Operational Design Domain Monitoring with Uncertain Measurements

Thibault Charmet, Véronique Cherfaoui, Javier Ibanez-Guzman, Alexandre Armand

Operational Design Domain Monitoring with Uncertain Measurements

Thibault Charmet[1,2], Véronique Cherfaoui[1], Javier Ibanez-Guzman[2], Alexandre Armand[2]

***Abstract*— The increasing automation in Intelligent Vehicles (IVs) necessitates robust safety measures. This paper proposes a method to address this challenge using the concept of Operational Design Domains (ODDs) and real-time monitoring. We present a framework for calculating an Operational Domain (OD) membership degree that accounts for measurement uncertainties and fuzzy boundaries within the ODD. This method utilizes fuzzy sets to represent the vagueness of real-world driving environments. Additionally, we introduce a taxonomy-based approach for formally defining ODDs independent of the vehicle's architecture. Our approach is demonstrated in simulation, by monitoring the vehicle's OD in real-time and by determining when it exits its defined ODD. The results pave the way for verifiable safety rules based on OD membership degrees.**

I. INTRODUCTION

Intelligent vehicles (IV) increasingly feature higher levels of automation. As a result, there is much concern across industry and academia regarding operational safety. While there is no consensus yet on the most reliable way to validate advanced driver-assistance systems (ADAS) and autonomous driving systems (ADS), several methods exist, such as scenario-based testing and real-time operational domain (OD) restriction based on the specification of the operational design domain (ODD). In our case, we focus on the role of the ODD concept as a safety guarantee for IVs and how it can act as a safeguard for complex mobile systems when combined with real-time monitoring of their OD. The ODD concept is often used, either during the design phase of the V cycle, to define the environment in which the designed system should be able to operate, or during the operation phase, as specifications describing the OD in which the system can evolve. This is a transparent way to specify minimum safety requirements for systems, that can be understood by both various non-technical end users like regulators and by experts. It is defined by [1] as "Operating conditions under which a given driving automation system or feature thereof is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics." For example, a valet parking system could be designed to operate in an indoor or outdoor parking lot, but not on the road. A SAE level 3 [1] traffic jam pilot could be designed for driving in slow traffic, during the day, in favorable weather conditions. ODD descriptions are useful tools to describe the capabilities of a system through the situation it can manage. The ODD is often used to describe the suitable operating conditions for a system. This is done via documents, tables, etc. However, this use remains generally abstract. This paper aims to contribute to the development of a formal framework for defining and monitoring the ODD.

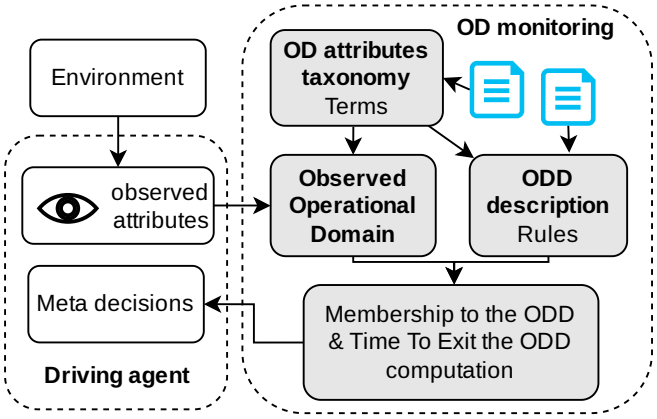

Fig. 1. Functional architecture.

A. Operational Design Domain Monitoring

Operational Design Domain Monitoring or Operational Domain Monitoring is the task of determining, whether or not a system is operating in the domain for which it was designed. This monitoring can be performed: 1) in advance, during the mission planning phase, using a priori information, 2) in a dynamic manner, during the trip, using live information.

1) Monitoring from a priori information: Before the trip, the available map and a priori information can be used to anticipate which sections of the trip will be outside the ODD. Generally speaking, this concerns information such as the structure of the road network, road geometry, and known areas (geofencing, school zones, etc.). It can also include traffic and other density/probability information that can be aggregated by a cloud service. In this case, it is possible to anticipate, and in the case of a level 3 system, to return the driving task to the user at the appropriate moment.

2) Monitoring from live information: On the other hand, some information cannot be anticipated, or only to a limited extent, due to its nature. This may involve road users, who are only detected once they are in range of the vehicle's

*This work has been carried out within SIVALab, a shared laboratory between Renault and Heudiasyc (Université de technologie de Compiègne, CNRS), and financed by Renault and the ANRT (Association nationale de la recherche et de la technologie) as a CIFRE PhD.

[1]Université de technologie de Compiègne, CNRS, Heudiasyc laboratory (Heuristics and Diagnosis of Complex Systems), Compiègne, France name.surname@hds.utc.fr

[2]Ampère, Guyancourt, France name.surname@renault.com

sensors, erroneous a priori information, or other dynamic information such as environmental conditions. Moreover, the observation of these elements is uncertain, and this uncertainty must be taken into account to determine whether the environment in which the system is evolving is outside the ODD. In this category are: road users and objects, observed behaviors, operational constraints, the state of various connectivity services, GNSS, etc., as well as environmental conditions such as weather, visibility, luminosity, particular situations, and events like roadworks.

It is clear that certain attributes can be considered as both a priori and live, depending on the nature of the information source. As pointed out in [2], multiple strategies could be used to process different types of attributes. Live information is challenging. As they need to be measured by sensors, they are dynamic, generally have high uncertainty, and cannot be anticipated very early. Therefore, this will be the subject of this paper.

