Handling Uncertainties of Data-Driven Models in Compliance with Safety Constraints for Autonomous Behaviour

Michael Kläs
Fraunhofer IESE
Kaiserslautern, Germany
michael.klaes@iese.fraunhofer.de

Rasmus Adler
Fraunhofer IESE
Kaiserslautern, Germany
rasmus.adler@iese.fraunhofer.de

Ioannis Sorokos
Fraunhofer IESE
Kaiserslautern, Germany
ioannis.sorokos@iese.fraunhofer.de

Lisa Joeckel
Fraunhofer IESE
Kaiserslautern, Germany
lisa.joeckel@iese.fraunhofer.de

Jan Reich
Fraunhofer IESE
Kaiserslautern, Germany
jan.reich@iese.fraunhofer.de

***Abstract*—Assuring safety is a key challenge for market introduction of many kinds of autonomous systems. This is especially true in cases where data-driven models (DDMs) such as deep neural networks are used to perceive or anticipate hazardous situations. Treating failures of such models in the same way as failures in traditional software appears insufficient, due to the less predictable nature of DDM failures. Although existing safety standards do not yet sufficiently address this finding, research widely acknowledges that residual uncertainty remaining in the outcome of DDMs is a fact that needs to be dealt with. In this context, Uncertainty Wrappers constitute a model-agnostic framework to obtain dependable and situation-aware uncertainty estimates. In contrast to many approaches, they are special in providing explainable and statistically-safeguarded uncertainty estimates. Although these properties appear beneficial, the question of integrating them into an effective risk management approach remains open. This paper intends to further stimulate and contribute to this discussion by exemplifying how dependable uncertainty estimates may become part of a dynamic risk management approach at runtime. We explore how this can be achieved in the context of the Responsibility-Sensitive Safety model.**

Keywords—machine learning, safety, uncertainty, dynamic risk management, responsibility-sensitive safety

I. MOTIVATION

Autonomous systems can achieve predefined objectives in accordance with the demands of a given operational situation without recourse to human control nor to imperative programming [1, p. 26]. A prominent example of an autonomous system is an autonomous vehicle. A key enabling technology for autonomous vehicles are Data-Driven Models (DDMs). DDMs, such as neural networks, can be trained to recognize patterns implicit in datasets, e.g. recognizing traffic signs trained from respective image sets.

The main challenge for market introduction of fully autonomous vehicles is the assurance of safety. However, autonomous vehicles are not the only example where potentially severe losses hinder market introduction. In the medical domain, autonomous surgery, autonomous infusions of analgesia, or an artificial pancreas autonomously infusing insulin are further examples where safety is still an unsolved open issue. In the production domain, the full potential of collaborative robots (cobots) and automated guided vehicles cannot be exploited due to safety issues.

An obvious approach for dealing with risks associated with safety is to constrain the autonomous behavior. For instance, a vehicle driving behind another vehicle should in general always keep a distance that is large enough so that a crash can be avoided in case of a sudden full-brake of the leading vehicle. For an autonomous vehicle, this safety objective can be formalized in a mathematical equation describing the safe distance, which depends on physical variables such as the speed of the vehicles. From this objective, a safety constraint can be established and implemented, demanding that the current distance must always be larger than the formalized safe distance. The Responsibility-Sensitive Safety (RSS) Model from Mobile Eye describes this approach in more detail [2]. The fundamental concept behind this exemplary approach is also transferrable to other applications. For instance, one could establish safety constraints for autonomous infusions considering relevant physiological variables to keep the dosage of the injection grout within safe boundaries.

DDMs outperform traditionally programmed software for specific tasks, e.g. many computer vision tasks like object classification. Considering the aforementioned RSS constraint or other similar safety constraints, DDMs could be used to classify the type of the front vehicle. This could help to thus better estimate the frictional coefficient of the leading vehicle and thus its braking capability, which is an important parameter in the safety constraint. However, it is challenging to assure safety when it is predicated on correct outputs of DDMs. Traditional safety measures against systematic software failures, like code reviews or white-box testing, are not sufficiently effective or applicable for DDMs [3]. In contrast, addressing DDMs requires approaches for handling uncertainty such as [4] [5] [6].

Software is essential for implementing safety constraints such as RSS constraints. It monitors values of variables in a constraint by processing sensor information and it controls actuators to influence some variables so that the safety constraint will not be violated. This software is obviously safety-critical and should comply with relevant safety standards. This becomes challenging if DDMs are involved, because safety standards do not sufficiently address the usage of DDMs. There are various ways in which DDMs could improve the accuracy and correctness of determining variables of both the RSS and other safety models. Accuracy and correctness are not only important for fulfilling safety constraints in all situations, but also for assuring that their implementation will not compromise performance or availability objectives. For these reasons, we will propose an approach that enables the usage of DDMs for ensuring safety constraints while improving performance and availability.

In previous work, we proposed Uncertainty Wrappers to quantify the likelihood that the DDM output is wrong, i.e., its uncertainty [7] [8]. In this paper, we will discuss how this quantitative uncertainty assessment could be used for safeguarding safety constraints dynamically during operation. This shall contribute to current discussions in research, industry, and standardization about acceptance criteria for using DDMs in a safety-critical context.

