An Application of Scenario Exploration to Find New Scenarios for the Development and Testing of Automated Driving Systems in Urban Scenarios

Barbara Schütt[1][a], Marc Heinrich[1], Sonja Marahrens[2], J. Marius Zöllner[1] and Eric Sax[1]
[1]*FZI Research Center for Information Technology, Karlsruhe, Germany*
[2]*IPG Automotive GmbH, Karlsruhe, Germany*

Keywords: Advanced Driving Assistance Systems, Automated Driving, Scenario-based Testing, Scenario Exploration.

Abstract: Verification and validation are major challenges for developing automated driving systems. A concept that gets more and more recognized for testing in automated driving is scenario-based testing. However, it introduces the problem of what scenarios are relevant for testing and which are not. This work aims to find relevant, interesting, or critical parameter sets within logical scenarios by utilizing Bayes optimization and Gaussian processes. The parameter optimization is done by comparing and evaluating six different metrics in two urban intersection scenarios. Finally, a list of ideas this work leads to and should be investigated further is presented.

1 INTRODUCTION

The development of semi-automated, automated and autonomous vehicles has played an important role in the software and hardware departments of automotive manufacturers during recent years. The consulting company Gartner has already anticipated *autonomous things* as a "hot topic" several times in previous years and is now going one step further. The current report "Top 10 Strategic Technology Trends for 2022" predicts *autonomic systems* as a main area of interest: systems that can not only make autonomous decisions, but additionally are able to adapt and change their behavior according to the environment (Gartner, 2021).

One type of autonomous or autonomic system are automated vehicles. A major challenge besides their development is to ensure that the system is sufficiently safe and can be approved and permitted on public roads. A widely discussed testing approach is scenario-based testing: According to (Otten et al., 2018), one goal is to take realistic field trial test drives into simulation environments, where predefined scenarios often serve as a basis for the derivation of relevant test cases in automated assessment and, thus, reducing the needed amount of real test drives. Moreover, the proper representation and usage of scenarios during the development process support a seamless development and testing of automated driving functions, as well as the specification of requirements and automated derivation of test cases (Bach et al., 2016). However, the specification of scenarios can include parameter ranges, where only a sub-set of these ranges might bring insight into the performance of an automated driving function or hold critical scenarios. Additionally, introducing new parameters or parameter ranges in a scenario increases the number of scenarios exponentially.

Novelty and Main Contribution to the State of the Art

The novelty and main contribution of this paper is a parameter evaluation for finding challenging and critical scenario parameters in predefined parameter ranges. Thus, we

- optimize the parameters for different intersection scenarios with different criticality metrics to find interesting scenarios and to save simulation time, and
- use the information gained by the optimization process to further assess these parameters and their meaning for redefining parameter ranges and the evaluation and assessment of an automated driving function.

[a] https://orcid.org/0000-0001-8439-0322

Schütt, B., Heinrich, M., Marahrens, S., Zöllner, J. and Sax, E.
An Application of Scenario Exploration to Find New Scenarios for the Development and Testing of Automated Driving Systems in Urban Scenarios.
DOI: 10.5220/0011064600003191
In *Proceedings of the 8th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2022)*, pages 338-345
ISBN: 978-989-758-573-9; ISSN: 2184-495X

Structure

The paper is structured as follows: Section 2 briefly introduces topics related to this work, e.g., scenario abstraction levels or Bayes optimization. Section 3 explains the proposed exploration algorithm and setup, including the simulation environment setup and evaluation. In section 4, we conclude and give a short overview of possible future work.

2 RELATED WORK

In the context of this work, the terms *scenario* and *scene* are used as summarized by (Steimle et al., 2021). A scene is a snapshots of a traffic constellation. A scenario is a sequence of scenes and describes the temporal development of the behavior of different actors within this sequence.

Finding new scenarios is a relevant step for defining new test cases to assess an automated driving system's safety. (Bussler et al., 2020) use evolutionary learning to find relevant parameter sets within logical scenarios and utilize Euclidean distance and time-to-collision for their fitness evaluation to find more critical scenarios. Another approach proposed by (Baumann et al., 2021) uses reinforcement learning combined with the metrics headway and time-to-collision to gain new test cases. Additionally, (Abeysirigoonawardena et al., 2019) use Bayes Optimization and Euclidean distance to generate training scenarios for a driving function to learn to avoid pedestrians by reinforcement learning. However, their work does not produce a scenario set suitable for testing since their approach always uses the current state of the driving function which changes over the course of the experiments. Further, there are other approaches to find new scenarios, e.g., extracting scenarios from recorded data sets as shown by (King et al., 2021) and (Zofka et al., 2015) or by experts planning and designing scenarios from scratch.

