2020 IEEE Intelligent Vehicles Symposium (IV)
October 20-23, 2020. Las Vegas, USA

Automatic Generation of Road Geometries to Create Challenging Scenarios for Automated Vehicles Based on the Sensor Setup*

Thomas Ponn[1], Thomas Lanz[1] and Frank Diermeyer[1]

Abstract— **For the offline safety assessment of automated vehicles, the most challenging and critical scenarios must be identified efficiently. Therefore, we present a new approach to define challenging scenarios based on a sensor setup model of the ego-vehicle. First, a static optimal approaching path of a road user to the ego-vehicle is calculated using an A* algorithm. We consider a poor perception of the road user by the automated vehicle as optimal, because we want to define scenarios that are as critical as possible. The path is then transferred to a dynamic scenario, where the trajectory of the road user and the road layout are determined. The result is an optimal road geometry, so that the ego-vehicle can perceive an approaching object as poorly as possible. The focus of our work is on the highway as the Operational Design Domain (ODD).**

I. INTRODUCTION

For the introduction of automated vehicles (AV) where the vehicle is responsible for performing the driving task (automation level 3 and higher according to SAE [1]), their safety must be assessed. The aim is to exceed a minimum level of safety that has not yet been defined. However, the procedure for the efficient and economical implementation of this safety verification is an open problem in automotive engineering. A distance-based procedure based exclusively on real tests is no longer economically feasible due to the enormous effort involved. According to WACHENFELD AND WINNER [2], approximately 6.6 billion kilometers are required under representative conditions to determine the safety level of an Autobahn chauffeur on German motorways.

In real-life road traffic, the proportion of comparatively simple situations that do not provide any added value for proof of safety is high, which is why the so-called scenario-based approach (like in the German funded project PEGASUS [3]) is restricted specifically to important and relevant scenarios. While the number of tests performed by virtual simulation is rising and the quality and performance of simulation is improving, even with this tool only a limited number of tests can be performed. But due to continuous parameters and an open parameter space, theoretically an infinite number of test scenarios can be defined [4]–[7].

From this infinite number of possible test cases, one must select a manageable amount that is especially meaningful. These particularly good and challenging test cases are also called edge or corner cases. This is explicitly important for the type approval of automated vehicles because only a very limited number of tests can be carried out for the economic implementation of the type approval. An automated vehicle consists of several modules [8] and test cases can be defined that represent a specific challenge for a certain module. In the context of this publication, the most challenging scenarios for the perception module of AV, especially for the use case of driving on German motorways, shall be defined. Because there are various possibilities in the implementation of an AV (e.g. different sensor setups), the efficient selection of challenging scenarios must be system-specific.

This contribution therefore presents a novel approach for a system-specific definition of particularly challenging scenarios for the perception module of AV. This work is based on a previously published work of the author [9]. Additionally, the presented approach is part of the overall methodology of the author to identify relevant scenarios for the type approval of AV presented in [10].

II. RELATED WORK

This chapter provides an overview of different techniques for defining and identifying scenarios for testing AV. Subsequently, it is shown in which aspects the present work represents an extension to the current state of the art.

A. Testing of Automated Vehicles

For the assessment of AV, the scenario-based approach already mentioned in Section I is used. The basic idea of the scenario-based approach is to verify the most relevant and critical scenarios that could occur instead of all possible ones so that the overall driving distance can be significantly reduced [11, p.46]. The basic assumption is that the greater part of everyday driving is irrelevant, e. g. driving on a straight and empty highway.

There are various approaches in literature on how to generate relevant scenarios:

Traffic analysis: For example, accident databases can be used to identify especially critical scenarios when analyzing real traffic situations [12]–[14]. In addition, a criticality metric can be used to filter out critical situations from test drives that represent relevant scenarios in which no accident occurred [15]. The exposure of individual scenarios can also be taken into account in order to make a more precise statement about the safety level of the vehicle [16].

Combinatorial testing and Design of Experiments: For simple systems, an n-wise combination of the parameters (i.e. each parameter value is combined with each parameter value) is still feasible, but for higher

*The research project was funded and supported by TÜV SÜD Auto Service GmbH.

[1] Thomas Ponn, Thomas Lanz and Frank Diermeyer are with the Institute of Automotive Technology, Technical University of Munich, 85748 Garching, Germany {ponn,diermeyer}@ftm.mw.tum.de,ga87veq@mytum.de

978-1-7281-6673-5/20/$31.00 ©2020 IEEE

degrees of automation this is no longer feasible even using virtual simulation [17]. A reduced number of combinatorial test cases based on scenario importance is part of [18]. SCHULDT ET AL. [19] discretize value-continuous parameters into equivalence classes and therefore reduce the number of test cases for a necessary test coverage.

Search based methods: All search-based approaches presented in the following have in common that they require a (simplified) model of the driving function. BEN ABDESSALEM ET AL. [20] use Evolutionary Algorithms to generate critical scenarios for a vision-based automatic emergency braking system. In [21] Monte Carlo Tree Search and Deep Reinforcement Learning are used to find scenarios that lead most likely to a system failure. In order to define critical scenarios, Evolutionary Algorithms are used in [22]. Thereby the safe drivable area, i. e. the solution space for the planning module of the automated vehicle, is minimized. Differential Evolution and Particle Swarm Optimization are used in [23] to generate critical scenarios for a Emergency Braking Assist. For this purpose, surrogate models of the overall system are implemented to reduce the computation time during optimization.