The paper is structured as follows: in Section II, we discuss uncertainty in the context of safety standards and summarize previous work about quantifying DDM uncertainty. Next, in Section III, we establish an autonomous vehicle scenario as a working example, incorporating the RSS model. For this example, we then discuss how Uncertainty Wrappers can be used to contribute to safeguarding the safety constraint. Further, we discuss implications and how our approach relates to alternative options in Section IV. Finally, in Section V, we conclude with a summary and brief outlook towards future work.

II. Background and Related Work

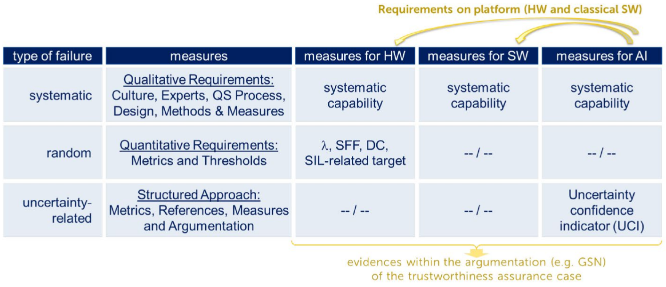
Previous [7] and related work, e.g. [4], propose to monitor the uncertainty of DDMs when using them in a safety-critical context. Specifically, they propose means to estimate the uncertainty of the output that the DDM generates. However, from a safety perspective, it is still unclear how to deal with this uncertainty because uncertainty handling differs fundamentally from commonly accepted safety engineering principles as they are described in functional safety standards.

Accordingly, we will begin by discussing how uncertainty in the output of DDM relates to existing and upcoming safety standards. We will then present previous work on addressing DDM uncertainty.

A. *Safety Standards and VDE-AR-E 2842-61*

The basic functional safety standards IEC 61508 and derivative standards (e.g., ARP4754-A and ISO26262) describe state-of-the-art safety engineering for software and hardware, but they do not describe how to deal with safety-critical DDMs. For non-DDM system elements, they apply the concept of Safety Integrity Levels (SILs). SILs effectively act as a kind of 'meta-requirement', encapsulating the rigor of assurance activities commensurate to risks of specific severity and likelihood combinations. For instance, SIL 4 of the IEC 61508 corresponds to roughly 10^9 hours per dangerous failure, requiring the most rigorous set of safety validation and verification requirements to be applied towards assuring the system's relevant functions. By developing and deploying the safety requirements alongside the system's nominal requirements, following the SIL concept implies that the residual risk remaining is acceptable to allow the system to deploy, operate and eventually decommission. However, a SIL does not describe how to deal with the 'uncertainty' that a DDM output fulfills some safety-critical properties.

The application rule VDE-AR-E 2842-61 tries to overcome this gap by introducing 'uncertainty-related' failures for DDMs and other kinds of Artificial Intelligence (cf. Fig. 1). In part one it states, "in the future it might be possible to calculate a fault rate for AI elements, consequently called λ_{AI}". Currently, it introduces an Uncertainty Confidence Indicator (UCI) as a concept to deal with uncertainty-related failures. Any claim about the achievement of a certain UCI level needs to be argued in an assurance case along with the evidence, e.g., based on metrics and specialized (test) datasets. However, it does not explain in detail how such an assurance case should look like and how to deal with 'uncertainty-related' failures considering the safety-criticality of the DDM.


Requirements on platform (HW and classical SW)
evidences within the argumentation (e.g. GSN) of the trustworthiness assurance case

type of failure	measures	measures for HW	measures for SW	measures for AI
systematic	Qualitative Requirements: Culture, Experts, QS Process, Design, Methods & Measures	systematic capability	systematic capability	systematic capability
random	Quantitative Requirements: Metrics and Thresholds	λ, SFF, DC, SIL-related target	-- / --	-- / --
uncertainty-related	Structured Approach: Metrics, References, Measures and Argumentation	-- / --	-- / --	Uncertainty confidence indicator (UCI)

Fig. 1. Traditional vs uncertainty-related failures in VDE-AR-E 2842-61

Our proposal to estimate and handle 'uncertainty-related' failures can be incorporated in an assurance case and provide a basis to argue that the residual risk due to DDM failures is acceptably low and not merely As Low As Reasonably Practicable (aka ALARP principle). This latter distinction is significant as we would use ALARP only to argue that we have done as much as we can but not to claim that this best effort is sufficient. Our proposal shall provide means to argue that fixed risk thresholds as they are given by acceptance criteria like MEM (Minimum Endogenous Mortality) are fulfilled as we handle residual uncertainty in compliance with given thresholds for acceptable residual risk (see part B below).

B. *Uncertainty in ML and Uncertainty Wrappers*

Managing uncertainty is an active research topic in the field of machine learning [9]. Key objectives are improving our understanding regarding potential sources of uncertainty and quantitatively answering how much can we rely on a specific output value provided by a DDM.

Several classifications have been proposed for the sources of uncertainty. However, the distinction between aleatoric and epistemic uncertainty is probably the best known and most cited. In this context, 'aleatoric' means that uncertainty is caused by unavoidable randomness and has no systematic cause. Instead, 'epistemic' refers to uncertainty that is systematic, in the sense that it is related to phenomena that could be considered in principle but are not sufficiently addressed by a given model [10]. The core motivation behind this separation is that the aleatoric part of uncertainty has to be accepted but the epistemic part should be reduced as much as possible, e.g., by collecting more and 'better' data. However, the distinction between the aleatoric and epistemic parts of uncertainty depends strongly on the viewpoint. Results provided for different DDMs are hard to compare since changes in the considered context including the data and space of potential hypotheses directly effects the meaning of the concepts [9].

