Efficient Splitting of Test and Simulation Cases for the Verification of Highly Automated Driving Functions

Eckard Böde[1] (✉), Matthias Büker[1](✉), Ulrich Eberle[2], Martin Fränzle[1], Sebastian Gerwinn[1], Birte Kramer[1] (✉)

[1] OFFIS - Institut für Informatik, Escherweg 2, 26121 Oldenburg, Germany,
{boede, bueker, gerwinn, fraenzle, kramer}@offis.de
[2] Opel Automobile GmbH, Bahnhofsplatz, 65423 Rüsselsheim am Main, Germany
ulrich.eberle@opel.com

Abstract. We address the question of feasibility of tests to verify highly automated driving functions by optimizing the trade-off between virtual tests for verifying safety properties and physical tests for validating the models used for such verification. We follow a quantitative approach based on a probabilistic treatment of the different quantities in question. That is, we quantify the accuracy of a model in terms of its probabilistic prediction ability. Similarly, we quantify the compliance of a system with its requirements in terms of the probability of satisfying these requirements. Depending on the costs of an individual virtual and physical test we are then able to calculate an optimal trade-off between physical and virtual tests, yet guaranteeing a probability of satisfying all requirements.

Keywords: verification, simulation, highly automated driving, statistical verification, testing, advanced driver assistant systems, optimal trade-off

1 Introduction

Advanced driver assistant systems (ADAS) and highly automated driving functions (HAD) are increasingly complex and their dependency on the environmental situation is increasing. An important step in bringing such systems into the market is to guarantee their safe operation. To this end, the reaction of these systems to all potential inputs needs to be verified. Due to their complexity not only individual inputs but sequences of inputs need to be checked. An analytical verification which exhaustively checks all input combinations and sequences is infeasible[3]. Therefore, it is not only important to develop these functions in a safe way but also use testing for their verification. During testing the system is probed at specific points (input sequences) from which the safety of the

[3] Besides the prohibitively large computational complexity, this also requires an accurate, formal description of possible environments.

system, or more generally the compliance of the system with the elicited requirements, can be inferred. Using statistical arguments, one can estimate the necessary number of tests (number of test-kilometers to drive) in order to guarantee a level of safety which is most likely as high as the level to be achieved without the ADAS under test [1]. For HAD this number will presumably be even higher. As determined in [1] and [2] the scale of such physical tests is also prohibitively large as the costs for such tests (which need to be performed with every newly developed ADAS) amount to the order of hundreds of millions of Euros. Thus, on the one hand, the complexity of the systems forces one to use tests but, on the other hand, physical tests are not sufficiently cost-efficient. A potential solution to this dilemma is to replace physical components with virtual ones mimicking the behavior of their physical counterparts denoted *Virtual Integration*. Depending on which part is replaced with a virtual substitute, these test are called for instance model-in-the-loop (MIL), software-in-the-loop (SIL), hardware-in-the-loop (HIL), or vehicle-in-the-loop (VIL), see [3] or [4]. Such a virtual setup can not only be used for safety assessment, but also for early-phase development, thereby achieving a much more cost-efficient development cycle. However, when replacing parts of a real operational environment with virtual components one has to guarantee a sufficiently realistic behavior of the virtual components such that results obtained from a simulation can be transferred to a non-virtual situation.

Although physical tests are again necessary to estimate the accuracy of the models used, such validation of models would only need to be performed once for each model. Hence, the overall costs for virtual and physical tests could still be feasible. In this paper, we address the question of feasibility by optimizing the trade-off between virtual tests for verifying safety properties of highly automated driver assistant systems and the physical tests for validating the models used for such verification. To this end, we follow a quantitative approach based on a probabilistic treatment of the different quantities in question. That is, we quantify the accuracy of a model in terms of its probabilistic prediction ability. Similarly, we quantify the compliance of a system with its requirements in terms of the probability of satisfying these requirements — obtained as an average across all possible uncertainties. As these probabilities are often unknown but have to be estimated based on the finite amount of test-samples, we additionally account for the statistical estimation uncertainty. Depending on the costs of an individual virtual and physical test we are then able to calculate an optimal trade-off between physical and virtual tests, yet guaranteeing a probability of satisfying all requirements.

Note that such a probabilistic treatment is mainly for practical reasons, similar to the arguments in [1]. We expect a human driver (one of several other traffic participants, from the perspective of a system-under-test) to be subject to randomness. That is, even if the initial situation is identical, the reaction of a traffic participant can be different between repetitions. In order to still quantify the level of compliance with the requirements we merely require the system to satisfy the requirement up to a pre-specified confidence level. For the same reasons we also can only measure the current level of safety with a similar uncertainty, see [1].

The results presented in this paper are meant to be of a generic nature i.e., applicable from a test-process perspective in general. As such, we do not analyse particular test instances, but investigate more the general conditions in which such a framework is applicable. Furthermore we are using an abstract notion of validity, thus we are not giving concrete checkpoints that would allow a test engineer to decide whether a given model is a valid replacement of the real world. Despite this, it allows us to make predictions about the test processes that have a practical impact. In particular, we illustrate how many tests would be needed (both physical and virtual) under the assumptions of an optimal split. Additionally, we can directly calculate the potential savings in costs compared to a pure physical test as illustrated by Winner *et al.* [1] while achieving the same quantitative guarantee of safety (see Section 3.5).

