IEEE Open Journal of
Intelligent Transportation
Systems

Received 15 November 2019; revised 23 February 2020 and 24 May 2020; accepted 1 June 2020.
Date of publication 11 June 2020; date of current version 8 July 2020.

Digital Object Identifier 10.1109/OJITS.2020.3001801

Efficient Simulation Based Calibration of Automated Driving Functions Based on Sensitivity Based Optimization

NICOLAS FRAIKIN [1], KILIAN FUNK [1], MICHAEL FREY [2], AND FRANK GAUTERIN [2]

[1] Department of Automated Driving Functions, BMW AG, 80788 Munich, Germany

[2] Institute of Vehicle System Technology, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany

CORRESPONDING AUTHOR: N. FRAIKIN (e-mail: nicolas.fraikin@bmw.de)

This work was supported by the KIT-Publication Fund of the Karlsruhe Institute of Technology.

ABSTRACT Increasing demands on reliability and safety of automated driving functions require an augmented usage of simulation tools for the efficient calibration of these functions. However, finding an optimal solution can be costly, especially when the objective function is represented by scenario simulations. To face these challenges, a novel optimization scheme for simulation based calibration problems, that enables reduced computational effort is introduced. The approach is based on sensitivity analyses that provide scenario specific influential parameter spaces. Using these information, all parameter combinations are checked for reference candidates obtained in preceding iterations that are expected to have an equivalent solution as the new set. Thus, expensive simulation runs can be replaced by taking results from a reference set. The so called 'scenario simulation reduction' approach is applied to the parameterization of an SAE level 3 automated driving function with a genetic algorithm as optimizer. In order to take modeling inaccuracies into account, a robustness analysis with respect to simulation model parameters is conducted. Finally, a validation of the optimization scheme is performed using an extensive sampling approach. Studies confirm that negligible errors occur that are not expected to disturb optimization progress.

INDEX TERMS Automated driving, optimization, complexity reduction, simulation.

I. INTRODUCTION

FUTURE automated driving functions will offer extended functionalities that enable handling more situations than systems currently available in the market. Especially systems of SAE level 3 or 4 [1] need to be able to safely maneuver through urban areas, correctly interpret the traffic environment (traffic lights, signs, construction sites) and quickly react to unforeseen incidents. Increased capabilities of driver assistance systems cause a more time-consuming calibration process since the complexity and dimensionality of parameter spaces grow. While safety assessments and validation studies are already widely conducted virtually [2], [3], [4], [5], calibration tests are mainly performed on the target hardware. However, simulation environments can be used to efficiently obtain solutions with high maturity levels as starting points for a manual calibration in the vehicle. Besides computational advantages, simulation based parameter studies enable a cheaper, risk-free testing of driving scenarios and an improved reproducibility and analyzability [16]. Even though conducting optimizations in a simulation environment is by far more efficient than tuning parameters manually on the target hardware, computational efforts are still high since every system evaluation is represented by simulating a set of scenarios relevant for the respective driving function. In order to enable fast calibrations and short feedback loops in the development process, new approaches for an optimization with minimum computational effort are required. Next to reducing simulation times, the transferability of results to the target environment needs to be taken into account.

The duration of an optimization run is mainly influenced by the optimization algorithm (and its parameterization) as well as the size of the parameter space. The performance of an optimizer is problem-specific so that its suitability needs to be evaluated with respect to the problem [6]. Sensitivity analyses are widely used to reduce parameter spaces and therewith constrain the search space to influential regions. Since the optimal solution is expected to be located within these areas, the number of relevant parameters to the target output can be reduced and it may converge earlier. In most application examples, a sensitivity analysis (SA) is conducted first and the results afterwards used to remove certain parameters from the optimization process. The studies of Zhu *et al.* [7] expose the efficiency of such sensitivity based optimizations compared to using gradient-based optimizations and Design-of-Experiment-methods for optimizing multibody dynamics of a vehicle system. Further publications in the field of geometric design optimization confirm the effectiveness of this approach. Examples include the parameterization of models for a speed catamaran [8], wheel loader working device [9] or earthmoving mechanism [10]. In general, the concept of combining sensitivity analyses and optimizations can be applied to any calibration problem to improve system understanding and the efficiency of optimization runs [11]. It is especially useful when objective function calculations are computationally expensive and the overall number of system evaluations can be reduced [12]. Therefore, influence analyses are widely applied to reduce dimensionalities of vehicle calibration problems. Since the objective function is often characterized by one or more simulation runs, system evaluations are costly. Examples include the parameterization of transmission systems [13], engines [14] or hybrid electric vehicle operating strategies [15].

The aforementioned application examples as well as the calibration problem considered in this work are based on simulation models which represent the actual dynamics of the system. They are denoted as 'simulation optimization problems' [16]. Since the effect chain for calibrating powertrain systems consists of few components with well-defined interfaces, parameter optimizations can be performed using only the simulation models of the respective component (e.g., the engine, transmission) [13], [17], [18]. On the contrary, the effect chain for automated driving functions is more complex (see Figure 1). Many different sub-models (that represent the environment model, vehicle dynamics, . . . ,) are required to enable a closed-loop simulation of the whole system [19]. The more interacting and complex these sub-models are, the more likely are modeling inaccuracies to occur along the chain of effects that are not negligible. Therefore, the presented optimization method examines the robustness of optimal parameter combinations towards changing simulation model behavior. In general, high robustness for a parameter means that its magnitude and effect direction remain unchanged regardless the value of other parameters. However, in the context of this work robustness of parameter combinations with respect to vehicle model parameters are calculated. If optimal calibration parameter combinations are robust, it can be concluded that equivalent results occur regardless of the value of simulation model parameters. As a result, the respective solution is assessed to be stable towards deviating dynamics under real conditions. On the contrary to previous contributions from the field of 'simulation optimization problems' that address vehicle system calibrations, the model quality is taken into account.

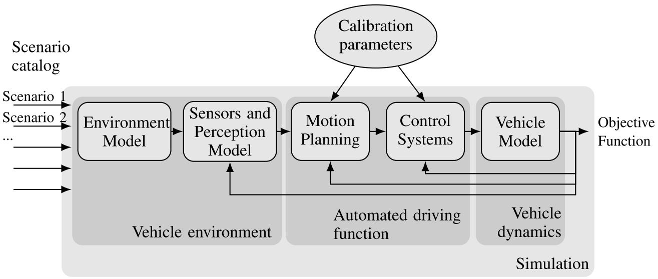


FIGURE 1. Optimization problem for the calibration of automated driving functions.

In order to minimize computational efforts, aforementioned literature references already show the potential of sensitivity analyses. In most cases influence information represent dependencies between an objective function and input parameters. However, as mentioned before, driving functions are optimized with respect to a scenario catalogue (see Figure 1). Therefore, the parameters' influence is additionally dependent on the driving maneuver. Even though sensitivity values could be averaged among all scenarios, valuable information would get lost that could decrease simulation times. As a result, to best use scenario specific influence information, the parameter space cannot be limited beforehand as suggested in other papers. Instead, the presented calibration scheme integrates scenario specific sensitivity information into the optimization algorithm to decide whether a simulation run of a parameter combination can be omitted.

If the initial search space cannot be notably reduced, a combination of SAs and optimizations can be computationally more expensive than conducting a simulation based parameterization alone. However, it should be noted that influence analyses are not only helpful to increase optimization efficiency but also to improve system understanding. Especially for complex models such as the effect chain shown in Figure 1, parameter effects and interdependencies can be exposed that may not be obvious to the calibration engineer. As a result, obtained information from the SA can not only be used as described in the presented scenario simulation reduction method but also for subsequent automated or manual calibrations in the vehicle.

The remainder of the paper is organized as follows. The first section provides the theoretical background to this work. In the following, a method is introduced to efficiently use sensitivity information for an optimization with minimum computational effort. The sensitivity based optimization method is thereafter used to find optimal solutions for a SAE level 3 driving function. An optimizer for solving the calibration problem is identified based on an experimental comparative study. Finally, a validation approach is described and outcomes of its execution analyzed.

II. THEORETICAL BACKGROUND

The following section provides the theoretical background for the calibration approach derived later. Therefore, a complexity reduction method, optimization algorithms and robustness measures are described.

A. COMPLEXITY REDUCTION OF THE PARAMETER SPACE

The introduced optimization approach in this work is based on a complexity reduction for an identification of influential areas in the parameter search space. The method introduced in a previously published contribution by the authors of this paper [20] deals with the same problem as addressed in this work and shown in Figure 1: Simulation based calibration of automated driving functions, i.e., finding parameter combinations for an optimal performance in a scenario catalog.

The method is based on a sensitivity analysis called 'elementary effects method' (EEM) introduced by Morris [21]. For the EEM an initial random sampling of size r (e.g., latin hypercube sampling) is conducted over the input space. In a second step, for each parameter combination n_p new sets are created following a 'one-factor-at-a-time' (OAT) radial variation scheme using Sobol's quasi-random numbers [22]. On the contrary to trajectory and cell based designs which suggest a uniform sampling over each input parameter space, the radial design provides the best performance with respect to result and efficiency as exposed by the studies of Campolongo *et al.* [23]. Convergence analyses carried out in [20] confirm that the SA provides reproducible results for different input samples using the radial approach. Thus, the number of samples for the original EEM is $r \cdot (n_p + 1)$. The output of Morris method are two metrics which indicate the main effect of each parameter (μ) and the degree of nonlinearity and interdependencies to other parameters (σ). The EEM is based on the calculation of elementary effects denoted as EE_i which are defined as relative changes of the objective function subject to changes of a single parameter i at a time. For the computation of μ_i all elementary effects are averaged. σ_i denotes the standard deviation:

$$\mu_i = \frac{1}{r} \cdot \sum EE_i \tag{1}$$

$$\sigma_i = \sqrt{\frac{1}{r-1} \cdot \sum (EE_i - \mu_i)^2} \tag{2}$$

To enable a relative comparison of parameters among each other these two metrics are summarized to one relative sensitivity using an euclidean distance in the $\mu - \sigma$-space: $s^{rel} = \sqrt{\mu^2 + \sigma^2}$. Using the EEM, the complexity reduction method aims to derive scenario specific influential parameter spaces. Therefore, conservative initial bounds for every parameter are estimated. Within these bounds, the EEM is applied to obtain sensitivity values for each parameter. Moreover, each parameter domain is split equally into n_{Int} intervals to limit initial bounds to influential bounds. Thereafter, the relative sensitivity s^{rel} for every interval per parameter is calculated with Morris' method while keeping remaining domains unchanged. Having calculated s^{rel} for all intervals and parameters, those intervals with a relative sensitivity below a certain threshold s^{rel}_{min} can be neglected for further evaluations. Since the method is applied to every interval for each parameter a total of

$$n_{CR} = n_p n_{Int} r \cdot (n_p + 1) \tag{3}$$

system evaluations needs to be performed.

Since experimental studies have shown that for the calibration of vehicular driving systems areas at upper and lower bounds of conservatively estimated parameter ranges tend to have smaller impact [20], the initial domain can be narrowed down based on this method. After the redefinition of sensitive ranges, samples laying inside the influential hyperbox can be further processed for a global sensitivity analysis of all parameters. If a parameter's global sensitivity remains below the above mentioned threshold, it can be completely neglected for further optimizations.

