Supplementary material for the paper "Fitness Functions for Testing Automated and Autonomous Driving Systems"

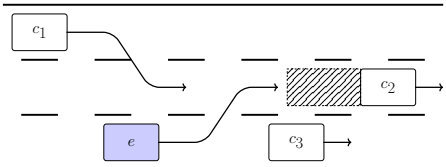
Florian Hauer[1], Alexander Pretschner[1], and Bernd Holzmüller[2]

[1] Technical University of Munich, Arcisstraße 21, 80333 Munich, Germany
{florian.hauer,alexander.pretschner}@tum.de
[2] ITK Engineering GmbH, Im Speyerer Tal 6, 76761 Ruelzheim, Germany
bernd.holzmueller@itk-engineering.de

Abstract. The work "Fitness Functions for Testing Automated and Autonomous Driving Systems" describes fitness function templates and how they can be combined and applied for testing named systems. It focuses on the methodological aspect. This document provides supplementary material for interested readers. The fitness function, which has been derived for the use case example of the main work, is applied on a technical level. It serves as input for search-based techniques to generate test cases.

1 Recapitulation

It is assumed that the reader has read the main work. Only the most important parts are recapitulated. In the main work's example (see Figure 1)), the ego vehicle e accelerates from standstill and approaches the other car c_3, which is driving at a lower velocity than e. Thus, e changes to the middle lane, while simultaneously c_1 changes also to the middle lane behind e. During this scenario, e must not violate the safety distances, e.g. the one to c_2 (shaded area in Figure 1). Each other car c_i has a parameter for its longitudinal starting position s_{0,c_i}, a starting time t_{start,c_i} for accelerating from standstill, and a desired velocity v_i it tries to reach and hold throughout the scenarios. In addition, the lane change of c_1 is triggered at a specific time, described by parameter t_{lc,c_1}. These parameters span a ten-dimensional space of possible test scenarios.



Paramter	Lower Bound	Upper Bound
Starting positions s_{0,c_i} [m]	0	300
Starting times t_{start,c_i} [s]	0	3
Target velocities v_i [km/h]	80	140
Lane change time t_{lc,c_1} [s]	6	13

Fig. 1. Parameterized scenario of a highway scenario with a schematic depiction as well as parameters and their domains, resulting in a ten-dimensional space of possible test scenarios

In the main work, the following fitness functions for single- as well as multi-objective search have been derived with the help of the presented templates: The combined fitness function for single-objective search does look as following. Powers of ten are used as offsets o_i, e.g. $o_1 = 10^3$ and $o_2 = 10^4$.

$$f_{single} = \begin{cases} \alpha + o_4, & e \text{ does not change lanes} \\ \begin{cases} \beta + o_3, & s_{c_2}(t_{e,start}) < s_e(t_{e,start}) \\ \begin{cases} \gamma + o_2, & s_e(t_{c_1,start}) < s_{c_1}(t_{c_1,start}) \\ \begin{cases} \delta + o_1, & \text{not simultaneous} \\ \epsilon, & \text{otherwise} \end{cases} \end{cases} \end{cases} \end{cases}$$

For an application of multi-objective search, the templates need to be changed, e.g. ϵ only can be computed if all scenario-specific properties underlying the other templates are fulfilled.

$$\widetilde{\epsilon} = \begin{cases} \infty, & \alpha + \beta + \gamma + \delta > 0 \\ \epsilon \end{cases}$$

For the experiments, ∞ is replaced by sufficiently large constant similar to o_i. By that, minimum and average fitness values of the optimization process can be shown. Preparing the other templates analogously yields the final vector of fitness values. α stays unchanged as it does not depend on other templates.

$$f_{multi} = [\alpha \quad \widetilde{\beta} \quad \widetilde{\gamma} \quad \widetilde{\delta} \quad \widetilde{\epsilon}]$$

The presented parameterized scenario, which constitutes a search space, together with the fitness function(s) serve as input for search-based techniques to yield test cases.

2 The System and the Experiment Setup

The widely used simulation tool CarMaker from IPG Automotive [5] is used for experiment purposes. Its MATLAB Simulink [1] API provided the means to connect the simulation with optimization, for which we used the single- and multi- objective optimizers of the MATLAB global optimization toolbox [2]. Note that the technological aspect is not the focus and more advanced techniques could be used. The population size is set to 60 and the number of generations to 15, resulting in 900 simulation executions, which means each system is tested in 900 test scenarios during the optimization. To show both the results of single- and multi-objective search, the first experiment will be conducted using single-objective search, the second one using multi-objective search. For highlighting the difference of a faulty and a "correct" system, the first experiment will be done using a faulty one, the second using a "correct" one. The experiments were executed multiple times to rule out randomization effects, but only one experiment run is presented.