Motivated from a more practice-oriented perspective, a classification orthogonal to aleatoric and epistemic was recently proposed by Kläs and Vollmer [11]. Their onion shell model distinguishes between three key sources of uncertainty, which can also be mathematically separated – regardless of the specific implementation of the DDM. It considers uncertainty directly attributed to limitations in the DDM (*model fit*), differences in the quality level of the model inputs (*input quality*), and the possibility that the model is applied outside its originally intended application scope (*scope compliance*) [7].

The first type of uncertainty can be directly tackled by improving the DDM, e.g., acquiring more data or adjusting the learning approach. It is measured by traditional validation metrics such as accuracy or the true positive rate on unseen, representative data. The second one can be addressed by modeling the influence of input quality on the uncertainty in the DDM output. For example, missing or imprecise input data will commonly influence the quality of the DDM output. The third one relies on not only appropriately defining the target application scope of the DDM but also monitoring compliance during the DDM application, e.g., checking boundary conditions and using novelty or out-of-distribution detection techniques to identify unusual settings [7].

Building upon these foundations, 'Uncertainty Wrappers' were proposed as a DDM-agnostic pattern to provide dependable uncertainty estimates [8]. As Fig. 2 illustrates, an uncertainty wrapper deals with a DDM that it encapsulates as a black box. Factors that are expected to influence input quality or scope compliance are modeled in a quality and scope model, respectively. Considering a strongly simplified application example for traffic sign recognition [5], the precipitation rate measured by a rain sensor could represent a quality factor indicating limitations in visibility. The GPS coordinate could indicate as a scope factor whether the model is applied outside its target application scope (e.g., a specific country). In opposite to many existing approaches, uncertainty wrappers accept a confidence requirement as an input when providing an uncertainty estimate, which makes their estimate dependable on a statistically interpretable level. We will further discuss the implications of choosing a specific confidence level for uncertainty estimates in Section III.

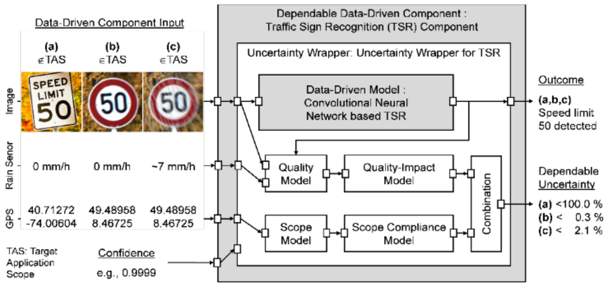


Fig. 2. Illustration of uncertainty wrapper pattern [5]

III. Dynamic Risk Management using Uncertainty Wrappers

In this section, we discuss how to exploit dependable uncertainty estimates, as provided by Uncertainty Wrappers, to establish useful safety constraints. First, we will introduce a safety constraint and a safety monitor for ensuring that the constraint is fulfilled. Second, we will extend this safety monitor with a DDM to determine a variable in the safety constraints more precisely and thus avoid worst-case assumptions that affect performance. Third, we will explain how to integrate Uncertainty Wrappers in the safety monitor. In doing so, we ensure that violations of the safety constraint are sufficiently unlikely and that we establish sufficient confidence in this safety objective.

A. Running Example: Responsibility-Sensitive Safety Model

It is often hard to specify safe behavior for an autonomous system. However, a comprehensible specification that defines what we mean by "safe behavior" is necessary for assuring and arguing safety. Approaches for specifying safe autonomous behavior often end up with some safety constraints. A prominent example in this regard is published by Intel/Mobileye in 2017, coined *Responsibility-Sensitive Safety (RSS)* [2]. In this paper, we focus on the RSS constraint for keeping a safe distance to a vehicle in front, aka the 'lead' (denoted as *L*), from our own vehicle, aka the 'follower' vehicle (denoted as *F*). The formula for determining the safe distance is given in Equation (1).

$$d_{safe} = \left[\begin{array}{c} v_F\,\rho + \frac{1}{2}\,a_{max,acc,F}\,\rho^2 + \frac{(v_F + \rho\,a_{max,acc,F})^2}{2a_{min,brake,F}} \\ - \frac{v_L^2}{2a_{max,brake,L}} \end{array} \right]_+ \qquad (1)$$

Effectively, the first three terms together represent the stopping distance of the follower vehicle considering (i) a reaction distance based on follower speed v_F and reaction time ρ, (ii) an acceleration distance (assuming the follower constantly accelerates with $a_{max,acc,F}$ during reaction time), and (iii) the follower braking distance, when the follower constantly brakes with deceleration $a_{min,brake,F}$. The last term represents the leader's braking distance. To estimate the actual safe distance d_{safe}, we subtract the leader's braking distance from the follower stopping distance. By means of this formula, we can define our safety constraint for the current distance d as follows: $d \geq d_{safe}$.

