Optimization of System Parameters for an Online Driving Style Recognition

Dominik Dörr[1], Konstantin D. Pandl[2] and Frank Gauterin[1]

***Abstract*— An online driving style recognition system using fuzzy logic has recently proven to work well and showed potential to optimize its parameters. This paper is about the efficient parameter optimization of such a system. To overcome combinatorial explosion, we introduce heuristics to express the main influential parameters of the system, which itself is divided into two layers. First, we use a method called Design of Experiments in order to identify the most important parameters of general high-level system parameters. The low-level layer consists of fuzzy logic systems, which are the core of the driving style recognition system. For this, we introduce a way to efficiently describe the main characteristics of a fuzzy system by very few parameters. Both sets of identified parameters are then separately optimized with an established multidimensional evolutionary algorithm. We show that using Design of Experiments is superior to a random selection of the high-level parameters, as it increases the optimization gain by 76.5% in average. All in all, the target function, which represents a weighted classification error, was reduced by 43.9% on the test data set. The optimization method can be used to calibrate the system on real-world driving data. The combination of Design of Experiments, evolutionary optimization and fuzzy logic parametrization can also be used to optimize arbitrary other complex nonlinear systems.**

I. INTRODUCTION

A. Motivation

Systems which can take the driving style into account are becoming more widespread in the automotive environment. There are many applications that can get an added value by taking the driving style into account. To generate a signal for the driving style, a system is needed which classifies the driving style of the actual driver. These systems often have many parameters which can be adjusted. To generate a more suitable driving style signal, these parameters can be optimized. The driving style recognition system is very complex and has many parameters and nonlinearities, which makes it hard to model the system for optimization. Therefore it is not easy to find a suitable optimization method.

B. State-of-the-Art

There are many possibilities to generate a driving style signal. In [1] the driving style recognition is conducted by using the lateral and longitudinal acceleration. However it is also stated, that the current driving situation must be taken into account. Aljaafreh et al. [2] and Ly et al. [3] use the values from inertial sensors to determine the driving style or to distinguish between different drivers. Also smartphones can be used as sensors to identify a driver [4].

Optimization methods can be classified into direct methods, which use the target function value, and indirect methods, which use derivatives. For complex models, there might be no derivative like in the case described in this paper, so indirect optimization methods have to be used. Design of Experiments (DoE) is an established method for both, estimating the influence of a parameter on a system and modeling a system to optimize that model with ease, later.

Evolutionary Algorithms are a common way to optimize nonlinear systems, especially if the search space is complex and not very well known.

For fuzzy logic fine tuning, which is another part of the driving style recognition system, there are several approaches. Gürocak et. al. [5] treat fuzzy systems as a multivariate function and perform the optimization on it. Another common approach is the use of sensitivity analysis methods for a preselection of the parameters, as in Ruano et. al. [6]. The usage of evolutionary algorithms for fuzzy system optimization is also possible, a profound approach is given in Akbarzadeh-T. et al. [7].

C. Contributions of this work

This paper provides a method to optimize the parameters of the driving style recognition system, which was introduced in [8]. The method in fact is a combination of two established methods. First, a method called Design of Experiment is used to identify the most important parameters. Afterwards, these parameters are optimized by a multidimensional evolutionary optimization algorithm. After optimizing the inherited system parameters, the fuzzy logic itself, which is the core of the driving style recognition system, is optimized. Therefore, the fuzzy logic is described by only a few parameters. These parameters are then optimized by the same algorithm which was used for the system parameters.

II. SYSTEMS AND METHODS

A. The Driving Style Recognition System

The driving style recognition system regarded in this work is a system which analyzes the driving style of a driver and classifies it into three classes (comfortable, normal and sporty). More details on the system are given in [8]. The regarded driving style recognition system is an online system, which means that the driving style is computed while the vehicle is driving. It is implemented in Matlab/Simulink. We use Fuzzy logic to classify the different driving styles. A

[1]D. Dörr and F. Gauterin are with the Institute of Vehicle System Technology, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany {dominik.doerr, frank.gauterin}@kit.edu