2.1 Scenario Abstraction Levels

(Menzel et al., 2018) suggest three abstraction levels for scenario representation. The most abstract level of scenario representation is called *functional* and describes a scenario via linguistic notation using natural, non-structured language terminology. The main goal for this level is to create scenarios that are easily understandable and open for discussion between experts. It describes the base road network and all actors with their maneuvers, such as a right-turning vehicle or road crossing cyclist. The next abstraction level is the *logical* level and refines the representation of functional scenarios by a detailed representation with the help of state-space variables. These variables or parameters can, for instance, be ranges for road width, vehicle positions, and their speed, or time and weather conditions. The most detailed level is called *concrete* and describes operating scenarios with concrete values for each parameter in the parameter space. Therefore, one logical scenario can yield many concrete scenarios, depending on the number of variables, size of parameter ranges, and step size for these ranges.

2.2 Bayes Optimization and Gaussian Process

Bayes optimization (BO) proceeds by maintaining a global statistical model of a given objective function $f(\mathbf{x})$ iteratively and consists of two main steps (Greenhill et al., 2020): The first step is the **Gaussian process**, which is used to represent the predicted mean $\mu_t(\mathbf{x})$ and the uncertainty $\sigma_t(\mathbf{x})$ for each point $\mathbf{x}$ of the input space, with the given set of observations $\mathcal{D}_{1:t} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ...(\mathbf{x}_t, y_t)\}$, where $\mathbf{x}_t$ is the process input and y_t the corresponding output at time t. After that, an **Acquisition function** is used to evaluate the beliefs about the objective function regarding the input space, based on the predicted mean $\mu_t(\mathbf{x})$ and uncertainty and chooses the most promising setting $\sigma_t(\mathbf{x})$.

2.3 Scenario Metrics

Scenario metrics are used to assess the quality of a scenario regarding the aspect that needs to be evaluated. According to (Schütt et al., 2021), scenario quality can be assessed at three different levels of resolution: nanoscopic (a scenario segment is evaluated, e.g., a single time step), microscopic (a complete scenario is evaluated, e.g., one concrete scenario), and macroscopic (a set of scenarios is evaluated, e.g., a logical scenario). Before a metric for the evaluation process is chosen, the usage, goals, and purpose of a scenario need to be clear, e.g., (Schönemann et al., 2018) propose a hazard analysis and risk evaluation to determine safety goals and show their approach on the example of a valet parking system. The formulated safety goals can be used in following steps to choose the metrics for scenario evaluation or to determine the performance of an automated driving system concerning its requirements.

2.4 Simulation Tools

Commercial tools for automotive simulation among others are available from dSPACE (dSPACE, 2021), and IPG (IPG Automotive GmbH, 2021). Both simulation tools provide modules for map and scenario creation, sensor models and dynamic models, to name some examples. A further tool is Carla, an open-source simulator with a growing community and based on the game engine Unreal (Dosovitskiy et al., 2017). It offers several additional modules, e.g., a scenario tool which includes its own scenario format, a graphical tool for creating scenarios, a ROS-bridge, and SUMO support. SUMO is an open-source software tool for modeling microscopic traffic simulation from DLR (Lopez et al., 2018). It specializes on big scale of traffic simulation and can be used for evaluating traffic lights cycles, evaluation of emissions (noise, pollutants), traffic forecast, and many others.

3 DIRECTED SCENARIO EXPLORATION

3.1 Optimization Setup

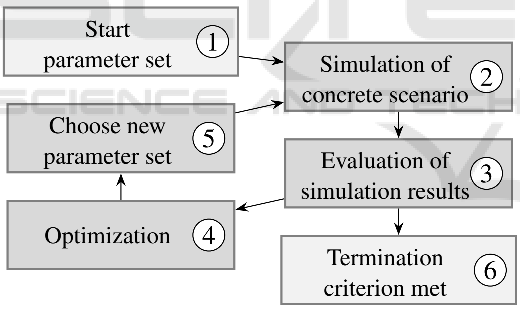

Figure 1: Optimization workflow.

