**Virtual Testing – Proposed Annexes**

**Annex I - Virtual Testing Toolchain Example**

Virtual Testing is introduced to reduce the burden of physical tests and effectively provide evidence on the ADS performance across the operational domain. However, no one simulation tool can be used to test all aspects of the ADS software, this is why manufacturers will exploit the attributes of various simulation tools to develop confidence in the safety of the full system.

Each virtual testing tool will have its own strengths and weakness based on the speed and cost of execution and the level of fidelity achieved. Typically lower fidelity tools are used to cover a vast number of scenarios to obtain a general understanding of the systems performance. Then it is possible to increase the level of fidelity within a subset of scenarios to validate the performance of the ADS in a statistically relevant number of realistic scenarios. A manufacturer’s virtual testing toolchain may consist of the following tools:

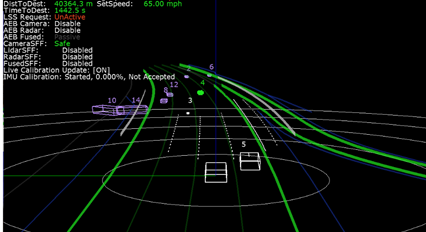
**Perception simulation**

Perception simulation can be used to train and validate the perception algorithms of the ADS software with physical accurate sensor models in combination with ground-truth data. This can be done in open-loop since the planning and control algorithms are bypassed.



**Planning & Control (P&C) simulation**

P&C simulation can be used to validate the control algorithms of the ADS software with basic sensor models. This can be done faster than real time so is an effective way to test the control system over a vast number of scenarios.



**Full AV Stack simulation (MIL, SIL or HIL)**

Full AV Stack simulation can accurately render sensor data streams that represent a wide range of environments and scenarios. The ADS software processes the simulated data as if it were coming from the sensors of a vehicle actually driving on the road and sends actuation commands back to the simulator. This allows engineers to test rare conditions, such as rainstorms, snowstorms, or sharp glare at different times of the day and night. Each scenario can be tested repeatedly, adjusting multiple variables such as road surfaces and surroundings, weather conditions, other traffic, and time of day.

HIL can be used to test the entire hardware component or ECU before the real vehicle is available and to test the interactions/ networks of the components within the virtual prototype e.g. conduct E/E failure test of hardware components.



**Vehicle in the Loop (VIL)**

VIL provides a validation environment for ready-to-drive vehicles in combination with a virtual environment simulation. It allows to execute complex and safety critical scenarios on vehicle level.

VIL on Test Beds:

VIL on test beds combines this with the advantages of a lab and focuses on flexibility in scenario generation and reproducibility of scenario execution. It allows additionally to test the real sensors and perception in the loop.



VIL on Test bed may consist of the following elements:

* Longitudinal dynamics: The longitudinal dynamics are emulated by the test bed. This can either be a chassis dynamometer or a wheel hub / powertrain test bed. High dynamic dynamometers in combination with a vehicle dynamics simulation allow the execution of various maneuvers and scenarios including high dynamic maneuvers at the limits (realistic wheel slip, etc.)
* Lateral dynamics: In case of lateral dynamics, including the steering is required, test beds can be extended by additional devices to allow steering. Ideally steering is not only allowed but also the resulting reaction forces are emulated properly to avoid error states and to ensure a proper operation together with the AV function
* Interface virtual environment simulation: Depending on the use case and the requirements, there are different possibilities: Object list injection (no sensor, no perception in the loop), raw data injection (no sensor but perception in the loop), over-the-air stimulation of the sensor (sensor and perception in the loop). Using the over-the-air stimulation, there are no modification on the vehicle required. Also, a mixed operation is possible.

VIL on proving grounds:

VIL on proving grounds focuses more on the interaction between the driver/passenger and the vehicle. In this configuration the real acceleration (longitudinal and lateral) of the vehicle can be experienced by the driver/passenger (difference to Vehicle-in-the-Loop at test beds). A judgment and rating by the real driver are possible.

VIL Test bed may consist of the following elements:

* Longitudinal dynamics: The real longitudinal dynamics are available
* Lateral dynamics: The real lateral dynamics are available
* Interface virtual environment simulation: Typically, the interface between the vehicle and the virtual environment is done via object list injection. Also, raw data injection is possible. Real sensors cannot be considered (with a few exceptions for very simple sensors like ultrasonic).

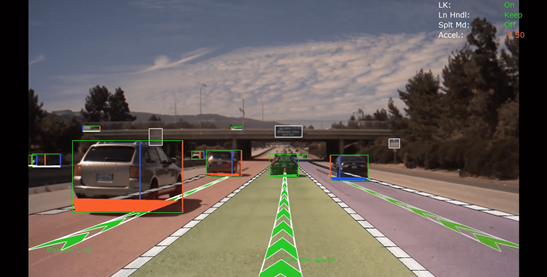
**Driver in the Loop (DIL)**

DIL virtual testing can be helpful to support the assessment of this category of functional requirement by analysing the interaction between the driver and the ADS in a safe and controlled environment.



**Software Reprocessing (SwR)**

SwR involves playing back previously recorded sensor data, rather than synthetic data, to the ADS software to accurately assess the perception performance in an open loop system.