B. Contribution and outline

The objective of this paper is to propose a method for estimating the degree of membership of a system's observed operational domain (OD) to its ODD (See Figure 1). This degree of membership, ranging from 0 to 1, is used to decide whether or not the system is within its ODD. It takes into account the uncertainty of the observations made. We also introduce the possibility of using fuzzy sets to model the numerical attributes of an ODD, as a tool to better represent the vagueness of real-world boundaries. In order to do that, we present a method for representing and formally defining an ODD textually, representing the known limits of a driving system, independently of the system's architecture. This definition is based on a set of attributes of the driving environment (road network, road users, weather conditions, etc.), organized in the form of a taxonomy.

We validate this method in simulation by monitoring, in real-time, the operational domain of a vehicle, and determining when it leaves a defined ODD. The result takes the form of a degree of OD membership to the ODD, that depends on the uncertainty of the observed environment and fuzzy boundaries. To illustrate it, two simple use cases are presented, including one inspired by an accident scenario. Potential applications for meta-decision will be discussed using the transition-of-control (ToC) fallback required in level 3 ADS as an example.

We will first look at the literature on the topic of ODD monitoring in Section II, before presenting the ODD description method in Section III. Then, the proposed method to monitor uncertain OD attributes is presented in Section IV and the results in Section V. Our results show that this approach allows to define verifiable rules in order to monitor the domain of operation of driving systems.

II. RELATED WORKS

Related works can be grouped into 4 categories: ODD monitoring, ODD identification, ODD description, and ODD attributes. This work is based on [3] and uses the methodology provided to build an ODD monitoring system. Several works have approached ODD monitoring. [4] addressed the need for functional boundaries (i.e. an ODD) and a methodology to define them for driving systems. [5] address the restriction of the ODD based on the degraded capabilities of the system. [2] recognizes that the monitoring task should be divided based on the type of ODD attributes and that different strategies may be applied. [6] monitor the ODD of a 2D laser-based localization algorithm by using machine learning (ML) algorithms to look at the extracted feature first. It verifies if the inputs contain distinguishable information that the localization system can use. [7] presents a functional architecture for reasoning with known system capabilities and environment monitoring. Each capability has a contract with preconditions and guarantees. A dependency tree therefore exists between capabilities, so that when one of them becomes unavailable, those downstream also lose their guarantees. An Operational Domain Monitor (ODM) is used to retrieve external information used as input to determine the service quality of a capacity. [8] employs a statistical approach to identify risky situations by computing their level of compliance with fuzzy requirements. The approach aggregates the values of relevant characteristics of situations from fleet logs. A level of acceptable risk can be defined to accept the most unlikely occurrences. The values are then compared to the requirements to obtain a compliance score between 0 and 1.

Other works are focused on the upstream task of identifying the limits of given systems, that is to say, their ODD. [9] does out-of-ODD research via counter example trained machine learning (ML). [10] assess the ODD of lane support system (LSS) in different weather and road conditions. [11] uses a scenario based approach to quantify risk and identify some ODD attributes, demonstrating it using two learning-based agents.

The task of ODD monitoring requires the ODD to be described first. It should be noted that there is no single way of representing the ODD in the literature. Works like [12], [13] are based on ODD descriptions. Either using domain specific language or YAML-based descriptions. [14] proposed a two-level language to describe the ODD. The ODD description is represented by a structured natural language that can be converted into a SQL-like, machine-interpretable formal language. Other projects like OpenODD are under conception [15].

Similarly, since these descriptions are based on attributes of the ODD representing elements of the environment, several works have been organizing them as a taxonomy of elements [16], [17], [18], [19] and more recently [20], [21]. [22] provides an extensive survey of the literature around ODD, including ODD monitoring.

No work has yet explored how to use uncertainty from perception and situation assessment to compute the membership of the OD to the ODD. In order to quantify risk, accurate uncertainty quantification and propagation from the perception systems to downstream systems is essential. Thus,

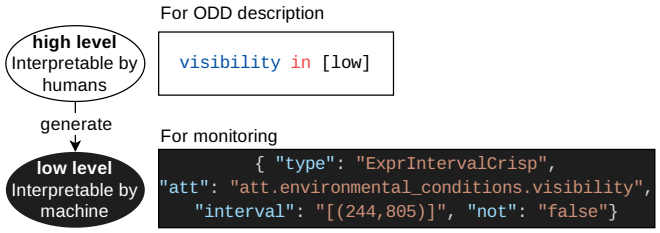

Fig. 2. The two levels of ODD description. The high-level description goal is to be as readable as possible while being usable to generate a lower-level version that is directly machine-readable.

like other systems, the ODD monitoring output should be representative of the uncertainty from the input. This work focuses mainly on ODD monitoring, and secondly on ODD description and attributes. The ODD will be considered as already identified.

III. ODD DESCRIPTION

A. High and low level ODD description

An ODD description must contain the information needed to define all the operating conditions under which a system has been designed. Or, under Closed-World Assumption [23], all the operating conditions under which a system has not been designed for. Even with automatic ODD identification methods, the ODD description should always be easily readable by non-technical users and experts. On the other hand, ODD monitoring requires the ODD description to be machine-readable. Thus, the role of ODD description is to allow for the most human-readable formal description that can be machine evaluated. A statement like ”Can’t drive faster than 30 km/h with low or worse visibility.” will be written as follows:

```
reject target_speed > 30 and visibility <= low
```

As such language cannot be directly machine interpreted, a lower level description is required. Tools like Xtext or its heir Langium allow parsing high-level custom languages, called Domain Specific Language (DSL), and converting them in lower level languages or data structures. Here, the ”low-level” ODD description will take the form of a JSON structure that contains all the information needed to describe the ODD statements. In Figure 2, there is a sample of the human-readable ”high-level” and its equivalent machine-readable ”low-level” JSON description. In this paper, the simplest human-readable format will be used to illustrate ODD statements.