B. OPTIMIZATION ALGORITHMS

In general, optimization algorithms can be divided into gradient-based and gradient-free direct search methods. The first category needs derivatives of the objective function and is based on deterministic, mathematical operations. Algorithms are mostly not parallelizable and computationally expensive [24]. Since simulation optimization problems are nonlinear and allow a derivative representation only with an unreasonably high effort, gradient-based approaches are inapplicable for the discussed problem in this work.

The suitability of an algorithm from the class of global gradient-free optimizers is mainly dependent on the respective calibration problem [6]. Preceding research in the field of vehicle system calibrations has mostly been carried out with the genetic algorithm (GA) [13], [15] and particle swarm optimization method (PSO) [25], [26], [27]. Therefore, these two population-based approaches are described and experimentally compared in the following.

1) GENETIC ALGORITHM

The GA is based on Darwin's theory of natural selection and reproduction governed by rules that assure the survival of the fittest. In each iteration a new population of individuals (parameter combinations) is produced by means of selection and reproduction. These two phases are based on probabilistic rules that utilize a fitness function (objective function). The genetic operations *selection* and *reproduction* performed in each iteration of the GA as well as the once performed *initialization* are exemplary shown in Figure 2 and can be described as follows:

- *Initialization:* In the first iteration a starting population is generated randomly (e.g., by latin hypercube

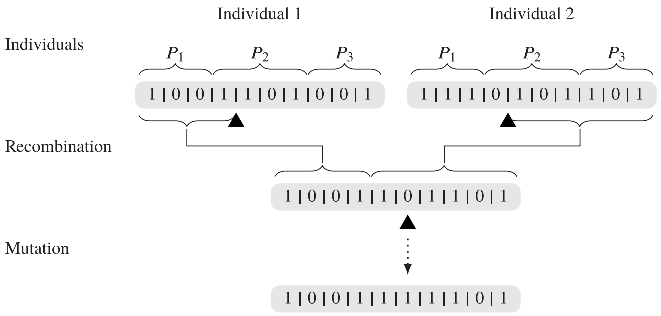


FIGURE 2. **Application of recombination and mutation operators on a binary string for the genetic algorithm.**

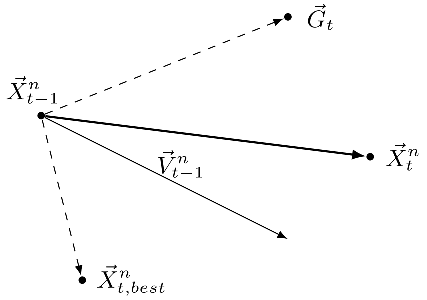


FIGURE 3. **Particle motion as implemented in the PSO-algorithm.**

sampling or halton sequence). Alternatively individuals may be placed based on expert knowledge.

- *Selection:* As a first step for creating the next generation, the fittest individuals from the last iteration are chosen for later reproduction.
- *Reproduction:* Taking the selected individuals, a new generation is created through *recombination* and *mutation*:
  - *Recombination:* During recombination the genetic information of two selected individuals are beneficially combined to create new offspring. Thus, recombination operations push the population to best available solutions and support convergence.
  - *Mutation:* Mutation is used to avoid getting stuck in local optima and support breadth-first-search. It introduces genetic diversity by randomly altering one or more genes of each individual.

For each genetic phase probabilities are defined that influence optimization progress towards convergence speed. The selection probability p_s determines how many individuals are chosen for the selection operator. Accordingly, the recombination probability p_r decides how many individuals are recombined and the mutation probability p_m influences the number of mutations per iteration.

Each individual in nature stores genetic information in a chromosome. For the GA chromosomes can be represented by bit arrays so that continuous variables are approximated with a binary decomposition [28]. For a binary representation of a parameter with n_{Values} values, the number of bits k needs to be chosen so that the condition $2^k \geq n_{Values}$ is met. Decimal values associated with each bit array are obtained by splitting the parameter domain equally in 2^k parts and allocate every bit array to one value along the domain. The parameter combination (individual) is finally created by attaching all parameter specific bit arrays. Genetic operations are performed on the summarized array (see Figure 2).

2) PARTICLE SWARM OPTIMIZATION

The PSO is inspired by the swarm behaviour of organisms and its common search for food or resting areas [29]. On the contrary to the GA, individuals collaborate on finding an optimal solution. Therefore, particles (or parameter combinations) are moved through the search space whereby positions and velocities are updated in every iteration [30]. The intelligence for a particle's movement is based on a combination of three operations: *evaluation*, *comparison* and *imitation* [31]. The *evaluation* is conducted through the objective function. A *comparison* is made based on the particle's own best known position as well as the entire swarm's best position. In the *imitation* phase all particles aim to derive promising actions from dominating individuals.

The three phases can be mathematically described as follows: In the beginning, an initial population with size N is randomly sampled in the search space. Each individual n is assigned a position vector $\vec{X}^n_t \in \mathbb{R}^{n_p}$ and a velocity vector $\vec{V}^n_t \in \mathbb{R}^{n_p}$. The total number of parameters is denoted by n_p. In every iteration t the own historically best parameter combination $\vec{X}^n_{t,best} \in \mathbb{R}^n_p$ of each particle and the global best parameter combination $\vec{G}_t \in \mathbb{R}^n_p$ are saved. Based on these reference points every individuals' next movement can be determined using the following equations of motion:

$$\vec{V}^n_t = \cdot \vec{V}^n_{t-1} + c_1 \cdot \vec{r}^n_t \cdot (\vec{X}^n_{t,best} - \vec{X}^n_{t-1}) + c_2 \cdot \vec{s}^n_t \cdot (\vec{G}_t - \vec{X}^n_{t-1}) \tag{4}$$

$$\vec{X}^n_t = \vec{X}^n_{t-1} + \vec{V}^n_t \tag{5}$$

The constants $c_1, c_2 \in [0, 1]$ regulate the orientation on the individual and globally best solution. Therefore, they define the algorithm's parameterization towards breadth-first and depth-first search. The factors $\vec{r}^n_t \in \mathbb{R}^{n_p}$ and $\vec{s}^n_t \in \mathbb{R}^{n_p}$ represent imitation heuristics and are calculated in every step of the algorithm. Figure 3 shows the particle motion implemented in the PSO.

C. ROBUSTNESS ANALYSIS

In literature different definitions for robustness exist. However, in the context of this work a robustness analysis is applied to examine influences of (slightly) different parameterizations compared to an optimal parameter combination on the objective function. In other words, a robustness analysis allows to expose the steepness of local optima in the parameter search space. Steepness in this case means that only one very good solution exists and samples in its immediate environment (i.e., with slightly deviating parameterizations)

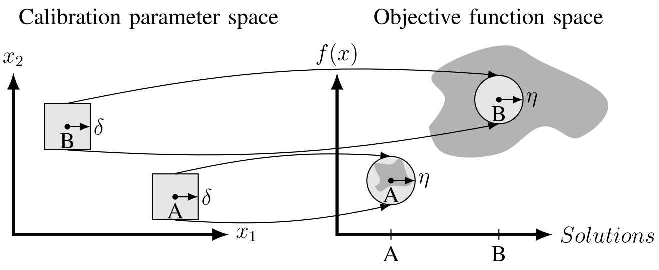


FIGURE 4. Definition of neighborhoods in the calibration parameter space (left) and objective function space (right). Solution B does not fulfill robustness conditions defined by η whereas solution A does.

provide a worse objective function value. Hence, if a solution was found at a steep peak, it is not stable towards slightly changed parameter values and thereby not robust. Many algorithms for finding robust solutions in multi-objective [32], [33], [34] and single-objective [35], [36] optimization problems were presented in the past. Since the analyzed problem in this work has only one objective function, approaches for multi-objective cases are not further discussed.

The calculation of a robustness measure for a parameter combination $x^S \in \mathbb{R}^{n_p}$ with objective function value $f(x^S)$ involves analyzing its immediate neighborhood. Figure 4 illustrates the concept for a parameter space with two parameters (x_1, x_2). In this figure, parameter combinations sampled inside a δ-hyperbox in the calibration parameter space around solution A and B (left plot) cause objective function values laying inside the respective dark gray polygons (right) in the objective function space. The variable η denotes a predefined robustness threshold. Since the polygon associated with solution A remains inside a hyperbox with radius η it is assessed robust. On the contrary, small deviations from parameter combination B cause larger objective function deviations that do not fulfill robustness conditions.

In this work, the approach is adapted to examine stability of optimal solutions with respect to changing environment conditions represented by vehicle model parameters. Therefore, parameter combinations in the calibration parameter space are held constant whereas vehicle model parameters are varied inside a δ-hyperbox. If these variations cause deviations from the optimal objective function value larger than η, the corresponding calibration parameter combination is assessed not robust towards changing vehicle dynamics. The robustness measures therewith provide an estimate whether calibration parameter combinations found in a simulation setup provide a similar performance under real conditions.

For a robustness evaluation of one solution, new parameter combinations (H) are randomly sampled inside a δ-hyperbox. The variable y may denote a parameter combination of vehicle model parameters with y^S being the combination associated with x^S. The respective objective function values are defined by $f(y^i)$ with $i \in 1, \ldots, H$. For the decision whether robustness is achieved the mean deviation between new function values $f(y^i)$ and $f(y^S)$ is compared to η: $\frac{1}{H}\sum_{i=1}^{H} \frac{|f(y^i)-f(y^S)|}{|f(y^S)|} < \eta$. Since outliers may be neglected through averaging, alternative measures such as the maximum absolute difference $\max_{i=1,\ldots,H} \frac{|f(y^i)-f(y^S)|}{|f(y^S)|} < \eta$ or a factor-dependent evaluation $\frac{|f(y^i)-f(y^S)|}{|f(y^S)|} < \eta_i, i = 1, \ldots, H$ can be used [37]. A maximum hyperbox size δ_{max} that still fulfills one of the above mentioned conditions is calculated through means of optimization and serves as robustness metrics. The higher the maximum hyperbox size, the more stable is a solution towards parametric changes. Alternatively to optimizations, the allowable hyperbox size δ_{max} can be calculated by taking multiples of an initial δ_{min} until robustness conditions are not met anymore. The maximum number of box enlargements that still enables parametric stability determines the 'degree of robustness'.

III. SCENARIO SIMULATION REDUCTION METHOD FOR AN EFFICIENT OPTIMIZATION OF CALIBRATION PARAMETERS

The goal of the optimization scheme introduced in this section is to beneficially use the complexity reduction (CR) approach as described before to reduce needed simulation runs for solving the optimization problem shown in Figure 1. The CR method provides information about influential parameter bounds and parameters, both with respect to a specific scenario. If overlapping non-influential regions for all parameters and scenarios exist, a decreased parameter search space can be easily derived and used for optimization by taking the smallest common area. However, it is expected that the relevant parameter search space varies depending on the characteristics of a scenario. Moreover, by reducing the search space to smallest common areas, potentials of scenario specific sensitivity information would be neglected.