2 Related Work

To use simulation for the verification of ADAS/HAD is a commonly proposed solution for the problems stated above. There are serveral approaches that offer ideas on how to integrate simulation and test in the verification process.

Virtual integration. In [5] an overview of virtual integration methods with their current use in practice, as well their limitations is given. Already for ADAS with environmental perception safety cannot be shown economically only by real test drives due to the high complexity of the systems and tests. Finding the right balance between real test drives and virtual integration tests can be considered as an optimization problem with respect to the effort of building and parameterizing simulation models and the efficiency gain won by simulation techniques. Beside the repeatability and efficiency the additional value of simulation techniques is in particular the possibility to perform tests of the whole system already in early design phases. With regard to the V-model the four virtual integration techniques relevant for the development of ADAS are MIL, HIL, SIL and VIL, also compare [4]. The current limitations of virtual integration are stated as the simulation models a) do not always meet the necessary realism or b) do not have the real-time capability needed for virtual integration methods (or both).

Taxonomy for testing of advanced driver assistance systems. In the survey [3] a taxonomy of approaches for testing advanced driver assistance systems is presented. This concerns different characterizations of test criteria and metrics to quantify the quality of observations or models. Further, different methods to determine the test reference (ground truth) are discussed i.e. either measurement-based, by simulation or by a mixture of both. Finally, the definition of test scenarios is regarded where actual tests are performed to be checked against the reference. In the present paper, we follow the suggested approach by comparing the virtual model with a reference, similar to [6], where the special case of vision based systems is considered.

Combining design time testing and runtime monitoring. In [7] a scenario-based approach is presented that combines testing at design time and monitoring during runtime of an ADAS. This allows to identify the set of relevant scenarios by simulation and thus reducing field testing to these instead of testing all possible scenarios, which would be infeasible for complex ADAS systems. Furthermore, missing scenarios identified at field tests may be fed back into the set of scenarios for design time simulation to improve test coverage.

Assigning test cases to test methods. A method for assigning test cases for automated driving functions to X-in-the-Loop test methods is proposed in [8]. The authors make use of their virtual modular test kit, which is a concept to systematically test automated driving functions in virtual environments. Its goal is to reduce the overall number of necessary tests by a systematic test case generation while keeping the test coverage at the same level. Depending on the test case different requirements arise concerning the set of applicable X-in-the-loop methods. The assignment method has two steps. First, the X-in-the-loop methods are characterized by a Kiviat diagram. On the z-axis of the diagram different assessment scales are plotted like quality of results, operational costs, etc. Second, the requirements of the test cases to the X-in-the-loop methods are represented in a Kiviat diagram as well. By matching the diagrams the set of applicable X-in-the-loop methods can be determined. By defining assessment functions describing the quality of the models, operational costs, etc the best rated method with respect to the defined assessment functions can be identified.

3 Stochastic Methods for Splitting Simulation and Testing

Although simulation offers a thorough investigation and verification of a system under test, testing real components against real environments will always be part of the verification process to guarantee the possibility of a transfer of results obtained in a simulation environment to the deployment phase. To optimize the cost efficiency of the overall verification process reducing and shifting effort towards virtual simulation is of major interest. In this chapter we will present the quantitative basis for an optimal trade-off between real world tests and virtual simulations to achieve a desired level of dependability.

3.1 Preliminaries

As mentioned in the introduction, our approach relies on a probabilistic argument which aims at quantifying the degree to which we can guarantee that the system requirements are fulfilled by the system across all possible situations the system under test might encounter throughout its lifetime. As shown by Winner *et al.* [1], such guarantee can be obtained with purely physical tests. However, these physical tests can also be used to validate a surrogate model of the reality, which in turn can then also be used to verify a system against this model of reality. Before going into details of the approach we first fix the notation of the stochastic variables that we will use in the following.

By the **real system under test** (S_r) we understand the system under test as it will be implemented. Analogously to the **virtual model** (S_v) it receives an input (which could be either provided by a model or the real world) and generates a corresponding response. The **reality or ground truth world model** ($\mathcal{W}$) is considered to be the desired environment for the system under test. The reality can be observed via measurements from a reference system, which provides sequences of measurements. These traces (sequences of measurements) can be compared with the sequences of the **simulation model** ($\mathcal{M}$) which can generate traces of virtual inputs for a system under test. Based on the generated traces, two models or a model and its real-world counterpart can be exchangeable. In that case we say the model $\mathcal{M}$ is a **valid** model of the real world $\mathcal{W}$, denoted by $\mathcal{M} \equiv_{\mathcal{R}} \mathcal{W}$ to check whether a system under test fulfills some **requirements** ($\mathcal{R}$). These are a set of logical formulae, which should be **satisfied** ($\vdash$) for all relevant situations of real world scenes in which the system under test operates.

