IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 26, 2021, accepted July 22, 2021, date of publication August 9, 2021, date of current version August 24, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3103912

Test Scenarios Construction Based on Combinatorial Testing Strategy for Automated Vehicles

HONG SHU[1], HAORAN LV[1], KANG LIU[1], KANG YUAN[2], AND XIAOLIN TANG[1], (Member, IEEE)

[1]College of Mechanical and Vehicle Engineering, Chongqing University, Chongqing 400044, China
[2]College of Electronics and Information Engineering, Tongji University, Shanghai 200092, China

Corresponding author: Hong Shu (shuhong@cqu.edu.cn)

This work was supported by Chongqing Technology Innovation and Application Development Special Major Theme Special Project, Chongqing Science and Technology Bureau, under Grant cstc2019jscx-zdztzxX0039.

ABSTRACT Scenario-based testing is an important verification and certification measure to evaluate the safety of automated vehicles. In view of the existing test scenario composition methods, which may miss some critical scenario problems that have low occurrence probability, we fully combined the ego-vehicle with the possible relative positions and movement directions of surrounding traffic participants based on a complex scenario group. We applied scenario-screening rules to obtain the functional test scenarios with different traffic environments and driving task complexities, which ensured the coverage of the test scenarios and reduced the number of test scenarios. The problem arose that the amount of test cases was too large after the discretized combination of test scenario parameters, so we adopted a three-way combinatorial testing strategy to greatly reduce the number of test cases. Taking the complicated lane changing scenario of the ego-vehicle as an example, the simulation method was adopted, and the critical test cases were obtained by screening through safety indicators. Finally, the K-medoids clustering method was used to further reduce the number of critical test cases, and a pairwise combinatorial test strategy was used to combine dynamic scenario and static scenario elements to obtain critical test cases for closed-road testing.

INDEX TERMS Automated vehicles, combinatorial testing strategy, functional test scenarios, critical test cases, safety indicators.

I. INTRODUCTION

Automated vehicle technology has developed rapidly in recent years. The failure rate of automated vehicles at present is higher than that of traditional vehicles, and their safety needs to be improved. The test of automated driving is an important verification and certification measure to evaluate the safety of automated vehicles. To ensure the safety of automated vehicles is to ensure the correctness of their perception, decision-making, path planning algorithms, control algorithms, and the normal operation of the hardware system. Automated vehicle testing is divided into public-natural-road testing, closed-road testing, virtual testing, and hardware-in-the-loop testing. Since extremely dangerous scenarios are difficult to encounter in actual roads, seven billion kilometers of testing are required in the natural-road environment [1], making the cost of testing too high to be feasible. Scenario-based testing is considered to be an economical and feasible method for safety assessment of Highly Automated Driving (HAD) vehicles. By identifying critical scenarios with test significance, this method can be used for virtual simulation testing, hardware-in-the-loop testing, and closed-road scenario testing, which can shorten the test cycle, accelerate the safety test verification and certification of automated vehicles, and reduce the burden of test mileage on public-natural roads.

According to Schuldt *et al.* [2], a scenario is defined as something that "describes the temporal development between several scenes in a sequence of scenes." Scenarios are further divided into function, logic, and specific scenarios [3]. Functional scenarios are described by language, logical scenarios give the value range of scenario description parameters, and specific scenarios, also called test cases, are defined by specific scenario parameter values.

The associate editor coordinating the review of this manuscript and approving it for publication was Chih-Yu Hsu.

The sources of test scenarios for automated vehicles include accident data, real-world driving data, scenario catalogs, and expert knowledge. Test regulations for Advanced Driver Assistance Systems (ADAS) mainly come from the analyses of accident databases, such as the General Estimate System (GES) database released by the U.S. Highway Administration in 2015, which lists six categories and 13 types of collision scenarios [4]. The China Deep Accident Database extracts typical accident scenarios through the analysis of the Chinese accident database [5]. However, automated vehicles above the L3 level are currently in the testing stage and the accident data accumulation is insufficient, so the testing method used to approve ADAS cannot be transferred to HAD [6]. The German PEGASUS project [7] is mainly based on Field Operational Tests, Naturalistic Driving Studies, and traffic accidents to establish a scenario database as well as test and verify the automated driving system based on the scenario database. The European AdaptIVe test project proposes three types of test conditions covering short-distance, urban, and highway conditions, including 33 main scenarios and 36 alternative scenarios [8], but the number of test scenarios is small.

Based on a five-layer scenario model, Ponn *et al.* [9] manually determined the weak links of the tested automated system based on sensor analysis, driving behavior analysis, and consideration of the complexity of traffic conditions. This was done to adjust the test scenario, to adapt to the found weaknesses, and to extend the logical scenario to a complex scenario and test the planning algorithm of the tested vehicle, but no optimization model of the scenario parameters was given. Zhou *et al.* [10] proposed to start from a basic test scenario with only one influencing factor and gradually increase the number of influencing factors and collision avoidance behavior to form a complex test scenario, but the formation mechanism of the scenario was not clear enough. Based on ontology, the formal representation of knowledge and its relationships, Bagschik *et al.* [11] used an adaptive hierarchical model to represent scenarios, creating automated vehicle test scenarios in five layers, but the scenarios created completely depended on expert knowledge. Khastgir *et al.* [12] used constrained randomization technology to create ADAS randomized test scenarios and test cases in the driving simulator, which can better reflect the authenticity of the scenario, but they did not give a reasonable solution for the number of test cases.

Amersbach *et al.* [6] decomposed the HAD function into six layers, performed specific tests on each layer, and determined the failure criteria of the test cases according to the fault-tree analysis. They reduced the test workload by eliminating redundant test cases and aggregating test cases that were subsets of each other. Rocklage *et al.* [13] integrated the combined interactive testing methods with simple trajectory planners to generate static- and mixed-scenario test sets with variable coverage. Erdogan *et al.* [14] used the scenario database and the test case generator to select and configure scenarios and created parameterized test cases according to each scenario. The action parameters of each scenario could be modified to generate more test cases, but there are too many scenario combinations.

Due to the infinite diversity of actual road traffic conditions, most scenarios have no test significance, and therefore how to reduce the number of test scenarios and ensure high coverage of test scenarios are important issues that need to be resolved. In response to this problem, a scenario-based test method has been developed in recent years. By finding critical scenarios for testing to reduce the test workload, this method has proven to be an economical and feasible method for safety verification of HAD. Hallerbach *et al.* [15] proposed a generic simulation-based toolchain to identify critical scenarios. For specific scenarios, safety indicators and traffic quality indicators were compared with specific thresholds to determine whether the scenario was a critical scenario. In order to perform virtual tests on the safety of automated vehicle motion planning, Althoff *et al.* [16] optimized the drivable area of the vehicle to reduce the motion space of the vehicle and automatically generate critical scenarios. Klischat *et al.* [17] improved this method and proposed a method for automatically generating critical scenarios based on minimizing the solution space of the vehicle under test. This was accomplished by using evolutionary algorithms to solve the optimization problem, which can optimize the criticality of complex scenarios. The above methods generated one critical scenario for each calculation.