In order to use full potential of the CR method outcome, this approach aims to integrate scenario specific sensitivity information in the optimization algorithm to minimize simulation efforts in each iteration. As pointed out before, population-based optimizers seem most promising for solving the herein considered calibration problem. The individuals are created by the optimizer in each iteration and need to be evaluated (in parallel) before the algorithm can proceed. In the herein considered optimization problem evaluating an objective function for one individual (or parameter combination) means simulating all n_s scenarios and summarizing their individual results to one combined objective function value using a weighted sum. This approach is chosen to obtain a single-objective optimization problem (more details are provided in Section IV-A).

By taking scenario specific influential parameter spaces into account, a simulation run might be obsolete if parametric changes compared to a previously evaluated parameter combination are expected to have a negligible influence. In this case the objective function value for the current parameter combination can be copied from a previously evaluated reference parameterization and an expensive simulation run can be avoided. Parametric changes are expected to be negligible if a new parameter combination only differentiates from a reference combination inside non-sensitive

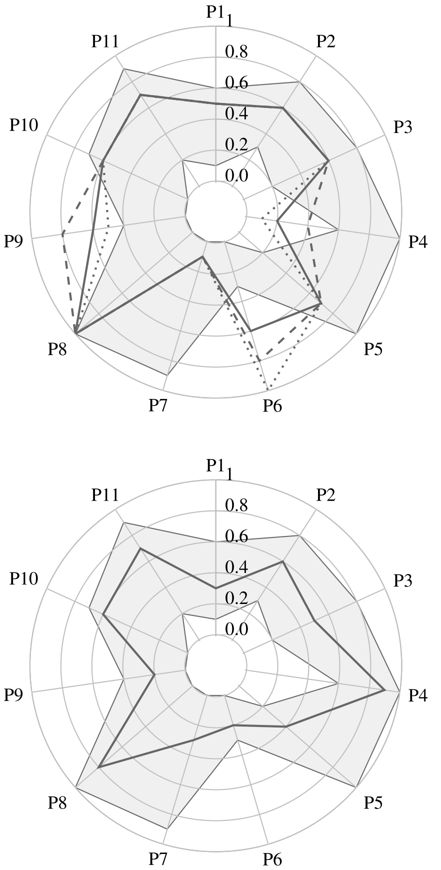


FIGURE 5. Top: Example for a replacement of a scenario simulation (solid line) through previously evaluated parameter combinations (dashed and dotted lines) since they only deviate from the reference combination within non-influential (white) areas. Bottom: Example for a parameter combination that does not qualify for scenario simulation reductions since all parameters lay within influential (gray) area.

areas. That means, if a parameter of the current combination is located within influential bounds it should have the exact same value in the reference combination. However, parameters laying inside non-sensitive regions might take any other value within this region since parametric changes in this area were assessed non-influential by the CR method. During optimization, all simulated parameter combinations and objective function values are stored in a database and serve as references.

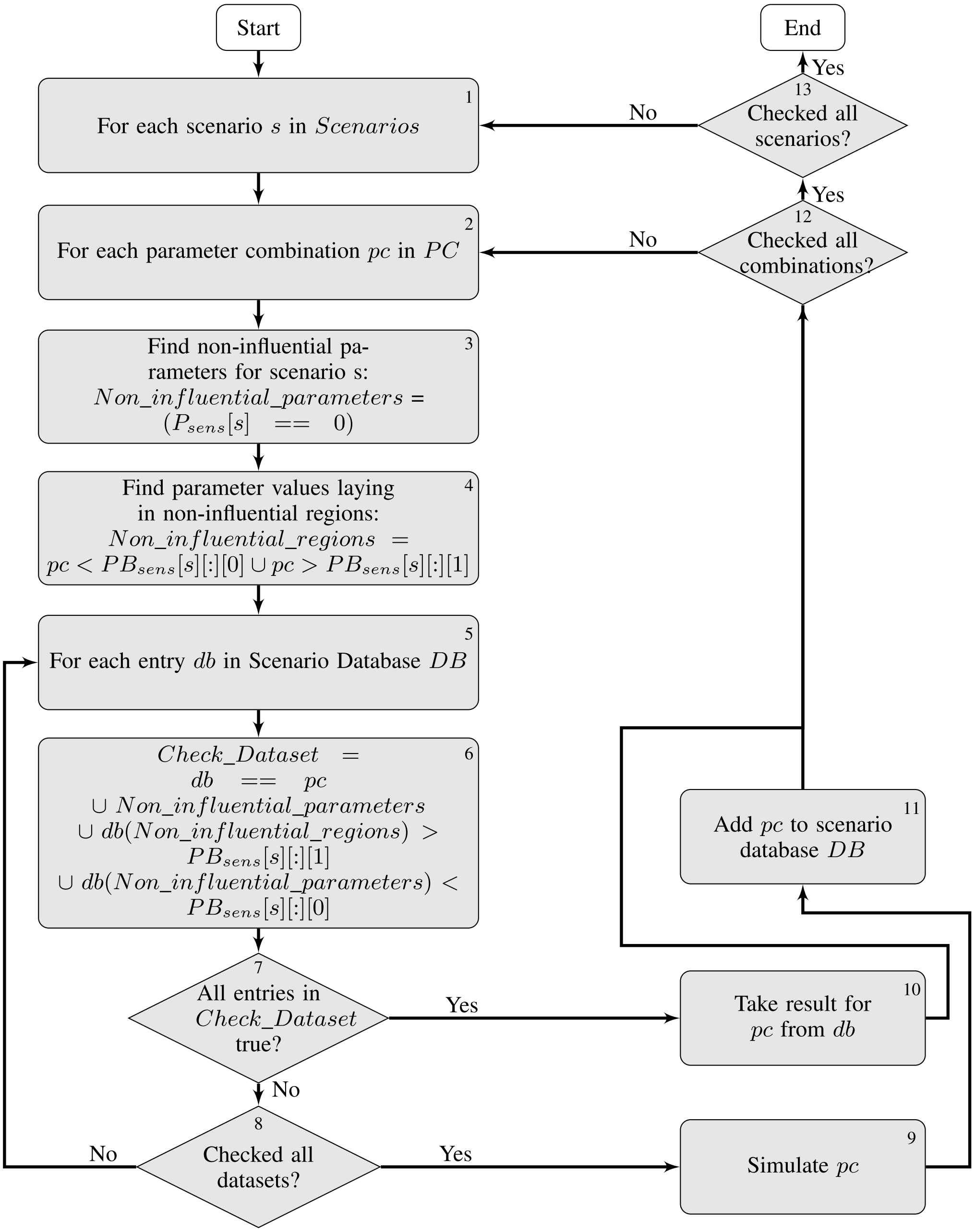
Fig. 5 shows an example where a scenario simulation can be avoided by taking objective function values of a reference combination (upper plot) and a contrary example where a simulation is necessary (lower plot). In this example, 11 calibration parameters (P1-P11) are considered.

Gray areas represent influential regions for the respective scenarios whereas white areas are non-influential. The solid line in the upper plot may represent a parameter combination generated by the optimizer. Combinations described by dashed and dotted lines have the same value for those parameters where the new parameter combination (solid line) lays within influential areas. For parameters with values inside non-sensitive areas these combinations do not align with the new combination (solid line) but vary within white (non-sensitive) areas. Thus, they may serve as valid references for the newly created parameter combination by the optimizer and make its simulation run unnecessary. The parameter combination in the lower plot does not allow scenario simulation reductions since all parameters lay inside sensitive areas. It can not be ruled out that small deviations from this parameter combination have a decisive impact on optimization progress.

Fig. 6 shows a flowchart for the described approach. It can be understood as a process description for evaluating all samples from a population with minimum computational expenses. Sensitive parameter bounds PB_{sens} and parameters P_{sens} are herein obtained by applying the complexity reduction method beforehand. For each scenario (first dimension) and parameter (second dimension) they store sensitive upper and lower bounds or parameter influence information (third dimension). P_{sens} is 1 if the respective parameter is expected to have an impact and 0 otherwise. The scenario catalogue is denoted by *Scenarios*. After the optimizer has created a new population of individuals *PC* it is checked for each scenario *s* (1) and parameter combination *pc* (2) whether parameters were assessed non-influential (3). Thus, the variable *Non_influential_parameters* is an n_p-dimensional boolean vector which is *true* for parameters without influence. In the following it will be checked which parameter values of the current combination are laying inside non-influential regions (4). The corresponding boolean vector is denoted as *Non_influential_regions*. Based on that information about the current combination *pc* the database *DB* is searched (5) for valid references fulfilling conditions in block (6). Valid reference combinations are those, whose parameters have the same value as *pc* ($db == pc$) or were assessed non-influential (*Non_influential_parameters*) or lay inside the same non-influential regions as the respective parameters from *pc* ($db(Non\_influential\_regions) > PB_{sens}[s][:][1] \cup db(Non\_influential\_regions) < PB_{sens}[s][:][0]$). If a database combination was found that fulfills all conditions (7), it is chosen as valid reference (10) making simulations obsolete. If the whole database was screened (8) and no qualified candidate was found, the respective combination needs to be simulated (9) but can itself serve as a potential reference (11) for future evaluations. The scenario database *DB* is available after simulating the first parameter combinations and extends its size after every iteration. Therefore, it is obvious that chances for a simulation replacement increase the further the optimization algorithm has proceeded.

IV. APPLICATION OF THE METHOD FOR THE CALIBRATION OF AN AUTOMATED DRIVING FUNCTION

Having derived the method theoretically, its potential with respect to computation time and optimization result is evaluated with a realistic use case: Simulation based calibration of an automated driving function. An optimizer for solving the described problem is selected using a comparative study of the GA and PSO. The chosen optimizer is integrated into the optimization scheme and applied to the problem. Finally,


Start
End
Yes
13
Checked all scenarios?
No
1
For each scenario s in $Scenarios$
Yes
12
Checked all combinations?
No
2
For each parameter combination pc in PC
3
Find non-influential parameters for scenario s: $Non\_influential\_parameters = (P_{sens}[s] \quad == \quad 0)$
4
Find parameter values laying in non-influential regions: $Non\_influential\_regions \ = pc < PB_{sens}[s][:][0] \cup pc > PB_{sens}[s][:][1]$
5
For each entry db in Scenario Database DB
6
$Check\_Dataset \ = db \ == \ pc \cup Non\_influential\_parameters \cup db(Non\_influential\_regions) > PB_{sens}[s][:][1] \cup db(Non\_influential\_parameters) < PB_{sens}[s][:][0]$
11
Add pc to scenario database DB
7
All entries in $Check\_Dataset$ true?
Yes
10
Take result for pc from db
No
8
Checked all datasets?
No
Yes
9
Simulate pc

Population size	n	Current population	$PC \in \mathbb{R}^{n \times n_p}$
Number of parameters	n_p	Sens. parameters	$P_{sens} \in \mathbb{N}^{n_s \times n_p}$
Number of scenarios	n_s	Sens. parameter bounds	$PB_{sens} \in \mathbb{R}^{n_s \times n_p \times 2}$
Scenario list	$Scenarios$	Scenario Database	$DB \in \mathbb{R}^{n_s \times N \times n_p}$
Number of entries in DB	N		

FIGURE 6. Flowchart for the evaluation of each population including a check for reference datasets that can replace computationally expensive simulation runs.

the best solutions found by the method are examined with respect to their robustness. The section concludes with a discussion of the results.