3 The System

The architecture of our demonstration system follows the general control paradigm, which contains the notions of sensing, long-term planning (or decision-making), short-term planning, tracking and actuation [9,3,4]. Sensors are not used for the purposes of this work. The quantities that would be provided by the sensors are directly fed from the simulation to the decision-making. The system-internal control flow is depicted in Figure 2. The decision making computes whether a lane change is necessary or beneficial (using [8]). If so, the best gap and timing between cars on the best lane is chosen. In the case that a previously chosen gap changes significantly, re-computation is tried. If the gap became infeasible, the lane change is aborted. In the next iteration, a new gap may be chosen.

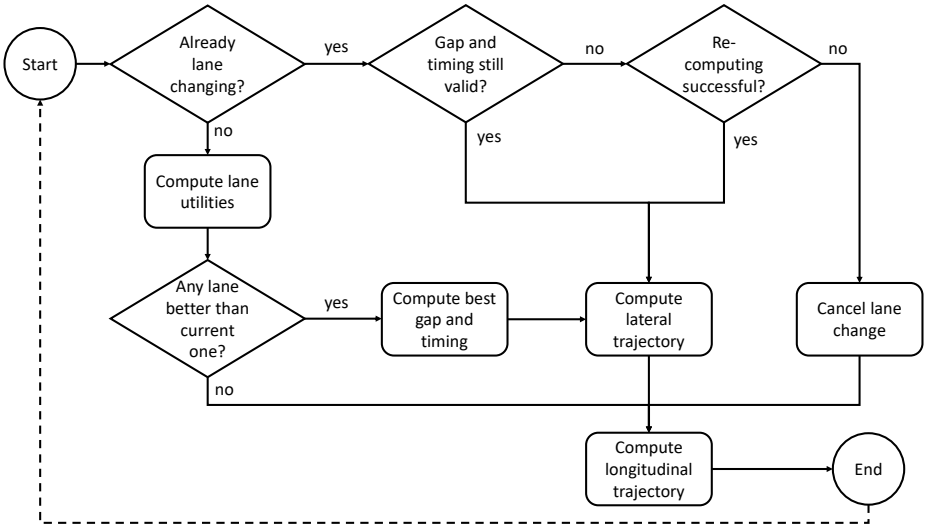


Fig. 2. System architecture overview

The short-term planning of longitudinal and lateral trajectories follows the concepts of a state-of-the-art approach [6,7] (see Figure 3). This model predictive control approach first predicts the positions x_i and velocities v_i of surrounding cars c_i for each sample time step t_k over a short time horizon (cf. Figure 3). For safety distance planning, a time gap τ is used. The safety corridor for the lane change (dashed green lines) is bounded by the distance $tg(v_i) = \tau \cdot v_i$ to the predicted positions x_i of other cars c_i. Quadratic optimization is used first to find a safe longitudinal and afterwards a safe lateral trajectory within these bounds and within the physical limitations of the ego car. The objective is to keep the velocity at each time step of the trajectory as close to the desired velocity of 120km/h and the acceleration as low as possible. The tracking of the trajectories

is done by a PID controller for the longitudinal velocity, controlling the position of the gas and the brake pedal, and by a PI controller for the lateral position, controlling the steering wheel angle. Both are tuned for a physical model of a sports car with a weight of 1,410kg, length of 4.18m, width of 1.83m and height of 1.35m. The model is provided by CarMaker.

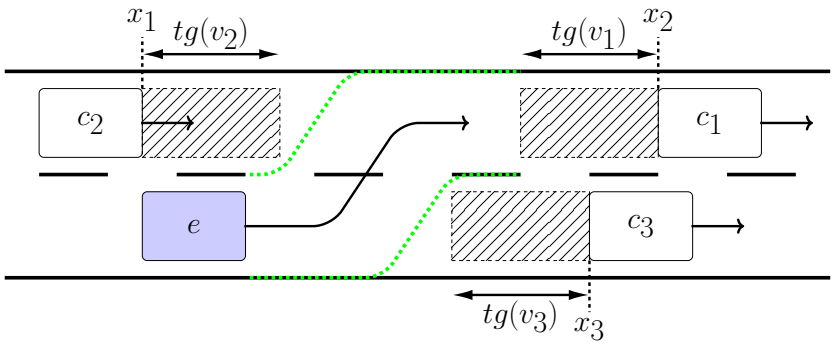


Fig. 3. Schematic simplified depiction of the predicted positions x_i and time gaps $tg(v_i)$ at a specific prediction time t_k