Another type of method is to find critical scenarios through natural driving data. Feng *et al.* [18] defined the criticality of a scenario as the combination of maneuvering challenges and exposure frequency. Maneuvering challenges refer to the probability of an automated vehicle encountering an event of interest in a scenario, which is estimated using an alternative model of the automated vehicle. Exposure frequency represents the probability of this happening on the road and is calculated from natural driving data. In order to reduce the computational complexity, the multi-start optimization method and seed filling method were used to search for critical scenarios. Xia *et al.* [19] screened out 80 samples of cut-in dangerous driving conditions from natural driving data. The hierarchical clustering method was used to analyze the proportion of real hazards and the degree of risk in 43 samples of the first type of cut-in dangerous conditions that the Autonomous Emergency Braking (AEB) system played a role in, and four critical test scenarios of the AEB system under the cut-in dangerous conditions were obtained. Zhao *et al.* [20] established a statistical model of the cut-in vehicle scenario based on the natural driving database, extracted critical scenarios through importance-sampling theory and the cross-entropy method, estimated the conflict, collision, and injury rate between the automated driving ego-vehicle and the cut-in vehicle, and realized the safety-accelerated evaluation of vehicles. On this basis, Huang *et al.* [21], [22] used the segmented mixed-distribution and kernel function methods to model the behavior of the cut-in vehicle and accelerated the evaluation of the ego-vehicle. The above methods are mostly

for simple scenarios, requiring a large amount of natural driving data to be acquired, which is time-consuming and costly. Additionally, the frequency of dangerous and complex scenarios in the natural driving data is low, and it is easy to miss some critical scenarios.

Most of the existing test scenarios are relatively simple or are small in number. The critical scenarios are determined by natural driving data, accident data, expert opinions, and scenario catalogs, which make it easy to miss some critical scenarios with a low probability of occurrence. In response to this problem, we adopted a full combinatorial testing strategy to form functional scenario groups with test values, based on research on the creation of functional scenario groups in pairwise combinations [23] and complex scenario groups. The method generated a wide coverage of scenarios, including simple and complex functional scenarios, with both clear logic of scenario composition and fast speed of scenario generation. Aiming to solve the problem of too large a number of scenario parameter combinations, we adopted a three-way combinatorial test strategy to parameterize scenarios so that the critical scenarios generated had high coverage and a small number of calculations. Taking a complex lane changing scenario as an example, the scenario parameters generated by the combination were screened through safety indicators, and a significantly reduced number of critical test cases were obtained through clustering. Compared with other methods, the safety index we designed was able to evaluate the safety of lane changing conditions and generate a significantly reduced number of critical test cases for closed field testing, with relatively high scenario coverage.

The rest of this paper contains the following contents: Section II describes the construction method of the functional test scenario based on the full combinatorial test strategy. Section III contains simulations, three-way combinatorial test strategies, safety indicators, and clustering methods to create critical test cases. Section IV describes the use of a two-way combination strategy to combine dynamic scenario and static scenario elements to generate critical test cases for closed-site testing. Lastly, Section V is the conclusion and prospects.

II. THE CONSTRUCTION FRAMEWORK OF FUNCTIONAL TEST SCENARIO GROUP BASED ON FULL COMBINATORIAL TEST STRATEGY

A. GENERATING FUNCTIONAL TEST SCENARIOS WITH A FULLY COMBINATORIAL TESTING STRATEGY

Each scenario is composed of static and dynamic elements. The static elements of the scenario include roads, transportation facilities, surrounding landscape, weather, and obstacles. Along with these static elements is the time of day, which includes day, night, sunny day, rainy day, snowy day, and foggy day. The dynamic elements of the scenario include the driving state of the automated vehicle, the driving state of the surrounding interfering vehicles, pedestrians, and sounds. When the static elements of the scenario are determined, the changes in the dynamic elements of the scenario constitute a different scenario. These scenarios assume that only the ego-vehicle and surrounding interfering vehicles are considered and that the sensor perception system is not malfunctioning. G and H represent the number of traffic participants. When the static elements of the scenario are determined, the scenario group is formed by the combination of the relative positions and movement directions of the ego-vehicle and the surrounding G traffic participants, which covers the scenarios with the largest number of traffic participants and most complex driving tasks, and altogether these combinations are called complex scenario groups. In this complex scenario group, if H ($1 \leq H < G$) traffic participants are arbitrarily removed, the scenario group formed by the combination of the relative position and movement direction of the remaining traffic participants and the ego-vehicle has less driving task complexity than the complex scenario group. Therefore, considering the comprehensiveness and swiftness of scenario construction, we built a functional scenario library based on complex scenario groups, and the steps were as follows:

1) The relative positions of the ego-vehicle and traffic participants were determined. The combinations of possible relative position and movement direction of the ego-vehicle and the surrounding G traffic participants were then analyzed. The complex scenario group was determined based on the premise of covering the most complex combination of driving tasks, and thus the relative position ranges of the ego-vehicle and G traffic participants were determined.

2) The relative movement directions of the ego-vehicle and traffic participants were determined. For a complex scenario group with a given set of static elements, the possible movement directions of the ego-vehicle were determined along with the movements of each interfering vehicle that affected the movement directions of the ego-vehicle, according to the level and function of the automated vehicle and the test target based on the perception, decision-making, planning, and control functions of the ego-vehicle. If the longitudinal and lateral movements of the interfering vehicle (including stationary) had little or no impact on the movement of the ego-vehicle, the interfering vehicle was deleted from the scenario. If the movement of the interfering vehicle in a certain direction did not cause interference to the designated movement of the ego-vehicle or the interference had little influence, the movement direction of the interfering vehicle was deleted.

3) The functional scenarios were formed. Each possible movement direction determined by the ego-vehicle function was fully combined with the possible movement directions of each interfering vehicle respectively (including the situation where any interfering vehicle did not exist) to form functional scenarios. Therefore, we used the Pairwise Independent Combinatorial Testing (PICT) tool to achieve this full combination by adding necessary motion constraints and selected coverage standards of parameter combinations to generate all combinatorial scenario groups automatically.

4) The scenario selecting rules were determined. We selected the scenarios with test values from the full

combinatorial scenario group, which formed a functional scenario library. The principles of scenario screening were as follows: 1) delete the scenarios that cannot be realized in real road traffic scenarios; 2) scenarios contained in other combined scenarios are deleted; 3) delete similar scenarios that have less impact on the movement of the ego-vehicle.

5) The complexity of the road traffic environment and the complexity of driving task was calculated. For each test scenario, we used the analytic-hierarchy process and graph-entropy method to calculate the complexity of the traffic road environment. For the driving task of the test scenario, we used the graph-entropy method to evaluate the complexity of the driving task using the four aspects of the number of operation steps, operation logic, traffic environment information, and task information.

B. EXAMPLES OF USING THE FRAMEWORK TO GENERATE FUNCTIONAL TEST SCENARIOS

The road section is the scenario that appears most frequently in the traffic environment. Taking a three-lane scenario with the ego-vehicle in the middle lane as an example, a complex scenario group of the ego-vehicle with nine interfering vehicles appearing in its surroundings simultaneously is shown in Fig. 1. EV represents the ego-vehicle (the automated vehicle under test) and C1-C9 represent the surrounding interfering vehicles and their possible location area. To ensure that the interfering vehicles had an impact on the movement of the ego-vehicle, C4 and C6 were longitudinally located in front of the ego-vehicle to ensure that C4 and C6 could perform cut-in actions. Similarly, the positions of C1 and C3 were selected. The positions of the other interfering vehicles also ensured that each vehicle could perform lane-changing operations.

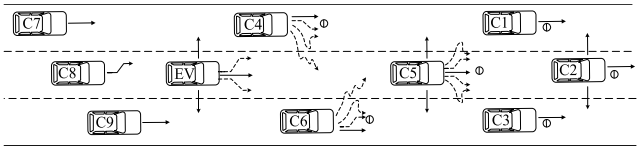

FIGURE 1. Directions of movement of the interfering vehicles that affect the movement of the EV.