The optimization is an iterative process and is outlined in Fig. 1. First, a start parameter set is selected (1), and simulated as summarized in step (2). The results are evaluated (3), a new parameter set is chosen (5) by the optimization algorithm (4), and it is simulated again (2). This step is repeated until a termination criterion is met (6). The open-source project common Bayesian optimization library (COMBO) is employed in the experiments since it offers Bayesian optimization that uses automatic hyperparameter tuning, Thompson sampling as a method of picking the next best candidate, and random feature maps for better performance (Ueno et al., 2016). Throughout this work, Bayesian optimization and Gaussian processes were used. However, other optimization algorithms might be used since the focus of this work does not lie on the optimization itself.

3.2 Simulator Setup

The simulation tool CarMaker [1] serves as a basis for the simulator setup (IPG Automotive GmbH, 2021). As an open integration and test platform, CarMaker provides a central control unit running the closed-loop simulation. It includes all proprietary and external models according to the given scenario and constraints. In this case, proprietary models of the complete simulation environment comprise the road, environment and traffic models. Six external models were integrated as FMUs (Functional Mock-up Unit) via an extended version of the Open Simulation Interface (OSI) (ASAM OSI, 2021), realizing a setup with three sensor models (camera, lidar (Linnhoff et al., 2021), and radar), an autonomous driving function (see section 3.3), a motion control model and a vehicle dynamics model. This simulator setup is shown in Fig 2. The output quantities of each simulation are handed over to the optimization setup described in section 3.1. The next scenario to be executed is then chosen directly by the optimization setup via script commands, leading to the optimization workflow pictured in Fig 1.

3.3 Driving Function

To calculate the trajectory of the ego vehicle, a lightweight and highly automated driving function is used. The function is centered around a modified, curvature-aware version of the Intelligent Driver Model (IDM) as used in (Zofka et al., 2016), initially introduced in (Treiber et al., 2000). To achieve a modular system, the function is implemented using the Robot Operating System (ROS) framework (Quigley et al., 2009). The system comprises six modules, as shown in Fig. 3: a sensor fusion module to join the information from the three sensors, a tracking algorithm to keep track of occluded traffic participants, a filter module to extract the relevant objects, a routing algorithm, a localization part to create an estimate of the own global position using odometry information, and finally, a trajectory module, planning a trajectory with a velocity profile.

The routing algorithm uses a high-definition map to extract the road topology. The relevant objects identified within the object filter are projected onto the path of the ego vehicle. Thereby, a distance and differential velocity can be calculated to be used

[1]CarMaker from IPG Automotive in version 8.0.2

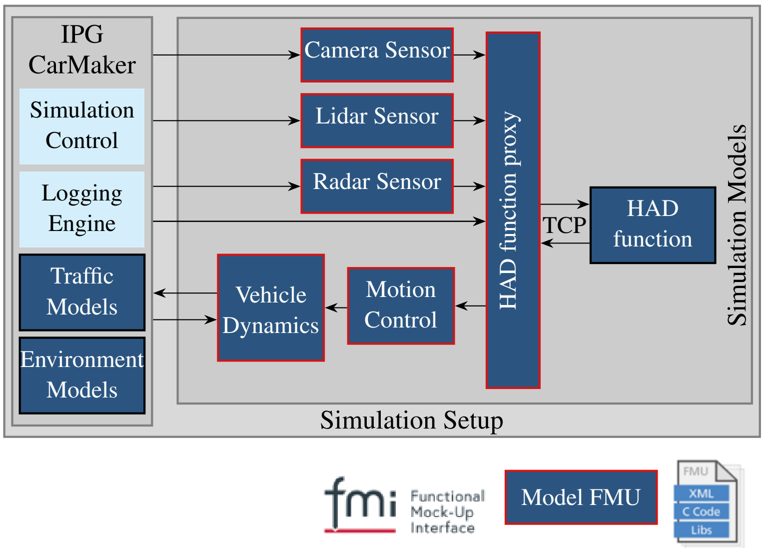

Figure 2: Modular structure of the simulation environment.