Challenging scenarios: Scenarios that are particularly difficult to master are referred to as challenging scenarios. The basic assumption here is that an increased difficulty of the test case results in an increased probability of the occurrence of system failures. GAO ET AL. [24] define a complexity index based on the Analytic Hierarchy Process that characterizes the traffic situation based on the difficulty. In addition, it is shown that a Lane Departure Warning System fails more frequently in more complex situations. WANG ET AL. [25] propose a new method to determine the complexity of a traffic scene by the quantified road semantic complexity and the traffic element complexity. The former evaluates the static environment and the latter the dynamics of other road users according to their difficulty for an AV. However, the presented metric is not validated.

For reasons of completeness, the formal methods will be presented as an alternative concept to testing, although this approach will not be discussed in the further course of the work. The verification of the system by formal methods as for example in [4], [26]–[30] aims at mathematically proving the safety of AV. If formal methods without assumptions and restrictions are applicable to the overall system, any kind of simulated or real test would be obsolete. However, so far there is no procedure that achieves this at the overall system level and therefore tests will also be necessary in the future to demonstrate the safety of AV.

B. Contributions

The first question to be answered is why the existing procedures are not sufficient. The methods of traffic analysis require a large amount of data, which is very cost-intensive to generate. The problem with the use of accident databases is that these contain almost exclusively accidents of human drivers. A direct transfer to AVs is not possible. Combinatorial tests have the disadvantage that they do not reduce the number of tests to the required extent. Search-based methods have the disadvantage that a simulation model of the entire system is necessary, with which a large number of simulations must be carried out. On the one hand, high-fidelity simulation models with high computational effort must be used for a realistic evaluation of safety. On the other hand, when AVs are certified by a technical service, not all simulation models may be available for the technical service and search-based approaches may not be applicable. The current literature with regard to challenging scenarios focuses mainly on the evaluation of existing scenarios and not on the definition of new challenging scenarios. Therefore, a method that actively defines challenging scenarios is presented here. The focus will be on the perception of the system and the approach is especially suitable for technical services to support the certification of automated vehicles without having all simulation models of the system available.

The basis for the method presented here is an existing framework from our previous work on the calculation of a 3D grid with detection probabilities (Fig. 1). Based on the used sensor setup, environmental conditions and the object to be detected, the sensor coverage and a three-dimensional grid with detection probabilities are calculated with phenomenological sensor models. With the existing framework, weak spots of the sensor setup can already be identified and the focus of the safety assessment of the vehicle can be set on these weak spots. Building on this, the present paper shows how a path finding algorithm can be used to determine a worst case approaching path of an object towards the ego-vehicle and to derive an optimal road geometry for this test scenario. The worst case approaching path is the path with which an object must approach an ego-vehicle in order to be perceived as poorly as possible. This constellation represents the greatest challenge for the perception of the vehicle and can lead to critical scenarios that are of crucial importance for the safety assessment.

III. METHODOLOGY

This chapter describes the individual blocks drawn in blue in Fig. 1. A 3D grid with the detection probabilities P_D serves as the basis. This means that a probability is assigned to each node in the grid with which an object (e.g. another vehicle) is correctly detected by the ego-vehicle under previously defined environmental conditions. All nodes outside the field of view of the sensors have a detection probability of zero. The basic idea is to iterate through the nodes of the grid in order to find the optimal approaching path $\mathbf{x}_\mathrm{oap}$ where the cost between the start node v_1 (start position of the object[1]) and the end node v_N (position of ego-vehicle) is minimal. The detection probability P_D and the length of the path d are considered as assessment quantities

[1] The object approaching the ego-vehicle is further referred to as the challenger.

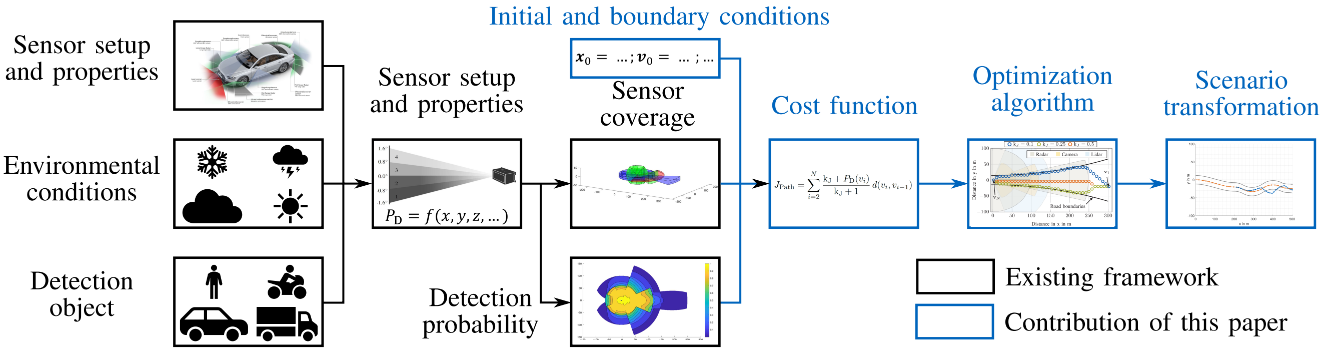
Sensor setup and properties
Environmental conditions
Detection object
Sensor setup and properties
$P_D = f(x, y, z, ...)$
Initial and boundary conditions
$x_0 = ... ; v_0 = ... ; ...$
Sensor coverage
Detection probability
Cost function
$J_{\mathrm{Path}} = \sum_{i=2}^{N} \frac{k_J + P_D(v_i)}{k_J + 1} d(v_i, v_{i-1})$
Optimization algorithm
Scenario transformation
Existing framework
Contribution of this paper

Fig. 1: Connection between already published framework [9] and the contribution of the current publication.

of $\mathbf{x}_{\mathrm{oap}}$. We want to define particularly challenging scenarios and therefore the path should be as short as possible so that the required reaction time of the ego-vehicle to the challenger is as short as possible. At the same time P_{D} of $\mathbf{x}_{\mathrm{oap}}$ should be minimal so that the probability is increased that the ego-vehicle detects the challenger late. This means that a path as short as possible with a low detection probability is optimal for our application.