According to the current automated driving functions above level two, the ego-vehicle's U-turn was temporarily not considered, and the possible movement directions of the ego-vehicle were straight, left/right lane change, and left/right yaw. Considering the influence of the interfering vehicles on the movement of the ego-vehicle under the possibility of actual traffic, the possible movement directions of the interfering vehicle C5 were straight, left/right lane change, left/right lane change failure, left/right yaw, and standstill. One of the possible movement directions of C4 was to continuously change lanes from the left lane to the right lane, which could test the ego-vehicle's lane keeping plus following function. All the possible directions of movement of C4 were straight, right lane change, right lane change fail, lane changing continuously to the right, and standstill. Similarly, the possible movement directions of C6 were determined. C1 and C3 were far away from the ego-vehicle, and their lane-changing and left/right yaw behaviors had little impact on the motion of the ego-vehicle. Therefore, the possible movement directions of C1 and C3 were straight and stationary. The possible movement directions of C2 that had an impact on the ego-vehicle were straight, left/right yaw, and standstill. When the ego-vehicle changed lanes, it was necessary to judge whether it could change lanes according to the relative distance and relative speed between the ego-vehicle and the front and rear vehicles in the adjacent lane. The possible movement of C8 that affected the ego-vehicle was overtaking. When C8 overtakes, the ego-vehicle must be courteous, and the lane change will be cancelled. The possible movement direction of C7 and C9 that affected the ego-vehicle was straight. We fully combined the possible movement directions of the ego-vehicle with the surrounding nine interfering vehicles and considered the situation where each interfering vehicle did not exist. The following describes the scenario construction based on the control function of the ego-vehicle.

1) THE EGO-VEHICLE GOING STRAIGHT

When the ego-vehicle was going straight, it was in the adaptive cruise control mode or active brake-assisting control mode. The movements of the interfering vehicles C7, C8, and C9 had no effect on the ego-vehicle and were not included in the scenario. Similarly, straight and stationary movements of vehicles C1, C3, C4, and C6 had no effect on the ego-vehicle and were not considered. The possible movement directions of the ego-vehicle and the interfering vehicles were fully combined. We used the PICT combination test case generation tool and added necessary constraints (such as scenarios that cannot be realized in reality) to automatically generate scenarios, and then, according to the scenario screening conditions, we obtained 34 functional scenarios with test values, some of which are shown in Fig. 2.

2) THE EGO-VEHICLE CHANGING LANE

When the ego-vehicle changed lanes to the left, it was in the active lane-changing control mode, that is, adaptive cruise control plus lane-change mode. We adopted the scenario construction method to obtain six functional scenarios with test values, as shown in Fig. 3. In the same way, six functional scenarios with test values were obtained when the ego-vehicle changes lanes to the right.

3) THE EGO-VEHICLE YAWING

When the ego-vehicle was yawing to the left or right, it was in the active lane-keeping following control mode. In addition to the lane-keeping mode, some ego-vehicles also have the lane-keeping lateral following mode. We adopted the scenario construction method to obtain 24 functional scenarios with test values, some of which are shown in Fig. 4.

In summary, we had constructed 70 functional test scenarios with different levels and functions for three-lane conditions. When the ego-vehicle is in the left or right lane at

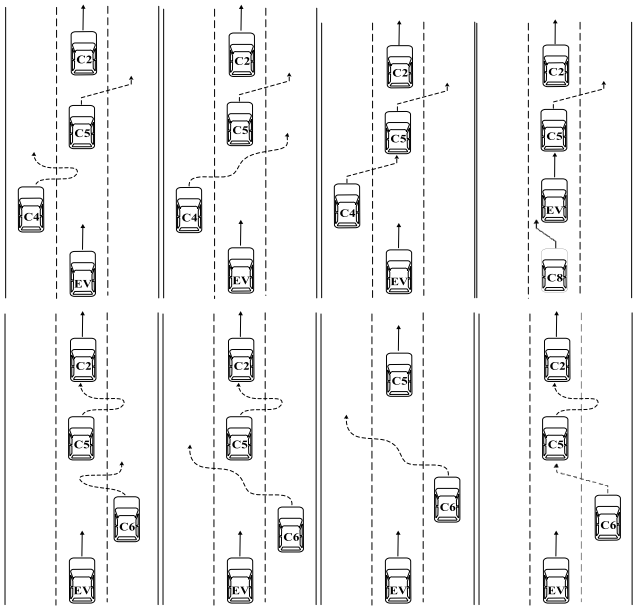

FIGURE 2. Part of the functional scenarios with the EV going straight.

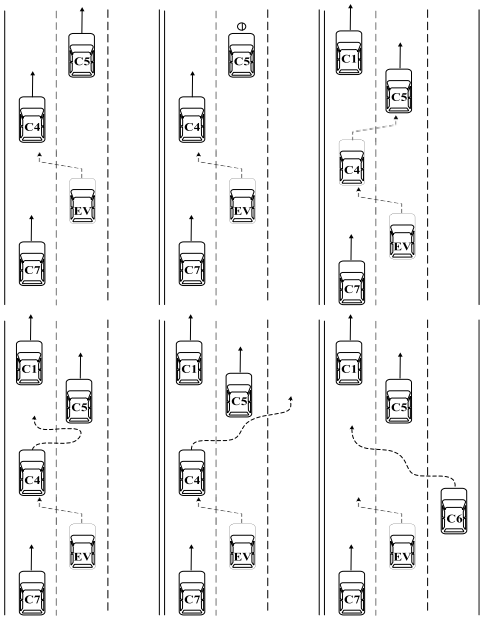

FIGURE 3. Functional scenarios with the EV changing lane to the left.

three-lane conditions, the corresponding functional test scenarios can be determined in the same way. Using this method, it is possible to establish functional test scenario group for other scenarios such as intersections without traffic lights, roundabouts, and etc.

III. CRITICAL TEST CASES GENERATION BASED ON COMBINATORIAL TESTING STRATEGY AND CLUSTERING

A critical test scenario is a test scenario that is dangerous, but no collision occurs. Carrying out safety test verification of automated vehicles based on critical scenarios can reduce

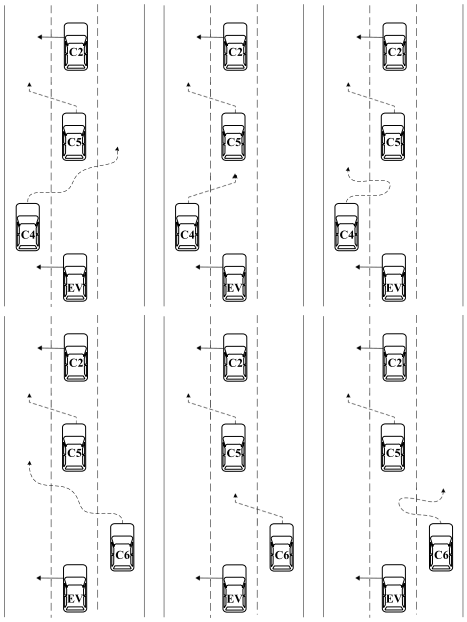

FIGURE 4. Part of the functional scenarios with the EV left yawing.

test costs and test mileage on public roads. First, we parameterized functional scenarios to form logical scenarios and reduced the number of specific test scenarios through combinatorial testing methods. Next, we selected critical test cases for virtual simulation testing through simulation and safety indicators and finally obtained fewer critical test cases for closed-field testing through clustering methods.