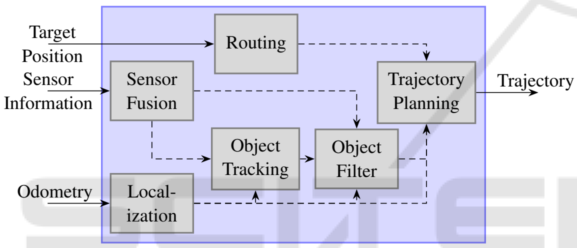

Figure 3: Modular structure of the driving function.

within the IDM. Filtering is done considering the type of an object, as well as its position relative to the road network and the ego vehicle. Moreover, the curvature of the path is considered by converting it to a velocity limit using maximum lateral acceleration. This velocity limit is treated as a separate object for the IDM to achieve smooth cornering behavior.

A gateway architecture was used to comply with the standardized FMI/OSI interface described in section 3.2 while maintaining the platform's independence. Thereby, a TCP proxy was integrated into the simulator as an FMU. The proxy forwards the messages via TCP to the communication layer of the driving function, where the messages are then converted into equivalent ROS messages. With this, the driving function can be run within a docker environment.

3.4 Logical Scenario

As shown in Fig. 4, two logical intersection scenarios are used for the experiments. Each scenario consists of an ego vehicle (E), a pedestrian (P), a second car (C), and a truck (T). Both scenarios vary in the actors' starting position and maneuvers. In scenario A,

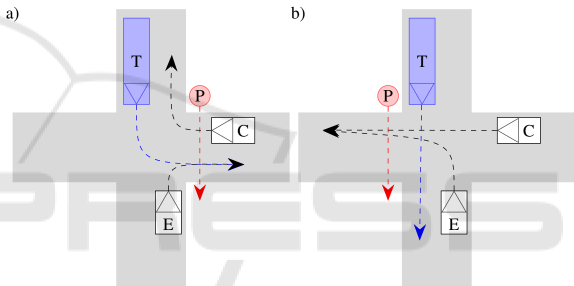

Figure 4: Two experimental scenarios: a) ego turns right, b) ego turns left. C: car, E: ego vehicle, P: pedestrian, T: truck.

the ego vehicle is turning right and, therefore, crosses the trajectories of the pedestrian and the truck but not the car's trajectory, whereas, in scenario B, the ego vehicle is turning left and crosses the trajectories of all three adversary traffic participants. The two scenarios lead to different behaviors of the ego vehicle since it reacts to other participants blocking its route. The following ranges were chosen as parameter ranges, for which the optimal parameter sets have to be found during the scenario exploration:

- **Pedestrian Delay:** The pedestrian waits for a given time $t_{P_{delay}}$ in s before crossing the road, where $t_{P_{delay}} \in \{0.0, ..., 7.0\}$. 50 samples with a step size of 0.14 s were taken.
- **Ego Position:** The ego vehicle starts at a given s-coordinate $s_{E_{start}}$ in m along the road, where $s_{E_{start}} \in \{27.99, ...77.99\}$. 250 samples with a step size of 0.2 m were taken.
- **Car Speed:** The maximum speed $v_{C_{max}}$ that the other car is allowed to achieve in m/s, where $v_{C_{max}} \in \{12.5, ..., 30.0\}$. 50 samples with a step

size of $0.35\,\mathrm{m/s}$ were taken.

This setup results in 625.000 scenarios. If each scenario takes approximately 30 s for simulation execution and calculation of criticality metrics, the execution of all 625.000 scenarios takes more than 217 days of non-stop simulation on a single machine. Additionally, the simulation time grows exponentially, with scenarios getting more complex by additional parameters and parameter ranges.

3.5 Optimization Problem

The goal of the scenario exploration is to find all critical scenarios involving the ego vehicle and the pedestrian, the most vulnerable road user (VRU) in this scenario. Therefore, the criticality between the ego vehicle and the pedestrian is optimized. The criticality is measured by criticality metrics, and the optimization aims to find scenarios evaluated to what degree they contain a potentially critical situation. Five criticality metrics were utilized as the objective function to be optimized by the Bayesian optimization:

- **Euclidean Distance:** Direct distance between the center of mass of two vehicles.
- **Trajectory Distance:** Distance between two traffic participants along their trajectories and road network.
- **Worst-time-to-collision (WTTC):** Metricetric based on time-to-collision (TTC) (Hayward, 1972), but without the TTC's limitation to car following scenarios (Wachenfeld et al., 2016).
- **Gap Time (GT):** The predicted distance in time between the two traffic participants crossing an intersection (Allen et al., 1978).
- **Post-encroachment-time (PET):** The actual distance in time between the two traffic participants crossing an intersection (Allen et al., 1978).