A. PROBLEM DESCRIPTION

The underlying simulation optimization problem was already sketched in the beginning of this paper (see Figure 1). In this application example the trajectory planning module of a level 3 driving function (according to SAE definition [1]) is parameterized for highway scenarios. The scenario catalog consists of six scenarios that are defined as follows:

- S1: Change lane on straight road (overtake slower vehicle on the right)
- S2: Change lane on a curved road (overtake slower vehicle on the right)
- S3: Keep lane on a straight road

- S4: Keep lane on a curved road
- S5: Adapt speed to a traffic sign (Acceleration maneuver)
- S6: Adapt speed to a curve (Deceleration maneuver)

In order to measure the performance for each scenario, objective functions need to be defined. In this work a machine learning based approach is used for an objective assessment of driving behavior presented by Moser *et al.* [38]. This approach uses multivariate time series classification to evaluate driving comfort for lane keeping, lane change and acceleration/deceleration maneuvers. It provides comfort metrics on a discrete scale from 0-7 where 7 represents maximum driving comfort.

As mentioned above, objective function values from every scenario are first computed and afterwards summarized to one objective criterium. In this case, objective function values of all scenarios are weighted equally using a weighting factor w. The chosen characteristics lead to a balanced prioritization of all maneuvers and only serves as an example. However, in an industrial development process different weighting factors might be chosen depending on the requirements on the driving function. The summarized objective function $F(ps)$ is calculated from the scenario specific results $F_j(ps)$ following (6):

$$F(ps) = \sum_{j=1}^{6} w_j \cdot F_j(ps) \quad with \quad w_j = \frac{1}{6}, \quad j = 1, \ldots, 6 \quad (6)$$

For subsequent studies 11 parameters are chosen from the trajectory planning module that are expected to have a high influence on the planned trajectory. They mainly serve as weight factors for a cost function evaluating a trajectory's suitability. A more detailed description of the parameters and its influence with respect to lateral and longitudinal planning is given in Table 3 in the Appendix section. Simulations are executed on a performance cluster with four parallel operating machines (CPU 2.6 GHz, RAM 8 GB).

B. OPTIMIZER SELECTION

Due to their successful application for similar problems the GA and PSO are compared in order to select the most suitable method.

To keep computation efforts in a reasonable order of magnitude, the experimental study is conducted with a simplified calibration problem. However, the effect chain shown in Figure 1 remains unchanged to ensure validity of the results. Studies are performed using only one driving scenario (S3) and eight (P1 - P8) out of 11 calibration parameters. The 7-scale comfort evaluator is used as present in the original problem. As described above, optimizers can be parameterized and their performance influenced in terms of search strategies (see Section II-B) To ensure a comparability, both algorithms shall have a similar, balanced parameterization in terms of breadth-first and depth-first search.

The particle swarm optimization provides two parameters (c_1 and c_2) that have a direct impact on the search strategy (see eqn. (4) and (5)). For an equally weighted consideration of the orientation towards the global best and locally best solution, both factors are set to the same value: $c_1 = c_2 = 0.5$ [39].

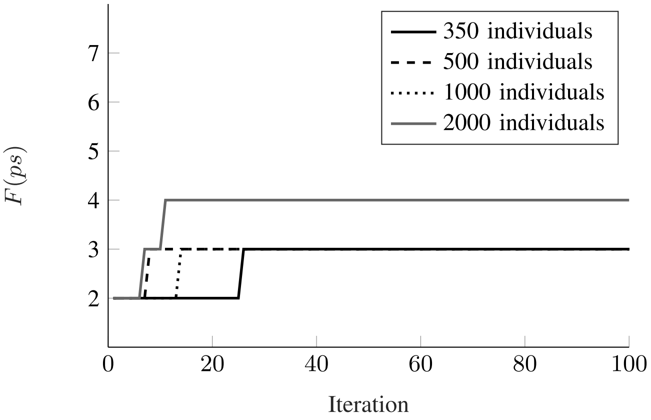


FIGURE 7. Optimization results for the particle swarm optimization.

On the other hand, the GA consists of more parameters that influence the characteristics of its operations. The selection probability p_s needs to be set so that convergence can be ensured but getting stuck in local optima is avoided. It is defined to $p_s = 0.8$ [40]. Recombination operations should also not be applied too frequently to prevent a too fast convergence. Therefore, p_r is set to 0.7 based on applications for comparative problems [15]. The probability p_m is set to 0.2 [41] to avoid turning the optimization into a random search. Parameter values are summarized in Table 4 in the Appendix section.

Next to the algorithm parameterization, optimization performance is mainly influenced by the population size (number of individuals). More parameter combinations per iteration increase the search space coverage but decrease convergence speed. A too low number of individuals causes higher risks to get stuck in local optima. Since the population size is used in both optimizers, it is systematically varied (350, 500, 1000 and 2000 individuals) for the experimental comparison study. The optimization is performed three times for each population to take the heuristic part of both optimizers into account. Only the best run is plotted for analysis. The initialization is not created by a random sampling but with a deterministic approach (Halton-sequence, [42]) that provides the same starting population for constant population sizes.

Optimization results for both optimizers are shown in Figure 7 and 8.

Both plots expose that the best solution is found using the highest population size whereas with less individuals per iteration only local optima are found. This observation coincides with previous research and can be explained with the fact that the density of samples in the search space increases with growing population sizes. Hence, it is more likely to find good solutions. On the other hand, choosing

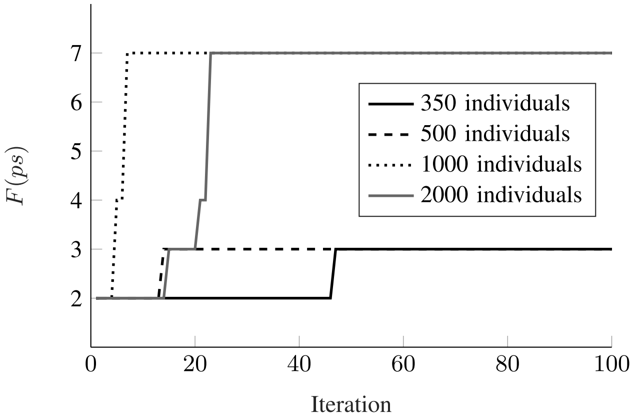


FIGURE 8. Optimization results for the genetic algorithm.

smaller population sizes can be beneficial in terms of computation time. However, stochastic variances are more likely when running the optimization algorithm repeatedly under the same conditions (see plots for 350 and 500 individuals as well as results provided in Table 5 in the Appendix section).

For further analyses both optimization algorithms seem promising. Even though the GA reaches higher objective function values, it can be assumed that the same performance is possible using an adapted PSO parameterization with respect to its breadth-first search operator. However, when comparing computation times of both optimizers it has to be taken into account that for the PSO more simulations have to be run even though the population size and iteration number are the same. The reason is, that the GA applies its genetic operators only to a selected subset of individuals whereas the other samples remain unchanged. Consequently, a new simulation run for an identical parameter combination in different iterations can be omitted. On the basis of this computation time advantage and observed superiority in terms of optimization result for the herein examined calibration problem the GA is chosen for further analyses.

C. OPTIMIZATION RESULTS AND COMPUTATION TIME

Using the GA, the optimization scheme as shown in Figure 6 is applied to solving the above defined calibration problem. As a first step prior to the actual optimization, a complexity reduction is applied to obtain scenario specific influential regions of the parameter space. Results are provided in Figure 9 through spider plots following the same notation as in Figure 5. Parameters without any gray segments are globally non-sensitive and non-influential along the whole domain.

The results expose that sensitive areas vary between scenarios. Thus, a consistent reduction of parameter bounds for all maneuvers can not be defined prior to optimization. Instead, parameter specific influence regions are used toreduce the number of simulations following the optimization scheme in Figure 6. The implemented GA uses the aforementioned bit representation with discretizations and bit numbers as defined in Table 6 in the Appendix section. The parameterization of genetic operators such as selection, crossover and mutation is the same as in the comparative study but with a stronger mutation operator ($p_m = 0.5$ [43]) to enable a distinct breadth-first search. Initialization is performed using latin hypercube sampling. The optimization scheme is repeated with 500, 1000 and 2000 individuals per iteration for 20 iterations in total. Since a population size of 1000 provides the best performance with respect to optimization result and computation time, its results are further discussed. Outcomes of the optimization runs conducted with 500 and 2000 individuals are provided in Table 7 in the Appendix section.

Fig. 10 shows the composition of each population over the iterations in the left plot.

Interestingly, at least one solution with an objective function value equal to the final optimum is already found in the first iteration (‘maximum’ line). On the other hand, the mean plot proves that the population in total seems to improve so that more alternative parameter combinations with high function values appear over time. The graph to the right in Figure 10 shows the total number of alternative samples with $F(ps) = 5.67$, which is the result for the best solution. It can be confirmed that with increasing number of iterations more parameter combinations with equal function values are found. Since the number of alternative solutions does not seem to change after the 15th iteration, convergence can be assumed.

Next to optimization results, computational efficiency is analyzed. The accumulated number of required and saved scenario runs is shown in Figure 11. It can be seen that the total number of simulations per iteration is 6,000 (1,000 individuals and 6 scenarios) in the first iteration but smaller in the following. That is due to the computation time advantage of GAs, that only select a subset of individuals for recombination and mutation while keeping remaining parameter combinations untouched. Thus, the requested number of simulations by the GA is not 120,000 (6 x 20 x 1,000) but approximately 43,000. With a duration of 10 s per evaluation in average, an original computation time of 116 hours would be necessary without usage of the scenario simulation reduction approach. However, the number of simulations could be reduced to 36,471. Consequently, computationally expensive evaluations of 6,416 parameter combinations could be avoided but qualified reference sets used instead. That leads to a computational advantage of approximately 17.82 hours which equals an efficiency increase of 15.36% for this optimization problem.