[2]K. D. Pandl is a student of Karlsruhe Institute of Technology who has written his bachelor thesis at the Institute of Vehicle System Technology, 76131 Karlsruhe, Germany konstantin.pandl@gmail.com

detailed view into fuzzy logic and its use can be found in [9] or [10].
The system is divided into subsystems, while each subsystem generates a separate driving style signal for different driving situations regarding the class of road (urban, rural, highway) the car is driven on. There are many parameters in the system, for example different thresholds, weighting factors and duration of events. For the fuzzy logic, there are additional parameters. For this work a very simple version of fuzzy logic only with triangular and trapezoidal functions is sufficient. Therefore, the parameters are only the width of the triangles and trapezes and their center point. In Figure 1 an example of the membership functions for the longitudinal acceleration behavior on urban streets is shown. The red and solid trapezoidal function is the membership function of the variable sporty, the blue and dashed triangular function is the membership function of the variable normal and the black and dotted trapezoidal function is the membership function of the variable comfortable. The red arrows indicate half of the width of the triangle and the length of the gradient of the trapezoidal functions. The green circles indicate the center value of this fuzzy logic. The values of both the half triangular width and the center point are determined during the optimization of the fuzzy logic.

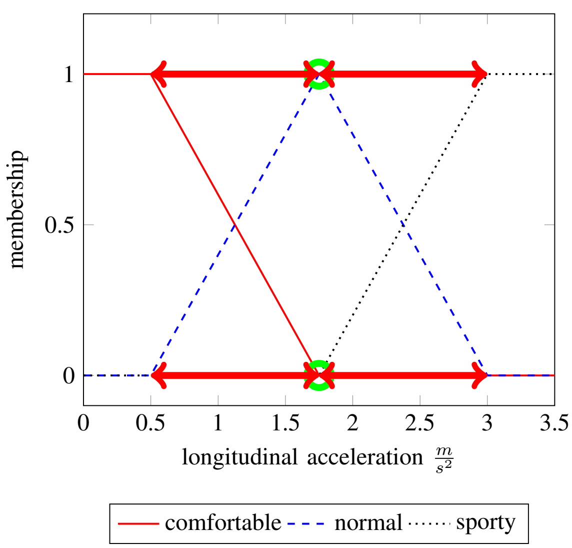

Fig. 1. Fuzzy parameters for input variable longitudinal behavior urban

B. Design of Experiments

DoE is a method to identify a statistical relationship between input parameters and output data of a system. An emphasis is put on the design which consists of several measuring points. Several types of design plans exist, a special focus is set on the size of the design plan which comes along with the cost of executing it.
DoE can be used for three purposes: First, one can identify the influence of a parameter on the output data and thereby decide whether the parameter influences the result significantly. Second, one can approximate a complex model - like the Driving Style Recognition System - by a mathematical model of lower complexity. From this model, one can then find a global optimum easily, e.g. by an indirect optimization method. In the regarded optimization problem, we can not expect the system to be approximated well by a simpler mathematical model since there are many nonlinearities. Their origin lies in the system itself. Fuzzy logic systems are non-linear because the fuzzy rules create non-linear functions. Additionally there are different subsystems and many inputs are used in more than one subsystem and each subsystem also combines many inputs. This creates a non-linear dependency between the inputs. Such a simple system would be strongly over-optimized on the training drive data. Third, one can create a space-filling design plan, evaluate the complex model at every point and consider the best resulting point as the optimum. In our case, we have 56 parameters with a relatively broad range for each. Therefore, such a space filling design plan would be too large to execute. However, it is possible to use the DoE for the first purpose, identifying the most significant parameters and optimize them in later steps. Non-significant parameters can be ignored in order to reduce the complexity of the optimization problem. We use the DoE instead of other methods for sensitivity analysis because it could handle the interactions between different inputs and a suitable design plan can be computed easily.

C. Optimization Algorithms

For the optimization, the Non-dominated Sorting Genetic Algorithm 2 (NSGA-2) is used. This is a multicriteria algorithm which allows us to optimize the system with multiple test drives in order to avoid over-optimization. The basic principle of such an Evolutionary Algorithm is shown in Figure 2.

The termination criterion can be a fixed amount of time or iterations or it may depend on the progress of the optimization. More details on the NSGA-2 are given in [11]. Since NSGA-2 is a multicriteria algorithm, the individuals are sorted into Pareto fronts before the selection step. Within such a Pareto front, it is not possible to select an individual with a better target function in one dimension without worsening the target function in another dimension. Usually there is one leading Pareto front and several Pareto fronts behind it.

