

Uni-DAS

14. Workshop
Fahrerassistenz
und
automatisiertes Fahren

FAS 2022



Copyright Uni-DAS e. V.

Alle Rechte, auch das des auszugsweisen Nachdrucks, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

ISBN: 978-3-941543-65-2

Titelbild: Fahrzeuge UNICARagil, gefördert vom Bundesministerium für Bildung und Forschung

Uni-DAS e. V.
Otto-Verndt-Straße 2
64287 Darmstadt

Uni-DAS

**14. Workshop
Fahrerassistenz
und
automatisiertes Fahren**

FAS 2022

09. – 11.05.2022
Kloster Bonlanden, Berkheim

Vorwort

In diesem Jahr findet der Workshop Fahrerassistenzsysteme zum 14-ten Mal statt. Er hat sich mittlerweile als fester Bestandteil der „FAS-Community“ im deutschsprachigen Raum etabliert und bietet Expertinnen und Experten aus universitärem und industriellem Umfeld ein einzigartiges Forum zur interdisziplinären Diskussion. Der Workshopcharakter wird durch Verzicht auf Parallelveranstaltungen, Zeit für Diskussionen innerhalb und außerhalb der wissenschaftlichen Sessions, Kleingruppenarbeit zu einem aktuellen Thema, sowie die Beschränkung der Teilnehmerzahl geprägt. Dadurch ist der Workshop für viele etwas Besonderes, was sich nicht zuletzt in der fortwährenden Überbuchung der Veranstaltung widerspiegelt.

Zur Zeit des ersten Workshops im Jahr 2002 waren Fahrerassistenzsysteme erst kurz im Markt, und die wenigen verfügbaren Komfortfunktionen waren Oberklassefahrzeugen vorbehalten. Inzwischen sind diese Systeme in nahezu allen Fahrzeugklassen weit verbreitet. Gleichzeitig erwarten wir noch in diesem Jahr die erste Level 3 Funktion in einem Serienfahrzeug. Für viele Forscher und Entwickler stellt der Übergang von Assistenzfunktionen zum automatischen Automobil eine herausfordernde Aufgabe dar, deren Lösung in nicht mehr allzu ferner Zukunft erwartet wird. Die Ausgestaltung weiterer Fahrfunktionen auf dem Weg dorthin ist jedoch auch für Experten ungewiss. Unter dem Titel „Rolle und Ausgestaltung der Technischen Aufsicht“ sollen im Workshopteil der Veranstaltung mögliche Vorgehensweisen und Ausgestaltungen erörtert und potentielle Lösungsansätze und Antworten gefunden werden.

Die einzelnen Vorträge des Workshops und Beiträge dieses Tagungsbandes beleuchten ein breites Spektrum des aktuellen Wissensstandes im Bereich der Fahrerunterstützung und des automatischen Fahrens.

Unser Dank gilt allen Helfern bei der Organisation des Workshops, insbesondere Frau Annika Meyer und Herrn Till Beemelmanns sowie den Mitgliedern von Uni-DAS e.V.. Wir wünschen – nach langer Covid-bedingter Virtualität – allen Teilnehmern neue Einsichten, spannende Diskussionen und das Knüpfen neuer und Vertiefen alter Kontakte im realen Raum.

Karlsruhe und Aachen im Mai 2022

Christoph Stiller und Lutz Eckstein

Inhaltsverzeichnis

Prädiktion & Planung

1	Learning a Diverse and Cooperative Policy for Predicting Roundabout Traffic Situations M. Sackmann, H. Bey, U. Hofmann, J. Thielecke	1
2	Kontextbasierte Bewegungsplanung automatisierter Fahrzeuge an Kreuzungen mit vernetzten Lichtsignalanlagen J. Müller, J. Strohbeck, M. Mertens, M. Buchholz	11
3	Trajectories as Markov-States for Long Term Traffic Scene Prediction J. Reichardt	21
4	Approach to Maintain a Safe State of an Automated Vehicle in Case of Unsafe Desired Behavior C. Popp, S. Ackermann, H. Winner	35

Absicherung I

5	Aktualitätsverifikation semantischer HD-Karten für das urbane automatisierte Fahren J.-H. Pauls, C. Stiller	47
6	Über das verantwortliche Fahren Autonomer Fahrzeuge H.-P. Schöner	57
7	Analyse von Dekompositionsprozessen zur Umsetzung einer modularen Absicherung automatisierter Fahrzeuge B. Klamann, H. Winner	69

8	Funktionale Sicherheit KI-basierter Systeme im Automobilsektor D. Gesmann-Nuissl, I. Tacke	85
---	--	----

Perzeption

9	YOLinO: Echtzeitfähige Schätzung von linienförmigen Kartenelementen im Kontext des automatisierten Fahrens A. Meyer, C. Stiller	99
10	Understanding the Domain Gap in LiDAR Object Detection Networks J. Richter, F. Faion, D. Feng, P. B. Becker, P. Sielecki, C. Gläser	109
11	General Panoptics: Combining Semantic Segmentation and Classical Methods for a Fast LiDAR Panoptic Segmentation L. Beer, H.-J. Wünsche	119

Absicherung II

12	A Safety-Certified Automotive SDK to Enable Software-Defined Vehicles J. Becker	129
13	Towards «type approval» of automated vehicles: Means of safety validation and simulation-based methods, in particular W. Kröger, A. Ayoub	141
14	Inferenz von Verhaltensattributen der verhaltenssemantischen Szeneriebeschreibung für die Entwicklung automatisierter Fahrfunktionen F. Glatzki, H. Winner	151

Human Factors

- | | | |
|----|---|-----|
| 15 | Manöverbasierte Lenkgesten für das teil- und hochautoamtisierte Fahren: Ergebnisse und Ausblick
F. Flemisch, N. Herzberger, M. Usai | 165 |
| 16 | Longitudinal Acceleration during Lane Changes - A Human-Centered Investigation for Automated Driving
J. Ossig, S. Cramer, K. Bengler | 175 |
| 17 | Controllability of lateral drift failures while driving with SAE Level 2 Advanced Driver Assistance Systems
N. Schneider, L. Ahrens, A. Pruksch | 183 |

Absicherung III

- | | | |
|----|--|-----|
| 18 | OMEGAFormat: A Comprehensive Format of Traffic Recordings for Scenario Extraction
M. Scholtes, M. Schuldes, H. Weber, N. Wagener, M. Hoss, L. Eckstein | 195 |
| 19 | Scenario-based X-in-the-Loop Test for Development of Driving Automation
F. Reisgys, J. Plaum, A. Schwarzhaupt, E. Sax | 207 |
| 20 | How to Derive Behavioral Requirements for Automated Driving from a Behavior-Semantic Scenery Description
M. Lippert, H. Winner | 217 |

X

Learning a Diverse and Cooperative Policy for Predicting Roundabout Traffic Situations

Moritz Sackmann*, Henrik Bey*, Ulrich Hofmann† and Jörn Thielecke*

Abstract: Predicting other drivers' trajectories is challenging. We address the issue by introducing a method to derive a driving policy based on multi-agent reinforcement learning. For this, we let multiple vehicles interact in a roundabout scenario and reward desirable behavior. While typically, all vehicles follow the same policy, we foster diversity by assigning different preferences, e.g., cautious or sporty driving, to each vehicle during the training stage. These preferences are part of the policy network inputs as well as the reward function. This enables us to learn one single policy that can express different driving styles.

Keywords: Multi-Agent Reinforcement Learning, Behavior Modelling, Cooperative Behavior

1 Introduction

Predicting other drivers' trajectories by modeling their behavior is an important challenge in automated driving. One prominent application is cooperative behavior planning, i.e., estimating the influence of the own plan on other vehicles [1, 2].

A driver behavior model is a function that maps the current local observation of a driver to their next action. Repeated execution of the model coupled with a kinematic model leads to the prediction of the driver's trajectory. There are many possible ways to obtain driver models: Manual specification [3], learning from observed driver behavior [4–7], and reinforcement learning (RL) [8].

As the manual specification of driver behavior is a Sisyphean task, many recent works focus on learning driver behavior from observations of real-world driving. While the idea of directly learning to imitate driver behavior is appealing due to its conceptual simplicity, it suffers from accumulating errors [4–7], which we investigated in [4]. Moreover, the learned policy is inherently limited by the scope of the data used for training. It is unlikely to learn appropriate reactions to situations that are not represented in the training data, which for example leads to collisions of approximately 3% of all vehicles in the closed-loop simulation [4].

To sidestep these issues, this work builds upon the idea of RL, where the behavior model (policy) is learned by interacting with a simulated environment that can generate

*Institute of Information Technology, FAU Erlangen-Nürnberg, (e-mail: firstname.lastname@fau.de).

†Pre-Development of Automated Driving, AUDI AG, 85045 Ingolstadt (ulrich.hofmann@audi.de).

Acknowledgment: This work is a result of the research project @CITY – Automated Cars and Intelligent Traffic in the City. The project is supported by the Federal Ministry for Economic Affairs and Energy (BMWi), based on a decision taken by the German Bundestag. The author is solely responsible for the content of this publication.

an arbitrary amount of training data to improve the policy. During the simulation, each state transition is assigned a reward, and the goal of the procedure is to find a policy that maximizes the sum of rewards.

Outline When applying RL to learning a driving policy, a range of interesting questions arises: 1.) How can we derive a behavior model from a minimal set of assumptions, e.g., the goal to move forward and to avoid collisions? 2.) The interaction with other drivers is one key feature of behavior models. How can we learn a cooperative behavior model without any model of how other drivers behave and react to us? 3.) How can the behavior model be adapted to represent different types of drivers, e.g., more and less cautious ones?

The foundations of multi-agent reinforcement learning (MARL), questions 1 and 2, are discussed in section 2, while our approach to learning a diverse policy, question 3, is the subject of section 3. Finally, the properties of the learned policy are presented in section 4.

The core contribution of our work is the introduction of a method to train one single adjustable policy that can be used to represent different driving styles. Further contributions are the introduction of a reward function that fosters cooperative behavior as well as the application of the independent proximal policy optimization method [9] for learning the policy.

2 Technical Background

Reinforcement Learning The field of RL is concerned with solving partially observable Markov decision processes [10], which model sequential decision problems: one agent makes an observation $o \in O$ and performs an action $\alpha \in A$. As a consequence of his action, his state $s \in S$ transitions stochastically to the next state¹ $s^+ \sim T(\cdot|s, \alpha)$, according to the conditional transition density $T(\cdot|s, \alpha)$. Each transition is assigned a reward $\mathcal{R} : S \times A \rightarrow \mathbb{R}$. Each new state $s^+ \in S$ stochastically leads to a new observation $o^+ \sim \mathcal{O}(\cdot|s^+)$, forming the basis for the next decision of the agent.

In our case, each agent can select an action $\alpha = (a, \delta)$, composed of acceleration a and steering angle δ . The state transitions are performed through the kinematic bicycle model [11]. The observations are generated by a simulation model and describe the local environment of that agent. This procedure is visualized in Fig. 1. We assign the rewards through a manually defined function that rewards forward movement and penalizes collisions and leaving the track. The reward function is described in section 3. This function might also be learned from real data, for example proposed by [12], but this is beyond the scope of this work.

The goal of RL is to find a possibly stochastic policy $\pi(\cdot|o)$, i.e., which action to choose at which observation, that maximizes the sum of discounted rewards $g_k = \sum_{i=k}^N \gamma^{i-k} r_{i-k}$, starting from the initial state $k = 0$. The reward in the i -th step $r_i \in \mathbb{R}$ is determined by \mathcal{R} . We denote g_k as the return. The discount factor $\gamma \in [0, 1]$, typically close to 1, ensures a preference for policies that gather high rewards fast. Moreover, it supports the convergence of g_k for large numbers of steps N .

¹We follow the convention of [10] to use \sim as a sampling operator: $s^+ \sim T(\cdot|s, \alpha)$ indicates that s^+ is a sample from a random variable that is distributed according to the conditional density $T(X = s|s, \alpha)$.

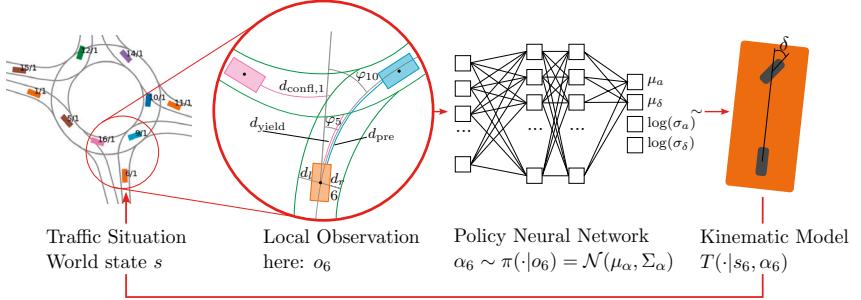


Figure 1: One step of the policy roll-out: The local observations of each vehicle are generated from the traffic situation. Observations include information on relevant vehicles, e.g., conflicting and preceding vehicles. The observation vector is the input to the policy network. An action is sampled from the action distribution. The kinematic model determines the next vehicle state. While this diagram visualizes the steps for vehicle #6, the same procedure is simultaneously executed for each vehicle in the situation.

Proximal Policy Optimization (PPO) PPO [13] is a policy gradient algorithm to solve RL problems. One central advantage of policy gradient methods compared to value-based RL approaches is the ability to handle continuous high-dimensional observation and action spaces [10]. For the sake of brevity, we introduce a simplified algorithm here and refer to the literature [13, 14] for a comprehensive description. At its core, two neural networks are responsible for learning the policy: The *policy network* π_θ with parameters θ predicts the mean and covariance ($\mu_\alpha, \Sigma_\alpha$) of an action distribution, in our case a 2D Gaussian distribution with diagonal covariance matrix. The *value network* $v_\phi : O \rightarrow \mathbb{R}$ with parameters ϕ predicts the expected value of an observation, i.e., the return g_k , when executing the current policy starting in state s_k .

After randomly initializing the policy and the value network, the following steps are executed repeatedly during one training epoch:

1. Collect multiple trajectories, i.e., policy roll-outs, by randomly sampling actions from the current policy $\alpha \sim \pi_\theta(\cdot|o)$ and interacting with the environment until termination, e.g., a collision. Calculate the relative likelihood $p = \pi_\theta(\alpha|o)$ of the sampled actions. Calculate the return g_k , starting from each state. Store each *experience* $e_k = (o_k, \alpha_k, p_k, g_k)$ without particular order in the current set of experiences $E = \{e_1, e_2, \dots\}$.
2. Estimate the advantage $A_k = g_k - v_\phi(o_k)$ of each experience in E and store it along with the experience. The advantage is a positive real number if the return g_k is higher than estimated by v_ϕ ; otherwise, it is negative.
3. Train the value network v_ϕ to predict returns g based on the current observation by minimizing $\sum_{(o,g) \in e \in E} (v_\phi(o) - g)^2 / |E|$ using gradient descent with respect to the value network parameters ϕ .
4. Train the policy network to increase the probability of selecting actions with positive

advantages \mathcal{A} and decrease the probability of actions with negative advantages. To do so, maximize $\sum_{(o, \alpha, p, \mathcal{A}) \in e \in E} (\pi_\theta(\alpha|o)/p)\mathcal{A}$ using gradient ascent with respect to θ .

The main principle of each training epoch can be explained as follows: Due to the stochastic sampling in step 1, some trajectories receive higher rewards than expected. Consequently, in step 2, the advantages of observations along these trajectories are positive, and the corresponding actions are reinforced in step 4, i.e., their relative likelihood $\pi_\theta(\alpha|o)$ is increased by shifting the predicted mean action μ_α towards the good actions α . Usually, Σ_α also decreases, which again increases the likelihood of α . Conversely, $\pi_\theta(\alpha|o)$ is pushed away from actions with negative advantages. Simultaneously, step 3 creates an updated baseline estimate of the expected returns of observations required to determine the truly advantageous actions in the next epoch. As the expected returns of an observation change when the policy changes, the value network needs to be updated along with the policy.

In practice, numerous improvements increase the stability of PPO, mainly by restricting the step length during the gradient ascent of the policy parameters, similar to a trust region optimization [13], and by reducing the variance of the advantage estimate [14].

Homogenous Multi-Agent Reinforcement Learning Until now, we have been focusing on a single agent learning to drive through a roundabout. Clearly, the interaction with other vehicles plays a central role in driver behavior models. This leads to three requirements: First, other vehicles must be part of the simulated world, i.e., being represented in the state vector s . Secondly, the observation model must give hints on relevant other vehicles. And thirdly, we need a model of how other vehicles behave to simulate the interaction with them.

Back to square one: After all, the goal was to learn a behavior model, but the learning already requires a behavior model. This paradoxical situation is resolved by multi-agent reinforcement learning (MARL): In a traffic situation, MARL treats each vehicle as an agent that interacts with the environment, which includes all other agents, to maximize its own reward. Recent work [9] introduces the independent PPO (IPPO) method, which decomposes the n -agent MARL problem into n single-agent RL problems that can be solved with PPO.

As we consider a situation consisting exclusively of cars, we previously trained a *single* policy used by all agents [8]. Thus, each agent executes the same policy, and the policy and value network are updated based on the collective experiences of all agents. This leads to homogenous behavior among agents, e.g., similar velocities in the roundabout and similar bumper-to-bumper distances in the queue before the entrance to the roundabout. In the following sections, we propose and evaluate a concept to foster heterogeneous behavior among agents to better reflect the diverse nature of human behavior in traffic.

3 Learning a Diverse Policy

We aim to find a safe and cooperative behavior model that can be applied independently by all agents in a roundabout traffic situation. Compared to previous works, we foster diverse behavior among agents by assigning different preferences $\rho_i = (\Delta t_{\min,i}, d_{\min,i}, \omega_{\text{acc},\text{lat},i})$ to each agent. The preferences are reflected in the reward function: At the start of the

Table 1: Components of the observation vector o

Feature	Symbol	Unit
Velocity	v	m/s
Distance to left and right boundary	d_l, d_r	m
Heading relative to lane in $\{0, 5, 10, 20\}$ m	$\varphi_{0..20}$	rad.
Road curvature in $\{0, 5, 10, 20\}$ m	$c_{0..20}$	m^{-1}
Preceding vehicle's velocity	v_{pre}	m/s
Distance to preceding vehicle	d_{pre}	m
Distance to next yield line	d_{yield}	m
Conflicting vehicle's velocity	$v_{\text{conf},1}$	m/s
Distance of conflicting vehicle to conflict zone	$d_{\text{conf},1}$	m
2 nd conflicting vehicle's velocity	$v_{\text{conf},2}$	m/s
Distance of 2 nd conflicting vehicle to conflict zone	$d_{\text{conf},2}$	m
Distance to next priority merge zone	d_{merge}	m
Non-priority vehicle's velocity	v_{nonpr}	m/s
Distance of non-priority vehicle to merge zone	d_{nonpr}	m
Preference: Minimum time gap	Δt_{\min}	s
Preference: Minimum safety distance	d_{\min}	m
Preference: Lateral acceleration weight	$\omega_{\text{acc,lat}}$	—

simulation, we assign a random minimum time gap $\Delta t_{\min,i} \in (0.5, 1.8)$ s, a minimum safety distance to the preceding vehicle $d_{\min,i} \in (1, 5)$ m and a lateral acceleration weight $\omega_{\text{acc,lat},i} \in (0.3, 1.5)$ to each agent. These values are also part of the observation vector and consequently can influence the actions selected by the policy.

To learn such a diverse policy, we formulate a MARL problem that we solve using the IPPO approach: The world state s contains the road layout, the kinematic states of all agents, and their planned route, e.g., entering through the first entry and leaving at the third exit.

The observation model $\mathcal{O}(\cdot|s, i)$ determines the observation o_i of the i -th agent. Similar to [3, 4, 7, 8], an observation o_i is a vector of manually selected descriptive features of the current local environment of the i -th agent. In our case, it includes 24 features that describe the agent's state and relation to the road, relevant other vehicles, and the preferences ρ_i . An extensive list of features is given in Tab. 1. Notably, the observation vector includes information on relevant other vehicles, also indicated in the observation step of Fig. 1: The preceding vehicle, the closest two vehicles with priority (Conflicting vehicles) when approaching the roundabout, and the closest non-priority vehicle driving towards the roundabout, when being in the roundabout. The relation to these vehicles is described by their velocities and distances. If a vehicle is not available, these values assume reasonable defaults, i.e., large distances and average velocities. Apart from this representation in the feature vector, communication between agents or policies is neither possible nor needed.

The policy selects an action by mapping the observation to a distribution over the action space. We restrict the feasible actions to physically plausible limits, i.e., steering angle $\delta \in (-\pi/8, \pi/8)$ rad and acceleration $a \in (-7, 3)$ m/s². The next vehicle state is determined according to the kinematic bicycle model [11].

Reward Function At each step, a reward is assigned to the i -th vehicle by the reward function

$$r(s, \alpha, i) = (\omega_{\text{vel}} r_{\text{vel}} + \omega_{\text{crash}} r_{\text{crash}} + \omega_{\text{offtr}} r_{\text{offtr}} + \omega_{\text{acc,lon}} r_{\text{acc,lon}} \\ + \omega_{\text{acc,lat},i} r_{\text{acc,lat}} + \omega_{\text{tg}} r_{\text{tg}} + \omega_{\text{dist}} r_{\text{dist}} + \omega_{\text{coop}} r_{\text{coop}})(s, \alpha, i).$$

It is a linear combination of different rewards. The importance of each reward can be altered via the corresponding weight ω . We set all weights to 1 for our experiments, except for $\omega_{\text{acc,lat},i}$, as described above.

The first three rewards,

$$\begin{aligned} r_{\text{vel}}(s, \alpha, i) &= 1 - |v_i - v_{\max}| / v_{\max} && \text{clipped to } [0, 1], v_{\max} = 9 \text{ m/s} \\ r_{\text{crash}}(s, \alpha, i) &= -100 - 20v_i / (1 \text{ m/s}) && \text{if } i \text{ crashed,} \\ r_{\text{offtr}}(s, \alpha, i) &= -200 && \text{if } i \text{ is off track,} \end{aligned}$$

are essential for the driving functionality: r_{vel} incentivizes progress along the track, whereas r_{crash} and r_{offtr} penalize collisions and leaving the track. By assigning negative rewards to longitudinal and lateral accelerations,

$$\begin{aligned} r_{\text{acc,lon}}(s, \alpha, i) &= -a_{\text{lon}}^2 / a_{\max}^2 && \text{clipped to } [-1, 0], a_{\max} = 5 \text{ m/s}^2 \\ r_{\text{acc,lat}}(s, \alpha, i) &= -a_{\text{lat}}^2 / a_{\max}^2 && \text{clipped to } [-1, 0], \end{aligned}$$

the model learns to use moderate accelerations if possible. The penalty on lateral accelerations effectively limits the velocity in the roundabout. The additional terms

$$\begin{aligned} r_{\text{tg}}(s, \alpha, i) &= -1 && \text{if } \Delta t_i < \Delta t_{\min,i}, \\ r_{\text{dist}}(s, \alpha, i) &= -10 && \text{if distance to preceding}(i) < d_{\min,i} \end{aligned}$$

facilitate learning a safe behavior because they directly penalize undesirable behavior, i.e., low time gaps and low distances. Thus, the RL agent does not need to experience a collision for learning to avoid these situations. The thresholds $d_{\min,i}$ and $\Delta t_{\min,i}$ differ per agent, fostering different driving styles.

Finally, the cooperation reward

$$r_{\text{coop}}(s, \alpha, i) = r(s, \alpha, j) \quad \text{if } i \text{ has to give way to } j$$

assigns the reward of the closest vehicle in the roundabout j to vehicle i . This reward is only assigned when i is close ($< 2.5 \text{ m}$) to entering the roundabout. Effectively, this encourages cooperative behavior: If vehicle j can drive unhindered, i also benefits from the rewards of j . On the other hand, if the entry of i forces j to brake strongly, i will also receive the resulting penalty.

Improving Robustness One major motivation for this work is to improve the robustness, i.e., collision and off-track rate, of the policies compared to our previous behavioral cloning approach [4]. While behavioral cloning does not allow for manual tweaking of the policy, e.g., through the penalties r_{tg} and r_{dist} in safety-critical situations, our previous RL-based approach [8] is a first step in this direction. Compared to [8], we now use the IPPO method, which is better suited for MARL than the previously used soft actor-critic (SAC),

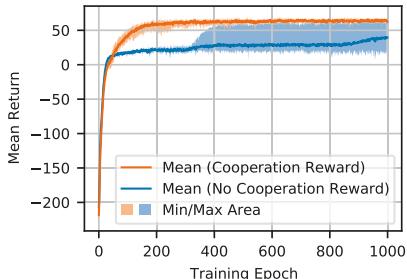


Figure 2: Minimum, mean, and maximum return g_0 at each training epoch, six repetitions. Higher values are better.

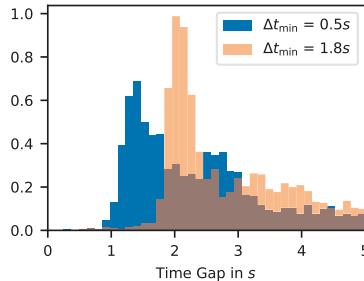


Figure 3: Histogram of time gaps of 2500 vehicles after 100 simulation steps. In these simulations, all vehicles use the same Δt_{\min} , either 0.5s or 1.8s.

because it exclusively uses experiences from the current policy. SAC and other *off-policy* methods learn from an experience buffer which also contains experiences from policies of earlier training epochs. This increases the *sample efficiency*, i.e., decreases the number of experiences required to converge to a good policy. However, MARL violates the underlying assumption of stationary environment dynamics as these change along with the policy. For example, it might be feasible to enter the roundabout with low speed at early training epochs, as all other agents also drive slowly. The same behavior might be dangerous at later epochs, as the other agents have learned to drive faster; the environment dynamics have changed.

We further improve robustness by enforcing a relatively large minimum Σ_α during training. The large variance of the action distribution causes all vehicles to act more randomly, such that the policy needs to be more robust to avoid critical situations, as the randomized actions could otherwise lead to collisions.

4 Experiments

The training is performed simultaneously in 50 randomly initialized roundabouts with 1 to 20 vehicles. On average, we collect experiences of 650 vehicles per simulation step. Each simulation terminates after 200 steps with a simulation step size of $dt = 0.2\text{s}$. The large simulation horizon of 40s is required for the agents to experience the long-term effects of their actions, e.g., that waiting at the roundabout entry eventually enables one to drive in.

The training progress is visualized in Fig. 2. Within 10 episodes, most agents learn to stay on track and subsequently further refine their behavior to avoid collisions while slowly increasing their velocity. The training typically converges after approximately 200 epochs, equivalent to 1 hour on an i7-9700 CPU @ 3 GHz. To ensure reproducibility, we repeat the training six times with randomly initialized neural network parameters and random initial simulation states.

We found that the cooperation reward r_{coop} is essential for learning a policy that respects

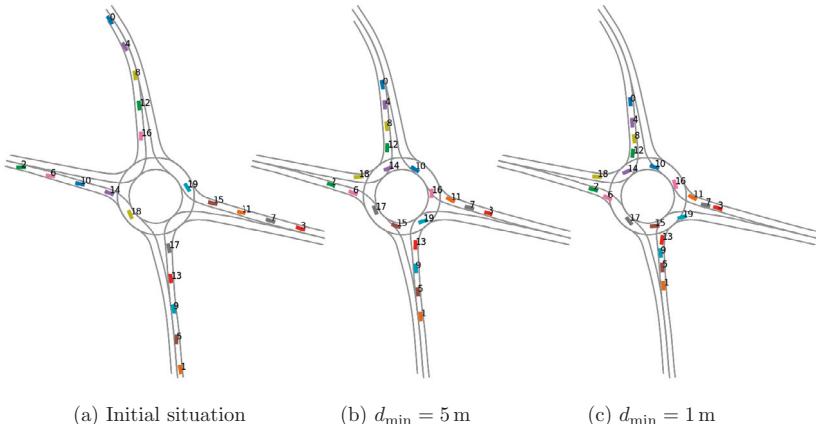


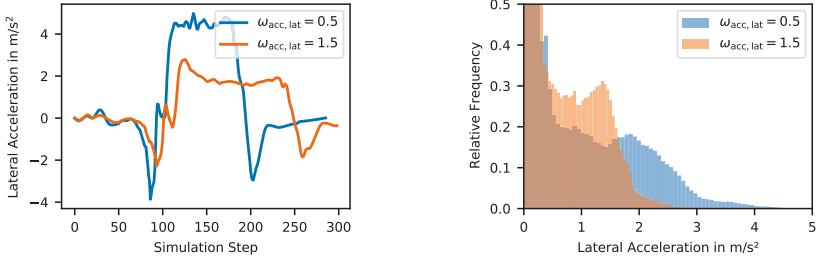
Figure 4: Effect of varying the minimum safety distance d_{\min} of all vehicles, while keeping all other parameters fixed: After 15 s of simulation, the standstill distance between the vehicles at the entry lanes is clearly distinguishable.

the right of way when entering the roundabout. Agents trained without cooperation reward often squeeze into the roundabout, regardless of the impact on other vehicles. Ultimately, this behavior often causes deadlocks, as more and more vehicles enter the roundabout. This effect is reflected in the mean returns during training in Fig. 2: Policies trained without cooperation reward require considerably more training epochs to achieve returns similar to those trained with cooperation reward. Without cooperation, three out of six policies never achieve good performance due to the above-described deadlock phenomenon. Effectively, the cooperation term adds a reflection of the common interest of all agents—flowing traffic—to the otherwise purely egoistic training goal, facilitating the convergence of the IPPO method. All subsequent experiments investigate the cooperative policies.

After training, the policy is executed deterministically by selecting the mean value of the action distribution instead of sampling. The preference values are randomly sampled for each vehicle. We evaluate the policy by executing it in 200 randomly generated situations for 200 steps with $dt = 0.1 \text{ s}$, containing a total of 2570 vehicles, which is equivalent to 14 hours of driving. Compared to training, the step size dt is halved to improve the policy performance by enabling faster reaction times. None of the six policies leaves the track in any case. The best two policies produce 0 collisions, while the worst policy produces 5 collisions (0.2%), significantly less than in our previous RL-based works (approx. 1%) [8] and our supervised learning based approach (approx. 3%) [4].

To demonstrate that our method learns an adaptive policy that can be changed at execution time via the preference vector ρ , we demonstrate the effect of altering the desired time gap Δt_{\min} , the minimum safety distance d_{\min} , and the lateral acceleration weight $\omega_{\text{acc,lat}}$ in the following. To emphasize the effects, we always assign the same ρ to all vehicles in a situation.

Keeping all other parameters fixed, we evaluate the effect of altering Δt_{\min} on the



(a) Lateral accelerations experienced by one vehicle driving through an empty round-about

(b) Histogram of lateral accelerations of 2500 vehicles during 200 time steps of simulation ($N \approx 500.000$)

Figure 5: Influence of lateral acceleration cost weight $\omega_{\text{acc}, \text{lat}}$ on the behavior of vehicles. Higher $\omega_{\text{acc}, \text{lat}}$ values penalize lateral accelerations, leading to lower overall accelerations as vehicles reduce their velocity in curves. This is shown for one single vehicle in 5a and for a large number of vehicles in 5b.

time gaps occurring after 100 simulation steps in Fig. 3. Almost no time gaps below the respective Δt_{\min} can be observed.

Next, we vary the minimum safety distance d_{\min} . The effect is shown in Fig. 4, where the policy is applied to the same initial situation for 150 steps. The effect of switching between $d_{\min} = 5 \text{ m}$ and $d_{\min} = 1 \text{ m}$ is clearly visible when comparing Fig. 4b and 4c.

Finally, the effects of different lateral acceleration costs are visualized in Fig. 5. A lower cost weight $\omega_{\text{acc}, \text{lat}}$ permits higher lateral accelerations, enabling the vehicles to drive with higher velocities in the roundabout, and vice versa.

5 Conclusion and Outlook

Our proposed approach learns a diverse policy that can represent different driving styles through its preferences ρ : Minimum time gap, minimum safety distance, and lateral acceleration penalty weight. At the same time, we maintain the main advantage of training a single policy: Efficient training by leveraging the experiences of all agents to train the same policy [15]. We demonstrate the emerging properties of the learned policies. Moreover, we introduce a cooperation term that enables the otherwise purely egoistic IPPO method to reliably converge in our roundabout setting.

Future work could address the online estimation of the preference vector, given observations of real-world driver behavior. Similar ideas have been applied in [2, 16] for manually formulated parametric models. Subsequent predictions can then adapt to individual driver behavior. Additionally, uncertainty can be expressed in the low-dimensional preference space. A set of representative trajectory predictions can then be generated by rolling out the policy with different preference values.

References

- [1] C. Hubmann et al. "Automated Driving in Uncertain Environments: Planning With Interaction and Uncertain Maneuver Prediction". In: *IEEE Transactions on Intelligent Vehicles*. Vol. 3. Mar. 2018, pp. 5–17.
- [2] H. Bey et al. "Handling Prediction Model Errors in Planning for Automated Driving Using POMDPs". In: *IEEE Intelligent Transportation Systems Conference*. Indianapolis, 2021.
- [3] J. Schulz et al. "Interaction-Aware Probabilistic Behavior Prediction in Urban Environments". In: *IEEE International Conference on Intelligent Robots and Systems*. Madrid, 2018.
- [4] M. Sackmann et al. "Multi-Step Training for Predicting Roundabout Traffic Situations". In: *IEEE Intelligent Transportation Systems Conf.* Indianapolis, 2021.
- [5] L. Bergamini et al. "SimNet: Learning Reactive Self-driving Simulations from Real-world Observations". In: *IEEE Int. Conf. on Robotics and Automation*. 2021.
- [6] S. Suo et al. "TrafficSim: Learning to Simulate Realistic Multi-Agent Behaviors". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
- [7] J. Schulz et al. "Learning Interaction-Aware Probabilistic Driver Behavior Models from Urban Scenarios". In: *IEEE Intelligent Vehicles Symposium*. Paris, 2019.
- [8] F. Konstantinidis et al. "Parameter Sharing Reinforcement Learning for Modeling Multi-Agent Driving Behavior in Roundabout Scenarios". In: *IEEE Intelligent Transportation Systems Conference*. Indianapolis, 2021.
- [9] C. S. de Witt et al. *Is Independent Learning All You Need in the StarCraft Multi-Agent Challenge?* pre-print, arXiv: 2011.09533v1. Nov. 2020.
- [10] R. S. Sutton and A. G. Barto. *Reinforcement learning: an introduction*. Second edition. Cambridge, Massachusetts: The MIT Press, 2018.
- [11] J. Kong et al. "Kinematic and dynamic vehicle models for autonomous driving control design". In: *IEEE Intelligent Vehicles Symposium*. Seoul, 2015.
- [12] P. Abbeel and A. Y. Ng. "Apprenticeship Learning via Inverse Reinforcement Learning". In: *International Conference on Machine Learning*. 2004.
- [13] J. Schulman et al. *Proximal Policy Optimization Algorithms*. arXiv: 1707.06347v2. Aug. 2017.
- [14] J. Schulman et al. *High-Dimensional Continuous Control Using Generalized Advantage Estimation*. arXiv:1506.02438v6. 2018.
- [15] J. K. Gupta, M. Egorov, and M. Kochenderfer. "Cooperative Multi-agent Control Using Deep Reinforcement Learning". In: *Autonomous Agents and Multiagent Systems*. Ed. by G. Sukthankar and J. A. Rodriguez-Aguilar. Cham: Springer, 2017, pp. 66–83.
- [16] S. Hoermann, D. Stumper, and K. Dietmayer. "Probabilistic long-term prediction for autonomous vehicles". In: *IEEE Intelligent Vehicles Symposium*. Los Angeles, 2017.

Kontextbasierte Bewegungsplanung automatisierter Fahrzeuge an Kreuzungen mit vernetzten Lichtsignalanlagen

Johannes Müller*, Jan Strohbeck*, Max Mertens* und Michael Buchholz†

Zusammenfassung: Die Vielzahl an Situationen, welche im urbanen Straßenverkehr vorkommen, stellt aktuelle Bewegungsplanungsverfahren in automatisierten Fahrzeugen immer noch vor große Herausforderungen. Während die Modellierung zusätzlicher Szenarien für klassische Verfahren oft sehr mühsam ist, erleichtert die kontextbasierte Bewegungsplanung diese Erweiterung. Um dies zu demonstrieren, wird in diesem Beitrag ein bestehendes kontextbasiertes Verfahren um das Überqueren einer Kreuzung mit vernetzter Lichtsignalanlage erweitert. Diese stellt auch prädiktive Informationen zur Verfügung, welche durch die Bewegungsplanung im Fahrzeug gewinnbringend verwendet werden. Das vorgestellte Verfahren wird simulativ und im realen Verkehr evaluiert. Gegenüber einer rein reaktiven Planung zeigen sich dabei deutliche Vorteile.

Schlüsselwörter: Bewegungsplanung, Lichtsignalanlage, vernetztes automatisiertes Fahren

1 Einleitung

Vernetzte automatisierte Fahrzeuge müssen im urbanen Raum eine Vielzahl verschiedener Szenarien beherrschen. Dabei sollen alle Spezifika des einzelnen Szenarios berücksichtigt werden, gleichzeitig muss der Planungsalgorithmus aber möglichst generell mit allen Szenarien umgehen können. Diese konkurrierenden Anforderungen stellen heutige Bewegungsplanungsverfahren immer noch vor große Herausforderungen. Klassische Ansätze teilen die Bewegungsplanung in Handlungs- und Trajektorienplanung auf [1]. Die Handlungsplanung muss also für alle antizipierten Szenarien immer eine adäquate Verhaltensentscheidung treffen, für welche die Trajektorienplanung dann eine gültige Trajektorie berechnen muss. In den dabei üblicherweise verwendeten Zustandsautomaten erfolgt die Modellierung neuer Szenarien indirekt: Für die neue Situation muss zunächst ein adäquates Verhalten modelliert werden. Zu diesem müssen Entscheidungskriterien definiert werden, welche einerseits die Situation beschreiben und andererseits die Auswahl der vorgeesehenen Verhaltensentscheidung ermöglichen. Dabei müssen die Kreuzabhängigkeiten der jeweiligen Kriterien bedacht werden. Entsprechend sind viele existierende Verfahren zum

*J. Müller, J. Strohbeck und M. Mertens sind Akademische Mitarbeiter am Institut für Mess-, Regel- und Mikrotechnik – Universität Ulm, Albert-Einstein-Allee 41, 89081 Ulm (E-Mail: {johannes-christian.mueller, jan.strohbeck, max.mertens}@uni-ulm.de).

†Michael Buchholz ist Akademischer Oberrat am Institut für Mess-, Regel- und Mikrotechnik – Universität Ulm, Albert-Einstein-Allee 41, 89081 Ulm (E-Mail: michael.buchholz@uni-ulm.de).

Lösen von Kreuzungsszenarien mit vernetzter Lichtsignalanlage, z.B. [2, 3], nicht einfach auf andere Szenarien wie das Einfädeln in eine Vorfahrt-Achten-Kreuzung übertragbar.

Moderne Verfahren, z.B. Methoden auf Basis von Markov-Entscheidungsprozessen [4], selbstbestärkendem Lernen [5, 6] oder Ende-zu-Ende-Lernverfahren benötigen zwar keine explizit modellierten Kriterien mehr, dafür aber einen großen Trainingsdatensatz, der die zu lösende Situation beschreibt [7].