B. ODD description syntax

Syntactically, an ODD is a list of statements, accepting or rejecting a type of OD. Two types of ODD descriptions could be used: permissive and restrictive. In a permissive description, every OD is accepted by default and each statement starts with ”reject” to add additional constraints. We could have the following statements: ”Heavy rain or worse is not OK.” ”Highways in low or worse visibility are not OK.”

```
reject rain.intensity >= heavy
reject road_importance in [motorways] and
    visibility <= low
```

Restrictive description, are the opposite, no OD is accepted by default, and each statement adds a valid OD. In this paper, only permissive descriptions will be used.

In statements, a type of OD is represented by a combination of expressions. Each expression verifies that an attribute in the environment belongs to an interval or a list of possible values.

Note, we could use an expression/condition pair like ”reject Expression when Condition” or ”if Condition reject Expression” but this is equivalent in terms of logic to ”reject Expression and Condition”. In an ODD statement, the starting keyword ”reject” means ”if Expression then out-of-ODD”. Thus, adding a condition is equivalent to extending the expression with ”and condition”: ”if Condition and Expression then out-of-ODD”.

The ”low-level” ODD description is the same as the ”high-level”, except it is organized as a JSON structure. Each statement is composed of a status (reject or accept), and one or many expressions. The expressions are organized hierarchically as an abstract syntax tree (AST). This is a structure easy to evaluate during execution, used to save the operators to be applied between expressions and their priorities (see Section IV-D for the evaluation). Each expression contains an OD attribute name, and the value, interval, fuzzy interval, or list to compare the OD attribute value to during execution.

C. ODD attributes taxonomy

The ODD attributes taxonomy is a tree-like structure allowing to define the different attributes used in ODD description and OD representations. This is the first step in creating a semantic relationship between the named elements of the environment and their machine representation. The second step is to have functions to observe/measure the said attribute as detailed in Section V-A. In practice, this is a YAML file organizing OD attributes and their metadata in a tree structure. It is based on multiple works, including existing taxonomies [16], [17], [18], [19]. Each attribute is unique, and can be identified by its path in the taxonomy. When this is unambiguous, the name of the attribute or the last elements of the path can simply be used. For example, `visibility` is the shorthand for:

```
att.environmental_conditions.visibility
```

and `rain.intensity` for:

```
att.environmental_conditions.weather.\
   precipitation.rain.intensity
```

Each attribute is enriched with different types of information. Firstly, descriptive information such as the name, description, and metric (for numerical attributes) is used to define as precisely as possible what the attribute represents and how it should be measured. For example, the difference can be made between the rain intensity detected directly by a rain sensor and that received by weather information.

Next comes the type of attribute. This can be numerical values (visibility, distances, etc), single-choice groups (type of lane markings, road, etc), multiple-choice groups (type of road users in sight, etc), and Booleans (pedestrian interaction, etc). In this paper, only numeric attributes will be addressed. However, most of the techniques presented will also work with the other attribute types. Numeric attributes will have two other information items. Firstly, the maximum range that the value can take (e.g. $[0,\infty]$ for a distance). Secondly, a list of categories, describing common presets for some attributes. For example, "visibility in [low]" will be equivalent to "visibility in [(244,805)]". The advantage of this approach is that the high-level ODD descriptions are simplified, improving usability while maintaining the benefits of numeric representations, such as order (poor $<$ low $<$ moderate $<$ good).

IV. ODD MONITORING

As mentioned in Section III-B, an ODD is described as a set of statements, each of which adds a constraint (in the case of a permissive description) or a valid OD (in the case of a restrictive description).

Each statement is composed of expressions, separated by Boolean operators. The value of an expression can be evaluated using a membership function, which will depend on the nature of the value and the interval.

A. Membership function definition

The membership value μ_{att} of an expression, represents the degree to which the value $x \in \mathbb{R}$ of an attribute att belongs to a union of intervals $I_{att} = \bigcup_{i=1}^{n}[a_i, b_i]$. For numerical attributes representing measurements with no uncertainties, the membership function $\mu_{att} : \mathbb{R} \to \{0,1\}$ is defined by:

$$\forall x \in \mathbb{R}, \begin{cases} \mu_{att}(x) = 1 & \text{if } x \in I_{att} \\ \mu_{att}(x) = 0 & \text{else.} \end{cases} \quad (1)$$

B. Membership function for uncertain observations

The particularity of live information is that it always comes with a level of uncertainty linked to its measurement. Here, uncertain values will be represented as a normal distribution. To determine how much the value lies within an interval, the sum of the areas under the curve is computed for each sub-interval within it. Let the function f_{att} be the probability density function of the Normal distribution representing the value x of an attribute att. The membership function $\mu_{att} : \mathbb{R}^2 \to [0,1]$ is defined by:

$$\mu_{att}(x) = \sum_{i=0}^{n} \int_{a_i}^{b_i} f_{att}(x)\ dx \quad (2)$$

C. Membership function with fuzzy intervals

To represent the vagueness of the real world, we introduce the possibility of defining fuzzy intervals in the ODD description. In classical set theory, an element either belongs or does not belong to a set. In fuzzy set theory [24], an element can belong to a set with a certain membership degree between 0 and 1. This membership degree is assigned using a membership function. Instead of a classical interval like $[a,b]$, it would be a trapezoidal fuzzy interval characterized by the quadruple $[a_s, a_c, b_c, b_s]$ of real numbers [25]. Here, only trapezoidal fuzzy intervals are used, but this could extend to any kind of fuzzy interval (Figure 3).