The basic advantages of Evolutionary Algorithms are the universal applicability and the information exchange between good areas in the search space by crossing [12]. In our system we could not use a gradient based optimization algorithm. Because of the very non-linear system, such an algorithm would possibly go in the wrong direction or it can get stuck in a local optimum instead of finding the global maximum. This is why we need an algorithm which could escape from such a local optimum, which the Evolutionary Algorithms can achieve by crossing and mutation. The advantages of the NSGA-2 algorithm are the low complexity in the sorting into Pareto fronts and the usage of an elitist approach, which generally improves the speed of convergence of such

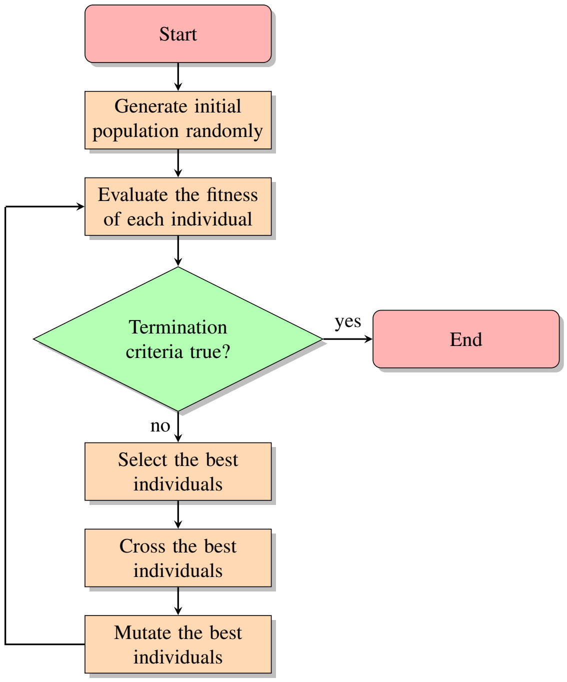

Fig. 2. The process of an Evolutionary Optimization Algorithm

an algorithm as stated in [13]. Furthermore, the NSGA-2 algorithm does not need the specification of a sharing parameter, which is required in other evolutionary optimization algorithms to ensure diversity in the population. The absence of such a parameter makes it easier to handle the algorithm and focus on other parameters [11]. These other parameters are especially the population size, the crossover rate and the mutation rate.

The binary tournament selection is used to select the best individuals, this means that two individuals are selected randomly and the one in the leading Pareto front wins. If the two individuals are in the same Pareto front, the one with the higher crowding distance wins. The crowding distance is a value calculated for every individual. The closer the surrounding individuals of an individual in the target function space are, the lower the crowding distance value is. The goal of this step is to search in areas of low individual density in the target function range. For crossing, the binary tournament selection is executed twice. We get two individuals which we cross and we get two crossings, this process is defined by equation (1) and (2)

$$x_k = \alpha_k * a_k + (1 - \alpha_k) * b_k \tag{1}$$

and

$$y_k = (1 - \alpha_k) * a_k + \alpha_k * b_k \tag{2}$$

a and b are the two individuals which should be crossed, we get the two individuals x and y, α is a randomly generated vector and k runs from 1 to the number of parameters of an individual.

In the mutation process, exactly one parameter of an individual is assigned a random value within a plausible and wide search range, predefined for each parameter.

Due to the probabilistic elements in the algorithm, the results might differ between several runs.

III. THE OPTIMIZATION PROCESS

A. Optimization goal

There are three classification states that can appear as the output of the driving style recognition system. The recognized driving style could be exactly the one that is performed. This is called `correct classification`. When the recognized driving style is next to the one which is performed, for example the recognized driving style is *sporty* and the performed one is *normal*, it is named as `differing classification`. A `wrong classification` occurs when the performed driving style is *sporty* and the recognized one is *comfortable* or vice versa.

As the optimization target and therefore also as measure for the quality of the optimization, a weighted sum of the wrong and the differing classifications is used. Since a `wrong classification` is worse than a `differing classification`, the weight of the `wrong classification` has twice the weight of the `differing classification`. The optimization goal is the minimization of that error function. Since the system is divided into two parts, two optimizations are done for each part. For the first, the general system parameters, Design of Experiments is done prior to the parameter optimization.