4 The Results

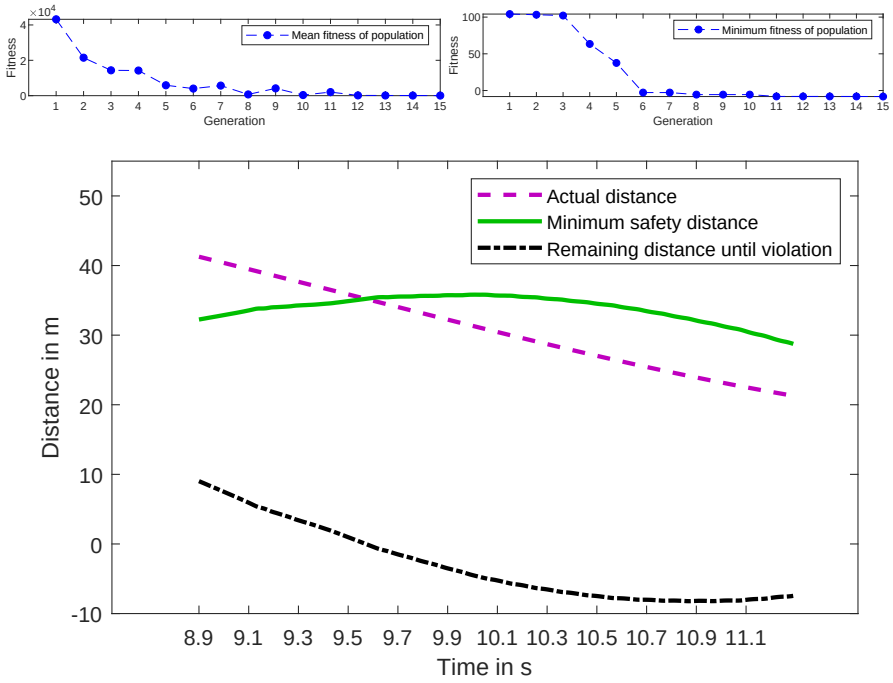
For both experiments, we show the minimum and average fitness of the population (of the genetic optimization) at each iteration. However, for the multi-objective optimization, the sum of all fitness values x_i of the fitness value vector is used instead of a single value. For the identified worst-cases, the distance of the ego vehicle to c_2 during the lane change of e is depicted, followed by the parameter values of the worst-cases.

4.1 Results of the Single-Objective Search with a Faulty System

The course of the minimum and average fitness indicates that the way of combining the templates to a fitness function for single-objective search does not hinder the optimization to converge. In the worst-case, the ego vehicle approaches c_3 and changes the lane behind c_2. Shortly after e started to change lanes, also c_1 changes lanes. Since e drives at a greater velocity than c_2, it needs to slow down and adjust to the velocity of c_2. However, as depicted in Figure 4, the remaining buffer until the violation of the safety distance is negative from scenario time 9.6s on with the minimum at 10.9s. This means that in the worst-case, the ego vehicle violated the safety distance maximal by 8.21 meters (the fitness value is -8.21). The system showed faulty behavior.

4.2 Results of the Multi-Objective Search with a "Correct" System

For the second experiment (see Figure 5), the system internal computation of the safety distance is changed such that the system keeps a greater safety distance. Similar to above, the course of the minimum and average fitness shows convergence. Due to the large numbers, the fitness improvements at later stages are difficult to identify. Similar to the worst-case in the first experiment, also this time, the ego vehicle is driving faster than c_2. However, even in the worst-case scenario, the remaining buffer does not get negative. Instead, 6.23 meters of distance are kept in addition to the safety distance. Therefore, the system is believed to be safe. The array of fitness values x_i is [0 0 0 0 6.23].