We have to assure at design-time that the safety constraint is fulfilled in all operational situations that an autonomous system will encounter. To this end, we have to determine the variables in the safety constraint so that unavoidable deviations between actual and determined values do not lead to a violation of the safety constraint. Effectively, we must ensure that the current distance d is not measured too high and the target distance d_{safe} is not underestimated in order to ensure that the corresponding controller controls the distance safely. Consequently, we have to ensure that $d_{estimated} \leq d_{actual}$ and $d_{safe,\ estimated} \geq d_{safe,\ actual}$ holds. An example of how this can be done without DDMs is presented in [12]. The cited example considers a platooning scenario where trucks drive very close to minimize fuel consumption. In such a scenario, the safe distance should be determined to

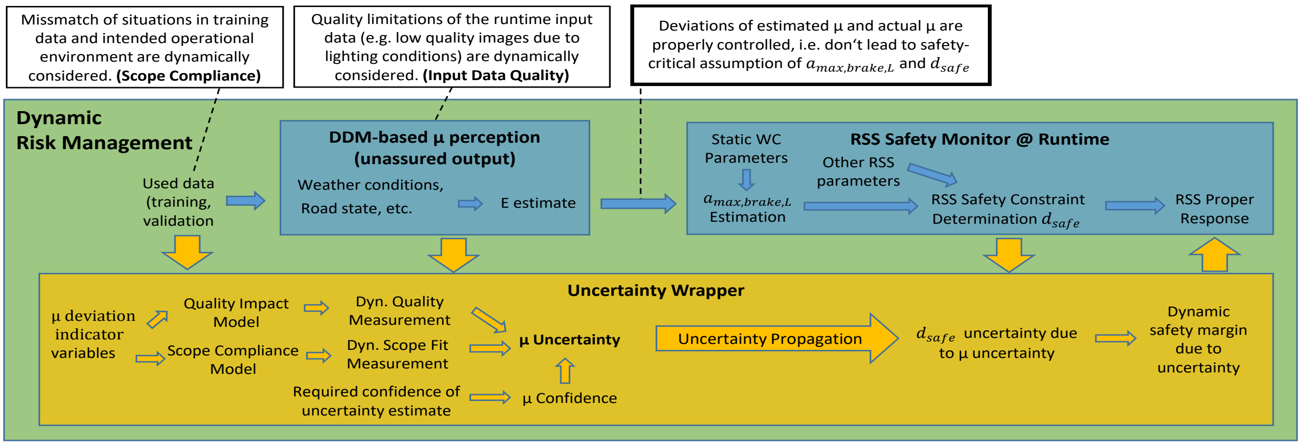

Missmatch of situations in training data and intended operational environment are dynamically considered. (Scope Compliance)
Quality limitations of the runtime input data (e.g. low quality images due to lighting conditions) are dynamically considered. (Input Data Quality)
Deviations of estimated μ and actual μ are properly controlled, i.e. don't lead to safety-critical assumption of $a_{max,brake,L}$ and d_{safe}
Dynamic Risk Management
Used data (training, validation
DDM-based μ perception (unassured output)
Weather conditions, Road state, etc.
E estimate
Static WC Parameters
RSS Safety Monitor @ Runtime
Other RSS parameters
$a_{max,brake,L}$ Estimation
RSS Safety Constraint Determination d_{safe}
RSS Proper Response
Uncertainty Wrapper
μ deviation indicator variables
Quality Impact Model
Scope Compliance Model
Dyn. Quality Measurement
Dyn. Scope Fit Measurement
μ Uncertainty
Required confidence of uncertainty estimate
μ Confidence
Uncertainty Propagation
d_{safe} uncertainty due to μ uncertainty
Dynamic safety margin due to uncertainty

Fig. 3. Uncertainty Wrappers enable using data-driven models for assured dynamic RSS Monitoring

be as precise as possible. Any worst-case approximation increases unnecessarily the distance whenever the worst-case is not present and this lowers the benefit of platooning. For this reason, we do not want to worst-case approximate the maximum leader deceleration $a_{max,brake,L}$ and determine it as precisely as possible. For this purpose, we analyze which other variables or constants like physical constants we need to know for determining it during operation. One of these variables is the *friction coefficient* μ of the lead vehicle. The worst-case for this variable would be a maximal friction, because the follower has to keep a larger distance if the leader has higher friction and thus a smaller breaking distance. Equations (2) and (3) concretize the physical relationship between $a_{max,brake,L}$ and μ. In general, the effective maximum leader deceleration limit $a_{max,brake,L}$ is influenced by all driving resistance forces acting in longitudinal direction (brake system brake force, air resistance, rolling resistance, road inclination resistance). For the sake of exemplification, but without loss of generality, Equation (2) only considers the effective deceleration to be bound by the maximum brake force the brake system is capable of generating and the traction limit determining, how much of the generated force can be transferred effectively on the road. While the brake system's limit is mainly influenced by construction, brake pad wear and the pedal force a driver is likely to apply, the traction limit depends on the vehicle's mass m, the gravitational constant g and, most relevant for this paper, the friction coefficient μ.

$$a_{max,brake,L} = \min(\frac{F_{b,traction,limit}}{m}, \frac{F_{b,brakesystem,limit}}{m}) \qquad (2)$$

$$F_{b,traction,limit} = m \cdot g \cdot \mu \qquad (3)$$

For the remainder of the paper, we will use the determination of μ by means of a DDM as an example for a variable in a safety constraint. We assume that apart from v_F and v_L all other variables are kept static at runtime with reasonably chosen and argued worst-case assumptions. This example is realistic as no direct physical sensors exist to date for measuring μ and components such as the *Road Condition Observer* [13] propose the use of DDMs to approximate μ based on other indicator variables such as weather conditions or road state.