A. PARAMETERIZATION OF TEST SCENARIOS BY THE COMBINATORIAL TESTING STRATEGY

For a designated functional scenario, a logical scenario was first formed. Then the logical scenario was discretized, and a specific scenario was formed through parameter combination. If the full combination of scenario parameters was used, the number of scenarios would be large, and the calculation time would be too long to be realized. Since the combinatorial testing method has advantages in reducing test costs and maintaining high coverage, we adopted a combinatorial testing strategy to achieve scenario-parameter combination.

The combinatorial testing strategy is a method of selecting test cases that is widely used in computer software testing and quality control. It generates test cases by combining the values of input parameters of different test objects based on a certain combination strategy. The t-way coverage requires that every possible combination of valid values of the t parameter is included in a certain test case in the test suite [24]. PICT is a test case generation tool developed by Microsoft that implements any t-way ($1 \leq t \leq N$) test strategy. The core generation algorithm of the combined test case is a greedy heuristic algorithm [25]. Gao *et al.* [26] proposed a test scenario generation method that combines complexity and combinatorial testing methods and used the analytic hierarchy process to determine the contribution of the 16 influencing

factors of the Lane Departure Warning (LDW) system to the scenario complexity. Improved combined testing algorithms were used to reduce the number of scenarios to be tested and increase the complexity of scenarios. Xia *et al.* [27] used an improved PICT method to generate a more compact LDW system test suite, which ensured complete coverage of the specified t-way combination. Tuncali *et al.* [28] used a combination of two-way coverage arrays to generate discrete parameter combinations of test scenarios and presented automatic falsification methods to identify challenging scenarios in the perception system of automated vehicles. Research by Kuhn *et al.* [29] showed that, for browsers, more than 75% of bugs are triggered by the interaction of two or less parameters, and about 95% are triggered by the interaction of three or fewer parameters. In NASA database applications, 98% of failures were triggered by a three-way combination [30]. Three-way combined testing can significantly reduce the number of test cases while ensuring relatively high test quality and scenario coverage. Therefore, we adopted a three-way combinatorial test strategy to carry out the parameterized combination of scenarios.

The generation of the critical test scenario of a two-lane automated vehicle's left lane change was taken as an example for research. As shown in Fig. 1, it was assumed that there is an interfering vehicle C5 in front of the ego-vehicle, an interfering vehicle C4 in front of the left adjacent lane, and an interfering vehicle C7 behind it. C4, C5, and C7 were all in straight-driving conditions, and there were no interfering vehicles in other places. The ego-vehicle will change lane to the left, which created a complex lane-changing scenario.

We parameterized the motion states of the ego-vehicle and the three interfering vehicles to form logical scenarios. Considering the low-speed, medium-speed, and high-speed lane-changing scenarios of the ego-vehicle, the driving conditions of the ego-vehicle were divided into three categories: congestion conditions, suburban conditions, and highway conditions. The suburban driving conditions were taken as an example, and the speed range of the ego-vehicle and the interfering vehicle in the initial state was set to 40∼80 km/h. An interval of 5 km/h was used to discretize the speed of the ego-vehicle and the interfering vehicles, such that there were nine values for the initial speed of the ego-vehicle and interfering vehicles. When the ego-vehicle changed lanes, it was assumed that the interfering vehicles had either constant speed or constant deceleration, or were in a stationary state. The interfering vehicle C5 was in the same lane as the ego-vehicle and the interfering vehicle C4 was in the adjacent lane in the initial state, and the acceleration states of these two interfering vehicles were less likely to be in a dangerous state than the deceleration state. In order to form a challenging scenario for the ego-vehicle's motion planning algorithm, the deceleration range of C5 and C4 was set to -8 m/s^2 $\sim$0 m/s^2, with an interval of 0.5 m/s^2 for discretization, defining 17 deceleration values for those interfering vehicles. For the interfering vehicle C7, although its acceleration state posed a greater threat to the ego-vehicle, C7 generally would not choose to accelerate if there was a vehicle in front of it in the same lane in the actual driving process. If it was set to decelerate, it would have less impact on the safety of the ego-vehicle, so the interfering vehicle C7 was set to a constant speed state. The initial distances between the ego-vehicle and each interfering vehicle were greater than the minimum longitudinal safety distances, and the maximum deceleration time of the interfering vehicle C5 was set to 15 s. The maximum deceleration time of the interfering vehicle C4 was set to 3 s, and then it was at a constant speed or at a standstill to ensure that the ego-vehicle was between the interfering vehicles C5 and C7 after the lane change was completed. If the full combination were to be adopted, the number of test cases where the ego-vehicle changes lanes to the left would be 1896129, which is too large to be feasible.

We used the three-way combinatorial testing method to combine the scenario parameters, and in the end, we arrived at 3019 test cases. The generated parameter combination covered any combination of three scenario parameters, ensuring a high coverage rate of scenario parameters while also greatly reducing the number of calculations.

B. SAFETY EVALUATION INDICATORS FOR CRITICAL TEST CASES

The criticality of test cases is generally evaluated by safety indicators. At present, the most commonly used safety evaluation index is the Time-To-Collision (TTC). The traditional TTC is the ratio of the relative distance and the relative speed between the vehicles. If only the TTC is used as the evaluation index, it is easy to miss the critical test cases where the relative speed of the two vehicles is small but the relative distance is smaller. The point distances between the ego-vehicle and the interfering vehicles were included in the evaluation index as a supplement. Additionally, the impact of the longitudinal deceleration of the ego-vehicle on the driver and passengers was considered, and the maximum value of the longitudinal deceleration during the lane change of the ego-vehicle was also included in the evaluation indexes. The combination of multiple evaluation indicators can reduce the number of false-positive and false-negative evaluations and improve the accuracy of scenario evaluation. Since there were three interfering vehicles, there were three index values for the TTC indexes and the corner distances between the ego-vehicle and the interfering vehicles respectively, giving seven index values in total. So long as the threshold of any indicator was reached, it was considered to be a critical test case.

1) TIME-TO-COLLISION

The TTC is one of the most common safety indicators for determining critical test cases. The smaller the value of the TTC, the more critical the scenario. However, the TTC used in the past literature was calculated under the assumption that both the ego-vehicle and the front interfering vehicle were simplified as mass points, which is not suitable as a safety indicator for the ego-vehicle lane changing scenario [31].

In order to make the TTC suitable for the evaluation of lane-changing conditions, we referenced Qin *et al.* [32], regarded the ego-vehicle and the interfering vehicle as rectangles, and considered the heading angle of the ego-vehicle and the centroid positions of the two vehicles to calculate the longitudinal distance between them. Taking the longitudinal distance between the ego-vehicle and the preceding vehicle in the adjacent lane as an example, the relative position relationship between the two vehicles is shown in Fig. 5.

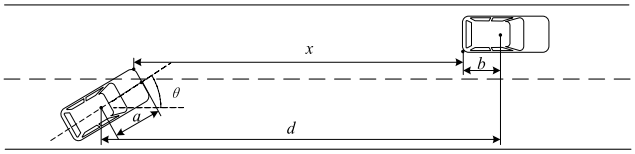

FIGURE 5. The relative position of the ego-vehicle and the front vehicle in different lane.