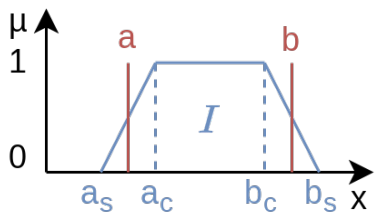

Fig. 3. Fuzzy interval. a_c and b_c are the core of the interval, inside which the membership value is 1. a_s and b_s are the support of the interval, outside which the membership value is 0. $\mu_I : \mathbb{R} \to [0,1]$ is the membership function of a value x to a fuzzy interval $I = [a_s, a_c, b_c, b_s]$.

Boolean	Fuzzy	Symbol
NOT(x)	1 - x	$\neg x$
AND(x,y)	MIN(x,y)	$x \wedge y$
OR(x,y)	MAX(x,y)	$x \vee y$

TABLE I

FUZZY LOGIC OPERATORS SORTED BY PRECEDENCE. $\neg$ HAS PRIORITY OVER $\wedge$ WHICH HAS PRIORITY OVER $\vee$.

For measurement values without uncertainty, the membership function $\mu_{att} : \mathbb{R} \to [0,1]$ of the value x of an attribute att to an interval $I_{att} = \bigcup_{i=1}^{n}[a_{s_i}, a_{c_i}, b_{c_i}, b_{s_i}]$ is defined by :

$$\mu_{att}(x) = \mu_{I_{att}}(x) = max(\mu_I(x), \forall I \in I_{att}) \quad (3)$$

For uncertain measurement values, the membership function $\mu_{att} : \mathbb{R}^2 \to [0,1]$ is defined by :

$$\mu_{att}(x) = \sum_{i=0}^{n} \int_{a_{s_i}}^{b_{s_i}} \mu_{I_{att}}(x).f_{att}(x)\ dx \quad (4)$$

D. Uncertain statement evaluation using fuzzy logic

Using the statement's expressions values just computed, the complete statement can be evaluated. As the measures feeding the expressions are uncertain, the resulting membership value will be between 0 and 1 (i.e. not classic booleans). To combine these values using Boolean operators, fuzzy logic can be used. In fuzzy logic, values, instead of being true or false, are real numbers between 0 and 1, representing degrees of truth. This is an extension of classical logic, allowing the propagation of measurement uncertainty and the fuzziness of intervals to the result. See table I for the fuzzy logic operators used. The various expressions of the statement are organized in an abstract syntax tree. This is a tree representing the structure of the logical relationships between the expressions, defined in the high-level language using operators and parentheses. For real-time evaluation, once the membership values of the expressions have been computed, they are combined, taking into account this structure and the precedence of logic operations.

As permissive descriptions have intervals describing values out of the ODD, an additional negation is applied on "reject" statements. This way, the ODD membership associated

with any statement uniformly represents how much an OD is in an ODD. The final ODD membership is a conjunction of all the statement values:

$$\left.\begin{array}{l} reject\ A \\ reject\ B \end{array}\right\} reject\ A\ and\ B$$

E. Evaluation smoothing

Although the measurements used to find attribute values are generally filtered (e.g. Kalman filter, etc.), the values obtained may still fluctuate too much. This can lead the monitoring system to be unstable, often creating brief "out of ODD" alerts. To smooth the membership value and make it independent of its update frequency, a sliding window average is used, based on the measurement time. Considering an arrangement $Q = ((t_0, \mu(t_0)), \ldots, (t_n, \mu(t_n))$ (like a queue) composed of n pairs $(t, \mu(t))$. $\mu(t)$ is a membership value and t its recorded time. t_0 is the oldest recorded time. The arrangement average is the sum of the memberships, weighted by the time between each record, divided by the total time between the first and the last record.

$$\mu_{avg} = \frac{1}{t_n - t_0} \sum_{i=1}^{n} (t_i - t_{i-1}) \times \mu(t_i) \tag{5}$$

The downside is that longer window length will increase the "reaction" time to events. The advantage is that it provides a time threshold, thereby preventing the generation of alerts for events that are shorter than this threshold. It adds a temporal aspect to the description, "it takes N seconds before leaving the ODD". Thus, the length of the window should be adapted to the fluctuating nature of the attributes. A slow-changing, but highly uncertain attribute should have a longer time window, while a fast-changing less uncertain attribute a short time window.

F. Time to exit the ODD (TTE)

One application of the membership computation is to use the membership value of the predicted attributes to determine how long the system will take to exit the ODD.

The notion of time to exit the ODD (TTE) is introduced, representing the estimated time for the system's OD to leave the ODD. The TTE value simply corresponds to the minimum time before a membership falls below a threshold. This low threshold represents, under which membership values the system is considered "out-of-ODD". This threshold can be regarded as a hyperparameter that determines the prudence of the TTE estimation. Values close to 0 indicate that a system is only considered "out of the ODD" when it is almost certainly out. Inversely, values close to one correspond to a conservative estimation of the TTE. In the same way that the Time To Collision (TTC) can be used to make driving decisions , the TTE can be used as an input to make meta-decisions regarding when to use a given system.