All metrics above require to minimize their output value to optimize the criticality within the logical scenario.

3.6 Experiments and Results

All experiments evaluate the criticality of the scenario regarding the ego vehicle and the pedestrian as the most VRU in this scenario. Some scenarios might lead to critical situations between the ego vehicle and other traffic participants. However, these scenarios are neglected. In general, different metrics cannot be compared directly, e.g., a critical scenario in Fig 6 b) which is indicated by a red dot is not equally critical to a red colored scenario in Fig 6 c).

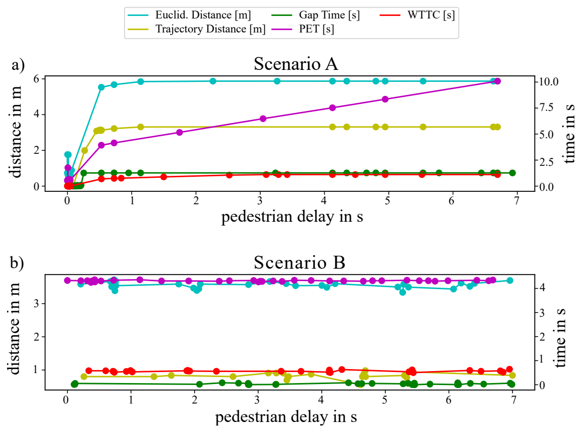

Figure 5: Bayes optimization results of both scenarios on a one-dimensional parameter space. The metrics used for optimization are Euclidean and trajectory distance, gap time, post-encroachment-time, and worst-time-to-collision.

3.6.1 Experiment 1

In the first set of experiments, only the **pedestrian delay** is varied and optimized throughout all simulations, with a set value of $s_{E_{start}} = 60.0\,\mathrm{m}$ in scenario A, $s_{E_{start}} = 67.0\,\mathrm{m}$ in scenario B, and $v_{C_{max}} = 15.0\,\mathrm{m/s}$ in both scenarios. The results are shown in Fig. 5 a) for scenario A and in b) for scenario B. In scenario A, critical scenarios are found for a delay near 0.0 s, and for all metrics except PET, there are no changes in criticality for a delay over approximately 1.5 s. Although PET values change after that, scenarios are not critical since the result is growing. Further, (Allen et al., 1978) set the threshold for critical scenarios to $PET < 1.5\,\mathrm{s}$. The used criticality metrics in scenario B indicate no change in criticality for a varying pedestrian delay, and therefore, the pedestrian delay has no influence on the outcome of scenario B for the chosen values of the other two parameters.

3.6.2 Experiment 2

In the second setup, all three parameters are varied and optimized as described in section 3.1 and 3.4. The results for these experiments are shown in Fig. 6, where a) shows results for the Euclidean distance, b) trajectory distance, c) post-encroachment-time, d) worst-time-to-collision, and e) gap time. In both scenarios, the **car speed** seems to have no or almost no visible influence on simulation results regarding the criticality of the pedestrian's situation. Therefore, a three-dimensional plot can be reduced to the two dimensions of **pedestrian delay** and **ego s-coordinate**. However, this does not mean that there is no influence at all, and outliers or deviations in the plot that are not congruent with other values around them might be influenced by the car's speed. An explanation for the