B. Identifying the most important parameters

We use DoE in order to identify the most important parameters. The driving style recognition system consists of three subsystems (urban, rural and highway) and some general system parameters, so 4 designs are used. The design plan should meet the following requirements:

- Continuous parameter search space: Since our parameter search space is continuous, the design plan should be able to handle this.
- Broad and uniform coverage of the search space: This requirement makes sure that every aspect of the system is considered when analyzing the design plan.
- Small size: Since the driving style simulation system is complex, we want to save executions by a small design plan size.
- Linear constraint: For some pairs of parameters a linear constraint is necessary, e.g. a lower boundary should be smaller than a corresponding upper boundary.

The complexity of a full factorial design grows exponentially with the number of parameters, therefore, we need a more efficient approach. Siebertz et al. [14] suggest Monte Carlo method, Orthogonal Designs, Latin Hypercube and Uniform Designs for the requirement of a uniform distribution of the test points in the search space. We use Latin Hypercube designs since they are less complex in creating compared to Orthogonal Designs and Uniform Designs and smaller than Monte Carlo designs [15]. The design is created using SAS JMP. This software can not handle our linear constraint requirement. To meet the requirement, the design is slightly modified after creation: Test points violating

the condition are permuted in order to fulfill these linear constraints.
The target function of the design is the 1:2 weight of differing and wrong classification portion since this is also the target function which we want to optimize later. Having the final design plan, the experiments are simulated and the screening report is calculated afterwards. In the screening report, we get an overview of the estimated influence of each parameter.

C. System parameter optimization

Now we can optimize the most significant parameters with the NSGA-2 algorithm. We use a population of 72 individuals, a crossover rate of 80% and a mutation rate of 35%.
In order to avoid over-optimization, we optimize in two dimensions, each dimension is the 1:2 weight of differing and wrong classification portion for the simulation of the driving style recognition system on one specific test drive.
We implement the NSGA-2 algorithm using parallel computing, which roughly improves the speed of optimization by the number of cores.
The termination criterion is fulfilled if all of the individuals are in the same Pareto front.

D. Optimization of fuzzy logic parameters

To optimize the fuzzy logic systems, we have to parametrize them first. As we have some lower order dependencies between the fuzzy systems itself through the general system parameters, we have a complex search space and we decided to use evolutionary optimization again, compared to other approaches as in [5]. We use some of the ideas in [7] in a simpler way to parametrize the main characteristics of a fuzzy logic system by few parameters, which allows us to optimize the fuzzy logic systems as well. The input function of every system consists of two trapezoidal shaped functions, representing the comfortable and sporty driving style, and a triangular shaped function, representing the normal driving style, as seen in Figure 1. We can parametrize those functions efficiently by defining a parameter representing the distance between the left/right upper boundary of the comfortable/sporty trapezoid and the apex of the triangle on the x-axis, as shown by the red arrows in Figure 1. Furthermore, we define a parameter for the location of the apex on the x-axis, shown with green circles, which is also the location of a feature point for each trapezoid. For some fuzzy systems, that parameter is always zero and does not need to be optimized, e.g. for the speeding behaviour, a value of 0 $\frac{km}{h}$ is the normal apex x-position of the triangle, since the triangle in the middle represents the normal and legal driving style. The very left end of the comfortable trapezoid and the very right end of the sporty trapezoid do not need to be parametrized since we can expect that the parameter settings are already optimal in those extreme scenarios. The relevant area for the optimization is only in the middle of the triangle and the corresponding trapezoid ends.
The output function of every fuzzy logic system consists of a middle triangle with a certain width. This width is the same for every system, so it adds one parameter.
All in all, we can efficiently parametrize the fuzzy systems by 18 parameters. Since we have weighting factors for every fuzzy system in the system parameters, DoE is not necessary here, as opposed to [6], for example, where the authors use a sensitivity analysis method for the preselection of parameters. If wished, we can reduce the 18 parameters by setting a minimum boundary on the weighting factor for which a fuzzy system will be optimized. However, we skip this step, as our previous system parameter optimization showed that the number of parameters is feasible.
Again, we use a population of 72 individuals, a crossover rate of 80% and a mutation rate of 35%. The termination criterion is the same as in the system parameter optimization.
It shows that in general randomly generated individuals of the fuzzy logic systems have a better target function value than randomly generated individuals for the system parameters. A possible explanation is that many system parameters influence the system behaviour early at the input level and effect the further progress of the system significantly. For this reason, we start with the optimization of system parameters first and then optimize the fuzzy logic parameters.