V. RESULTS

A. Experimental setup

The method presented has been tested in simulation using Carla Simulator [26] which will serve as the environment for the system (Figure 1). Since the measurements made directly in Carla are perfect, uncertainty will be added to the measurements. If we define x as the true value, the measured uncertain value $\tilde{x}$ can be expressed as:

$$\tilde{x} \sim \mathcal{N}(x + \varepsilon, \sigma_u^2) \tag{6}$$

Where $\varepsilon \sim \mathcal{N}(\mu_e, \sigma_e^2)$ is the measurement error, which is normally distributed with variance σ_e^2 and bias μ_e. It represents the discrepancy between reality and the measured value, i.e. random error and systematic error. σ_u^2 is the measure uncertainty, the expected error of the measure. This is often estimated using the statistical dispersion of measured values against a ground truth.

For the various use cases presented below, the error, uncertainty are available in the table II.

Parameters	ego loc.	bus loc.	vis.
Measurement variance σ_e^2	0.5	0.5	10
Measurement bias μ_e	0.5	0.5	0
Measurement uncertainty σ_u^2	2	2	10

TABLE II

ERROR AND UNCERTAINTY VALUES FOR THE ATTRIBUTES USED IN THE TWO USE CASES. THE ATTRIBUTES ARE THE EGO LOCATION, BUS STOP LOCATION, AND VISIBILITY

B. Statement evaluation from uncertain observations

To illustrate the statement evaluation from uncertain observations, let's take a simple use case in which the ego vehicle moves along a road approaching a bus stop. The ODD rule used by monitoring is, "The distance to the closest on-lane bus stop cannot be less than 20 m".

```
reject bus_stop_on_lane_distance in [(0,20)]
```

Figure 4 shows membership variation due to the measurement error from both ego and bus stop location being propagated to the computed distance. With the time-based sliding windows (of 0.5 second) smoothing the result, and because the ego speed is almost constant, the curve is similar to a sigmoid from a cumulative distribution function. This is expected, as the membership values are computed from the area under the curve of the distance uncertainty distribution, and the distance decreases at an almost constant rate. The average membership reaches a value of less than 0.5 about 0.2 seconds after the ground truth. This threshold corresponds to an estimate similar to the one that would be obtained by using only the mean of the measured distribution. A confident threshold of 0.05 can also be considered. The system will be "out-of-ODD" 1 s after the ground truth but with greater confidence. Inversely, conservative thresholds greater than 0.5 will give early "out-of-ODD" signals. Regarding the standard membership, for both the 0.5 and 0.05 thresholds, two "out-of-ODD" signals happened, while only one with the averaged membership, thus limiting the number of "out-of-ODD" alerts.

To go further, the matching of a membership curve to the membership ground truth can be seen as an optimization task.

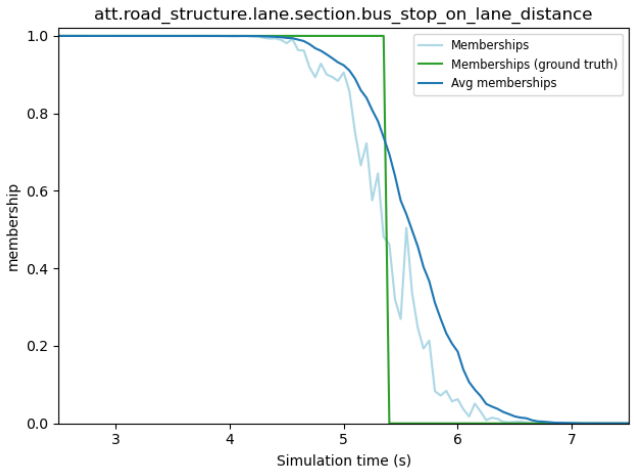

Fig. 4. Value of the OD membership to the ODD over time. `Memberships` (light blue) is the membership evaluated from the uncertain measurement. `Avg memberships` (blue) is the same membership, averaged over the last 0.5 s. `Memberships (ground truth)` (green) is the membership evaluated from the real values.

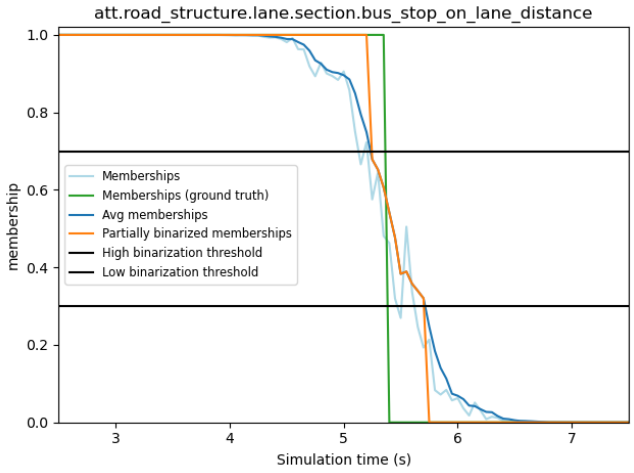

Fig. 5. Value of the OD membership to the ODD over time. Similar to Figure 4, except that the sliding windows length has been reduced to 0.2 seconds. The `Partially binarized memberships` (orange) are membership values equal to 1 when the average membership value is above a high threshold (here 0.7), and equal to 0 when below a low threshold (0.3).