Im Gegensatz dazu kann das kontextbasierte Bewegungsplanungsverfahren [8] einfach auf neue Szenarien erweitert werden. Die Kernidee dieses Verfahrens besteht darin, die einzelnen Situationen durch die jeweils geltenden Randbedingungen, Modellannahmen und Heuristiken zur Bestimmung möglicher Trajektorienkandidaten zu beschreiben. Diese Beschreibung wird *Situationskontext* genannt und vereinfacht die Modellierung erheblich. Die Handlungs- und Trajektorienplanung wird in einem Schritt mit Hilfe eines Sampling-Verfahrens gelöst, welches über alle verfügbaren Handlungsoptionen eines Situationskontexts iteriert und die dazu berechnete Trajektorie auswählt, die alle Randbedingungen erfüllt und insgesamt die geringsten Kosten aufweist. Durch die explizite Beschreibung von Randbedingungen und Modellannahmen kann das Planungsverfahren auf mengenbasierte Ansätze zur Behandlung von Unsicherheiten zurückgreifen, welche dem Verfahren Sicherheitsgarantien verleihen.

In diesem Beitrag wird ein Situationskontext für Kreuzungen mit vernetzten Lichtsignalanlagen modelliert. Damit wird das kontextbasierte Bewegungsplanungsverfahren [8], welches bisher an Vorfahrt-Achten-Kreuzungen mit Infrastrukturunterstützung zur Auflösung von Verdeckungen evaluiert wurde, für die Anwendung an Kreuzungen mit vernetzten Lichtsignalanlagen erweitert. Es dient der Umsetzung des Anwendungsfalls „Überqueren einer Kreuzung mit vernetzter Infrastruktur“ aus dem EU-Projekt ICT4CART (www.ict4cart.eu). Vernetzte Lichtsignalanlagen können dem vernetzten Fahrzeug hierbei nicht nur den aktuellen Signalstatus pro Spur übertragen, sondern auch eine Schätzung für den Umschaltzeitpunkt in den nächsten Signalzustand. Dazu kommt die standardisierte Nachricht SPATEM (Signal Phase and Timing Extended Message) des Europäische Instituts für Telekommunikationsnormen (ETSI) zum Einsatz. Die vorgestellte Modellierung berücksichtigt dabei sowohl Unsicherheiten in dieser Schätzung als auch Unsicherheiten in den prädizierten Trajektorien weiterer Verkehrsteilnehmer. Das erweiterte Verfahren wird simulativ evaluiert und experimentell an einer Kreuzung mit vernetzter Lichtsignalanlage verifiziert. Dabei zeigt das erweiterte Verfahren an der untersuchten Kreuzung gegenüber einer rein reaktiven Planung deutliche Vorteile.

2 Situationskontext für Kreuzungen mit vernetzten Lichtsignalanlagen

Der Situationskontext für Vorfahrt-Achten-Kreuzungen mit vernetzter Infrastruktur zur Auflösung von Verdeckungen wird in [8] näher beschrieben. Die Modellierung des Situationskontexts für Kreuzungen mit vernetzten Lichtsignalanlagen bleibt strukturell nahezu gleich, lediglich das Risikomodell muss neu erstellt werden. Daher wird dieses im Abschnitt 2.2 genauer vorgestellt. Zuvor wird im Abschnitt 2.1 der sich insgesamt ergebende Situationskontext für die Kreuzung mit Lichtsignalanlage beschrieben.

2.1 Resultierender Situationskontext

Gemäß [8] wird der Situationskontext allgemein durch

$$\mathcal{C}_{[s_j, s_{j+1}]} = (\mathcal{S}_C, \mathcal{S}_{RA}, \mathcal{L}_B) \quad (1)$$

mathematisch beschrieben und enthält vier Teile: ein Gültigkeitsintervall $[s_j, s_{j+1}]$, welches durch die 1D-Positionen s_j und s_{j+1} entlang der geplanten Route charakterisiert ist, eine Menge von Randbedingungen \mathcal{S}_C , eine Menge von regularisierenden Annahmen \mathcal{S}_{RA} und einer Liste von Verhaltensoptionen \mathcal{L}_B . Randbedingungen sind beispielsweise die Verkehrsregeln oder die Straßengeometrie, welche die sichere Kurvengeschwindigkeit begrenzt. Diese müssen von der Planung stets eingehalten werden. Regularisierende Annahmen, wie z.B. die Einhaltung der Verkehrsregeln durch andere Verkehrsteilnehmer, sind dagegen deutlich schwächere Modellannahmen, die in manchen Situationen verletzt werden können. Weitere Beispiele für regularisierende Annahmen sind die Verteilungsannahmen im Risikomodell oder Heuristiken, mit denen die Liste der Verhaltensoptionen bestimmt wird. Die Liste der Verhaltensoptionen enthält alle in der aktuellen Situation sinnvollen Möglichkeiten, wie sich das Fahrzeug verhalten kann. Diese sind gemäß

$$\mathcal{B} : (\mathcal{P}, \langle (\mathbf{x}_f, t_f) \rangle_i, I) \quad (2)$$

durch drei Komponenten charakterisiert: den Referenzpfad \mathcal{P} , die Zielsequenz $\langle (\mathbf{x}_f, t_f) \rangle_i$, entlang der die Längsplanung die Trajektorie generiert, und die Priorität I . Die Priorität I wird anhand der Heuristiken in \mathcal{S}_{RA} festgelegt und bestimmt, wie stark präferiert die zu der Handlungsoption korrespondierende Handlungsentscheidung ist.

Analog zum Situationskontext für Vorfahrt-Achten-Kreuzungen enthält \mathcal{S}_C auch für den Kontext „Kreuzung mit vernetzter Lichtsignalanlage“ ein Tempolimitprofil $v_{\max}(s, t)$. Dieses passt sich an vorausfahrende Fahrzeuge an und stellt dadurch sicher, dass das Ego-Fahrzeug in einer Folgefahrt nicht näher als den Sicherheitsabstand auf das vorausfahrende Fahrzeug auffährt. Weiterhin sind auch die minimale und maximale Beschleunigung sowie das maximal zulässige Restrisiko in beiden Kontexten in \mathcal{S}_C enthalten. Auf Grund des Überholverbots im Kreuzungsbereich sind die Verhaltensoptionen auf einen Referenzpfad entlang der Fahrspur begrenzt. Für beide Szenarien enthält der Kontext die regularisierende Annahme, dass sich die anderen Verkehrsteilnehmer an die Verkehrsregeln halten. Außerdem haben beide Kontexte das Anhalten mit konstanter Verzögerung an der Haltelinie bzw. hinter einem bremsenden Vorausfahrer als Fail-Safe-Strategie in \mathcal{S}_{RA} hinterlegt. Für komfortables Anhalten werden jeweils Zielsequenzen mit variierender Endzeit t_f , aber mit identischem Zielzustand $\mathbf{x}_f = [s_{stop}, v = 0, a = 0]^T$ generiert, wobei v die Geschwindigkeit und a die Beschleunigung bezeichnen.

Unterschiedlich ist jedoch das Risikomodell, das im folgenden Abschnitt 2.2 beschrieben wird. Zudem werden anstatt einer Liste von durch Infrastruktursensorik detektierten Objekten die Zustände einer Lichtsignalanlage und deren Prädiktion übertragen.

Schließlich beinhaltet \mathcal{L}_B die Liste der Verhaltensoptionen, welche durch die entsprechenden Heuristiken generiert wurden. Wie für die Vorfahrt-Achten-Kreuzung wird auch hier das Überqueren der Kreuzung höher priorisiert als das Anhalten an der Kreuzung. Dadurch überquert das Fahrzeug die Kreuzung, wann immer es dafür eine gültige Trajektorie gibt.

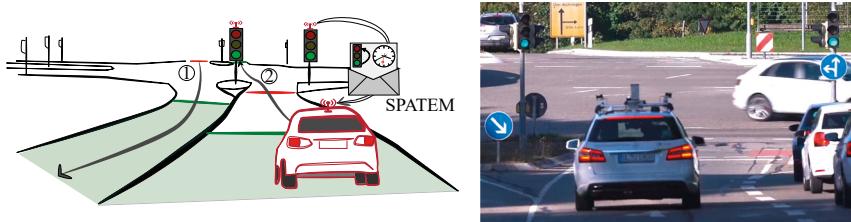


Abbildung 1: Die Skizze links zeigt eine Kreuzung mit vernetzter Lichtsignalanlage, die Haltelinie (rot) sowie PNR und PGA (grün) für ein Überqueren der Kreuzung. Im Foto rechts ist der automatisierte Versuchsträger zu sehen, der von der vorgestellten Bewegungsplanung gesteuert wird. Das Fahrzeug bremst bereits komfortabel während der grünen Ampelphase, da durch die von der Lichtsignalanlage bereitgestellten Informationen klar ist, dass ein Überqueren nicht mehr möglich sein wird.

2.2 Risikomodell

Mit dem Risikomodell wird die Wahrscheinlichkeit berechnet, dass das Ego-Fahrzeug in einen Zustand kommt, für den die Bewegungsplanung nicht mehr garantiert eine gültige Lösung findet [8]. Die Eingangsdaten des Risikomodells bestehen primär aus einer Objektliste, welche der Bewegungsplanung von der Fahrzeugperzeption zur Verfügung gestellt wird, dem Zustand des Ego-Fahrzeugs, welcher von der Lokalisierung bereitgestellt wird, und Zustandsprädiktionen für die Lichtsignalanlage, welche per Vehicle-to-anything-Kommunikation (V2X-Kommunikation) übermittelt werden.

Auf Basis dieser Eingangsdaten wird zunächst mit Hilfe von mengenbasierten Verfahren nach [9] eine invariabel sichere Zustandsmenge (ISZ) bestimmt, welche sich dadurch auszeichnet, dass jeder enthaltene Zustand garantiert Startpunkt einer gültigen Trajektorie ist. Im Idealfall lässt sich eine Trajektorie finden, welche vollständig in der ISZ liegt, sodass deren Sicherheit formal garantiert werden kann [9]. Im Allgemeinen wird jedoch nur per Randbedingung erzwungen, dass das Ende der Trajektorie in der ISZ liegt. Entsprechend kann es vorkommen, dass Trajektorienkandidaten zeitweise außerhalb der ISZ liegen und später (wieder) in diese eintreten. Der Austrittspunkt aus einer ISZ heißt dann Point of No Return (PNR) und der Wiedereintrittspunkt Point of Guaranteed Arrival (PGA) [9], siehe Abbildung 1. Beide Punkte können mit mengenbasierten Verfahren bestimmt werden. Für ein Kreuzungsszenario ist der PNR anschaulich der Punkt, an dem das Ego-Fahrzeug gerade noch vor der Haltelinie der Kreuzung bzw. hinter einem vorausfahrenden Fahrzeug anhalten kann. Der PGA hingegen markiert den Punkt, ab dem die Kreuzungssituation sicher erfolgreich gelöst wurde, beispielsweise durch Einfädeln in eine Lücke auf der Zielpur oder durch erfolgreiches Überqueren der Kreuzung.

Das Restrisiko einer Trajektorie setzt sich nun aus zwei Teilen zusammen: Einerseits besteht das Risiko, dass mindestens eine der zur Konstruktion der ISZ verwendeten regularisierenden Annahmen in der aktuellen Situation unzutreffend ist. Andererseits besteht ein Risiko, dass das Fahrzeug zwischen PNR und PGA in einen Zustand gerät, für den die Sicherheit nicht mehr gewährleistet werden kann. Im Allgemeinen lässt sich das Risi-

komodell also durch die Gleichung

$$p_{\text{risk}}(\mathcal{T}_n) = (1 - p_{\text{RA}}) + p_{\text{RA}} \int_{t_{\text{PNR}}}^{t_{\text{PGA}}} \int_{s_n(t) - s^-(t)}^{s_n(t) + s^+(t)} f(s, t) ds dt \quad (3)$$

beschreiben, wobei p_{RA} die Wahrscheinlichkeit angibt, dass die regularisierenden Annahmen korrekt sind. Weiterhin sind s^- und s^+ die Sicherheitsabstände zu anderen Verkehrsteilnehmern, t_{PNR} und t_{PGA} die Zeiten, an denen das Ego-Fahrzeug den PNR bzw. den PGA erreicht, und $s_n(t)$ die 1D-Position auf der Trajektorie. Die Funktion $f(s, t)$ gibt die Wahrscheinlichkeit an, dass das Ego-Fahrzeug einen bestimmten 1D-Ort entlang des Referenzpfads zur Zeit t nicht überfahren darf. Für den konkreten Situationskontext gibt es dafür zwei Gründe: Entweder sperrt die Lichtsignalanlage zu einem Zeitpunkt t die Kreuzung, d.h. alle Orte zwischen Haltelinie und PGA, oder ein Ort s wird voraussichtlich zum Zeitpunkt t von einem Verkehrsteilnehmer mit Vorrang belegt. Um die Wahrscheinlichkeit einer Belegung durch andere Verkehrsteilnehmer zu bestimmen, werden für jedes Objekt der Objektliste mehrere mögliche zukünftige Trajektorien mit dem neuronalen Netz aus [10] prädiziert. Aus den Prädiktionen werden die Trajektorien aussortiert, die entweder die eigenen Trajektorienkandidaten sicher nicht schneiden, oder Verkehrsregeln verletzen und damit unplausibel sind. Dies ergibt die Liste relevanter prädizierter Trajektorien \mathcal{L}_{RPT} . Diese werden auf den Trajektorienkandidat projiziert, mit Hilfe eines Kalman-Filters zeitlich interpoliert und das Risikointegral über den Trajektorienkandidaten integriert. Damit berechnet sich das Risiko durch andere Verkehrsteilnehmer zu

$$p_{\text{RPT}} = \sum_{s_T \in \mathcal{L}_{\text{RPT}}} \int_{t_{\text{PNR}}}^{t_{\text{PGA}}} \Phi\left(\frac{s_T(t) - s^+(t) - s_n(t)}{\sigma_T}\right) - \Phi\left(\frac{s_T(t) + s^-(t) - s_n(t)}{\sigma_T}\right) dt, \quad (4)$$

wobei $\Phi(\cdot)$ die Stammfunktion der aus der Kalman-Filterung der projizierten Trajektorien s_T stammenden Normalverteilung mit Varianz σ_T^2 bezeichnet. In der praktischen Implementierung wurde das Integral durch die in [8] gegebene Iteration approximiert, um Rechenzeit zu sparen.

Die Wahrscheinlichkeit p_{TL} , dass die Lichtsignalanlage die Durchfahrt sperrt, bevor das Ego-Fahrzeug in die Kreuzung eingefahren ist, wird in Abhängigkeit des aktuellen Zustands sowie t_{\min} , t_{erw} und t_{\max} , also dem frühesten, dem erwarteten und dem spätesten Umschaltzeitpunkt, modelliert. Diese Umschaltzeitpunkte sowie die Konfidenz für t_{erw} werden von der Lichtsignalanlage per SPATEM übertragen, wobei aus der Konfidenz die Varianz σ_{TL}^2 berechnet wird. Dazu wird der Umschaltzeitpunkt als normalverteilt mit Mittelwert t_{erw} und Varianz σ_{TL}^2 angenommen, die Verteilung aber bei t_{\min} und t_{\max} hart zu Null bzw. Eins abgeschnitten. Dies führt auf

$$p_{\text{TL}} = \begin{cases} 1, & \text{falls grün und } t_{\text{stop}} \geq t_{\max} \text{ oder rot und } t_{\text{stop}} \leq t_{\min}, \\ 0, & \text{falls grün und } t_{\text{stop}} \leq t_{\min} \text{ oder rot und } t_{\text{stop}} \geq t_{\max}, \\ \Phi\left(\frac{t_{\text{stop}} - t_{\text{erw}}}{\sigma_{\text{TL}}}\right), & \text{falls grün und } t_{\min} < t_{\text{stop}} < t_{\max}, \\ 1 - \Phi\left(\frac{t_{\text{stop}} - t_{\text{erw}}}{\sigma_{\text{TL}}}\right), & \text{falls rot und } t_{\min} < t_{\text{stop}} < t_{\max}, \end{cases} \quad (5)$$

wobei t_{stop} die Zeit angibt, zu dem das Fahrzeug die Haltelinie erreicht. Für die praktischen Tests wurde für ein konservativeres Verhalten $t_{\text{stop}} = t_{\text{PGA}}$ gesetzt. Um zusätzlich die Gelbphasen mit zu berücksichtigen, werden t_{\min} , t_{erw} und t_{\max} entsprechend korrigiert.

Falls die Lichtsignalanlage keine Schätzung für den Umschaltzeitpunkt überträgt (oder der Ampelzustand nur aus einem Kamerabild extrahiert wurde), wird als Risiko $p_{TL} = 0$ bei grünem Ampelzustand und sonst $p_{TL} = 1$ angenommen. Dies entspricht der rein reaktiven Planung. Die Berechnung von (3) erfolgt für eine Kreuzung mit Lichtsignalanlage insgesamt also durch

$$p_{\text{risk}} = (1 - p_{RA}) + p_{RA}(p_{TL} + (1 - p_{TL})p_{RPT}). \quad (6)$$

3 Evaluation

Die Evaluation des vorgestellten Verfahrens erfolgt in zwei Schritten: Zunächst demonstrieren wir anhand von Simulationen den Effizienzgewinn, der sich durch Ausnutzen der von vernetzten Lichtsignalanlagen übermittelten Informationen ergibt. Dann validieren wir anhand von Tests an einer Pilotanlage das Verhalten des Verfahrens im realen Straßenverkehr.

3.1 Simulationsbeispiele zum Nutzen der Zustandsprädiktion

Die Simulationen wurden mit Hilfe unseres Simulationsframeworks DeepSIL [11] durchgeführt. Dieses wurde dafür um ein Modell für Lichtsignalanlagen erweitert. Jedes Simulationsszenario wird hierfür zweimal durchlaufen. Das eine Mal wird dem Planungsverfahren der prädizierte t_{erw} , sowie der früheste t_{\min} und späteste t_{\max} Umschaltzeitpunkt bereitgestellt. Das andere Mal hingegen kennt das Fahrzeug nur den aktuellen Zustand der Lichtsignalanlage und kann daher sein Verhalten nur reaktiv anpassen. Beide Male schaltet die Lichtsignalanlage tatsächlich zur Zeit $t = 0$ s. Die Simulationsszenarien sind dabei so gewählt, dass das Fahrzeug kurz vor dem Umschalten an der Lichtsignalanlage ankommt. Abbildung 2 zeigt die Trajektorien des Ego-Fahrzeugs mit und ohne Prädiktion beim Umschalten der Lichtsignalanlage von grün auf rot (links) und von rot nach grün (rechts) im Vergleich. Es zeigt sich, dass das Ego-Fahrzeug die Prädiktionsinformation beim Umschalten von grün nach rot effektiv dafür nutzen kann, früher und damit sanfter zu bremsen. Beim Umschalten von rot nach grün adaptiert das Ego-Fahrzeug seine Geschwindigkeit frühzeitig, sodass es weniger bremsen und früh wieder moderat beschleunigen kann. Dadurch wird das Überqueren der Kreuzung energie- und zeiteffizienter als das rein reaktive Fahren.

3.2 Fahrversuche im realen Straßenverkehr

Im nächsten Schritt wird im realen Straßenverkehr validiert, dass das vorgestellte Verfahren angemessenes Verhalten für das Ego-Fahrzeug generiert. Dazu wird die Kreuzung bei der Halle Nord am Berliner Ring in Ulm betrachtet. Die Lichtsignalanlage wurde von Partnern im Projekt ICT4CART so umgerüstet, dass sie ihren frühesten, erwarteten und spätesten Umschaltzeitpunkt per hybrider Kommunikation [12] sowohl über ETSI ITS-G5 als auch über LTE/5G an das Ego-Fahrzeug überträgt. Abbildung 3 zeigt 35 Trajektorien von Versuchsfahrten an der Pilotkreuzung aus zwei verschiedenen Richtungen, wobei die Lichtsignalanlage bei Ankunft an der Kreuzung in 15 Szenarien grün und in 20 Szenarien rot zeigte. Aufgetragen ist die Position s über der Zeit t und die Geschwindigkeit v über

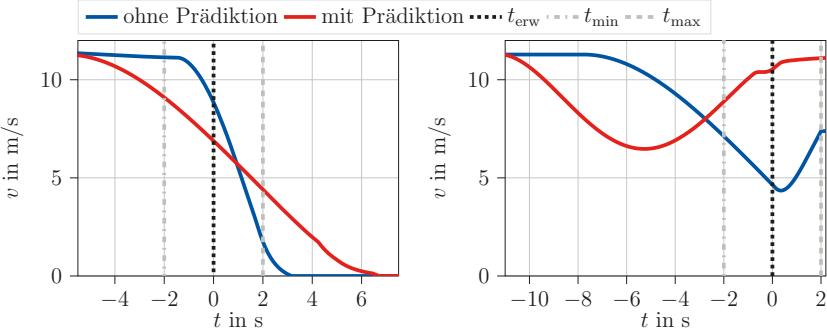


Abbildung 2: Simulierte Geschwindigkeitsverläufe $v(t)$ beim Umschalten der Lichtsignalanlage von grün über gelb nach rot (links) und von rot über rot-gelb nach grün (rechts).

der Position s jeweils bis zum Stopp des Fahrzeugs (bei rot) oder bis zum Erreichen des PGA (bei grün). Es zeigt sich, dass das Fahrzeug vorschriftsmäßig bei rot an der Kreuzung anhält (in rot gezeichnete Kurven) und bei grün über die Kreuzung fährt (in grün gezeichnete Kurven). Außerdem ist klar erkennbar, dass das Fahrzeug auch auf vorausfahrende Fahrzeuge reagiert und deshalb in den in magenta markierten Kurven deutlich vor der Stop-Linie (Position $s = 0$) anhält. Die in blau dargestellte Trajektorie korrespondiert zu einem Szenario, in dem die Lichtsignalanlage während der Anfahrt auf grün umschaltet. In dem Szenario fährt zudem ein weiteres Fahrzeug vor dem Ego-Fahrzeug, auf welches das Ego-Fahrzeug Rücksicht nehmen muss. Die entsprechende Verhaltensadaption ist im Geschwindigkeitsverlauf des Ego-Fahrzeugs deutlich zu erkennen. Außerdem ist zu erkennen, dass das Ego-Fahrzeug auf Basis der Prädiktion schon frühzeitig abbremsst, um noch mit Fahrt an der Kreuzung anzukommen, wenn die Lichtsignalanlage auf grün umschaltet. So kann das Fahrzeug ohne anzuhalten über die Kreuzung fahren und erreicht, wie in der Simulation dargestellt, eine höhere Zeiteffizienz.

4 Zusammenfassung und Ausblick

In dieser Arbeit wurde ein bestehendes kontextbasiertes Planungsverfahren auf einen neuen Use-Case, das Überqueren von Kreuzungen mit vernetzten Lichtsignalanlagen, erweitert. Es zeigte sich, dass dies aufgrund des kontextbasierten Verfahrens nur einer Modellerweiterung bedarf und dadurch deutlich weniger mühsam ist als bei verhaltensbasierten Verfahren. In Simulationen wurde gezeigt, dass das Verfahren deutliche Effizienzvorteile bringt, wenn die prädictierten Umschaltzeitpunkte der Lichtsignalanlage zur Verfügung stehen. Schließlich wurde im Straßenverkehr evaluiert, dass die mit dem Verfahren generierten Trajektorien auch unter realen Einsatzbedingungen ein angemessenes Verhalten des Ego-Fahrzeugs widerspiegeln.

Zukünftig bietet es sich an, das kontextbasierte Planungsverfahren auf weitere Use-Cases, wie zum Beispiel Kreisverkehre, Fahrbahnverengungen oder Baustellen, zu erweitern. Außerdem könnte das vorgestellte Verfahren von Heuristiken profitieren, die z.B.

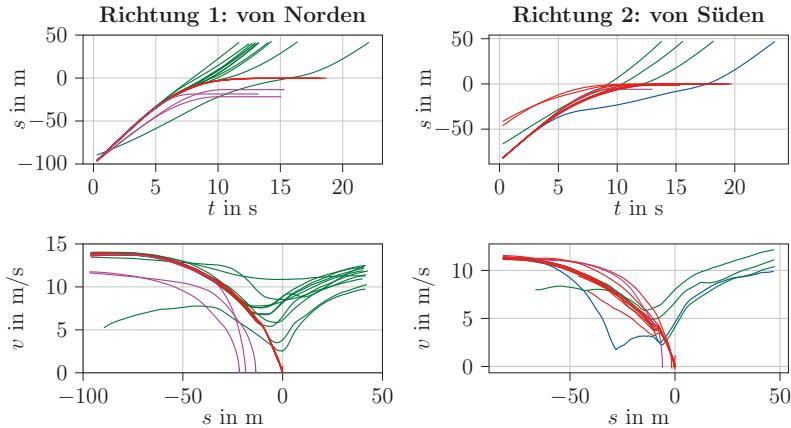


Abbildung 3: Experimentelle Verifikation im realen Straßenverkehr. Die Kreuzung mit vernetzter Lichtsignalanlage wird aus zwei verschiedenen Richtungen her angefahren. Zu sehen sind 16 Anfahrten bei rot ohne Fahrzeug voraus (rot), 4 Anfahrten bei rot mit Fahrzeug voraus (magenta), 14 Anfahrten bei grün und eine Anfahrt (blau), bei der die Lichtsignalanlage auf grün umschaltet. Die Trajektorien enden beim Stopp bzw. beim Erreichen des PGA.

mit Reinforcement-Learning-Verfahren gelernt wurden.

Danksagung

Teile dieser Arbeit wurden im Rahmen des Projekts ICT4CART durchgeführt, welches durch das Forschungs- und Innovationsprogramm Horizon 2020 der Europäischen Union unter der Fördernummer 768953 finanziert wird. Weitere Teile dieser Arbeit wurden finanziell vom Bundesministerium für Wirtschaft und Klimaschutz im Rahmen des Programms „Hoch- und vollautomatisiertes Fahren in anspruchsvollen Fahrsituationen“ (Projekt LUKAS, FKZ 19A16010I) unterstützt.

Literatur

- [1] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, “A survey of motion planning and control techniques for self-driving urban vehicles,” *IEEE Trans. Intell. Veh.*, vol. 1, no. 1, pp. 33–55, Mar. 2016.
- [2] C. Sun, J. Guanetti, F. Borrelli, and S. J. Moura, “Optimal eco-driving control of connected and autonomous vehicles through signalized intersections,” *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3759–3773, 2020.
- [3] G. Mahler and A. Vahidi, “An optimal velocity-planning scheme for vehicle energy efficiency through probabilistic prediction of traffic-signal timing,” *Trans. Intell. Transp. Sys.*, vol. 15, no. 6, pp. 2516–2523, 2014.

- [4] C. Hubmann, J. Schulz, G. Xu, D. Althoff, and C. Stiller, “A belief state planner for interactive merge maneuvers in congested traffic,” in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Maui, HI, USA, Nov. 2018, pp. 1617–1624.
- [5] F. Gritschneider, P. Hatzelmann, M. Thom, F. Kunz, and K. Dietmayer, “Adaptive learning based on guided exploration for decision making at roundabouts,” in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Gothenburg, Sweden, Nov. 2016, pp. 433–440.
- [6] M. Meghjani, Y. Luo, Q. H. Ho, P. Cai, S. Verma, D. Rus, and D. Hsu, “Context and intention aware planning for urban driving,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Macau, China, Nov. 2019, pp. 2891–2898.
- [7] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, “A survey of autonomous driving: Common practices and emerging technologies,” *IEEE Access*, vol. 8, pp. 58 443–58 469, Mar. 2020.
- [8] J. Müller, M. Herrmann, J. Strohbeck, and M. Buchholz, “Motion planning for connected automated vehicles at occluded intersections with infrastructure sensors,” accepted for publication in *IEEE Trans. Intell. Transp. Syst.*, <https://arxiv.org/pdf/2110.11246.pdf>, Nov. 2021.
- [9] C. Pek, M. Koschi, M. Werling, and M. Althoff, “Enhancing motion safety by identifying safety-critical passageways,” in *Proc. IEEE Annu. Conf. Decis. Control (CDC)*, Melbourne, VIC, Australia, Dec. 2017, pp. 320–326.
- [10] J. Strohbeck, V. Belagiannis, J. Müller, M. Schreiber, M. Herrmann, and M. Buchholz, “Multiple trajectory prediction with deep temporal and spatial convolutional neural networks,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Virtual, Nov. 2020.
- [11] J. Strohbeck, J. Müller, A. Holzbock, and M. Buchholz, “Deepsil: A software-in-the-loop framework for evaluating motion planning schemes using multiple trajectory prediction networks,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Prague, Czech Republic, Sep. 2021, pp. 7075–7081.
- [12] M. Buchholz, J. Strohbeck, A.-M. Adaktylos, F. Vogl, G. Allmer, S. C. Barros, Y. Lassoued, M. Wimmer, B. Häfty, G. Massot *et al.*, “Enabling automated driving by ict infrastructure: A reference architecture,” 8th Transp. Res. Arena TRA 2020 (conf. cancelled), 2020.

Trajectories as Markov-States for Long Term Traffic Scene Prediction

Jörg Reichardt

Abstract: We propose the use of traffic participant trajectories with constant time horizon as Markov states for tracking and trajectory prediction. We show how constant time length trajectories can be tracked with minimal computational overhead over kinematic state tracking. The same representation can be used to model future trajectories. In conjunction with multi-object multi-hypotheses tracking architectures it allows for an efficient representation of multimodal distributions over the future of traffic scenes with heterogeneous participants.

Keywords: Bayes filter, tracking, trajectory prediction, situation interpretation

1 Introduction

Safe and comfortable navigation of an autonomous vehicle necessitates anticipatory planning, i.e. the ability to form expectations and make predictions about the future behavior of dynamic agents in the environment.

The basis for such predictions is an accurate estimate of the present state of dynamic agents based on past observations. Naturally, such state estimates are probabilistic due to uncertainties in the measurement process or unobservable quantities such as agent intent.

State space models are uniquely suited for this task as they provide a solid probabilistic framework to sequentially absorb observations into estimates of the agents' current state and track their motion over time.

The standard technique for this are Bayesian Filter architectures of which the Kalman Filter is the prime example [1]. Under the assumption of Gaussian densities for states and observations together with linear motion and observation models, it yields closed form prediction and update equations that are both numerically efficient and stable. Commonly, kinematic states of agents are tracked. Figure 1 illustrates that these are generally not sufficient to predict the future evolution of a traffic scenario. Many contributions exist that augment the kinematic state space with latent variables modeling agent intent that need to be inferred from observations [2, 3]. Intent is not directly measurable and often not signaled unequivocally and thus has to be inferred from past observations. However, that is a challenging problem as it is not clear how to describe the space of driver intentions. Should one use maneuver intentions that are mutually exclusive and collectively exhaustive? Does that mean one cannot make a lane change while taking a turn? Also, does this apply to all traffic participants? What are the maneuvers of a cyclist or a pedestrian?

*Continental AG, Siemensstr. 12, 93055 Regensburg (e-mail: joerg.reichardt@continental-corporation.com)

Since all the obtainable information about a driver's intent must be in its past behavior, i.e. its past trajectory, it is tempting to simply maintain a list of past kinematic states and infer driver intent on the fly based on this "historic" data. Note that a list of past states is not the same as a list of past observations - state tracking algorithms disentangle the uncertain data associations that arise in multi-object tracking giving unique physical objects unique tracking ids without which single object trajectories cannot be formed.

While promising, using such lists has the obvious drawback of being both memory and compute intensive. This is particularly the case when tracking multiple objects with multi-hypotheses tracking algorithms that potentially maintain hundreds of data association hypotheses in order to be able to track effectively through clutter and occlusion [4, 5, 6]. Since each data association hypothesis effectively represents a different history, one would have to maintain hundreds of lists of past kinematic states.

We seize upon the idea of taking the trajectory of an object as the basis for predicting its future motion. However, instead of maintaining a list of kinematic states, we promote an object's past trajectory over a constant time horizon Δt to its *Markov state*. We assume $\Delta t \gg \delta t$, i.e. much larger than the cycle time δt at which new measurements are obtained and the filter updates are executed. The compatibility of tracked past and hypothesised future trajectory in the light of current evidence then plays the decisive role in estimating the probability of future trajectories.

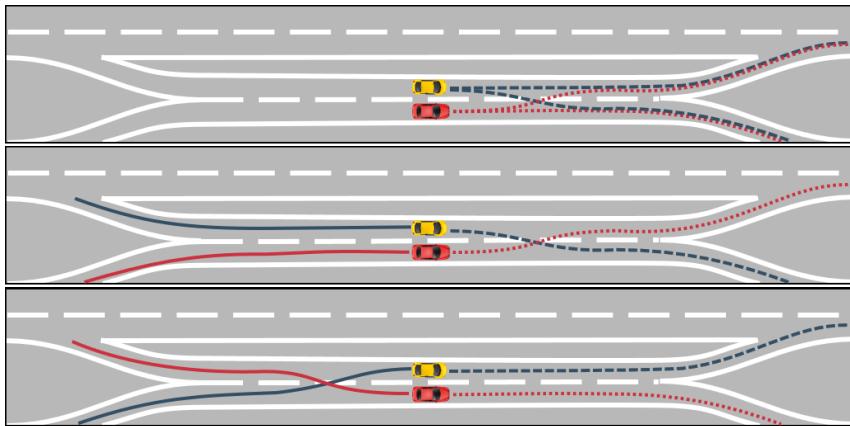


Figure 1: **Top:** Given only kinematic states, each vehicle has 2 plausible path options, resulting in a total of 4 equally likely future scenarios for the traffic scene. **Middle and Bottom:** With past trajectories given, the uncertainty about future scenarios may be largely resolved.

Our contribution is outlined as follows: we introduce a parametric trajectory representation and corresponding linear motion and observation models for its parameters, which, together with Gaussian densities over parameters, yield closed form Kalman update equations. We show that the parameters of this representation have a spatial interpretation. We then employ this representation in predicting future trajectories and finally demonstrate how this representation can be integrated into a multi-hypotheses tracking frame-

work that permits the estimation of multi-modal mixture distributions over the futures of entire traffic scenarios.

It is understood that all results presented are elementary and as such are not new. However, we feel that the exposition unifies a number of existing ideas and possibly provides a fresh perspective and novel synthesis that may still be of interest.

2 Trajectory Tracking

In contrast to paths, which are curves in space, trajectories are curves in space *and* time. The proposed parameterization rests on separating these into a set of *fixed* basis functions of time only, and a parameterization that has spatial semantics.

We choose a set of $n + 1$ fixed basis functions $\phi_k(\tau) : \mathbb{R} \rightarrow \mathbb{R}$ to describe a trajectory in each of d dimensions of space. We differentiate in $\tau \in \mathbb{R}$ a re-scaled time variable that denotes time along a tracked past trajectory with $\tau = 0$ corresponding to $t - \Delta t$ and $\tau = 1$ corresponding to t . It is convenient to introduce $\Phi(\tau) \in \mathbb{R}^{n+1}$ as the vector of basis functions $\Phi(\tau) = [\phi_0(\tau), \phi_1(\tau), \phi_2(\tau), \dots, \phi_n(\tau)]$ evaluated at τ .

Next, we introduce our state vector $\mathbf{x}_t \in \mathbb{R}^{(n+1)d}$. We can generate any point $\mathbf{c}_t(\tau) \in \mathbb{R}^d$ on a trajectory from linear combinations of the basis functions $\mathbf{c}_t(\tau) = (\Phi^T(\tau) \otimes \mathbf{I}_d)\mathbf{x}_t$ and thus identify our state variables as the coefficients of this linear combination. Here \mathbf{I}_d is a $d \times d$ identity matrix and \otimes denotes the Kronecker product that simply distributes our basis functions across the d spatial dimensions. An agent's current position at time t will be $\mathbf{c}_t(1)$ and an agent's position at time $t - \Delta t$ will be $\mathbf{c}_t(0)$.

It is instructive to interpret the entries in our state vector \mathbf{x}_t as *control points*, i.e. as having spatial semantics. Separated into individual coordinates and basis vectors \mathbf{e}_i of the d dimensional space we have (dropping the index t for brevity):

$$\mathbf{c}(\tau) = \sum_{i=1}^d \sum_{k=0}^n \phi_k(\tau) x_{kd+i} \mathbf{e}_i = \sum_{k=0}^n \phi_k(\tau) \mathbf{p}_k$$

Thus, a trajectory can also be interpreted as a weighted combination of $(n + 1)$ control points \mathbf{p}_k with weights that vary with time via the basis functions. The first d entries in \mathbf{x}_t correspond to \mathbf{p}_0 , the second d entries correspond to \mathbf{p}_1 , etc. If we arrange the control points as the rows of a matrix $\mathbf{P} \in \mathbb{R}^{(n+1) \times d}$ we can write most succinctly:

$$\mathbf{c}(\tau) = \Phi^T(\tau) \mathbf{P}$$

We thus have a compact representation of a trajectory in terms of its control points. We will make use of all notations introduced in our further discussion. We stress the importance of having a parameterization that has spatial semantics: Trajectory prediction and scenario prediction in particular are about the interaction of curves extending through time and space with purely spatial features. Hence, a common footing can provide the basis for effectively modelling the interactions of trajectories and the static environment.

We are free in our choice of basis functions and can even learn them from data. A classical choice are monomials $\phi_k = \tau^k$. Then, we are effectively using a Taylor expansion of the trajectory in each spatial dimension. This provides provable limits on the approximation error if we are able to establish limits on the higher derivatives. This is possible for physical agents due to physical constraints. We can further express preference for

comfortable trajectories by placing zero mean priors on the value of the coefficients corresponding to jerk and higher derivatives [7, 8, 9]. We denote the vector of monomial basis functions by $\Phi_M(\tau)$. Another classical choice is to use Bernstein Polynomials, effectively using Bézier curves as trajectory/state representation [10, 11, 12, 13]. The corresponding vector of Bernstein Polynomials is $\Phi_B^T(\tau) = \Phi_M^T(\tau)\mathbf{M}$ with $\mathbf{M} \in \mathbb{Z}^{(n+1) \times (n+1)}$ defined as [14]:

$$M_{ij} = \begin{cases} \binom{n}{j-1} \binom{n-j+1}{i-j} (-1)^{(i+j) \bmod 2} & \text{if } j \leq i \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

This choice renders the location of the control points especially intuitive with \mathbf{p}_0 and \mathbf{p}_n always at the end points, $\mathbf{p}_0 - \mathbf{p}_1$ and $\mathbf{p}_n - \mathbf{p}_{n-1}$ tangent to the trajectory at the endpoints and the entire trajectory confined to the convex hull of the control points.

Figure 2 shows an example of such a trajectory representation with Bernstein Polynomials as basis functions.

Due to fixed basis functions, we have a linear observation model for $\mathbf{c}_t(\tau)$ and its time derivatives by construction. Our formalism allows to make observations of the trajectory at any point in time. However, for tracking applications, we are most interested in the observation model at $\tau = 1$, the end of the trajectory at current time t . The natural observable is of course $\mathbf{c}_t(\tau)$ and its derivatives with respect to time, i.e. velocities and accelerations. Due to our re-scaling of time, we have $dt = \Delta t d\tau$:

$$\frac{d^n}{dt^n} \mathbf{c}_t(\tau) = \underbrace{\frac{1}{(\Delta t)^n} \left(\frac{d^n}{d\tau^n} \Phi^T(\tau) \otimes \mathbf{I}_d \right)}_{\mathbf{H}_n(\tau)} \mathbf{x}_t \quad (2)$$

This amounts to a constant observation matrix for every value of τ that can be easily combined. For example, let us assume we are observing an object's position x, y and velocity v_x, v_y organized in an observation vector $\mathbf{o}_t = [x, y, v_x, v_y]$, the corresponding linear observation model \mathbf{H} is then given by stacking the rows of \mathbf{H}_0 and \mathbf{H}_1 from above. The corresponding observation noise \mathbf{R} can then be used to reflect measurement noise.