According to Fig. 5, the calculation formula of the longitudinal distance x between the ego-vehicle and the front vehicle when considering the heading angle is

$$x = d - a\cos\theta + \mathrm{w}\sin\theta/2 - b \tag{1}$$

where x is the relative longitudinal distance between the two vehicle, d is the longitudinal distance between the centroids of the two vehicles, a is the distance between the ego-vehicle's centroid and the front end of the ego-vehicle, b is the distance between the front vehicle's centroid and the rear end of the front vehicle, w is the width of the ego-vehicle, and θ is the heading angle of the ego-vehicle.

Considering that the interfering vehicle may accelerate or decelerate in the lane-changing scenario, the TTC that takes acceleration/deceleration into account was selected as one of the safety indicators for screening scenarios. We calculated the TTC by solving (2) [33], choosing the lowest positive root as the value of the TTC. If there is no positive root, the TTC is infinite. It is generally considered that 0 s<TTC <0.5 s is a pre-collision, 0.5 s≤TTC<1 s is an emergency state, 1 s≤TTC<2.5 s is a dangerous state, and TTC≥2.5 s is a normal driving state. We selected the scenario satisfying 0 s<TTC<2.5 s as the critical test case.

$$\ddot{x}t_T^2/2 + \dot{x}t_T - x = 0 \tag{2}$$

where t_T is the value of the TTC.

2) DISTANCE BETWEEN THE CORNERS OF TWO VEHICLES

During the lane change, the right-front corner point A of the ego-vehicle can easily collide with the left rear corner point A' of C5, the left front corner point B of the ego-vehicle can easily collide with the right rear corner point B' of C4, and the left rear corner point C of the ego-vehicle can easily collide with the right front corner point C' of C7, as seen in Fig. 6. The distance between the two corner points reflects the safety of the longitudinal and lateral movement of the ego-vehicle during the lane change, which can be used as a safety indicator for critical scenarios. If the distance between the two corners was less than 1.8 m, we considered the scenario to be dangerous and regarded it as a critical test case.

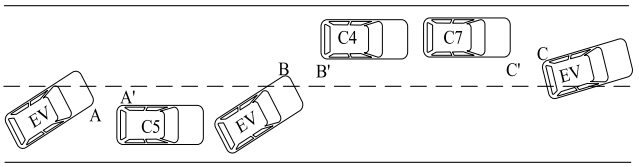

FIGURE 6. Schematic diagram of the collision corner position between the EV and the interfering vehicles.

Taking the centroid coordinates of the ego-vehicle and the interfering vehicle as reference points, we obtained the distances between the corner points of the two vehicles after deduction.

$$AA' = \sqrt{\begin{aligned}&(x_0 + a\,cos\,\theta + \frac{1}{2}w\,sin\,\theta - x_{C5} + b_{c5})^2 \\ &+(y_0 + a\,sin\,\theta - \frac{1}{2}w\,cos\,\theta - y_{C5} - \frac{1}{2}w_{c5})^2\end{aligned}} \tag{3}$$

$$BB' = \sqrt{\begin{aligned}&(x_0 + a\,cos\,\theta - \frac{1}{2}w\,sin\,\theta - x_{C4} + b_{C4})^2 \\ &+(y_0 + a\,sin\,\theta + \frac{1}{2}w\,cos\,\theta - y_{C4} + \frac{1}{2}w_{c4})^2\end{aligned}} \tag{4}$$

$$CC' = \sqrt{\begin{aligned}&(x_0 - b_e\,cos\,\theta - \frac{1}{2}w\,sin\,\theta - x_{C7} - a_{C7})^2 \\ &+(y_0 - b_e\,sin\,\theta + \frac{1}{2}w\,cos\,\theta - y_{C7} + \frac{1}{2}w_{c7})^2\end{aligned}} \tag{5}$$

where x_0 and y_0 are the horizontal and longitudinal coordinates of the centroid of the ego-vehicle in the inertial coordinate system; a_{c7} is the distances between the centroid and the front end of vehicle C7; b_e, b_{c4} and b_{c5} are the distances between the centroid and the rear end of the vehicle of the ego-vehicle, C4, and C5, respectively; x_{c5} and y_{c5} are the horizontal and longitudinal coordinates of the centroid of C5 in the inertial coordinate system; x_{c4} and y_{c4} are the horizontal and longitudinal coordinates of the centroid of C4 in the inertial coordinate system; and x_{c7} and y_{c7} are the horizontal and longitudinal coordinates of the centroid of C7 in the inertial coordinate system; w_{c4}, w_{c5} *and* w_{c7} are the width of vehicle C4,C5, and C7, respectively.

3) THE MAXIMUM OF LONGITUDINAL DECELERATION

The deceleration of the ego-vehicle will affect the safety and ride comfort of the driver and passengers. Generally, if the deceleration is greater than 2 m/s^2 it affects the comfort of passengers, and it affects safety if it is greater than 3 m/s^2. Therefore, we took the maximum deceleration as a safety indicator with a threshold of 3 m/s^2.

C. CRITICAL TEST CASES GENERATION BASED ON SIMULATION AND SAFETY INDICATORS

Aimed at the simulation of automated vehicles, we developed a new algorithm for automated vehicle decision-making,

motion planning, and control based on Model Predictive Control (MPC) [34]. The effectiveness of the automated vehicle dynamics model has been verified by real vehicle tests, and a variety of simple and complex lane change, cut-in, and lane keeping scenarios were simulated to verify the effectiveness of the algorithm. The algorithm decoupled the traditional automated vehicle multi-control variable MPC motion-planning and control framework and proposed an automated vehicle MPC motion-planning and tracking control framework based on longitudinal and horizontal separation focused on longitudinal safety priority, which improved the calculation speed of the algorithm. In addition, the vehicle kinematics and dynamics constraints were added to the motion-planning and control process to achieve a safe, stable, and comfortable motion-planning and control process.

According to the ego-vehicle lane-changing motion-planning and control algorithm, we established a co-simulation model of ego-vehicle lane-changing based on the software MATLAB/Simulink and CarSim. The dynamic parameters of the test scenarios obtained by the three-way combination of suburban driving conditions were simulated, the scenarios where the ego-vehicle failed to change lanes and collided with an interfering vehicle were deleted, and then the critical scenario evaluation index thresholds were screened to obtain 185 critical test cases. Among them, there were 64 groups meeting the TTC threshold requirements, 157 groups meeting the deceleration threshold requirements, and 77 groups meeting the corner point distance threshold requirements. These critical test cases can be used for virtual simulation and hardware-in-the-loop simulation tests.

D. CRITICAL TEST CASES BASED ON THE CLUSTERING METHOD

In section C, critical test cases were obtained through simulation and safety indicators of the lane-changing process of automated vehicles. However, for closed-road testing, too many test cases will lead to an increase in test costs, so to reduce the number of critical test cases we used the clustering method.

Clustering is the process of dividing a data set into many clusters according to a certain guiding ideology. The principle of the division is to make the result of clustering meet the conditions that the gap between data objects in the same cluster is as small as possible, while the gap between processing objects in different clusters is as large as possible. We used the clustering method to classify critical test cases with the same characteristics and test the representative data in the cluster to improve the test efficiency.

For clustering we used the K-medoids algorithm, which uses the object at the center of the cluster as a reference, handles outliers well, and has good robustness. In this way, the data points obtained by clustering are data points that exist before the clustering. This means the scenarios obtained by the clustering also met the critical test case indicator threshold, that is, the critical test cases. K-medoids is based on distance measurement, and the dimensions of the different variables contained in the test scenarios to be clustered are different. If the dimensions of different variables are too different, it may cause a small number of variables to dominate the clustering trend and affect the clustering effect. Therefore, the data needed to be standardized before calculation, so we used the method of range standardization.