Considering the categories of functional requirements currently being considered, virtual testing seems particularly relevant for assessing requirements related to:

* ADS should drive safely and ADS should manage safety critical situations. These are the requirements where virtual testing can play the most prominent role. MIL/SIL, HIL and VIL virtual testing can all be used to assess these requirements at different stages of vehicle verification and validation.
* ADS should interact safely with the user. DIL virtual testing can be helpful to support the assessment of this category of functional requirement by analysing the interaction between the driver and the ADS in a safe and controlled environment.
* ADS should safely manage failure modes and ADS should ensure a safe operational state. The use of virtual testing in these two categories is also very promising but would probably require further research work. SIL virtual testing could include simulated failures and maintenance requests. HIL and VIL virtual testing could be used to assess how the system would react to the occurrence of a real malfunctioning induced to the real system.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Functional Requirement | SIL | HIL | VIL | DIL | SwR |
| ADS should drive safely | Y | Y | Y | - | Y |
| ADS should interact safely with the user | Y | Y | Y | Y | - |
| ADS should manage safety-critical situations | Y | Y | Y | - | Y |
| ADS should safely manage failure modes | Y | Y | Y | -- | - |

The table below describes all available test environments. The main difference in these test environments is in the application of virtual and real stimuli and in the items being tested.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Virtual Testing Tool | Software | Hardware | Vehicle | Driver | Environment |
| Perception | Real | Virtual | Virtual | Virtual | Virtual |
| Planning & Control | Real | Virtual | Virtual | Virtual | Virtual |
| Full AV Stack (SIL) | Real | Virtual | Virtual | Virtual | Virtual |
| Full AV Stack (HIL) | Real | Real | Virtual | Virtual | Virtual |
| Vehicle in the Loop | Real | Real | Real | Virtual | Virtual |
| Driver in the Loop | Virtual | Virtual | Virtual | Real | Virtual |
| Software Reprocessing | Real | Virtual | None | None | Real |
| Proving Ground | Real | Real | Real | Real | None |
| Real World Test | Real | Real | Real | Real | Real |

**Annex II – Correlation methodologies**

The validation of a virtual testing toolchain shall be based on the quantitative evaluation of a set of KPIs with respect to the real-world data. The assessment returns a measure of correlation which has to be checked against a prescribed correlation threshold. It is recognized that no method for correlating sim-real data is suitable for all virtual testing tools, it is therefore the responsibility of the ADS manufacturer to justify the chosen correlation methodologies.

The computation of the correlation is carried out comparing either time-series or probability distributions depending on the data availability and the virtual testing setup. Deterministic virtual testing environments such as MIL and SIL will originate deterministic results with no possibility of assessing the confidence intervals. Similarly, real-world testing leveraging on a single execution per each test does not allow assessing confidence intervals. Thus, when a MIL testing environment is compared to a single execution for validation purposes, only time-series comparison analysis is possible.

On the other side, a HIL or VIL testing environment is subject to a certain degree of stochasticity, which implies that multiple repetitions will originate a statistical distribution of the results. An analogous result is obtained via the execution of several repetitions for a given proving ground scenario. This way of proceeding allows carrying out statistical testing on the collected data distributions.

**Graphical comparison**

Graphical comparisons provide a first validation step which displays the goodness of the simulation model. Nonetheless, the subjectivity inherent to the qualitative nature of the assessment implies that graphical comparisons are only suitable to support the credibility of the developed toolchain. A proper validation methodology shall be based on the quantitative methods described below.

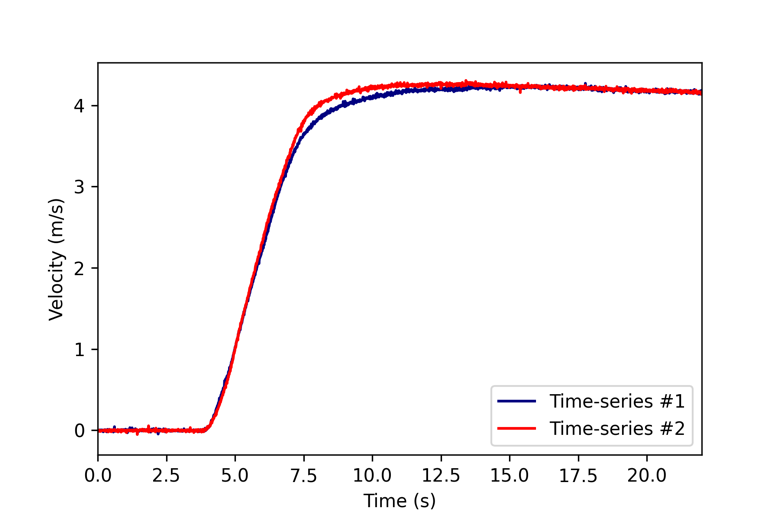
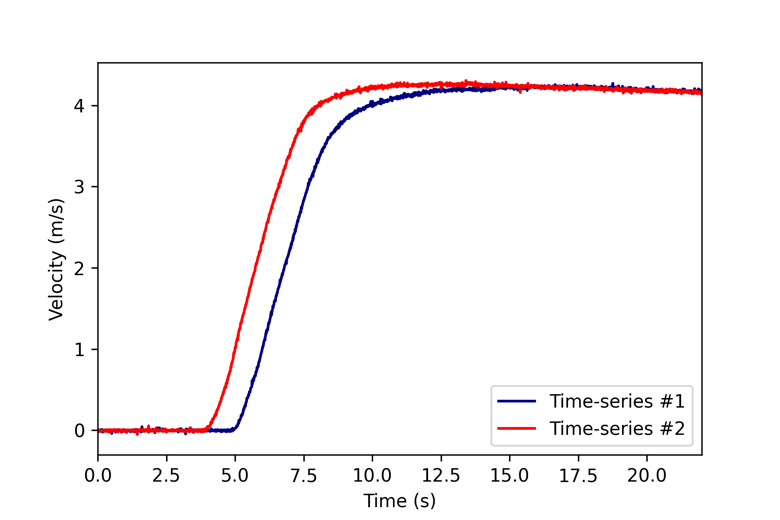
**Scalar data comparison**

Scalar data comparisons are useful tools to compare significant values of a signal. When only the pick values of a signal is relevant (e.g. the maximum yaw-rate during an emergency obstacle avoidance maneuver) for the sake of validation, the Relative Error Criterion (REC) [1] difference amplitude criterion is a suitable metrics

**Time-series comparison**

Despite being the first step into the quantitative evaluation, scalar data provide limited information about the agreement of the signals. The study of time-series affords to investigate the correlation of the simulation-generated evidence with the real-world data to a greater extent.