Furthermore, we are looking at **samples drawn from the reality** (X^w). These are discrete sequences of measurements taken from a system equipped with a reference sensor system. This reference system is able to provide sequences of accurate measurements comparable to **samples generated from the co-simulation of the models** (X^s) which are sequences of virtual measurements of the co-simulation of the virtual environment and the system model.

Definition 1. *We write $S_{\mathcal{W}} = (S_r, \mathcal{W})$ for the real system under test with input from the real world.. Analogously we write $S_{\mathcal{M}} = (S_v, \mathcal{M})$ for the virtual model with input generated from a simulation model.*

Thus read e.g., $P\left(S_{\mathcal{W}} \vdash \mathcal{R}\right)$ as: the probability that the real system under tests satisfies a set of requirements. If a quantity cannot be directly assessed but has to be inferred from other observations or measurements we annotate this with a hat symbol. For example, if we have no access to a probability p, but have a method to estimate this probability, we denote the estimate by $\hat{p}$.

For the verification of a system under design, we are interested in the probability that the designed system will satisfy all requirements when facing environments generated from reality, i.e., the true world model. This probability can be written in terms of conditional probabilities assuming a particular model used for simulation and the probability that this model is an accurate description of the real world. With the law of total probability we arrive at:

$$\begin{aligned} P\left(S_{\mathcal{W}} \vdash \mathcal{R}\right) =& P\left(S_{\mathcal{W}} \vdash \mathcal{R} \mid\ S_{\mathcal{M}} \equiv_{\mathcal{R}} S_{\mathcal{W}}\right) P\left(S_{\mathcal{M}} \equiv_{\mathcal{R}} S_{\mathcal{W}}\right) \\ &+ P\left(S_{\mathcal{W}} \vdash \mathcal{R} \mid\ S_{\mathcal{M}} \not\equiv_{\mathcal{R}} S_{\mathcal{W}}\right) P\left(S_{\mathcal{M}} \not\equiv_{\mathcal{R}} S_{\mathcal{W}}\right) \\ \geq& P\left(S_{\mathcal{W}} \vdash \mathcal{R} \mid\ S_{\mathcal{M}} \equiv_{\mathcal{R}} S_{\mathcal{W}}\right) P\left(S_{\mathcal{M}} \equiv_{\mathcal{R}} S_{\mathcal{W}}\right) \\ \approx& P\left(S_{\mathcal{M}} \vdash \mathcal{R} \mid\ S_{\mathcal{M}} \equiv_{\mathcal{R}} S_{\mathcal{W}}\right) P\left(S_{\mathcal{M}} \equiv_{\mathcal{R}} S_{\mathcal{W}}\right) \quad . \end{aligned} \tag{1}$$

That is, we first split the probability that the real system under test in its desired environment will satisfy the requirements into two cases depending on whether composition of the model and the virtual environment can be regarded as a valid replacement. If this is indeed the case, we can replace the pair $S_{\mathcal{W}}$ with its virtual counterpart. As a result we have split the overall verification effort into a part which can be evaluated purely in a virtual environment (first term in equation (1)) and a part which evaluates the validity of the virtual model.

3.2 Validity of a Virtual Model

Validating a virtual model against its real counterpart is a challenging task (see [3]). In equation (1), we deduced from the validity of a model that we can use the model as a replacement for the real system within the satisfaction of the requirements. However, there are different notions of validity. We could classify a model to be valid with respect to a particular environment, if such replacement is allowed for a particular requirement. The latter interpretation, for example, is the basis for determining a test-method (including the selection of a virtual model) according to the method described in [8]. In the present paper we would like to follow a more generic approach. That is, we would like to define a notion of validity such that the replacement of the virtual model is valid for all requirements.

For this to hold, we have to at least ensure that all sequences of measurements from the real world X^w could be generated within the virtual environment, i.e., finding corresponding X^s. Please note that we assume all traces to be discrete. To avoid that the virtual model dominantly explores part of its sample space, which are not possible to observe in the real world, a stronger notion of validity would also require that for all traces in the virtual model, there exists a corresponding trace (sequence of measurements) in the real world. Even if traces are possible to generate for both systems, virtual model and real world, it could happen that different kinds of traces are differently favoured. That is some traces might be more likely to be generated in the real world compared to the likelihood of generating them using the virtual model. To summarise, we have the three (increasingly stronger) notions of validity:

1. All sequences of real world observations are also possible within the virtual model
2. Additionally, for each possible sequence within the virtual model, there exists an identical sequence of measurements within the real world
3. For each sequence of measurements there exists an identical sequence within the respective other model. Additionally, the likelihood of generating such sequence is also equal.

For simplicity, we only consider the first notion of validity within this paper. It should also be noted that all notions can be further relaxed by not requiring the existence of an identical sequence, but the existence of a sequence which is close to the required one.

3.3 Splitting Simulation and Testing for Ubiquitous Requirements

Given these notations, we can formulate the following properties as conditional probabilities. These are modeled as probabilities as the models used to describe the environment and potentially for the system model as well are likely to contain stochastic variables and hence the satisfaction of requirements is potentially also subject to this encoded variability. For example, with the abbreviations introduced above, we can write down the probability that the system model satisfies the requirements, given that the simulation model is an accurate description of the (needed aspects of the) true world model. Note that in this section we assume that the requirements can be validated both via simulation as well as via physical testing. That is, we assume that the environmental model used for simulation provides all necessary information to evaluate whether a single sample generated from the model satisfies the requirements.