Let us stress the benefits of linearity at this point. To determine the $n + 1$ linear coefficients necessary for our state representation, only $n + 1$ linear measurements at arbitrary times τ are necessary. For example, using $n + 1 = 6$ polynomial basis functions, obtaining position, velocity and acceleration at start and end point completely determines the entire trajectory in between. This effectively corresponds to a two point Taylor expansion of the trajectory to degree 2 which results in the same accuracy as a Taylor expansion to degree 4 at a single point [15]. Together with the physically imposed limits on the higher derivatives of the motion of massive objects, our formalism should allow a highly accurate representation of physical trajectories over time horizons Δt of a few seconds. We will come back to this point when discussing trajectory prediction.

Ego motion compensation is equally trivial as the control points transform as points fixed in space under sensor movement. Augmenting the state vector to homogeneous coordinates renders ego motion compensation another linear transform. This enables tracking of the dynamic environment directly from the perspective of the autonomous vehicle, i.e. in ego-coordinates. In moving sensor applications, the ability of being able to transform our state representation to any new coordinate systems is a key advantage

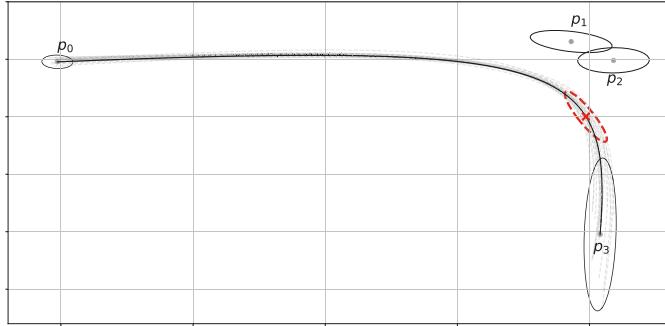


Figure 2: Example of a trajectory representation with $n = 3, d = 2$. The solid line corresponds to the mean trajectory and the dashed lines are samples from the density. Parameters, i.e. control points, are indicated at the mean values and with 95% confidence intervals. The red dashed ellipse encloses a 95% confidence region for the position of the object at $\tau = 0.7$.

over alternative state representations such as LSTM recurrent neural networks that are often employed in trajectory prediction tasks.

The next component we need in a filtering architecture is the motion model. We will consider two ways to derive a linear motion model. First, let us consider the case in which we can find a linear derivative operator \mathbf{D} for our vector of basis functions:

$$\dot{\Phi}^T(\tau) = \Phi^T(\tau)\mathbf{D} \quad (3)$$

For the monomial basis functions, this derivative operator is

$$\mathbf{D}_M = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 2 & 0 & \dots \\ 0 & 0 & 0 & 3 & \dots \\ \dots & & & & \dots \end{bmatrix}$$

We now observe the associativity in:

$$\dot{\mathbf{c}}(\tau) = \dot{\Phi}^T(\tau)\mathbf{P} = (\Phi^T(\tau)\mathbf{D})\mathbf{P} = \Phi^T(\tau)(\mathbf{D}\mathbf{P}) = \Phi^T(\tau)\dot{\mathbf{P}} \quad (4)$$

which gives us a linear motion model for the control points $\dot{\mathbf{P}} = \mathbf{D}\mathbf{P}$ and consequently for our state vector:

$$\frac{d}{dt}\mathbf{x} = \frac{1}{\Delta t}(\mathbf{D} \otimes \mathbf{I}_d)\mathbf{x} \quad (5)$$

for a given $\delta t = \Delta t \delta \tau$ this can be integrated analytically as a matrix exponential to yield:

$$\begin{aligned}\mathbf{F}_{\delta t} &= \exp \left(\frac{\delta t}{\Delta t} (\mathbf{D} \otimes \mathbf{I}_d) \right) \\ \mathbf{x}_{t+\delta t} &= \mathbf{F}_{\delta t} \mathbf{x}_t\end{aligned}\tag{6}$$

We denote the derivative for Bernstein Polynomials can be obtained from \mathbf{D}_M via a similarity transform: $\mathbf{D}_B = \mathbf{M}^{-1} \mathbf{D}_M \mathbf{M}$.

The second way of generating a motion model is derived from a fit procedure: Assume at time t , we had $m \leq n+1$ samples of a trajectory $\mathbf{c}_t(\tau_i)$ from different times τ_i , $i \in \{1, \dots, m\}$ between $\tau_1 \geq 0$ and $\tau_m \leq 1$. We arrange these samples as the rows of an $m \times d$ matrix \mathbf{C}_t . We now form an $m \times (n+1)$ matrix \mathbf{B} so that row i of \mathbf{B} corresponds to $\Phi(\tau_i)$. Then, we could estimate the control points, i.e. our state vector, as a Bayesian least square fit to the samples of the trajectory:

$$\mathbf{P}_t = (\mathbf{B}^T \mathbf{R}^{-1} \mathbf{B} + \Sigma_P^{-1})^{-1} \mathbf{B}^T \mathbf{R}^{-1} \mathbf{C}_t \tag{7}$$

Here Σ_P is the covariance matrix of a possible zero mean Gaussian prior and \mathbf{R} the covariance of Gaussian observation noise. Note that a zero mean prior is a natural assumption if one of our basis functions is constant as is the case for both Φ_M and correspondingly Φ_B . Further, this prior is the natural expression for any inductive biases one may wish to express for the motion model.

With this in mind, the way to propagate a trajectory forward in time by δt now is the following: From the current state estimate, i.e. the control points \mathbf{P}_t , we generate samples along the mean of the trajectory at $n+1$ equal spaced $\tau'_i = \delta t / \Delta t + i(1 - \delta t / \Delta t) / n$, $0 \leq i \leq n$ and then re-estimate the control points from these samples pretending they were obtained at $\tau_i = \tau'_i - \delta t / \Delta t$. To do this, as before, we construct the $(n+1) \times (n+1)$ matrices \mathbf{B} and \mathbf{B}' from $\Phi(\tau_i)$ and $\Phi(\tau'_i)$, respectively. We then get transformed control points from

$$\mathbf{P}_{t+\delta t} = (\mathbf{B}^T \mathbf{B} + \Sigma_P^{-1})^{-1} \mathbf{B}^T \mathbf{B}' \mathbf{P}_t \tag{8}$$

Since we are estimating the new control points from samples of the mean, we don't need to consider the observation covariance matrix \mathbf{R} in this expression. Without a prior, this formula reduces to

$$\mathbf{P}_{t+\delta t} = \mathbf{B}^{-1} \mathbf{B}' \mathbf{P}_t \tag{9}$$

and it should be noted that (9) is consistent with our earlier approach (6) if a linear derivative operator exists:

$$\mathbf{B}^{-1} \mathbf{B}' = \exp \left(\frac{\delta t}{\Delta t} \mathbf{D} \right) \tag{10}$$

We thus have a linear motion model $\mathbf{F}_{\delta t}$ for the control points that only depends on our choice of basis functions and correspondingly for the state vector \mathbf{x}_t we find

$$\begin{aligned}\mathbf{F}_{\delta t} &= ((\mathbf{B}^T \mathbf{B} + \Sigma_P^{-1})^{-1} \mathbf{B}^T \mathbf{B}') \otimes \mathbf{I}_d \\ \mathbf{x}_{t+\delta t} &= \mathbf{F}_{\delta t} \mathbf{x}_t\end{aligned}\tag{11}$$

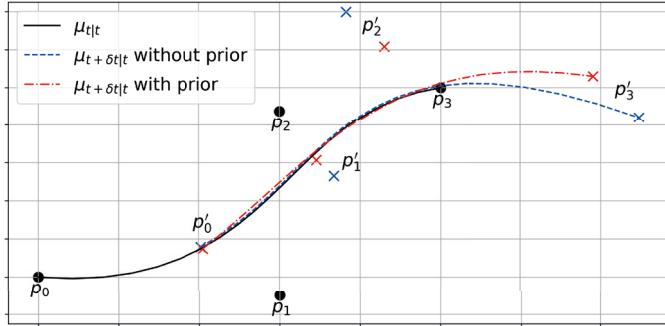


Figure 3: Example of the action of the motion model with $\delta t = 0.3\Delta t$. Note how the motion model without prior follows the trajectory exactly for all points in the past, while the model with prior deviates slightly from the past estimation.

Figure 3 illustrates this on a cubic Bézier curve and exemplifies the differences of the motion model with and without the use of a prior.

With this, we have a compact representation of object trajectories together with linear observation and motion models that enables closed form updates under a Gaussian density assumption. Thus, it can serve as a drop in replacement in any object tracker, both single object or multi-object. Observations at the most recent time point $\tau = 1$ are naturally absorbed into the tracked trajectory via the observation matrix $\mathbf{H}(\tau)$ and the linear motion model $\mathbf{F}_{\delta t}$ propagates the trajectory forward in time.

3 Trajectory Prediction

So far, we only absorb actual observations and thus are limited to track past trajectories. However, if the proposed representation is accurate for past trajectories, then it is equally adequate for future trajectories. For future trajectories, we use the semantics of $\tau = 0$ corresponding to t and $\tau = 1$ corresponding to $t + \Delta t$.

As discussed, for $n+1$ basis functions, we only need $n+1$ linear measurements to fully determine the trajectory over the entire duration of Δt . With $n+1 = 6$ and position, velocity and acceleration given at the start $\tau = 0$, we only need 3 more observations and the natural choice is position, velocity and acceleration at the end $\tau = 1$. We denote these hypothetical measurements "pseudo-observations" \mathbf{o}^f at $\tau = 1$ to differentiate them from actual observations \mathbf{o} used for tracking. This concept of pseudo-observations at a constant time horizon is motivated by the rich literature on driver modeling [7] and preview control of vehicles [16, 17].

For vehicles, the road topology gives us a finite number of path options in the form of center-lines and we choose to restrict the position of our endpoints to lie on these

path options [18, 19, 20]. We further choose to restrict velocities and accelerations at the endpoints to be tangent to the path option. We thus only have to find the position along the path option, the longitudinal velocity and acceleration. This choice deliberately rules out the possibility for pseudo-observations on lane boundaries for example. For the time horizons considered here, there will generally not be enough evidence to distinguish such a hypothesis from either a completed lane change or a keep lane hypothesis. Note that this does not rule out predicting an object's position away from center-lines at intermediate times between t and $t + \Delta t$ - it just rules this out for pseudo-observations at $t + \Delta t$. If an object indeed crosses a lane boundary at $t + \Delta t$, our measurements will certainly lend support to the hypothesis of a lane change at some later point in time $t' \in (t, t + \Delta t)$ when we consider pseudo-observations at $t' + \Delta t$.

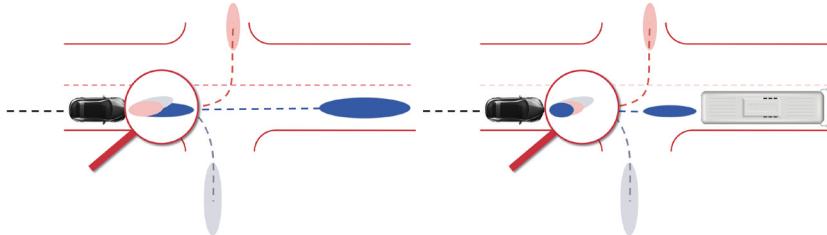


Figure 4: The static and dynamic environment shape the long term path options for a traffic participant. The distribution of vehicle states at $t + \Delta t$ is generally multi-modal and influenced not only by the static environment but also by other traffic participants. This multi-modal structure of the long term options shapes the expected state distribution in short time horizons $t + \delta t$.

Figure 4 illustrates this concept for a single vehicle in two different environmental contexts. In each environment, there are $K = 3$ path options for the vehicle. The pseudo-observations along these path options are uncertain in position, velocity and acceleration. However, since we can assume continuity of trajectories in position, velocity and acceleration, the hypothesized object state at $\tau = 1$ and the current vehicle state at $\tau = 0$ fully define a trajectory and we can propagate back the uncertainty of the pseudo-observation from $\tau = 1$ to $\tau = \delta t / \Delta t$ when we will receive the next actual observation of the vehicle dynamics. Note that due to the multitude of path options, we obtain a *mixture* distribution for expected vehicle dynamics in a short time horizon. This means, conditioned on the K pseudo-observation, we could consider K different motion models:

$$P_{t+\delta t|t}^k = \int d\mathbf{x}_t P(\mathbf{x}_{t+\delta t} | \mathbf{x}_t, \mathbf{o}_k^f) P_{t|t} \quad (12)$$

The probability, which of the path options is likely taken, is then evaluated in light of new evidence together with a prior probability $P(\mathbf{o}_k^f)$ of the path option (which may be uniform):

$$P(\mathbf{o}_k^f | \mathbf{o}_{t+\delta t}) = \frac{P(\mathbf{o}_k^f) \int d\mathbf{x}_{t+\delta t} P(\mathbf{o}_{t+\delta t} | \mathbf{x}_{t+\delta t}) P_{t+\delta t|t}^k}{\sum_{k'} P(\mathbf{o}_{k'}^f) \int d\mathbf{x}_{t+\delta t} P(\mathbf{o}_{t+\delta t} | \mathbf{x}_{t+\delta t}) P_{t+\delta t|t}^{k'}} \quad (13)$$

A sensible choice for the pseudo-observations is to select the furthest point in the path together with velocity and acceleration that is reachable without violating comfort levels [8, 21, 22]. We are free to choose the uncertainties at $\tau = 1$. This approach is similar to [18, 23] but instead of forward generating the hypotheses, we interpolate back from a single step prediction. Another excellent choice is to employ a Gaussian Process conditioned on the past trajectory and the path ahead to provide an estimate of both mean and covariance of our pseudo-observations [24]. The covariances of the pseudo-observations must be large enough as to cover the option space plausibly, but small enough as to be discriminative in the near future.

Figure 5 illustrates this concept numerically in a toy example. A single vehicle approaches the crossing. The road topology provides three path options. We construct plausible end states as pseudo-observations at $t + \Delta t$ with $\Delta t = 5s$ for each of these options together with associated uncertainties. From a common initial condition, we simulate three actual vehicle trajectories along these path options. Note how these actual trajectories deviate from the mean hypothesized trajectories due the holonomic constraints of the vehicle and the cost function of the vehicle controller.

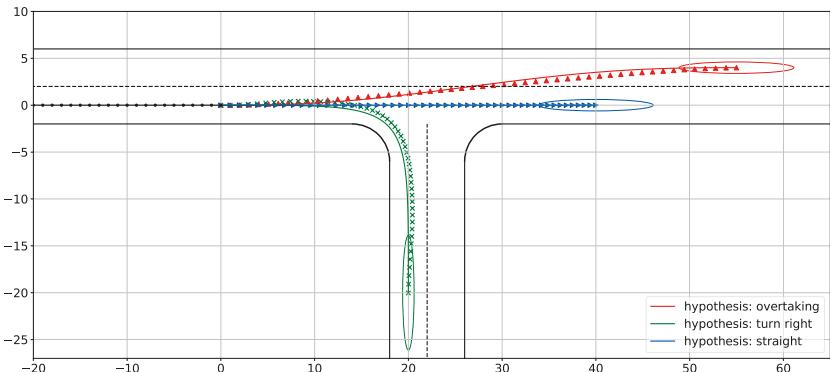


Figure 5: A vehicle approaches a crossing at time $t = 0$ with $v_0 = 10m/s$, $a_0 = 0$ with three hypotheses of future motion: a right turn, going straight across and decelerating possibly due a leading vehicle, and changing lanes and accelerating to overtake the leading vehicle. We show 95% confidence ellipses for pseudo-observations at $t+5s$ and the resulting mean trajectory for each hypothesis. With individual markers, we denote three actual trajectories obtained from simulating a vehicle along these path options for $5s$.

We are now interested at what point along the vehicle trajectory, it is possible to differentiate which of the options is actually chosen. Figure 6 illustrates this. If we only compare the vehicle's position to the expectation from each hypothesis, we reach high confidence at around $1s$ into the scene, (Figure 6, top). If we additionally consider the velocity, high confidence is reached at about $0.5s$ (Figure 6, middle). Also note how the turn right maneuver is competing with the decelerating straight maneuver early into the

scene as both maneuvers are decelerating and the simulated vehicle veers slightly to the left to increase the turn radius before the right turn. The bottom of Figure 6 shows the marginal covariances of the hypothesized trajectories at $t = 1\text{s}$ into the scene together with the actual measurement of the three simulated trajectories.

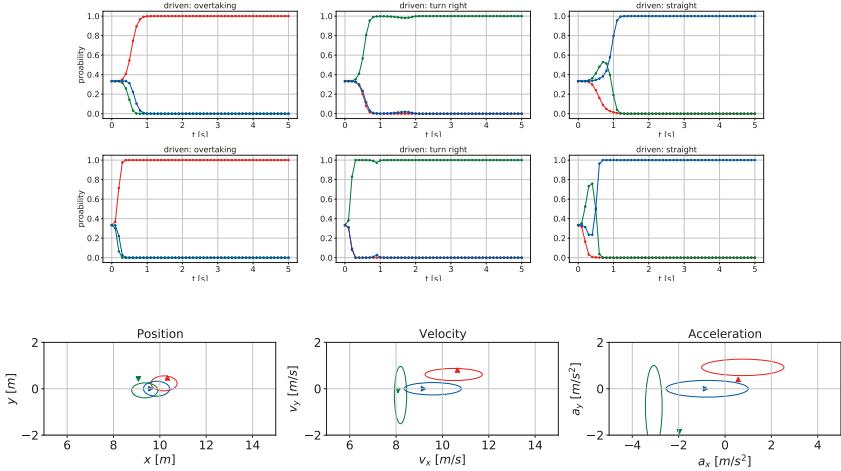


Figure 6: **Top:** The posterior probability of each hypothesis from Figure 5 for each of the actually driven trajectories when considering only position information. **Middle:** When considering both position and velocity. **Bottom:** Distribution of the hypotheses dynamical variables at $t = 1\text{s}$ into the situation together with measured quantities from driven trajectories. Colors and markers correspond to Figure 5.

In a live system running at sub second cycle times δt the difference between future options may be too small to differentiate between them, in particular as current state estimates and pseudo-observations are updated at this cycle time, too, and by construction, the end of a tracked trajectory is always 100% compatible with all future trajectories resulting from pseudo-observations. Hence, the current vehicle state does not provide any evidence about which of the future options an object is going to take. We can, however, check the compatibility of the past trajectory with a future option at the current point in time. For this, we form a transition trajectory from observations made at $t - \alpha\Delta t$ on the past tracked trajectory and at $t + (1 - \alpha)\Delta t$ on the hypothesised trajectory. Typical values for $\alpha\Delta t$ are in the range of 0.5s identified in our preceding analysis. On this transition trajectory, we find the expectation for the current kinematic object state at $\tau = \alpha$. The higher the compatibility between past and future trajectory, the closer this expectation will match the current kinematic vehicle state. This effectively corresponds to an "observation likelihood" for future trajectories. This approach is similar in spirit to multiple model filters [25], but here, the "models" are constructed from pseudo-observations for given path options.

We can leverage this observation likelihood directly in the framework of a multi-hypotheses tracking algorithm. Before the data association step, we augment each local hypothesis (corresponding to a single tracked object) with its possible path-options and pseudo-observations. The entries in the cost matrix during the data association step are then calculated from the observation likelihood of the transition trajectories as described above [5]. That is, we effectively employ (12) in the calculation of the observation likelihood. The standard data association algorithm (Murty's Algorithm [26, 27]) now produces a ranking of the most likely assignments of object detections to local hypotheses under the constraint that all object detections are assigned and each local hypothesis is assigned at most one observation. Each of these assignments forms a new global hypothesis weighted by its total likelihood and, thus, provides a multi-modal multi-object representation of the scene. We alter this algorithm slightly by adding the additional group constraint that in each global hypothesis, only one of the possible futures for a local hypothesis can be present. This constraint can be fulfilled trivially for the most likely global hypothesis by considering only the maximum entry in each group of future trajectories. As Murty's algorithm generates alternative assignments from this globally optimal solution, we only need to adapt the exchange rules in order to reflect the additional constraints. The resulting ranking of data associations represents a probability weighted multi-modal multi-object representation of the current traffic scene and its future development. The ranking can then be further pruned by applying additional constraints such as traffic rules. The pruning of highly probable hypotheses that violate traffic rules can be taken as a consideration of an imminent violation of traffic rules.

4 Conclusion

We have introduced a parametric representation for object trajectories as Markov states for object tracking together with corresponding linear motion and observation models that allow closed form Kalman filtering under a Gaussian density approximation. Albeit we have illustrated and motivated these concepts primarily with vehicles, they are by no means limited to represent vehicle trajectories. Rather, they apply to all physical traffic participants that cannot change their state of motion arbitrarily fast. We have further shown how this representation can be used in trajectory prediction tasks and how it can be integrated into multi-object multi-hypothesis trackers to represent consistent probabilistic multi-modal distributions over the future of entire traffic scenarios.

5 Acknowledgements

The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Climate Action within the project “KI Wissen – Entwicklung von Methoden für die Einbindung von Wissen in maschinelles Lernen”. The author would like to thank the consortium for the successful cooperation and Jörg Dietrich for a careful review of the manuscript.

References

- [1] S. S. Haykin, ed., *Kalman filtering and neural networks*. Adaptive and learning systems for signal processing, communications, and control, New York: Wiley, 2001.
- [2] T. Gindele, S. Brechtel, and R. Dillmann, “A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments,” in *13th International IEEE Conference on Intelligent Transportation Systems*, (Funchal, Madeira Island, Portugal), pp. 1625–1631, IEEE, Sept. 2010.
- [3] M. Schreier, V. Willert, and J. Adamy, “An Integrated Approach to Maneuver-Based Trajectory Prediction and Criticality Assessment in Arbitrary Road Environments,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, pp. 2751–2766, Oct. 2016.
- [4] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, “Multiple Hypothesis Tracking Revisited,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015.
- [5] A. F. Garcia-Fernandez, J. L. Williams, K. Granström, and L. Svensson, “Poisson Multi-Bernoulli Mixture Filter: Direct Derivation and Implementation,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, pp. 1883–1901, Aug. 2018.
- [6] K. Granström, L. Svensson, Y. Xia, J. Williams, and F. García-Fernández, “Poisson Multi-Bernoulli Mixtures for Sets of Trajectories,” *arXiv:1912.08718 [cs, eess, stat]*, Dec. 2019. arXiv: 1912.08718.
- [7] C. C. Macadam, “Understanding and Modeling the Human Driver,” *Vehicle System Dynamics*, vol. 40, pp. 101–134, Jan. 2003.
- [8] I. Bae, J. Moon, J. Jhung, H. Suk, T. Kim, H. Park, J. Cha, J. Kim, D. Kim, and S. Kim, “Self-Driving like a Human driver instead of a Robocar: Personalized comfortable driving experience for autonomous vehicles,” *arXiv:2001.03908 [cs, eess]*, Jan. 2020. arXiv: 2001.03908.
- [9] H. Hayati, D. Eager, A.-M. Pendrill, and H. Alberg, “Jerk within the Context of Science and Engineering—A Systematic Review,” *Vibration*, vol. 3, pp. 371–409, Oct. 2020.
- [10] F. de Dilectis, D. Mortari, and R. Zanetti, “Bézier Description of Space Trajectories,” *Journal of Guidance, Control, and Dynamics*, vol. 39, pp. 2535–2539, Nov. 2016.
- [11] J. J. Faraway, M. P. Reed, and J. Wang, “Modelling three-dimensional trajectories by using Bézier curves with application to hand motion: *Modelling Three-dimensional Trajectories*,” *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 56, pp. 571–585, Nov. 2007.
- [12] R. Hug, W. Hübner, and M. Arens, “Modeling continuous-time stochastic processes using N-Curve mixtures.,” *CoRR*, 2019.

- [13] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, “Optimal trajectory generation for dynamic street scenarios in a Frenet Frame,” in *2010 IEEE International Conference on Robotics and Automation*, (Anchorage, AK), pp. 987–993, IEEE, May 2010.
- [14] S. Ray and P. Nataraj, “A Matrix Method for Efficient Computation of Bernstein Coefficients.,” *Reliab. Comput.*, vol. 17, no. 1, pp. 40–71, 2012.
- [15] R. Estes and E. Lancaster, “Two-point taylor series expansions,” tech. rep., Goddard Space Flight Center, Greenbelt, Maryland, 1966.
- [16] M. F. Land and D. N. Lee, “Where we look when we steer,” *Nature*, vol. 369, pp. 742–744, June 1994.
- [17] O. Lappi, “Future path and tangent point models in the visual control of locomotion in curve driving,” *Journal of Vision*, vol. 14, pp. 21–21, Oct. 2014.
- [18] D. Petrich, T. Dang, D. Kasper, G. Breuel, and C. Stiller, “Map-based long term motion prediction for vehicles in traffic environments,” in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, (The Hague, Netherlands), pp. 2166–2172, IEEE, Oct. 2013.
- [19] C. Yang, M. Bakich, and E. Blasch, “Nonlinear constrained tracking of targets on roads,” in *2005 7th International Conference on Information Fusion*, vol. 1, pp. 8–pp, IEEE, 2005.
- [20] C. Yang and E. Blasch, “Fusion of tracks with road constraints,” tech. rep., AIR FORCE RESEARCH LAB WRIGHT-PATTERSON AFB OH SENSORS DIRECTORATE, 2008.
- [21] G. Arechavaleta, J.-P. Laumond, H. Hicheur, and A. Berthoz, “An Optimality Principle Governing Human Walking,” *IEEE Transactions on Robotics*, vol. 24, pp. 5–14, Feb. 2008.
- [22] A. Takahashi, T. Hongo, Y. Ninomiya, and G. Sugimoto, “Local Path Planning And Motion Control For Agv In Positioning,” in *Proceedings. IEEE/RSJ International Workshop on Intelligent Robots and Systems '89. (IROS '89) 'The Autonomous Mobile Robots and Its Applications*, (Tsukuba, Japan), pp. 392–397, IEEE, 1989.
- [23] Q. Tran and J. Fir, “Modelling of traffic situations at urban intersections with probabilistic non-parametric regression,” in *2013 IEEE Intelligent Vehicles Symposium (IV)*, (Gold Coast City, Australia), pp. 334–339, IEEE, June 2013.
- [24] J. Wiest, M. Hoffken, U. Kresel, and K. Dietmayer, “Probabilistic trajectory prediction with Gaussian mixture models,” in *2012 IEEE Intelligent Vehicles Symposium*, (Alcal de Henares , Madrid, Spain), pp. 141–146, IEEE, June 2012.
- [25] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, “Interacting multiple model methods in target tracking: a survey,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, pp. 103–123, Jan. 1998.

- [26] K. G. Murty, “Letter to the Editor—An Algorithm for Ranking all the Assignments in Order of Increasing Cost,” *Operations Research*, vol. 16, pp. 682–687, June 1968.
- [27] M. Motro and J. Ghosh, “Scaling data association for hypothesis-oriented mht,” in *2019 22th International Conference on Information Fusion (FUSION)*, pp. 1–8, IEEE, 2019.

Approach to Maintain a Safe State of an Automated Vehicle in Case of Unsafe Desired Behavior

Christoph Popp*, Stefan Ackermann*, Hermann Winner*

Abstract: For automated driving, higher levels of automation pose new challenges in terms of safety. In this paper, we develop a generic behavior safety framework that maintains a safe vehicle state even in case of system failures. It is applicable to different configurations of automated driving system architectures. We verify the designed generic behavior safety framework by applying it to two different architectures from both projects PRORETA 5 and UNICARagil. The previously defined safety requirements are met with both applications, which indicates that the developed generic safety framework is also valid for other configurations of automated driving systems.

Keywords: automated driving, minimal risk condition, safety, vehicle behavior

1 Introduction

Providing safety for automated vehicles remains an unsolved challenge for stakeholders of automated driving. In operation, an automated vehicle shall provide safe behavior at all times. Unsafe desired behavior led to fatal accidents in the past. For example, there were accidents involving vehicles from Tesla in 2016 [1] and Uber in 2018 [2]. The published accident reports reveal that both accidents were caused by faults in the processing sequence between environment perception and desired behavior planning combined with failed vehicle behavior monitoring by the safety driver. These vehicles were not in SAE level 4 operation mode.

Those fatal accidents prove that the state of the art automated driving functions (perception, interpretation and behavior planning) do not yet have the necessary capabilities to generate safe vehicle behavior at any given time. We propose our Behavior Safety Framework (BSF). It is a necessary component to aim to keep automated vehicles in a safe state for state of the art automated vehicles. In this paper, we present the state of the art for safeguarding the desired behavior of automated vehicles and elaborate the lack of a BSF providing safe vehicle behavior in all driving situations. We further identify the requirements for such a safety framework and design a generic modular architecture. We verify this architecture in a formal way and also by applying the generic architecture to two specialized applications within different automated driving system (ADS) configurations. We conclude that our generic architecture meets our set of requirements and that the verification, both formally and by applying it to two example applications, indicate that the generic architecture is valid for other ADS configurations as well.

*Institute of Automotive Engineering (FZD) at Technical University of Darmstadt, 64287 Darmstadt
(e-mail: firstname.lastname@tu-darmstadt.de).

2 Methodology

The purpose of this paper is to provide a functional safety assessment of the desired behavior of an automated vehicle equipped with a level 4 ADS according to SAE J3016 [3] and to generate a safe desired behavior if needed. We begin with a survey of the state of the art for safeguarding the desired behavior of a highly automated vehicle in Section 3. By comparing the state of the art with our problem statement, we conclude that it does not provide a generic modular architecture of a BSF. This conclusion motivates the development of such a framework in Section 4. In Section 4.1 and 4.2, we provide definitions and derive the requirements for a modular BSF, based on which we design a generic architecture for the BSF in Section 4.3. The verification of our architecture is done using two approaches in Section 5. For the first approach, we formally compare the capabilities of our BSF with the requirements. For the second approach, we apply our generic architecture to two applications with different ADS configurations and verify that the requirements are met even with different ADS configurations. Finally, we review our results and give an outlook on future research in Section 6.

3 State of the Art

The literature overview contains several approaches for safeguarding automated vehicles. Some of these are briefly presented below.

Shalev-Shwartz et al. [4], Nistér et al. [5], and based on these, de Iaco et al. [6] provide formal behavioral descriptions that prevent collisions between road users. If all road users adhere to mathematically defined safety distances in the longitudinal and lateral directions as well as other behavioral guidelines and react appropriately in dangerous situations, no collisions can occur. Other behavioral guidelines are among others to "not cut-in recklessly", that "right-of-way is given, not taken" in cooperative situations and to "be careful of areas with limited visibility" [4].

Molina et al. [7] present an architecture approach for vehicle behavior safeguarding where the outputs from the behavior planning modules are not tested directly, but indirectly by monitoring the vehicle behavior. The monitoring system is equipped with environmental perception separated from the primary environment perception and intervenes when criticalities are detected. It then overrides the controller output of the primary system and specifies a risk-minimizing vehicle behavior corresponding to the situation.

Stolte et al. [8] develop an ADS for an unmanned protective vehicle for the highway hard shoulder. In the event of component or system failures, or if the defined system limit is exceeded, the vehicle brakes to standstill. The vehicle estimates its own perception quality and uses sensor redundancies to ensure the required safety.

Pek et al. [9] present a safety layer that can be used for existing motion planners. During the operation of the automated vehicle, all legally possible movements of other road users are predicted and the safety of the current traffic situation is thus assessed in real time. In case of identified unsafeties, a combination of reachability analysis and convex optimization is used to determine drivable fail-safe emergency trajectories that end in a safe area with vehicle standstill. Similarly, the concept of Stahl et al. [10] checks the results of specific modules whose functional safety can only be insufficiently proved. In case of unsafety, a previously defined emergency braking trajectory is used.

Further approaches for emergency trajectories are presented, among others, by Funke et al. [11], Hilgert et al. [12] and Mehmed et al. [13]. For emergency maneuvers, Reschka [14] considers three options: Decelerating with constant steering angle, decelerating along the last calculated path, or stopping at a suitable place, for example on the roadside. Ackermann and Winner [15] present another option for an emergency maneuver: The behavior planner searches for safe stop locations and plans a minimal risk maneuver as described in ISO/TR 4804 [16] and ISO/DIS 21448 [17] to these locations. The purpose of this is to reach the minimal risk condition for each emergency situation.

In summary, literature contains many different approaches to increase safety of automated vehicles. We listed formal behavior rule definitions and various concepts to ensure the safety of the whole ADS. Still, the state of the art does not provide a generic architecture of a BSF that fully safeguards the desired behavior of an ADS for an automated vehicle. In most of the safeguarding concepts from literature, it is assumed that the vehicle environment is always perceived by the perception system according to reality. Thus, false-negative object detections, e.g. due to damaged or decalibrated sensors, are neglected. Also, failures of sensors or the entire ADS are barely addressed in the literature.

4 Behavior Safety Framework

4.1 Definitions

In this chapter, we define the terminology that we use for the development of the BSF.

Automated Driving System (ADS)

First, we present our definition of an automated driving system (ADS), which is shown in Figure 1.

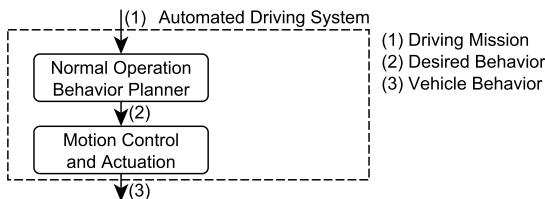


Figure 1: Definition of the ADS.

The input to the ADS is a desired driving mission. The normal operation behavior planner contains all functional submodules of sensing and planning that are required for the automated driving task. This includes the submodules for collection and interpretation of environmental sensor data, the localization and dynamic state estimation of the ego-vehicle, the prediction of future object behavior and the desired behavior planning. The result is the desired vehicle behavior, which is expressed as a desired trajectory. It contains information about desired poses and dynamics of the ego-vehicle for the upcoming time. The result from the motion control and actuation is the real vehicle behavior.

Due to a variety of possible combinations of the functions of an automated vehicle, different ADS configurations are conceivable. Some of the functions can be AI-assisted

and different strategies for environment interpretation or desired behavior planning can be pursued. However, end-to-end learning is excluded by this ADS architecture. Otherwise, an intervening safety function can not be integrated.

Behavior Safety Framework (BSF)

The BSF extends this basic ADS by adding safeguarding functions. In order not to reduce the solution space in advance, we do not define specific interfaces of interaction between the BSF and the other modules in Figure 1 for now.

Safe State

For the definition of the safe state, we use the definition from ISO/TR 4804 [16]: A safe state is an “operating mode that is reasonably safe”.

Safety of Desired Behavior

Regarding the desired behavior of the automated vehicle, the terms ‘safety’ and ‘unsafety’ need to be defined. Blokland and Reniers [18] do not directly refer to automated vehicles but in general to industrial safety and state that there is actually ”no commonly agreed upon definition of ‘safety’ nor of its opposite ‘unsafety’”. Today, safety is mostly defined by an absence of accidents”. According to this, situations with near misses are still considered to be safe and furthermore, a situation can only be classified as safe or unsafe retrospectively. Since we have to evaluate a priori whether the desired behavior is safe, we cannot use this definition. A dictionary definition of unsafety is the ”exposure to danger or risk” [19]. Junietz [20] defines it in a similar, but more detailed way in the context of automated driving: a system is in an unsafe state, if ”the corresponding risk is not accepted” . For the following considerations, we use the definition of Junietz.

4.2 Requirements

In this chapter, we define requirements for our BSF to safeguard the desired behavior of an automated vehicle. The task of this framework is to maintain the safe state of an automated vehicle, even with unsafe desired behavior. This shall not be performed by our BSF verifying the algorithms of the ADS, but by verifying the result of the ADS. This result is the planned desired behavior, usually defined as desired trajectory. Additionally, an appropriate risk-minimizing reaction shall be triggered in the event of detected unsafe desired behavior.

The BSF is comparable to a horse in a carriage team. The coachman leads the horse with the harness to a desired behavior. So in most cases the coachman influences the behavior of the horse. However, if a coachman leads a horse into a roadside ditch, the horse will reject this desired behavior and stop the carriage before the roadside ditch to avoid being injured. In this case, we cannot claim that the horse verifies the planning of the coachman, but only the result of the planning. In the same way, we do not presume to verify the algorithms of an ADS, but to verify that the automated vehicle does not pose an unreasonable risk.

The first requirement describes this basic functionality, from which further requirements are derived. We use the definition for the minimal risk condition as described in ISO/TR 4804 [16] and ISO/DIS 21448 [17].

Requirement 1 *The BSF shall transition the ego vehicle to a minimal risk condition in case of an identified unsafety of the desired behavior.*

Thus, the BSF shall perform an emergency maneuver to maintain the safe state of the automated vehicle. To avoid collisions with obstacles during the emergency stop maneuver, the BSF shall have the capability to perceive and interpret the relevant vehicle environment.

As addressed in Requirement 1, in case of detected unsafe states, the task of the safety framework is to achieve a minimal risk condition or to reduce the hazard and risk to an acceptable level by appropriate intervention in the vehicle behavior. For this, it is also crucial that unsafe states are identified fast enough.

Requirement 2 *The BSF shall detect unsafeties in the desired behavior and generate a suitable intervention fast enough to reduce the risk to an acceptable level.*

The detection of those unsafeties in the desired behavior includes on the one hand to be aware of whether the capabilities of the ADS-modules are sufficient to plan a safe desired behavior. This can involve the collection and interpretation of environmental sensor data, the localization and dynamic state estimation of the ego-vehicle, the prediction of future object behavior and the desired behavior planning. On the other hand, potential accidents with static or dynamic objects caused by performing the desired behavior shall also be detected.

Requirement 3 *The BSF shall be aware of whether the current capabilities of the vehicle are sufficient for safe vehicle operation.*

Requirement 4 *The BSF shall detect whether performing the desired behavior would lead to an accident.*

The safety framework shall be able to actively influence the vehicle behavior. Therefore it shall be able to set interface-compliant commands to the corresponding actuators for longitudinal and lateral vehicle movement. This can be done either directly by actuator commands or indirectly by sending the desired emergency behavior to the motion controller.

Requirement 5 *The BSF shall have direct or indirect access to the relevant actuators of the vehicle.*

In case of functional deficiencies of the BSF itself, the safety of the vehicle cannot be ensured anymore. Thus, they also need to be detected.

Requirement 6 *The capabilities of the BSF shall be monitored online.*

As a part of the ADS, the BSF is critical for the safety of the vehicle behavior. The safety framework shall therefore fulfill high safety requirements. To ensure this, the dedicated functionality must be verifiable, which corresponds to the seventh requirement.

Requirement 7 *The safety and functionality of the BSF shall be testable and verifiable.*

4.3 Functional Architecture

Figure 2 illustrates the generic functional architecture of the BSF. The desired behavior planned by the normal operation behavior planner is the input of the BSF. The BSF has five functional submodules: the ADS health state data reception, the environment perception data reception, the interpretation of the relevant environment, the emergency behavior generation and the safe behavior selection.

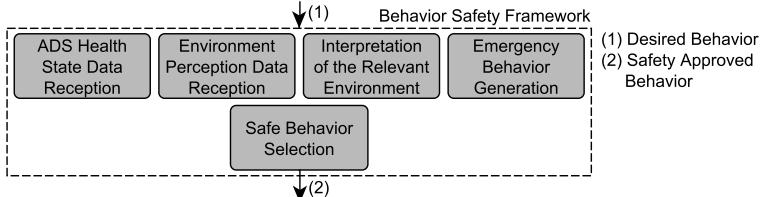


Figure 2: Generic functional architecture of the BSF.