Several tools exist to quantify the distance between time-series. Before any attempt of comparison can be pursued, the time-series have to be synchronized and resampled based on the lowest frequency between real-world and the simulated data. A widespread solution for the synchronization is to adopt the Time-of-Arrival (ToA) criterion. ToA implies the definition of a reference starting time for the signals which is derived from the first time the signal reached a pre-defined amplitude.



Once opportunely synchronized and resampled, the time series can be analyzed according to a distance function. Distance estimation is typically carried out by applying some norm function to the vector of residuals. For instance, the norm (Euclidean distance) reads as:

where is the total list of samples. The normalization of the norm over the total number of samples yields the Root Mean Square Error (RMSE):

Alternative norms can be used to quantify the discrepancies between the time-series, which are susceptible to different features or error signals. For instance, the norm returns the maximum absolute value of the error

Recently developed metrics allow separating the contribution of *phase* error (thus the shape of the time-series) to the contribution of the *magnitude* error between the signals, thus providing more insights on possible inconsistencies affecting the model. A recent report published by Sandia [2] investigates such techniques. In particular, the Sprague-Geers [3] metric is presented therein. The same criterion is also adopted to validate virtual models for seats within the field of aviation [4]. The metric is based on establishing the integral distance between the signals

and the phase difference

combined into the total error

An alternative analysis that can be carried is establishing the *correlation* between the signals. Several tools to calculate the correlation have been proposed in the literature [5]. Among them, a commonly adopted tool is the Pearson correlation

Values of close 1 suggest good agreement between the signals, whereas correlation degrades approaching 0.

**Statistical testing**

Statistical testing is concerned with verifying whether the null hypothesis, *i.e.,:* “the model is an accurate representation of the real-world phenomena,” cannot be rejected given the evidence generated by the simulation. Statistical testing is particularly useful when dealing with non-deterministic virtual testing environments or multiple repetitions of the same driving scenario on the proving ground.

A common statistical test is the well-known T-test which analyzes whether two distributions have a significatively different mean. T-test can be performed on both one-sample or two-sample datasets. A one-sample case study involves determining whether the mean of a population () is statistically different from a given reference mean (. The “”-value can be calculated as

where is the standard deviation of the sample. One can reject the null-hypothesis is the value exceeds the critical value resulting from the sample size and significance level.

A typical example for the one-sample T-test is investigating whether the experimental mean of a quantity differs significatively from the distribution of the same quantity deriving from multiple repetitions on a HIL/VIL setup. Similarly, comparing multiple repetitions on a proving ground with the evidence derived from a deterministic environment originates a one-sample exercise. Conversely, two-sample T-test is found when two distributions are compared. The comparison of more than two distributions can be carried out by exploiting ANOVA.

While the T-test is mainly concerned with studying the mean of distributions, alternative tests exist which do not make assumptions on input data normality. For instance, the Kolmogorov-Smirnov test evaluates the maximum vertical distance in the Cumulative Distribution Functions (CDFs) of the input distributions.

**Annex III – Validation examples**

This section presents modeling and validation approaches for three models’ classes: lane and camera, RADARs and LiDARs, and vehicle dynamical models. The first two paragraphs are concerned with describing the realization of virtual perception modules, which, together with the virtual vehicle models described in the last paragraph, enable interfacing the ADS with the simulation environment. The contribution includes examples of metrics and relevant KPIs which afford the determination of the fidelity level returned by the virtual solution.

The focus of the current discussion is on the simulation models *per se* (intrinsic properties). Nonetheless, a complete assessment of the fidelity level provided by the integrated virtual testing toolchain might also include investigating the *sensor-grade* realism offered by the virtual environment. That is how the simulation engine is capable of faithfully rendering real-world characteristics that are relevant for the perception systems but that might not necessarily match human vision peculiarities.

1. **Lane Model Validation**

Lane model validation is considered to provide a practical example on how the validation is performed as a part of the credibility assessment. Accurate representation of lane models are required for perception algorithm used for most lateral support systems e.g. lane keep assist, lane centering, lane change assist etc. In order to demonstrate that the lane models are fit for purpose we have used the processes defined in the credibility assessment. Vehicle dynamics is not considered during this process because the chassis dynamic will have negligible impact on the ability to detect the lane markings. The process consist of the following elements:

* Subsystem – camera model
* Sensor System – camera model with virtual lane markings.
* Integrated System – Lane detection algorithms

**Camera Model Validation**

Simulation needs to provide accurate image (intrinsic property) from the correct position (extrinsic property) for all cameras for a given scene. Specific intrinsic camera-related phenomena that should be considered during the validation include:

* Lens distortion: optical aberration due to projection
* Vignette: darkening of the screen border.
* Grain jitter: white noise injection.
* Bloom: presence of fringes around bright areas
* Auto exposure: image gamma adaption to darker or brighter areas.
* Lens flares: reflection of bright objects on the lens.
* Depth of field: blurring of objects near or very far away of the camera.
* Exposure time: shutter opening duration

Below is a non-exhaustive list of tools that can be used to support the camera model validation.

|  |  |  |
| --- | --- | --- |
| Tool | Image | Purpose |
| Macbeth Color chart Test |  | * To determine the camera color space of the camera * To determine the parameters for camera noise modelling * To learn about the exposure characteristics |
| OECF chart Tests |  | * Is designed for evaluating the opto-electronic-conversion-function of a camera. |
| SFR Chart |  | * To measure sharpness, contrast and lens effects |
| Lens Flare Characterization |  | * To differentiate the static and the dynamic components (dark shot noise) a video has to be recoded * To determine the lens characteristic for lens flares and ghosting artifacts |
| FTheta Calibration |  | * At every position, tilt the checkerboard target both horizontally and vertically up to 45 degrees * To determine the ftheta polynomial and to compare it with a more precise lens measurement |

April Tags is a visual fiducial system, useful tool to supporting the validation of the extrinsic camera related properties. The tags provide a means of identification and 3D positioning, even in low visibility conditions. The tags act like barcodes, storing a small amount of information (tag ID), while also enabling simple and accurate 6D (x, y, z, roll, pitch, yaw) pose estimation of the tag.