IV. RESULTS

The results were generated using the vehicle simulation software CarMaker. The used driver was parametrized with three different parameter sets for a comfortable, a normal and a sporty driving style. The driving styles were mixed in each simulation, so that the changing between different driving styles could be simulated, too.

A. Setup and parameters

In [8], we used three different parametrized driving styles on only one circuit to ascertain the function of the driving style recognition system. The driver parameters for the three different driving styles were the same in this work, but two additional circuits around Karlsruhe were used to prevent an over optimization to the used road. The three circuits used in this work are called *KIT-round, KIT-round-north* and *KIT-round-east*. The different routes were simulated with and without regular traffic, because the behavior of other road users has an enormous impact on the driving behavior of the ego-vehicle. For the traffic, additional driver parameters are taken into account.
In the beginning, a simulation with basic parameters was conducted. These parameters were determined by choosing average values from literature or by tuning them manually during different simulations. In the following optimization process, the system parameter optimization was conducted first. After that, the fuzzy logic parameters where optimized as a second step.

For generating the results shown below, we use three different data sets with different mixes of the driving style on three different roads. As training data sets we applied the *KIT-round* with *Mix 2* and the *KIT-round-east* with *Mix 3*. To check if the optimized parameters are applicable for other

roads and mixes, we used a third data set, called the test data set. For this test data set we chose the *KIT-round-north* with *Mix 1*.
Furthermore, the search space for each of the system parameters is defined. This is not only needed for the optimization itself but also for DoE. Exemplary general parameters are shown in TABLE I.

TABLE I
GENERAL SYSTEM PARAMETERS SEARCH SPACE

Parameter	**initial value**	**lower boundary**	**upper boundary**	**unit**
Time interval output control	500	50	1500	ms
Lower threshold normal driving style	1.7	1	2	–
Upper threshold normal driving style	2.3	1	3	–
Initialization time at start	6000	500	10000	ms

B. Results of Design of Experiments

DoE is performed for the system parameters, individually for the 4 general parameters and for the 3 subsystems which have 54 parameters in total. For the optimization, we consider a parameter to be relevant if its p-value is less or equal than 10% with respect to the weighted target function. Out of the 58 parameters, 19 turned out to have a significant influence and are considered further for the optimization.

C. Results after System parameter optimization

To test the effectiveness of DoE in the parameter selection, we optimize both, the 19 significant parameters and five times a random selection of 19 parameters, of which we calculate the average, and compare it.
After optimizing the significant system parameters identified with DoE, the weighted sum of wrong and differing classifications could be decreased by at least 35% and at most 61.5% in the training data sets and by 13.2% in the test data set. The results are shown in Figure 3. It shows that the optimization of the system parameters improves the performance on the training data a lot. The test data set is improved, too, but not that much as the training data sets.
After optimizing 19 randomly selected parameters, which we did five times, we get an average weighted sum of wrong and differing classifications of around 27% on every dataset. Over the average of the weighted error function of both, the two training data sets and the test data set, this means an absolute improvement of 7.9 percentage points without DoE or 13.9 percentage points with DoE, respectively. All in all, we see the improvement without DoE is much less on the training data and on the test data, there is no improvement at all when optimizing randomly selected parameters. DoE increases the absolute average optimization gain by 76.5%.