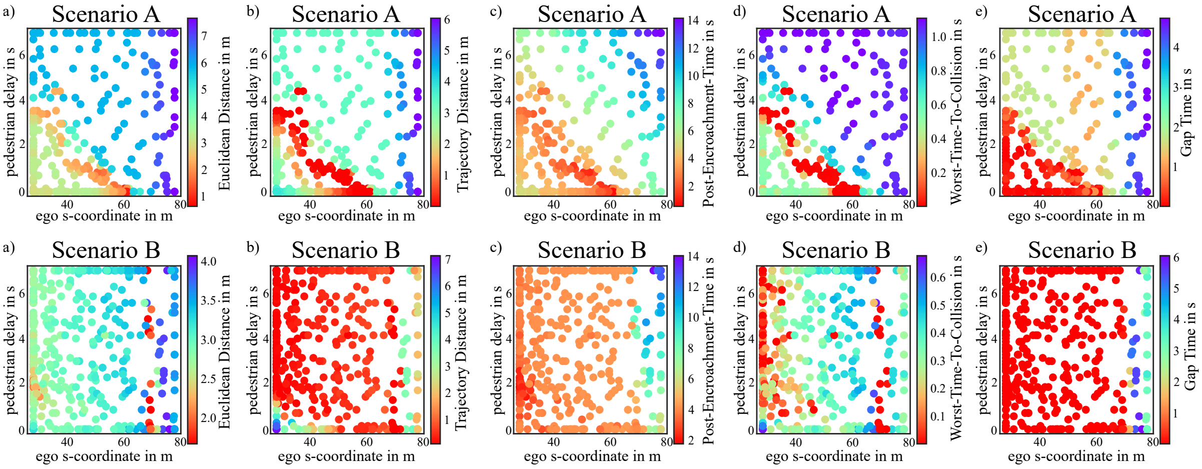

Figure 6: Bayes optimization results of both scenarios on a three-dimensional parameter space. The metrics used for optimization are Euclidean (a) and trajectory (b) distance, post-encroachment-time (c), worst-time-to-collision (d), and gap time (e) and are shown in a two-dimensional space since the variable car speed has almost no influence on the scenario outcome.

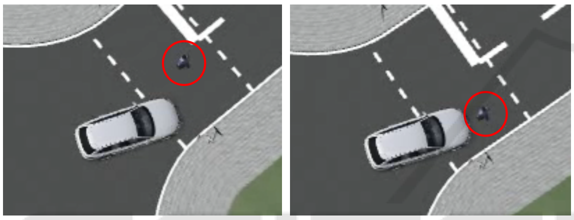

Figure 7: Screenshot from a less critical scenario (left) and critical scenario where the ego vehicle almost hits the pedestrian (right).

lack of influence could be, that in scenario A, the ego vehicle's trajectory and the car's trajectory are not intersecting, and the ego has no need to react to the car. Also, the other car is starting close to the intersection in both scenarios, it might simply not have had enough time to accelerate until it reaches the intersection and always pass the ego vehicle at the same speed.

In general, scenario A has three different outcomes regarding the criticality of the pedestrian's situation: the ego vehicle reaches the intersection and stops for the pedestrian who is crossing the street. This result can be observed in scenarios at the bottom left corner in Fig. 6 a)-e) where the criticality decreases for most metrics. The second result consists of the ego vehicle passing the intersection before the pedestrian or the truck, which are mostly scenarios at the top half and right half in Fig 6 a)-e). The last variant are scenarios where the ego vehicle reaches the intersection right after the truck and, therefore, cannot see the pedestrian. These scenarios are the critical scenarios around the diagonal near the bottom left corner in Fig. 6 a)-e).

In scenario B, the ego vehicle not only reacts to the pedestrian and the truck but also the other car. The outcomes of scenario B are the same as in scenario A. However, they are distributed differently over the parameter space: the less critical area in the bottom left corner, which can be clearly observed in Fig. 6 b) and d) results from a parameter set, where the ego vehicle intersects the pedestrian's path after they crossed the road. The left part of Fig. 7 shows a screenshot of this situation taken during the simulation. On the right side, where the s-coordinate has its highest value, the ego vehicle passes the intersection before the pedestrian, followed by a critical line around 70 m with near-collisions. In the middle and left part of the s-coordinate, the ego vehicle waits for the pedestrian to pass, and the critical cluster at delay 2 s results from interference with the truck.

The comparison of the results of both scenarios leads to the following conclusions:

- Scenario A has a higher variance in criticality than scenario B,
- car speed has no recognizable influence in scenario A and almost no influence in B,
- some metrics are more sensitive, e.g., trajectory distance and gap time, and
- some metrics lead to similar patterns in criticality, e.g., bottom left corner in scenario A or scenarios with a small ego vehicle s-coordinate in scenario B.