Definition 2 (Satisfaction of requirements for a given simulation model). *Let $S_{\mathcal{M}}$ denote the virtual model of the system under test with input generated from a virtual model $\mathcal{M}$ and $\mathcal{R}$ denote the requirements we would like the system to satisfy. We write the conditional probability of the system satisfying the requirements under the assumption that the simulation model $\mathcal{M}$ is an accurate description of the real world $\mathcal{W}$ as:*

$$P_{\vdash}^{M} := P\left(S_{\mathcal{M}} \vdash \mathcal{R} \mid \ S_{\mathcal{M}} \equiv_{\mathcal{R}} S_{\mathcal{W}}\right) \quad . \tag{2}$$

Due to the potential stochastic variability encoded into the simulation model, this probability is a property of the simulation model. As simulation models are typically too complex to be analyzed symbolically, this probability cannot be calculated exactly but can be approximated or bounded by means of a statistical analysis. To this end, samples from the simulation model can be generated and estimates of the probability can be obtained. This is the main goal of simulation based verification. In order to rigorously quantify the level of certainty associated with such a verification process it is important to keep track of the sample uncertainty incurred by the simulation based verification.

Assume we have generated m samples $X_1^s, \ldots, X_m^s$ using the simulation model. For each of these samples we can test whether the requirements are satisfied for the particular trace, $X_i^s \vdash \mathcal{R}$. Based on these results, we can, for example, estimate the probability (2) by the relative frequency of the samples satisfying the requirements (ideally, all traces satisfy the requirements, i.e., the relative frequency will be 1). If we denote this estimate $\hat{P}_{\vdash}^{M}$, we can statistically bound the probability that this estimate will deviate from the true probability $P_{\vdash}^{M}$ by more than any given ϵ_s. The resulting bound on the probability depends on three variables: the confidence δ_s, the accuracy ϵ_s and the number of samples m used for this estimate. If two of these are given the others can usually be calculated based on the other two (see below for some specific examples).

$$P_{X_1^s,\ldots,X_m^s}\left(\left|\hat{P}_{\vdash}^{M} - P_{\vdash}^{M}\right| \geq \epsilon_s(\delta_s, m)\right) \leq \delta_s \quad . \tag{3}$$

In words, such a formula bounds the likelihood of the results being a fluke (judged by the estimation being further than $\epsilon_S(\delta_S, m)$ from the true value apart) as a result of unlucky observed data. Here, we have written $\epsilon_S(\delta_S, m)$ to stress the fact that the accuracy ϵ can be calculated from the other two parameters δ_S, m.

Now we have to combine this probability with the probability that the model represents the relevant information sufficiently well. Here, we assume that we represent all relevant information which is needed to answer whether the specified requirements are satisfied on a single trace basis. In other words, we say that the simulation model represents the ground truth model if a trace of observations of the real world is considered possible in the simulation model and testing this trace with respect to the requirements on both models leads to the same answer.

$$P_{\equiv_\mathcal{R}} := P\left(S_\mathcal{M} \equiv_\mathcal{R} S_\mathcal{W}\right) \ . \tag{4}$$

As we do not have access to the mathematical description of the world model, we need to estimate this probability based on observations of the real world, similar as we have estimated (2) via sampling from the model. Again, we need to keep track of the residual uncertainty associated with such an empirical estimation procedure. Specifically, for n observations $X_1^w, \ldots, X_n^w$ of the real world, we have:

$$P_{X_1^w,\ldots,X_n^w}\left(\left|\hat{P}_{\equiv_\mathcal{R}} - P_{\equiv_\mathcal{R}}\right| \geq \epsilon_w(\delta_w, n)\right) \leq \delta_w \ . \tag{5}$$

These different estimates can be used to bound the overall probability of interest (see equation (1)). In fact, for the first term in equation (1), as we have no access to the true probabilities, we can use their sample-based estimates from equations (3) and (5) to obtain:

$$P\left(S_\mathcal{W} \vdash \mathcal{R}\right) \geq \left(\hat{P}_\vdash^M - \epsilon_s\right)\left(\hat{P}_{\equiv_\mathcal{R}} - \epsilon_w\right) \qquad \text{with } P \geq (1-\delta_s)(1-\delta_w) \ . \tag{6}$$

If all physical tests, i.e., observations of the behavior of the system, satisfy the requirements and could also have been generated by the simulation model both estimates $\hat{P}_\vdash^M, \hat{P}_{\equiv_\mathcal{R}}$ are 1. In this case the equation simplifies to:

$$\begin{gathered} P\left(S_\mathcal{W} \vdash \mathcal{R}\right) \geq \left(1-\epsilon_s\right)\left(1-\epsilon_w\right) \geq 1 - \epsilon_s - \epsilon_w \\ \text{with } P \geq (1-\delta_s)(1-\delta_w) \ . \end{gathered} \tag{7}$$

For simplicity, we have omitted the dependence of δ_s, δ_w, m, n on the accuracies ϵ_w, ϵ_s. The above equation suggests that the effort to spend on either simulation or physical tests amount to the same contribution to the overall safety guarantee (satisfaction of the requirements). However, due to the multiplication of residual uncertainties, i.e., confidences δ_s, δ_w, we might want to allow for a smaller confidence in the simulation model thereby requiring a larger confidence in the simulation analysis while obtaining the same level of overall confidence and safety estimate. In other words, we can achieve the same safety guarantee with different splits between physical and simulation tests. This degree of freedom can therefore be exploited to obtain an optimal trade-off with respect to the resulting costs.