D. Results after fuzzy logic parameter optimization

After the optimization of the system which was optimized on its significant parameters by DoE, the fuzzy logic parameters described above are optimized with basically the

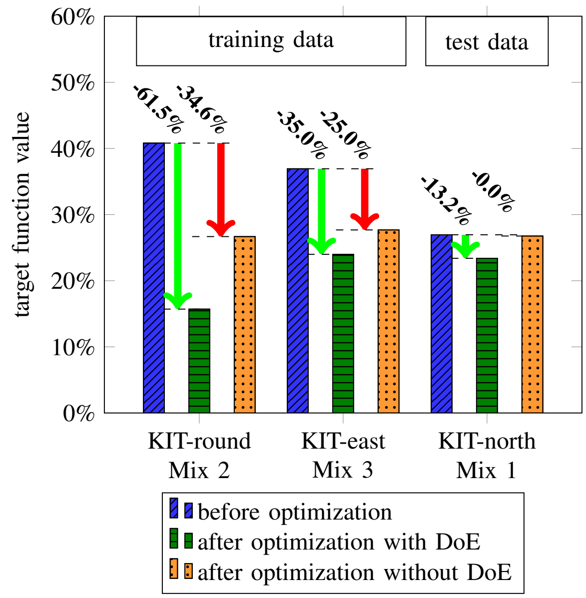

Fig. 3. Results system parameter optimization

same Evolutionary Algorithm. The weighted error could be further decreased with the algorithm terminating after 38 iterations. The results are shown in Figure 4. It can be seen that the optimization of the fuzzy logic parameters improves the test data set a lot more than the training data sets, which are nevertheless improved, too. This is contrary to the optimization of the system parameters. The training data sets are improved by 13.8% and 8.4% and the test data set could be improved by 35.3%.

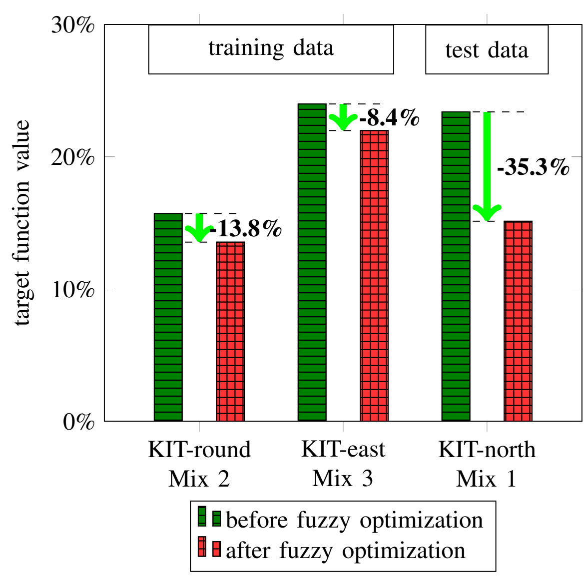

Fig. 4. Results fuzzy parameter optimization

E. Overall results

After completion of both optimizations the weighted sum of wrong and differing classifications could be decreased by 66.8% and 40.5% on the training data sets and by 43.9% on the test data set. The results are shown in Figure 5. They show that there is no over-optimization on the training data sets, because the test data sets are improved in a similar extend like the training data sets are.

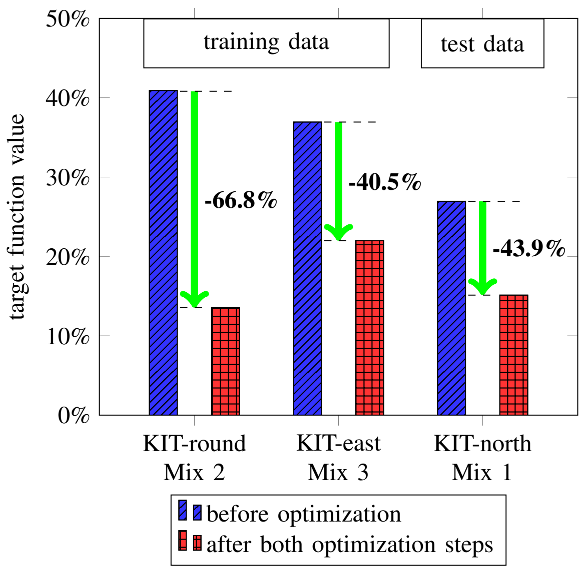

Fig. 5. Results overall

V. CONCLUSIONS

A. Conclusion

DoE has proven to be a good method to select which parameters to optimize, because without DoE the improvement in the error is much less after optimizing the system parameters. Fuzzy logic systems can be optimized by efficiently parameterizing the main characteristics of it, which has lead to significant improvements of the systems as well. The combination of both methods is a well suited tool for optimizing complex systems with a lot of nonlinearities like the driving style recognition system regarded in this work.