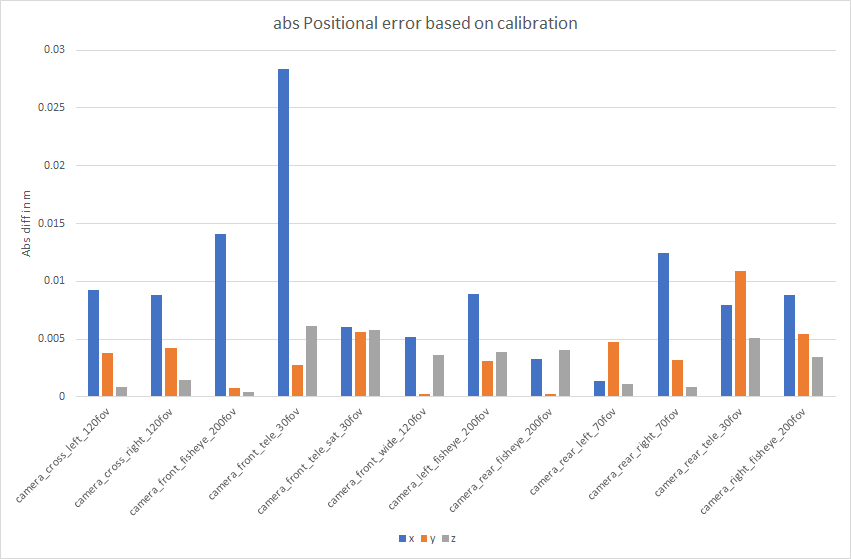
Camera front wide 120fov:

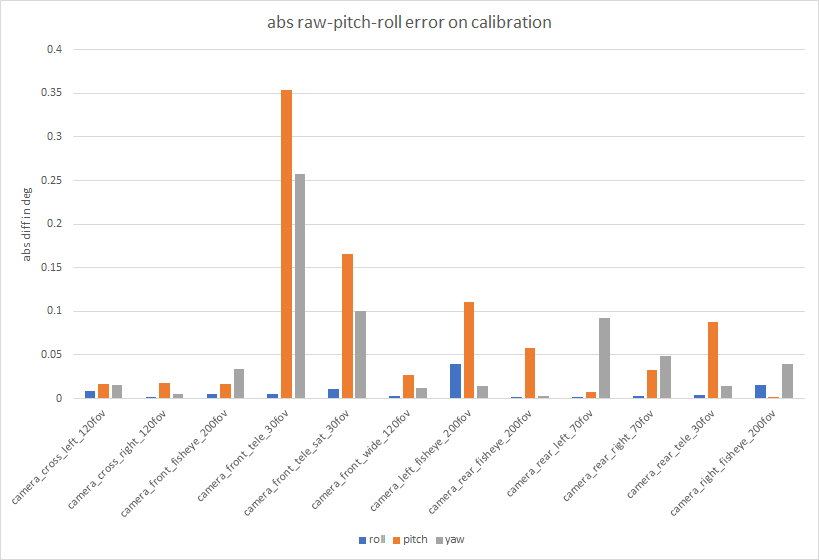
|  |  |
| --- | --- |
| Simulation | Real |
|  |  |

Camera\_left\_fisheye\_200fov

|  |  |
| --- | --- |
| Simulation | Real |
|  |  |

The April tag chart positions and orientations are well constrained in the scene as they are visible from multiple cameras. Thresholds may be set on error derived from difference in absolute position / angle of the April Tags





**Sensor System Validation**

The purpose of the sensor system validation is to demonstrate that camera models provide accurate results in the virtual environment which the system under test will be operated in. Pre-defined KPIs can be used to determine performance of the virtual sensor system. For the purpose of lane models the contrast ratio between lane marking and road surface is used to demonstrate the performance of the sensor system in both physical and virtual environments.

A simple framework for dividing sensor performance into several equivalence classes is shown here, as an example. The method relies on efficiently dividing the equivalence classes of the *conditions* that have a significant effect on a sensor performance metric, in this case, the brightness contrast ratio between the lane marking and the road surface. A requirement pattern can be formed that combines performance achievement with certain environmental or scenario-specific conditions.

A generic requirement pattern can be considered, as follows:

The {KPI} shall be {greater than} {KPI Threshold} if {Conditions Exist}.

The requirement pattern can be repeated with different conditions, as needed, in order to 1) fully cover all external conditions, including the extreme ends, and 2) define the boundary values at which performance requirements may change depending on the conditions, for example, relaxing the false-positive detection rate of a lane boundary if it is snowing. If this requirement pattern is well-defined across all possible conditions, independently verified, and has commitment from the developers to fulfill the requirements, then the problem of “functional insufficiencies” in sensor performance will likely be reduced or eliminated altogether.

A method of division of the conditional classes follows this simple structure, as an example:

* Class 1: Nominal conditions - These are the ideal, best-case conditions.
* Class 2: Average conditions - These are the expected, real-world conditions that likely require significant development effort compared to Class 1 conditions, e.g. inner quartile range.
* Class 3: Worst-acceptable conditions - These are the worst conditions in which some level of performance will be guaranteed, e.g. 95’th percentile range. These likely require a tradeoff between the minimum required performance level and the remaining development effort. Beyond this class, no performance requirements are obligated. (Note, this can be tailored, as needed)