A. Cost Function

First, we need a mathematical representation of the distance d and the detection probability P_{D}. For d, the Euclidean distance between the current node v_j and the previous node v_{j-1} is used. For the detection probability $P_{\mathrm{D}}(v_j)$, the existing values of the grid from our previous work in [9] are used for each node v_j. It is advantageous for the cost function if the ratio between the detection probability $P_{\mathrm{D}}(v_j)$ and the distance $d(v_j, v_{j-1})$ can be adjusted. This is done with the weighting factor k_{J}. Equation 1 shows the cost function J_{Node} that describes the cost of the approaching path between two nodes v_j and v_{j-1}.

$$J_{\mathrm{Node}} = \frac{\mathrm{k_J} + P_{\mathrm{D}}(v_j)}{\mathrm{k_J} + 1} d(v_j, v_{j-1}) \tag{1}$$

Note that for all $\mathrm{k_J}$ the distance d is taken into account, since also diagonal steps are possible in the orthogonal grid and these are longer than horizontal and vertical steps. The total cost of the path J_{Path} consisting of N nodes can be calculated according to Equation 2.

$$J_{\mathrm{Path}} = \sum_{j=2}^{N} \frac{\mathrm{k_J} + P_{\mathrm{D}}(v_j)}{\mathrm{k_J} + 1} d(v_j, v_{j-1}) \tag{2}$$

The choice of $\mathrm{k_J}$ has a considerable influence on the result, which is examined in more detail in Section IV. The higher $\mathrm{k_J}$ in Equation 2, the less P_{D} taken into account and thus the greater focus given to the shortest possible approaching path. Preliminary investigations in [31] have revealed that values in the range of $0.1 - 1$ are promising for $\mathrm{k_J}$.

B. Optimization Algorithm

This section describes the calculation of the path through the three-dimensional grid with the lowest cost. It also shows how to reduce the number of valid nodes in the search space. Within our work, two different optimization algorithms are examined. First, the A-star (A*) algorithm, a heuristic search algorithm [32, chap. 6.3.2], and second, the Ant Colony Optimization with problem-specific adjustments, which is one of the nature-inspired path finding algorithms [33]. In our preliminary investigations [31], the A* algorithm proved to be superior to the Ant Colony Optimization in terms of runtime, quality and simple parameter selection. In this paper, we will therefore only discuss the A* algorithm.

The following description of the functionality of the A* algorithm is based on [32, chap. 6.3.2]. One characteristic of the A* algorithm that indicates its suitable use is that the optimal path is always found between the start and end nodes, if one exists. The cost of the path for the A* algorithm via v_j is determined by a heuristic evaluation function according to Equation 3,

$$f(v_j) = g(v_j) + h(v_j) \tag{3}$$

where $g(v_j)$ is the cost of the path from the start node v_1 to the current node v_j and $h(v_j)$ is an estimate of the remaining cost from the current node v_j to the end node v_N. Thus f(v_j) approximates the total cost of the path from v_1 via v_j to v_N. For $g(v_j)$ the cost function from Equation 2 from v_1 to v_j can be used directly. When selecting $h(v_j)$, note that $h(v_j)$ must not overestimate the actual cost from v_j to v_N. Therefore, $h(v_j)$ uses the cost function from Equation 2 from v_j to the target node v_N using a vanishing detection probability ($P_{\mathrm{D}}(v_j) = 0\,\%$ for all nodes from $j+1$ to N). Thus, the A* algorithm defines an iterative process, which begins at v_1 and finds $\mathbf{x}_{\mathrm{oap}}$ to the desired end node v_N.

In order to increase the efficiency of the algorithm, the number of nodes in the 3D grid can be limited to the relevant nodes. Only approach paths to the ego-vehicle from the front are considered, therefore all nodes behind the ego-vehicle can be removed from the grid. In addition, a maximum distance in x-direction (vehicle longitudinal direction) of 300 m is defined, which corresponds approximately to the maximum range of sensors in the automotive industry. In addition, this paper considers the use case of German motorways. In Germany, the design of motorways is clearly regulated in the German Motorway Construction Guideline [34]. This means, for example, that minimum curve radii as well as minimum radii of crest and hollow are specified. Based on the minimum values of these parameters, only nodes within a theoretically possible motorway course need to be considered. All other nodes can be removed from the grid. For the design class EKA 1 B standard cross section 43,5 (four lanes in each driving direction) according to the German Motorway Construction Guideline [34], the valid search space shown by section planes in Fig. 2 is obtained.

In summary, the optimization algorithm calculates an optimal approaching path $\mathbf{x}_{\mathrm{oap}}$ based on a grid with detection probabilities and a cost function. This represents the optimal relative approach of a challenger to the ego-vehicle so that it can perceive the challenger as poorly as possible.