It should be noted, that computational benefits of this method are mainly dependent on the size of insensitive areas which influences the chance to find valid references during optimization. However, if further analyses with the same setup but for example changing weighting factors of the scenarios (see (6)) are performed, the same database DB can be reused and extended. As a result, the number of saved

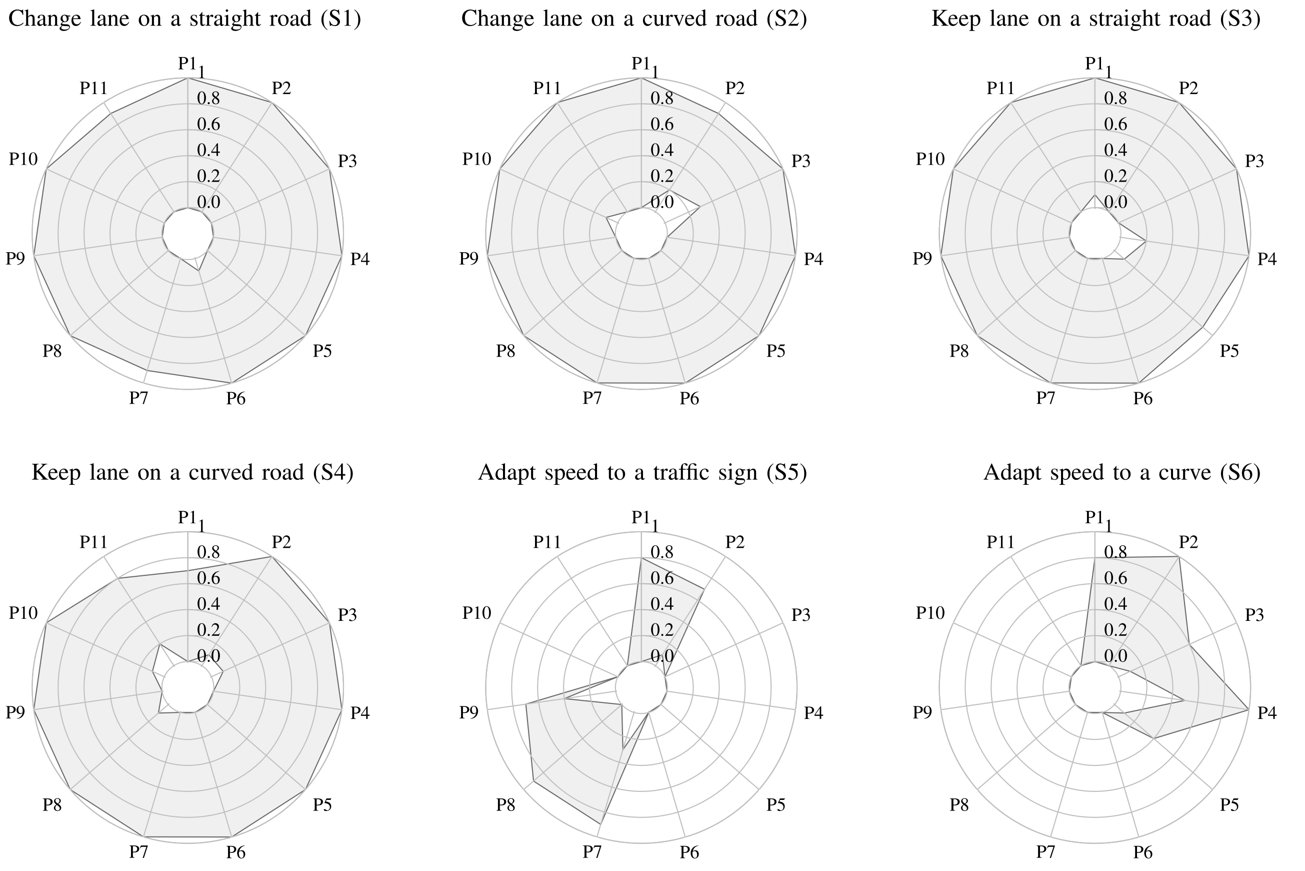


FIGURE 9. Result of the complexity reduction method. Gray areas represent influential regions in the parameter space.

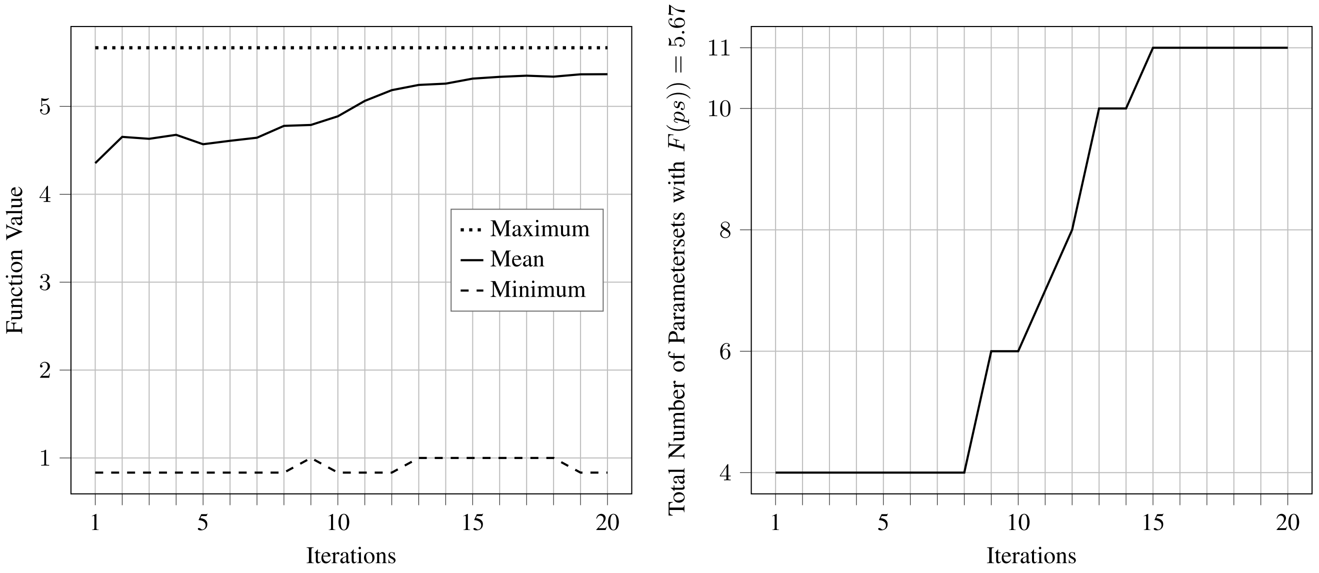


FIGURE 10. Overview of the population distribution per iteration (left) and number of parametersets with a function value of 5.67 (right).

scenario runs increases the more often an optimization is executed. Since requirements for driver assistance systems frequently change throughout the development process, the same optimization with changing characteristics is often conducted. Therefore, computational efficiency might not only increase by 15.36% as for a onetime execution but much more considering the repeated optimization with the same setup.

D. ROBUSTNESS ANALYSIS

The optimization allowed finding 11 equivalent parameter combinations that all enable an objective function value of

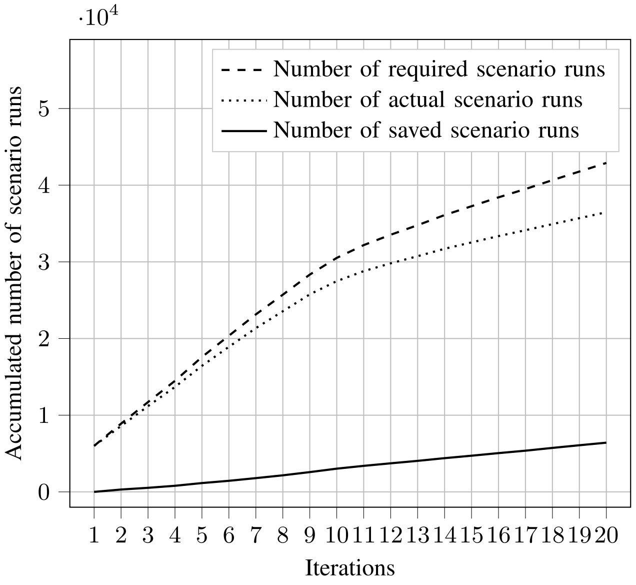


FIGURE 11. Accumulated number of saved scenario simulation runs.

$F(ps) = 5.67$ (see Figure 10 right). To further rank these solutions, a robustness analysis with respect to vehicle model parameters is applied.

The vehicle model used for the simulation based optimization is a so called 'hybrid vehicle model' that consists of a single track model and a neural network (details see [44]). Since a variation of parameters related to the neural network would require a new training every time, the robustness analysis is performed based on single track model parameters. Values used during the optimization as well as parameter bounds for the following analysis are given in Table 8 in the Appendix section. The minimum hyperbox radius δ is defined to 0.01. Robustness is given if none of the scenario specific objective function values ($F_j(ps)$) in a $k\delta$-environment deviates more than 1 from the respective solution. Therefore, the robustness condition represents the minimum deviation possible in the discrete comfort scale. 11 different hyperbox radii are examined ($k \in 0, 1, \dots, 10$).

Fig. 12 shows the optimal parameter combinations (left) and corresponding robustness information (right). The actual values can be found in Table 9 in the Appendix section. Solid lines herein represent the highest robustness level whereas dashed lines stand for a very low level of robustness. It can be noted that those solutions that lay close to each other in the left plot seem to be stable (Degree of robustness ≥ 10) against small parametric changes or vehicle model dynamics, respectively. Those five parameter combinations have equal values for parameters P5 to P11 but vary slightly for P1 to P4. On the other hand, the six remaining sets lay at discrete locations in the parameter search space and are classified with a lower robustness level. The results expose a dependency between the location of parameter combinations and its robustness. Combinations laying close to each other in the calibration parameter space also tend to be more robust towards changing vehicle dynamics. Consequently, these parameterizations should be preferred for later usage on the target hardware. Optimal solutions with lower level of robustness might also perform well. However, the risk that a reliable transferability to the real setup is not given is much higher since their performance is expected to change rapidly with changing environment conditions.

E. DISCUSSION OF THE RESULTS

When analyzing results of the complexity reduction method (see Figure 9) it can be noted that influential areas are comparably large for scenarios 1-4 and smaller for S5 and S6. That is due to interdependencies among parameters for S1-S4 which result in larger combined sensitivities s^{rel}. Since the threshold s^{rel}_{min} was conservatively estimated to avoid neglecting influential effects, these interactions are assessed relevant for almost all parameter domains. Even if there is only one highly interacting parameter, chances are high that the search space cannot be much reduced when using a conservative sensitivity threshold. Consequently, the computational efficiency advantage for the optimization decreases. To avoid such limitations, it might be beneficial to derive a sensitivity threshold experimentally. Therefore, s^{rel}_{min} might turn out to be larger and more irrelevant parameter effects with respect to the objective function can be neglected. Moreover, the method's potential can be improved if each SA metric is evaluated individually using separate thresholds for μ and σ. Especially if main effects dominate interdependencies such as for robust models, assessing SA metrics separately might outperform using combined sensitivities and equal weights for μ and σ in terms of parameter space reductions. However, the calculation of valid sensitivity thresholds is computationally expensive and has to be performed for each optimization problem individually.

