[13] What it Was Like to Drive 100 Years Ago. Edgar Snyder Assoc. 2015.

[14] Edwin Olson. The Moores Law for Self-Driving Vehicles. Medium blog

[15] US Department of Transportation, National Highway Traffic Safety Administration. Pre-Crash Scenario Typology for Crash Avoidance Research. DOT HS 810 767 (2007).

[16] T.Harmon, G.Bahar, F.Gross . Crash Costs for Highway Safety Analysis. FHWA Report FHWA-SA-17-071.

[17] US Department of Transportation, National Highway Traffic Safety Administration. Description of Light-Vehicle Pre-Crash Scenarios for Safety Applications Based On Vehicle-to-Vehicle Communications. DOT HS 810 767 (2013).

[18] The PEGASUS Method https://www.pegasusprojekt.de/files/tmpl/Pegasus-Abschlussveranstaltung/PEGASUS-Gesamtmethode.pdf

[19] Hardi Hungar, Frank Kster, Jens Mazzega: Highway Pilot Test Specifications for Highly Automated Driving Functions: Institute of Transportation Systems, German Aerospace Center (DLR)

[20] Cox, R.T. (1946). Probability, Frequency, and Reasonable Expectation. American Journal of Physics. 14 (1): 110.

[21] J. Pearl. Causality: Models, Reasoning, and Inference, Cambridge University Press, 2000

[22] R.Dechter. Bucket elimination: A unifying framework for probabilistic inference. Artificial Intelligence. Volume 113, Issues 12, September 1999, Pages 41-85

[23] Standard: Performance-Based Sneak Circuit Analysis (SCA) Requirements (AIAA S-102.2.5-2019)

[24] James Li. Sneak Circuit Analysis: Lessons Learned from Near Miss Event. International Journal of Mathematical, Engineering and Management Sciences Vol. 2, No. 1, 3036, 2017

[25] Clifton A. Ericson. Hazard Analysis Techniques for Safety. Wiley Interscience 2005, ISBN 0-471-72019-4.

[26] Laura Fraade-Blanar, Marjory S. Blumenthal, James M. Anderson, N.Kalra, Measuring Automated Vehicle Safety. RAND Corp, 2018.

[27] N. Kalra: Challenges and Approaches to Realizing Autonomous Vehicle Safety, Self-Driving Cars: Road to Deployment. Subcommittee on Digital Commerce and Consumer Protection, Committee on Energy and Commerce, United States House of Representatives, 115th Congress, First Session, Feb 2017.

[28] Why Waiting for Perfect Autonomous Vehicles May Cost Lives. RAND Blog 11/7/2017.

[29] IIHS Status Report newsletter, Vol. 52, No. 3, May 25, 2017

[30] A Framework for Automated Driving System Testable Cases and Scenarios. DOT HS 812 623.

[31] Office of Highway Policy Information. Table VM-1 - Highway Statistics 2016: Annual Vehicle Distance Traveled in Miles and Related Data, by Highway Category and Vehicle Type.

[32] Sneak Circuit Analysis Handbook: Apollo Model. Document D2-118341-1. The Boeing Company Contract NAS9-10364.

[33] S. Shalev-Shwartz, S. Shammah, A. Shashua. On a Formal Model of Safe and Scalable Self-driving Cars. Mobileye, 2017

[34] D.Jackson, M.Thomas, and L.I.Millett, Editors: Software for Dependable Systems: Sufficient Evidence? UK National Research Council; Engineering and Physical Sciences; Computer Science and Telecommunications Board; Committee on Certifiably Dependable Software Systems (2007)

[35] J.Li. Sneak Circuit Analysis: Lessons Learned from Near Miss Event International Journal of Mathematical, Engineering and Management Sciences Vol. 2, No. 1, 3036, 2017

[36] James L. Vogas. Sneak Analysis of Process Control Systems. Boeing Aerospace Operations, Inc. (2006)

[37] Standard: Performance-Based Sneak Circuit Analysis (SCA) Requirements (AIAA S-102.2.5-2019)

[38] Robbie Miller. Closing the Curtains on Safety Theater. Medium Blog

[39] V.Spears. GMs Driverless Fleet Should Be Held to High Safety Bar NHTSA Is Told. Insurance Coverage Law Center

[40] K.G. Larsen, B.Thomsen. Partial specifications and compositional verification. Theoretical Computer Science 88 (1991) 15-32

[41] Demonstration of Tesla 2-car ahead detection. https://www.youtube.com/watch?v=6yaUj7yVtaQ&feature=youtu.be&t=99

[42] Virtual Test Drive. MSC Software.

[43] Leica Geosystems https://leica-geosystems.com/en-us

[44] Smartdrive www.smartdrive.net

[45] Waymo Safety Report, Waymo.

[46] R.M.A. van der Heiden, S.T. Iqbal, C.P. Janssen. Priming Drivers before Handover in Semi-Autonomous Cars. CHI 2017

[47] Numerical Recipes in C, 198892 Cambridge University Press, SBN 0-521-43108-5, pp. 123128.

[48] The Cauchy distribution. Wikipedia

[49] M.T.Ribeiro, S.Singh, C.Guestrin. Why should I trust you? Explaining the predictions of any classifier. 22nd ACM SIGKDD (2016)

[50] A.Shrikumar, P.Greenside, A.Kundaje. Learning Important Features Through Propagating Activation Differences. 22nd ACM SIGKDD (2016), arXiv:1605.01713v3

[51] S. M.Lundberg, G.G.Erion, SI.Lee. Consistent Individualized Feature Attribution for Tree Ensembles. ICML 2017 arXiv:1706.06060

[52] M.Ancona, E.Ceolini, C.ztireli, M.Gross, Towards better understanding of gradient-based attribution methods for Deep Neural Networks. ICLR 2018. arXiv:1711.06104v4

[53] J.Tobin, R.Fong, A.Ray, J.Schneider, W.Zaremba, P.Abbeel. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2017) arXiv:1703.06907v1

[54] A.Prakash, S.Boochoon, M.Brophy, D.Acuna, E.Cameracci, G.State, O.Shapira, S.Birchfield. Structured Domain Randomization: Bridging the Reality Gap by Context-Aware Synthetic Data. arXiv:1810.10093v1

[55] S.H.Huang, K.Bhatia, P.Abbeel, A.D.Dragan. Establishing Appropriate Trust via Critical States, arXiv:1810.08174v1.

[56] M.Kuhn, K.Johnson. Applied Predictive Modeling 1st ed. 2013, Corr. 2nd printing 2018 Edition, ISBN-13: 978-1461468486.