C. Scenario Transformation

If $\mathbf{x}_{\mathrm{oap}}$ from the challenger to the ego-vehicle is calculated with the A* algorithm, then this represents the optimal course that the challenger must choose when the ego-vehicle is stationary. Since in reality both vehicles move, the trajectories of the ego-vehicle and the challenger have to be calculated from $\mathbf{x}_{\mathrm{oap}}$, which we call scenario transformation. If both vehicles execute the calculated trajectory (which also defines the street geometry of the scenario), $\mathbf{x}_{\mathrm{oap}}$ results as a relative movement between the vehicles. To achieve this, we need to make assumptions about the velocities of the ego-vehicle v_{ego} (drives with recommended speed on German motorways) as well as the challenger v_{ch} (slow moving vehicle) and set maximum allowable values for the pitch and yaw angles and rates of both vehicles, respectively (Table I).

For a better transformation into a valid road profile, $\mathbf{x}_{\mathrm{oap}}$ is interpolated so that the data for the defined relative speed between ego-vehicle and challenger is available for each time step i ($\Delta t = 0.01\,\mathrm{s}$). In addition, $\mathbf{x}_{\mathrm{oap}}$ is smoothed because no driving physics is taken into account during path optimization and the path can therefore contain corners. A global and a relative coordinate system are used for the calculation of the trajectories (Fig. 3). Subsequently, $\mathbf{x}_{\mathrm{oap}}$ must be converted from the relative to the global coordinate system. The aim of the algorithm is to determine the changes in position in every time step i of the ego-vehicle $\Delta\mathbf{x}_{\mathrm{ego},i}$ and of the challenger $\Delta\mathbf{x}_{\mathrm{ch},i}$ so that $\mathbf{x}_{\mathrm{ch,rel}}(t)$ corresponds as closely as possible to $\mathbf{x}_{\mathrm{oap}}$ (Fig. 4).

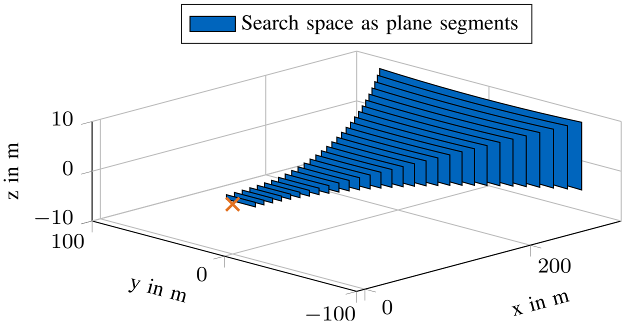

Fig. 2: Valid search space for design class EKA 1 B according to the German Motorway Construction Guideline [34] with minimum curve radius of $R_{\mathrm{min}} = 720\,\mathrm{m}$, minimum crest radius of $H_{\mathrm{K,min}} = 10\,000\,\mathrm{m}$ and minimum hollow radius of $H_{\mathrm{W,min}} = 5700\,\mathrm{m}$. The position of the ego-vehicle is marked with an orange cross.

TABLE I: Assumptions and maximum values used for the scenario transformation algorithm

Symbol	Value	Unit	Description
v_{ego}	130	$\mathrm{km\,h^{-1}}$	ego-vehicle velocity
v_{ch}	80	$\mathrm{km\,h^{-1}}$	challenger velocity
$\dot{\Psi}_{\mathrm{max}}$	0.22	$\mathrm{rad\,s^{-1}}$	max. yaw rate of both vehicles
$\Delta\Psi_{\mathrm{max}}$	0.21	rad	max. yaw angle difference between both vehicles to increase stability of algorithm
$\dot{\Theta}_{\mathrm{max}}$	0.22	$\mathrm{rad\,s^{-1}}$	max. pitch rate of both vehicles
Θ_{max}	6.0	%	max. inclination of highways [34]

At the beginning of the calculation, the ego-vehicle starts at the origin of the global coordinate system and the challenger at v_1 of $\mathbf{x}_{\mathrm{oap}}$ with their defined velocities in positive x-direction of the global coordinate system. It is important that the ego-vehicle is at some stage at the starting position of the challenger because the trajectories of both vehicles must be within the same road. Until this point is reached ($x_{\mathrm{ego,glob}} = x_{\mathrm{ch,start,glob}}$), the road geometry between the ego-vehicle and the challenger can be specified by the ego-vehicle movement. After this point, the ego-vehicle must follow the road course that the challenger has already defined. Thus, when the ego-vehicle reaches the starting position of the challenger, a change in the calculation method of the optimal road course arises. Therefore, the calculation is divided into these two areas (Algorithm 1 and 2).

In the first part of the calculation (Algorithm 1), the position change of the ego-vehicle $\Delta\mathbf{x}_{\mathrm{ego},i}$ in each time step i is determined between the starting point of the ego-vehicle at v_N and the starting point of the challenger at v_1 by means of an interpolation with a smoothing spline. The calculations in Algorithm 1 consist of geometric correlations that are shown simplified in 2D in Fig. 3 and 4. For a detailed formulation of the equations, the interested reader is referred to [31, chap. 4.5]. At the end of the algorithm, the trajectories of both vehicles can be calculated from the constant speeds and the

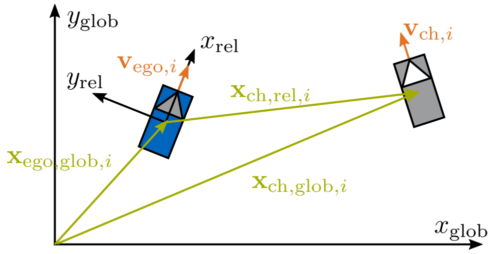
y_{glob}
x_{rel}
y_{rel}
$\mathbf{v}_{\mathrm{ego},i}$
$\mathbf{v}_{\mathrm{ch},i}$
$\mathbf{x}_{\mathrm{ch,rel},i}$
$\mathbf{x}_{\mathrm{ego,glob},i}$
$\mathbf{x}_{\mathrm{ch,glob},i}$
x_{glob}

Fig. 3: Representation of the relative and global coordinate system in time step i.