The lefthand submodule in Figure 2 receives information about the capabilities of the automated driving system based on the status and health data of the ADS submodules. These capabilities enable an evaluation of the safety of the desired behavior. The environment perception and the environment interpretation functions are required for two tasks. On the one hand, they need to verify whether the desired behavior is safely compatible with the environment or whether e.g. collisions may occur. On the other hand, depending on the kind of emergency strategy, they might be required for planning a safe emergency behavior, which is done by the righthand submodule in Figure 2. The last one is the safe behavior selection, which chooses either the originally planned desired behavior or the emergency behavior to be sent to the motion controller or directly to the relevant actuators of the vehicle. This decision is based on the results of the other submodules and thus always leads to a safety approved behavior output of BSF. The presented architecture only specifies the required functions represented by submodules. In order to keep the BSF generic, we do not define data flow or interfaces between them.

5 Evaluation and Discussion

We developed our architecture of a BSF based on the requirements given in Section 4.2. For verification we apply two approaches. In Subsection 5.1, we formally compare the capabilities of our generic architecture with the specified requirements. In Subsection 5.2, we use two different ADS configurations for verification. Once again, we compare the requirements presented in Section 4.2 with the capabilities of these ADS configurations.

5.1 Formal Verification

For formal verification, the capabilities of the generic architecture of our BSF are compared to the requirements. Our architecture includes the emergency behavior generation submodule. This module generates the emergency behavior to maintain the safe state of the automated vehicle. Thus, our architecture satisfies Requirement 1. Requirement 2

requires a sufficiently fast safety response of the BSF in case of unsafe desired behavior. A formal verification of this requirement is not possible, since it is dependent on the specific implementation of the BSF. It will be verified in practical testing.

To monitor the health of the ADS, our architecture provides the ADS health state monitoring submodule. This submodule is used to determine the capabilities of the vehicle and compare them with the requirements for the driving mission. Requirement 3 is therefore fulfilled. We also added the submodules for environment perception data reception and interpretation of the relevant environment to satisfy Requirement 4.

Our modular architecture of the BSF allows a functional separation of the submodules, so that the desired behavior of different ADS configurations can be monitored with our framework. Defined interfaces to the relevant vehicle actuators enable sending commands corresponding to the desired behavior and thus, Requirement 5 is fulfilled. All submodules of the BSF determine their own health status, so that the health status of the BSF can be aggregated. As a result, our generic architecture also fulfills Requirement 6. The modularity of the BSF enables individual testability of the submodules. The functional separation between the normal operation behavior planner and our BSF allows the BSF to be tested independently of the ADS. Our BSF thus also meets Requirement 7.

5.2 Application

To demonstrate the technical implementation of the BSF, we introduce two examples. Section 5.2.1 illustrates the example of trajectory monitoring in the context of an AI-driven ADS. As another example, we use a fallback system for an automated vehicle in Section 5.2.2. Both systems are used in an automated vehicle equipped with an ADS that is designed for level 4 according to SAE J3016, as presented in Section 4.1.

5.2.1 Safety Check Module for Monitoring AI Planned Trajectories

The so called Safety Check (SC) module is developed in the scope of the project PRORETA 5 [21]. As Nascimento et al. [22] point out in their study, besides their great potential in various fields of automated driving, AI-approaches also pose risks regarding safety. Safeguarding the results of these AI algorithms and thus the vehicle behavior is the motivation of the SC concept. Consequently, the SC only uses conventional approaches without AI in order not to have the same AI-caused safety issues as the modules of the normal operation behavior planner. Since the considered ODD includes a speed limit of 30 km/h, braking to standstill is mostly preferable to evasion maneuvers in safety critical situations. Thus, the approach to reach a minimal risk condition is to decelerate to standstill along the path of the last safe trajectory. This emergency strategy does not require additional environmental sensors, so no extra hardware is needed for the SC concept.

In the ADS, the SC module is placed between the normal operation behavior planner and the motion controller. That means that the desired behavior of the normal operation behavior planner, which is represented by a desired trajectory, is first checked by the SC module before being sent to the motion controller if no unsafe condition is identified.

As shown in Figure 3, the SC module contains the submodules for environment verification & trajectory safety check and for the system health check. The SC has access to the same sensor data as all other modules in the ADS but uses diversitary approaches of interpreting them. It is also able to detect sensor degradations for environmental or

vehicle dynamic sensors. The safety of the desired trajectory is checked by verification of the environmental perception of the normal operation behavior planner and by looking for collision-critical objects in and around the driving corridor. The system health check submodule observes all functional modules in the ADS as well as all sensors that sense the environment or the dynamic behavior of the vehicle. If any of them fails, the state of the ADS is considered to be unsafe. Both submodules in the SC are sending out Boolean safety flags to the trajectory selection. Based on their values, the trajectory selector chooses either the desired trajectory from the normal operation behavior planner or an emergency trajectory instead, which is provided by the righthand submodule in Figure 3.

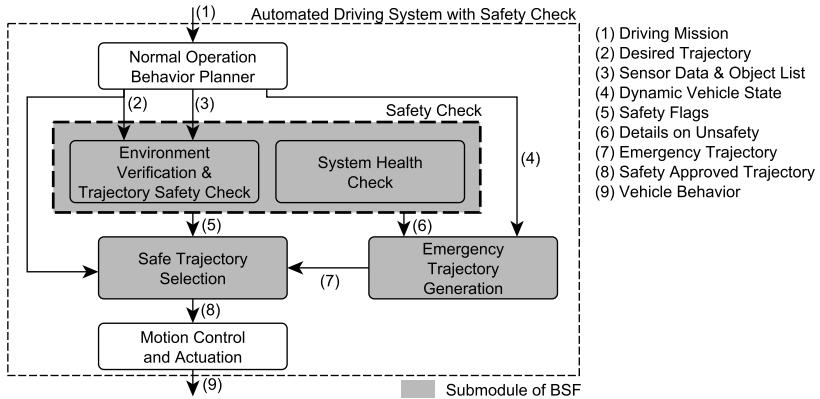


Figure 3: Functional sketch of the Safety Check module.

Comparing this application to the generic BSF, all of the submodules defined in Figure 2 are covered. The ADS health state data reception of the generic BSF corresponds to the system health check. The emergency behavior generation and the safe behavior selection correspond to the emergency trajectory generation and the safe trajectory selection, respectively. The environment perception of the generic BSF is not separately shown in Figure 3, because the perception system of the normal operation behavior planner is used. The environment interpretation of the generic BSF is implemented as an environment verification in the SC, as described above.

Regarding the requirements defined in Section 4.2, Requirements 1 and 3 are met by the ADS with SC. Degraded capabilities of the vehicle that are not sufficient for safe operation are identified by the submodules inside the safety check box in Figure 3. In case of unsafety, an emergency trajectory is immediately generated and sent to the motion controller to transition the vehicle to standstill. This also happens in case of collision-critical objects, that are detected by the environment verification and trajectory safety check. Requirement 4 is thus met. Since the environmental perception and interpretation is reduced to a verification task along the desired driving corridor and the emergency trajectory generation only adapts the speed profile along the path of the desired trajectory, computation time related issues are not expected. This satisfies Requirement 2.

The emergency trajectory uses the same format as the desired trajectory from the normal operation behavior planning, so the motion controller can process both kinds of

trajectories in the same way. Therefore, indirect access to the relevant vehicle actuators is given and the ADS with SC also meets Requirement 5. In case of SC breakdown, the safe trajectory selection gets no information about the safety of the desired trajectory and thus does not forward any trajectory to the motion control. If the motion control has to wait too long for a new trajectory, it initiates an emergency stop by itself and thereby covers Requirement 6. However, this is not supposed to happen due to reliable design of the SC module. The ADS with the SC is designed in a modular way using clearly defined interfaces. This enables testing individual modules and thus fulfills Requirement 7.

The presented SC architecture is easy to be implemented and promising for driving at low speed. For higher velocities, the simple emergency strategy of the SC concept is not reasonable anymore. Then, the planning of collision-free emergency paths to standstill that differ from the originally planned path is more important. A safety concept that also considers higher velocities than the SC module is presented in the following section.

5.2.2 Emergency Stop System Safe Halt

For a second application, we use the emergency stopping system Safe Halt for an automated vehicle [15]. The functional architecture is presented in Figure 4. This emergency stop system is engaged when the capabilities of the automated vehicle are no longer sufficient for the safe performance of its driving mission. The emergency stop system provides a minimal risk maneuver that leads the automated vehicle to the minimal risk condition. The minimal risk maneuver is monitored for collision objects by means of an independent environment perception system.

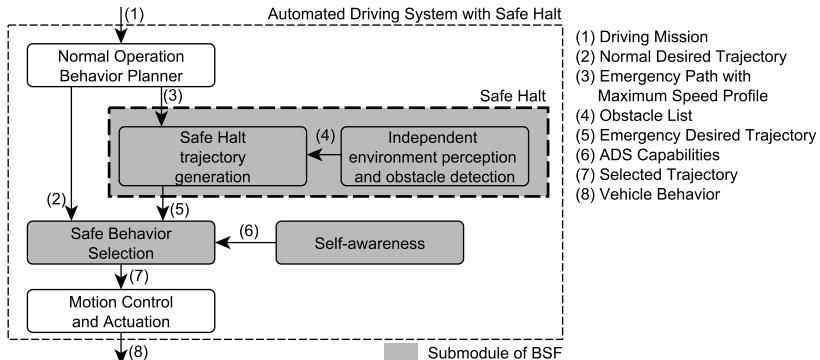


Figure 4: Functional sketch of the ADS with *Safe Halt* emergency stop application.

This ADS configuration includes all submodules of the generic BSF that we presented in Chapter 4. If the capabilities of this ADS with Safe Halt are compared with the requirements in Section 4.2, this ADS configuration meets Requirement 1, because the emergency stop system with the independent environment perception is able to transition the vehicle to a minimal risk condition even with severe degradation of the normal operation behavior planner.

The demonstrated ADS configuration with Safe Halt meets Requirement 2 in a prototype implementation. By providing the normal desired trajectory and the emergency

trajectory in parallel, switching between both in the behavior selection is performed with minimal latency. The aggregation of the vehicle capabilities and the comparison with the requirements for the driving mission is also performed sufficiently fast in a prototypical implementation to enable a suitable desired behavior response.

Requirement 3 is satisfied by self-awareness determining vehicle capabilities and comparing them to driving mission requirements. If these capabilities are not sufficient for the driving mission, the behavior selection switches from the normal desired trajectory to the emergency trajectory and the vehicle performs an emergency stop behavior. The Safe Halt functionality provides collision free trajectories based on the separated environment perception. The system therefore fulfills Requirement 4.

The modular architecture of Safe Halt is integrated into the generic ADS architecture in an interface-compliant manner. The interfaces between the normal operation behavior planner, the safe behavior selector and the Safe Halt emergency stop are defined. This definition allows motion control and actuation to be used for the emergency stop maneuver as well. The presented ADS configuration thus fulfills Requirement 5.

The emergency stop system Safe Halt reports its health status to the ADS self-awareness. The system therefore also meets Requirement 6. Due to the modular architecture of Safe Halt, the modules are testable separately. The system thus also fulfills requirement 7.

Overall, this example shows that the generic architecture of the BSF can also be inserted into this ADS configuration and still meet all the requirements for a BSF. The architecture of the BSF enables the safeguarding of the desired behavior of automated vehicles even in the case of complete failures of the normal operation behavior planner.

6 Conclusion

In this paper, we present a generic architecture for a BSF for an automated vehicle with a level 4 ADS according to SAE J3016. We demonstrate the application of the architecture in two different ADS configurations. The verification of these applications indicate that the presented architecture is also valid for other ADS configurations. Our generic architecture of a BSF can be used to safeguard complex, possibly AI-supported, desired behavior planners as well as emergency stop systems to transition an automated vehicle to a minimal risk condition when needed. We see a further need for research in the reliable determination of vehicle capabilities in order to identify unsafe trajectories by an ADS internal evaluation. Our future work aims at verification and validation of the presented architecture. This will be done by experimental tests using silent testing methods in the automated vehicle under real conditions.

Acknowledgments

We kindly thank Continental AG for their great cooperation within PRORETA 5, a joint research project of TU Darmstadt, University of Bremen, TU Iași and Continental. This research is also accomplished within the project “UNICARagil” (FKZ 16EMO0286). We acknowledge the financial support for the project by the Federal Ministry of Education and Research of Germany (BMBF).

References

- [1] National Transportation Safety Board, “Collision Between a Car Operating With Automated Vehicle Control Systems and a Tractor-Semitrailer Truck Near Williston, Florida, May 7, 2016,” Highway Accident Report NTSB HAR-17/02, National Transportation Safety Board, 2017.
- [2] National Transportation Safety Board, “Collision Between Vehicle Controlled by Developmental Automated Driving System and Pedestrian, Tempe, Arizona, March 18, 2018,” Highway Accident Report NTSB HAR-19/03, National Transportation Safety Board, 2019.
- [3] Society of Automotive Engineers, “SAE J3016: Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems.,” 2021.
- [4] S. Shalev-Shwartz, S. Shammah, and A. Shashua, “On a Formal Model of Safe and Scalable Self-driving Cars,” *arXiv:1708.06374 [cs, stat]*, Oct. 2018. arXiv: 1708.06374.
- [5] D. Nistér, H.-L. Lee, J. Ng, and Y. Wang, “The Safety Force Field,” white Paper, Nvidia, 2019.
- [6] R. De Iaco, S. L. Smith, and K. Czarnecki, “Universally Safe Swerve Manoeuvres for Autonomous Driving,” *arXiv:2001.11159 [cs]*, Jan. 2020. arXiv: 2001.11159.
- [7] C. B. S. T. Molina, J. R. d. Almeida, L. F. Vismari, R. I. R. González, J. K. Naufal, and J. Camargo, “Assuring Fully Autonomous Vehicles Safety by Design: The Autonomous Vehicle Control (AVC) Module Strategy,” in *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pp. 16–21, June 2017. ISSN: 2325-6664.
- [8] T. Stolte, A. Reschka, G. Bagschik, and M. Maurer, “Towards Automated Driving: Unmanned Protective Vehicle for Highway Hard Shoulder Road Works,” in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, (Gran Canaria, Spain), pp. 672–677, IEEE, Sept. 2015.
- [9] C. Pek, S. Manzinger, M. Koschi, and M. Althoff, “Using online verification to prevent autonomous vehicles from causing accidents,” *Nature Machine Intelligence*, vol. 2, pp. 518–528, Sept. 2020.
- [10] T. Stahl, M. Eicher, J. Betz, and F. Diermeyer, “Online Verification Concept for Autonomous Vehicles – Illustrative Study for a Trajectory Planning Module,” in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, p. 7, 2020.
- [11] J. Funke, M. Brown, S. M. Erlien, and J. C. Gerdes, “Collision Avoidance and Stabilization for Autonomous Vehicles in Emergency Scenarios,” *IEEE Transactions on Control Systems Technology*, vol. 25, pp. 1204–1216, July 2017.

- [12] J. Hilgert, K. Hirsch, T. Bertram, and M. Hiller, “Emergency path planning for autonomous vehicles using elastic band theory,” in *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)*, vol. 2, (Kobe, Japan), pp. 1390–1395, IEEE, 2003.
- [13] A. Mehmed, M. Antlanger, W. Steiner, and S. Punnekkat, “Forecast Horizon for Automated Safety Actions in Automated Driving Systems,” in *Computer Safety, Reliability, and Security* (A. Romanovsky, E. Troubitsyna, and F. Bitsch, eds.), vol. 11698, pp. 113–127, Cham: Springer International Publishing, 2019. Series Title: Lecture Notes in Computer Science.
- [14] A. Reschka, *Fertigkeiten- und Fähigkeitengraphen als Grundlage des sicheren Betriebs von automatisierten Fahrzeugen im öffentlichen Straßenverkehr in städtischer Umgebung*. PhD Thesis, Technische Universität Braunschweig, Braunschweig, 2017.
- [15] S. Ackermann and H. Winner, “Systemarchitektur und Fahrmanöver zum sicheren Anhalten modularer automatisierter Fahrzeuge..,” in *13. Workshop Fahrerassistenzsysteme und automatisiertes Fahren*, (Walting), 2020.
- [16] ISO, “ISO/TR 4804:2020 - Road vehicles — Safety and cybersecurity for automated driving systems — Design, verification and validation,” 2020.
- [17] ISO, “ISO/PAS 21448 -Road vehicles - Safety of the intended functionality,” 2019.
- [18] P. Blokland and G. Reniers, “Measuring (un)safety. A broad understanding and definition of safety, allowing for instant measuring of unsafety,” *Chemical Engineering Transactions*, vol. 77, pp. 253–258, Sept. 2019.
- [19] Dictionary.com, “Definition of unsafety.” <https://www.dictionary.com/browse/unsafety>, Sept. 2021.
- [20] P. Junietz, *Microscopic and Macroscopic Risk Metrics for the Safety Validation of Automated Driving*. PhD Thesis, Technische Universität Darmstadt, Darmstadt, 2019.
- [21] TU Darmstadt, “PRORETA 5,” 2021. <https://www.proreta.tu-darmstadt.de/proreta/index.en.jsp>, visited 2021-11-05.
- [22] A. M. Nascimento, L. F. Vismari, C. B. S. T. Molina, P. S. Cugnasca, J. B. Camargo, J. R. d. Almeida, R. Inam, E. Fersman, M. V. Marquezini, and A. Y. Hata, “A Systematic Literature Review About the Impact of Artificial Intelligence on Autonomous Vehicle Safety,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 4928–4946, 2020.

Aktualitätsverifikation semantischer HD-Karten für das urbane automatisierte Fahren

Jan-Hendrik Pauls* und Christoph Stiller†

Zusammenfassung: In diesem Beitrag wird ein Gesamtsystem zur Aktualitätsverifikation hochauflösender, semantischer, aber dennoch kompakter Karten vorgestellt. Es basiert auf einer geeigneten semantisch-parametrischen Kartendarstellung, die automatisiert erstellt und aktualisiert werden kann. Durch die Kombination semantischer Detektionen und Lidar-Tiefenmessungen kann nicht nur eine hochgenaue 6D-Lokalisierung realisiert werden, die robust gegenüber Änderungen ist. Es wird hiermit auch eine evidenzbasierte, ternäre Aktualitätsverifikation ermöglicht, die aus den Messdaten eines einzelnen Fahrzeugs Änderungen detektieren oder Karteninhalte freigeben kann. Schließlich wird erläutert, wie man Verifikationsergebnisse von physischen Elementen, wie Verkehrsschildern, auf abstrakte Karteninhalte, wie Tempolimits, übertragen kann.

Schlüsselwörter: Aktualitätsverifikation, Semantische HD-Karten, Lokalisierung

1 Einleitung

Komfortables und sicheres Fahren erfordert ein umfassendes Verständnis der Umgebung. Automatisierte Fahrfunktionen müssen dieses entweder zur Laufzeit aus Sensordaten herleiten oder auf in Karten gespeicherte Informationen zurückgreifen. Da eine Inferenz zur Laufzeit mit aktuellen Methoden und Sensoren nicht mit hinreichender Qualität möglich ist, werden hochauflösende, semantische Karten auf absehbare Zeit unabdingbar sein.

Die Verwendung von Karten bringt dafür ein anderes Problem mit sich: Die Aktualitätsverifikation der Karteninhalte. So könnte sich beispielsweise ein in der Karte annotiertes Tempolimit seit der Erstellung oder letzten Aktualisierung der Karte geändert haben. Um also Karteninhalte sicher nutzen zu können, muss ihre Aktualität mithilfe der aktuellen Sensordaten geprüft und dies an die Fahrfunktionen weitergegeben werden. Durch Verdeckungen sind aber nicht immer alle Karteninhalte sichtbar, was in der Verifikation berücksichtigt werden muss.

Dieser Beitrag stellt ein Gesamtsystem vor, das eine passende, automatisiert erzeugbare Kartendarstellung umfasst, die Grundlage einer robusten Lokalisierung sowie einer evidenztheoretischen Verifikation der Karteninhalte ist. Schließlich wird vorgeschlagen, wie mithilfe einer relationalen Annotation Änderungen der physischen Kartenelemente auf abstrakttere Karteninhalte übertragen werden können.

* Jan-Hendrik Pauls leitet die Forschungsgruppe *Kartierung und Lokalisierung* am Institut für Mess- und Regelungstechnik (MRT) des Karlsruher Instituts für Technologie (KIT), pauls@kit.edu.

† Prof. Dr.-Ing. Christoph Stiller leitet das Institut für Mess- und Regelungstechnik (MRT) des Karlsruher Instituts für Technologie (KIT), stiller@kit.edu.

2 Verwandte Arbeiten

Verwandte Arbeiten lassen sich anhand der einzelnen Teile des Gesamtsystems gliedern, der hochauflösenden Kartierung, der robusten Lokalisierung und der Aktualitätsverifikation. Für die Verarbeitung der erkannten Änderungen innerhalb einer Karte sind den Autoren keine verwandten wissenschaftlichen Publikationen bekannt.

2.1 Hochauflösende, semantische Kartierung

Hochauflösende Karten beschränkten sich anfangs auf sensorspezifische Kartenmerkmale [1, 2]. Die Extraktion vektorieller semantischer Merkmale, die sowohl kompakt speicherbar als auch sensorübergreifend wahrnehmbar sind, war lange Zeit nur mit manuellem Aufwand möglich.

Arbeiten zur Automatisierung konzentrierten sich zuerst auf Straßenmarkierungen [3] sowie die Extraktion semantischer Merkmale mittels ungelerner Verfahren [4]. Erst das Aufkommen leistungsstarker maschineller Lernverfahren, insbesondere tiefer neuronaler Netze, und die kostengünstigere Verfügbarkeit hochauflösender Lidar-Sensoren erlaubt die Detektion diverser Merkmale in hoher Genauigkeit.

Viele Arbeiten, die semantische Objekte kartieren, nutzen eine möglichst generische, d.h. klassenübergreifende Darstellung der Objekte, wie Quadriken [5]. Während man hierdurch keinerlei Modellwissen aufprägt, verliert man gleichzeitig an Genauigkeit. Im Gegensatz dazu nutzt das in diesem Beitrag vorgestellte Verfahren, wie bereits in [6] gezeigt, die semantische Klassifikation, um eine an die Klasse angepasste, kompakte, parametrische Darstellungsform zu bestimmen, deren Parameter aus Bild- und Lidarinformationen geschätzt werden. Dies erlaubt die automatische Kartierung mit einer Genauigkeit in derselben Größenordnung wie bislang nur manuelle Annotationsverfahren.

2.2 Robuste Lokalisierung

Wie auch bei der Kartierung waren robuste Lösungen für das Lokalisierungsproblem lange Zeit sensorspezifisch: Nur innerhalb derselben Sensordomäne konnten so dicht eindeutige Merkmale bestimmt werden, dass auch noch unter Änderungen der Umgebungsbedingungen oder bei veralteten Karteninformationen eine Lokalisierung möglich war [2].

Weniger eindeutige Merkmale erforderten Verfahren, die mehrere Hypothesen parallel verfolgen und auswerten können [7, 8]. Eine Alternative ist die Überprüfung der Eindeutigkeit von Karteninformation bei der Kartenerstellung [9].

In diesem Beitrag wird ein auf der hochgenauen semantischen Kartendarstellung aufbauendes Verfahren vorgestellt, das eine selbsteinschätzende Lokalisierung zur Laufzeit ermöglicht. Die Eindeutigkeit kann hierbei über die sehr hohe Genauigkeit der Messungen und deren intelligente Auswertung erreicht werden. Im Gegensatz zu rekursiven Filtern [8] umgeht das Verfahren durch die Verwendung nur einer (zeitlich begrenzten) Messung die Gefahr, bei überlappenden, aber teilkonsistenten Kartenänderungen zu einer falschen Lösung zu “driften”.

2.3 Aktualitätsverifikation

Das Problem der Aktualitätsverifikation ist eng mit dem Problem der Lokalisierung verbunden. Meist kann ein Abgleich zwischen Karten- und Sensordaten nur erfolgen, wenn die Pose relativ zur zu verifizierenden Karte bekannt ist. Eine Ausnahme ist ein unter anderem von den Autoren vorgestellter Ansatz, der eine Merkmalsreduktion durchführt, die robust gegenüber Lokalisierungsfehlern ist [10].

Um die starke Kopplung zu reduzieren, kann man Änderungen mittels Flottendaten detektieren [11]. Zeitvariante Lokalisierungsfehler können sich so ausgleichen. Um nicht nur allmählich Änderungen zu detektieren, sondern Karteninhalte instantan zu verifizieren, ist man aber allein auf die Sensordaten des Ego-Fahrzeugs beschränkt.

Eine andere Lösungsmöglichkeit ist, Lokalisierung und Änderungsdetektion gekoppelt zu lösen [8, 12]. Angelehnt an diese Idee wird in diesem Beitrag vorgeschlagen, die Aktualitätsverifikation basierend auf einer der Lokalisierung ähnlichen Datenassoziation durchzuführen. Hierbei ist aber nicht die Pose, sondern die Wahrscheinlichkeit, ein Kartenmerkmal beobachtet zu haben, das Ergebnis.

3 Semantisch-parametrische Kartendarstellung

Grundlage der Aktualitätsverifikation ist eine passende Kartendarstellung. Automatisierte Fahrfunktionen erfordern in der Regel semantisch-relationale, physische Merkmale wie Straßen- und Fahrstreifenberandungen, (Verkehrs-)schilder, Ampeln¹ und die hierdurch ableitbaren, abstrakten Inhalte wie Fahrstreifen, Verkehrsregeln sowie einen topologischen Zusammenhang zur Routenplanung. Beispielhaft sei hier eine Lanelet2-Karte [13]. Karteninhalte müssen kompakt gespeichert werden, um Aktualisierungen über mobile Datenübertragungswege zu ermöglichen. Während Lanelet2 dies nicht explizit erfordert, wird dieser Beitrag schließlich zeigen, dass eine parametrische 3D-Darstellung aller Schilder und Ampeln für verifizierbare Karten hilfreich ist und ebenbürtig zu parametrischen Fahrstreifenberandungen zum Standard werden sollte.

Da Schilder, Ampeln und Masten mit üblicher Sensorik aus wesentlich größerer Entfernung detektiert werden können als Straßenmarkierungen, eignen sie sich auch besser zur hochgenauen Lokalisierung, die notwendige Voraussetzung der Aktualitätsverifikation ist. Daher konzentriert sich dieser Beitrag im Folgenden beispielhaft auf diese drei Klassen.

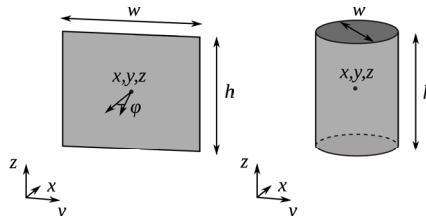


Abbildung 1: Visualisierung der parametrischen Darstellungen für Schilder (links) sowie Ampeln und Masten (rechts) wie bereits in [6] veröffentlicht.

¹Im Folgenden wird der geläufigere Begriff *Ampel* für den Fachterminus *Signalgeber* verwendet.

Eine fast vollständig automatisierte Erstellung einer solch semantisch-parametrischen 3D-Karte wurde bereits vorgestellt [6]. Kurz umrissen können semantische Detektionen in Kamerabildern mit Tiefeninformationen aus Lidar-Sensoren kombiniert werden. Eine Kernidee ist hierbei, dass die semantische Klasse der Detektion hierbei über die parametrische Darstellung der Karteninhalte entscheidet: Schilder werden als um die Hochachse rotierte Rechtecke repräsentiert während Ampeln und Masten durch Zylinder dargestellt werden (siehe Abbildung 1). Die Darstellungen sind hierbei kompakte parametrische Annäherungen, die gleichzeitig eine treffende Rückprojektion in ein Kamerabild, eine gute Datenassoziation und eine zuverlässige Schätzung der Darstellung ermöglichen.

Das Messmodell entnimmt den Sichtstrahl zum Mittelpunkt sowie Höhe und Breite der Repräsentation aus den Bounding Boxen der Detektionen. Durch Projektion der Lidar-Punktwolke ins Kamerabild können durch die Detektionsmaske relevante Punkte ausgewählt werden. Diese ergeben die Tiefe des Mittelpunkts sowie, im Fall von Schildern, den Rotationswinkel. Wie in [6] gezeigt erreicht die so automatisierte Kartierung hierbei – abgesehen von Fehldetections des neuronalen Netzes – mit ca. 1° Orientierungs- bzw. weniger als 10 cm Translationsfehler annähernd menschliche Annotationsqualität. Ein typisches Kartierungsergebnis ist in Abbildung 2 veranschaulicht.



Abbildung 2: Beispielhaftes Kartierungsergebnis an einer urbanen Kreuzung in Karlsruhe. Masten sind in Hellblau, Ampeln in Grün und Schilder in Gelb gezeigt.

4 Änderungsrobuste Lokalisierung

Neben einer passenden Karte benötigt die Verifikation eine hochgenaue Lokalisierung, die gegenüber Änderungen der Karte robust sein muss. Genau genommen ist die Lokalisierung, d.h. die Bestimmung der Egopose im Koordinatensystem der Karte, für die Verifikation ebenso wichtig wie die Assoziation von Detektionen zu Karteninhalten. Gleichzeitig erlauben hinreichend viele assoziierte Detektionen die Berechnung einer hochgenauen 6D-Pose. In diesem Beitrag wird das Lokalisierungsproblem daher durch das Finden geeigneter Assoziationen gelöst.

Die Lokalisierung in einer möglicherweise veränderten Karte ist hierbei besonders herausfordernd, da die sonst oft geltende Annahme, dass die Kartenmerkmale als gegeben anzunehmen sind, nicht mehr gilt. Diese Herausforderung kann dennoch gemeistert werden, da die zuvor vorgestellte Kartenrepräsentation durch die Kombination der semantisch-

parametrischen und der geometrischen Dimensionen Detektionen sehr eindeutig werden lässt. Anders gesagt passen Fehldetectionsen nur sehr selten gut zu Kartenelementen.

Um mögliche Assoziationshypothesen zu finden kann man nun einen Assoziationsgraphen $\mathcal{A} = (\mathcal{G}, \mathcal{E})$ aufspannen. Knoten dieses Graphen sind hypothetische Paarungen (l, d) eines Kartenelements $l \in L$ mit einer Detektion $d \in D$. Von allen möglichen Paarungen können durch semantisch-parametrische Kompatibilität bereits viele Kombinationen ausgeschlossen werden, beispielsweise weil die Höhe und Breite eines Schilds zu verschieden ist. Anschließend werden alle verbleibenden Paarungen auf paarweise geometrische Kompatibilität geprüft. Dies bedeutet, dass zwei Paarungen $(l_{i_1}, d_{j_1}), (l_{i_2}, d_{j_2})$ in \mathcal{A} genau dann verbunden werden, wenn $\|l_{i_1} - l_{i_2}\| - \|d_{j_1} - d_{j_2}\| < \tau$, wobei τ ein Schwellwert ist.

Eine passende Assoziationshypothese ist damit eine Menge von individuell semantisch-parametrisch und paarweise geometrisch kompatiblen Paarungen von Karteninhalten und Detektionen. Mögliche Assoziationshypothesen θ können durch Umformung der Wahrscheinlichkeit einer Mehrzielmessung verglichen werden [15, S. 423], wobei wir der in [15] eingeführten Notation folgen.

$$f(\theta|D, L) = e^\lambda \prod_{i:\theta(i)=0} (1 - p_D(l_i)) \prod_{i:\theta(i)>0} f(d_{\theta(i)}|l_i) p_D(l_i) \prod_{j:\theta^{-1}(j)=0} \lambda c(d_j) \quad (1)$$

Hierbei bezeichnet $p_D(l_i)$ die Detektionswahrscheinlichkeit eines Kartenelements und $f(d_{\theta(i)}|l_i)$ die Ähnlichkeit der zugeordneten Paarung. Die Parameter λ und $c(d_j)$ charakterisieren Fehldetectionsen. Typische Werte zeigen, dass die Wahrscheinlichkeit einer Assoziationshypothese mit jeder Paarung um einen Faktor von ca. 10^3 wächst. Die wahrscheinlichsten Hypothesen sind also jene, die die meisten Paarungen umfassen. Dies wird auch als *Maximum Consensus*-Datenassoziation bezeichnet.

In \mathcal{A} sind Assoziationshypothesen Cliques im Sinne der Graphentheorie. Es gilt somit, die größten Cliques in \mathcal{A} zu finden. Während die Bestimmung der größten Clique in einem Graph ein bekanntes Problem der Informatik ist, sind wir an allen Cliques $\Theta^p(\hat{\theta}) \subseteq \Theta$ interessiert, die um höchstens p kleiner sind als die größte Clique $\hat{\theta}$. Eine Anpassung des Bron-Kerbosch-Algorithmus [14] ermöglicht eine sehr effiziente Suche nach $\Theta^p(\hat{\theta})$.

In urbanen Szenarien ergeben sich hierbei häufig Cliques, die größtenteils übereinstimmen, aber in wenigen Paarungen voneinander abweichen. Für eine reine Lokalisierung wäre das Ergebnis, d.h. die resultierende Pose, quasi identisch. Für eine Änderungsdetection ist jedoch jede individuelle Paarung (l_i, d_j) interessant, da potentiell d_j die Existenz von l_i bestätigt. Die Wahrscheinlichkeit jeder Paarung ergibt sich durch Summierung über alle relevanten Assoziationshypothesen:

$$f((l_i, d_j)|D, L) = \sum_{\theta \in \Theta: \theta(i)=j} f(\theta|D, L) \approx \sum_{\theta \in \Theta^p(\hat{\theta}): \theta(i)=j} f(\theta|D, L) \quad (2)$$

Somit können nicht nur ausschließlich hinreichend sichere Paarungen für die Po-senschätzung verwendet werden. Es können auch Situationen mit mehrdeutigen Lokalisierungslösungen aktiv erkannt werden, wodurch eine Fehllokalisierung vermieden werden kann. Hierbei ist anzumerken, dass Systeme, die eine möglicherweise ausbleibende Kartenfreigabe erfordern, ebenso eine fehlende Lokalisierung tolerieren können sollten.

Neben einer hochgenauen und robusten 6D-Pose \mathcal{P} erhält man also auch für jede Landmarke eine Wahrscheinlichkeit, dass diese durch eine Detektion bestätigt werden konnte. Beide Ergebnisse werden im folgenden Abschnitt für die Sichtbarkeitsprüfung bzw. für die Aktualitätsbestätigung der Landmarken benötigt.

5 Ternäre, evidenzbasierte Verifikation

Die Verifikation von Karten, wie sie in diesem Beitrag vorgestellt wird, unterscheidet sich in zwei Details von vorherigen Arbeiten [8, 12]. Sie setzt nicht auf Flottendaten [11], sondern allein auf die Beobachtungen eines einzelnen Fahrzeugs. Außerdem konzentriert sie sich auf mehrschichtige, semantische HD-Karten. Während reine Lokalisierungskarten allmählich aktualisiert werden können, da im Regelfall trotz einer Änderung hinreichend viele unveränderte Merkmale zur hochgenauen Lokalisierung existieren, erfordern semantische HD-Karten eine unmittelbare Verifikation.

Um ein Tempolimit, das nur mit einem Schild signalisiert wird, zu verifizieren, muss man genau dieses Schild lokalisieren und auf Existenz prüfen. Da nur die Messdaten eines einzigen Fahrzeugs zur Verfügung stehen, kann es aber auch zu Verdeckungen kommen. Dies motiviert eine ternäre Verifikation: Jedes physische Kartenelement kann entweder *aktuell*, *veraltet* oder *nicht beobachtbar* sein.

5.1 Evidenzbasierte Messmodelle

Für jedes Kartenelement l_i können aber nur für die beiden Hypothesen *aktuell* und *veraltet* Evidenzen gesammelt werden. Hierfür werden zwei beispielhafte Evidenzquellen vorgeschlagen, die Datenassoziation und eine Sichtbarkeitsprüfung. Die verbleibende Evidenzmasse entspricht dann der Nicht-Beobachtbarkeit als dritte Hypothese.

Wenn l_i in der Datenassoziation verwendet wurde, dient dies als Evidenz, dass ein solches Kartenelement in geometrischer, semantischer und parametrischer Form sehr ähnlich beobachtet wurde. Die Formeln für die Evidenzmassen aus der Datenassoziation (DA) lautet daher

$$\begin{aligned} m^{DA}(l_i \text{ veraltet}) &= 0 \\ m^{DA}(l_i \text{ aktuell}) &= \lambda \sim f((l_i, d_j) | D, L) \\ m^{DA}(\Omega) &= 1 - \lambda. \end{aligned}$$

Hierbei ist $f((l_i, d_j) | D, L)$ die bereits eingeführte Assoziationswahrscheinlichkeit und Ω die Gesamtheit der Hypothesen. Bei einer Verdeckung kann l_i nicht in der Datenassoziation auftauchen. Gleichzeitig ist dies aber keine Evidenz für eine Veränderung.

Um Evidenz für Änderungen zuzuweisen, schlagen wir daher eine Sichtbarkeitsprüfung (SP) vor, die die zuvor bestimmte, hochgenaue 6D-Pose \mathcal{P} verwendet. Durch die Transformation ins Sensorkoordinatensystem kann mittels Lidar gemessen werden, wie viele Strahlen R_\odot durch l gingen, und dies in Verhältnis zu allen auf l_i fallenden Strahlen R gesetzt werden. Die Evidenzmassen ergeben sich somit durch

$$\begin{aligned} m^{SP}(l_i \text{ veraltet}) &= \sigma \sim \frac{|R_\odot|}{|R|} \\ m^{SP}(l_i \text{ aktuell}) &= 0 \\ m^{SP}(\Omega) &= 1 - \sigma. \end{aligned}$$

Im Gegensatz zu vorherigen Arbeiten kann somit nicht nur eine Veränderung detektiert, sondern sowohl aktiv die Aktualität gemessen als auch das Ausbleiben von Informationen für beide Hypothesen modelliert werden.

5.2 Aggregation und Ergebnisse

Mit den soeben vorgestellten Messmodellen kann nun in jedem Zeitschritt Evidenz gesammelt und über die Zeit aggregiert werden. Die Übersetzung der Evidenzen in eine passende Ausgabe ist ein Klassifikationsproblem, das im einfachsten Fall die Hypothese mit der größten Evidenzmasse ausgibt.

Für die Evaluation wurden reale Karteninhalte zufällig verändert. Durch diese simulierte Änderung ist somit das Sollergebnis bekannt. Erste Ergebnisse zeigen eine Änderungsdetektionsrate (Recall) von ca. 95% sowie eine Genauigkeit (Precision) von 98%.

6 Verarbeitung der Änderungsergebnisse

Wie im vorhergehenden Abschnitt beschrieben können für jedes physische Kartenelement l Evidenzen ermittelt und aggregiert werden. Für viele Fahrfunktionen ist aber nicht die physische Kartenschicht, sondern abstraktere Merkmale, wie Tempolimits, Ampelzuordnungen oder Abbiegevorschriften interessant und müssen daher ebenfalls verifiziert werden.

Eine direkte Verifikation dieser Merkmale, beispielsweise durch das Verhalten anderer Verkehrsteilnehmer, ist durchaus denkbar, erfordert aber entsprechende Befahrungsstatistiken oder Flottendaten. Damit ein einzelnes Fahrzeug dennoch diese abstrakten Merkmale verifizieren kann, wurde im Kartenformat Lanelet2 [13] neben einer direkten Annotation solch abstrakter Inhalte auch die Möglichkeit vorgesehen, die abstrakte Eigenschaft mit den Elementen der physischen Kartenschicht in Beziehung zu setzen.