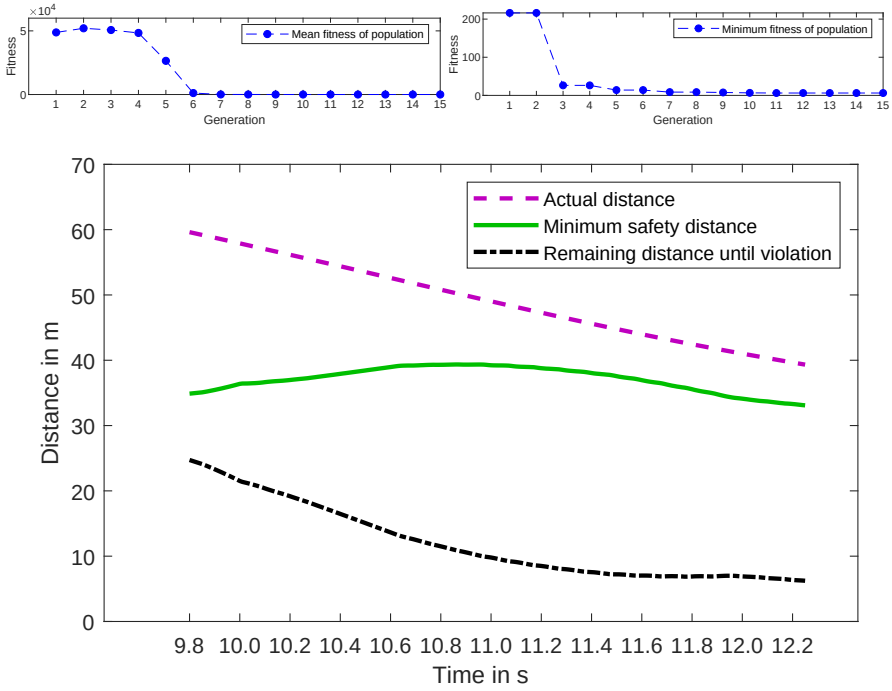


(a) Distances during the worst-case scenario of the first experiment

$s_{0,c_1} = 4.700$	$t_{start,c_1} = 2.531$	$v_1 = 24.439$	$s_{0,c_2} = 94.100$	$t_{start,c_2} = 2.778$
$v_2 = 27.106$	$s_{0,c_3} = 25.764$	$t_{start,c_3} = 0.364$	$v_3 = 25.906$	$t_{lc,c_1} = 8.059$

(b) Parameter values of the worst-case scenario of the first experiment

Fig. 4. Results of the first experiment: Faulty system



(a) Distances during the worst-case scenario of the second experiment

$s_{0,c_1} = 26.979$	$t_{start,c_1} = 1.842$	$v_1 = 28.938$	$s_{0,c_2} = 108.094$	$t_{start,c_2} = 2.373$
$v_2 = 33.634$	$s_{0,c_3} = 102.413$	$t_{start,c_3} = 2.537$	$v_3 = 30.678$	$t_{lc,c_1} = 10.052$

(b) Parameter values of the worst-case scenario of the second experiment

Fig. 5. Results of the second experiment: "Correct" system

5 Further comments

Videos that show the 3D animation of the worst-case scenarios can be found under the following URLs (the red car is the ego vehicle):

- Worst-case of experiment one with single-objective search and faulty system: https://mediatum.ub.tum.de/1474288
- Worst-case of experiment two with mutli-objective search and "correct" system: https://mediatum.ub.tum.de/1474289

References

1. MATLAB Simulink. The MathWorks Inc., Natick, MA, USA (2016b)
2. MATLAB Global Optimization Toolbox. The MathWorks Inc., Natick, MA, USA (341 2016b)
3. Aeberhard, M., Rauch, S., Bahram, M., Tanzmeister, G., Thomas, J., Pilat, Y., Homm, F., Huber, W., Kaempchen, N.: Experience, results and lessons learned from automated driving on germany's highways. IEEE Intelligent Transportation Systems Magazine **7**(1), 42–57 (2015)
4. Berger, C., Rumpe, B.: Autonomous driving-5 years after the urban challenge: The anticipatory vehicle as a cyber-physical system. arXiv preprint arXiv:1409.0413 (2014)
5. CarMaker: version 7.1. IPG Automotive GmbH, Karlsruhe, Germany (2018)
6. Nilsson, J., Brännström, M., Coelingh, E., Fredriksson, J.: Longitudinal and lateral control for automated lane change maneuvers. In: American Control Conference (ACC), 2015. pp. 1399–1404. IEEE (2015)
7. Nilsson, J., Brännström, M., Coelingh, E., Fredriksson, J.: Lane change maneuvers for automated vehicles. IEEE Transactions on Intelligent Transportation Systems **18**(5), 1087–1096 (2017)
8. Nilsson, J., Silvlin, J., Brannstrom, M., Coelingh, E., Fredriksson, J.: If, when, and how to perform lane change maneuvers on highways. IEEE Intelligent Transportation Systems Magazine **8**(4), 68–78 (2016)
9. Wei, J., Snider, J.M., Kim, J., Dolan, J.M., Rajkumar, R., Litkouhi, B.: Towards a viable autonomous driving research platform. In: Intelligent Vehicles Symposium (IV), 2013 IEEE. pp. 763–770. IEEE (2013)