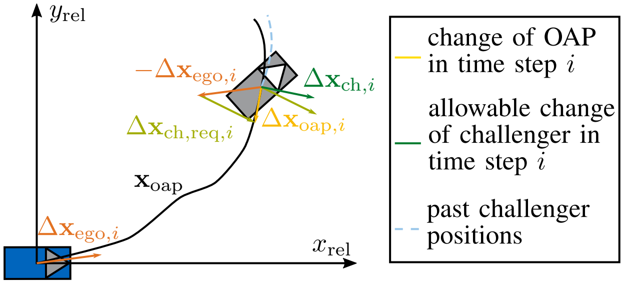
y_{rel}
$-\Delta\mathbf{x}_{\mathrm{ego},i}$
$\Delta\mathbf{x}_{\mathrm{ch},i}$
$\Delta\mathbf{x}_{\mathrm{ch,req},i}$
$\Delta\mathbf{x}_{\mathrm{oap},i}$
$\mathbf{x}_{\mathrm{oap}}$
$\Delta\mathbf{x}_{\mathrm{ego},i}$
x_{rel}
change of OAP in time step i
allowable change of challenger in time step i
past challenger positions

Fig. 4: Geometric relationships for calculating the trajectories in time step i, resulting in the optimal approximation path.

Algorithm 1: Calculation of trajectories until ego-vehicle reaches the start pos. of the challenger at v_1.

```
Input: $\mathbf{x}_{\text{oap}}$, $\Delta\mathbf{x}_{\text{ego},i}$, $v_1$, $v_N$
Parameter: $v_{\text{ego}}$, $v_{\text{ch}}$, $\Delta t$, $\dot{\Psi}_{\max}$, $\Delta\Psi_{\max}$, $\dot{\Theta}_{\max}$, $\Theta_{\max}$
Output: $\mathbf{x}_{\text{ego}}(t)$, $\mathbf{x}_{\text{ch}}(t)$ and road geometry up to $v_1$
$t_i = 0$
while $x_{\text{ego,glob}} \leq x_{\text{ch,start,glob}}$ do
    calc $\Delta\mathbf{x}_{\text{oap,i}}$            // required relative path
    calc $\Delta\mathbf{x}_{\text{ch,req},i}$        // required challenger path
    check $\dot{\Psi}_{\max}$, $\Delta\Psi_{\max}$, $\dot{\Theta}_{\max}$, $\Theta_{\max}$      // of ch.
    calc $\Delta\mathbf{x}_{\text{ch},i}$   // actual challenger pos. change
    $t_i \leftarrow t_i + \Delta t$
end
```

Algorithm 2: Calculation of trajectories after ego-vehicle reaches the start pos. of the challenger at v_1.

```
Input: $\mathbf{x}_{\text{oap}}$, $x_{\text{sp}}$, $v_1$
Parameter: $v_{\text{ego}}$, $v_{\text{ch}}$, $\Delta t$, $\dot{\Psi}_{\max}$, $\Delta\Psi_{\max}$, $\dot{\Theta}_{\max}$, $\Theta_{\max}$
Output: $\mathbf{x}_{\text{ego}}(t)$, $\mathbf{x}_{\text{ch}}(t)$ and road geometry after $v_1$
$t_i = 0$
while $x_{\text{ego,glob}} \leq x_{\text{ch,glob}}$ do
    calc $\Delta\mathbf{x}_{\text{ego,req},i}$      // required ego pos. change
    check $\dot{\Psi}_{\max}$, $\Delta\Psi_{\max}$, $\dot{\Theta}_{\max}$, $\Theta_{\max}$      // of ego
    calc $\Delta\mathbf{x}_{\text{ego},i}$            // actual ego pos change
    calc $\mathbf{x}_{\text{ch,rel},i}$    // relative challenger position
                                // due to pos change of ego
    calc arguments of while-loop of Algorithm 1
end
```

calculated paths $\Delta\mathbf{x}_{\text{ego},i}$ and $\Delta\mathbf{x}_{\text{ch},i}$. In the second part of the calculation (Algorithm 2), the path of ego-vehicle $\Delta\mathbf{x}_{\text{ego},i}$ must be within the road boundaries already defined by the challenger and is therefore given within limits.

In summary, the scenario transformation uses $\mathbf{x}_{\text{oap}}$, which represents the relative movement between the challenger and the ego-vehicle, and calculates the necessary trajectories of the two vehicles under certain assumptions, so that $\mathbf{x}_{\text{oap}}$ results as a relative movement. From the trajectories, the required road geometry can be derived as a surrounding envelope with the width of the standard cross section used.

The calculated trajectory of the challenger as well as the derived road geometry represent the input for the most critical scenario with respect to the sensor setup of the ego-vehicle and can be executed in virtual simulation. The conversion into a simulation compatible format and the execution of the simulation are not part of this publication.

IV. RESULTS

This chapter first presents the results of the calculation of $\mathbf{x}_{\text{oap}}$ using the A* algorithm (Section III-B), and the second part describes the results of the scenario transformation (Section III-C).