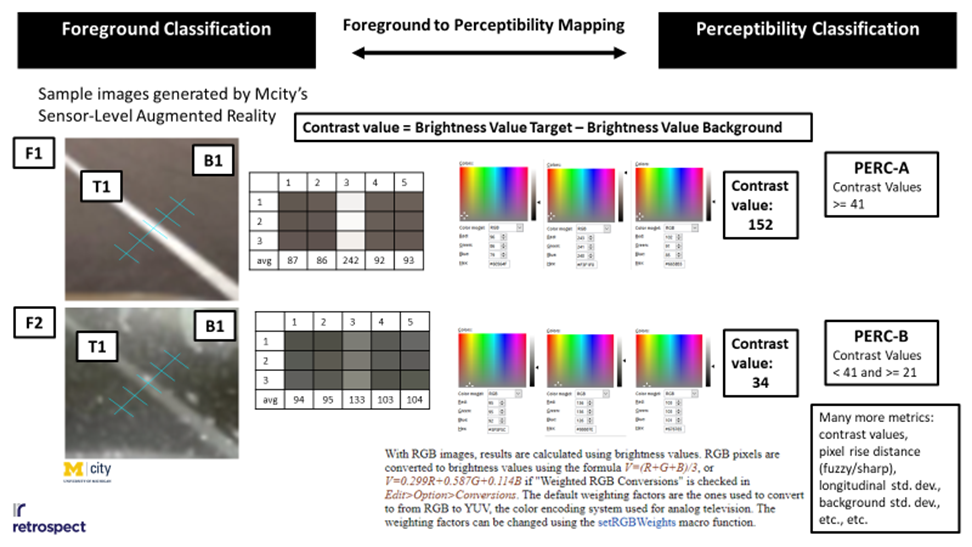
Finally, the conditions themselves may need to be separated into parameters according to the dependence or independence from each other into a minimal parameter set which adequately captures the environmental and scenario-specific conditions. For each sensing modality that has been considered so far, which includes: camera (visible light), radar, LiDAR, ultrasonic, and infrared cameras, the following generic sensor model has shown to be repeatable and useful in analyzing all environmental conditions for all sensing modes. It is broken down into three distinct parameters: Foreground, Target, and Background.

Diagram

Description automatically generated

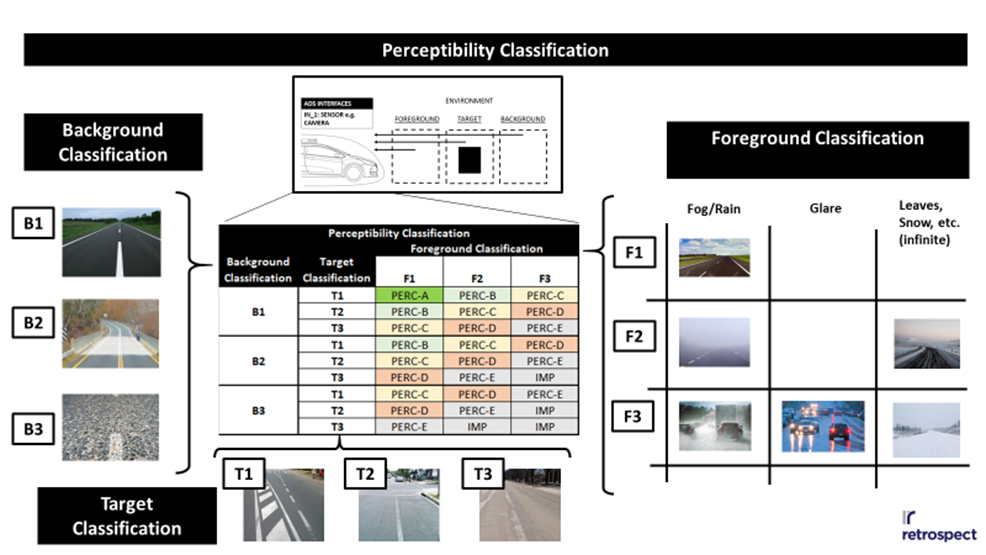
In the example of a camera-based system doing lane detection, the “Target” in this case would be the lane, itself. Many attributes may need to be developed to fully capture all the desired attributes of the Target, such as color, position, curvature, dash type, sharpness (or blurriness), etc. In this case, the attribute of interest is the contrast ratio of the lane with respect to the road surface. The road surface would be the “Background” in the sensing model. The “Foreground” could be anything between the sensor and the Target, such as fog, rain, or clear air, as well as, debris, objects, accumulated snow, etc.

In the following figure, a sample image from a sensor-level augmented reality demonstration is used to show how the Foreground, in this case snow, can alter the contrast value of the lane (Target) with respect to the road (Background). This highlights the usefulness of the conditional classification, as it can provide a consistent interface point between the infinite variations and combinations of environmental conditions and the finite set of the performance requirements that the developers must commit to deliver.



Internal or external classifications of the overall performance capability may (or may not) be useful to consider. For lane models, each parameter and their associated range of variables (Target, Foreground, and Background) should be considered during this phase. The sensor system should be qualified using known KPIs, such as very deterministic static scenarios, at first. This will allow the system to be validated against a measurable KPI. After that it can be extended to varying weather conditions etc. A test matrix can then be established that considers the variation of input parameters. Large variation in real and simulated results provide evidence where there may be limitations in the tool. Any sensor performance limitations should be noted during the assessment to put restrictions on what data can be generated to support the assessment of the ADS.

As performance limitations are encountered due to uncontrollable environmental conditions, the designers may either make reductions to a minimum performance level, as discussed above, or they may be able to strategically shift the “Target” in order to detect the environmental conditions, themselves. Considering the snowy example above, in light snow conditions the “Target” could be the lane, itself. However, in heavy snow conditions, the “Target” may be the heavy snow, itself, that is to say, the lane detection camera must detect heavy snow.



It is up to the designers to determine what is useful for the overall system goals and the given technical capabilities, but the intent with the sensor system validation approach is to show how the designers and testers may fully specify performance and safety requirements in their development contracts, and show evidence of the fulfillment of their contracts. This can be done with little risk of finding out late in development that they are unable to meet the expected performance. Even if this should happen, the easiest solution may simply be to renegotiate the performance levels and communicate those to all stakeholders. The greatest source of risk would be in failing to write the requirements, in the first place, and plan for a “wait and see” approach with respect to sensor performance.