The sample to be clustered contains multiple variables. For each variable, the standardized calculation formula is as follows [19]:

$$x_{ij}^{*} = \frac{x_{ij} - \min\limits_{1\le i\le N_c} x_{ij}}{\max\limits_{1\le i\le N_c} x_{ij} - \min\limits_{1\le i\le N_c} x_{ij}} \tag{6}$$

where x_{ij}^{*} is the normalized variable, $i = 1, 2 \ldots N_c$ and $j = 1, 2 \ldots M$, x_{ij} is the variable to be clustered, N_c is the number of critical lane-changing scenarios, and M is the number of variables to be clustered.

The standardized variable value ranges from 0 to 1 and is dimensionless. The elbow method was used to determine the number of clusters. As the number of clusters K increases, the data segment is more detailed, the degree of aggregation of each cluster increases, and the Sum of the Squared Errors (SSEs) gradually decreases. When the K value is less than the optimal number of clusters, the SSE value decreases more significantly with the increase of K value. After K reaches the optimal number of clusters, the SSE value decreases relatively slowly with the increase of K value. The relationship between the SSE and the cluster number K is similar to the "elbow" shape, and the "elbow" corresponds to the optimal K value. The calculation formula of SSE is as follows [35]:

$$\mathrm{SSE} = \sum_{i=1}^{k} \sum_{m_p \in C_q} \left| m_p - m_q \right|^2 \tag{7}$$

where SSE is the sum of the squared errors, C_q is the qth cluster, m_p is the sample point in C_q, m_q is the mean value of all samples in C_q, and k is the number of classification groups.

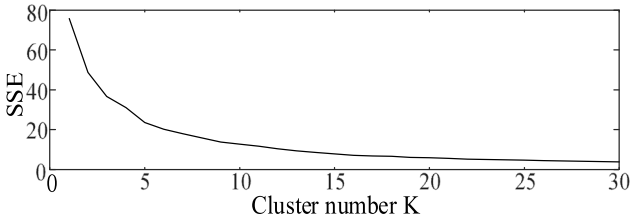

FIGURE 7. Relationship between cluster number K and SSE.

The upper limit of clustering was selected as 30, then each K value was clustered and the corresponding SSE was written down. The relationship between K and SSE is as shown in Fig. 7. It can be seen from Fig. 7 that the K value corresponding to "elbow" was seven, so the optimal cluster number was seven. The parameters of the seven sets of critical test scenarios obtained are shown in Table 1, with

TABLE 1. Dynamic scenario parameters based on K-medoids clustering.

	V_{0e} km/h	V_{0c4} km/h	a_{c4} m/s^2	V_{0c5} km/h	a_{c5} m/s^2	V_{0c7} km/h	a_{c7} m/s^2
1	50	60	-1.5	65	-3	55	0
2	65	65	-2	45	-2.5	45	0
3	65	65	-2	75	-5	45	0
4	70	75	-1	50	-1	65	0
5	75	75	-0.5	60	-3.5	55	0
6	80	50	-1	65	-3	45	0
7	80	75	-3.5	65	-2.5	40	0

the initial speed of the ego-vehicle and interfering vehicles C4, C5, and C7 represented by V_{0e}, V_{0c4}, V_{0c5}, and V_{0c7}, respectively, and the acceleration of interfering vehicles C4, C5, and C7 represented by a_{c4}, a_{c5}, and a_{c7}, respectively.

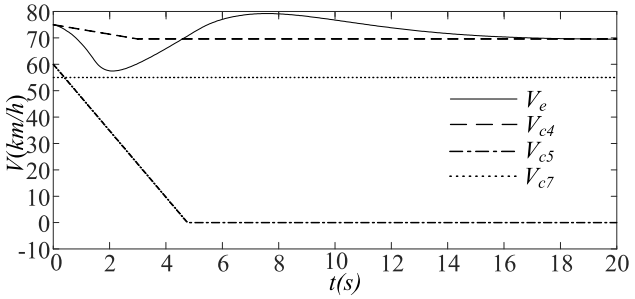

FIGURE 8. The speeds of the ego-vehicle and the interfering vehicles.

After analyzing the simulation results of the critical test cases after clustering, it was found that for the seven sets of critical test cases in suburban conditions, there were two sets that met the TTC threshold requirements, six sets that met the deceleration threshold requirements, and two sets that met the corner point distance threshold requirements. The critical test case with an initial speed of 75 km/h for the ego-vehicle obtained by clustering was taken as an example to analyze. In this critical test case, the initial speeds of the ego-vehicle, C4, C5, and C7 were 75 km/h, 75 km/h, 60 km/h, and 55 km/h, respectively, and the deceleration of C4 and C5 were -0.5 m/s^2 and -3.5 m/s^2, respectively, with C7 at a constant speed. Due to the restrictions of suburban conditions, the maximum speed of the interfering vehicles and the ego-vehicle should not exceed 80 km/h, and the minimum speed should not be less than 0 km/h. The speeds of the ego-vehicle and the interfering vehicles, the longitudinal acceleration of the ego-vehicle, the yaw rate of the ego-vehicle, and the lateral acceleration of the ego-vehicle obtained by the simulation are shown in Fig. 8-11, respectively. It can be seen from Fig. 8 that before and during the lane change of the ego-vehicle, when the front vehicle C5 decelerated, the ego-vehicle decelerated in order to maintain a safe distance. After the ego-vehicle successfully changed lanes, it first accelerated and then decelerated to keep the minimum safe distance from C4 as quickly as possible. Finally, the two vehicles' speeds were equal, and the vehicles were kept a certain safe distance. At the time of 0.95 seconds, the deceleration of the

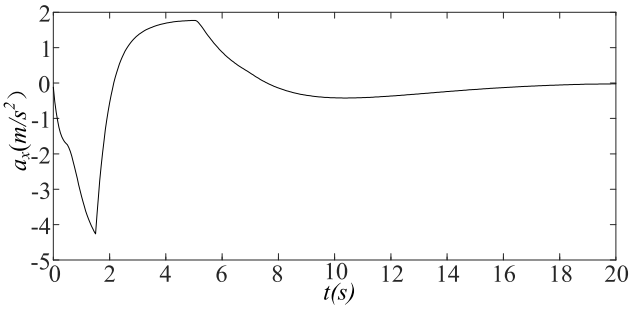

FIGURE 9. The longitudinal acceleration of the ego-vehicle.

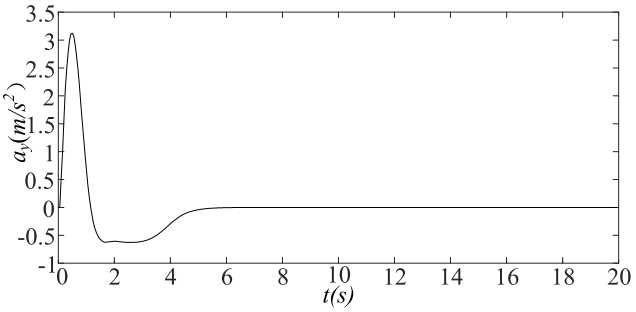

FIGURE 10. The lateral acceleration of the ego-vehicle.