B. Dynamic RSS Parameter Estimation Assurance

Having introduced the RSS safety constraint and the assurance claim associated to the $d_{safe,estimated}$, Fig. 3 presents the integration of the *Uncertainty Wrapper* concept into a runtime safety monitoring channel based on the RSS safety constraint. The blue highlighted part shows the nominal RSS Safety Monitor that shall be made more dynamic through the dynamic determination of the road friction μ in a data-driven component. If we introduce this new component, the argumentation claim required before would have to change. Previously, this claim would assure that "static assumptions used for $a_{max,brake,L}$ are valid in the intended operating context". Instead, the claim changes to assure that the "dynamically estimated μ does not lead to a safety-critical estimation of $a_{max,brake,L}$", i.e. an overestimated braking distance. Since data-driven components cannot be assured by the same means as traditional software components, new approaches are required to deliver evidence that the usage of highly performant DDM-based perception techniques *within a safety function* do not violate the original safety constraint. For instance, such a violation can occur, if scope compliance or the input data quality is not adequately assured (Fig. 3, Top Left).

In a similar way as the *RSS Safety Monitor* monitors safety violations of the nominal driving function (also known as "Doer-Checker" pattern), the *Uncertainty Wrapper* (UW) monitors the DDM specifically for safety-critical deviations between DDM output and the ground truth. These deviations are formally captured in an uncertainty estimate (figuratively described as "Checking the Checker" pattern). Thus, the DDM output is enriched by an uncertainty value describing how likely the outputted μ value deviates in a safety-critical manner (Fig. 3, middle yellow block). To that end, the UW performs *assurable* (i.e. non-DDM) dynamic measurements of the quality of the input data and of how well the training and validation data matches the current operating environment.

Following this approach of "checking the checker", a natural question may be, who checks the new checker and so on. This refers to having a confidence threshold defining how much we want the uncertainty result to be trustable, more formally the residual probability of the uncertainty estimate being wrong. The UW can be parameterized with such a confidence value, thus constructively braking the "checking the checker" recursion.

Having an estimate of both the μ value and associated uncertainty, the next step is to analyze which effect the μ uncertainty will eventually have on the uncertainty of

$d_{safe,estimated}$ and on the likelihood that this will lead to the unsafe critical condition $d_{safe,estimated} < d_{safe,actual}$. In Fig. 3, this step is called *Uncertainty Propagation* and may use physical models such as the ones introduced in Equations (2), (3) to determine the relation between μ and $a_{max,brake,L}$.

At this point, the dynamic uncertainty analysis in the UW is finalized and the right conclusions need to be drawn to assure the distance setpoint $d \geq d_{safe,actual}$, i.e. how much additional distance the following vehicle should have from its lead vehicle to account for the uncertainty. This additional space is the dynamically determined margin d_{margin} (Fig. 3, right of yellow block) that is added to $d_{safe,estimated}$ to ensure that the safe distance is not underestimated and the condition $d_{safe,estimated} + d_{margin} > d_{safe,actual}$ holds with a sufficiently high probability. The latter probability threshold can be derived from the acceptable residual risk concerning a potential violation of our safety constraint.

In summary, UWs enable the use of DDM within safety monitoring channels such as RSS. The effect is an RSS Proper Response leading to an improved system performance due to dynamic RSS parameter measurements and at the same time provides evidence the residual risk of violating the safety constraint due to uncertainties is acceptably low.

C. Expressing RSS in terms of Uncertainty Wrappers

In this section, we first make explicit what we intend to obtain on the level of the RSS safety constraint by applying an UW for the DDM that estimates the friction value μ. Then we explain what this would imply for information that needs to be provided by the UW and how this information could be obtained. Finally, we discuss the relationship between the required confidence and the uncertainty estimates provided.

As argued in the previous section, in the end we are interested in the additional distance d_{margin} we need to consider due to uncertainty in our DDM-based estimate of μ. We can obtain this margin if we have a cumulative uncertainty distribution U_d over the condition $d_{safe} \leq d$ as illustrated in Fig. 4. Assuming that we only accept that the condition is violated with probability u_{max}, we would use the distribution to find the distance $d_{u_{max}}$ with $d_{u_{max}} = {U_d}^{-1}(u_{max})$ and obtain d_{margin} as the difference $d_{u_{max}} - d_{safe,estimated}$.

Obviously, DDM outcomes can not only underestimate but also overestimate the actual d_{safe}. However, we can ignore uncertainty related to overestimation from the perspective of assuring the considered safety constraint since this kind of uncertainty does not affect the fulfillment of the constraint. It only affects the utility of the DDM outcome to achieve a safe distance that is as small as possible.

Since we assumed in our example, for the sake of simplicity, that μ is the only variable that introduces uncertainty, we can directly derive the cumulative uncertainty distribution U_d given the distribution U_μ. Using the physical model in Equation (2) and (3) that specifies the relationship between μ and $a_{max,brake,L}$ together with the right part of Equation (1) as a function f, we get $u_d = f(U_\mu(\mu))$.