**Integrated System Validation**

Finally the complete integrated system is tested. This includes the sensor system with the integrated perception algorithms. Simulated and real world data are collected from the same environment and synchronized. State changes perception algorithms can then be compared to check if the simulated results match the real world performance. The correlation threshold would determine if lane detection algorithms are used to support: LDW, LKAS or ADS. 

|  |  |
| --- | --- |
| Simulation | Real |
|  |  |

After demonstrating that the lane model is accurate enough, the virtual testing tool can be used to support the assessment of lane detection algorithms. Virtual tests can be used to dramatically speed up the validation process and provide enough evidence that the system works as expected across the ODD. Once a base line correlation of the models and tool chain is achieved, the virtual testing tool can used to validate a large span of behaviors and confirm safe responses to unexpected situations. By applying variations and randomization of the different inputs, the system response is being tested across a wide range of scenarios and stimulus, and more confidence in its performance is gathered. The confidence can be reflected by coverage metrics ( measured on the input data and/or ODD ranges) , where higher measured coverage correlates to higher confidence in the system performance, as it was tested over a wider set situations.



1. **LiDAR/RADAR Model Validation**

**Modelling approaches**

The LiDARs/RADARs modelling approaches can informally be divided into fidelity levels depending on the target application for the M&S. In particular, three reference classes [6] can be derived:

* *“Low” fidelity models:* retrieve the traffic objects’ list and status directly from the virtual environment ground-truth. This modelling paradigm does not afford statistical aspects related to the perception, such as false positives/negatives rate. Low fidelity models might however include basic sensor modelling such as accounting for the sensor’s Field of View (FoV) and occlusions to filter the whole object list;
* *“Medium” fidelity model:* similarly to the low fidelity, medium fidelity models retrieve the objects’ status from the virtual environment kernel. Nonetheless, medium fidelity sensors introduce detection probability (false positive and false negative), the effect of objects’ shape and material on the detection, and environmental effects such as atmospheric degradation;
* *“High” fidelity model:* take advantage of advanced and computationally expensive rendering techniques to model physical processes happening in the real sensor. High fidelity sensors take as input the simulation rendered 3D environment following ray-tracing/rasterization. These sensor models are then allowed to operate with a similar input with respect to their physical counterparts.

  
Each fidelity level can be associated with a corresponding validation procedure. For instance, only “high” and “medium”-fidelity levels provide simulated raw-data that can be investigated against the real-world recording. Conversely, “low” fidelity model can only deliver information related to the object/detection level. Hence any validation procedure requiring raw data as an input cannot be embraced.

**Metrics and KPIs for explicit LiDAR/RADAR Model Validation**

The validation of a sensor model is concerned with establishing whether the developed sensor model is a viable solution for the purpose of performing ADS certification via virtual testing. “Explicit” validation techniques directly compare the direct output of the virtual model with respect to the real counterpart for the same set on input when applicable.

The ADS validation shall rely on the highest fidelity modelling approaches in virtual tests where the perception system plays a critical role. Hence, the annex is mainly concerned with the validation of “medium” and “high”-fidelity LiDAR/RADAR models. Such models are typically validated by exploiting the generated “point-clouds” (PC) or at the “occupancy-grid” (OG) level.

OGs are derived from the PCs where a cell () is assumed to be free () or occupied () if the probability of detecting an obstacle in the cell is greater than 0.5.



OGs deriving from simulation tests and real-world tests can be compared exploiting one of the following methods:

* OGs pixel-loss:

;



* OGs Pearson correlation:

;



* OGs ratio:

;



As an alternative validation procedure, the virtual and real point clouds (PC) can be characterized taking advantage of a distance function, such as:

* PCs Euclidean distance:



* PCs Pearson correlation:

.



Based on the evidence provided in [7], [8], the following correlation thresholds have been reported in literature:

|  |  |  |
| --- | --- | --- |
| Metric | Literature correlation | Optimum |
| OG Pearson | 0.59 – 0.76 | 1 |
| OG ratio | 0.2 – 0.5 | 1 |
| PC Pearson | 0.57 – 0.59 | 1 |

**Implicit LiDAR/RADAR Model Validation**

The perception system of an ADS is the element which acts as an interface between the simulation environment and the actual ADS. Thus, any information retrieved by the sensors is forwarded to the ADS. The validation of a sensor model shall then not disregard the impact that even small discrepancies between the real and virtual model can have on a complex system such as the ADS.

“Implicit” validation techniques establish the validity of the sensor model by including the perception algorithms [9] in the validation chain. The comparison is then carried out by establishing the difference between the simulation derived and real-world detected/tracked traffic objects.

The evaluation of implicit metrics can be carried out by directly compare the distance between the and coordinates of the tracked obstacles over the duration of the experiment using the techniques highlighted in Annex II. Alternatively, the Intersection-over-Union (IoU) metric