Assuming a fixed cost c_s for each simulation run and c_w for each physical test to validate the simulation model, we therefore can solve the following constrained optimization problem for a given overall confidence level X and a safety level Y:

$$\begin{aligned}&\min_{n,m}\,(c_s m + c_w n) \quad s.t.\\&(1-\delta_s)(1-\delta_w) \geq X \text{ and } (1-\epsilon_s(\delta_s, m))(1-\epsilon_w(\delta_w, n)) \geq Y \ \ .\end{aligned} \tag{8}$$

Although we optimize the costs within the above optimization problem, we only do so under the constraint that a certain level of safety has to be guaranteed. Such optimization problem can be solved (at least numerically) if the estimation accuracy functions ϵ_w, ϵ_s are given. For the particular situation in which we are aiming at estimating a probability - which is specified in terms of a binary indicator variable (satisfaction of the requirements) - we can use a Bernoulli bound to obtain a specific form of the accuracy functions, for example a Clopper-Pearson bound (see [9] and Section 3.5).

If the simulation model is only used for generating the environment of the system under test and is therefore independent of the system under test the physical tests to validate that model need be performed only once for a model. The model in turn can be used to verify more than one system without requiring additional physical tests for model validation provided that the samples that were generated from the model were generated for each system under test. However, if the model is allowed to change or adapt to the physical test data that has been acquired, validation of the model corresponds to bounding the prediction performance of a learning system, as the model *learns* from the physical test data. Although this is possible, the calculations are more involved and we therefore postpone this discussion.

3.4 Splitting Simulation and Testing Based on Type of Requirements

In the previous section, we assumed that all requirements can be tested either using simulation or physical tests. Additionally, we also measured the quality of a simulation model based on its ability to generate traces and leading to the same answer regarding the satisfaction of the requirements. In practice, there are certain requirements which are outside the scope of the simulation model. For instance, the model might not include certain variables within its representation that are relevant for some requirements. That is, we might have a model of the vehicle dynamics at hand, but would like to test a requirement which specifies how a route-planning component should work. In these situation Schuldt *et al.* [8] proposed a method to judge which type of test-method (for example HIL or MIL should be applied. In particular, they also suggested that the quality of the provided simulation model should be taken into account when selecting a suitable test-method. Using the results from the previous sections, we can provide a quantitative measure which supports their method.

Also, we can use similar calculations as above to provide an overall measure in satisfying the desired requirement. Specifically, assume we have given two types of requirements $\mathcal{R}_1, \mathcal{R}_2$ each of which specifies the desired behaviour for different parts of the system under test. Assume further that we have two simulation models $\mathcal{M}_1$ and $\mathcal{M}_2$, each modelling the respective part of the system under test and their respective inputs. Then we can write the overall probability of satisfying the requirements as

$$\begin{aligned} P\left(S_{\mathcal{W}} \vdash \mathcal{R}_1 \wedge S_{\mathcal{W}} \vdash \mathcal{R}_2\right) &= P\left(S_{\mathcal{W}} \vdash \mathcal{R}_1 | S_{\mathcal{W}} \vdash \mathcal{R}_2\right) P\left(S_{\mathcal{W}} \vdash \mathcal{R}_2\right) \\ &\overset{\text{ind.}}{=} P\left(S_{\mathcal{W}} \vdash \mathcal{R}_1\right) P\left(S_{\mathcal{W}} \vdash \mathcal{R}_2\right) \quad . \end{aligned} \tag{9}$$

Here we have assumed that the satisfaction of the second requirement does not affect the satisfaction of the first one. If both requirements restrict different parts of the system under test, this might be reasonable, however, it should be noted that all components within the system under test are likely to be connected via a certain computation path. Therefore, the independence assumption might be too strong. Similar to equation (1), we can now resolve each of the remaining terms in equation (9) using the respective models.