A. Optimal Approaching Path with A*

The results shown here are calculated on the basis of the coarse grid (Table II), resulting in a total number of 8 926 nodes. For the calculation of P_{D} of the grid according to [9], good weather conditions and a passenger car as the challenger are used. The ego-vehicle is in the origin, so $\mathbf{v}_N = (0\,\text{m}, 0\,\text{m}, 0.5\,\text{m})$. The challenger is outside the sensor coverage ($P_D = 0\,\%$) at $\mathbf{v}_1 = (300\,\text{m}, -20\,\text{m}, 2\,\text{m})$, so the algorithm can select the optimal entry point in the sensor coverage. In the following, the influence of k_{J} of the cost function from Equation 2 will be examined in more detail. $\mathbf{x}_{\text{oap}}$ for three different values of k_{J} are shown for the x-y-plane in Fig. 5. A detailed representation of the x-z-plane is not provided because k_{J} has no significant influence on it. Also, the choice of the grid size has only a small influence [31, chap. 5].

With varying k_{J}, $\mathbf{x}_{\text{oap}}$ changes significantly. For $k_{\text{J}} = 0.5$ (high distance costs) the algorithm chooses the shortest possible path to the ego-vehicle and P_{D} of the nodes has only a negligible effect. With lower k_{J}, more and more focus is placed on a lower P_{D} and a longer path is accepted. Fig. 6 shows P_{D} as a function of the x-distance. It can be seen that smaller k_{J} lead to lower P_{D}. However, the total length of the approaching path is inversely proportional (Table III).

As a consequence of these results, it can be concluded that small values of k_{J} are preferred for the creation of worst case scenarios regarding the sensor coverage of automated vehicles.

B. Scenario Transformation Results

To demonstrate the results of the scenario transformation, we use the fine grid from Table II, a weighting factor of $k_{\text{J}} = 0.25$, and a starting point of the challenger at $\mathbf{v}_1 = (200\,\text{m}, -20\,\text{m}, 2\,\text{m})$. This results in a scenario duration of $14.06\,\text{s}$ with the assumptions made in Table I. The ego-vehicle covers a distance of $507.7\,\text{m}$ and the challenger $312.4\,\text{m}$. Again, the visualization is reduced to the x-y-plane (Fig. 7). This shows the course of the road when

TABLE II: Used grid sizes with $x_{\min} = 0\,\text{m}$, $x_{\max} = 300\,\text{m}$ and $y_{\min}$, $y_{\max}$, $z_{\min}$ and $z_{\max}$ according to Fig. 2.

Grid	Δx in m	Δy in m	Δz in m
fine	1	1	0.2
middle	2	2	0.4
coarse	4	4	0.8

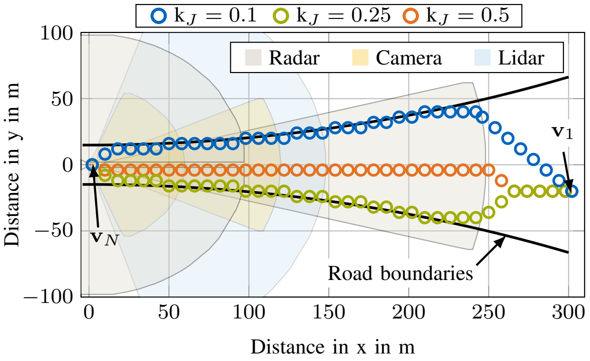

Fig. 5: Optimal approaching path $\mathbf{x}_{\text{oap}}$ with different weighting factors k_{J}.

TABLE III: Length of the optimal approaching path $L(\mathbf{x}_{\mathrm{oap}})$ with different weighting factors k_{J}.

k_{J}	0.1	0.25	0.5
$L(\mathbf{x}_{\mathrm{oap}})$	343.9 m	325.0 m	308.4 m

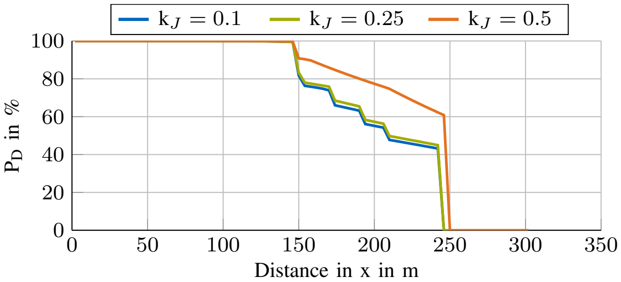

Fig. 6: Detection probability P_{D} with different weighting factors k_{J}. The lengths of the path are summarized in Table III. The high detection probabilities P_{D} of almost 100 % are due to multiple overlapping sensors and the assumed good weather conditions. This can change in adverse weather conditions [9].

both vehicles move, so that $\mathbf{x}_{\mathrm{oap}}$ is obtained as the relative movement between the two vehicles.

Fig. 7 shows the road boundaries as twice the width of the possible offset in y-direction between two vehicles for a standard cross-section of 43,5. The reference represents the ego-vehicle that is always in the middle of the boundaries. From the point of view of the ego-vehicle, the challenger can be to the left and to the right with the highest possible offset to still be on the shared street layout. The highest possible offset in the y-direction for the standard cross section 43,5 is 10.875 m.