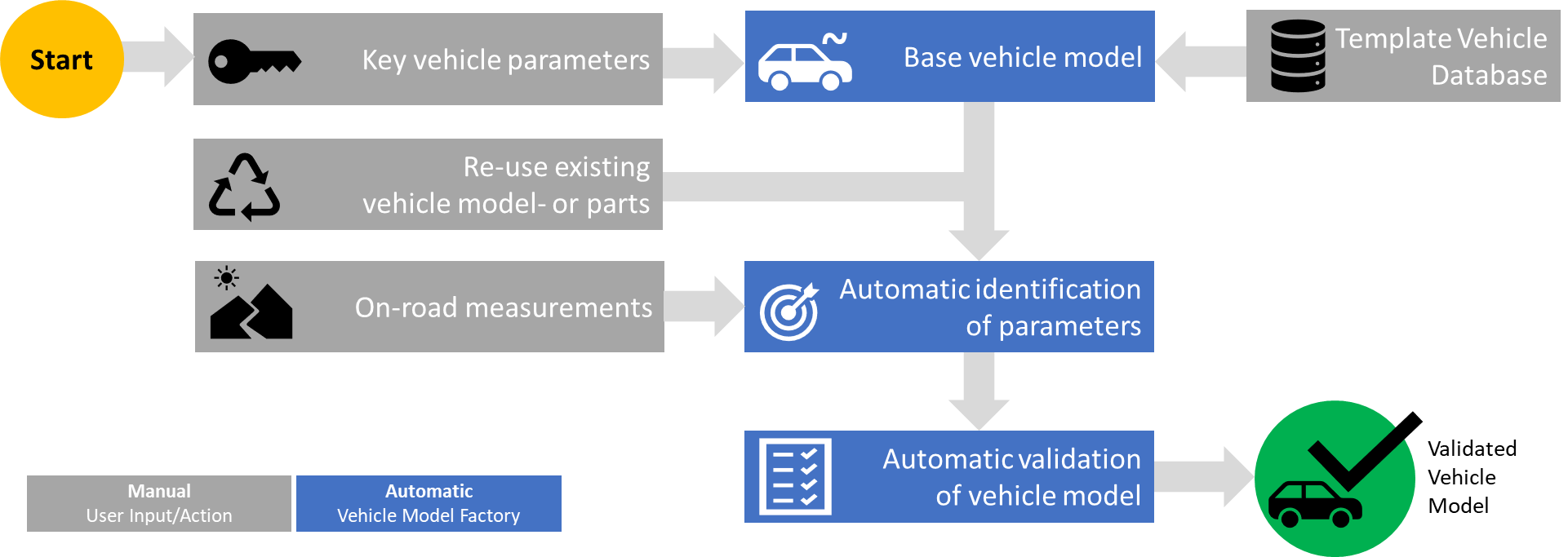
can be computed in case the detection layer returns bounding-boxes.

1. **Vehicle Dynamics Model Validation**

Beyond environment and sensor simulation described in the previous sections, vehicle dynamics simulation plays a major role in the virtual testing toolchain for certification and type approval. Building high fidelity vehicle dynamical models is a time- and cost-intensive activity since it requires precise component measurements (e.g. tire slips) to achieve the required accuracy quality on vehicle-level.

**Vehicle Model Factory (VMF) Approach** [10]

To reduce time, effort, and previous knowledge necessary to build up and validate a vehicle model, software tools can be used which guide the crafting high-fidelity models. For instance, the Vehicle Model Factory (VMF) approach requires 3 main inputs. Firstly, a minimal amount of key vehicle parameters, which can be obtained by vehicle-datasheets or workshop measurements (e.g.: vehicle corner weights, wheelbase, track-width, tire dimensions, etc.). Secondly, the test vehicle must be equipped with a minimum set of measurement- and recording equipment (CAN-access, inertial measurement unit, accelerometer, and GPS). The required instrumentation does not require any structural modifications on the vehicle. Finally, a set of predefined maneuvers must be performed on a test track.

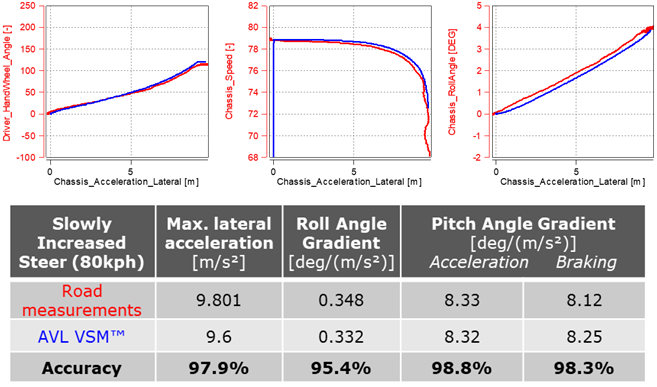


Vehicle Model Factory Workflow.

For the parameter identification task, groups are defined. For example:

* Driving resistance (Driving Resistance Coefficients),
* Weight distribution (Center-of-gravity position vertically and horizontally),
* Suspension (dynamic roll- and pitch behavior),
* Powertrain (motor/engine torque- and pedal-maps, gearbox- and total ratios).

The software tool allows automatic identification of the parameters based on the collected dataset and on the chosen template model. After the parameters are identified, they are inserted into the vehicle model and a simulation for validation is performed automatically.



Correlation quality following the vehicle model factory approach for a C-class vehicles suspension.