Eine solche Relation ist in Abbildung 3 abgebildet. Wird nun also festgestellt, dass das Verkehrsschild, das das Tempolimit anordnet, nicht als aktuell bestätigt werden kann, kann dieses Ergebnis direkt auf abstraktere, abgeleitete bzw. betroffene Karteninhalte propagiert werden. Fahrfunktionen erhalten somit für alle Schichten und Merkmale einer semantischen HD-Karte eine Aktualitätsinformation.

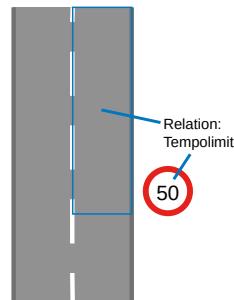


Abbildung 3: Beispielhafte Tempolimit-Relation, die das anordnende Verkehrsschild mit dem betroffenen Lanelet in Verbindung setzt.

7 Fazit

In diesem Beitrag wurde ein Gesamtsystem vorgestellt, das basierend auf einer passenden, automatisiert erzeugbaren Kartendarstellung eine hochgenaue und dennoch gegenüber Änderungen robuste Lokalisierung erlaubt. Die Lokalisierung ermöglicht hierbei Rückschlüsse auf die detektierten und assoziierten Karteninhalte. Zusammen mit einer Sichtbarkeitsprüfung können diese für eine ternäre Aktualitätsverifikation der Karteninhalte verwendet werden. Durch relationale Annotation von abgeleiteten Karteninhalten, wie Tempolimits, kann das Verifikationsergebnis von der physischen auf höhere Kartenebenen propagiert werden.

Das System zeigt erstmals, wie eine hochgenaue und dennoch robuste Lokalisierung in einer kompakten, semantischen HD-Karte funktionieren kann. Darüber hinaus erlaubt das System erstmals die Aktualitätsverifikation mit den Sensordaten einer einzigen Befahrung und kann so Karteninhalte für die Verwendung durch automatisierte Fahrfunktionen freigeben bzw. die Karte absichern. Dies gilt hierbei nicht nur für physische Elemente, sondern für alle Ebenen einer Karte.

Literatur

- [1] J. Levinson und S. Thrun, *Robust vehicle localization in urban environments using probabilistic maps*, 2010 International Conference on Robotics and Automation (ICRA), IEEE, S. 4372–4378, 2010.
- [2] H. Lategahn und C. Stiller, *Vision-only localization*, IEEE Transactions on Intelligent Transportation Systems, IEEE, 15. Jg., Nr. 3, S. 1246–1257, 2014.
- [3] M. Schreiber, F. Poggenhans und C. Stiller, *Detecting symbols on road surface for mapping and localization using OCR*, 17th International Conference on Intelligent Transportation Systems (ITSC), IEEE, S. 597–602, 2014.
- [4] M. Sefati, M. Daum, B. Sondermann, K. D. Kreisköther und A. Kampker, *Improving vehicle localization using semantic and pole-like landmarks*, 2017 Intelligent Vehicles Symposium (IV), IEEE, S. 13–19, 2017.
- [5] L. Nicholson, M. Milford und N. Sünderhauf, *QuadricSLAM: Dual Quadrics From Object Detections as Landmarks in Object-Oriented SLAM*, IEEE Robotics and Automation Letters, IEEE, 4. Jg., Nr. 1, S. 1–8, 2018.
- [6] J.-H. Pauls, B. Schmidt und C. Stiller, *Automatic Mapping of Tailored Landmark Representations for Automated Driving and Map Learning*, 2021 International Conference on Robotics and Automation (ICRA), IEEE, S. 6725–6731, 2021.
- [7] N. Sünderhauf und P. Protzel, *Switchable constraints vs. max-mixture models vs. RRR-a comparison of three approaches to robust pose graph SLAM*, 2013 International Conference on Robotics and Automation (ICRA), IEEE, S. 5198–5203, 2013.
- [8] M. Stübler, *Self-assessing localization and long-term mapping using random finite sets*. Dissertation, Fakultät für Ingenieurwissenschaften, Informatik und Psychologie, Universität Ulm, Ulm, 2018.
- [9] I. Hofstetter, M. Sprunk, F. Schuster, F. Ries und M. Haueis, *On ambiguities in feature-based vehicle localization and their a priori detection in maps*, 2019 Intelligent Vehicles Symposium (IV), IEEE, S. 1192–1198, 2019.
- [10] J.-H. Pauls, T. Strauß, C. Hasberg, M. Lauer und C. Stiller, *HD Map Verification Without Accurate Localization Prior Using Spatio-Semantic 1D Signals*, IEEE, 2020 Intelligent Vehicles Symposium (IV), S. 680–686, 2020.

- [11] D. Pannen, M. Liebner und W. Burgard, *HD Map Change Detection with a Boosted Particle Filter*, 2019 International Conference on Robotics and Automation (ICRA), IEEE, S. 2561–2567, 2019.
- [12] K. Jo, C. Kim und M. Sunwoo, *Simultaneous Localization and Map Change Update for the High Definition Map-Based Autonomous Driving Car*, Sensors 18, No. 9, S. 3145, 2018.
- [13] F. Poggenhans, J.-H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt und M. Mayr, *Lanelet2: A high-definition map framework for the future of automated driving*, 2018 21st International Conference on Intelligent Transportation Systems (IT-SC), IEEE, S. 1672–1679, 2018.
- [14] C. Bron und J. Kerbosch, *Algorithm 457: finding all cliques of an undirected graph*. Commun. ACM 16, 9 (Sept. 1973), S. 575–577, 1973.
- [15] R. Mahler, *Statistical Multisource-Multitarget Information Fusion*, Norwood, MA, USA: Artech House, 2007.

Über das verantwortliche Fahren Autonomer Fahrzeuge

Hans-Peter Schöner*

Zusammenfassung: Dieser Beitrag diskutiert zunächst die wichtigsten Faktoren, die verantwortungsvolles Fahren begründen, und darauf aufbauend einige technische Vorkehrungen, die hilfreich sind, um den Nachweis verantwortungsvollen Verhaltens beim autonomen Fahren zu unterstützen. Es verknüpft die Argumente mit dem Konzept der Taktischen Sicherheit: In Verkehrssituationen frühzeitig und proaktiv handeln, um nicht-beherrschbare Situationen mit möglicherweise hoher Unfallschwere zu vermeiden. Eine wichtige Konsequenz ist eine durchgängige Sicherheitsmetrik, die es ermöglicht, Gefahr als ‘Entfernung’ von einem Schadens-Ereignis zu messen. Verlässliche Kommunikation unter Verkehrsteilnehmern ermöglicht frühzeitiges Erkennen von Bedingungen für mögliche Gefahren. Abschließend wird die Kombination dieser Aspekte zur Bewertung der Fahr-Verantwortung von autonomen Fahrzeugen diskutiert.

Schlüsselwörter: Fahr-Verantwortung, Sicherheitsmetrik, Taktische Sicherheit, Verlässliche Kommunikation.

1 Der verantwortliche Fahrer

Als ‘verantwortliche Person’ wird normalerweise die Person wahrgenommen, die für Fehlverhalten bestraft werden könnte, wenn in ihrem Verantwortungsbereich etwas schief geht. Aber vor der Bestrafung besteht für die Person die Möglichkeit, sich zu verteidigen, indem sie ihr verantwortungsvolles Verhalten einem Richter darlegt. Dazu gehört insbesondere auch die Beantwortung von folgenden Fragen des Richters: „Können Sie erklären, wie dieser Unfall überhaupt passieren konnte?“; „Warum haben Sie diesen Unfall nicht viel früher kommen sehen?“; „Mit welcher Intention und wie haben Sie gehandelt, und warum haben Sie nicht ausreichend schnell und stark genug reagiert, um den Unfall noch zu vermeiden?“. Nur wenn der Verantwortliche auf diese Fragen überzeugend antworten kann, indem er dem Richter entweder sein dem Stand der Technik entsprechendes Verhalten oder die höhere Gewalt der Sachlage darlegt, wird er aus dem Verfahren ohne Schulterspruch oder gar Bestrafung für Fehlverhalten herauskommen.

* Dr. Hans-Peter Schöner ist CEO von ‚Insight from Outside‘-Consulting, www.ifo-consulting.com (e-mail: hans-peter.schoener@gmx.net).

Analysiert man den Fragenkatalog, lassen sich drei Hauptpunkte herauslesen:

- Der verantwortliche Fahrer muss *ein Verständnis für die Aufgabe und die Szenarien nachweisen*, in denen er für die Sicherheit verantwortlich ist: Was sind die Gründe für Gefährdungen (Gefahrenpotentiale), unter welchen Bedingungen und wie wahrscheinlich drohen Gefahren, welche Handlungen können Gefahren erhöhen, verringern oder zuverlässig abwenden.
- Der verantwortliche Fahrer muss die Entwicklung der Situation *kontinuierlich beobachten und dabei eine Bewertung der Sicherheit* durchführen, um mögliche Gefahren frühzeitig zu erkennen, so dass die weitere sichere Entwicklung der Situation noch gewährleistet werden kann. Er muss sein Wissen und Verständnis über verborgene Gefahren und Risiken einbeziehen und wahrscheinliche Entwicklungen der Situation antizipieren, insbesondere bei unvollständigen Informationen über das gesamte Szenario.
- Der verantwortliche Fahrer muss das erwartete Verhalten anderer und die eigenen Wahrnehmungs- und Reaktionsfähigkeiten einbeziehen, um die *Wirksamkeit seiner möglichen Reaktionen zu beurteilen*. Wetter-, Straßen- und Verkehrsverhältnisse müssen berücksichtigt werden, um schnell und stark genug reagieren zu können.

In einem Gerichtsverfahren spielen auch Zeugen eine Rolle; sie tragen zur Klärung durch beobachtbare Fakten bei: *Was ist passiert? Wie waren die Bedingungen? In welcher Abfolge* geschah etwas? Im Gegensatz zu einem Zeugen muss die verantwortliche Person jedoch zusätzliche Fragen beantworten, um das *Warum* zu erklären: Was waren Ihre *Absichten?* Welche *Schlüsse* haben Sie aus den beobachtbaren Fakten gezogen? Und nach welchen *Überlegungen* haben Sie *Ihr Verhalten* und *Ihre Reaktionen* ausgerichtet?

Andererseits hat die Verantwortung für eine Aufgabe oder einen Prozess nicht nur das Ziel, einen unbeschadeten Ausgang eines Rechtsstreits sicherzustellen. Das Hauptziel der Verantwortung besteht darin, die Aufgabe oder den Prozess zum Erfolg zu führen, *einschließlich* der Sicherstellung, dass aus dieser Aktivität keine unangemessenen Gefahren entstehen. Die Berücksichtigung der o.g. erwarteten Fragen eines Richters ist jedoch eine gute Grundlage, um sich bewusst zu machen, wie verantwortungsvolles Verhalten umgesetzt werden kann.

Alle diese Anforderungen an einen verantwortungsvollen Fahrer gelten nicht nur für menschliche Fahrer, sondern müssen auch bei technischen Umsetzungen eines verantwortungsbewussten Fahrers berücksichtigt werden. Die Fragen, die in einem Gerichtsverfahren gestellt werden, sind genau die gleichen, die ein Entwickler bei der Entwicklung des verantwortungsvollen Verhaltens einer technischen Lösung berücksichtigen muss. Nur durch solch verantwortungsvolles Verhalten lassen sich viele seltene, aber schwere Unfälle vermeiden – im langfristigen und umfassenden Einsatz werden auch seltene gefährliche Situationen mit immer höherer Wahrscheinlichkeit auftreten.

2 Das Konzept der ‘Taktischen Sicherheit’

Auftretenswahrscheinlichkeit, Beherrschbarkeit und Schadenshöhe von Unfallsituationen sind als die drei elementar beitragenden (multiplikativen) Faktoren für ein quantifiziertes Risikomanagement in vielen Anwendungen bekannt, wie ausführlicher in [3] und [4] erläutert. Wenn die Beherrschbarkeit als Wahrscheinlichkeit C gemessen wird, eine herausfordernde Situation schadlos zu meistern, dann ist der anwendbare Faktor zur Berechnung des Risikos (1-C). Der Risikobeitrag R eines einzelnen risikobeuragenden Szenarios ist also

$$R = E(1-C)S$$

mit

- E: Auftretenswahrscheinlichkeit (exposure) einer unfallträchtigen Situation (durch äußere Bedingungen oder eigenes Verhalten);
- C: Beherrschbarkeit (controllability) als Wahrscheinlichkeit für die Vermeidbarkeit von Schäden in einer akuten unfallträchtigen Situation;
- S: Schadenshöhe bzw. Schwere (severity) eines Unfalls, wenn ein Schaden nicht vermieden werden konnte.

Um das Gesamtrisiko gering zu halten, muss der verantwortliche Fahrer die Wahrscheinlichkeit einer gefährlichen Situation (E) einschätzen, und zwar insbesondere solche mit einem hohen möglichen Schweregrad S oder solchen mit einer geringen Beherrschbarkeit C. Ein guter menschlicher Fahrer wird immer bestrebt sein, die Exposition E gegenüber solchen Situationen durch aktive Änderung seines Verhaltens zu verringern: Da ein hoher Geschwindigkeitsunterschied der Hauptparameter ist, der zur Schwere beiträgt, verringert eine *Geschwindigkeitsreduzierung* im Allgemeinen die mögliche Schadenshöhe; und da die verfügbare Reaktionszeit (des Fahrers selbst, aber auch anderer Verkehrsteilnehmer) der Hauptbeitragsparameter zur Beherrschbarkeit ist, verringert ein *größerer räumlicher und damit auch zeitlicher Abstand* zu anderen Objekten das Risiko, eine Gefahrensituation nicht beherrschen zu können.

Daraus leitet sich das Konzept der *Taktischen Sicherheit* ab: Eine frühzeitige und sanfte Änderung des Verhaltens auf Basis von Indikatoren für möglicherweise gefährliche Situationen, um potentiell unkontrollierbare Situationen zu verhindern.

Solche taktischen Verhaltensänderungen bringen die Verkehrsteilnehmer in eine sicherere Position gegenüber potenziell unkontrollierbaren, schweren Unfallsituationen. Auf diese Weise vermeiden sie, im Falle einer aufkommenden Gefahr gezwungen zu sein, in einer herausfordernden kurzen Zeitspanne oder mit ungewöhnlich hoher Intensität auf eine plötzlich auftretende Situation zu reagieren. Solche frühzeitigen und sanften Änderungen des Verhaltens als Maßnahme der ‘taktischen Sicherheit’ reduzieren das *Risiko* im Straßenverkehr. Sie ergänzen die Begrenzung der *Schadenshöhe* in Kollisionen durch Maßnahmen der ‘Passiven Sicherheit’ sowie die Beherrschbarkeit von akuten *Gefahren* durch schnelle und heftige Eingriffe der ‘Aktiven Sicherheit’ in kritischen Situationen.

Die Notwendigkeit taktischer Verhaltensänderungen spiegelt die Tatsache wider, dass allgemeine Verkehrsregeln den Verkehr unter *normalen* Bedingungen sicher machen sollen, aber nicht ohne Änderungen auf *alle auftretenden* Verkehrssituationen angewendet werden können. Beispielsweise ist die Reaktionszeit eine Kombination aus *Wahrnehmungszeit* und *Aktionszeit*: Beide können stark von Verkehrs- und Wetterbedingungen abhängen, aber auch vom Zustand der Straße oder des eigenen Fahrzeugs. Ungewöhnliche Bedingungen erfordern ein außergewöhnlich vorsichtiges Verhalten. Auch kooperatives Verhalten zur gemeinsamen Bewältigung von Situationen, in denen die einfache Anwendung allgemeiner Regeln zu einer Verkehrsbehinderung oder -gefährdung führen würde, sind Ausdruck taktischen Sicherheitsverhaltens.

3 Die Bedeutung einer kontinuierlichen Sicherheits-Metrik

Für die taktische Sicherheit reicht es also nicht aus, allgemeine Verhaltensregeln strikt zu befolgen, sondern es muss für jedes einzelne Szenario separat *beurteilt* werden, ob die allgemeinen Regeln noch angemessen oder möglicherweise nicht sicher genug sind. Ein wesentlicher Bestandteil für die technische Umsetzung dieser Aufgabe ist eine geeignete Metrik für die Gefährlichkeit (oder Sicherheit) einer Situation. In Abb. 1 [3] wird (am Beispiel des sicheren zeitlichen Folgeabstands bei Autobahnfahrten) gezeigt, dass eine kontinuierliche Sicherheitsmetrik diesem Ziel viel besser dienen kann als eine traditionelle scharfe (binäre) Unterscheidung zwischen sicheren und unsicheren Situationen.

Im linken Teil der Abb. 1 ist das traditionelle binäre Konzept der Sicherheitseinstufung skizziert: Eine bestimmte Situation gilt als sicher, wenn ein relevanter Parameter, wie der zeitliche Sicherheitsabstand für die Autobahnfolgeaufgabe, über einem definierten Grenzwert liegt, und als unsicher unterhalb desselben Grenzwerts. Diese scharfe Grenze ist nützlich für die Entscheidung, ob eine Notfallmaßnahme eingeleitet werden muss, um eine Kollision in der letztmöglichen Sekunde zu vermeiden. Der geeignete Schwellwert hierfür hängt jedoch von der konkreten Situation ab: In normalen Fahrsituationen (1) würde ein geringerer Sicherheitsabstand ausreichen, in seltenen Fällen (2, wie z. B. bei einer nassen Fahrbahn) wird jedoch ein deutlich höherer Sicherheitsabstand benötigt. Und bei extremen Bedingungen (3, wie bei schlechter Sicht, Glatteis oder überschwemmter Fahrbahn) ist ein noch größerer Sicherheitsabstand nötig, um eine Kollision zuverlässig zu vermeiden – selbst bei sonst gleichartiger Verkehrskonstellation (wie plötzliches Abbremsen eines vorausfahrenden Fahrzeugs). Jede konkrete Wahl einer einzelnen Worst-Case-Situation und des entsprechenden Schwellwerts für die Auslegung führt zu einer Bewertung, die entweder zu restriktiv für normale Situationen oder zu fahrlässig für seltene und extreme Situationen ist. Eine taktische Anpassung an veränderte Bedingungen ist eine Lösung. Aber die binäre Sicherheitsmetrik ist dafür nicht hilfreich.

Die rechte Seite von Abb. 1 zeigt den Fall einer kontinuierlichen Sicherheitsmetrik. Dabei reicht die Skala für die Sicherheitsbewertung von *‘verlässlich sicher’* (für große relevante

Sicherheitsparameter) bis hin zu ‘*inakzeptabel unsicher*’ (für kleine relevante Sicherheitsparameter). Gemäß der Theorie der Fuzzy-Entscheidungsfindung [8] stellt die rote Gerade eine vereinfachte Beziehung zwischen messbaren Parametern und der Sicherheitsbewertung für eine bestimmte Verkehrssituation her. Eine solche Metrik erlaubt es, abhängig von den Bedingungen ein Sicherheitsniveau anzufordern: Im Regelfall werden alle normalen Situationen (1) mit mehr oder weniger großen Sicherheitsmargen abgedeckt; wenn die Bedingungen günstig sind, z. B. für eine begrenzte Zeit in einer gut verstandenen, überschaubaren und vorhersehbaren Situation, kann sogar vorübergehend eine reduzierte Sicherheitsmarge akzeptabel sein. Aber wenn die Bedingungen schwieriger (schlechte Wetter- oder Straßenbedingungen), weniger beobachtbar (Sichtbehinderungen oder verringerte Sensorreichweite) oder unvorhersehbar (Gegenverkehr auf Einbahnstraßen oder Störungsbedingungen) werden, muss ein Sicherheitsniveau näher an ‘*verlässlich sicher*’ (2 oder 3) ausgewählt werden. Die relevanten Sicherheitsparameter müssen entsprechend angepasst werden: Vor allem Fahrgeschwindigkeit, Folgezeitabstand und seitliche Abstände sind die wichtigsten taktischen Steuerungsparameter zur Risikominderung eines Verkehrsszenarios in Abhängigkeit der identifizierten Randbedingungen.

Konkrete praktische Vorschläge zur Implementierung einer kontinuierlichen Sicherheitsmetrik wurden insbesondere in zwei unabhängigen Literaturquellen [1] und [5] veröffentlicht. Im Anhang von [5] wird eine Literatürvorschau verschiedener Ansätze zur Bewertung sicherer Fahrabstände auf Autobahnen gegeben; diese kommt zu dem Schluss, dass die binäre Einteilung von Situationen in *sicher* und *unsicher* der Vielzahl möglicher Zustände nicht angemessen und für eine kontinuierliche Regelaufgabe eines sicheren Abstandes nicht geeignet ist.

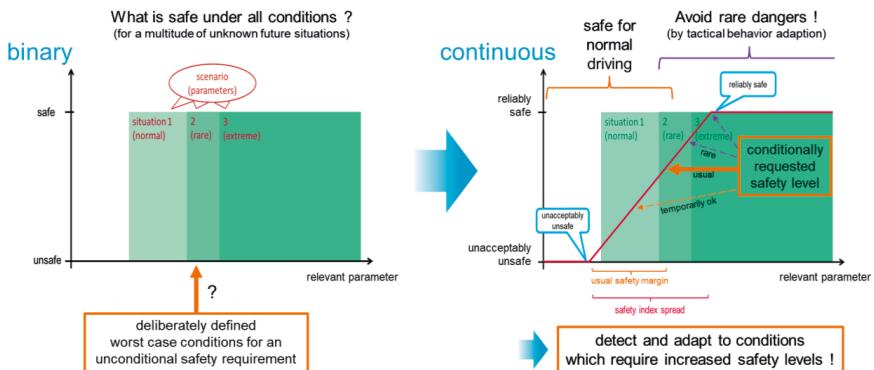


Abbildung 1: Übergang von binärer zu kontinuierlicher Metrik für sicheres Fahren
(z.B. zeitlicher Sicherheitsabstand als relevanter Sicherheitsparameter)

Im Haupttext von [5] wird eine kontinuierliche Sicherheitsmetrik vorgeschlagen, parametrisiert aus Beobachtungen des menschlichen Verhaltens während des Fahrens auf Autobahnen; ein ausreichender Sicherheitsabstand ist breit gestreut und wird je nach Gegebenheiten und individuellem Ermessen gewählt. In [1] wird die kontinuierliche Sicherheitsmetrik unter Verwendung des binären RSS-Sicherheitsmodells [2] abgeleitet, wobei jedoch berücksichtigt wird, dass dieses Modell unterschiedlich parametrisiert werden muss, je nachdem, welches maximale Bremskraftniveau als akzeptabel angesehen wird. Beide Ansätze kommen zu sehr ähnlichen Ergebnissen, obwohl ihr ursprünglicher Ansatz recht unterschiedlich ist: Beide ordnen eine kontinuierliche Sicherheitsfunktion dem Verkehrsszenario selbst zu, während die äußeren Bedingungen von Wetter, Straße, Fahrzeug und Beobachtbarkeit der Situation bestimmen, welches Niveau der Sicherheitsmetrik zu einer akzeptablen Sicherheit für die spezifische Situation führt.

Basierend auf einer solchen Metrik kann das resultierende Sicherheitsniveau eines beobachteten Verhaltens in einem bestimmten Szenario gemessen und als *Qualitätsindex* für das Verhalten des Fahrers verwendet werden. Die kontinuierliche Sicherheitsmetrik ermöglicht es zudem, eine *Regelgröße* für das Fahrverhalten des autonomen Fahrzeugs festzulegen: Es kann eine Sicherheitsreserve abhängig von den äußeren Bedingungen adaptiv zugewiesen werden und so das Fahrzeug mit einer angemessenen Reserve für Unvorhergesehenes gegenüber den Grenzen der Unbeherrschbarkeit fahren. Darüber hinaus benötigt künstliche Intelligenz zum Lernen einen Bewertungsindex, und die kontinuierliche Sicherheitsmetrik kann (nach Abschluss des Szenarios) als *retrospektiver Indikator* für das automatische Lernen von verbessertem Verhalten dienen.

Für das Testen und die Freigabe von autonomen Fahrzeugen kann die Anwendung kontinuierlicher Sicherheitsmetriken (ggf. für mehrere relevante Parameter) als Nachweis dafür dienen, dass ein autonomes Fahrzeug das Gefahrenniveau eines Szenarios ‘versteht’ und sich an die äußeren Bedingungen anpassen kann. Das Konzept hat ein hohes Potenzial, kann aber durch weitere Forschung und Entwicklung noch ausgebaut werden, um sich als wesentlicher Bestandteil für die Sicherheitsbewertung autonomer Fahrzeuge zu etablieren.

4 Die Rolle von kooperativem Sensieren für die Sicherheit autonomer Fahrzeuge

Eine zweite wesentliche Komponente für die taktische Sicherheit ist die Fähigkeit, nach vorne zu schauen und Anhaltspunkte für die Vorhersage der weiteren Entwicklung eines Szenarios zu haben. Aufgrund der Tatsache, dass der vorausfahrende Verkehr die Sicht jeglicher bordeigener Sensorik beeinträchtigen kann, gibt es Risiko-Szenarien mit großer

Schadenshöhe, die insbesondere bei schneller Autobahnfahrt nicht vollständig vom Fahrzeug selbst aus im Sinne der Regelungstechnik ‘beobachtbar’ sind [3]. Eine solche unvollständige Beobachtbarkeit erfordert eine Verhaltensänderung zur sicheren Seite, da sie unter ungünstigen Bedingungen die Gefahr einer schweren Kollision impliziert. In solchen Szenarien können menschliche Fahrer schwache Hinweise erkennen (oft als ‘Intuition’ oder ‘Siebter Sinn’ bezeichnet) oder Warnhinweise anderer Verkehrsteilnehmer verstehen und als Anlass für taktisches Sicherheitsverhalten anwenden. Eine gleichwertige Technologie für autonome Fahrzeuge fehlt noch. Angesichts der heute gut funktionierenden Dienste, die auf Crowdsourcing-Informationen basieren, kann eine Kommunikation zwischen Fahrzeugen über Verkehrs-, Straßen- und Wetterbedingungen – unter einigen zusätzlichen Voraussetzungen – diese Lücke schließen.

Eine mögliche Lösung ist die Einrichtung einer ‘Echtzeit-Verkehrssicherheits-Datenbank’ (‘Real-time Traffic Safety Data Base’, RTS-DB) mit vertrauenswürdigen und zeitgestempelten Informationen über den tatsächlichen Verkehr, das Wetter und die Straßenbedingungen, wie sie z.B. in der EU schon seit einiger Zeit vorgesehen, aber bisher nur in Forschungsprojekten umgesetzt wurde [6], [7]. Diese Datenbank sollte Verkehrswarnungen entsprechen, wie sie über Radio oder Verkehrsfunk an menschliche Fahrer verteilt werden. Aber es sollte präziser, vertrauenswürdiger und mit Informationen über *Vollständigkeit* und *Aktualität* der Informationen ausgeführt sein. Nur auf diese Weise können diese Warnungen als *verlässliche* Informationen für Sicherheitsanwendungen verwendet werden. Um verlässlich und vollständig zu sein, benötigt die Datenbank nicht nur Informationen über Orte erhöhter Gefahren, sondern ergänzend auch bestätigende Informationen über normal fließenden Verkehr ohne ungewöhnliche Gefahren. Auf diese Weise werden auch Orte mit *fehlenden* ausreichenden, verlässlichen oder aktuellen Informationen über die Verkehrslage offensichtlich. Die Datenbank muss von autonomen Fahrzeugen lesbar sein und als zusätzlicher Input für die taktische Verhaltensentscheidung eines automatischen Fahrsystems dienen. Zusätzlich wäre sie auch für ADAS-Systeme der *Taktischen Sicherheit* zur Unterstützung menschlicher Fahrer verwendbar.

In [3] und [4] wird ein mögliches Einstiegskonzept für eine RTS-DB vorgeschlagen, das die Fahrgeschwindigkeit für jeden Fahrstreifen auf Schnellstraßen erfasst und bereitstellt, siehe Abb. 2. Kernpunkt dieses Konzepts ist die Behandlung hoher Geschwindigkeits-*unterschiede* auf einer Schnellstraße als ein Sicherheitsrisiko; sie können zu schweren Kollisionen führen, insbesondere wenn sie unerwartet und zudem hinter vorausfahrendem Verkehr verborgen sind. Crowdsourcing-Geschwindigkeitsprofile entlang Autobahnen erfüllen die Vollständigkeits- und Aktualitätsanforderungen mit relativ geringen Bandbreiten- und Realitäts-zu-Datenbank-Latenzanforderungen.

Das Konzept nach Abb. 2 sammelt Geschwindigkeitsinformationen von jedem beitragenden Fahrzeug auf der Straße. Das Fahrzeug meldet mindestens seine eigene Geschwindigkeit, Fahrstreifen und Längsposition; mit einem Radarsensor können entsprechende Informationen auch über andere Objekte auf der Straße (einschließlich nicht

mit Sensoren bestückter Fahrzeuge und Hindernisse) erfasst und geliefert werden. Insbesondere, wenn sowohl langsam fahrende Lkw als auch eher schnell fahrende Luxus-Pkws diese Datenerhebung unterstützen, kann schon mit einer relativ geringen Teilnehmerzahl ein vollständiges Bild über den Verkehrsfluss erstellt werden. Zusätzliche Quellen dieser Informationen können straßenseitige Sensoren sein. Die Daten können für Längsabschnitte der Straße aggregiert und in Geschwindigkeitsklassen in Schritten von z.B. 30 km/h klassifiziert werden. Wichtig ist dabei die Information zur Aktualität und Datendichte in Zeit und Raum, beispielsweise durch Bereitstellung der Anzahl der Geschwindigkeitsmeldungen pro Abschnitt in einem gleitenden letzten Zeitintervall.

Diese Daten werden zu verlässlichen Informationen, wenn unterstützende Metadaten von einem Dienstleister zusätzlich bereitgestellt werden können, insbesondere wie viele *konsistente* Datenberichte pro Straßenabschnitt und in einem definierten aktuellen Zeitintervall verfügbar sind [4]. Der Information ‘*auf Spur i fließt der Straßenverkehr mit der angezeigten Geschwindigkeit*’ ist ab einer gewissen Anzahl konsistenter Meldungen in den letzten Minuten (z. B. die letzten 3 Minuten) zu trauen. Aber auch das Wissen, dass es keine oder nur sehr wenige Datenpunkte gibt, ist eine auswertbare Information, die in Verbindung mit der Kenntnis des aktuell vorliegenden Erfassungshorizonts des eigenen Fahrzeugs genutzt werden kann, um daraus auf ein sicheres oder möglicherweise gefährliches eigenes Fahrverhalten zu schließen. Die Kommunikation der Datenbank bietet so eine redundante, diversitäre und weitreichende Quelle für die Geschwindigkeit des vorausfahrenden Verkehrs.

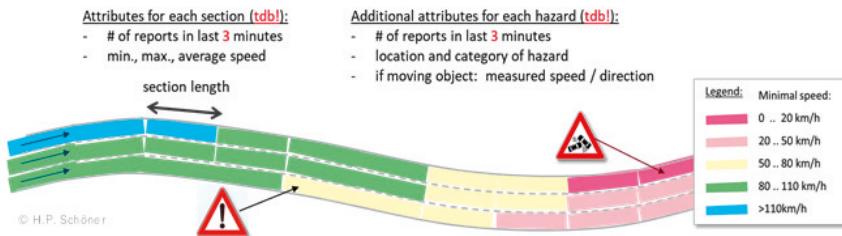


Abbildung 2: Konzept für eine vollständige und verlässliche Gefahrenwarnung auf Schnellstraßen [3]

Aus Sicht des Autors sollte eine solche Echtzeit-Sicherheits-Datenbank zumindest für autonome Fahrzeuge mit hohen Fahrgeschwindigkeiten obligatorisch sein, solange ein fahrzeugbasiertes Sensorset alleine nicht garantieren kann, das vollständige Bild der vorausliegenden Straße zu liefern. Die genaue Spezifikation der Dateninhalte ist noch offen für die technologische Entwicklung auf der Basis bestehender Grundlagen [6, 7].

Eine Crowdsourcing-Datensammlung kann offen gestaltet sein und die Funktionalität bezogen auf spezielle Gefährdungen oder Services im Laufe der Zeit erhöht werden. Ein großer Schritt für die Sicherheit ist aber alleine der Übergang von ‘*vagen Informationen über vereinzelte Verkehrssereignisse*’ auf ein ‘*vollständiges, aktuelles und verlässliches Wissen über den Verkehrsfluss*’ auf der vorausliegenden Strecke.

5 Bewertung des verantwortlichen Fahrens Autonomer Fahrzeuge

Sowohl die ‘Kontinuierliche Sicherheits-Metrik’ als auch die ‘Real-time Traffic Safety Data Base’ können das Risikomanagement für *automatisierte* oder *autonome* Fahrzeuge nachhaltig verbessern. Darüber hinaus können sie auch dazu beitragen, die Sicherheit von Fahrzeugen mit *menschlichen* Fahrern zu verbessern. Beide benötigen keine fahrerlosen Fahrzeuge auf der Straße, um das Konzept umzusetzen; sie können ebensogut in von Menschen geführten Fahrzeugen eingeführt werden. Für einen zügigen Aufbau dieser Sicherheitskonzepte im Zusammenhang mit der Einführung autonomer Fahrzeuge könnte jedoch eine Verordnung hilfreich sein, da ein einzelner Hersteller kaum in der Lage ist, den erforderlichen Standard ohne eine solche Verordnung zu setzen.

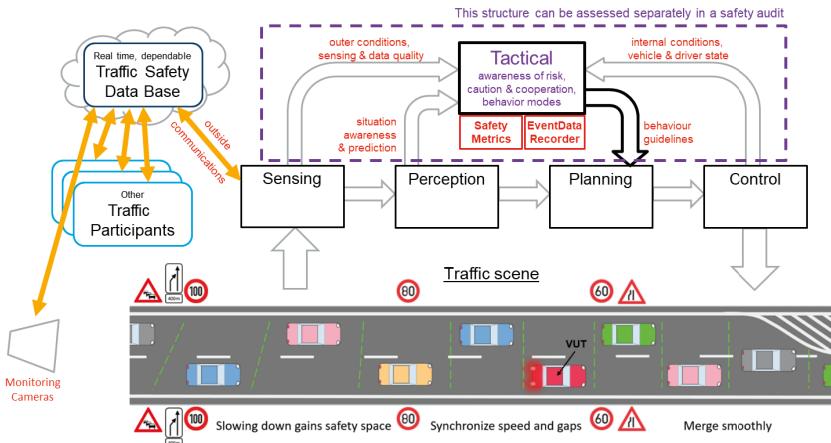


Abbildung 3: Beurteilung sicherheitsrelevanten Verhaltens von Autonomen Fahrzeugen

Ein generisches Blockdiagramm, wie Sicherheitsmetriken und kooperatives Sensieren zur Implementierung von Taktischer Sicherheit in einem Autonomen Fahrsystem (Autonomous Driving System, ADS) beitragen können, ist in Abb. 3 [3] dargestellt; die

Struktur innerhalb des gestrichelten Rechtecks könnte als ‘*Taktisches Verhaltensmodul*’ bezeichnet werden. Die Aufgabe dieses Moduls besteht darin, das Zusammenwirken des gesamten ADS in Bezug auf die impliziten Verantwortlichkeiten eines Fahrers explizit darzustellen. Es berücksichtigt Informationen aus den verschiedenen Hauptbausteinen eines autonomen Fahrsystems (Umfelderfassung, Situationsbewertung, Fahrplanung, Steuerung), zum Beispiel:

- Bewertung der Integrität des Gesamtsystems (abgeleitet aus der Überprüfung des Zustands der Fahrzeugsysteme, insbesondere der Erfassungs- und Steuerungsfähigkeiten),
- Beurteilung der herrschenden Umgebungsbedingungen (abgeleitet aus Wahrnehmung und Vorhersage der Verkehrssituation aus eigener Sensorik sowie aus der Außenkommunikation),
- Gesamtrisikobewertung (abgeleitet aus der Szenarioerkennung, äußeren Bedingungen und den Sicherheitsmetriken).

Aus all diesen Eingaben generiert das taktische Verhaltensmodul ein Risikobewertung für das autonome Fahrsystem, leitet das notwendige Maß an Vorsicht und Kooperation ab und stellt die Parameter für die Steuerung entsprechend ein. In besonderen Fällen schaltet es zwischen verschiedenen grundsätzlichen Verhaltensmodi der Steuerung um. Zusammenfassend führt das taktische Verhaltensmodul eine Einschätzung der Sicherheit der Situation durch und steuert Verhaltensrichtlinien zur Risikominderung.

Der ‘Event Data Recorder’ (EDR), wie er bereits von mehreren Verordnungen gefordert wird, wäre ein Bestandteil dieses taktischen Verhaltensmoduls, der die notwendigen Daten bereitstellt und speichert, die für taktische Sicherheitsentscheidungen benötigt und verwendet werden; im Falle eines Gerichtsverfahrens kann der EDR als *Zeuge* für den untersuchten Sachverhalt dienen. Der Block ‘Safety Metrics’ berechnet aus diesen Daten die quantitative Einschätzung von Sicherheits- bzw. Gefährdungslage, und damit die verantwortungsgemäße *Begründung* für taktische Verhaltensentscheidungen. Diese helfen bei Freigabe- und Zulassungsprüfungen (und ggf. bei Gerichtsverfahren) zu beurteilen, ob die Situation bezüglich Gefährdungen ausreichend analysiert und ‘verstanden’ wurde, und ob die Verhaltens-Entscheidungen des Systems angemessen waren. Die Sicherheitsmetriken und das taktische Sicherheitsmodul liefern also den entscheidenden Nachweis, ob bzw. dass die *Verantwortung* für die Fahraufgabe ausreichend wahrgenommen wurde.

Zusammenfassung

Für die verantwortungsvolle Fahraufgabe ist der Begriff des *Risiko-Bewusstseins* essentiell. Die Verwendung einer kontinuierlichen Sicherheitsmetrik bietet einen Mechanismus, um diesen Begriff in Fahrzeugsteueralgorithmen zu handhaben. Diese bietet eine Abstraktionsebene auf Basis der *Gefährlichkeit von Situationen*, die die Chance hat, die Komplexität der Sicherheitsbewertung autonomer Fahrzeuge zu vereinfachen. Sie erlaubt zudem zu beobachten, ob die Fahrzeugssteuerung die Fähigkeit hat, die Gefahren einer Situation einzuschätzen, und dann zu verifizieren, dass sie angemessen dazu handelt.

Die kooperative Erfassung und Nutzung von Verkehrsdaten in einer ‘Echtzeit-Verkehrssicherheits-Datenbank’ ist der zweite wichtige Faktor, um noch bestehende Handicaps bei der Wahrnehmung von Verkehrs-, Straßen- und Umwelt-Risikobedingungen zu überwinden. So können überragende menschliche Fähigkeiten zur Beurteilung der Gesamtsituation anhand zahlreicher unscharfer Hintergrundinformationen (wie Wettervorhersagen, Straßenzustandsinformationen, Warnlichter, etc.) durch proaktive Sicherheitsmaßnahmen autonomer Fahrzeuge unter Einbeziehung der Informationen der Echtzeit-Datenbank kompensiert werden.