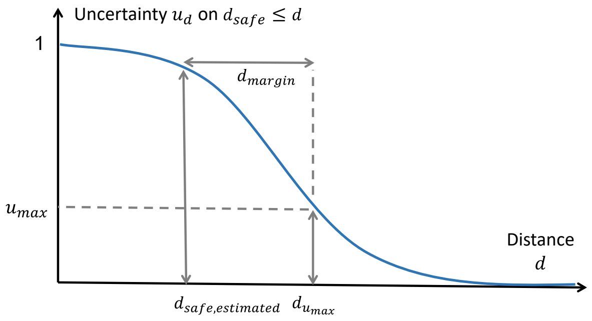


Fig. 4. Cumulative uncertainty distribution U_d

The cumulative uncertainty distribution U_μ, which needs to be determined, is illustrated in the upper part of Fig. 5. A simple option to obtain U_μ is extending the basic UW concept to provide not only an uncertainty estimate for the DDM outcome $\mu_{estimated}$, but also for the outcome extended by the margins μ_{margin_i} with $i = 1 \ldots n$. This results in n further uncertainty estimates that approximate U_μ, as seen in the lower part of Fig. 5.

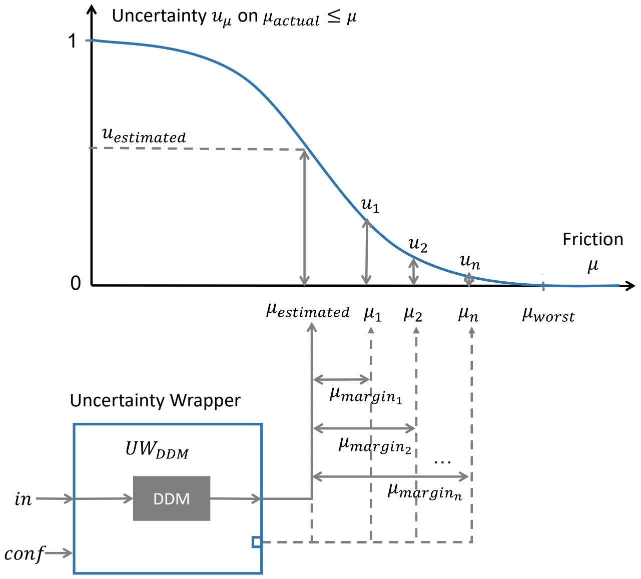


Fig. 5. Cumulative uncertainty distribution U_μ based on UW

Obtaining additional uncertainty estimates for different margins does not incur a performance overhead during operation, since the UW uses decision trees to provide the uncertainty estimates. Thus, the additionally required uncertainty estimates can be precalculated and stored together with the standard uncertainty estimate in the respective leaves of the tree. Only the required storage would be increased from O(m) to O(nm) where m is the number of leaves in the applied decision tree.

In the following, we discuss the implications and relevance of requesting a certain level of confidence for the considered uncertainty estimates. In this setting, we need to remind that the uncertainty estimates provided by the UW are empirically determined on a representative data sample called the calibration dataset. Because the calibration dataset is a sample from possible observations in the target application scope but it does not encompass them all, there is the possibility that the actual uncertainty in the DDM outcomes is underestimated. To reduce the chance of providing systematically overconfident uncertainty estimates and achieve a defined level of confidence in our arguments, we need to consider a statistical margin for the uncertainty estimate itself. The size of this margin can be determined by calculating a confidence boundary based on a requested confidence level [8]. The size of the margin depends, on the one hand, on the number of available relevant samples in the calibration dataset and, on the other hand, on the observed ratio between the DDM outcomes fulfilling or violating the considered condition, $\mu_{estimated} + \mu_{margin_i} > \mu_{actual}$.

The confidence level $conf$ gives a lower limit on the likelihood that the results of the uncertainty estimation approach do not underestimate the uncertainty under the given conditions. Specifically, it limits the related type I error α to $1 - conf$. If such a statement is not provided by the uncertainty estimation approach, the validity of its estimates cannot be justified in a safety argumentation. To give an example, one would obviously not argue that the output of a DDM includes 0% remaining uncertain when it was tested only on a single data point on which it provided a correct result (although it was correct for 1/1 = 100% of the tested cases) [8].

If we assume μ_{worst} is a worst-case upper bound on μ in the target application scope (cf. Fig. 4) and apply the law of total probability, we can derive for each μ a holistic uncertainty estimate that considers not only the estimated uncertainty but also the uncertainty attributed to the approach used to determine the uncertainty estimate:

$$u_{\mu}^{conf} = 1 - conf\big(1 - u_{\mu}\big) \qquad for\, \mu < \mu_{worst} \quad (4)$$

$$u_{\mu}^{conf} = 0 \qquad for\, \mu \geq \mu_{worst} \quad (5)$$

Equation (4) considers the cases in which we can be both certain and confident that $\mu_{actual} \leq \mu$. In all other cases, this means in cases where we are either uncertain about $\mu_{actual} \leq \mu$ or cases where we might be certain in $\mu_{actual} \leq \mu$ but are not confident in our approach to obtain this certainty, we need to fall back to our worst-case approximation for μ, cf. Equation (5). The equations indicate that neither uncertainty nor confidence can be ignored. It seems also reasonable to target for a confidence that is at least as high as the targeted certainty, especially, since it is commonly much easier to provide uncertainty estimators with higher confidence than getting DDMs that provide output with lower uncertainty. We can increase confidence simply by increasing the quantity of calibration data or accepting a slightly more conservative margin. The equations also indicate that even if we dynamically estimate the value of a parameter, we still need a reasonably chosen worst-case value μ_{worst}.Therefore, we also propose to choose for the confidence the highest probability that is reasonably practicable, and at least a probability that is higher than $1 - u_{max}$.