**AEB Simulation example**

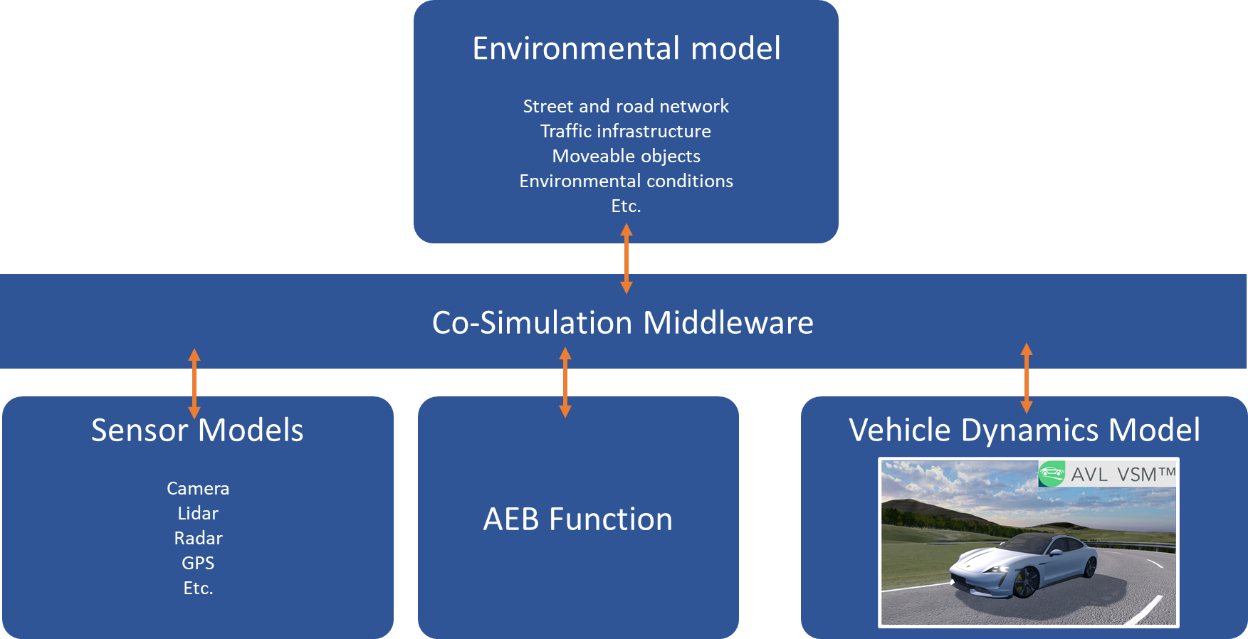
For a virtual certification of ADAS/AD functions, validated vehicle dynamics models are crucial to achieve a maximum correlation between the virtual and the real-world system behavior. For instance, accurate tire and brake models enable realistic Automated Emergency Brake (AEB) testing in terms of deceleration and braking distance results. Validated suspension models will affect the virtual sensor output, such as Radar, Lidar or Camera, in a realistic manner, including e.g. pitching and rolling motion of the chassis. In general, a correlation analysis as part of creating a digital vehicle twin for virtual ADAS testing should be conducted at several levels:

* Vehicle dynamics behavior
* Vehicle geometrics (digital 3D twin)
* Environment sensor output (raw data)
* Perception output (detected and classified objects)
* ADAS/AD controller behavior

To showcase the importance of accurate vehicle dynamics models for ADAS testing, a simulation of an AEB function was conducted, and the results of two different virtual vehicle configurations of a limousine passenger car analyzed. Thereby configuration 1 featured summer tires and configuration 2 winter tires.

As part of a larger General Safety Regulation (GSR) scenario database, a simple AEB scenario was simulated, where the vehicle under test approaches a stationary Target Object Front (TOF) at varying velocities, namely 20 kph, 40 kph and 60 kph. The AEB controller is based on Time to Collision (TTC) thresholds that initiate the different braking modes. The emergency braking is actuated as soon as the TTC exceeds the critical threshold.

In total, six simulations were conducted. Three for the configuration with summer tires, and three using winter tires, having a reduced longitudinal tire grip. The results show that winter tires lead to a substantially longer braking distance. For 60 kph, the braking distance increases from 14.77 m to 17.51 m. The 20 kph case resulted in a 2.2 m increase in braking distance when changing the tires.



Usage of validated virtual prototypes for vehicle dynamics simulation in overall simulation architecture.

Contrarily, the maximum longitudinal deceleration declines when changing from summer to winter tires. The different cases show a reduction in the range between 0.6 (for 20 kph) and 1.8 m/s² (for 60 kph). Although these results are no surprise and can be expected in real-world tests, the simulations demonstrate the effect of slight model variations on the AEB performance. In simulations with low-fidelity dynamics models, this effect might be overlooked, and critical scenarios or collisions might remain undetected.

**References**

[1] “proposed\_cm-s-014\_modelling\_simulation\_-\_for\_consultation.pdf.” Accessed: Jul. 30, 2021. [Online]. Available: https://www.easa.europa.eu/sites/default/files/dfu/proposed\_cm-s-014\_modelling\_simulation\_-\_for\_consultation.pdf

[2] K. A. Maupin and L. P. Swiler, “Validation Metrics for Deterministic and Probabilistic Data,” p. 52.

[3] M. A. Sprague and T. L. Geers, “A spectral-element method for modelling cavitation in transient fluid–structure interaction,” *Int. J. Numer. Methods Eng.*, vol. 60, no. 15, pp. 2467–2499, Aug. 2004, doi: 10.1002/nme.1054.

[4] “ARP5765B: Analytical Methods for Aircraft Seat Design and Evaluation - SAE International.” https://www.sae.org/standards/content/arp5765b/ (accessed Jul. 29, 2021).

[5] J. P. C. Kleijnen, “1999: VALIDATION OF MODELS: STATISTICAL TECHNIQUES AND DATA AVAILABILITY,” p. 8.

[6] B. Schlager *et al.*, “State-of-the-Art Sensor Models for Virtual Testing of Advanced Driver Assistance Systems/ Autonomous Driving Functions,” vol. 3, no. 3, p. 30, 2020.

[7] A. Schaermann, A. Rauch, N. Hirsenkorn, T. Hanke, R. Rasshofer, and E. Biebl, “Validation of vehicle environment sensor models,” in *2017 IEEE Intelligent Vehicles Symposium (IV)*, Los Angeles, CA, USA, Jun. 2017, pp. 405–411. doi: 10.1109/IVS.2017.7995752.

[8] T. Hanke *et al.*, “Generation and validation of virtual point cloud data for automated driving systems,” in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Yokohama, Oct. 2017, pp. 1–6. doi: 10.1109/ITSC.2017.8317864.

[9] A. Ngo, M. P. Bauer, and M. Resch, “A Multi-Layered Approach for Measuring the Simulation-to-Reality Gap of Radar Perception for Autonomous Driving,” *ArXiv210608372 Cs Eess*, Jun. 2021, Accessed: Jun. 25, 2021. [Online]. Available: http://arxiv.org/abs/2106.08372

[10] M. Oswald, “Vehicle Model Factory - Automatic Generation of Validated Virtual Prototypes,” presented at the Internationales Symposium für Entwicklungsmethodik, Wiesbaden, Nov. 2021.