Diese zusätzlichen Hinweise auf das Risiko einer Situation sind unerlässlich für *taktische Verhaltensentscheidungen* mit dem Endziel eines verantwortungsvollen Fahrens: *Halten wir uns bewusst von Gefahren fern, die wir möglicherweise nicht kontrollieren können!*

Literatur

- [1] Mattas, Konstantinos; e.al.; 2020. "Fuzzy Surrogate Safety Metrics for real-time assessment of rearend collision risk". In: Accident Analysis and Prevention 148 (2020)
- [2] MobilEye. 2018. "Implementing the RSS Model on NHTSA Pre-Crash Scenarios". https://www.mobileye.com/responsibility-sensitive-safety/rss_on_nhtsa.pdf
- [3] Schöner H.P., Antona-Makoshi J.: "Testing for Tactical Safety of Autonomous Vehicles". 30th Aachen Colloquium Sustainable Mobility, 2021.
- [4] Schöner, Hans-Peter; Antona-Makoshi, Jacobo; 2022. "Tactical Safety for Autonomous Vehicles on Highways". in Winner, Hermann (editor) e.al.: Handbook Assisted and Automated Driving. 4th edition, Springer Nature, to be published 2022.
- [5] Schöner, Hans-Peter; Pretto, Paolo; Sodnik, Jaka; Kaluza, Bostjan; Komavec, Mojca; Varesanovic, Damir; Chouchane, Hanna; Antona-Makoshi, Jacobo; 2021. "A safety score for the assessment of driving style". Traffic Injury Prevention, 22:5, 384-389, DOI: 10.1080/15389588.2021.1904508
- [6] SensorIS; 2020. SensorIS_specification_v1.1.1. <http://www.sensoris.org/presentations>
- [7] SRTI Ecosystem; 2021. <https://www.dataforroadsafety.eu/>
- [8] Zadeh, Lotfi A. 2008. Fuzzy logic. Scholarpedia. 3(3):1766. doi: 10.4249/scholarpedia.1766

Analyse von Dekompositionsprozessen zur Umsetzung einer modularen Absicherung automatisierter Fahrzeuge

Björn Klamann^{*} und Hermann Winner[†]

Zusammenfassung: Die in der Automobilindustrie übliche Sicherheitsvalidierung auf der Systemebene verstärkt für immer komplexere Systeme, wie automatisierte Fahrzeuge, den Nachteil, dass Änderungen an einzelnen Komponenten eine erneute Freigabe des gesamten Systems erfordern. Mit dem Konzept einer modularen Absicherung erfolgt die Sicherheitsvalidierung bereits auf Modulebene. In diesem Beitrag werden mit einem neu entwickelten Ansatz Dekompositionsprozesse vom System zu Modulen hinsichtlich möglicher Ungewissheiten und Ungenauigkeiten untersucht. Aus den Erkenntnissen werden Regeln definiert, deren Einhaltung dem Entstehen von Ungewissheiten bei einer modularen Absicherung entgegenwirkt.

Schlüsselwörter: Automatisiertes Fahren, Modularare Absicherung, Dekomposition.

1 Einleitung und Definitionen

Nach heutigem Stand der Technik konzentriert sich die Sicherheitsvalidierung in der Automobilindustrie auf die Fahrzeug- bzw. die Systemebene. Da das System Fahrzeug jedoch zunehmend komplexer geworden ist, beinhalten verknüpfte Entwicklungsprozesse inzwischen bereits auf niedrigen hierarchischen Ebenen umfassende Tests, z. B. auf Komponentenebene. Dies ist einerseits eine Konsequenz erreichter Fortschritte in Bereichen der Software- und Hardware-in-the-Loop-Tests [1], aber gleichzeitig ein Grund für intensive Weiterentwicklungen im Bereich des simulativen Testens. Trotz einhergehender Verbesserungen in der Systemzuverlässigkeit ist die endgültige Sicherheitsvalidierung nach wie vor nur auf Systemebene zulässig [2]. Es wurde bereits gezeigt, dass Feldtests auf Systemebene mit zunehmender Fahrzeugkomplexität insbesondere bei der Einführung hochautomatisierter Fahrfunktionen nicht mehr wirtschaftlich zu bewältigen sind [3]. Für automatisierte Fahrfunktionen wird das szenariobasierte Testen als vielversprechender Ansatz gesehen (siehe z. B. [4, 5]). Nach einer Sicherheitsvalidierung auf Systemebene ist es jedoch bereits aufgrund geringer Änderungen oder Systemvariationen notwendig, alle Szenarien erneut zu testen, sofern nicht anderweitig nachgewiesen werden kann, dass die Änderungen keinen Einfluss auf die Ergebnisse der Sicherheitsvalidierung haben. Darüber hinaus zeigen Amersbach und Winner [6], dass eine praktikable Anwendung des

* Björn Klamann ist wissenschaftlicher Mitarbeiter am Fachgebiet Fahrzeugtechnik der TU Darmstadt, Otto-Berndt-Straße 2, 64287 Darmstadt (e-mail: bjoern.klamann@tu-darmstadt.de).

† Hermann Winner ist Professor für Fahrzeugtechnik an der TU Darmstadt, Otto-Berndt-Straße 2, 64287 Darmstadt (e-mail: hermann.winner@tu-darmstadt.de).

szenariobasierten Testens aufgrund der erforderlichen Anzahl an Szenarien immer noch eine große quantitative Herausforderung darstellt. Das von uns vorgestellte Konzept der modularen Absicherung mit Modulen als Subsysteme des übergeordneten Systems "Fahrzeug" stellt den Vorteil in Aussicht, dass Änderungen an einem einzelnen Modul nur die wiederholte Absicherung der modifizierten Module erfordern. Hierdurch sind auch verschiedene Kombinationen von Modulen für verschiedene Anwendungsfälle umsetzbar, ohne dass für jede einzelne Kombination eine eigene Absicherung erforderlich ist [7]. Zur Realisierung einer Absicherung auf Modulebene, stellt sich die Frage nach den Unterschieden zwischen dem Entwicklungs- und Testprozess auf Systemebene und besagter Modulebene. In der vorliegenden Arbeit wird daher der Dekompositionsprozess von System- auf die Modulebene betrachtet. Dabei dekomponierte Informationen unterliegen Ungewissheiten¹ und Ungenauigkeiten², die in der folgenden Analyse aufgedeckt werden. Als Modul wird ein Subsystem mit modularen Eigenschaften verstanden, wodurch es eine relative Unabhängigkeit zu anderen Subsystemen seines Systems aufweist [9, S. 28, 10, S. 209]. Im Projekt UNICARagil wird darüber hinaus zwischen zweckdifferenzierten modularen Architekturen unterschieden. Im vorliegenden Beitrag wird hieraus zusammen mit den Definitionen aus der zitierten Literatur folgende Definition abgeleitet:

„Ein **Modul** ist eine Gruppierung von Komponenten mit höherer Kopplung untereinander als zu Komponenten anderer Module in Bezug auf den Zweck der jeweiligen modularen Architektur. Ein Modul ist daraus folgend relativ unabhängig von anderen Modulen in Bezug auf den Zweck der jeweiligen modularen Architektur, besitzt aber wohldefinierte Schnittstellen zu diesen.“

Zweck einer modularen Architektur kann die Darstellung von Zusammenhängen der Funktionen, Hardwarekomponenten oder Softwareteilen sein. Für die von uns angestrebte modulare Absicherung gilt der Zweck, eine möglichst gute Testbarkeit bzw. Absicherung zu erreichen. Die modularen Architekturen sind dabei voneinander abhängig. Mit der Definition für ein Modul und der Darstellung der Zusammenhänge zwischen den erwähnten Hierarchieebenen nach der ISO 26262 [2] und Steimle et al. [11] ergibt sich die Darstellung nach Abbildung 1.

¹ **Ungewissheit** (eng.: uncertainty) beschreibt das Fehlen der vollständigen Gewissheit, d. h. das Vorhandensein mehrerer Möglichkeiten. Das wahre Ergebnis/Zustand/Ergebnis/Wert ist nicht bekannt. (Übersetzung nach Hubbard et al. [8]) Ungewissheit kann in diesem Beitrag über ein System, Modul oder der jeweiligen Umgebung vorliegen, sodass das deren wahres Verhalten nicht für alle Bedingungen bekannt ist. Das unbekannte Verhalten ist jedoch nicht notwendigerweise unsicher, weshalb wir von der Übersetzung „Unsicherheit“ abssehen.

² **Ungenauigkeit** beschreibt eine Abweichung, z. B. eines Simulationsmodells von der Realität, dessen Maß das Residuum ist.

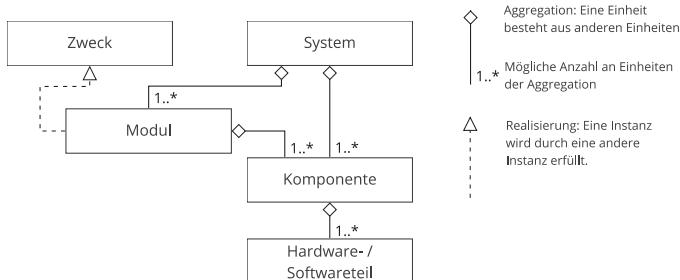


Abbildung 1: In diesem Beitrag betrachtete Hierarchieebenen und Zusammenhänge in Bezug auf Module, dargestellt als UML-Klassendiagramm.

2 Methodik und Struktur des Beitrags

Zur Ableitung möglicher fehlerhafter Schlüsse, die bei Anwendung eines modularen Absicherungsprozesses im Vergleich zur Absicherung auf Systemebene gezogen werden, ist die Aufdeckung möglicher Ungewissheiten und Ungenauigkeiten im Dekompositionsprozess notwendig. Dekomposition ist allgemein die Aufspaltung einer größeren Einheit in kleinere Einheiten. In der Systemtechnik wird angenommen, dass die dabei entstehenden kleineren Einheiten die Eigenschaften der größeren Einheit erfüllen [12, S. 127]. Diese Annahme ist jedoch erst nach Nachweis der Erfüllung der Eigenschaften der größeren Einheit möglich.

Während sich die Sicherheitsvalidierung in der Automobilindustrie vorwiegend auf das Testen konzentriert [13, S. 75], schließt der vorgestellte Ansatz neben dem Testprozedere auch den Entwicklungsprozess mit ein. Die Beherrschung des Entwicklungsprozesses ist für das Konzept einer modularen Absicherung zwingend erforderlich, da eine vollständige Testabdeckung nicht erreichbar ist [14, S. 208].

Im Entwicklungsprozess können falsche Schlussfolgerungen gezogen werden, sodass die Implementierung der Funktion davon betroffen wird. Der englische Begriff für eine fehlerhafte Implementierung („fault“) wird einheitlich in verschiedenen Domänen verwendet (siehe z.B. [2], [14], [15]). Die deutsche Übersetzung „Fehlzustand“ nach IEC [15] impliziert ggf. nur das Fehlen eines Zustands in der Implementierung. Eine fehlerhafte Implementierung aufgrund falscher Schlussfolgerungen wäre dadurch nicht eindeutig beschrieben. Wir verwenden daher den Begriff „Fehlerzustand“ nach Spillner und Linz [14, S. 7]. In der Softwareentwicklung ist eine falsche Schlussfolgerung im Entwicklungsprozess mit dem englischen Begriff „error“ belegt, der einen „fault“ verursacht. Der Begriff „error“ wird in der ISO 26262 [2] oder nach Avizienis et al. [16] dagegen als Abweichung von einem korrekten Wert oder Zustand während des Betriebs verstanden, der erst durch einen „fault“ ausgelöst wird. Für die Entstehung des Fehlerzustands im Entwicklungsprozess wird kein Begriff genannt. Wir führen daher für eine falsche Schlussfolgerung bzw. falsche Ergebnisse im Entwicklungsprozess den Begriff „Irrtum“ ein. Dieser ist ebenfalls als mögliche Übersetzung von „error“ nach IEC aufgeführt [15]. Schlussendlich folgt nach Stolte et al. [17] aus einem Irrtum bzw. einem Fehlerzustand ggf. eine Beendigung des

gewünschten Verhaltens in Form eines Ausfalls oder Versagens (eng.: „failure“). Während der Begriff Ausfall in der Literatur breite Anwendung findet (siehe z.B. [12, S. 97], [15], [18, S. 48]), sind die Autoren des vorliegenden Artikels der Auffassung, dass der Begriff missverständlich ist und eine vollständige Beendigung einer Funktion impliziert. Dies ist für klassische Systeme im Automobil für die Mechanik und E/E-Systeme oft zutreffend, passt jedoch nicht zu einem fehlerhaften Verhalten einer Software aufgrund eines Irrtums in der Spezifikation. Daher wird der Begriff Versagen als Beendigung des beabsichtigten Verhaltens bevorzugt.

Die konkreten Irrtümer hängen von der konkreten Umsetzung des weiteren Entwicklungsprozesses ab. Diese Prozesse sind je nach Domäne und Unternehmen jedoch stark unterschiedlich und werden in anderen Publikationen analysiert (siehe z. B. [19, 20]), weshalb die vorliegende Arbeit den Dekompositionsprozess möglichst allgemein betrachtet. Die Identifizierung möglicher Irrtümer im Dekompositionsprozess erfolgt daher mit Hilfe einer Fehlerbaumanalyse ausgehend von dem Zustand eines nicht abgesicherten Moduls aufgrund von Irrtümern im Dekompositionsprozess. Zur Ergänzung des entwickelten Fehlerbaums verwenden wir einen ähnlichen Ansatz wie die Systemtheoretische Prozessanalyse (STPA) von Leveson et al. [21] Für beide identifizierten übergeordneten Irrtümer wird eine jeweils passende vereinfachte Architektur modelliert. Durch die Fehlerbaumanalyse identifizierte Irrtümer werden den modellierten Entitäten oder Verbindungen zugeordnet. Entitäten und Verbindungen ohne zugewiesene Irrtümer werden detailliert auf weitere mögliche Irrtümer durch die Zerlegungsprozesse analysiert. Aus den aufgedeckten Fehlermöglichkeiten werden daraufhin Regeln für den Dekompositions- und Testprozess definiert.

3 Irrtümer im Dekompositionsprozess

Die Analyse geht von dem Top-Ereignis aus, dass Module nach erfolgter modularer Absicherung nicht ausreichend abgesichert sind. Dies ist für den Dekompositionsprozess unter der Annahme, dass die Spezifikation auf Systemebene korrekt und vollständig ist, auf lediglich zwei allgemeine Ursachen zurückzuführen. Nach der Dekomposition und Spezifikation der Module könnten diese einerseits die Anforderungen des Gesamtsystems im Verbund nicht erfüllen, obwohl die Spezifikation eingehalten wird. Andererseits könnte eine vollständig korrekte Spezifikation nicht erfüllt sein. Für die zweite Ursache beschränken wir uns darauf, dass Modultests nicht offenlegen, dass die Spezifikation nicht erfüllt wird. Zur Aufdeckung der Irrtümer im Entwicklungsprozess, die zur Nichterfüllung der Spezifikation führen, wäre der gesamte Entwicklungsprozess zu analysieren. Dieser ist, wie zuvor erwähnt, abhängig von Unternehmen und Produkt und daher für eine generalisierte Analyse nicht geeignet. In dieser Arbeit wird folglich die Annahme getroffen, dass Modultests grundsätzlich eine Abweichung von der Modulspezifikation aufdecken können, sofern die richtigen Tests richtig durchgeführt würden. Die Ableitung solcher Testfälle und entsprechender Testumgebungen wird außerdem wesentlich vom Fokus der folgenden Analyse, dem Dekompositionsprozess, beeinflusst. Für eine modulare Absicherung müssen die Tests schlussendlich eine aggregierte Aussage über die Leistungsfähigkeit bzw. das Restrisiko des Systems liefern.

3.1 Irrtümer in Modulspezifikationen

Die erste Ursache, dass die spezifizierte Funktion eines Moduls zusammen mit den Funktionen anderer Module nicht die Anforderungen auf Systemebene erfüllt, wird als erster möglicher Irrtum angenommen, für den weitere untergeordnete Ursachen identifiziert werden. Die Ableitung weiterer Ursachen geschieht mit Unterstützung einer möglichst generischen funktionalen Architektur. Abbildung 2 zeigt eine solche Architektur, die aus einem System mit den Modulen 1 und 2 besteht, die die Funktionen F1, F2 und F3 haben.

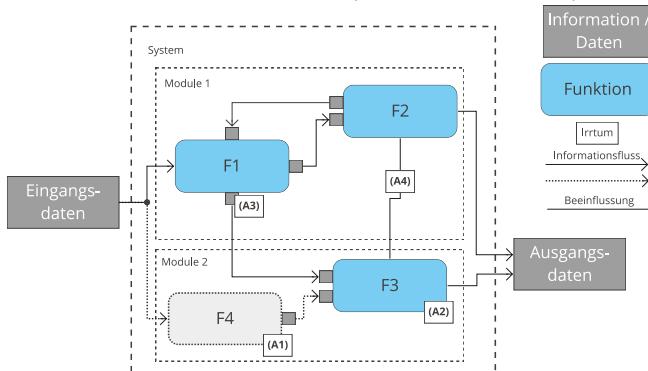


Abbildung 2: Beispielhafte abstrakte funktionale Architektur mit Irrtum an der Stelle ihres Auftretens.

Dargestellt durch einen grauen Block, stellt Funktion 4 eine fehlende Funktion dar, die eine Ursache für Irrtum A ist und als Irrtum A1 in den Fehlerbaum nach Abbildung 3 aufgenommen wird. Der Irrtum A1 kann dadurch entstehen und ggf. unentdeckt bleiben, dass Anforderungen zwar grundlegend erfüllt werden, dies jedoch für bestimmte Zustände nicht mehr gilt. In UNICARagil konnte dies mit dem anfänglichen Fehlen der von Homolla et al. [21] beschriebenen Posen-Offset-Korrektur beobachtet werden. Inkonsistente Lokalisierungsdaten aus verschiedenen Lokalisierungsdiensten führen ohne die beschriebene Posen-Offset-Korrektur zu einer zunehmenden Abweichung von der Soll-Trajektorie, sodass die Anforderungen auf Systemebene nicht mehr erfüllbar sind.

Irrtum A2 bezieht sich auf fehlerhaft oder unvollständig spezifizierte Funktionen, sodass Anforderungen auf Systemebene nicht erfüllt werden. Ein einfaches Beispiel hierfür ist die Anforderung auf Systemebene, die maximale Geschwindigkeit einzuhalten, die durch eine Drehzahlbegrenzung als Funktion umgesetzt wird. Ist diese Begrenzung jedoch mit einer falschen Getriebeübersetzung berechnet, kann die Geschwindigkeitsbeschränkung auf Systemebene überschritten werden, obwohl die Anforderung auf Modulebene erfüllt ist.

Ebenso kann mit Irrtum A3 nur die Schnittstellenbeschreibung fehlerhaft, unvollständig oder inkonsistent in sich oder zu anderen Schnittstellen sein. Ein Beispiel hierfür ist die Verwendung unterschiedlicher Koordinatensysteme als Referenz für Werteangaben.

Schnittstellen können darüber hinaus auch ungewünschte Interaktionen auslösen, die unter Irrtum A4 im Dekompositionsprozess nicht erkannt werden. Insbesondere in der Mechanik ist dies oft nicht vermeidbar, da z.B. die Verbindung zweier Bauteile nicht nur Kraft,

sondern auch Wärmeenergie überträgt, deren Übertragung aber nicht gewünscht ist. Zwischen zwei Softwarediensten kann ebenfalls eine ungewünschte Interaktion, z.B. bei gemeinsamer Nutzung einer Recheneinheit auftreten, sodass für mindestens einen Dienst nicht mehr ausreichend Rechenkapazität zur Verfügung steht.

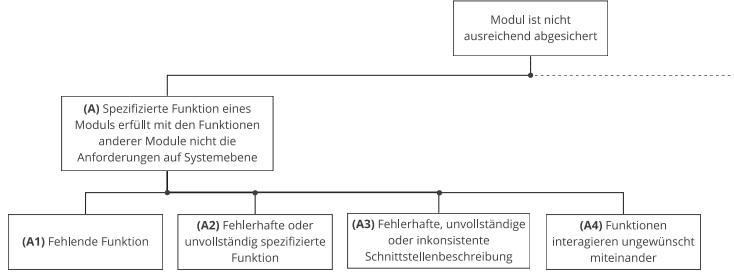


Abbildung 3: Fehlerbaum für Irrtum A.

3.2 Irrtümer in Modultests

Der folgende Abschnitt zeigt mögliche Ungewissheiten im Entwicklungsprozess auf, die als Konsequenz von Ungewissheiten oder Ungenauigkeiten beim Testen verursachen. Der Irrtum, dass ein *Modul nicht ausreichend getestet* ist (B), ist entweder darauf zurückzuführen, dass die *richtigen Tests falsch durchgeführt* werden (B1) oder dass die *falschen Tests durchgeführt* werden (B2). Zur Erhöhung der Vollständigkeit des Fehlerbaums wird wieder eine Visualisierung möglicher Irrtümer eingesetzt. Abbildung 4 zeigt dazu einen generisch gehaltenen Prüfstand, der ein zu prüfendes Modul, eine spezifizierte Testumgebung sowie den Testfallgenerierungs- und Evaluationsprozess enthält.

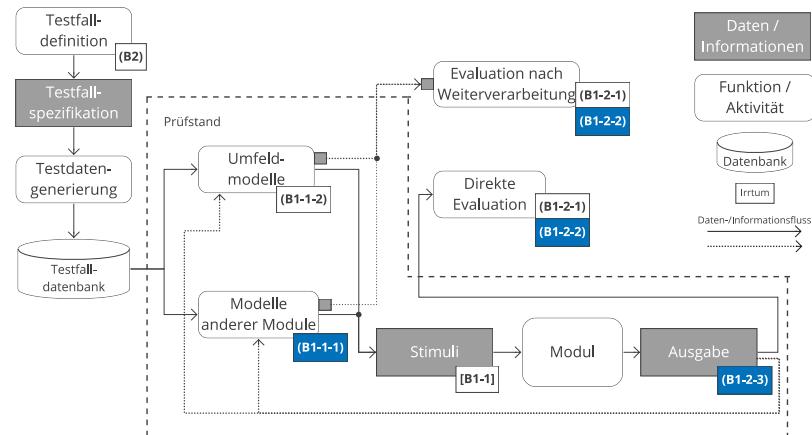


Abbildung 4: Beispielhafter generischer Prüfstandsaufbau für Modultests inkl. Testfallerzeugung und Evaluation. Irrtümer sind an der Stelle ihres Auftretens eingezzeichnet. Blaue Kästen sind mit der Analyse des Prüfstands ergänzte Irrtümer.

Falsch durchgeführte Tests lassen sich auf invalide Stimulation (B1-1) oder invalide Evaluation (B1-2) zurückführen. Dabei wird für die modulare Absicherung die Bedingung gesetzt, dass Module unabhängig von anderen Modulen testbar sind. Die Einschränkung folgt aus der Anforderung der modularen Absicherung, dass Module veränderbar sind, ohne die Absicherung anderer Module erneut durchführen zu müssen. Im parallelisierten Entwicklungsprozess ist außerdem vorteilhaft, nicht auf einen ausreichend reifen Entwicklungsstand anderer Module warten zu müssen. Die Unabhängigkeit bezieht sich also auf zukünftige Änderungen anderer Module sowie deren aktuellen Entwicklungsstand bzw. die exakte Implementierung der Funktionen nach dem Konzept der Kapselung [22, S. 332 f.]. Dies ist nur möglich, wenn der Einfluss geplanter Änderungen auf das untersuchte Modul bekannt ist. Die Ermittlung zukünftiger Änderungen als auch deren Einfluss sind jedoch nicht nur aufgrund unbekannter Ausrichtung der zukünftigen Ziele eines Produkts oder des Unternehmens schwierig abzuschätzen. Auch die Aufdeckung unbekannter Unbekannter (z.B. zuvor unbekannte Szenarien, die einen Unfall verursacht und damit einen Fehlerzustand in einem Modul aufgedeckt haben) machen zukünftige Änderungen nur beschränkt vorhersehbar. Vielversprechender ist dagegen der Einsatz vereinfachter Repräsentationen anderer Module, die lediglich das für das untersuchte Modul relevante Verhalten abbilden. Änderungen anderer Module müssen dementsprechend nur ggü. diesem spezifizierten Modellverhalten abgesichert werden.

Mit Hilfe des Prüfstandsaufbaus lassen sich als Ursachen die Verwendung invalider Modelle anderer Module (B1-1-1) und speziell der Umgebung (B1-1-2) herleiten.

Für eine invalide Evaluation können auch eine unzureichende Metrik (B1-2-1), unzureichende zugeordnete Bestehens-/Versagenskriterien (B1-2-2) sowie unzureichende

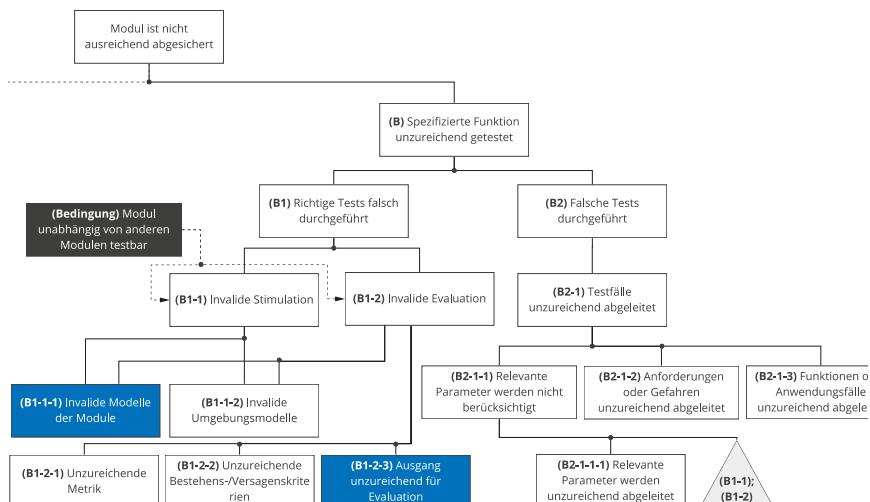


Abbildung 5: Fehlerbaum für Irrtum B

Ausgaben am Modul (B1-2-3) als Ursachen identifiziert werden.

Der rechte Ast des Fehlerbaums listet lediglich falsche Tests aufgrund unzureichender Ableitung von Testfällen (B2-1) auf, da hier nur der Dekompositionsprozess, nicht aber die Testfallgenerierung auf Systemebene betrachtet wird. Daraus ergibt sich der potenzielle Irrtum, dass relevante Parameter nicht berücksichtigt werden (B2-1-1), da diese unzureichend abgeleitet wurden (B2-1-1-1) oder da diese über die Simulationsmodelle fehlerhaft oder unvollständig berücksichtigt werden, sodass die Stimulation bzw. Evaluation insgesamt als invalide zu bewerten ist (B1-1, B1-2). Ein derartiger Irrtum trat im Projekt UNICARagil wie folgt auf. Ausgehend von der Annahme, den Regler durch hohe laterale Beschleunigungen herauszufordern, wurden Testdaten zur Abfahrt verschiedener Kreisbahnen erzeugt. Zur Initialisierung der Kreisfahrt wurde dabei zuerst geradeaus und dann in den Kreis eingefahren. Bei dieser Einfahrt zeigte sich eine größere Regelabweichung als während der gesamten Kreisfahrt. Zurückführen lässt sich dies auf die Gierbeschleunigung bei Einfahrt in die Kreisbahn, bei der der untersuchte Regler höhere Regelabweichungen erzeugt als bei lateraler Beschleunigung. Das Beispiel deckt keinen überraschenden Zusammenhang auf, illustriert jedoch wie Parameter unberechtigt vernachlässigt werden können.

4 Regeln für den Dekompositionsprozess

Anhand der zuvor entwickelten Irrtümer werden in Tabelle 1 Regeln definiert, die diese Irrtümer vermeiden. Hierzu werden die Irrtümer auf den niedrigsten Ebenen negiert und die Erkenntnisse aus deren Herleitung genutzt, um die Regeln zu konkretisieren. Die Einhaltung der Regeln im Entwicklungsprozess sind die Basis für einen erfolgreichen modularen Absicherungsprozess.

Tabelle 1: Regeln R1-R7 zur Vermeidung von Irrtümern im Dekompositionsprozess

Nr.	Regel
R1	Die funktionale Architektur muss Funktionen enthalten, die alle Anforderungen des Systems erfüllen. Somit müssen alle Funktionen für alle definierten Zustände auf die Anforderungen des Systems und vice versa rückverfolgbar sein.
R2	Die modularen Architekturen müssen die Schnittstellen zwischen den Modulen detailliert beschreiben, um sicherzustellen, dass die Funktionen gemeinsam die Anforderungen des Systems erfüllen. Die Schnittstellenbeschreibung muss hierzu eine Funktionsbeschreibung, die zulässigen Werte, ihr zeitliches Verhalten und mögliche Verläufe einschließlich eines Fehlermodells enthalten.
R3	Die modularen Architekturen müssen Abhängigkeiten zwischen den Modulen in allen möglichen Zuständen der Module bzw. des Gesamtsystems berücksichtigen.
R4	Die modularen Architekturen müssen auf unerwünschte Interaktion zwischen den Modulen untersucht werden.
R5	Die modularen Architekturen müssen Schnittstellen bereitstellen, die gültige Darstellungen anderer Module und der Umgebung in Bezug auf alle gewünschten Testfälle und deren Testziele ermöglichen. Daher sind Testfälle und Testziele vor der endgültigen Definition der modularen Architekturen zu definieren.
R6	Die modularen Architekturen müssen Schnittstellen bereitstellen, an denen alle erforderlichen Metriken und die zugehörigen Bestehens-/Versagenskriterien anwendbar sind. Hierfür können zusätzlich Modelle anderer Module oder der Umgebung verwendet werden. Daher sind Metriken und Bestehens-/Versagenskriterien vor der endgültigen Definition der modularen Architekturen zu definieren.
R7	Alle Informationen auf Modulebene (bspw. Funktionalitäten, Anforderungen, Anwendungsfälle, Parameter, Gefahren oder Testfälle) müssen durch mindestens eine Architektsicht auf die Systemebene und vice versa rückverfolgbar sein.

Regel 1 leitet sich insbesondere aus den Irrtümern A1 und A2 ab. Während die Notwendigkeit aller Funktionen zur Erfüllung der Anforderungen trivial erscheint, beinhaltet sie zusätzlich die Forderung nach Rückverfolgbarkeit von Funktionen und Anforderungen zwischen System- und Modulebene. Insbesondere für Systeme mit verschiedenen Entwicklungspartnern, die ggf. nur beschränkt Informationen untereinander austauschen, ist dies eine Herausforderung [12, S. 151]. Darüber hinaus ist die rein statische Betrachtung der Funktionen und Anforderungen eines dynamischen Systems nur zulässig, wenn die Funktionen und Anforderungen für alle dynamischen Zustände gleich formulierbar sind. Komplexe Systeme erfordern meist die Formulierung verschiedener Zustände (z.B. für verschiedene Betriebsmodi, Informationsverfügbarkeiten oder bei Richtungsabhängigkeiten), da sich ein Modul in diesen ggf. unterschiedlich verhält. Diese Zustände und zugehörige Zustandsübergänge sind bei der Spezifikation von Funktionen zu

berücksichtigen.

Regel 2 ist zu entnehmen, dass die Schnittstellen zwischen den Modulen so detailliert zu beschreiben sind, dass nicht nur die spezifizierte Modulfunktion dargestellt wird, sondern auch abgeleitet werden kann, dass alle Module im Verbund alle Anforderungen der Systemebene erfüllen. Obwohl die Beschreibung der Schnittstellen als wichtiger Meilenstein im Entwicklungsprozess gesehen wird, ist die Umsetzung einer technischen Schnittstellenbeschreibung in Entwicklungsprozessen oft lückenhaft [19, 221-224, 23, S. 314]. Derartige Lücken finden sich auch in Normen wie der ISO 26262 wieder, worin lediglich explizite Anforderungen an die Beschreibung der Hardware-Software-Schnittstellen gefordert werden [2, Teil 4]. Ein mögliches Konzept in Form einer detaillierten semantischen Schnittstellenbeschreibung wird in diesem Beitrag nach Vorstellung der weiteren Regeln erläutert.

Regel 3 erweitert Regel 2 durch die Berücksichtigung indirekter Abhängigkeiten über mehrere Module hinweg. Hierdurch werden Einflüsse von Modulen identifiziert, die keine Schnittstellen zum untersuchten Modul haben. Auf Grundlage von Sensitivitätsanalysen der jeweiligen Module lässt sich die Relevanz dieser Einflüsse quantifizieren. Der identifizierte Einfluss kann ebenfalls in einer semantischen Schnittstellenbeschreibung festgehalten werden, sodass der Einfluss bei der Absicherung des Moduls Berücksichtigung findet.

Regel 4 präzisiert Regel 3 für Einflüsse, die nicht erwünscht, ggf. aber nicht vermeidbar sind, wie bereits für Irrtum A4 beschrieben. Solche Einflüsse sind zu identifizieren und z.B. durch Sensitivitätsanalysen zu quantifizieren, um zu entscheiden, ob diese in Modultests zu berücksichtigen sind oder ob darauf verzichtet werden kann.

Aus den Regeln 5 und 6 lassen sich Anforderungen an die modularen Architektursichten ableiten. Regel 5 fordert, dass für alle notwendigen Modultests die Umgebung ausreichend valide repräsentiert werden kann und adressiert damit die möglichen Irrtümer B1-1-1 und B1-1-2. Im Rahmen einer modularen Absicherung muss diese Repräsentation mögliche Änderungen der repräsentierten Module berücksichtigen. Mögliche Änderungen sind hinsichtlich ihres Einflusses auf die Repräsentation bzw. das untersuchte Modul zu analysieren. Hierbei ist für jedes Modul ein Optimum zwischen Änderungskapazität und seiner spezifischen Konfiguration für einen bestimmten Anwendungsfall zu finden. Eine Erhöhung der Änderungskapazität bedeutet, dass umfangreichere Änderungen an Modulen zulässig sind, ohne dass andere unveränderte Module erneut abzusichern sind. Dies kann sich auf die Leistungsfähigkeit der Module auswirken, da diese mit vielfältigen Varianten/Konfigurationen kompatibel sein müssen und dazu weniger spezifisch konfiguriert werden.

Zur Einhaltung von Regel 6 und der Vermeidung der Irrtümer B1-2-1, B1-2-2 und B1-2-3 gelten ähnliche Randbedingungen für die Bestehens-/Versagenskriterien wie für die Modulrepräsentationen. Die Änderungskapazität verhält sich für Bestehens-/Versagenskriterien ebenfalls gegenläufig dazu wie spezifisch ein Modul konfiguriert ist, sodass ein Optimum aus beiden Kriterien angestrebt wird. Grundsätzlich muss zur Einhaltung der Regel für jede Schnittstelle definiert werden, welche Kriterien erfüllt werden müssen, damit eine Ausgabe zulässig ist. Die Ausgabe kann dabei direkt oder indirekt bewertet werden. Für indirekte Bewertungen wird die eigentliche Ausgabe zuerst weiterverarbeitet. Hierfür sind ggf. Repräsentationen anderer Module notwendig. Eine Metrik bewertet die jeweilige

finale Ausgabe.

Regel 7 fordert die allgemeine Rückverfolgbarkeit von Informationen zwischen Modul- und Systemebene, um die Irrtümer B2-1-1-1, B2-1-2 und B2-1-3 zu vermeiden. Sie erweitert damit Regel 1, fordert jedoch nicht explizit die durchgehende Konsistenz zwischen den Informationen verschiedener Architektsichten, wie sie nach Regel 1 zwischen Funktionen und Anforderungen notwendig ist. Stattdessen wird für alle Architektsichten gefordert, dass die Informationen innerhalb einer Architektsicht zwischen den Hierarchieebenen rückverfolgbar sind.

In UNICARagil werden Schnittstellen in einer dienstebasierten Architektur in funktionale Daten und Qualitätsdaten aufgeteilt [24]. Qualitätsdaten werden ergänzend zu den Daten zur Umsetzung der eigentlichen Funktion eingesetzt, um die aktuellen Fähigkeiten des Fahrzeugs zu bestimmen und durch einen Abgleich mit den geforderten Anforderungen zu prüfen, ob das Fahrzeug diese erfüllen kann (siehe z. B. [25]). Die Qualitätsdaten sind wie die funktionalen Daten in der Absicherung zu prüfen, sodass die Funktion der Selbstwahrnehmung sichergestellt wird. Qualitätsdaten werden aus Kenntnissen über potenzielle Irrtümer und daraus folgende Fähigkeitsdegradationen abgeleitet. Das Fehlermodell von Schnittstellen für die Absicherung von Modulen erfordert ergänzend dazu eine Beschreibung, welche Residuen an der Schnittstelle unter welchen Bedingungen auftreten. Die Herausforderung besteht jedoch weiterhin darin, das Sollverhalten an den Schnittstellen zu spezifizieren. Eine Betrachtung der Auswirkungen auf das Verhalten des Gesamtsystems ist dabei weiterhin erforderlich.

Aus den Regeln 2, 3, und 4 ergibt sich insbesondere die Notwendigkeit einer detaillierten Beschreibung der Schnittstellen. Schnittstellen lassen sich in syntaktischer Form und semantisch beschreiben. Die Syntax beschreibt dabei, wie Informationen repräsentiert und verarbeitet werden [22, S. 22]. Syntaktische Informationen der Schnittstelle bilden in der Entwicklung meist den Hauptfokus [26, S. 14], da ohne deren Abstimmung eine Kommunikation zwischen Modulen nicht möglich ist. Die Prüfung der Einhaltung an einer Schnittstelle kann durch einen manuellen Vergleich mit der Spezifikation erfolgen und durch Tests ergänzt werden, die versuchen, verschiedene Bereiche dieser Syntax anzusprechen. Neben der Prüfung, ob die Definition der Syntax eingehalten wird, lassen sich Schnittstellentests anwenden, in denen zur Stimulation des Moduls der mögliche Wertebereich systematisch angesprochen wird. Bei Schnittstellentests handelt es sich gleichzeitig um einen Funktionstest unter verschiedenen Eingaben. Hierbei wird durch die Systematik ein hohes Maß an Objektivität, üblicherweise jedoch eine geringe Effizienz bei der Aufdeckung von Fehlerzuständen erreicht. Der Grund hierfür ist, dass Fehlerursachen an einzelnen Stellen oder in einzelnen Zuständen [14, S. 153] und selten gleichverteilt über einen Wertebereich liegen. Daher werden Schnittstellentests durch Methoden wie Äquivalenzklassen- oder Grenzwerttests präzisiert.

In der Literatur und in Normen (z.B. AUTOSAR [27]) nur beschränkt adressiert ist die Semantik einer Schnittstelle. Die Semantik beschreibt als Ergänzung zur Syntax welche Informationen übertragen werden und welche logischen Abhängigkeiten existieren [22, S. 47]. Diese leiten sich üblicherweise aus den Anforderungen an das System bzw. aus den daraus dekomponierten und detaillierten Anforderungen an die Module ab. Ziel einer detaillierten semantischen Schnittstellenbeschreibung ist, die Erkenntnislücke aufgrund

fehlender Integrationstests zu schließen. Die Beschreibung sollte die Transformation bzw. Interpretation zwischen Systemstimulation und Modulstimulation bzw. Modulverhalten und Systemverhalten ermöglichen. Zur Abschätzung der Auswirkungen des Modulverhaltens auf Systemebene ist die Kenntnis des Sollverhaltens aller Module eines Regelkreises notwendig, womit Regel 2 und 3 berücksichtigt werden. Diese Verhaltensableitung ist keinesfalls an typisch prägnanten Stellen (z.B. an der Systemgrenze) zu stoppen, sondern muss immer bis zum Eingang des Moduls verfolgt werden. Damit werden auch Einflüsse auf die Modulteststimuli durch das eigene Modulverhalten berücksichtigt. Zusätzlich zu der jeweiligen Beschreibung sollten mögliche Anomalien, d.h. Abweichungen vom beschriebenen Soll-Verhalten an der Schnittstelle genannt werden. Diese können in der Modulentwicklung und in Modultests berücksichtigt werden und somit potenzielle Irrtümer aufdecken. Extrahierbar sind die Anomalien z.B. aus der Beschreibung der Qualitätsdaten oder anhand der 15 IQ(Informationsqualität)-Dimensionen [28, S. 26-27].