Figure 5, shows that by tuning the sliding windows length and by partially re-binarizing membership values close to 0 and 1, the curve difference with the membership ground truth can be reduced.

Using a trapezoidal fuzzy interval, a similar ODD statement could be:

```
reject bus_stop_on_lane_distance in [(0,0,15,25)]
```

The second interval bound (15,25) means that, without considering uncertainty, the membership value will start to decrease 25 meters before the bus stop and reach 0 once closer than 15 meters. It can help to represent ODD description that can be subjective by nature. For example,

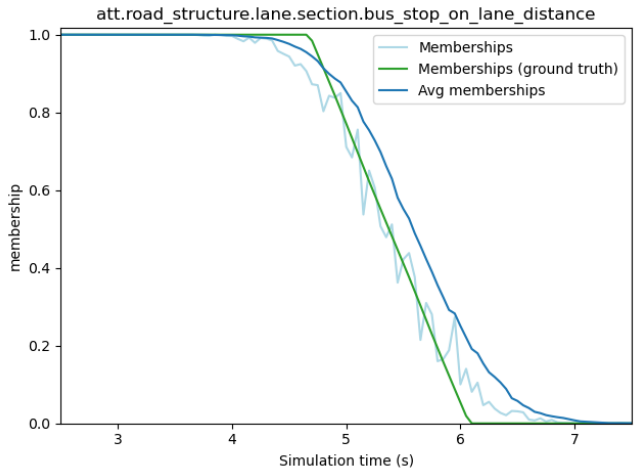

Fig. 6. Value of the OD membership to the ODD over time. Similar to Figure 4, except that the rejected interval for the distance with the bus stop has a fuzzy bound between 15 and 25 meters instead of just 20 meters.

the second interval bound (15,25) could be used to represent optimistic and pessimistic values. Combined with an adapted threshold, it can represent the prudence of the ODD exit estimation. A threshold close to 1 for a pessimistic estimation and close to 0 for an optimistic one. Thus, the time window where the ODD's membership decreases from 1 to 0 is wider (Figure 6). In this context, using fuzzy intervals is convenient to represent subjective or imprecise numeric ranges.

We saw here that in the presence of uncertain measured values, the membership computation provides a degree of truth for ODD exit, dependent on the localization uncertainty. The smoothed membership computation allows for a valid estimation of the true ODD membership while reducing the number of "out-of-ODD" alerts. Moreover, fuzzy intervals can be used to define an ODD description closer to the vagueness of the world. This result depends on two hyper-parameters, the sliding windows average, and the ODD exit threshold.

C. Time to exit the ODD (TTE)

Using the same use case, based on the predicted future distances to the bus stop, the time to exit the ODD (TTE) can be estimated. Thus, it depends on the expected trajectory of the vehicle and the uncertain location of the bus stop. The TTE is the time until the first membership value goes below a given threshold. In Figure 7 the threshold is 0.05 giving a TTE of 3.8 seconds. The prediction has a stair shape because it is dependent on the vehicle location prediction from the local planner, which is discreet. A meta-decision system could decide to deactivate the driving system and go back to manual driving when the TTE is low. For level 3 ADS, this is called a transition-of-control (ToC) fallback. Then, it can decide to trigger a minimal risk maneuver (MRM) like an emergency stop when the TTE is too low, or the ODD is left. In Figure 7, as an example, the transition-of-control TTE threshold is 2 seconds (yellow area), and the MRM TTE threshold is 0 seconds (red area). Here, the times are given

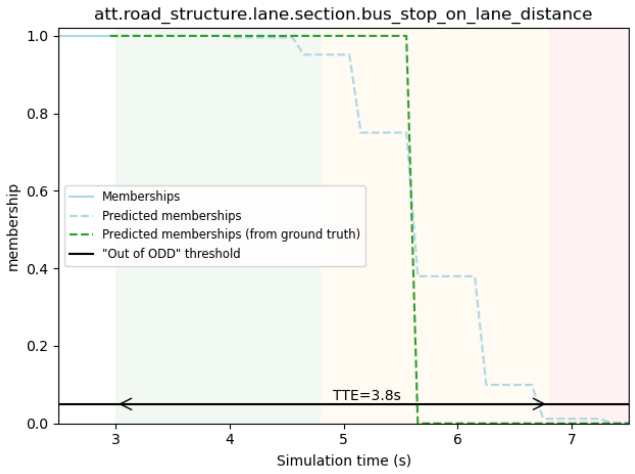

Fig. 7. Past value (light blue solid line) and predicted value (light blue dashed line) of the OD membership to the ODD. Predicted memberships are the memberships computed from the predicted measured values (here the distance to the bus top). The predicted membership from ground truth (green dashed line) is the membership value of the predicted attribute values without uncertainty or error (no bus stop and ego localization uncertainty). The TTE is the time until the first membership value goes below the "Out of ODD" threshold. An example of meta-decision based on the TTE is represented by the background colors: white is the past, in green everything is fine, in yellow (TTE=2s) the driving system deactivation is requested, in red (TTE=0s) a minimal risk maneuver (MRM) is requested.

as an example to better illustrate the application. In practice, the expected time to hand over to the driver would likely be around 30 seconds. While this is not a problem when based on a priori information (road network, etc), it is not always possible to obtain this level of anticipation with dynamic information (presence of pedestrians, distances, etc).