3.6.3 Experiment 3

In the third setup, the pedestrian delay $t_{P_{delay}}$ is replaced by a new variable $y_{C_{start}}$ to see if the car speed has more influence on the scenario outcome:

- **Car Position:** The y-coordinate $y_{C_{start}}$ in m fulfils

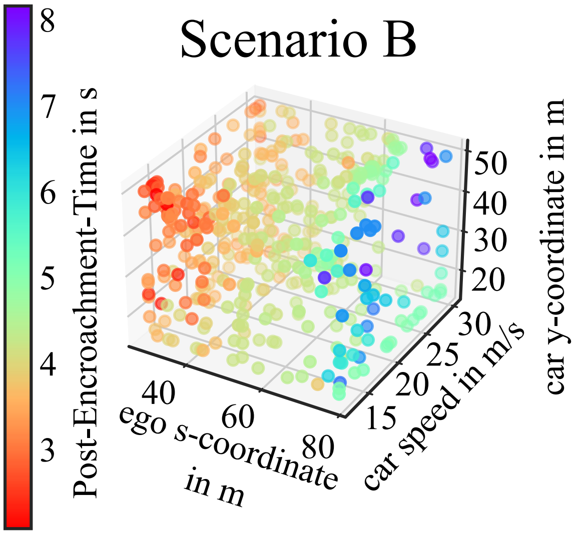

Figure 8: Exemplary Bayes optimization results of scenario B on a three-dimensional parameter space for the metric post-encroachment-time.

$y_{C_{start}} \in \{15.0, \ldots, 50.0\}$. 50 samples with a step size of 0.7 m were taken.

All three variables are varied and optimized, and the five previously mentioned metrics are used. The results for scenario A do not show any influence of the car's speed or y position on the outcome. This is not surprising since there is no trajectory intersection between the ego vehicle and the car. The results show, the outcome only depends on the s-coordinate. As Fig. 8 shows, in scenario B the ego s-coordinate still has the most influence on the outcome. However, in some areas in the parameter space car speed and y-coordinate also affect the scenario criticality.

3.7 Results

In our experiments, we could show that even though different metrics were used, they led to similar critical scenario clusters although, these metrics are not comparable in the severity of the measured criticality and their sensitivity. Moreover, our approach led to a reduction of the amount of necessary simulation: instead of executing more than half a million scenarios, only about 430 were executed in experiments 2 and 3, respectively. Additionally, we were able to show that the variable *car speed* has no influence in scenario A and can be neglected to reduce the number of scenarios or replaced by another variable with more influence.

4 CONCLUSION AND FUTURE WORK

In this work, we used an optimization algorithm to find critical scenarios for the develoing and testing of automated and highly automated driving systems. Bayes optimization with Gaussian process was utilized in combination with five criticality metrics from the automotive domain to calculate the process output. This approach was used in two different experiments and evaluated accordingly.

Derived from the evaluation of the experiments and the results of this work, additional questions arise. Results of the same scenario, models, criticality metrics, and the same driving function could be used by different simulation tools and compared regarding their deviation. Furthermore, it is harder to measure criticality for other scenarios, i.e., scenarios with more traffic participants. Our experiments only focused on criticality metrics between the ego vehicle and the most VRU, the pedestrian. However, such a choice might not always be obvious or changing during one scenario, e.g., in urban rush hour traffic with a high density of traffic participants. Metrics to evaluate the relation between the ego and more than one adversary traffic participant or the whole scenario situation are needed to make more objective conclusions. In future work, the problem of finding more objective metrics for scenarios will be approached to be able to find critical situations between the ego vehicle and the sum of all other traffic participants.

ACKNOWLEDGEMENTS

This research is funded by the "Simulations-basiertes Entwickeln und Testen von automatisiertem Fahren (SET Level)-Simulation-Based Development and Testing of Automated Driving," a successor project to the project "Projekt zur Etablierung von generell akzeptierten Gütekriterien, Werkzeugen und Methoden sowie Szenarien und Situationen zur Freigabe hochautomatisierter Fahrfunktionen (PEGASUS)" and a Project in the PEGASUS Family, promoted by the German Federal Ministry for Economic Affairs and Energy (BMWi) under the grant number 19 A 19004.

REFERENCES

Abeysirigoonawardena, Y., Shkurti, F., and Dudek, G. (2019). Generating adversarial driving scenarios in high-fidelity simulators. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8271–8277. IEEE.

Allen, B. L., Shin, B. T., and Cooper, P. J. (1978). Analysis of traffic conflicts and collisions. Technical report.