Ein Beispiel für die Notwendigkeit einer semantischen Beschreibung an den Schnittstellen ist das Verhalten des Kurswinkels in einer Trajektorie, wie sie in UNICARagil vom Trajektorienplaner an den Trajektorienregler übermittelt wird. Der Kurswinkel kann für ein notwendiges Ausweichmanöver auf Fahrzeugebene sprunghaft verlaufen. Der Trajektorienregler muss für ein Gelingen des Ausweichmanövers den Kurswinkel möglichst schnell und präzise einregeln. In anderen Szenarien kann es dagegen vorkommen, dass aufgrund kurzzeitiger falsch-positiver Objekte oder Trajektorienkorrekturen der Soll-Kurswinkel sprungartig verläuft. Der Trajektorienregler kann dies jedoch nicht von einem Kurswinkelverlauf für ein Ausweichmanöver unterscheiden und würde ebenfalls mit hoher Querbeschleunigung reagieren. Der Trajektorienplaner muss solche sprungartigen Verläufe daher vermeiden, da aus unnötig hohen Beschleunigungen ein unsicheres Verhalten auf Systemebene erzeugt werden kann. In einer erweiterten semantischen Schnittstellenbeschreibung lassen sich solche Anforderungen an jede Schnittstelle bzw. jeden Parameter der Schnittstelle adressieren. Gleichzeitig können auch mögliche Anomalien wie ein möglicher sprungartiger Verlauf des Kurswinkels, aufgenommen und von allen Abonnenten der Schnittstelle abgenommen oder abgelehnt werden. In zukünftigen Arbeiten werden wir im Rahmen des Projekts UNICARagil systematisch Schnittstellenbeschreibungen erarbeiten und deren Eignung hinsichtlich einer modularen Absicherung untersuchen.

5 Zusammenfassung

Der vorliegende Beitrag stellt das Konzept einer modularen Absicherung anhand eines strukturierten Vorgehens für Entwicklung und Testen vor, das die Unterschiede zwischen System und Modulen berücksichtigt. Als neuer Ansatz zur Identifizierung möglicher Irrtümer im Dekompositionsprozess vom System bis zum abgesicherten Modul wird eine Fehlerbaumanalyse zusammen mit der Analyse einer funktionalen Architektur und eines Prüfstandes vorgestellt. Zunächst wird die Systemebene mit der Modulebene im Hinblick auf die verfügbaren Informationen und Testmethoden verglichen. Die daraus resultierenden Schlussfolgerungen werden genutzt, um Irrtümer abzuleiten, die bei der Dekomposition dieser Informationen für die Entwicklung und den Test von Modulen auftreten können.

Daraus abgeleitete Regeln für den Dekompositionsprozess sind in dargestellter Form noch nicht verifizierbar, können in konkreten Entwicklungsprozessen jedoch als robuster Ausgangspunkt zur Ableitung spezifischer Regeln bzw. Anforderungen dienen.

Auf Systemebene bestehen generell weniger Ungewissheiten und Ungenauigkeiten als auf untergeordneten Modulebenen. Daher sind für eine Absicherung auf Systemebene weniger Analysen erforderlich als in diesem Beitrag für eine modulare Absicherung vorgestellt. Im Vergleich liegt der Analyseaufwand für eine modulare Absicherung initial höher. Allerdings verspricht dieser eine Reduzierung des zunehmenden Testaufwands für automatisierte Fahrfunktionen, da mit ihr ein baugleiches Modul für eine Vielzahl von Fahrzeugen verwendet werden kann, ohne nach Änderungen angrenzender Module Regressionstests zu erfordern. Hinzu kommt die Möglichkeit der funktionalen Dekomposition [29], die auch für Module, die weniger funktionale Ebenen beinhalten als das Gesamtsystem, eine Testaufwandsreduktion verspricht. Daher erwarten wir ein stark steigendes Interesse, sowohl in Bereichen der Forschung als auch der Durchführung, an dem vorgestellten und an weiteren neuen Ansätzen zur Umsetzung einer modularen Absicherung.

Danksagung

Diese Forschungsarbeiten wurden im Rahmen des Projekts "UNICARagil" durchgeführt (FKZ 16EMO0286). Wir bedanken uns für die finanzielle Unterstützung des Projekts durch das Bundesministerium für Bildung und Forschung (BMBF).

6 Literatur

- [1] I. Raghupatruni, S. Burton, M. Boumans, T. Huber und A. Reiter, „Credibility of software-in-the-loop environments for integrated vehicle function validation“ in *20. Internationales Stuttgarter Symposium*, Wiesbaden, 2020, S. 299–313.
- [2] *Road vehicles - Functional Safety*, ISO 26262:2018, International Organization for Standardization.
- [3] W. Wachenfeld und H. Winner, „The Release of Autonomous Vehicles“ in *Autonomous Driving: Technical, Legal and Social Aspects*, M. Maurer, B. Lenz, H. Winner und J. C. Gerdes, Hg., s.l.: Springer, 2016, S. 425–449, doi: 10.1007/978-3-662-48847-8_21.
- [4] M. Wood et al., „Safety First for Automated Driving“, Aptiv, Audi, Baidu, BMW, Continental, Daimler, Fiat Chrysler Automob., Here, Infineon, Intel, Volkswagen, 2019. [Online]. Verfügbar unter: <https://www.daimler.com/dokumente/innovation/sonstiges/safety-first-for-automated-driving.pdf>. Zugriff am: 16. September 2021.
- [5] *Road vehicles — Safety of the intended functionality*, ISO DIS 21448:2021, International Organization for Standardization (ISO).
- [6] C. Amersbach und H. Winner, „Defining Required and Feasible Test Coverage for Scenario-Based Validation of Highly Automated Vehicles**“ in *The 22nd IEEE*

- Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand, 2019, S. 425–430, doi: 10.1109/ITSC.2019.8917534.
- [7] T. Woopen *et al.*, „UNICARagil - Disruptive Modular Architectures for Agile, Automated Vehicle Concepts“ in *27th Aachen Colloquium*, Aachen, 2018, doi: 10.18154/RWTH-2018-229909.
 - [8] D. W. Hubbard, *How to measure anything: Finding the value of "intangibles" in business*, 3. Aufl. Hoboken, New Jersey: Wiley, 2014.
 - [9] J. Göpfert, *Modulare Produktentwicklung: Zur gemeinsamen Gestaltung von Technik und Organisation*. Wiesbaden: Deutscher Universitätsverlag, 1998.
 - [10] D. D. Walden, G. J. Roedler, K. Forsberg, R. D. Hamelin und T. M. Shortell, Hg., *Systems engineering handbook: A guide for system life cycle processes and activities*; *INCOSE-TP-2003-002-04*, 2015, 4. Aufl. Hoboken, NJ: Wiley, 2015.
 - [11] M. Steimle, T. Menzel und M. Maurer, „Toward a Consistent Taxonomy for Scenario-Based Development and Test Approaches for Automated Vehicles: A Proposal for a Structuring Framework, a Basic Vocabulary, and Its Application“, *IEEE Access*, Jg. 9, S. 147828–147854, 2021, doi: 10.1109/ACCESS.2021.3123504.
 - [12] J. Schäuffele, T. Zurawka und R. Carey, *Automotive Software Engineering: Grundlagen, Prozesse, Methoden und Werkzeuge Effizient Einsetzen*, 6. Aufl. Wiesbaden: Springer Fachmedien Wiesbaden GmbH, 2016.
 - [13] H.-L. Ross, *Functional Safety for Road Vehicles*. Cham: Springer International Publishing, 2016.
 - [14] A. Spillner und T. Linz, *Basiswissen Softwaretest: Aus- und Weiterbildung zum Certified Tester – Foundation Level nach ISTQB®-Standard*, 6. Aufl. Heidelberg: dpunkt.verlag, 2019.
 - [15] International Electrotechnical Commission, „International Electrotechnical Vocabulary: IEC ref 192-03-02 (error)“, 1. Feb. 2015. [Online]. Verfügbar unter: <https://www.electropedia.org/iev/iev.nsf/display?openform&ievref=192-03-02>. Zugriff am: 5. November 2021.
 - [16] A. Avizienis, J.-C. Laprie, B. Randell und C. Landwehr, „Basic concepts and taxonomy of dependable and secure computing“, *IEEE Trans. Dependable and Secure Comput.*, Jg. 1, Nr. 1, S. 11–33, 2004, doi: 10.1109/TDSC.2004.2.
 - [17] T. Stolte *et al.*, „A Taxonomy to Unify Fault Tolerance Regimes for Automotive Systems: Defining Fail-Operational, Fail-Degraded, and Fail-Safe“, *IEEE Transactions on Intelligent Vehicles*, 2021, doi: 10.1109/TIV.2021.3129933.
 - [18] M. Hillenbrand, *Funktionale Sicherheit nach ISO 26262 in der Konzeptphase der Entwicklung von Elektrik/Elektronik Architekturen von Fahrzeugen*. Zugl.: Karlsruhe, KIT, Dissertation, 2011. Hannover, Karlsruhe: Technische Informationsbibliothek u. Universitätsbibliothek; KIT Scientific Publishing, 2012.
 - [19] R. Sanchez, „Building real modularity competence in automotive design, development, production, and after-service“, *IJATM*, Nr. 3, S. 204–236, 2013, Art. no. 54918, doi: 10.1504/IJATM.2013.054918.
 - [20] R. Sanchez und T. Shibata, „Modularity Design Rules for Architecture Development: Theory, Implementation, and Evidence from Development of the Renault-Nissan

- Alliance "Common Module Family" Architecture“, *Data Science and Service Research Discussion Paper*, Nr. 80, S. 1–44, 2018.
- [21] T. Homolla, G. Gottschalg und H. Winner, „Verfahren zur Korrektur von inkonsistenten Lokalisierungsdaten in modularen technischen Systemen“ in *Uni-DAS 13. Workshop Fahrerassistenz und automatisiertes Fahren. FAS 2020*, 2021. [Online]. Verfügbar unter: <http://tuprints.ulb.tu-darmstadt.de/18522/>
 - [22] M. BROY und M. Kuhrmann, *Einführung in die Softwaretechnik*. Berlin, Heidelberg: SpringerVieweg, 2021.
 - [23] N. Leveson, *Engineering a safer world: Systems thinking applied to safety*. Cambridge, Massachusetts: MIT Press, 2012.
 - [24] A. Mokhtarian, A. Kampmann, B. Alrifaei und S. Kowalewski, „The Dynamic Service-oriented Software Architecture for the UNICARagil Project“ in *29th Aachen Colloquium Sustainable Mobility*, 2020, S. 275–284.
 - [25] M. Nolte, I. Jatzkowski, S. Ernst und M. Maurer, „Supporting Safe Decision Making Through Holistic System-Level Representations & Monitoring -- A Summary and Taxonomy of Self-Representation Concepts for Automated Vehicles“, 27.07.20. [Online]. Verfügbar unter: <http://arxiv.org/pdf/2007.13807v2.pdf>. Zugriff am: 26. Februar 2022.
 - [26] F. Bachmann *et al.*, „Documenting Software Architecture: Documenting Interfaces“, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA CMU/SEI-2002-TN-015, 2002. [Online]. Verfügbar unter: <http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=5939>. Zugriff am: 25. Februar 2022.
 - [27] AUTOSAR: *Generic Structure Template*, AUTOSAR consortium, Dez. 2017.
 - [28] K. Hildebrand, M. Gebauer und M. Mielke, Hg., *Daten- und Informationsqualität: Die Grundlage der Digitalisierung*, 5. Aufl. Wiesbaden: Springer Vieweg, 2021.
 - [29] C. T. Amersbach, „Functional Decomposition Approach - Reducing the Safety Validation Effort for Highly Automated Driving“. Dissertation, TU Darmstadt, Darmstadt, 2020.

Funktionale Sicherheit KI-basierter Systeme im Automobilsektor

Dagmar Gesmann-Nuissl* und Ines Tacke†

Zusammenfassung: Je umfassender Künstliche Intelligenz (KI) in sicherheitskritische Abläufe und Entscheidungsprozesse eingebunden wird, desto häufiger wird sich die Frage stellen, ob und inwie- weit die (funktionale) Sicherheit dieser KI-basierten Systeme gewährleistet werden kann – eine Frage, die insbesondere für das Inverkehrbringen dieser Systeme in den Europäischen Binnenmarkt relevant wird. Nur KI-basierte Systeme (Anwendungen und/oder Funktionen), denen eine funktionale Sicherheit im Rahmen der Konformitätsbewertung attestiert werden kann, sind zertifizierungs- und zulassungsfähig. Trotz aller Bemühungen auf EU-Ebene bleibt der Nachweis der funktionalen Sicherheit der KI-basierten Systeme derzeit nur eingeschränkt möglich, da die zugrundeliegenden Produktvorschriften die KI-basierten Systeme entweder gar nicht kennen oder dort die Anforderungen an deren funktionale Sicherheit nicht oder nur unvollständig geregelt sind. Dass dies auch für den Automobil-Sektor gilt, ist hinlänglich bekannt und bedarf daher Überlegungen, wie dieses Defizit überwunden werden kann. Hier wird nach der Darstellung der regulativen Zusammenhänge die These vertreten, dass die Leitlinien aus dem Flugverkehrssektor zur Zulassung von KI-Systemen auch für den Automobil-Sektor nutzbar gemacht werden könnten, um dem erklärten Ziel einer zeitnahen Regelzulassung sicherheitskritischer KI-Anwendungen auch im Automobil-Sektor näher zu kommen.

Schlüsselwörter: Künstliche Intelligenz, funktionale Sicherheit, Sicherheitsnachweisführung, Fahrzeugzulassung, Standardisierung.

1 Einleitung

„Vision Zero“ – die Vermeidung aller Verkehrstoten – ist erklärtes Ziel des aktuellen Verkehrssicherheitsprogrammes der Bundesregierung. [1] Zur Erreichung dieses Ziels soll künstliche Intelligenz (folgend KI) eine Schlüsselrolle einnehmen. Dies gilt sowohl für die Mobilität der Gegenwart als auch der Zukunft. So kann beispielsweise eine KI-basierte Bildverarbeitung in Echtzeit zur Objekterkennung ein Spurhalteassistenzsystem (SAE-Stufe 1) ebenso unterstützen, wie das fahrerlose autonome Fahren der SAE-Stufe 5. [2] Um auf allen Automatisierungsstufen das Ziel der „Vision Zero“ nicht zu verfehlten, spielt die funktionale Sicherheit der KI-basierten Systeme eine entscheidende Rolle, da sie darauf abzielt, die Allgemeinheit vor Fehlfunktionen dieser Systeme zu schützen, die auf dem Weg

* Dagmar Gesmann-Nuissl leitet an der TU Chemnitz die Professur für Privatrecht und Recht des geistigen Eigentums und begleitet das Forschungsprojekt „KI-bezogene Test- und Zulassungsmethoden – Anwenderkompass für den intelligenten Schienenverkehr“ auf normativer Ebene, TU Chemnitz, Thüringer Weg 7, 09126 Chemnitz (dagmar.gesmann@wirtschaft.tu-chemnitz.de).

† Ines Tacke ist wissenschaftliche Mitarbeiterin an ebendieser Professur, (ines-maria.tacke@wirtschaft.tu-chemnitz.de).

bis zur Lösungsfindung entstehen können. Insofern manifestiert sich in ihrer Ausgestaltung auch der Bereich, den die Gesellschaft im Hinblick auf bestimmte Entwicklungsstufen von Produkten zu akzeptieren bereit ist. Dies gilt nicht nur für das Fahrzeug insgesamt, sondern auch für alle darin verbauten KI-basierten bzw. KI-gestützten lernenden Systeme, die künftig auch ohne den Fahrer auskommen. Werden solche KI-basierten Systeme in Fahrzeuge integriert, muss auch weiterhin die gewohnte funktionale Sicherheit gewährleistet sein. Dabei gilt: Je autonomer zukünftig KI-Algorithmen Fahrfunktionen oder sicherheitskritische Entscheidungsprozesse im Rahmen der Fahrzeugführung (mit-)tragen, desto klarer müssen die neuen Anforderungen an die funktionale Sicherheit im Hinblick auf diese KI-Subfunktionen definiert sein; eine Fehlerkorrektur seitens des Menschen ist jedenfalls ex tunc beim autonomen Fahren nicht mehr vorgesehen.

Mit der Festlegung der funktionalen Sicherheit von KI-basierten Systemen tut sich die derzeitige Rechts- und Normungspraxis angesichts der zahlreichen auch nicht-technischen Berührungspunkte der KI (z.B. ethische Aspekte) sowie der Tatsache, dass sich existierende und etablierte Absicherungsprozesse sowie die Nachvollziehbarkeit von Funktionsabläufen nicht ohne weiteres auf maschinelle Lernverfahren übertragen lassen, recht schwer, wie nachfolgend im Regelungsrahmen des Automobil-Sektors aufgezeigt wird, bevor mögliche Lösungsansätze aus dem Flugverkehrssektor vorgestellt werden und deren Übertragbarkeit überprüft wird.

2 Regelungsrahmen der funktionalen Sicherheit – ein Überblick

Deutschland nimmt mit seinem hochentwickelten Rechtsrahmen zur Zulassung innovativer Fahrtechnologien international eine Vorreiterrolle ein. [3] Bereits im Juni 2017 wurden die Regelungen zu Kraftfahrzeugen mit hoch- oder vollautomatisierten Fahrfunktionen eingeführt (§§ 1a-1c StVG). Im Juli 2021 traten zudem die Regelungen für Kraftfahrzeuge mit autonomen Fahrfunktionen in festgelegten Betriebsbereichen hinzu (§§ 1d-11 StVG). Insofern wurde der straßenverkehrsrechtliche Rahmen für den Regelbetrieb von Fahrzeugen der SAE-Stufen 3 – 4 geschaffen, deren Fahrfunktionen zunehmend durch Techniken aus dem Bereich der künstlichen Intelligenz unterstützt werden (z.B. maschinelles Sehen; Klassifizierung von Objekten etc.).

2.1 State of the Art

Damit derart optimierte Fahrzeuge in den Regelbetrieb gelangen oder in festgelegten Betriebsbereichen eingesetzt werden können, müssen diese Fahrzeuge einschließlich ihrer KI-basierten integrierten Systeme zugelassen sein, wobei sie – abhängig vom Automatisierungsgrad der Fahrfunktionen – unterschiedlichen sicherheitstechnischen Anforderungen unterliegen, die bislang aber nur zum Teil die Belange der KI hinreichend berücksichtigen.

2.1.1 Hoch- und vollautomatisierte Fahrfunktionen/-systeme

Die Zulassung von Fahrzeugen mit hoch- oder vollautomatisierten Fahrfunktionen erfordert eine konventionelle Verkehrszulassung (§ 1 Abs. 1 StVG) ergänzt um die Zulassung der hoch- oder vollautomatisierten Fahrfunktionen (§ 1a Abs. 3 StVG). Bezüglich der sicherheitstechnischen Anforderungen wird dabei zuvörderst auf die 159 internationalen Rege-

lungen der ‚United Nations Economic Commission for Europe‘ (folgend UNECE) verwiesen (§ 1a Abs. 3 Nr. 1 StVG). [4] Sie beschreiben die technischen Anforderungen an die Fahrzeugkomponenten orientiert an Fahrfunktionen, wie z.B. Bremsen (UNECE Nr. 79) oder die automatisierten Systeme, wie z.B. das automatische Spurhalteassistenzsystem (UNECE Nr. 157). Dabei kennt die neue Generation der UNECE-Regelungen auch die Unterstützung durch KI, indem z.B. beim Einsatz von ‚Lern-Algorithmen‘ eine Datenaufzeichnung zwingend vorgeschrieben wird (vgl. 3.2. Annex 4 UNECE Nr. 157). Allerdings treffen die eher begleitenden Aussagen zur KI keine spezifischen Aussagen zur Sicherstellung ihrer funktionalen Sicherheit (z.B. Vermeidung von Fehlinterpretationen) oder lösen auch nicht das allgemein bekannte Black-Box-Problem; die UNECE-Regelungen bleiben vielmehr technologienneutral.

Alleiniger Maßstab für die Funktionssicherheit bleiben daher die industrieseitigen technischen Standards (für Fahrerassistenzsysteme etwa vorgeschrieben in 7. Anhang 4 UNECE-Nr. 157); zu funktionaler Sicherheit die ISO 26262 ‚Funktionale Sicherheit — Straßenfahrzeuge‘ sowie weitere wichtige Spezifizierungen enthaltend die ISO/PAS 21448 ‚Straßenfahrzeuge — Sicherheit der beabsichtigten Funktion‘.

Traditionell wird die ISO 26262 zur Analyse möglicher Gefährdungen durch fehlerhaftes Verhalten konventioneller, sicherheitsrelevanter Systeme – wie etwa Ausfälle – angewendet. [5] Der Standard gilt als Grundlage für E/E-Systeme (elektrische und elektronische Systeme) in Fahrzeugen. Er spezifiziert die Gefahrenanalyse und Risikoabschätzung für die Bestimmung des Gefahrenlevels des Fahrzeugs und seiner Systeme (Hard- und Software). Durch die Analyse des Kontexts der beabsichtigten Anwendungsfälle können nicht nur Gefährdungen, sondern effektive und wirksame Gegenmaßnahmen identifiziert werden. [6] Mit fortschreitender Automatisierung gerieten jedoch auch über die Fehlfunktion von Komponenten hinausgehende Sicherheitsaspekte in den Fokus, weshalb im Jahr 2018 das Profil des Standards geschärft und Raum für ISO/PAS 21448 (folgend SOTIF) geschaffen wurde. [7] Die ISO/PAS 21448 wurde im Jahr 2019 neben die ISO 26262 gestellt, um noch vorhandene Lücken bei der Analyse von Gefahren innovativer Technologien zu adressieren. [8] [9] Mit ihrem use-case-basierten Ansatz ermöglicht sie seitdem auch die Identifikation von Unzulänglichkeiten der Systeme (Grenzen der bestimmungsgemäßen Funktionalität und Leistungsgrenzen) während des sicheren Betriebs. Mit ihr lassen sich Limitationen und Grenzen vor allem auch komplexer Algorithmen-basierter Systeme und Funktionen identifizieren, [10] etwa die mangelnde Fähigkeit eines Systems, einen weißen LKW vor wolkenbedecktem Himmel zu detektieren. [11] Insofern bilden die Funktionalitäten, bei denen ein angemessenes Situationsbewusstsein sicherheitskritisch ist und die von komplexen Sensoren und Verarbeitungsalgorithmen abgeleitet werden, den Anwendungsschwerpunkt der SOTIF. [12] [13] Allerdings bleibt sie bislang hauptsächlich auf automatisierte Fahrfunktionen der SAE-Stufe 2 ausgerichtet. [12] Für komplexere Systeme sind demnach zusätzliche Sicherungsmaßnahmen erforderlich, die noch nicht definiert sind. [10] [13] Zusammenfassend lässt sich bezogen auf die Regelungen zur funktionalen Sicherheit von hoch- und vollautomatisierten Fahrsystemen daher feststellen, dass sie durch ihren technologienneutralen Ansatz zwar eine Überregulierung vermeiden und Innovationen großzügig zulassen. Allerdings sind insbesondere die älteren Regelungen und Standards vor dem Hin-

tergrund herkömmlicher, unzweifelhaft deterministischer Software entwickelt worden. Da-her adressieren sie nicht die besonderen, sich insbesondere aus dem datengetriebenen Lern- prozess ergebenden, sicherheitstechnischen Herausforderungen von KI. [14] [15] Insofern werden zurecht Bedenken geäußert, auch, weil die sicherheitstechnische Regulierung und die damit erst möglich werdende regelbasierte Sicherheitsprüfung von Produkten ein entscheidendes Merkmal zum Aufbau von Vertrauen und der Marke ‚Made in Germany‘ darstellen. Bleibt es dagegen bei einem regelungsfreien Raum wird dies den Aufbau von Ver- trauen in die Technologie der KI als wichtigen Baustein für die erfolgreiche Marktdurch- setzung KI-basierter Systeme verhindern. Ferner können klar zu erfüllende Vorgaben die Entwicklungskosten bei den System-Herstellern senken, da sie dann eben nicht – wie derzeit – „ins Blaue hinein“ agieren, wenn sie die funktionale Sicherheit nachweisen müssen. Aus den genannten Gründen besteht daher weiterhin Regelungsbedarf zur funktionalen Sicher- heit von KI-Applikationen in automatisierten Fahrsystemen sowie zum Nachweis derselben.

2.1.2 Autonome Fahrfunktionen/-systeme

Die Voraussetzungen für die Zulässigkeit des Betriebs eines Fahrzeugs mit autonomer Fahr- funktion sind in § 1e Abs. 1 StVG angelegt. Sie müssen neben der Regelzulassung (§ 1 Abs. 1 StVG) weitere Voraussetzungen erfüllen, insbesondere den technischen Voraussetzungen des § 1e Abs. 2 Nr. 1-10 StVG entsprechen. [16] Dabei lassen sich die erforderlichen tech- nischen Voraussetzungen in Kategorien einteilen, die ihrerseits den in den internationalen Beschlüssen festgehaltenen technischen Sicherheitsanforderungen an autonome Fahrsys- teme entsprechen müssen [17] [18]. Zur Erlangung der Betriebserlaubnis muss der Herstel- ler das Vorhandensein dieser (bislang eher rudimentär ausgestalteten) Sicherheitsanforde- rungen verbindlich erklären (§ 1f Abs. 3 Nr. 4 StVG). Der herstellerseitigen Erklärung kommt daher ein besonderes Gewicht zu, was auch in der nunmehr gesetzlich normierten Verpflichtung des Fahrzeugherstellers zur Risikobeurteilung zum Ausdruck kommt (§ 1f Abs. 3 Nr. 1-3 StVG). Allerdings fehlt es an der Präzisierung der Anforderungen, die einer solchen Erklärung zugrunde liegen sollen. [16]

Zwar enthält § 1j StVG neuerdings eine Ermächtigung zum Erlass einer die Regelungen ausgestaltenden Verordnung. Diese liegt bislang aber nur als Entwurf des BMDV vor (Autonome-Fahrzeuge-Genehmigungs-und-Betriebs-Verordnungsentwurf – AFGBV-E), wobei bereits sichtbar wird, dass das altbewährten Regelungsregime weitergeführt werden soll (Anhang 1 Teil 1 Nr. 7 AFGBV-E bzw. Anlage 4 AFGBV-E). [19] Insofern wird auch weiter ein herstellerseitiges Sicherheitskonzept betreffend der funktionalen Sicherheit der Systeme erforderlich sein, welches das Kraftfahrt-Bundesamt anschließend überprüfen muss (Anlage 4 Nr. 3 AFGBV-E). Als Maßstab gelten – sofern einschlägig – weiterhin die UNECE-Reglungen sowie, auch für die KI-basierten Systeme, die ISO 26262 sowie die ISO/PAS 21448 (Anlage 1 Teil 1 Nr. 7.2 AFGBV-E). Die hohe Bedeutung der industriesei- tigen Standards für das Sicherheitskonzept der Hersteller bleibt demnach auch bei autono- men Fahrfunktionen bestehen, mit all den zuvor genannten Defiziten. Ein Paradigmenwech- sel, hin zu mehr Verbindlichkeit in diesen hoch sensiblen, den Menschen ersetzenen Be- reichen, zeichnet sich gerade nicht ab. Im Gegenteil soll die Bedeutung der vorhandenen Normen bis zur Etablierung von rechtlich verbindlichen Maßstäben noch zunehmen. [20]

2.2 Aktuelle Aktivitäten der Regelsetzer im Automobil-Sektor

Zu den Anforderungen an KI-basierten Systeme, gibt es derzeit auf allen Regelungsebenen Aktivitäten, die auch den Automobil-Sektor anbetreffen. Zumeist verfolgen sie aber noch nicht das Ziel, die sicherheitstechnischen Anforderungen an derartige Systeme zu präzisieren oder die Nachweisführung (Erklärbarkeit der KI) näher auszugestalten, wie nachfolgend auszugsweise aufgezeigt wird.

2.2.1 Nationale Ebene

Vom nationalen Gesetzgeber wird derzeit einiges unternommen, um die Zulassung von Kraftfahrzeugen mit autonomer Fahrfunktion in festgelegten Betriebsbereichen zeitnah zu ermöglichen (vgl. §§ 1d - 11 StVG sowie AFGBV-E [19]). Allerdings lassen sich in den Regelungen der Durchführungsverordnung, die bislang nur im Entwurf¹ vorliegen, keine technischen Anforderungen an die KI-basierte Systeme oder Vorschläge zur Nachweisführung finden. Beschrieben werden die Anforderungen, die ein autonomes Fahrsystem fahrfunctionsbezogen im Regelbetrieb erfüllen müssen; das „technische make-up“ eines solchen Systems wird dort gerade nicht adressiert. Die Regelungen bleiben auch künftig technologienutral und verlangen danach, durch Normen untersetzt zu werden.

2.2.2. Europäische Ebene

Auf europäischer Ebene verbinden sich hohe Erwartungen an die KI-Verordnung⁴ der Europäischen Kommission, die derzeit im Entwurf vorliegt. [21] [22] Nach ihr werden KI-Systeme anhand ihrer Sicherheitsimplikationen (Kritikalität) in vier Risiko-Kategorien eingeteilt; KI-Systeme mit minimalem Risiko (Art. 69 KI-VO-E), geringem Risiko (Art. 52 KI-VO-E), hohem Risiko (Art. 6 ff. KI-VO-E) und unannehmbarem Risiko (Art. 5 KI-VO-E). Die im Straßenverkehr verwendeten KI-Systeme werden der Gruppe der ‚hochrisiko-KI-Systeme‘ zugeordnet, für welche der Gesetzgeber zunächst grundlegende Anforderungen definiert (Annex 3 Nr. 2 lit. a i.V.m. Art. 6 KI-VO-E). Da aber die KI-VO dem ‚New Legislative Framework‘ (NLF) folgt, ist bereits jetzt impliziert, dass die spezifischen sicherheitstechnischen Anforderungen an KI-basierte Systeme des Automobil-Sektors, also die Anforderungen an dessen funktionaler Sicherheit, auch weiterhin den harmonisierten Normen und Standards vorbehalten bleiben. [23] Es steht nicht zu erwarten, dass sie die Anforderungen an die (produktspezifische) funktionale Sicherheit oder an die (produktspezifische) Nachweisführung in der KI-VO selbst definieren wird.

2.2.3. Internationale Ebene (UNECE)

Auf internationaler Ebene spielt – wie bereits erwähnt - die UNECE eine besondere Rolle. In ihr werden arbeitsteilig, in thematisch organisierten Expertengruppen neue Regelungen auch für den Automobil-Sektor erarbeitet. Die Harmonisierung fahrzeugbezogener Regelungen verantwortet dabei das ‚World Forum for Harmonization of Vehicle Regulations‘ (Wp.29), innerhalb dessen die ‚Working Party on Automated/Autonomous and Connected Vehicles‘ (GRVA) mit Fragestellungen rund um automatisierte Fahrsysteme befasst ist. Sie

¹ Redaktionsschluss 31.03.2022.

bearbeitet in speziellen Unterarbeitsgruppen etwa Fragen zu Cyber-Security und Software-Updates (CS/OTA) oder zur Datenspeicherung (DSSAD/DER).

Für die Ausarbeitung der funktionalen Anforderungen an automatisierte und autonome Fahrsysteme (FRAV) sowie der Validierung von Fahrsystemen (VMAD) existieren spezielle Arbeitsgruppen, auf deren Arbeit hier wegen der systemimmanenter Schnittmenge zu KI-basierten Systemen näher eingegangen werden soll.

Durch die FRAV wurde zuletzt ein Vorschlag für eine Leitlinie für Sicherheitsanforderungen für automatisierte Fahrsysteme (Sicherheitsanforderungs-Leitlinie) in die GRVA eingebbracht, [24] in der Begrifflichkeiten für automatisierte Fahrsysteme definiert werden sollen, die für die Festlegung von Sicherheitsanforderungen an Fahrzeugsysteme von zentraler Bedeutung sein können; etwa Dynamic Driving Task. Daneben wurden Leitlinien für die Beschreibung von automatisierten Fahrsystemen formuliert, etwa zu den Mindestanforderungen an die Beschreibung der Operational Design Domain. Wichtiger Bestandteil des dynamischen Dokumentes sind darüber hinaus die Safety-Empfehlungen für automatisierte Fahrsysteme, welche zunächst verschiedene Sicherheitszustände beschreibt, die in einem nächsten Schritt in Zusammenarbeit mit der VMAD zu messbaren, verifizierbaren, nachweisbaren Anforderungen weiterentwickelt werden sollen. [25] Diesem Ansinnen folgend werden parallel seitens der VMAD dynamisch Leitlinien für Nachweis- und Testmethoden für automatisierte Fahrsysteme (NATM-Leitlinie) entwickelt. [26] Beide Leitlinien folgen einem technologienutralen Ansatz [27] und sollen nach ihrer Fertigstellung Entwicklern und Vertragsparteien der UNECE als Orientierung zu empfohlenen Verfahren zur Sicherheitsvalidierung automatisierter Fahrsysteme dienen. [27]

Allerdings wird aktuell von der GRVA erörtert, ob neben dem technologienutralen Ansatz ein Bedürfnis für eine KI-spezifischen Betrachtung und Ausformulierung existiert. [28][29] Das Mandat zu weiterführenden Überlegungen wurde der GRVA durch die WP.29 erteilt. [30] [31] In einem ersten Schritt sollen daher die für die Automobilbranche relevante Begrifflichkeiten rund um KI gesammelt und geschärft werden, um in einem weiteren Schritt die drängendsten Fragen zu den KI-basierten System – u.a. Erklärbarkeit und Verifizierbarkeit, insbesondere auch im Rahmen von Sicherheitsaudits – zu beantworten. [29] Inwieweit dabei die Arbeit der zuvor genannten Untergruppen (FRAV sowie VMAD) einfließen kann, bleibt abzuwarten.

2.2.4. Standardisierung

Sowohl nationale als auch internationale Normungsorganisationen arbeiten derzeit an Standards, welche die funktionale Sicherheit von KI-basierten Systemen sowie die Nachweisführung im Allgemeinen, aber auch mit einem Fokus auf die Mobilität und den Straßenverkehr näher ausgestalten sollen (vgl. Deutsche Normungsroadmap Künstliche Intelligenz) [32]. Von Seiten der Industrie wurde bereits im Jahr 2019 eine umfassende Aufstellung über den Stand der Technik zum Thema Fahrzeugsicherheit für Stufe 3- und Stufe 4-Systeme veröffentlicht. [15] Außerdem liegt bereits seit dem Jahr 2020 der ‚Technical Report‘ ISO/TR 4804 „Road vehicles — Safety and cybersecurity for automated driving systems — Design, verification and validation“ vor, der die sichere Entwicklung von Fahrsystemen der SAE-Stufen 3 und 4 adressiert [33] und zu einer ISO-Norm weiterentwickelt werden soll.

Neben diesen fahrzeugspezifischen Ansätzen sind auch Standards in der Entwicklung, welche sich mit KI-Funktionen/Systemen im Fahrzeugumfeld beschäftigen. [34] [35] Dazu gehören u.a. ISO/AWI PAS 8800 „Road Vehicles – Safety and Artificial Intelligence“, welche die Definition geeigneter Sicherheitsprinzipien, Methoden und Nachweise zur Erfüllung der Ziele von ISO 26262 und SOTIF festzulegen beabsichtigt, sowie ISO/IEC AWI 5469 „Artificial intelligence — Functional safety and AI systems“ [36] [37]. Ebenso wird im Rahmen der ISO/IEC FDIS 22989 „Information technology — Artificial intelligence — Artificial intelligence concepts and terminology“ an einer Taxonomie für KI gearbeitet. [38]

3 Blick über die Sektor-Grenze: Flugverkehr

Wie soeben aufgezeigt wurde, finden sich im Automobil-Sektor bisher noch keine verbindlichen Vorgaben oder Leitlinien zur funktionalen Sicherheit KI-basierter Fahrzeugsysteme oder zur KI-bezogenen Nachweisführung. Dies stellt die Hersteller solcher Systeme derzeit noch vor besondere Herausforderungen, da ihnen Orientierungen weitgehend fehlen. Da dies außerdem für andere Mobilitäts-Sektoren – etwa den Bahn-Sektor [39] – gleichermaßen gilt, kann sich ein Blick über die Sektoren-Grenze hinaus lohnen, um zu schauen, ob es dort Entwicklungen gibt, auf die rekrutiert werden kann. Dies soll nachfolgend zum Flugverkehrs-Sektor geschehen, da sich dieser als Vergleich-Sektor anbietet. Zum einen geht es auch dort um die Beförderung von Passagieren, Post (künftig auch mit Drohnen) und Cargo und zum anderen sind die Sicherheitsanforderungen an den Flugverkehr wegen der besonderen Gefährdungslagen (Geschwindigkeit, Wetterbedingungen, Sichtverhältnisse, künftig auch in Interaktion im urbanen Luftraum etc.) hoch. Schließlich wurde die automatisierte Unterstützung des menschlichen Piloten beim Führen des Flugzeugs schon frühzeitig eingeführt, sie ist dort bereits seit Jahren Stand der Technik; für die Zukunft sind ferner auch unbemannte Luftverkehrsfahrzeuge (z.B. Lufttaxis/eVTOL) geplant.

Insofern soll nachfolgend zunächst der sicherheitstechnische Zulassungsrahmen für Luftverkehrsfahrzeuge kurz dargestellt werden, bevor auf die sektorspezifischen Besonderheiten der verbauten autonomen und KI-basierten Systeme näher eingegangen wird.

3.1 Regelungsrahmen zur funktionalen Sicherheit

Für die Musterzulassung von bemannten und unbemannten Luftfahrzeugen für den Luftverkehr ist nach der Luftfahrt-Grundverordnung (Luft-GVO; VO EU 2018/1139), bis auf wenige Ausnahmen, [40] die EASA zuständig (Art. 77), die die Lufttüchtigkeit und die funktionale Sicherheit der Flugzeuge sowie der darin verbauten Systeme auf der Basis europäischer Vorschriften bewertet. Die Luft-GVO stellt abstrakte Anforderungen an die Sicherheit von Flugzeugen und Systemen und begründet in Verbindung mit der VO zu Lufttüchtigkeitszeugnisse (VO EU 748/2012) eine generelle Nachweisverpflichtung.