While in the x-z-plane no exceedances of the limits occur [31, p. 78], in Fig. 7 for the x-y-plane it can be seen that the challenger is outside the road limits in the region of $x = 400$ m. This means that in this area no road can be defined according to [34] in order to maintain $\mathbf{x}_{\mathrm{oap}}$ between the two vehicles. Apart from these short deviations, however, the procedure developed is suitable for defining the optimal road geometry for particularly challenging scenarios for the perception module of an automated vehicle. Approaches to overcome the limitations of the scenario transformation algorithm are discussed in Section V.

V. Discussion

For a more efficient calculation of $\mathbf{x}_{\mathrm{oap}}$, the grid can be reduced to 2D because the influence and effects in z-direction are negligible. The reason for this lies in the sensor models used to calculate P_{D} of the grid [9]. These show almost no dependence on the z-component. If, however, sensor models are used that consider a dependency in the z-direction, the 3D grid must be used instead.

The most potential for improvement exists in scenario transformation (Section III-C). Fig. 7 shows that the boundary conditions cannot be met over the entire duration of the scenario. The following aspects describe approaches with which compliance with the boundary conditions can be improved:

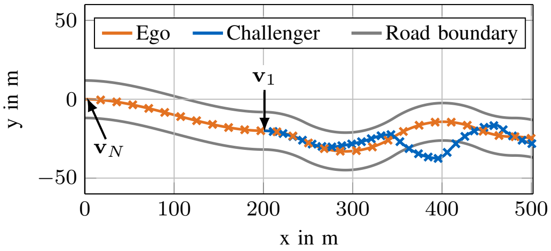

Fig. 7: Calculated optimal road geometry to generate the $\mathbf{x}_{\mathrm{oap}}$ between the two vehicles. From $\mathbf{v}_1$, i.e. from 200 m, the road can theoretically also be defined around the challenger.

- Consideration of vehicle dynamics already during path search in the A* algorithm.
- No exact position specification of the ego-vehicle to the starting point $\mathbf{v}_1$ of the challengers (deviations within one road width tolerable).
- Currently the trajectories are calculated forward in time. This means that starting from the initial distance between the vehicles, both drive forward and get closer with $\mathbf{x}_{\mathrm{oap}}$. A reversal of the process into a reverse simulation can bring advantages, because the more complex calculation of the road geometry (both are on the same road) is already carried out at the beginning of the calculation.

In addition, an adaptive velocity specification of the challenger may be necessary for the simulation execution of the generated critical scenarios. This is necessary if the ego-vehicle deviates from the assumed constant speed and the assumed relative speed between both vehicles must be re-established by the challenger.

VI. Conclusion

This contribution addresses a novel method for the definition of system-specific challenging scenarios for the safety assessment of automated vehicles with the Operational Design Domain of highways. Based on a 3D grid of the sensor coverage with corresponding detection probabilities at the nodes of the grid, the A* algorithm calculates a worst-case approaching path between the ego-vehicle and the challenger, so that the automated vehicle perceives the challenger as poorly as possible. The static approaching path is translated into a dynamic scenario using scenario transformation (Section III-C), which defines the challenger trajectory and road geometry. These scenarios represent the most critical scenarios from the point of view of the automated vehicle perception module and offer significant additional value in the safety assessment of automated driving on highways. In future work, visual obstructions caused by other objects can be considered. In addition, the generated scenario data will be exported directly to an OpenSCENARIO and OpenDRIVE file for automated connection to a simulation tool.

Acknowledgment and Contributions

Thomas Ponn (corresponding author) initiated and wrote this paper. He was involved in all stages of development

and primarily developed the research question as well as the concept. Thomas Lanz wrote his master thesis on sensor modeling and implemented the developed models during his thesis. Frank Diermeyer contributed to the conception of the research project and revised the paper critically for important intellectual content. He gave final approval of the version to be published and agrees to all aspects of the work. As a guarantor, he accepts responsibility for the overall integrity of the paper.

REFERENCES

[1] SAE J3016, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," 2018.

[2] W. Wachenfeld and H. Winner, "The release of autonomous vehicles," in *Autonomous Driving: Technical, Legal and Social Aspects*, M. Maurer, J. C. Gerdes, B. Lenz, and H. Winner, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 425–449. [Online]. Available: https://doi.org/10.1007/978-3-662-48847-8{_}21

[3] Deutsches Zentrum für Luft- und Raumfahrt e. V. (2018) Pegasus-project. Accessed on: 22.10.2019. [Online]. Available: https://www.pegasusprojekt.de/en/home

[4] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 903–918, 2014.

[5] Z. Huang, H. Lam, and D. Zhao, "Sequential experimentation to efficiently test automated vehicles," in *2017 Winter Simulation Conference (WSC)*, 2017, pp. 3078–3089.

[6] P. Junietz, Wachenfeld Walther, K. Klonecki, and H. Winner, "Evaluation of different approaches to address safety validation of automated driving," in *21st IEEE International Conference on Intelligent Transportation Systems, November 4-7, 2018, Maui, Hawaii,*, 2018, pp. 491–496.

[7] T. Menzel, G. Bagschik, and M. Maurer, "Scenarios for development, test and validation of automated vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1821–1827.

[8] C. Amersbach and H. Winner, "Functional decomposition—a contribution to overcome the parameter space explosion during validation of highly automated driving," *Traffic injury prevention*, vol. 20, no. sup1, pp. S52–S57, 2019.

[9] T. Ponn, F. Müller, and F. Diermeyer, "Systematic analysis of the sensor coverage of automated vehicles using phenomenological sensor models," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 879–885.

[10] T. Ponn, C. Gnandt, and F. Diermeyer, "An optimization-based method to identify relevant scenarios for type approval of automated vehicles," in *26th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, National Highway Traffic Safety Administration, Ed., 2019.