The optimization progress as shown in Figure 10 exposes that optimal solutions can already be found in the first iteration when samples are distributed randomly. This characteristics is not typical since the goal of optimization algorithms is to guide individuals towards optimal solutions so that the best parameter combinations appear after several iterations. One reason for this characteristics is the discrete scale of the objective function. Since $F(ps)$ can only take a limited number of values (from 0 to 7), gradients are very steep and solutions can only improve in whole numbers. Thus, chances of randomly hitting a good parameter combination with an equal function value as the final optimal solution are higher. Moreover, characteristics of the parameter space contribute to finding optimal solutions with random searches as optima do not necessarily lay close to each other but can be found at distinct places in the parameter search space (see Figure 12 left). Most of the optimal solutions are found between iteration 8 and 15 (see Figure 10). At these steps, the GA finds many well rated solutions at the same time. It is likely that this progress was caused by the mutation operator since the number of good solutions and population mean rapidly increases, which indicates that

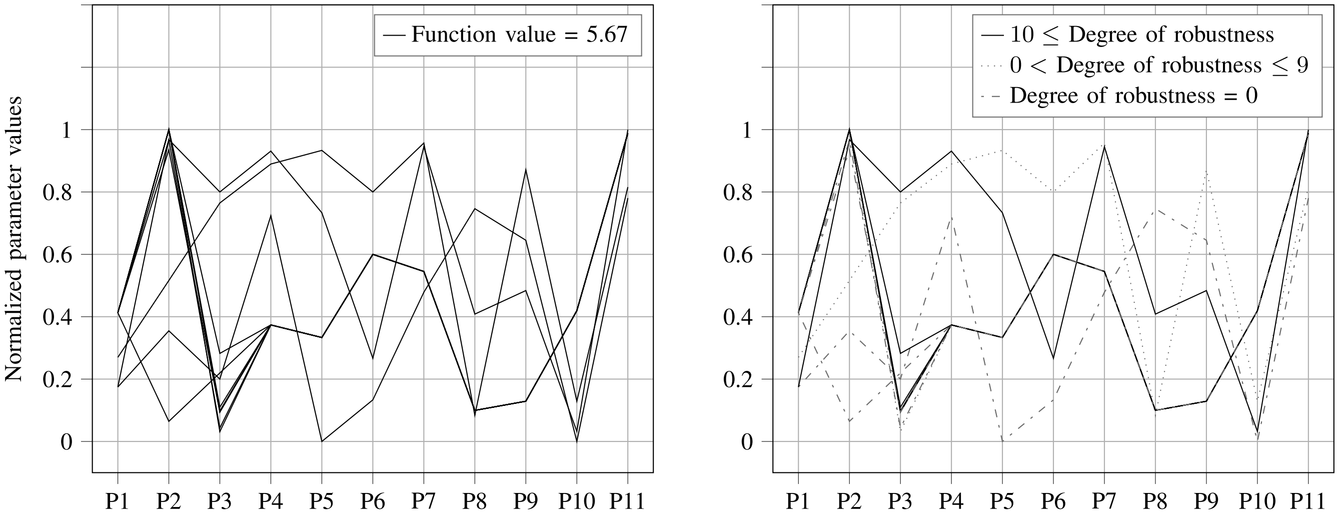


FIGURE 12. Best solutions found during optimization (left) and robustness information (right) for the parameter combinations. Values can be found in Table 9 in the Appendix section.

a new promising area was exploited. The 'minimum' line remains almost unchanged which can be reasoned with the aforementioned characteristics of the search space. Since high and low objective function values lay close to each other the minimum value does not change even though the optimizer identified areas that comprise many good solutions. The GA's high weighted mutation operator causes a quasi-random search for new candidates that often ends in finding parameter combinations with worse assessments and not always provides an improvement.

Findings presented in this section confirm that an efficiency increase for the simulation based optimization of automated driving functions can be achieved by applying the presented scenario simulation reduction approach. The transferability of virtually found parameter combinations to the real environment is ensured by analyzing robustness of final solutions towards changing model parameters. Providing more than one promising solution enables an increased flexibility for manual calibrations in the vehicle. The method is scalable so that an arbitrary number of scenarios or driving functions can be added. Also, solving a multi-objective optimization problem of this kind could be improved using this approach.

Whether an application for other optimization problems is useful depends on the characteristics of the problem. As mentioned before, this method aims to reduce the number of computationally expensive simulation runs. For an objective function evaluation that can be done quickly (e.g., an analytic function) this approach provides less advantages. Moreover, benefits only occur if the complexity reduction method computes a reduced parameter space compared to initially defined bounds. However, this can be expected for most vehicle control systems as concluded in previous work [20]. Theoretically, an optimizer can be freely chosen but algorithms using a discrete instead of a continuous scale contribute most to finding scenario/parameter-set-combinations whose simulations can be avoided. The reason is that there is only a finite (but arbitrarily high) number of samples available in the parameter search space. Consequently, chances for creating an individual that only differentiates along one or two dimensions but has otherwise identical values as a reference combination is higher than for optimizers using continuous parameter domains. For continuous scales it is recommended to define a maximum allowable deviation of parameters laying in sensitive areas to qualify parameter combinations for replacements with database values.
Especially recent aspirations of companies that develop software products to use simulation tools for testing and safety assessments, qualify for usage of this approach. Even though large computation clusters exist, the amount of simulations required is still high to enable full test coverage. Next to calibrating driving functions, the scenario simulation reduction approach could for example support safety assessments by minimizing the number of scenario parameter variations necessary for finding critical scenarios. In general, scenario based optimization approaches of any kind (such as for calibrating vehicle control systems, aviation systems, traffic control, ...,) might benefit from the optimization scheme introduced in this contribution.

V. VALIDATION OF THE METHOD

To validate presented optimization scheme it has to be tested if an error occurs when applying scenario simulation reductions according to the introduced method. Therefore, the validation studies intend to check if by taking objective function values for an arbitrary parameter combination (qualified for scenario simulation reductions according to Figure 6) from a reference set, errors remain in a negligible order of magnitude. Therefore, it could be tested how optimization results differ when running the GA with scenario simulation reductions and without. However, it has to be taken into consideration that the GA comprises random genetic operators and is not deterministic. For that reason, a systematic sampling approach is chosen instead to examine isolated effects of replacing scenario simulations with reference runs.

For the design of a reasonable validation study, underlying methods for the optimization scheme need to be taken into account. Identifying sensitive parameter regions is based on the local application of the EEM (see Section II-A). Sensitivity metrics σ and μ represent averaged main effects and interdependencies (see (1) and (2)). The risk is given that through averaging extreme values are neglected. That happens for example, if all elementary effects that were evaluated for a certain interval are zero except one being 0.1 . Let's assume 100 elementary effects are evaluated ($r = 100$), then s^{rel} would be 0.01 (see (3) and (1)). With a sensitivity threshold of, e.g., $s^{rel}_{min} = 0.02$, the associated parameter interval would be assessed non-influential. However, if the case appears that the one influential parameter combination is generated during optimization, it would be mistakenly qualified for a scenario simulation reduction and the objective function value taken from a reference set. As a result, optimization progress can be disturbed.

The meaning of main effects and interdependency orders in the scenario simulation reduction method is explained by using a fictive example shown in Figure 13. The plots are not related to the results described in previous sections but serve as an example to derive the validation approach. In this example, the parameter search space could be reduced for six parameters (P3, P4, P6, P9, P10, P11). Remaining parameters are influential along the whole domain and need to have the same value in a potential reference combination (see Figure 6). Since deviations for these parameters are not allowed for the scenario simulation reduction method, errors in their sensitivity information cannot influence optimization progress in a negative way. However, erroneous sensitivity values for the other six parameters can have an impact on the optimization result so that they are extensively examined during the following validation studies.

Since six parameters are affected by the parameter space reduction, six main effects for these parameters exist in this example. Moreover, interdependencies up to the 5th order can appear. How these effects become effective is shown in Figure 13. If a sample is only varied for one parameter at a time (first plot top left), main effects come into place. If more than one parameter value is varied at the same time, interdependencies are added to the main effects. For first order interdependencies (second plot), the initial sample (solid line) is varied along two dimensions (dashed and dotted lines). Accordingly, for 2nd order effects samples are varied along three dimensions and so on.

In order to check all relevant main effects and interdependencies between insensitive regions, an initial latin hypercube sampling (LHS) is performed inside the whole parameter space. Samples that qualify for a scenario simulation reduction (i.e., contain parameters laying in non-sensitive areas) are then extracted from the initial sample set. If that subset of samples covers all relevant parameter interdependencies and main effects for the respective scenario with a sufficient number of parameter combinations they are further processed. Otherwise, a new LHS with increased sample size is performed until all relevant effects are covered. It should be noted that the number of relevant effects $n_{Effects}$ increases exponentially the more parameters are affected by the complexity reduction ($n_{Effects} = 2^{n_{CR}} - 1$ with n_{CR} as number of parameters affected by the CR). Therefore, the subset of LHS-samples is larger the more effects (interdependencies and main effects) exist. Its size is denoted by n^*_{LHS}. Based on this subset, reference sets are generated for every LHS-sample that only vary randomly within non-sensitive regions compared to the LHS-basis (dashed and dotted lines in Figure 13). A total of n_{ref} reference sets are created for each LHS-sample so that the total number of validation samples $n_{Validation}$ per scenario can be calculated with

$$n_{Validation} = n^*_{LHS} \cdot n_{ref} \tag{7}$$

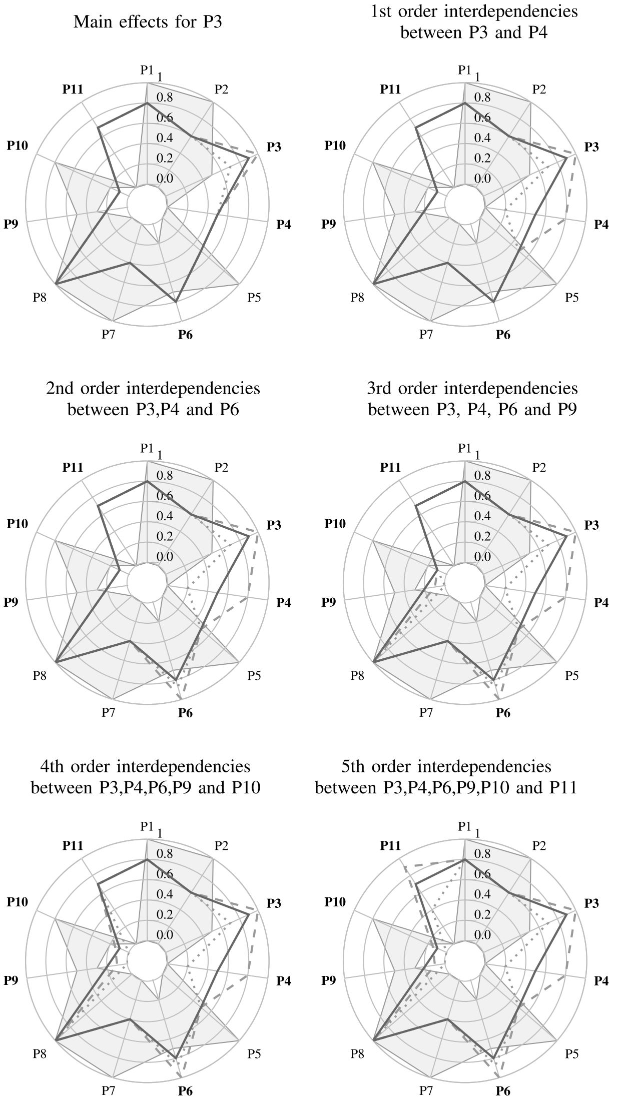


FIGURE 13. Exemplary visualization of samples for testing main effects and interdependencies of higher orders.

Table 1 provides an overview about sample sizes and numbers of insensitive regions per scenario.