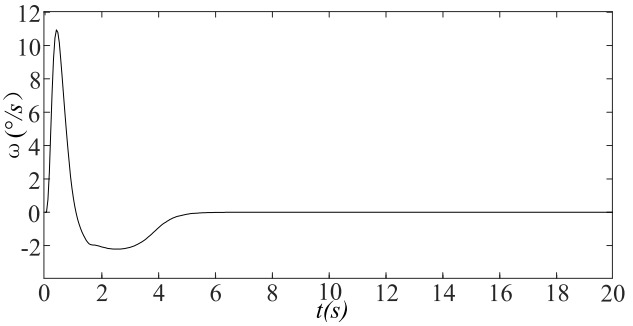

FIGURE 11. The yaw rate of the ego-vehicle.

ego-vehicle exceeded the threshold of 3 m/s^2 in order to avoid collision with the front vehicle C5. Then, at 1.5 seconds, the ego-vehicle was in the process of lane changing, and the TTC between the ego-vehicle and the front vehicle C4 in the adjacent lane was less than the threshold of 2.5 s, so this scenario was the critical test case.

IV. GENERATING TEST CASES BASED ON THE COMBINATION METHOD

According to the different levels of intelligent behavior of automated vehicles, it was necessary to design modular multi-level test scenarios with different degrees of difficulty. The critical test cases generated in Section III only contained the dynamic elements of the scenario, not the static elements of the scenario. We used a combinatorial testing method to design critical test cases containing static and dynamic elements to detect the perception and motion-planning capabilities of the ego-vehicle.

Based on the functional scenario when the ego-vehicle changes lanes on the left side of the two-lane road segment and the seven sets of critical dynamic scenario parameters obtained in TABLE 1, the specific critical scenarios were designed when the ego-vehicle changes lanes by considering the traffic elements including the time of day, road, traffic facilities, and dynamic elements. The time of day included weather and illumination, where weather was divided into sunny days, rainy days, snowy days, and foggy days, numbered 1–4, respectively, and illumination was divided into day, night, and flickering, numbered 1–3 respectively. The road included the length of the road and the number of lanes, where the road length was estimated according to each scenario, and the lane data were one-way and two-lane, numbered 1. Traffic facilities were lane lines, including white dashed lines and blurred lane lines, numbered 1–2. The traffic participant was a car, numbered 1. Seven critical lane-changing test cases were obtained by clustering dynamic element selection, numbered 1–7. We used the PICT toolbox to perform the method of pairwise combination and three-way combination to combine the above parameters and obtained 28 and 85 groups of critical test cases containing static environmental elements and dynamic elements, respectively. Considering the cost and conditional constraints of the actual field test, a pairwise combinatorial method was selected, and some of the generated critical test cases are shown in Table 2.

TABLE 2. Some of the generated critical test cases.

	Weather	Light	the Number of Lanes	Lane Lines	Traffic Participants	Critical Test Cases
T1	3	2	1	1	1	3
T2	4	1	1	2	1	7
T3	1	3	1	1	1	6
T4	1	2	1	2	1	1
T5	3	3	1	2	1	4
T6	2	1	1	1	1	4
T7	4	3	1	1	1	2
T8	4	3	1	2	1	3
T9	2	3	1	1	1	1
…	…	…	…	…	…	…

V. CONCLUSION

This paper introduced a method of generating functional test scenario groups based on complex scenario groups and using a full combinatorial testing strategy. This method can be used for the construction of functional test scenario groups for multi-lane road sections, intersections without traffic lights, roundabouts, and more. The paper also introduced a method of batch generation of critical test cases using combinatorial testing strategies. We adopted a three-way combinatorial testing strategy for the parameterized combination of scenarios, which greatly reduced the number of test cases. Additionally, multiple safety indicators were used to screen test scenarios and obtain critical test cases. On this basis, clustering was used to obtain representative critical test cases with a greatly reduced number, and dynamic scenarios and static scenario elements were combined to obtain critical test cases that could be used for closed-road testing. This method is suitable for the construction of critical test cases for automated vehicles. In the future, we will study the impact of the main parameters of the ego-vehicle (such as vehicle mass and air resistance coefficient) on critical scenarios, so that the critical test cases are suitable for various automated passenger cars for safety testing and verification. An optimization study of scenario parameters will also be carried out to obtain better critical test cases.

REFERENCES

[1] B. Klamann, M. Lippert, C. Amersbach, and H. Winner, "Defining pass-/fail-criteria for particular tests of automated driving functions," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Auckland, New Zealand, Oct. 2019, pp. 169–174.

[2] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer, "Defining and substantiating the terms scene, situation, and scenario for automated driving," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Gran Canaria, Spain, Sep. 2015, pp. 981–988.

[3] T. Menzel, G. Bagschik, and M. Maurer, "Scenarios for development, test and validation of automated vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Suzhou, China, Jun. 2018, pp. 1821–1827.

[4] *National Automotive Sampling System (NASS) General Estimates System (GES) Analytical User's Manual 1988-2015*, NHTSA, Washington, DC, USA, 2016.

[5] X. Li, X. Lian, and F. Liu, "Rear-end road crash characteristics analysis based on Chinese in-depth crash study data," in *Proc. CICTP*, Shanghai, China, Jul. 2016, pp. 1536–1545.

[6] C. Amersbach and H. Winner, "Functional decomposition: An approach to reduce the approval effort for highly automated driving," in *Proc. Tagung Fahrerassistenz*, Berlin, Germany, 2017, pp. 1–6.

[7] A. Zlocki and L. Eckstein, "Data handling in PEGASUS," Presented at the 2nd PEGASUS Symp., 2019. [Online]. Available: https://www.pegasusprojekt.de/files/tmpl/Symposium2019/1_3_PEGASUS_Data_Handling_Zlocki.pdf

[8] C. Rodarius, J. Dufils, F. Fahrenkrog, C. Roesener, A. Varhelyi, R. Fernandez, L. Wang, P. Seiniger, D. Willemsen, and L. van Rooijx, "Deliverable D7. 1 test and evaluation plan," in *Proc. AdaptIVe*, Berlin, Germany, 2015, pp. 12–71.

[9] T. Ponn, F. Diermeyer, and C. Gnandt, "An optimization-based method to identify relevant scenarios for type approval of automated vehicles," in *Proc. 26th Int. Tech. Conf. Enhanced Saf. Vehicles (ESV)*, Eindhoven, The Netherlands, 2019, pp. 1–19.

[10] J. W. Zhou and L. D. Re, "Reduced complexity safety testing for ADAS & ADF," in *Proc. 20th World Congr., Int. Fed. Autom. Control*, Toulouse, France, 2017, pp. 6159–6164.

[11] G. Bagschik, T. Menzel, and M. Maurer, "Ontology based scene creation for the development of automated vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Suzhou, China, Jun. 2018, pp. 1813–1820.

[12] S. Khastgir, G. Dhadyalla, S. Birrell, S. Redmond, R. Addinall, and P. Jennings, "Test scenario generation for driving simulators using constrained randomization technique," in *Proc. SAE World Congr. Exper.*, New York, NY, USA, 2017, Paper 2017-01-1672.

[13] E. Rocklage, H. Kraft, A. Karatas, and J. Seewig, "Automated scenario generation for regression testing of autonomous vehicles," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Yokohama, Japan, Oct. 2017, pp. 476–483.

[14] A. Erdogan, E. Kaplan, A. Leitner, and M. Nager, "Parametrized end-to-end scenario generation architecture for autonomous vehicles," in *Proc. 6th Int. Conf. Control Eng. Inf. Technol. (CEIT)*, Istanbul, Turkey, Oct. 2018, pp. 1–6.