If for one of the requirements there is no model available, we can simply perform physical tests to estimate the corresponding probability. In this case, assuming all tests have been passed, we can rewrite equation (7) to obtain

$$\begin{aligned} P\left(S_{\mathcal{W}} \vdash \mathcal{R}_1 \wedge S_{\mathcal{W}} \vdash \mathcal{R}_2\right) &\geq \left(1-\epsilon_s^1\right)\left(1-\epsilon_w^1\right)\left(1-\epsilon_w^2\right) \\ \text{with} \quad P &\geq (1-\delta_s^1)(1-\delta_w^1)(1-\delta_w^2) \quad . \end{aligned} \tag{10}$$

For the first requirement, we have $\left(1-\epsilon_s^1\right)\left(1-\epsilon_w^1\right)$ representing the accuracy of checking the satisfaction of the requirement times the probability of the model being valid. For the second requirement, we can omit the model validation probability, as we assume to perform real-world tests. Using such formulation, we can again optimise the costs under safety constraints. In the above formulation, we perform real-world tests for checking the validity of the model and checking the satisfaction of $\mathcal{R}_2$. However, we can re-use the same real-world test for both objectives. Therefore, the number n of real-world tests within (8), can be used for both ϵ_w^1 and ϵ_w^2.

3.5 Practical Considerations

In this section we investigate the potential of the approach as outlined in the previous sections from a more practical perspective. Applying the procedure outline in the previous section, one has to first set up a model for simulation then collecting independent observations of the real world, which allows to check whether these observations can also be generated by the simulation model, and finally performing the simulation-based tests. We therefore use the real-world observations only for validating the model here, although a double use, i.e., validating the model and checking requirements would be possible as well and would further strengthen the guarantees.

Validating a simulation model using a reference sensor system. In the previous sections we assumed that the validity of the simulation model can be checked on a single observation basis. In the simplest case, this can be achieved by verifying that a (sequence of) measurements can be reproduced within the virtual environment used as a simulation model. For example, the simulation model could consist of several modules integrated into a co-simulation platform. To be able to generate a simulation run, further parameters such as road topology, behavior of traffic participants, etc., have to be specified within the co-simulation platform. By choosing a suitable set of these parameters, one can try to mimic the sequence of measurements. If the observed sequence can be reproduced, the necessary check can be considered passed. If all measurement data can be reproduced, the corresponding estimate of the probability that the virtual model is an accurate description of the real world $\hat{P}_{\equiv_{\mathcal{R}}}$ in equation (5) and (7) is 1.

More precisely, one has shown that the virtual model is capable of reproducing the sensor measurements of the (potentially inaccurate) sensor setup used for recording. Therefore, if one aims at validating a system which should serve as a generator of ground truth data, one should use a sensor setup which can act as a reference, i.e., has the desired accuracy. With the help of an applied co-simulation platform, one could measure the (relative) positions of all objects and then reproduce the trajectories of all detected objects within the simulation. If the measurements also contain a visual component, one needs to show that the rendering procedure is capable of generating the recorded video sequence.

It should be noted, that the same procedure can also be used to validate components, such as sensor models, against their hardware counterparts. To this end, one would discretely measure pairs of signals, input and output signal, where input signals could be obtained via a reference sensor system and the output would be measured from the sensor one would like to model. To validate the virtual model of the sensor it would be checked whether it can reproduce all observed sequence of input-output pairs. In fact, having validated a sensor model would also mean that all inaccuracies of the sensor are captured within the model. By combining different validated models for components, one can then also conclude that the combined model is validated. However, the confidence in the combined model is reduced as the overall confidence is given by the product of the confidences for the individual components.

Exemplary optimization for a cost-efficient trade-off. To evaluate the practical impact and associated costs (savings) we calculate the optimal trade-off between simulation and tests as outlined in equation (8). To this end, we assume that both physical tests as well as simulation based tests have not revealed any violation of the requirements and model validation, respectively. Otherwise we assume that the underlying problem has been addressed and the corresponding tests have been successfully repeated. For the costs of a physical test we assume here $10\frac{€}{\mathrm{km}}$. Relative to these costs we assume a virtual kilometer within a simulation environment to cost a fraction of 0.01, that is here $0.1\frac{€}{\mathrm{km}}$. Note that these values are only illustrative figures, but are easily replaceable by more accurate values. For the desired overall confidence we are using 0.99 and our desired accuracy is set at $1 - 1.375\ 10^{-7}$ which roughly corresponds to half of the current empirical probability of no accident per km [4]

As mentioned in section 3.3, we can use the Clopper-Pearson confidence interval to determine the accuracy ϵ_W, ϵ_S for the simulation and physical test specific accuracies. Specifically, we have for both ϵ_W, ϵ_S :

$$\epsilon_W(\delta_W, k) = 1 - (\delta_W)^{\frac{1}{k}}\ , \quad \epsilon_S(\delta_S, k) = 1 - (\delta_S)^{\frac{1}{k}} \quad . \tag{11}$$

Therefore, in the case of no violations of any requirements, we have:

$$\begin{aligned} &\min_{n,m} (c_S m + c_W n) \qquad s.t. \\ &(1-\delta_S)(1-\delta_W) \geq 0.99 \qquad \wedge \qquad (\delta_S)^{\frac{1}{m}} (\delta_W)^{\frac{1}{n}} \geq 1 - 1.375 \cdot 10^{-7} \quad . \end{aligned} \tag{12}$$

As the costs for simulation and physical tests might vary between different systems, models, and companies, we illustrate the achievable trade-off for a range of possible costs. We plotted the results in Figure 1.