For validating the scenario simulation reduction approach, all n_{ref} samples around the respective LHS-sample and the sample itself need to have same objective function values. To evaluate whether this requirement is fulfilled, objective function deviations of all samples from its related basis LHS-sample are computed and plotted for each scenario in Figure 14. It can be seen that the majority of samples provides a difference of zero from their related LHS-sample. To examine the impact of deviations larger than zero a maximum allowable deviation of 1 from the basis sample is defined (similar to the robustness condition in the previous section). Based on that the number of samples causing deviations larger than 1 denoted by $n_{\Delta F(ps)>1}$ is divided through the total number of samples ($n_{\Delta F(ps)>1}+n_{\Delta F(ps)\leq 1}$) defining the ratio of erroneous samples R_{err}:

$$R_{err} = \frac{n_{\Delta F(ps)>1}}{n_{\Delta F(ps)>1} + n_{\Delta F(ps)\leq 1}} \tag{8}$$

Ratios are calculated for each scenario individually and results are given in Table 2.

The most critical amount of erroneous samples occurs for scenario S4 with a result of $R_{err} = 0.0108 = 1.08 \quad \%$. If an optimization is performed with a population size of 1,000 and the unlikely case occurs that all 1000 parameter combinations qualify for a scenario simulation reduction, 11 sets would be allocated an erroneous objective function value for S4 in this case. Taking the worst case error ratios for all scenarios into account the maximum number of mistakenly evaluated samples would be 22 out of 6,000 ($\sum_{i=1}^{6} R_{err}^{i} \cdot 1000 \quad for \quad i = 1, \ldots, 6$). Since the GA is a population-based optimizer, those few mistakenly calculated samples compared to the total number of individuals are not expected to disturb optimization progress. Moreover, it is unrealistic that the scenario simulation reduction can be applied to 100% of the parameter combinations. The ratio of qualified samples for scenario simulation reductions was much smaller in studies described in this work (approx. 16% throughout all iterations, see Figure 11). 6,416 samples could be avoided in total. Assuming a worst case error ratio of $R_{err} = 0.0108$ (which actually only occurred for S4) for all scenarios a maximum of 65 samples were allocated an erroneous objective function value throughout the whole process. This number seems negligibly small against the background that 6,000 samples were processed per iteration which equals $6,000 \cdot 20 = 120,000$ individuals for the whole optimization. As a conclusion, validation studies confirm that a replacement of scenario simulations using the scheme in Figure 6 causes negligible errors that are not expected to have an influence on optimization progress.

VI. CONCLUSION

The goal of this contribution was to develop an efficient optimization scheme for a simulation based calibration of automated driving functions taking into account the transferability of solutions to the target vehicle. As the examined problem can be classified as a simulation optimization problem, objective function evaluations are computationally expensive. In this case they were represented by simulations of a representative scenario catalog. To minimize computational expenses, a scenario simulation reduction method was introduced that enables a replacement of costly simulations if the respective parameterization is expected to cause the same result as a previously evaluated parameter combination during optimization.

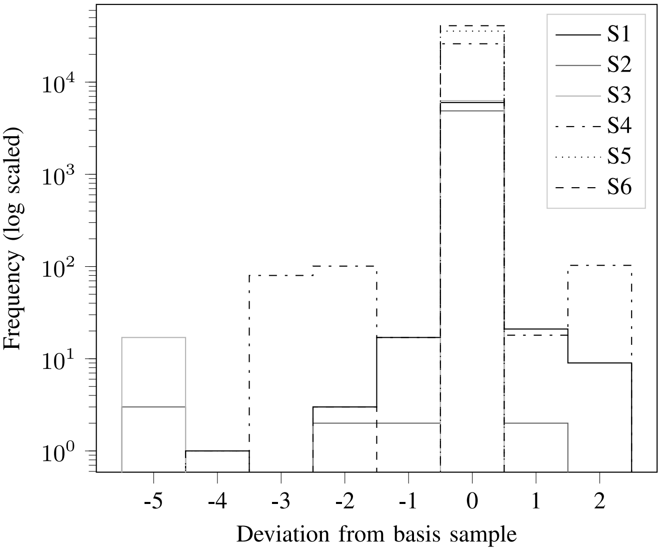


FIGURE 14. Objective function deviations per scenario to the respective basis LHS sample plotted in a stacked step histogram.

To evaluate efficiency of this approach, the method was applied for optimizing calibration parameters of a level 3 automated driving function. In order to find a suitable optimizer, a genetic algorithm and particle swarm optimization algorithm were experimentally compared with respect to optimization result and computation time. Since the GA outperformed the PSO in this case, it was used in combination with the scenario simulation reduction method to optimize the level 3 driving function. For this application example 11 parameters and a scenario catalog consisting of six representative maneuvers were considered. Parameters were optimized towards driving comfort.

The optimization provided a set of alternative calibration parameter combinations with equivalent function values. To further classify final solutions based on their stability towards changing environment conditions under real circumstances, a robustness analysis with respect to vehicle model parameters was applied. The results exposed a high degree of robustness for five out of 11 solutions laying close to each other and small robustness for the remaining six solutions that seem to be laying at steeper peaks in the parameter search space. For manual calibrations on the target hardware based on virtually obtained optimal parameter combinations, those solutions with larger robustness should be preferred whereas parameter combinations with smaller robustness degrees can rather be neglected.

TABLE 1. Sampling parameters for the validation.

Scenario	n_{CR}	n^*_{LHS}	n_{ref}	$n_{Validation}$
S1	3	207	30	6,210
S2	3	197	30	5,910
S3	3	210	30	6,300
S4	6	889	30	26,670
S5	11	1365	30	40,950
S6	10	1192	30	35,760

TABLE 2. Ratio of erroneous samples as exposed through validation studies.

S1	S2	S3	S4	S5	S6
0.0069	0.0014	0.0027	0.0108	0.0000	0.0001

TABLE 3. Description of the calibration parameters.

Parameter	Description
P1	Mutual smoothing factor for lat. and long. trajectories
P2	Weight factor for lat. planning
P3	Weight factor for lat. planning
P4	Weight factor for lat. planning
P5	Weight factor for lat. planning
P6	Weight factor for lat. planning
P7	Safety factor for lat./long. planning
P8	Safety factor for lat./long.planning
P9	Safety factor for lat. planning
P10	Safety factor for lat. planning
P11	Safety factor for lat. planning

TABLE 4. Parameterization of the GA and PSO for the comparative study of both optimizers.

Parameter		Value
PSO	c_1	0,5
	c_2	0,5
GA	p_s	0,8
	p_r	0,7
	p_m	0,2

TABLE 5. Optimization results for the comparative study of GA and PSO. Convergence is assumed if the best solution remains unchanged for at least 10 iterations.

Optimizer	Population size	Number of iterations until convergence is achieved	Best objective function value
PSO	350	26	3
	350	37	3
	350	1	2
	500	19	3
	500	8	3
	500	27	3
	1000	16	3
	1000	13	3
	1000	20	3
	2000	11	4
	2000	19	4
	2000	17	4
GA	350	56	3
	350	47	3
	350	1	2
	500	66	3
	500	13	3
	500	1	2
	1000	22	7
	1000	29	7
	1000	7	7
	2000	23	7
	2000	25	7
	2000	34	7

TABLE 6. Initial parameter ranges and discretizations for optimization.

Parameter	Lower Bound	Upper Bound	Discretization
P1	-15	14	0.5
P2	0	2	0.01
P3	0	200	1
P4	0	2,000	1
P5	0	1,500	1
P6	0	1	0.01
P7	0	200	1
P8	0	2,000	1
P9	0	10	0.1
P10	0	20	0.1
P11	0	200	1

TABLE 7. Optimization results for the GA applied to the calibration problem. Convergence is achieved if the number of alternative parameterizations remains unchanged for more than five iterations.

Population size	Best solution	Number of parameter combinations with F(ps) = 5.67	Number of iterations until convergence is achieved
500	F(ps)=5.67	6	12
1000	F(ps)=5.67	11	15
2000	F(ps)=5.67	11	10

TABLE 8. Parameter values of the single track model and parameter bounds for the robustness analysis.

Parameter	Initial value	Min.	Max.
Mass m	2060 kg	1200 kg	2600 kg
Cornering stiffness front axle $c_{\alpha,f}$	75000 $\frac{N}{rad}$	55000 $\frac{N}{rad}$	90000 $\frac{N}{rad}$
Cornering stiffness rear axle $c_{\alpha,r}$	150000 $\frac{N}{rad}$	120000 $\frac{N}{rad}$	180000 $\frac{N}{rad}$
Distance front axle - center of mass l_f	$1.66m$	$1m$	$2m$
Distance rear axle - center of mass l_f	$1.55m$	$1m$	$2m$
Yaw moment of inertia θ_z	3832 kgm^2	2000 kgm^2	4500 kgm^2

The herein described scenario simulation reduction approach allowed saving 6,416 scenario runs by replacing their function value with valid references. With approximately 42,000 required simulations in total an efficiency increase of around 15% could be achieved. A final large-scale validation study confirmed that negligible errors with respect to optimization progress occur when applying scenario simulation reductions according to the presented scheme.

The method derived in this work provides an efficient approach for solving scenario based optimization problems by suggesting a novel way of using sensitivity information to save simulation runs during each iteration. On the contrary to other contributions in the field of simulation optimization, the discrepancy between virtual and real environments is taken into account by assessing robustness of optimal solutions towards simulation model parameters. Comparable applications such as the calibration of other vehicle control systems might benefit from the approach presented in this work.

In further investigations other use cases could be examined. Especially the large number of scenario variations necessary to virtually perform safety risk assessments for automated driving functions offers potentials for this method to improve computational efficiency.

TABLE 9. Normalized values of optimal parameter combinations and corresponding robustness.

Parameter combination	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	DoG
1	0,175	0,355	0,2	0,724	0	0,133	0,478	0,746	0,645	0	0,78	0
2	0,413	0,968	0,043	0,374	0,333	0,6	0,545	0,1	0,129	0,419	0,988	0
3	0,27	0,516	0,765	0,89	0,933	0,8	0,957	0,084	0,871	0,129	0,816	1
4	0,413	0,064	0,22	0,374	0,333	0,6	0,545	0,1	0,129	0,419	0,988	2
5	0,175	0,968	0,8	0,931	0,733	0,267	0,945	0,408	0,484	0,032	1	2
6	0,413	1	0,031	0,374	0,333	0,6	0,545	0,1	0,129	0,419	0,988	4
7	0,413	1	0,094	0,374	0,333	0,6	0,545	0,1	0,129	0,419	0,988	≥ 10
8	0,413	1	0,098	0,374	0,333	0,6	0,545	0,1	0,129	0,419	0,988	≥ 10
9	0,413	0,935	0,094	0,374	0,333	0,6	0,545	0,1	0,129	0,419	0,988	≥ 10
10	0,413	1	0,282	0,374	0,333	0,6	0,545	0,1	0,129	0,419	0,988	≥ 10
11	0,413	1	0,11	0,374	0,333	0,6	0,545	0,1	0,129	0,419	0,988	≥ 10

APPENDIX

See Tables I–IX.