All in all, the target function, which represents a weighted classification error, was reduced by 43.9% on the test set. This equals a correct classification rate of around 85%. Regarding other methods in the literature, this is a good value. It is hard to compare different methods, because they differ in many ways. Some methods use discrete situations, which are evaluated, others are continuous like our method. But even the continuous methods can not be compared, because the ground truth is not the same. In many papers the results are not even given as numbers.

B. Outlook

The method could be expanded to an iterative or a parallel optimization. In the iterative optimization, two optimizations for the system parameters and the fuzzy logic parameters could be conducted in an iterative loop until a previously defined termination criterion is fulfilled. In the parallel optimization, system parameters and fuzzy systems are optimized in parallel. Special emphasis has to be set in the selection, which parameters should be optimized. The optimization could be extended to more routes or different cars, so that an over-optimization to these parameters could be prevented.

The next steps with the system are to validate it with real world data collected from a real car in real traffic and to implement it in a real car by using a rapid prototyping ECU. One problem when using real world data will be the generation of a reference signal for the driving style, because the reference signal from the simulation is not usable. But this is essential to perform the optimization. So it is important to carry on research in this field.

VI. ACKNOWLEDGMENT

The work presented in this publication is performed as part of the project ”e-volution” which is supported by the German Federal Ministry of Education and Research (BMBF), FKZ: 16EMO0071.

REFERENCES

[1] D.-I. B. Deml, D.-I. J. Freyer, and P. D. B. Faerber, “Ein Beitrag zur Prädiktion des Fahrstils,” in *VDI Berichte Nr. 2015*. VDI, 2007.

[2] A. Aljaafreh, N. Alshabatat, and M. Najim Al-Din, “Driving style recognition using fuzzy logic,” in *Vehicular Electronics and Safety (ICVES), 2012 IEEE International Conference on*, July 2012, pp. 460–463.

[3] M. V. Ly, S. Martin, and M. Trivedi, “Driver classification and driving style recognition using inertial sensors,” in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, June 2013, pp. 1040–1045.

[4] D. Johnson and M. Trivedi, “Driving style recognition using a smartphone as a sensor platform,” in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, Oct 2011, pp. 1609–1615.

[5] H. GÃ$\frac{1}{4}$rocak and A. de Sam Lazaro, “A fine tuning method for fuzzy logic rule bases,” *Fuzzy Sets and Systems*, vol. 67, no. 2, pp. 147 – 161, 1994. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0165011494900833

[6] M. Ruano, J. Ribes, G. Sin, A. Seco, and J. Ferrer, “A systematic approach for fine-tuning of fuzzy controllers applied to {WWTPs},” *Environmental Modelling & Software*, vol. 25, no. 5, pp. 670 – 676, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364815209001212

[7] A. H. M. Akbarzadeh-T, *Intelligent Control Systems Using Soft Computing Methodologies*. CRC Press, 2001, ch. Chapter 19 EVOLUTIONARY FUZZY SYSTEMS.

[8] D. Dörr, D. Grabengiesser, and F. Gauterin, “Online driving style recognition using fuzzy logic,” in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, Oct 2014, pp. 1021–1026.

[9] H. Zimmermann, *Fuzzy set theory and its applications*. Kluwer Academic, Dordrecht, 1985.

[10] L. A. Zadeh, “Fuzzy logic,” *Computer*, vol. 21, no. 4, pp. 83–93, 1988.

[11] S. A. Kalyanmoy Deb, Amrit Pratap and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, vol. 6, no. 2, 2002. [Online]. Available: http://www.iitk.ac.in/kangal/Deb_NSGA-II.pdf

[12] I. Gerdes, F. Klawonn, and R. Kruse, *Evolutionäre Algorithmen*. Vieweg Wiesbaden, 2004.

[13] E. Zitzler, K. Deb, and L. Thiele, *Evolutionary Computation, 8(2)*. MIT Press Journals, 2009, ch. Comparism of Multiobjective Evolutionary Algorithms: Empirical Results, pp. 173–195.

[14] K. Siebertz, D. v. Bebber, and T. Hochkirchen, *Statistische Versuchsplanung*. Springer-Verlag Berlin Heidelberg, 2010.

[15] S. I. Inc., *JMP®11 Design of Experiments Guide*. SAS Institute Inc. Cary, NC, 2014.