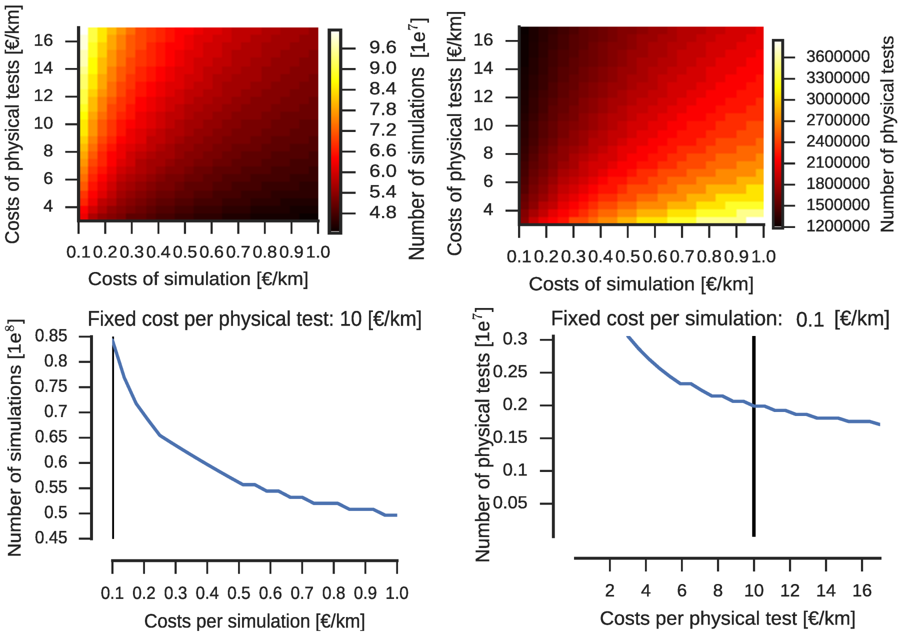

Fig. 1. The achievable trade-off between simulation-based and physical tests.

Here the upper two three dimensional diagrams belong together, meaning that if you choose a fixed cost for a physical test and a fixed cost per simulation you can estimate the number of needed simulations (left diagram) and physical

[4] www.adac.de/_mmm/pdf/statistik_7_1_unfallrisiko_42782.pdf

tests (right diagram) for an optimal trade-off from the color depicted in the diagrams. In the lower two diagrams we each fix one dimension of the diagrams above to observe how the needed number of simulations/physical tests changes compared to the cost per simulation/physical test. The intersection with the straight lines drawn into the diagram thus mark the optimal trade-off from the example given above.

It can be seen from the lower panel in Figure 1 that if a physical test is more costly the optimisation procedure will increase the number of simulations, as expected. However, the overall number of tests (both physical and virtual) is still quite high, which is due to the high safety targets and no additional assumptions.

Comparison with purely physical testing. We also investigated the potential savings in costs using this approach compared to a setting using only purely physical tests. In Figure 2 we plotted the relative savings in Euros when comparing a setting in which only real-world tests are performed to check the requirements to an optimal split. Thus this diagram does not say anything about the amount of tests necessary but only about the relative savings of performing an optimal split compared to pure physical tests. Although we only use real-world observations for model validation and not for testing requirements, the potential savings amount to over 90 percent of the costs associated with the several hundred million kilometers that were estimated to be sufficient using only real-world testing [1]. The savings are particularly dramatic in settings where the costs of a physical test are much higher than for an individual virtual test (upper left corner in Figure 2).

4 Conclusion

In this paper, we focused on the foundations for a quantitative analysis of splitting test-cases into virtual and physical tests, thereby taking into account the difference in costs for these two types. Although we did not use any further assumptions on the regularity of tests (e.g., nearby scenarios are more likely to produce similar satisfaction results with respect to requirements), the results show that the total savings in costs can be quite substantial ($\approx 90\%$ in the given example) when compared to the setting of testing all requirements purely in real situations. Additionally, such savings are likely to be multiplied, as the models used for simulations can be re-used, once they have been validated using the physical tests. Even when we have made slight changes in the simulation model we could include a prior belief about the quality of the simulation into our approach. With the help of a prior quality belief we could reduce the needed real-world observations even further.

As mentioned in the introduction, the results presented in this paper are meant to be of generic nature. In fact, from a very abstract perspective, the overall test-process is unchanged, but incorporates the quality metric as proposed by Winner *et al.* [1] and can be integrated into methods for selecting appropriate