[15] S. Hallerbach, Y. Xia, U. Eberle, and F. Koester, "Simulation-based identification of critical scenarios for cooperative and automated vehicles," *SAE Int. J. Connected Automated Vehicles*, vol. 1, no. 2, pp. 93–106, Apr. 2018.

[16] M. Althoff and S. Lutz, "Automatic generation of safety-critical test scenarios for collision avoidance of road vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Changshu, China, Jun. 2018, pp. 1–8.

[17] M. Klischat and M. Althoff, "Generating critical test scenarios for automated vehicles with evolutionary algorithms," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Paris, France, Jun. 2019, pp. 1–7.

[18] S. Feng, Y. Feng, C. Yu, Y. Zhang, and H. X. Liu, "Testing scenario library generation for connected and automated vehicles, Part I: Methodology," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1573–1582, Mar. 2021.
[19] L. Xia, X. C. Zhu, and Z. X. Ma, "AEB test scenarios under cut-in dangerous conditions," in *Proc. 14th Int. Forum Automot. Traffic Saf.*, Changsha, China, 2017, pp. 184–197.
[20] D. Zhao, H. Lam, H. Peng, S. Bao, D. J. LeBlanc, K. Nobukawa, and C. S. Pan, "Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 595–607, Mar. 2017.
[21] Z. Huang, D. Zhao, H. Lam, D. J. LeBlanc, and H. Peng, "Evaluation of automated vehicles in the frontal cut-in scenario—An enhanced approach using piecewise mixture models," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Singapore, May 2017, pp. 197–202.
[22] Z. Huang, Y. Guo, M. Arief, H. Lam, and D. Zhao, "A versatile approach to evaluating and testing automated vehicles based on kernel methods," in *Proc. Annu. Amer. Control Conf. (ACC)*, Wisconsin Center, ML, USA, Jun. 2018, pp. 4796–4802.
[23] H. Shu, K. Yuan, H. L. Xiu, Q. Xia, and S. He, "Construction of basic test scenarios of automated vehicles," *China J. Highw. Transp.*, vol. 32, no. 11, pp. 245–254,2019.
[24] M. Grindal, J. Offutt, and S. F. Andler, "Combination testing strategies: A survey," *Softw. Test. Verification Rel.*, vol. 15, no. 3, pp. 167–199, Sep. 2005.
[25] J. Czerwonka, "Pairwise testing in real world: Practical extensions to test case generator," in *Proc. 24th Pacific Northwest Softw. Qual. Conf.*, 2006, pp. 419–430.
[26] F. Gao, J. Duan, Y. He, and Z. Wang, "A test scenario automatic generation strategy for intelligent driving systems," *Math. Problems Eng.*, vol. 2019, Jan. 2019, Art. no. 3737486.
[27] Q. Xia, J. L. Duan, F. Gao, T. Chen, and C. Yang, "Automatic generation method of test scenario for ADAS based on complexity," in *Proc. Intell. Connected Vehicles Symp.*, Kunshan, China, 2017, Paper 2017-01-199.
[28] C. E. Tuncali, G. Fainekos, H. Ito, and J. Kapinski, "Simulation-based adversarial test generation for autonomous vehicles with machine learning components," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Suzhou, China, Jun. 2018, pp. 1555–1562.
[29] D. R. Kuhn and M. J. Reilly, "An investigation of the applicability of design of experiments to software testing," in *Proc. 27th Annu. NASA Goddard/IEEE Softw. Eng. Workshop*, New York, NY, USA, Dec. 2003, pp. 91–95.
[30] R. Kuhn, Y. Lei, and R. Kacker, "Practical combinatorial testing: Beyond pairwise," *Professional*, vol. 10, no. 3, pp. 19–23, 2008.
[31] S. M. S. Mahmud, L. Ferreira, M. S. Hoque, and A. Tavassoli, "Application of proximal surrogate indicators for safety evaluation: A review of recent developments and research needs," *IATSS Res.*, vol. 41, no. 4, pp. 153–163, 2017.
[32] J. X. Qin, W. W. Deng, and R. He, "Intelligent vehicle collision risk modeling and comprehensive evaluation method," in *Proc. 3rd Conf. Vehicle Control Intell. (CVCI)*, Hefei, China, Sep. 2019, pp. 1–5.
[33] D. Asljung, J. Nilsson, and J. Fredriksson, "Using extreme value theory for vehicle level safety validation and implications for autonomous vehicles," *IEEE Trans. Intell. Vehicles*, vol. 2, no. 4, pp. 288–297, Dec. 2017.
[34] K. Yuan, H. Shu, Y. J. Huang, Y. B. Zhang, A. Khajepour, and L. Zhang, "Mixed local motion planning and tracking control framework for autonomous vehicles based on model predictive control," *IET Intell. Transp. Syst.*, vol. 13, no. 6, pp. 950–959, 2019.
[35] G. Wu, J. Zhang, and D. Yuan, "Automatically obtaining K value based on K-means elbow method," *Comput. Eng. Softw.*, vol. 40, no. 5, pp. 167–170, 2019.

HONG SHU was born in Chongqing, China, in 1963. She received the Ph.D. degree in mechanical engineering from Chongqing University, Chongqing, in 2008. She is currently an Associate Professor with the College of Mechanical and Vehicle Engineering, Chongqing University. She has been a main researcher of over 20 research projects. She has published more than 50 academic articles. Her research interests include automated vehicle decision-making, control, evaluation, and environmental perception, electric vehicle dynamics and control, and vehicle eco-driving assistance technology.

HAORAN LV was born in Henan, China, in 1997. He received the B.S. degree in vehicle engineering from Wuhan University of Technology, Wuhan, China, in 2015. He is currently pursuing the M.S. degree in vehicle engineering with Chongqing University, Chongqing, China. He focuses on the construction of automated vehicle test scenario and naturalistic driving data analysis. His research interests include automated vehicles and data mining.

KANG LIU was born in Tangshan, China, in 1994. She received the B.S. degree in automotive service engineering from Hebei University of Engineering, Handan, China, in 2017, and the M.S. degree in vehicle engineering from Chongqing University, Chongqing, China, in 2020. Her research interests include test and evaluation of automated vehicles, construction of test case library, and simulation scenarios.

KANG YUAN was born in Hubei, China, in 1993. He received the B.S. and M.S. degrees in vehicle engineering from Chongqing University, Chongqing, China, in 2015 and 2019, respectively. He is currently pursuing the Ph.D. degree with the College of Electronics and Information Engineering, Tongji University, China. His research interests include machine learning, non-linear control and applications in decision-making, and motion control for connected and automated vehicles.

XIAOLIN TANG (Member, IEEE) received the B.S. degree in mechanics engineering and the M.S. degree in vehicle engineering from Chongqing University, Chongqing, China, in 2006 and 2009, respectively, and the Ph.D. degree in mechanical engineering from Shanghai Jiao Tong University, China, in 2015. From August 2017 to August 2018, he was a Visiting Professor with the Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, ON, Canada. He is currently an Associate Professor with the College of Mechanical and Vehicle Engineering, Chongqing University. He has led and has been involved in more than ten research projects, such as the National Natural Science Foundation of China. He has published more than 30 articles. His research interests include hybrid electric vehicles, vehicle dynamics, noise and vibration, and transmission control. He is also a Committee Member of Technical Committee on Vehicle Control and Intelligence of Chinese Association of Automation (CAA).