The probability u_{max} is directly related to the necessary risk reduction as it states the probability with which we can accept a violation of our safety constraint. In our example, we can derive this probability from the risk that exists in the hazardous situation where the distance is smaller than the safe distance. In this situation we cannot avoid a crash if the leading vehicle is performing a full brake. In order to derive u_{max} from such a hazardous situation, we see three options.

The first option is based on integrity levels, as proposed in most safety standards, and related target failures rates. We would simply determine the integrity level for the hazardous situation that we drive so close that we cannot avoid a rear-end collision. For this integrity level, we would then use the related probabilistic target values for deriving the probability u_{max}. One might end up in very low probabilities depending on the assumptions that one makes when deriving u_{max} from the target failure rates. For instance, ASIL D refers to 10^{-8} failures per hour. If we would transfer this failure rate directly to the critical failure mode of the μ perception component and assume that it is scheduled with 10 Hz, then the acceptable probability for the critical failure in one execution would be 2,8 10^{-13}. However, we could also argue that a few wrong outputs are not critical as they can be identified and removed. This would open the discussion about the independence between failures at different points in time. We omit this discussion here as it goes beyond the scope of the paper but we conclude that it is possible to derive u_{max} from integrity levels. The principle behind the target values of integrity levels is MEM (see Section II.A).

If the target values of MEM are not achievable, then it might be possible to argue safety by means of other risk acceptance criteria. This opportunity is the second option that we see for determining u_{max}. For instance, in the context of autonomous driving there is the idea to compare the safety performance of an autonomous vehicle with the average safety performance of a human driver and assure that the accidents will decrease when increasing automated driving. This approach is referred to as positive risk balance and already mentioned in ISO/SAE 21434.

A third option is to consider that the risk of violating the safety constraint is not always the same. For instance, the severity of a potential accident when violating the safe distance constraint depends on the current speed of the vehicles. Further, an accident will only occur if the leader actually brakes and the likelihood of braking also depends on the current situation. One approach to deal with this issue is to partition the space of critical situations where the safety constraint is violated. For each partition, we would then identify a potentially different u_{max} that has to be achieved. For instance, we could split the safety constraint into two constraints where one considers vehicle speeds below 5 km per hour and the other considers speed values of 5 km per hour and higher. For the safety constraint addressing low speed, we could then specify a lower u_{max} if we could argue that a potential collision would be less severe. Continuing along this line of thought, we would eventually arrive at a continuous function that dynamically determines u_{max} for the current operational situation.

IV. Discussion

One may argue that our proposal significantly increases the complexity of the safety concept. A simplistic alternative to dynamically estimating uncertainty during operation could be seen in applying statistical testing at design-time using a representative test dataset. We would then calculate an upper bound for the error probability of the DDM considering errors that would result in violating a given safety constraint. We would accept the DDM if the calculated upper bound on uncertainty as determined on the test dataset does not exceed the acceptable uncertainty u_{max}. However, this approach has a number of limitations:

(1) It would not be applicable if we follow the third option of deriving u_{max} (at the end of section III.C) and use a function that dynamically determines u_{max} for the current operational situation.

(2) The actual uncertainty u in the output of a DDM is usually not constant for the complete operational design domain, but depends on some influence factors that are not present in all situations (e.g., dirt on the sensor, last update on current weather data). That means we do not want to have a one-time determined average, but instead a situation-aware uncertainty estimate. Our proposed approach can provide such situation-aware uncertainty estimates, if the relevant influence factors have been identified. Identifying all relevant influence factors can be very challenging but is a topic outside the scope of our discussion. However, even if we would omit an influence factor, we could still determine the correct uncertainties for our considered situations. Fortunately, the only effective penalty then would be that our approach missed the opportunity of identifying more precise uncertainty bounds for those more fine-grained situations.

(3) A safety constraint might contain several parameters that are determined by DDMs, each with an uncertainty that varies on the given situation. If we combine the pieces of information on uncertainty, we can choose - on the level of the safety constraint - the value with the best utility that still has an acceptable uncertainty u_{max} of not violating the safety constraint. This combination is possible if we can provide for each parameter p a distribution U_p over the possible values of the parameter that the actual value will only deviate from with uncertainty u_p, respectively. If we obtain such distributions, we can compute the resulting uncertainty distribution for violating the respective safety constraint using appropriate uncertainty forward propagation methods.

The major limitation we currently view in our approach is the lack of objective evidence on its utility from a case study. While the Uncertainty Wrappers concept has been established and evaluated in comparison with other uncertainty estimation approaches in previous work, evaluating the effectiveness of its application with respect to its benefits to performance and availability of safety constraint monitoring as we have outlined in this paper remains to be seen.

Finally, our approach assumes that uncertainty cannot be avoided completely and that it has to be addressed with probabilistic claims. So far, safety standards consider probabilistic claims only for random hardware failure but not for software. When extending the probabilistic reasoning to DDMs, a question arises with regards to whether random hardware failures and uncertainty-related failures should be considered together in order to assure that the target failure rates from integrity levels are achieved. However, this question is not specific to our approach and relates to the question of combining a λ_{AI} and the traditional λ for hardware failure rates to an overall λ.