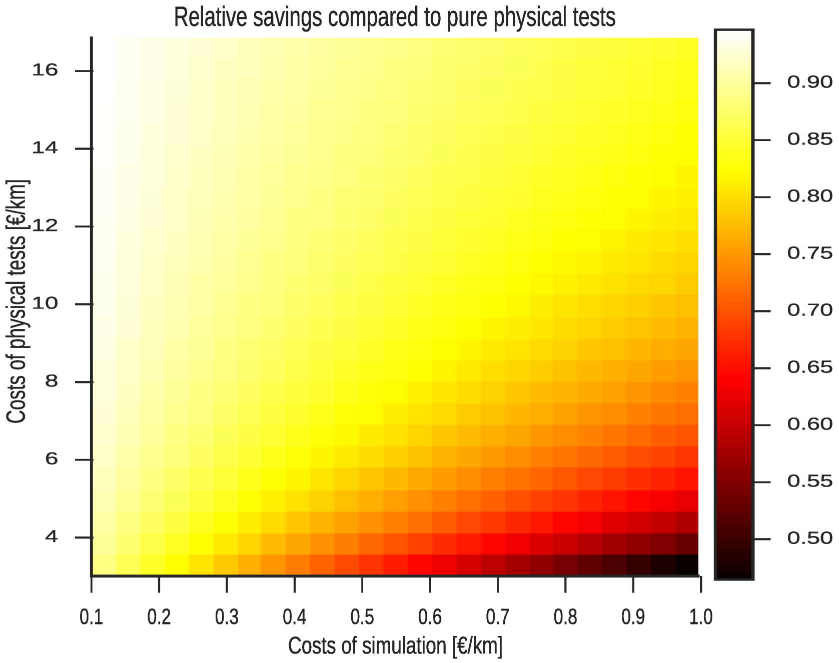

Fig. 2. Comparison of the optimal split and a purely physical testing setting.

test-methods such as [8]. Once a decomposition of the system (e.g. via [10]) has been identified in terms of which parts of the system can be safely replaced by virtual counterparts, this also provides guidance for a X-in-the-loop test setup. Furthermore, if one can identify critical scenarios either as done in PEGASUS[5] or simulation based as in [11] one can further reduce the overall needed effort.

The main challenge for our approach to hold is that a tool for model validation is missing. Given a model: how can we find out whether a sequence of measurements could be reproduced with this given model? This becomes even more difficult if we assume a different notion of validity as discussed in Section 3.2. To use simulation for the verification of highly automated driving functions we need to be able to decide how valid the simulation is. What are the important aspects? Which deviations from reality are allowed? Thus, rigorous model validation is needed to bring such systems into the market.

Moreover, we have seen that a substantial amount of simulation is necessary. Such simulations have a high computational complexity. Thus one would wish for possibilities to further reduce the simulation effort, e.g. with the use of Multilevel Monte Carlo Methods. [12].

Although the results presented in this paper are of quite general nature, we believe that the quantification and necessary formalisation of the different

[5] https://www.pegasusprojekt.de/en/about-PEGASUS

aspects might lead to not only a more efficient test process but also to a safer overall system.

Acknowledgments. This study was partially supported and financed by Opel Automobile within the context of PEGASUS (**P**roject for the **E**stablishment of **G**enerally **A**ccepted quality criteria, tools and methods as well as **S**cenarios and **S**ituations for the release of highly-automated driving functions), a project funded by the German Federal Ministry for Economic Affairs and Energy.

References

1. Winner, H.: Quo vadis, FAS? In: Handbuch Fahrerassistenzsysteme. Springer (2015) 1167–1186
2. Kalra, N., Paddock, S.M.: Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability? RAND Corporation (2016)
3. Stellet, J.E., Zofka, M.R., Schumacher, J., Schamm, T., Niewels, F., Zollner, J.M.: Testing of advanced driver assistance towards automated driving: A survey and taxonomy on existing approaches and open questions. In: Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on, IEEE (2015) 1455–1462
4. Hallerbach, S., Eberle, U., Köster, F.: Absicherungs- und Bewertungsmethoden für kooperative hochautomatisierte Fahrzeuge. In: AAET 2017, Braunschweig (2017) 369
5. Hakuli, S., Krug, M.: Virtuelle Integration. In Winner, H., Hakuli, S., Lotz, F., Singer, C., eds.: Handbuch Fahrerassistenzsysteme. Springer (2015) 125–138
6. Nentwig, M.: Untersuchungen zur Anwendung von computergenerierten Kamerabildern für die Entwicklung und den Test von Fahrerassistenzsystemen. PhD thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (2014)
7. Mauritz, M., Rausch, A., Schaefer, I.: Dependable ADAS by Combining Design Time Testing and Runtime Monitoring. In: FORMS/FORMAT. (2014) 28–37
8. Schuldt, F., Menzel, T., Maurer, M.: Eine Methode für die Zuordnung von Testfällen für automatisierte Fahrfunktionen auf X-in-the-Loop Verfahren im modularen virtuellen Testbaukasten. In: 10. Workshop Fahrerassistenz-systeme. (2015) 171
9. Clopper, C.J., Pearson, E.S.: The use of confidence or fiducial limits illustrated in the case of the binomial. Biometrika **26**(4) (1934) 404–413
10. Ammersbach, C., Winner, H.: Functional Decomposition - An Approach to Reduce the Approval Effort for Highly Automated Driving. In: Tagungsband Fahrassistenz. (2017)
11. Hallerbach, S., Xia, Y., Eberle, U., Koester, F.: Simulation-based identification of critical scenarios for cooperative and automated vehicles. In: SAE Technical Paper, SAE International (04 2018)
12. Giles, M.B.: Multilevel Monte Carlo Path Simulation. Operations Research **56**(3) (2008) 607–617