ACKNOWLEDGMENT

The authors would like to thank all of the developers from the Department of Automated Driving Functions of BMW for their support while developing the method described in this work.

REFERENCES

[1] "SAE on-road automated vehicle standards committee," in *Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems*, SAE Int., Warrendale, PA, USA, 2014.

[2] T. Helmerr, L. Wang, K. Kompass, and R. Kates, "Safety performance assessment of assisted and automated driving by virtual experiments: Stochastic microscopic traffic simulation as knowledge synthesis," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Las Palmas, Spain, 2015, pp. 2019–2023.

[3] D. Watzening, and M. Horn, *Automated Driving: Safer and More Efficient Future Driving*. Cham, Switzerland: Springer, 2016.

[4] H. Winner, W. Wachenfeld, and P. Junietz, *Validation and Introduction of Automated Driving. Automotive Systems Engineering II*. Cham, Switzerland: Springer, 2018, pp. 177–196.

[5] M. R. Zofka, S. Klemm, F. Kuhnt, T. Schamm, and J. M. Zöliner "Testing and validating high level components for automated driving: Simulation framework for traffic scenarios," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Gothenburg, Sweden, 2016, pp. 144–150.

[6] T. Weise, *Global Optimization Algorithms-Theory and Application*, Self-Published Thomas Weise, 2009.

[7] Y. Zhu, C. Sandu, D. Dopico, and A. Sandu, "Benchmarking of adjoint sensitivity-based optimization techniques using a vehicle ride case study," *Mech. Based Design Struct. Mach.*, vol. 46, no. 2, pp. 254–266, 2018.

[8] M. Diez, M. F. Campana, and F. Stern, "Design-space dimensionality reduction in shape optimization by Karhunen–Loève expansion," *Comput. Methods Appl. Mech. Eng.*, vol. 283, pp. 1525–1544, Jan. 2015.

[9] Y. Pan, C. Zhang, B. Yin, and B. Wang, "Fuzzy set based multi-objective optimization for eight-rod mechanism via sensitivity analysis," *Proc. Inst. Mech. Eng. D, J. Automobile Eng.*, vol. 233, no. 2, pp. 333–343, 2017.

[10] Y. Pan and L. Hou, "Lifting and parallel lifting optimization by using sensitivity and fuzzy set for an earthmoving mechanism," *Proc. Inst. Mech. Eng. D, J. Automobile Eng.*, vol. 231, no. 2, pp. 192–203, 2017.

[11] A. Saltelli, S. Tarantola, F. Campolongo, and M. Ratto, *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*, vol. 1. New York, NY, USA: Wiley, 2004.

[12] A. Saltelli*et al.*, *Global Sensitivity Analysis: The Primer*. Chichester, U.K.: Wiley, 2008.

[13] S. Kahlba and D. Bestle, "Optimal shift control for automatic transmission," *Mech. Based Design Struct. Mach.*, vol. 41, no. 3, pp. 259–273, 2013.

[14] G. Belgiorno, G. Di Blasio, and C. Beatrice, "Parametric study and optimization of the main engine calibration parameters and compression ratio of a methane-diesel dual fuel engine," *Fuel*, vol. 222, pp. 821–840, Jun. 2018.

[15] A. Sittig, G. Mumelter, and F. Rabenstein, "Development and calibration of hybrid electric vehicle operating strategies using simulations and a hybrid power train test bench," in *Proc. IEEE Veh. Power Propul. Conf.*, Chicago, IL, USA, 2011, pp. 1–5.

[16] A. Gosavi, *Simulation-Based Optimization*. Berlin, Germany: Springer, 2015.

[17] K. Nikzadfar, and A. H. Shamekhi, "More than one decade with development of common-rail diesel engine management systems: A literature review on modelling, control, estimation and calibration," *Proc. Inst. Mech. Eng. D, J. Automobile Eng.*, vol. 229, no. 8, pp. 1110–1142, 2015.

[18] S. Park, Y. Kim, S. Woo, and K. Lee, "Optimization and calibration strategy using design of experiment for a diesel engine," *Appl. Thermal Eng.*, vol. 123, pp. 917–928, Aug. 2017.

[19] H. Winner, S. Hakuli, F. Lotz, and C. Singer, Eds., *Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort*. Cham, Switzerland: Springer, 2016.

[20] N. Fraikin, K. Funk, M. Frey, and F. Gauterin, "Dimensionality reduction and identification of valid parameter bounds for the efficient calibration of automated driving functions," *Autom. Engine Technol.*, vol. 4, pp. 75–91, May 2019.

[21] M. D. Morris, "Factorial sampling plans for preliminary computational experiments," *Technometrics*, vol. 3, no. 2, pp. 161–174, 1991.

[22] I. Y. M. Sobol', "On the distribution of points in a cube and the approximate evaluation of integrals," *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, vol. 7, no. 4, pp. 86–112, 1967.

[23] F. Campolongo, A. Saltelli, and J. Cariboni, "From screening to quantitative sensitivity analysis. A unified approach," *Comput. Phys. Commun.*, vol. 182, no. 4, pp. 978–988, 2011.

[24] K. Deb, "Multi-objective optimization," in *Search Methodologies*. Boston, MA, USA: Springer, 2014, pp. 403–449.

[25] J. Pillas, F. Kirschbaum, R. Jakobi, A. Gebhardt, and F. Uphaus, "Model-based load change reaction optimization using vehicle drivetrain test beds," in *Proc. 14th Int. Stuttgarter Symp.*, Weilburg, Germany, 2014, pp. 1663–1673.

[26] R. Pfeil, M. Grimm, and H. C. Reuss, "Approach for the simulative evaluation of charging strategies for electrically-driven vehicle fleets," in *Proc. Shanghai Stuttgart Symp.*, 2017.

[27] H. Oschlies, "Komfortorientierte Regelung für die automatisierte Fahrzeugquerführung," in *Comfort-Oriented Control System for an Automated Lateral Keeping Function*. Heidelberg, Germany: Springer, 2019.

[28] M. Patrascu, A. F. Stancu, and F. Pop, "HELGA: A heterogeneous encoding lifelike genetic algorithm for population evolution modeling and simulation," *Soft Comput.*, vol. 18, no. 12, pp. 2565–2576, 2014.

[29] R. Eberhart, J. A. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Human Sci. (MHS'95)* Nagoya, Japan, 1995, pp. 39–43.

[30] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intell.*, vol. 1, no. 1, pp. 33–57, 2007.

[31] Y. Zhang, S. Wang, and G. Ji, "A comprehensive survey on particle swarm optimization algorithm and its applications," *Math. Problems Eng.*, vol. 2015, p. 38, Oct. 2015.

[32] C. Barrico, C. H. Antunes, "Robustness analysis in multi-objective optimization using a degree of robustness concept," in *Proc. IEEE Int. Conf. Evol. Comput.*, Vancouver, BC, Canada, 2006, pp. 1887–1892.

[33] K. Deb, H. Gupta, "Searching for robust pareto-optimal solutions in multiobjective optimization," in *Proc. 3rd Int. Conf. Evol. Multi Criterion Optim. (EMO-05)*, 2005, pp. 150–164.

[34] M. Ehrgott, J. Ide, and A. Schöbel, "Minmax robustness for multi-objective optimization problems," *Eur. J. Oper. Res.*, vol. 239, no. 1, pp. 17–31, 2014.

[35] J. Branke and C. Schmidt, "Faster convergence by means of fitness estimation," *Soft Comput.*, vol. 9, no. 1, pp. 13–20, 2005.

[36] S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence and Robustness*. Upper Saddle River, NJ, USA: Courier Corporation, 2011.

[37] K. Deb, and H. Gupta, "Introducing robustness in multi-objective optimization," *Evol. Comput.*, vol. 14, no. 4, pp. 463–494, Dec. 2006.

[38] U. Moser, N. Harmening, and D. Schramm, "A new method for the objective assessment of ADAS based on multivariate time series classification," in *Proc. 19th Int. Stuttgarter Symp.*, Wiesbaden, Germany, 2019, pp. 636–651.

[39] M. E. H. Pedersen, "Good parameters for particle swarm optimization," Hvass Lab., Copenhagen, Denmark, Rep. HL1001, pp. 1551–3203, 2010.

[40] H. M. Pandey, A. Chaudhary, and D. Mehrotra, "A comparative review of approaches to prevent premature convergence in GA," *Appl. Soft Comput.*, vol. 24, pp. 1047–1077, Nov. 2014.

[41] K. Deb and D. Deb, "Analysing mutation schemes for real-parameter genetic algorithms," *Int. J. Artif. Intell. Soft Comput.*, vol. 4, no. 1, pp. 1–28, 2014.

[42] L. Kuipers and H. Niederreiter, *Uniform Distribution of Sequences*, Courier Corporation, 2012.

[43] S. N. Sivanandam and S. N. Deepa, "Genetic algorithms," in *Introduction to Genetic Algorithms*. Heidelberg, Germany: Springer, 2008, pp. 15–37.

[44] N. Fraikin, K. Funk, M. Frey, and F. Gauterin, "A fast and accurate hybrid simulation model for the large-scale testing of automated driving functions," *Proc. Inst. Mech. Eng. D, J. Automobile Eng.*, vol. 234, no. 4, pp. 1183–1196, 2019, doi: 10.1177/0954407019861245.

NICOLAS FRAIKIN received the B.Sc. and M.Sc. degrees in automotive engineering from RWTH Aachen University in 2017. He is currently pursuing the Ph.D. degree with the Department of Automated Driving Functions, BMW Group, supervised by F. Gauterin (KIT). His research interests are simulation, optimization and control, and machine learning.

KILIAN FUNK received the Diploma degree in mechanical engineering from the Technical University of Munich in 1998, and the Ph.D. degree in technical mechanics in 2004. Since then, he has been with the BMW Group working in various research and development areas as a Product Developer and a Project Manager. He is currently a Senior Product Developer working on autonomous driving systems. His research interests are dynamical systems, optimization and control, modeling of complex systems, and software design.

MICHAEL FREY received the Diploma and Doctoral degrees in mechanical engineering from the University of Karlsruhe in 1993. He is the Manager of the Research Group Automation and of the Research Group Suspension and Propulsion Systems, Institute of Vehicle System Technology, KIT. His research interests are driver assistance systems, operational strategies, suspension systems, vehicle dynamics, and vehicle modeling and optimization.

FRANK GAUTERIN received the Diploma (M.Sc.) degree in physics from the Westfälische Wilhelms-Universität Münster, Germany, in 1989, and the Dr.rer.nat. (Ph.D) degree from the University of Oldenburg, Germany, in 1994. From 1993 to 2000, he was an Acoustics Engineer with Continental AG, Hannover, Germany, and from 2000 to 2006, he was the Director of noise vibration harshness (NVH) engineering with Continental AG with locations in Germany, USA, and Malaysia. Since 2006, he has been a Full Professor in vehicle science with the Karlsruhe Institute of Technology (KIT), the Head of the Institute of Vehicle System Technology, and a scientific spokesman for KIT-Center Mobility Systems. His research interests are vehicle NVH, tire acoustics, driver assistance systems, vehicle dynamics, and vehicle modeling and identification methods.