D. Multi expression statements

The second use case (Figure 8) is similar to a past ADAS accident [27]. In this accident, the perception system failed to recognize a stopped emergency vehicle in time. The emergency vehicle had emergency lights, during a foggy night, and the ego vehicle was cruising at high speed (87 kph). The emergency vehicle was finally detected 34 meters before impact. The perception failure was probably caused by a combination of factors, the flashing emergency lights, low luminosity and visibility being the main factors. In the use case we will be using, the difference is that only visibility is used as an aggravating factor for the perception capabilities. Visibility also progressively worsens as the vehicle enters the area.

Knowing that the perception system is unable to correctly detect emergency vehicles in low visibility conditions, it would be possible to use the presence of other emergency vehicles, which would not otherwise be considered obstacles, as clues indicating an unsuitable domain of operation. This can then be described as a situation out of the ODD as:

```
reject vehicles.in_sight in [emergency_vehicles]
and visibility <= low
```

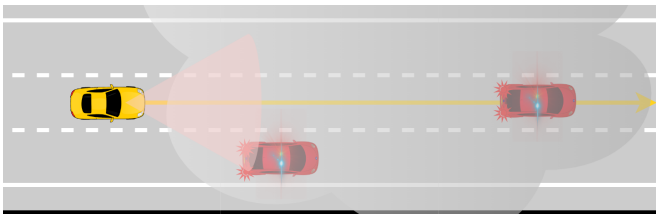

Fig. 8. Emergency vehicles use case. The ego vehicle (yellow) is driving straight past the first stopped emergency vehicle (red, on the side), and then toward a second emergency vehicle (red, in the lane) stopped on the road with emergency lights. In this use case, the combination of low visibility and the flashing lights of the emergency vehicles on the trajectory would be detected too late by a perception system not adapted for this type of situation. Here, the simulated detection range was 20 meters.

In Figure 9, as the vehicle enters a low visibility area, the expression value associated with visibility (in dotted light blue) starts to increase. This is because, in a reject statement, intervals represent rejected values. Thus being in the interval decreases the statement membership. The yellow area represents when the visibility value becomes low according to the ground truth (solid green line). However, the statement's membership does not decrease because the expression associated with emergency vehicles being in sight is still 0. It is only when the system also has the first emergency vehicle in sight, that the ODD membership starts to rapidly decrease (red area).

As soon as the conjunction of both the low visibility and the sight of the side emergency vehicle is observed, the monitoring system detects that this situation is out of the ODD. Then, depending on the system's level of autonomy, a transition-of-control or MRM can be triggered to avoid risky situations. Of course, there will not always be an emergency vehicle on the side in this type of situation, but this can be generalized to any element that can be used as a clue for unsuitable operating conditions, like warning triangles, warning lights, etc.

With this example, we showed that ODD statements can be composed of multiple expressions, narrowing down the accepted or rejected situations to better correspond to the known limits of a system.

VI. CONCLUSION

The main objective of Operational Design Domain (ODD) description and monitoring is to define transparent minimum requirements for a system, in order to safeguard it against known adverse operational conditions for which it was not designed. In this paper, we proposed a method to monitor live numerical ODD attributes (like dynamic elements, weather, road users, etc.), that often have uncertainty in their measures. The proposed ODD monitoring system estimates the degree of membership of a system's uncertain observed operational domain (OD) to its ODD. This degree of membership, ranging from 0 to 1, is used to decide whether or not the system is within its ODD. For this purpose, a formal way to describe an ODD was presented. It uses a high-level domain specific language (DSL) as a natural, human-readable format, which can be converted into

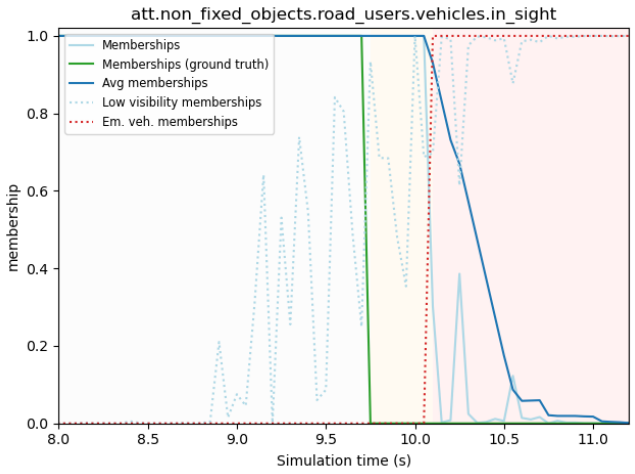

Fig. 9. Value of the OD membership to the ODD over time (solid lines). Value of the membership of the expressions composing the statement over time (dotted lines). `Low visibility memberships` (dotted light blue) are the values of the expression "low visibility". Values are close to 1 when the visibility is low. `Em. veh. memberships` (dotted red) are the values of the expression "emergency vehicle in sight". Values are close to 1 when emergency vehicles are in sight.

a lower-level structured description, interpretable at runtime. This description can be composed of multiple expressions, referencing driving environment attributes organized as a taxonomy. Fuzzy intervals were also included in the ODD description and monitoring as a tool to better represent the vagueness of real-world boundaries. Finally, two use cases were used to illustrate how ODD description and monitoring can be used to safeguard complex systems in simulation, and estimate the time to exit the ODD (TTE) from predicted observations.

In the future, we plan to test this approach with real driving systems in order to have more realistic attribute values and to identify relevant ODD descriptions for given systems.

References

[1] SAE International, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," 2021.