V. Conclusion and Further Work

Autonomous vehicles and other safety-critical autonomous systems require DDMs. A key challenge in this context is handling the uncertainty of DDM outputs. In particular, safety controllers for ensuring safety constraints such as RSS can hardly rely on outputs of DDMs due to the uncertainty. Therefore, safety controllers have to operate under worst-case assumptions for the information that DDMs could provide. However, these worst-case assumptions lower performance and/or availability of the nominal behavior. We argue that being able to determine the uncertainty of DDMs further enables their usage as a performant perception means within dynamic safety-constraint monitors with high integrity requirements.

We have proposed an approach for solving this challenge by determining the uncertainty during operation with a defined level of confidence and using the uncertainty estimates within a safety controller. The safety controller dynamically adjusts margins according to the current uncertainty in order to ensure that the violations of the safety constraint are sufficiently unlikely. In this way, performance and availability of the nominal behavior is improved.

By construction, the result of our approach is always better than working with worst-case assumptions, but it is application-specific. There is a trade-off to consider whether the benefits justify the effort to replace worst-case assumptions with dependable uncertainty estimates.

Another contribution of our approach concerns the safety assessment and assurance of safety-critical DDMs. Our approach provides a basis to argue that the residual uncertainty, after minimizing it at design-time as much as reasonably practicable, is adequately low and handled appropriately.

Our future work is related to the two contributions. First, we plan to evaluate the approach in different applications like truck platooning in order to quantify the benefits for performance and availability. The use cases of the research projects we are participating in, including those acknowledged below, offer us options for investigating autonomous system use cases. Second, we want to concretize the related safety argument and integrate it into an assurance case for machine learning [14] on the example of cobots and automated guided vehicles.

Acknowledgment

Parts of this work have been funded by the Observatory for Artificial Intelligence in Work and Society (KIO) of the Denkfabrik Digitale Arbeitsgesellschaft in the project "KI Testing & Auditing" and the Horizon 2020 project SESAME (Grant Agreement ID: 101017258). We would like to thank Mohammed Naveed Akram and Sonnhild Namingha for the initial review of the paper.

References

[1] H. Kagermann, N. Gaus, J. Hauck, J. Beyerer, W. Wahlster and H. Brackemann, „Autonome Systeme-Chancen und Risiken für Wirtschaft, Wissenschaft und Gesellschaft. Fachforum Autonome Systeme im Hightech-Forum. Abschlussbericht–Langversion, Berlin, Germany, 2017.

[2] S. Shalev-Shwartz, S. Shammah and A. Shashua, "On a formal model of safe and scalable self-driving cars", arXiv preprint:1708.06374, 2017.

[3] R. Salay and K. Czarnecki, "Using machine learning safely in automotive software: An assessment and adaption of software process requirements in ISO 26262," arXiv preprint:1808.01614, 2018.

[4] M. Henne, A. Schwaiger, K. Roscher and G. Weiss, "Benchmarking Uncertainty Estimation Methods for Deep Learning With Safety-Related Metrics," in Proceedings of the Workshop on Artificial Intelligence Safety (SafeAI 2020), New York, USA, 2020.

[5] T. Bandyszak, L. Jockel, M. Klas, S. Torsleff, T. Weyer and B. Wirtz, "Handling Uncertainty in Collaborative Embedded Systems Engineering," in Model-Based Engineering of Collaborative Embedded Systems, Cham, Springer, 2021, pp. 147-170.

[6] K. Aslansefat, I. Sorokos, D. Whiting, R. Kolagari and Y. Papadopoulos, "SafeML: Safety Monitoring of Machine Learning Classifiers through Statistical Difference Measure.," in International Symposium on Model-Based Safety and Assurance (IMBSA) 2020, Lisbon, Portugal, 2020.

[7] M. Kläs and L. Sembach, "Uncertainty Wrappers for Data-Driven Models," International Conference on Computer Safety, Reliability, and Security, vol. 11699, pp. 358-364, 2019.

[8] M. Kläs and L. Jöckel, "A Framework for Building Uncertainty Wrappers for AI/ML-based Data-Driven Components," in Third International Workshop on Artificial Intelligence Safety Engineering (WAISE 2020), Lisbon, Portugal, 2020.

[9] E. Hullermeier and W. Waegeman, Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods, Machine Learning 110, Cham, Springer, 2021, pp. 457-506. https://doi.org/10.1007/s10994-021-05946-3.

[10] A. Der Kiureghian and O. Ditlevsen, "Aleatory or epistemic? Does it matter?," Structural Safety, vol. 31, no. 2, pp. 105-112, 2009.

[11] M. Kläs and A. Vollmer, "Uncertainty in machine learning applications: A practice-driven classification of uncertainty," International Conference on Computer Safety, Reliability, and Security, pp. 431-438, 2018.

[12] J. Reich, D. Schneider, I. Sorokos, Y. Papadopoulos, T. Kelly, R. Wei, E. Armengaud and C. Kaypmaz, "Engineering of Runtime Safety Monitors for Cyber-Physical Systems with Digital Dependability Identities," in International Conference on Computer Safety, Reliability, and Security, 2020.

[13] B. Hartmann and A. Eckert, "Road condition observer as a new part of active driving safety," ATZ Elektron Worldwide, vol. 12, pp. 34-37, 2017.

[14] R. Hawkins, C. Paterson, C. Picardi, Y. Jia, R. Calinescu and I. Habli, "Guidance on the Assurance of Machine Learning in Autonomous Systems (AMLAS)," arXiv:2102.01564, 2021.