ASAM OSI (2021). ASAM OSI® (Open Simulation Interface). Accessed: Dec. 15, 2021.

Bach, J., Otten, S., and Sax, E. (2016). Model based scenario specification for development and test of automated driving functions. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 1149–1155. IEEE.

Baumann, D., Pfeffer, R., and Sax, E. (2021). Automatic generation of critical test cases for the development of highly automated driving functions. In *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, pages 1–5. IEEE.

Bussler, A., Hartjen, L., Philipp, R., and Schuldt, F. (2020). Application of evolutionary algorithms and criticality metrics for the verification and validation of automated driving systems at urban intersections. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 128–135. IEEE.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16.

dSPACE (2021). SIMPHERA - Web-based solution for simulation and validation in autonomous driving development. Accessed: Jan. 19, 2022.

Gartner (2021). Top strategic technology trends for 2022. https://www.gartner.com/en/information-technology/insights/top-technology-trends. Accessed: 2021-10-25.

Greenhill, S., Rana, S., Gupta, S., Vellanki, P., and Venkatesh, S. (2020). Bayesian optimization for adaptive experimental design: A review. *IEEE Access*, 8:13937–13948.

Hayward, J. C. (1972). Near miss determination through use of a scale of danger. In *Unknown*. Publisher: Pennsylvania State University University Park.

IPG Automotive GmbH (2021). CarMaker. Accessed: Dec. 20, 2021.

King, C., Braun, T., Braess, C., Langner, J., and Sax, E. (2021). Capturing the variety of urban logical scenarios from bird-view trajectories. In *VEHITS*, pages 471–480.

Linnhoff, C., Rosenberger, P., and Winner, H. (2021). Refining object-based lidar sensor modeling — challenging ray tracing as the magic bullet. *IEEE Sensors Journal*, 21(21):24238–24245.

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., and Wießner, E. (2018). Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE.

Menzel, T., Bagschik, G., and Maurer, M. (2018). Scenarios for development, test and validation of automated vehicles. In *2018 IEEE Intelligent Vehicles Symp. (IV)*, pages 1821–1827. IEEE.

Otten, S., Bach, J., Wohlfahrt, C., King, C., Lier, J., Schmid, H., Schmerler, S., and Sax, E. (2018). Automated assessment and evaluation of digital test drives. In *Advanced Microsystems for Automotive Applications 2017*, pages 189–199. Springer.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A. Y., et al. (2009). Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan.

Schönemann, V., Winner, H., Glock, T., Otten, S., Sax, E., Boeddeker, B., Verhaeg, G., Tronci, F., and Padilla, G. G. (2018). Scenario-based functional safety for automated driving on the example of valet parking. In *Future of Information and Communication Conference*, pages 53–64. Springer.

Schütt, B., Steimle, M., Kramer, B., Behnecke, D., and Sax, E. (2021). A taxonomy for quality in simulation-based development and testing of automated driving systems. *arXiv preprint arXiv:2102.06588*.

Steimle, M., Menzel, T., and Maurer, M. (2021). Towards a consistent terminology for scenario-based development and test approaches for automated vehicles: A proposal for a structuring framework, a basic vocabulary, and its application. *arXiv preprint arXiv:2104.09097*.

Treiber, M., Hennecke, A., and Helbing, D. (2000). Congested traffic states in empirical observations and microscopic simulations. *Phys. Rev. E*, 62:1805–1824.

Ueno, T., Rhone, T. D., Hou, Z., Mizoguchi, T., and Tsuda, K. (2016). Combo: an efficient bayesian optimization library for materials science. *Materials discovery*, 4:18–21.

Wachenfeld, W., Junietz, P., Wenzel, R., and Winner, H. (2016). The worst-time-to-collision metric for situation identification. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 729–734. IEEE.

Zofka, M. R., Kuhnt, F., Kohlhaas, R., Rist, C., Schamm, T., and Zöllner, J. M. (2015). Data-driven simulation and parametrization of traffic scenarios for the development of advanced driver assistance systems. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 1422–1428. IEEE.

Zofka, M. R., Kuhnt, F., Kohlhaas, R., and Zöllner, J. M. (2016). Simulation framework for the development of autonomous small scale vehicles. In *2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots, SIMPAR 2016, San Francisco, CA, USA, December 13-16, 2016*, pages 318–324.