[2] M. Gyllenhammar, R. Johansson, F. Warg, D. Chen, H.-M. Heyn, M. Sanfridson, J. Söderberg, A. Thorsen, and S. Ursing, "Towards an Operational Design Domain That Supports the Safety Argumentation of an Automated Driving System," in *10th European Congress on Embedded Real Time Systems (ERTS 2020)*, 2020.

[3] T. Charmet, V. Cherfaoui, J. Ibanez-Guzman, and A. Armand, "Overview of the Operational Design Domain Monitoring for Safe Intelligent Vehicle Navigation," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 2023, pp. 5363–5370.

[4] D. Wittmann, C. Wang, and M. Lienkamp, "Definition and identification of system boundaries of highly automated driving," in *7. Tagung Fahrerassistenzsysteme*, 2015.

[5] I. Colwell, B. Phan, S. Saleem, R. Salay, and K. Czarnecki, "An Automated Vehicle Safety Concept Based on Runtime Restriction of the Operational Design Domain," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1910–1917.

[6] Z. Alsayed, P. Resende, and B. Bradai, "Operational Design Domain Monitoring at Runtime for 2D Laser-Based Localization algorithms," in *2021 20th International Conference on Advanced Robotics (ICAR)*, 2021, pp. 449–454.

[7] F. Woerter, A. Kreutz, A. Salvi, M. Dreiser, and G. Weiss, "Enhanced System Awareness as Basis for Resilience of Autonomous Vehicles," in *CARS 2021 6th International Workshop on Critical Automotive Applications: Robustness & Safety*, 2021.

[8] E. Schwalb, A. Richter, and D. Rohne, "Validating Autonomous Behaviors against Partially Specified Ambiguous Requirements," in *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2022, pp. 1342–1349.

[9] H. Torfah, C. Xie, S. Junges, M. Vazquez-Chanlatte, and S. A. Seshia, "Learning Monitorable Operational Design Domains for Assured Autonomy," in *Automated Technology for Verification and Analysis*, ser. Lecture Notes in Computer Science, A. Bouajjani, L. Holík, and Z. Wu, Eds. Springer International Publishing, 2022, pp. 3–22.

[10] G. Pappalardo, R. Caponetto, R. Varrica, and S. Cafiso, "Assessing the operational design domain of lane support system for automated vehicles in different weather and road conditions," *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 9, no. 4, pp. 631–644, 2022.

[11] C. Sun, Z. Deng, W. Chu, S. Li, and D. Cao, "Acclimatizing the Operational Design Domain for Autonomous Driving Systems," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 2, pp. 10–24, 2022.

[12] Iain Whiteside, "ASAM OpenXOntology ProposalWorkshop 5 Ontologies and ODDS at Five," 2020.

[13] D. Rohne, A. Richter, and E. Schwalb, "Implementing ODD as single point of knowledge to support the development of automated driving," in *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2022, pp. 1364–1370.

[14] E. Schwalb, P. Irvine, X. Zhang, S. Khastgir, and P. Jennings, "A Two-Level Abstraction ODD Definition Language: Part II," in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2021, pp. 1669–1676.

[15] ASAM, "OpenODD: Concept Paper," 2021. [Online]. Available: https://www.asam.net/standards/detail/openodd

[16] E. Thorn, S. C. Kimmel, and M. Chaka, "A Framework for Automated Driving System Testable Cases and Scenarios," National Highway Traffic Safety Administration (NHTSA), Tech. Rep. DOT HS 812 623, 2018.

[17] BSI, *PAS 1883:2020 Operational Design Domain (ODD) Taxonomy for an Automated Driving System (ADS) - Specification.*, 2020.

[18] SAE Industry Technologies Consortia (ITC), "AVSC Best Practice for Describing an Operational Design Domain: Conceptual Framework and Lexicon," 2020.

[19] Krzysztof Czarnecki, "Operational Design Domain for Automated Driving Systems - Taxonomy of Basic Terms," Waterloo Intelligent Systems Engineering Lab (WISE), University of Waterloo, Tech. Rep., 2018.

[20] International Organization for Standardization (ISO), "ISO 34503:2023 : Road Vehicles — Test scenarios for automated driving systems — Specification for operational design domain," 2023.

[21] L. Mendiboure, M. L. Benzagouta, D. Gruyer, T. Sylla, M. Adedjouma, and A. Hedhli, "Operational Design Domain for Automated Driving Systems: Taxonomy Definition and Application," in *2023 IEEE Intelligent Vehicles Symposium (IV)*, 2023, pp. 1–6.

[22] M. A. Mehlhorn, A. Richter, and Y. A. W. Shardt, "Ruling the Operational Boundaries: A Survey on Operational Design Domains of Autonomous Driving Systems," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 2202–2213, 2023.

[23] R. Reiter, "A logic for default reasoning," *Artificial Intelligence*, vol. 13, no. 1, pp. 81–132, 1980.

[24] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[25] J. Fortin, D. Dubois, and H. Fargier, "Gradual Numbers and Their Application to Fuzzy Interval Analysis," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 2, pp. 388–402, 2008.

[26] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[27] Wall Street Journal, "Watch: Exclusive Tesla Footage Suggests Reasons for Autopilot Crashes," *WSJ*, Aug. 2023. [Online]. Available: https://www.wsj.com/video/series/in-depth-features/watch-exclusive-tesla-footage-suggests-reasons-for-autopilot-crashes/2FBEE1CA-56E1-4ACC-92B6-DED638B531CE