[11] J. Mazzega, F. Köster, K. Lemmer, and T. Form, "Testing of highly automated driving functions," *ATZ Worldwide*, vol. 118, no. 10, pp. 44–48, 2016.

[12] F. Fahrenkrog, L. Wang, T. Platzer, A. Fries, F. Raisch, and K. Kompaß, "Prospective Effectiveness Safety Assessment of Automated Driving Functions – From The Method to the Results," in *26th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, National Highway Traffic Safety Administration, Ed., 2019.

[13] J. So, I. Park, J. Wee, S. Park, and I. Yun, "Generating Traffic Safety Test Scenarios for Automated Vehicles using a Big Data Technique," *KSCE Journal of Civil Engineering*, 2019. [Online]. Available: https://doi.org/10.1007/s12205-019-1287-4

[14] L. Stark, M. Düring, S. Schoenawa, J. E. Maschke, and C. M. Do, "Quantifying Vision Zero: Crash avoidance in rural and motorway accident scenarios by combination of ACC, AEB, and LKS projected to German accident occurrence," *Traffic injury prevention*, vol. 20, no. sup1, pp. S126–S132, 2019.

[15] P. Junietz, F. Bonakdar, B. Klamann, and H. Winner, "Criticality metric for the safety validation of automated driving using model predictive trajectory optimization," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 04.11.2018 - 07.11.2018, pp. 60–65.

[16] E. d. Gelder and J. Paardekooper, "Assessment of automated driving systems using real-life scenarios," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 589–594.

[17] T. Ponn, D. Fratzke, C. Gnandt, and M. Lienkamp, "Towards certification of autonomous driving: Systematic test case generation for a comprehensive but economically-feasible assessment of lane keeping assist algorithms," in *VEHITS 2019-Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems*, 2019.

[18] L. Huang, Q. Xia, F. Xie, H. Xiu, and H. Shu, "Study on the test scenarios of level 2 automated vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 49–54.

[19] F. Schuldt, A. Reschka, and M. Maurer, "A method for an efficient, systematic test case generation for advanced driver assistance systems in virtual environments," in *Automotive Systems Engineering II*, H. Winner, G. Prokop, and M. Maurer, Eds. Cham: Springer International Publishing, 2018, pp. 147–175. [Online]. Available: https://doi.org/10.1007/978-3-319-61607-0_7

[20] R. Ben Abdessalem, S. Nejati, L. C. Briand, and T. Stifter, "Testing vision-based control systems using learnable evolutionary algorithms," in *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*, 2018, pp. 1016–1026.

[21] M. Koren, S. Alsaif, R. Lee, and M. J. Kochenderfer, "Adaptive stress testing for autonomous vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1–7.

[22] M. Klischat and M. Althoff, "Generating critical test scenarios for automated vehicles with evolutionary algorithms," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 2352–2358.

[23] H. Beglerovic, M. Stolz, and M. Horn, "Testing of autonomous vehicles using surrogate models and stochastic optimization," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–6.

[24] F. Gao, J. Duan, Y. He, and Z. Wang, "A test scenario automatic generation strategy for intelligent driving systems," *Mathematical Problems in Engineering*, vol. 2019, 2019.

[25] J. Wang, C. Zhang, Y. Liu, and Q. Zhang, "Traffic sensory data classification by quantifying scenario complexity," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1543–1548.

[26] M. Kamali, L. A. Dennis, O. McAree, M. Fisher, and S. M. Veres, "Formal verification of autonomous vehicle platooning," *Science of Computer Programming*, vol. 148, pp. 88–106, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167642317301168

[27] P. Koopman, B. Osyk, and J. Weast, "Autonomous Vehicles Meet the Physical World: RSS, Variability, Uncertainty, and Proving Safety," in *Computer Safety, Reliability, and Security*, A. Romanovsky, E. Troubitsyna, and F. Bitsch, Eds. Cham: Springer International Publishing, 2019, pp. 245–253.

[28] A. Mehmed, M. Antlanger, W. Steiner, and S. Punnekkat, "Forecast Horizon for Automated Safety Actions in Automated Driving Systems," in *Computer Safety, Reliability, and Security*, ser. Lecture Notes in Computer Science, A. Romanovsky, E. Troubitsyna, and F. Bitsch, Eds. Cham: Springer International Publishing, 2019, vol. 11698, pp. 113–127.

[29] N. Aréchiga, "Specifying Safety of Autonomous Vehicles in Signal Temporal Logic," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 58–63.

[30] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," *CoRR*, vol. abs/1708.06374, 2017. [Online]. Available: http://arxiv.org/abs/1708.06374

[31] T. Lanz, "Automatische Generierung relevanter Szenarien für den Test automatisierter Fahrzeuge unter Berücksichtigung der Sensorabdeckung," Master's thesis, Technische Universität München, München, 2019.

[32] W. Ertel, *Introduction to Artificial Intelligence*, 2nd ed., ser. Undergraduate Topics in Computer Science. Cham: Springer International Publishing, 2017. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-58487-4

[33] K.-L. Du and M. N. S. Swamy, "Ant colony optimization," in *Search and Optimization by Metaheuristics*, K.-L. Du and M. N. S. Swamy, Eds. Cham: Springer International Publishing, 2016, vol. 5, pp. 191–199.

[34] Forschungsgesellschaft für Straßen- und Verkehrswesen, "Richtlinien für die Anlage von Autobahnen," 2008.