Die technischen Anforderungen an bestimmte Flugzeugtypen sind recht kleinteilig in den für den jeweiligen Flugzeugtyp geschaffenen Zertifizierungsgrundlagen, den sog. Zertifizierungsspezifikationen (Certification Specification – CS) geregelt, die von der EASA erlassen werden. Die CS enthalten aus Sicht der EASA die notwendigen Anforderungen für die sichere Konstruktion und Funktion des entsprechenden Luftfahrzeugs und haben verbindliche Wirkung (Art. 116 Luft-GVO). Auf Basis der festgelegten CS erlässt die EASA in einem zweiten Schritt „annehmbarer Nachweisverfahren“ (sog. Acceptable Means of

Compliance – AMC), mit denen die CS vor allem in technischer Hinsicht konkretisiert wird. [41] Im Unterschied zu den CS haben die AMC nur eine unverbindliche Wirkung, wenngleich ein Antragsteller, der die Vorgaben des AMC erfüllt, keinen weiteren Nachweis im Hinblick auf die Lufttüchtigkeit seines Fahrzeugs erbringen muss. Da ein Abweichen von den Empfehlungen der EASA ohnehin einen erheblichen Begründungsaufwand nötig macht, findet das Abweichen in der Praxis auch nur ausnahmsweise statt.

Die CS, als umfassende Bauvorschriften, bilden die funktionale Sicherheit ab. Sie sind nach Typen festgelegt: „CS 25 large airplanes“ spezifiziert die Zulassung von turbinengetriebenen Großflugzeugen und deren verbauten Systemen, wie CS 25.735 – Bremsen und Bremsanlagen oder CS 25.733 – Fahrwerkbereifung. Daneben formuliert die CS 25.1309 „Equipment, systems and installations“ systemübergreifend die quantitativen Anforderungen an die Sicherheitsstandards der Systeme. Dabei werden an die, für die Durchführung des Fluges notwendigen Systeme – in Abgrenzung zu nicht notwendigen Systemen wie Infotainmentsystemen – strenge quantitative Funktionstüchtigkeitsanforderungen geknüpft. Hierfür werden fehlerabhängig quantitative Grenzen nach der Schwere des zu erwartenden Schadens festgeschrieben (ähnlich wie dies auch beim Zugverkehr in der CSM-RA angelegt ist).

Die AMC formulieren Vorgaben von Methoden, die es den Herstellern ermöglichen, den Nachweis betreffend der Einhaltung der Anforderungen der CS zu erbringen. Für verbaute Systeme bietet z.B. die AMC 25.1309 ‘System Design and Analysis’ unter anderem grobe Vorgaben und Richtlinien für das richtige Safety Assessment.

Neben diesen Regelungsrahmen treten zur weiteren technischen Konkretisierung industrie-seitige Standards, die die sichere Entwicklung von Flugzeugen und Flugzeugsystemen begleiten und in der Regel von EASA zur Konkretisierung technischer Spezifikationen oder Nachweismethoden anerkannt werden, wie etwa die Standards zur System Development Assurance von komplexen Systemen zur Vermeidung von Entwicklungsfehlern (ARP 4754 A oder ED-79A „Guidelines for Development of Civil Aircraft and Systems“ (2010)). Ein ähnliches Ziel verfolgt der ED-12C oder der DO-178C „Software Considerations in Airborne Systems and Equipment Certification“ (2011). Für datengetriebene Prozesse kann der ED-76A oder der DO-200B zusätzlich einschlägig sein, ein „Standard for Processing Aeronautical Data“, welcher unter anderem Anforderungen an Datenqualität festlegt sowie Vorgaben bezüglich der Methoden zur Datenverarbeitung enthält.

CS, AMC sowie industrieseitige Standards sind in ihrem Zusammenwirken durchaus vergleichbar zu den Regelungen, welche die Automobilbranche dominieren. Sie sind technologieoffen und noch vor dem Hintergrund konventioneller Software entwickelt, weshalb KI-spezifische Aspekte bislang nicht vorkommen. Insbesondere für die Sicherheitsnachweisführung beschäftigt sich der ED-12C / DO-178C mit traditioneller „Development Assurance“ und deckt damit KI-spezifische Besonderheiten, welche sich unter anderem aus dem maschinellen Lernprozess ergeben, nicht ab. [42] Insgesamt lässt sich sagen, dass auch für den Bereich des Flugverkehrs Regelungen und Standards speziell zur Sicherung von KI-spezifischen Gefährdungen noch ausstehen. [44] [45]

3.2 Aktuelle Aktivitäten der Regelsetzer im Flugverkehrs-Sektor

3.2.1 Lösungsansätze der EASA

Bereits im April 2021 hat die EASA ein ‚Concept Paper‘ publiziert, welches u. a. bisherige Regelungslücken durch sicherheitstechnische Besonderheiten von KI adressiert, einen Leitfaden für die Entwicklung aufzeigt und die Zulassung von KI (Stufe 1) nach eigener Aussage ermöglicht. [43] Um die Zulassung von KI speziell für den Einsatz in Flugsystemen zu ermöglichen, werden Besonderheiten im ‚Safety Assessment‘ adressiert, u.a. finden sich darin Aussagen zu Unsicherheiten im Lernprozess und zu auftretenden Veränderungen durch den Wechsel zwischen Trainings- und Inferenzplattform. Darüber hinaus hat die EASA den für KI-Anwendungen so sicherheitskritischen ‚Learning Assurance‘-Prozess mithilfe von 38 ‚Objectives‘ und zugehörigen, erläuternden ‚Means of Compliance‘ in eine von offizieller Seite anerkannte Struktur gegossen und klarer strukturiert. Hierdurch wird die Rechts- und Planungssicherheit gerade im Entwicklungsprozess von KI-Systemen erheblich gesteigert.

Die Besonderheit dieser Leitlinie liegt vor allem aber auch in der stufenweisen Einteilung der KI-Anwendungen, einer weitergehenden Klassifizierung nach der Kritikalität von KI. Die Einteilung erfolgt je nach Leistungsvermögen in Abhängigkeit zum Menschen (human centric approach) in drei Stufen. [43] Künstliche Intelligenz auf Stufe 1 adressiert insbesondere Systeme, die dem Menschen bei der Ausführung seiner Tätigkeiten assistieren. So erkennt KI in einem ‚runway detection system‘ aus aufgenommenen Bildern das Vorhandensein und die Eckpunkte einer Landebahn und gibt diese Information an einen Piloten weiter. [43] Auf Stufe 2 kooperieren das KI-System und der Mensch in der Ausführung einer Funktion, etwa indem das ‚runway detection system‘, zusätzlich mit einem Autopiloten verbunden und durch den eingriffsbereiten Mensch überwacht, die Landung durchführt. [43] Auf Stufe 3 führt die Funktion autonom Tätigkeiten aus, während der Mensch vor allem durch die Überwachung des Entwicklungsprozesses sowie durch die vorgeschriebene Produktbeobachtung seiner Aufsichtspflicht nachkommt. Ein Beispiel für eine Applikation der Stufe 3 ist die Zukunftsvision eines vollautonomen Lufttaxis, welches Personen eigenständig zu steigen lässt und sie selbstständig zu ihrem Zielort befördert.

Bisher wurde durch die EASA vor allem die KI-Stufe 1 zusammen mit den Zulassungsvooraussetzungen in dem genannten ‚Conpet paper‘ definiert und erörtert. Genaue Spezifikationen zu Stufe 2 und 3 werden mit weiteren Zulassungs-Leitlinien in den kommenden Jahren folgen.

3.2.2 Standardisierung

Parallel zu den Bestrebungen der EASA arbeiten die Normungsinstitute an Lösungen für die sicherheits- und nachweistechnischen Probleme die sich aus den Besonderheiten der KI gegenüber konventioneller Software ergeben. Für den Flugverkehrssektor befinden sich zurzeit ein ‚Process Standard for Development and Certification of Aeronautical Safety-Related Products Implementing AI‘, ein Standard zu ‚Use Case Considerations‘ bezüglich ‚Artificial Intelligence in Aeronautical Systems‘ sowie eine Taxonomie zu ‚Artificial Intelligence in Aeronautical Systems‘ in Entwicklung. [44] Allesamt Standards, die die Spezifika von KI-basierten Systemen ansprechen.

4 Übertragbarkeit auf den Automobil-Sektor?

Die Entschlossenheit, die der Luftverkehrssektor an den Tag legt, ist beachtlich. Der KI-Bezug sowie die sektorspezifische Klassifizierung der funktionalen Sicherheit nach Kritikalität gefällt; ebenso die damit einhergehenden Veränderungen zu den Methoden der Nachweisführung in den AMC. Für den Automobil-Sektor existiert derzeit keine derartige Kategorisierung der KI-basierten Systeme nach Kritikalität. Automatisierungsfunktionen – mit und ohne KI – werden international grundsätzlich nach der Taxonomie SAE J3016 von SAE-Stufe 0 bis SAE-Stufe 5 („No Automation“ bis „Full Automation“) in sechs Stufen eingeteilt. Dabei kann ein Kraftfahrzeug mit einem Fahrerassistenzsystem ausgestattet sein, welches die Fähigkeit besitzt, mehrere Automatisierungsfunktionen auf verschiedenen Stufen zu realisieren. Die bisherige Taxonomie beschreibt nur, welche Leistungsanforderungen der Hersteller mit der Fahrfunktion verbindet, die er anschließend einer bestimmten SAE-Stufe informativ zuweist. Obendrein bezieht sich diese Taxonomie ohne Normcharakter lediglich auf eine begrenzte Anzahl von Fahrfunktionen, welche die Rolle des menschlichen Fahrzeuginspektors bei der Ausführung eines Teils oder der gesamten Fahraufgabe (DDT) ändern. [45] Fahrfunktionen, welche lediglich kurzzeitig und/oder notfallmäßig eingreifen, wie Notbrems- oder Spurhalteassistenten, fallen ausdrücklich nicht unter den Anwendungsbereich der Taxonomie, obschon solche Notfallfunktionen oftmals besonders sicherheitskritische KI-Komponenten enthalten können. Hier scheint bezüglich der Zuweisung der Fahrfunktionen eine kritische Überprüfung erforderlich.

Darüber hinaus sollten alle KI-Systeme, ähnlich dem Luftverkehrs-Sektor, anhand ihrer Kritikalität in Stufen eingeteilt werden. Hiervon könnte der Zulassungsprozess gerade im Hinblick auf den Nachweis der funktionalen Sicherheit KI-bezogener Systeme profitieren. Leitlinien könnten die Zuordnung erklären und kommentieren. Eine solche Stufeneinteilung könnte ein übersichtliches und erstes grobes Bild davon zeichnen, welche Art von KI-Anwendungen aktuell zu welchen Gefährdungen führen und diese entweder einem höheren oder niedrigeren Level zugeordnet werden. In den jeweiligen Kritikalitätsstufen würden die Sicherheitsmaßnahmen sowie Methoden zum Sicherheitsnachweis festgelegt. Je nach Zuordnung wären anschließend die Voraussetzungen an die funktionale Sicherheit und den Nachweis zu definieren; in einem hohen Kritikalitätslevel wäre beispielsweise der Black-Box-Effekt (mangelnde Nachvollziehbarkeit der systemimmanenten Prozesse) nicht hinnehmbar, während er sich in einem niedrigeren Kritikalitätslevel als nicht beachtenswert erweisen könnte. So könnten die Lücken in der Nachweisbarkeit der funktionalen Sicherheit von KI-Systemen stufenweise durch Leitlinien adressiert werden. Insoweit könnte die Automobil-Branche von einem Blick über die Sektorengrenzen profitieren, zumal sich diese Vorgehensweise nahtlos in das derzeit vorgeschlagene europäische Regelungssystem zur KI (siehe oben 2.2.2) integrieren ließe und dieses Regelungsregime für den Automobil-Sektor sinnvoll ergänzen würde. Die bisherige Vorgehensweise, die Systeme gleicher Endfunktion ohne Ansehen der verwendeten Technik und unabhängig von einer KI-Kritikalitätskaskade zu betrachten, ist auf Dauer und in Ansehung der europäischen Gesetzgebungsvorhaben nicht ausreichend. Für den Flugverkehr ist – wie erläutert – eine solche Einteilung

von KI-Systemen sowohl von öffentlicher Seite durch die EASA als auch durch industrie-seitige Standardisierungsorganisationen (vgl. „AIR6987 / ER-xxx Artificial Intelligence in Aeronautical Systems: Taxonomy“) vorgesehen.

Mithilfe der Stufeneinteilung über die Kritikalität wird die Einführung von KI in den Realverkehr durch die Sicherheitsbehörde unter Beratung industrie-eigener Stakeholder engmaschig begleitet und so auch rechtlich mit abgesichert.

5 Fazit

Obwohl durch diverse Stakeholder in den letzten Jahren die Lücken bezüglich einer Regelung zur funktionalen Sicherheit von KI-Systemen und der Nachweisführung mit der Absicht thematisiert wurden, die Sicherheit für Mobilitätanwendungen zu stärken, [15] beginnt für den Straßenverkehr gerade erst die rechtswissenschaftliche Debatte. Zwar hat die Thematik mit der Novelle des StVG vom Juni 2021 im wahrsten Sinne des Wortes „Fahrt“ aufgenommen, [17] [46] [47] dennoch bleiben die Aspekte der funktionalen Sicherheit von KI-basierten Systemen eher defizitär. Aus diesem Grund lohnt sich ein komparativer Blick über die Grenze der Sektoren hinaus, in den Flugverkehrs-Sektor. Im Unterschied zum Automobil-Sektor liefert der Flugverkehrs-Sektor zur Zulassung von KI-basierten Systemen der Stufe 1 – niedrigster Kritikalitätslevel – bereits über einsatzbereite, durch die Zulassungsbehörde veröffentlichte Leitlinien zur Sicherheitsnachweisführung. Von der Regelungssystematik und der damit einhergehenden Praxiserfahrung könnte auch der Automobil-Sektor profitieren. Die Leitlinien der EASA könnten, wenn nicht als Blaupause, zumindest als Anregung für Regelungen im Automobil-Bereich herangezogen werden, um eine zeitnahe Zulassung KI-basierter Systeme nach geregelten Standardverfahren unter Berücksichtigung ihrer Kritikalität zu ermöglichen.

Literaturverzeichnis

- [1] BMVI, Verkehrssicherheitsprogramm 2021 bis 2030, abrufbar unter: <https://www.bmvi.de/SharedDocs/DE/Artikel/StV/Verkehrssicherheit/verkehrssicherheitsprogramm-2021-bis-2030.html> (05.10.2021).
- [2] Bansal/Krizhevsky/Ogale, ChauffeurNet: Learning to Drive by Imitating the Best and Synthesizing the Worst, Waymo Reaserch, abrufbar unter: <https://arxiv.org/pdf/1812.03079.pdf> (01.10.2021).
- [3] BMVI, Ofizieller Internetauftritt des BMVI, abrufbar unter: <https://www.bmvi.de/SharedDocs/DE/Artikel/DG/automatisiertes-und-vernetztes-fahren.html> (19.01.2021).
- [4] BT, Drs. 18/11300, Gesetzentwurf der Bundesregierung eines Gesetzes zur Änderung des Straßenverkehrsgesetzes vom 20.02.2017.

- [5] ISO, ISO 26262:2018 Road vehicles - Functional safety, abrufbar unter: <https://www.iso.org/standard/68383.html> (01.10.2021).
- [6] Ross, Funktionale Sicherheit im Automobil, 2. Auflage, Hanser.
- [7] Hosse/Krumbach, SOTIF -Bedeutung der Sollfunktion in der Serienentwicklung, ACIMobility Summit Braunschweig, September 2021.
- [8] ISO, Abstract ISO/PAS 21448:2019, abrufbar unter: <https://www.iso.org/standard/70939.html> (06.05.2021).
- [9] Kirovskii/Gorelov, Driver assistance systems: analysis, tests and the safety case. ISO 26262 and ISO PAS 21448, 2019/OP Conf. Ser.: Mater. Sci. Eng. 534 012019, S. 7, abrufbar unter: <https://iopscience.iop.org/article/10.1088/1757-899X/534/1/012019> (25.05.2021).
- [10] ISO, ISO / PAS 21448 Road vehicles - Safety of the intended functionality, abrufbar unter: <https://www.iso.org/standard/70939.html> (04.10.2021).
- [11] Handelsblatt, Erster Toter in selbstfahrendem Auto, Beitrag vom 30.06.2016, abrufbar unter: <https://www.handelsblatt.com/unternehmen/industrie/us-behoerden untersuchen-tesla-unfall-erster-toter-in-selbstfahrendem-auto/13815470.html?ticket=ST-8834208-er5l4IVwcByEvCkQpsWs-ap5> (05.10.2021).
- [12] ISO, Abstract ISO/PAS 21448:2019, abrufbar unter: <https://www.iso.org/standard/70939.html> (06.05.2021).
- [13] Sari, Fail-operational Safety Architecture for ADAS/AD Systems and a Model-driven Approach for Dependent Failure Analysis, 1. Auflage 2020, Springer Vieweg.
- [14] DEEL Dependable and Explainable Learning, Machine Learning in Certified Systems , White paper 2021.
- [15] Aptiv Services US/AUDI AG/Bayrische Motoren Werke AG/Beijing Baidu Netcom Science Technology Co./Continental Teves AG & Co oHG, Safety First for Automated Driving, abrufbar unter: <https://www.daimler.com/innovation/case/autonomous/safety-first-for-automated-driving.html> (06.10.2021).
- [16] Gatzke, Zulässigkeitsvoraussetzungen für den Betrieb autonomer Kraftfahrzeuge - Bringt das Gesetz zum autonome Fahren Rechtssicherheit?, NZV 2022, 62, 66.
- [17] Kleemann/Arzt, Das Gesetz zum "autonomen" Fahren in Deutschland - Automatisiertes Fahren findet seinen Weg im StVG, RAW 2021 99, 99 ff.
- [18] UNECE, ECE/TRANS/WP.1/165, Resolution of the Deployment of Highly and Fully Automated Vehicles in Road Traffic vom 20.09.2018., S. 11 f.
- [19] BMDV, Autonome Fahrzeug-Genehmigungs- und Betriebsverordnung-Entwurf, abrufbar unter: <https://fragdenstaat.de/dokumente/8397-blanko-vorlage-fuer-bmv/>, Bearbeitungsstand: 27.01.2021.

- [20] Wagner, Gesetz zum autonome Fahren - Streitpunkte im Gesetzgebungsverfahren, SVR 2021, 287, 287 ff.
- [21] Europäische Kommission, COM(2021) 206 final, Proposal for a regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain union legislative acts., abrufbar unter: <https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1623335154975&uri=CELEX%3A52021PC0206> (01.10.2021).
- [22] Bomhard/Merkle, Europäische KI-Verordnung – der aktuelle Kommissionsentwurf und praktische Auswirkungen, "RDI 2021, 276, 276 ff.
- [23] Ebers, Standardisierung Künstlicher Intelligenz und KI-Verordnungsvorschlag, RDI 2021, 588, 588 ff.
- [24] FRAV, GRVA-12-23, Guidelines and Recommendations concerning Safety Requirements for Automated Driving Systems.
- [25] German Delegation (Bast), FAV-26-07, Vision for FRAV final steps.
- [26] Technical Secretary, VMAD-24-01, Wrap-up of the 23rd WMAD IWG web-conference.
- [27] Co-chairs, VMAD-24-03-rev.3, New Assessment/Test Method for Automated Driving (NATM) - Guidelines for Validating Automated Driving System (ADS).
- [28] GRVA, ECE/TRANS/WP.29/GRVA/11, Report of the Working Party on Automated/Autonomous and Connected Vehicles on its eleventh session.
- [29] Secreteriat, Workingpaper of the Workshop on Artificial Intelligence and Vehicle Regulations, Stand des Workingpapers: 17.03.2022.
- [30] Wp.29, ECE/TRANS/WP.29/1159, Report of the World Forum for Vehiccle Regulations on its 184th session.
- [31] Secreteriat, GRVA-08-10, AC.2's response to GRVA requests.
- [32] DIN/DKE, Deutschen Normungsroadmap Künstliche Intelligenz, abrufbar unter: <https://www.dke.de/de/arbeitsfelder/core-safety/normungsroadmap-ki> (04.10.2021).
- [33] ISO, ISO/TR 4804:2020 Road vehicles — Safety and cybersecurity for automated driving systems — Design, verification and validation, abrufbar unter: <https://www.iso.org/standard/80363.html> (04.10.2021).
- [34] ISO, ISO/AWI PAS 8800 Road Vehicles — Safety and artificial intelligence, abrufbar unter: <https://www.iso.org/standard/83303.html> (17.01.2022).
- [35] ISO, ISO/IEC AWI TR 5469 Artificial intelligence — Functional safety and AI systems, abrufbar unter: <https://www.iso.org/standard/81283.html> (17.01.2022).
- [36] ISO TC22/SC32/WG14, GRVA-11-13, ISO PAS 8800 Road Vehicles - Safety and Artificial Intelligence,, abrufbar unter: <https://unece.org/transport/events/wp29grva-working-party-automatedautonomous-and-connected-vehicles-11th-session> (28.09.2021).

- [37] ISO, ISO/IEC AWI TR 5469 Artificial intelligence — Functional safety and AI systems, abrufbar unter: <https://www.iso.org/standard/81283.html> (06.10.2021).
- [38] ISO/IEC, FDIS 22989 'Information Technology - Artificial Intelligence - Artificial Intelligence concepts and technology', abrufbar unter: : <https://www.iso.org/standard/74296.html>(18 03 2022).
- [39] SR.CC, KI-bezogene Test- und Zulassungsmethoden, abrufbar unter: https://www.tu-chemnitz.de/wirtschaft/jura2/FuE_KI_SRCC.php und <https://www.smart-rail.cc/> (14.10.2021).
- [40] LBA, Musterzulassungen und Einzelstückzulassungen, abrufbar unter: https://www.lba.de/DE/Technik/Musterzulassungen/Musterzulassungen_node.html (22.02.2022).
- [41] Koschmieder, eVTOL-Lufttaxen – Aktuelle regulatorische Herausforderungen in den Bereichen Zulassung, Betrieb und Infrastruktur, EuZW 2020, 741, 742.
- [42] Daedalean/EASA, Concepts of Design Assurance for Neural Networks (CoDANN) Public Report Extract, abrufbar unter: <https://www.easa.europa.eu/document-library/general-publications/concepts-design-assurance-neural-networks-codann> (01.08.2021).
- [43] EASA, Concept Paper: First usable guidance for Level 1 machine learning applications – A deliverable of the EASA AI Roadmap, abrufbar unter: <https://www.easa.europa.eu/newsroom-and-events/news/easa-releases-consultation-its-first-usable-guidance-level-1-machine> (04.10.2021).
- [44] SAE International G-34, Artificial Intelligence in Aviation, abrufbar unter: <http://profiles.sae.org/teag34/> 804.10.2021).
- [45] SAE International, J3016 Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, abrufbar unter: https://www.sae.org/standards/content/j3016_202104/.
- [46] Schrader, Wohin steuert das autonome Fahrzeug - vorübergehend?, ZRP 2021, 109, 109 ff.
- [47] Lutz, Neue Vorschriften für das automatisierte und autonome Fahren – ein Überblick, DAR 2021, 182, 182 ff.

YOLinO: Echtzeitfähige Schätzung von linienförmigen Kartenelementen im Kontext des automatisierten Fahrens

Annika Meyer* und Christoph Stiller †

Zusammenfassung: Für viele Anwendungen im Bereich des automatisierten Fahrens fehlt es bisher an einer geeigneten, echtzeitfähigen Architektur, die Kartenmerkmale darstellen und damit eine Lokalisierung in Karten, kartenloses Fahren und auch Ende-zu-Ende-Ansätze unterstützen kann. Wir präsentieren daher YOLinO, eine neuronale Netzarchitektur, die mit mehr als 150 Bildern pro Sekunde Kartenmerkmale detektieren kann. Dies können neben Fahrstreifenrändern jegliche geometrischen Elemente einer Karte sein, die sich aus Liniensegmenten darstellen lassen, bspw. Markierungen und Bordsteine. Wir zeigen zudem, dass auch implizite Merkmale wie Mittellinien von Fahrstreifen schätzbar sind.

Schlüsselwörter: Liniendetektion, Deep Learning, Kartenmerkmale

1 Einleitung

Im automatisierten Fahren werden – wie in vielen anderen Bereichen – immer mehr Aufgaben durch tiefe neuronale Netze (NNs) übernommen. Während für viele Aufgaben, wie Objektdetektion oder pixelweise Segmentierung, bereits etablierte Repräsentationen existieren, fehlt eine passende Repräsentation für Linien und Linienzüge bislang.

Linienbasierte Objekte spielen aber für die Wahrnehmung und ein Umgebungsverständnis im automatisierten Fahren eine entscheidende Rolle: Nicht nur enthalten HD-Karten mit Fahrstreifenrändern, Markierungen oder Bordsteinen linienförmige Elemente [1]. Auch kartenlose, Ende-zu-Ende-gelernte Ansätze profitieren von einer geeigneten, kartenverwandten Repräsentation der Umgebung [2].

Aktuell werden Linien oft durch eine klassifizierte Menge von Pixeln repräsentiert. Diesen fehlt jedoch der geometrische Zusammenhang, den Linien aufweisen. Dies zeigt sich insbesondere bei der Repräsentation impliziter oder kreuzender Linien. Alternativ wird auf langsame, iterative Verfahren zurückgegriffen, die weit von Echtzeitfähigkeit entfernt sind, schlecht konvergieren und/oder nicht generisch anwendbar sind.

Dieser Beitrag stellt daher eine Linien-Repräsentation für tiefe neuronale Netze vor, die diese Probleme überwindet und eine echtzeitfähige Wahrnehmung und Darstellung von Linien und linienähnlichen Elementen erlaubt.

* Annika Meyer, M.Sc., ist wiss. Mitarbeiterin am Institut für Mess- und Regelungstechnik am Karlsruher Institut für Technologie, Kontakt: annika.meyer@kit.edu.

† Prof. Dr.-Ing. Christoph Stiller leitet das Institut für Mess- und Regelungstechnik am Karlsruher Institut für Technologie

2 Stand der Forschung

Häufig wird die Erkennung von Fahrstreifenrändern, Markierungen, der Fläche des Ego-Fahrstreifens [3] oder der Nachbar- und Gegenfahrstreifen [4] als *semantische Segmentierung* umgesetzt. Teilweise wird auch nur ein möglicher Fahrkorridor prädiziert [5]. Semantischer Segmentierung fehlt jedoch der geometrische Zusammenhang. Eine direkte Schätzung eines parametrischen Modells ist für das Problem deutlich zielführender und effektiver [6, 7]. Hierzu stellen [8] eine Architektur vor, die die Fahrstreifenränder nicht pixelweise klassifiziert, sondern als *Distanztransformation* zum nächsten Rand darstellt. Diese Form der Darstellung erleichtert eine Nachverarbeitung erheblich, ist jedoch in kreuzenden Szenarien nicht eindeutig.

Ein häufiger, jedoch stark vereinfachter Ansatz für Autobahnszenarien ist es, die Fahrstreifen im Kamerabild zu bestimmen, indem entlang bestimmter *Bildzeilen* [9, 10, 11, 12] oder *Ankerlinien* [13, 14] die Position von Punkten geschätzt wird, die auf einem Fahrstreifenrand liegen. Dies ist jedoch nicht auf Kreuzungen übertragbar. Eine Aufweichung der Anker stellt eine *Spline-Regression* dar, bei der ein Bild in Subbilder unterteilt und für jedes Subbild ein Punkt geschätzt wird, der Teil der Markierung bzw. des Fahrstreifenrands ist [15]. Diese Punkte werden dann als Kontrollpunkte für einen Spline verwendet. Für eine generische Darstellung von Fahrstreifenrändern eignen sich *gerichtete Graphen* [16], die von einem rekurrenten Netz prädiziert werden. Dabei wird in jeder Iteration ein Knoten geschätzt, der einem Randpunkt auf einem Fahrstreifen entspricht. Diese Repräsentation kann teilende und fusionierende Fahrstreifen darstellen, ist jedoch nicht einfach übertragbar auf Kreuzungen. [17] beschreiben diese Methodik für Kreuzungsänder, jedoch nicht für die Fahrstreifen.

Die Forschung im Bereich der Fahrstreifenerkennung betrachtet häufig lediglich nicht-kreuzende Bereiche der Straße und lässt die Komplexität von Kreuzungen unbeachtet. Zur Bestimmung von *Kreuzungsgeometrien* ist es wichtig, dass ein Algorithmus zusammenführende Fahrstreifen repräsentieren und trotz Verdeckungen Schlüsse ziehen kann. Viele Ansätze scheitern jedoch bereits daran, dass im Kamerabild horizontal verlaufende Linien nicht repräsentierbar sind. Von den bisher genannten zeigen lediglich [17] die Anwendbarkeit auf Kreuzungszenarien, wobei hier nur der Rand der Straße nicht der Rand der Fahrstreifen geschätzt wird und somit die Komplexitätsprobleme der Kreuzungen nicht gelöst werden. Kreuzungen werden eher mit Hypothesenschätzern detektiert, die jedoch ihrerseits auf Detektionsalgorithmen wie YOLinO angewiesen sind [18, 19].

3 Generische Liniensegmentschätzung

YOLinO nutzt das von der YOLO-Architektur [20] bereits bekannte Prinzip der Unterteilung eines Bildes in einzelne Zellen, für die dann jeweils mehrere Objekthypothesen geschätzt werden. Dabei wird ein vortragsgerichtetes, tiefes neuronales Netz eingesetzt, das in einem einzigen Prädiktionsschritt das gesamte Bild verwendet, um pro Zelle mehrere Prädiktoren zu schätzen. Mit YOLinO werden dabei Hypothesen zu Segmenten eines Linienzuges prädiziert, die in die jeweilige Zelle fallen. Abbildung 1b zeigt ein Beispiel.

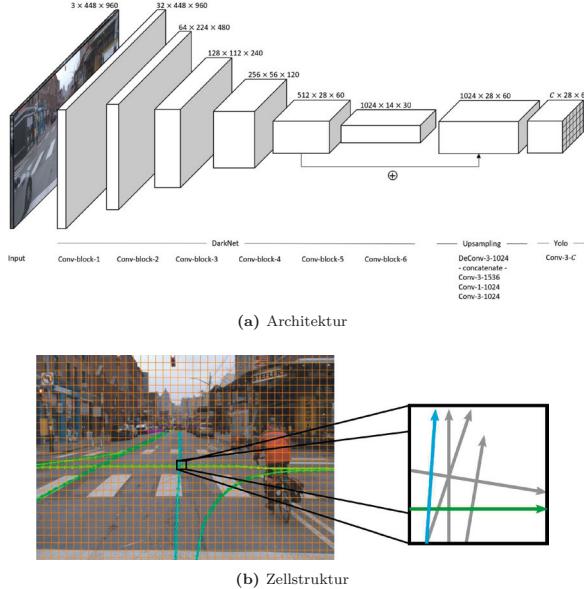


Abbildung 1: YOLinO-Prinzip: Für jede Zelle im Bild werden mehrere mögliche Liniensegmenthypthesen mit einer Konfidenz geschätzt. Zusätzlich kann für jedes Liniensegment implizit eine Richtung und eine Klassifikation prädiert werden. In b) werden hohe Konfidenzen farbig dargestellt und niedrige in grau. Bildquelle: Argoverse-Datensatz [21].

3.1 Architektur

Der Backbone für YOLinO ist generisch austauschbar, solange es sich um eine komprimierenden Encoder handelt. Für die Experimente in dieser Arbeit wird ein Darknet-19 [22] verwendet. Die Architektur (vgl. Abbildung 1a) besteht aus 19 faltenden Schichten (engl. convolutional) mit fünf Pooling-Schichten. Ein spezieller Decoder berechnet hieraus die dichte Linienrepräsentation im Zellgitter mit Zellen von jeweils 16×16 Pixel.

3.2 Linien-Repräsentation

Für jede Zelle wird eine durch die Architektur bestimmte Anzahl Prädiktoren geschätzt. Jeder Prädiktor $L = (g, c, p)$ ist definiert durch eine geometrische Beschreibung g , eine Klassenverteilung $c \in [0, 1]^{|C|}$ über alle Klassen C und eine Konfidenz $p \in [0, 1]$. Die Anzahl Prädiktoren pro Zelle bestimmt somit auch die maximale Anzahl an Liniensegmenten pro räumlichen Bereich.

Zur Repräsentation der Liniensegmente bestehen mehrere Optionen [23]. Ein Liniensegment wird bspw. über je einen Start- und einen Endpunkt als $L := (s, e)$ mit $s, e \in [0, 1]^2$ in kartesischen Koordinaten definiert. Der Ursprung des Koordinatensystems liegt dabei auf der oberen linken Ecke jeder Zelle. Das Koordinatensystem der Zelle wird dabei normiert, sodass die Zellbreite (und -höhe) genau 1 entsprechen. Durch die explizite

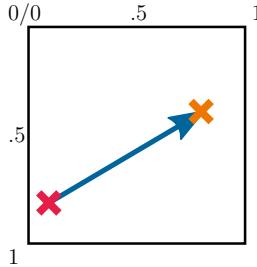


Abbildung 2: Segment mit Start- und Endpunkt

Formulierung als Start- und Endpunkt hat jedes Liniensegment eine implizite Richtung, die bspw. für Mittellinien als Fahrtrichtung interpretiert werden kann. Abbildung 2 zeigt die Repräsentation grafisch. Diese Darstellung bietet eine hohe Flexibilität, da Liniensegmente frei im Raum definiert werden können. Die Distanz zwischen zwei Liniensegmenten wird als euklidische Distanz definiert:

$$d(L, \hat{L}) := \|s - \hat{s}\|_2 + \|e - \hat{e}\|_2. \quad (1)$$

3.3 Verantwortlichkeit

Ein wichtiges Konzept für direkte, einstufige Prädiktionsverfahren mit neuronalen Netzen, bei denen mehrere Hypothesen in Form von Prädiktoren geschätzt werden, ist die Verantwortlichkeit dieser Prädiktoren [20]. Für ein optimales Ergebnis werden Prädiktoren innerhalb einer Zelle für unterschiedliche Typen von Linien trainiert. Das kann z.B. bedeuten, dass ein Prädiktor für horizontale Linien und ein anderer für vertikale Linien trainiert wird. Diese unterschiedliche Verantwortlichkeit wird durch ein geschicktes Design der Kostenfunktion ermöglicht, jedoch nicht explizit modelliert.

Die Verantwortlichkeit eines Prädiktors wird über die Distanz d zur Ground Truth (GT) bestimmt. Hat ein Prädiktor L_{ik} die kürzeste Distanz zu einer GT-Linie GT_{ij} in derselben Gitterzelle G_i so ist er für diese verantwortlich. Dies wird im Folgenden über eine Indikatorfunktion beschrieben:

$$\mathbb{1}_{ijk}^r = \begin{cases} 1 & \text{wenn } L_{ik} \text{ verantwortlich für } GT_{ij} \\ 0 & \text{sonst} \end{cases} \quad (2)$$

3.4 Kostenfunktion für Liniensegmente

Gleichungen 3-5 zeigen die für Liniensegmente angepasste Kostenfunktion aus [20]. Dabei werden in jeder Gitterzelle G_i für jede prädizierte Linie L_{ik} Kosten in Bezug zu den Liniensegmenten in der GT berechnet.

$$\mathcal{L}_{loc} = \sum_{i=1}^{|G|} \sum_{j=1}^{|GT_i|} \sum_{k=1}^{|L_i|} \mathbb{1}_{ijk}^r d(g_{ij}, g_{ik}) \quad (3)$$

$$\mathcal{L}_{conf} = \sum_{i=1}^{|G|} \sum_{k=1}^{|L_i|} (p_{ik} - \mathbb{1}_{ik}^r)^2 \quad (4)$$

$$\mathcal{L}_{class} = \sum_{i=1}^{|G|} \sum_{j=1}^{|GT_i|} \sum_{k=1}^{|L_i|} \mathbb{1}_{ijk}^r (c_{ik} - \hat{c}_{ij})^2 \quad (5)$$

Die geometrischen Lokalisierungskosten \mathcal{L}_{loc} fallen dabei nur an, wenn ein Prädiktor verantwortlich ist. Mit den beiden Konfidenzkostentermen \mathcal{L}_{conf} wird eine hohe Konfidenz entweder belohnt oder bestraft, je nachdem ob ein Prädiktor verantwortlich bzw. nicht verantwortlich ist. Die Klassifikationskosten des Prädiktors \mathcal{L}_{class} werden mit der quadratischen Distanz zwischen den prädizierten Pseudo-Wahrscheinlichkeiten c und dem One-Hot-Encoding¹ der GT \hat{c} berechnet. Die Gewichtung der einzelnen Kostenterme wurde empirisch bestimmt.

4 Non-Maximum Suppression

In direkten, einstufigen Prädiktionsverfahren mit neuronalen Netzen wird eine Vielzahl von Hypothesen inklusive einer Konfidenz für jede Hypothese geschätzt. Um nur die geeigneten Hypothesen zu erhalten, werden daher nur hinreichend konfidente Prädiktions ausgegeben.

Die YOLinO-Netzarchitektur prädiziert zur Laufzeit zu einem gegebenen Bild die passenden Prädiktions als Zellstruktur. Da jede Prädiktion als $L = (g, c, p)$ definiert ist, kann über $p > \tau_p$ gefiltert werden, sodass nur sicher geschätzte Prädiktions verarbeitet werden. Die gefilterten Prädiktions werden anschließend mit einem dichtebasierten Verfahren (DBSCAN [24]) geclustert und je Cluster ein Repräsentant berechnet.

Für den Clusteralgorithmus werden die Prädiktoren zunächst in einen geeigneten Raum transformiert. Hierfür wird jedes Liniensegment über $\tilde{L} = (m_u, m_v, l, d_u, d_v)^\top$ beschrieben, wobei m_u, m_v der Mittelpunkt des Liniensegments in Bildkoordinaten, l die Länge des Segments und d_u, d_v die normierte Richtung darstellen. Dazu wird zunächst die geometrische Beschreibung des Liniensegments in Bildkoordinaten $\tilde{g}_s, \tilde{g}_e \in [0, M] \times [0, N]$ umgerechnet und Gleichung 6-8 angewendet. Die Vorfaktoren λ_d, λ_ℓ und λ_m skalieren die Koordinaten und werden empirisch für jeden Datensatz bestimmt.

$$\begin{pmatrix} m_u \\ m_v \end{pmatrix} = \lambda_m \frac{\tilde{g}_e + \tilde{g}_s}{2} \quad (6)$$

$$\ell = \lambda_\ell \|\tilde{g}_e - \tilde{g}_s\| \quad (7)$$

$$\begin{pmatrix} d_u \\ d_v \end{pmatrix} = \lambda_d \frac{\tilde{g}_e - \tilde{g}_s}{\ell} \quad (8)$$

¹Vektor mit einer 1 an der Stelle der richtigen Klasse und ansonsten 0

Die resultierenden Cluster werden anschließend zu einem Repräsentanten zusammengefasst, der sich aus dem nach Konfidenz gewichteten Mittel der Clusterelemente ergibt. Die Konfidenz des Repräsentation wird aus dem Maximum des Clusters bestimmt.

5 Ergebnisse und Zusammenfassung

Für die Evaluation werden die prädictierten Linien abgetastet und pixelweise in den Raum (x, y, α) transformiert, der jeden Punkt mit der Pixelposition (x, y) und zudem die Richtung des Liniensegments α beschreibt. Mit dieser Darstellung werden alle Punkte als richtig-positiv betrachtet, die innerhalb eines Radius r als Nächstes zu einer Ground-Truth-Linie liegen. Daraus ergeben sich Recall, Precision und der F1-Wert.

Für den Vergleich mit dem TuSimple-Benchmark werden die prädictierten Liniensegmente nachverarbeitet. Dazu schlagen wir vor zunächst eine Connected-Component-Analyse durchzuführen, um die Instanzen zu identifizieren und einen Baum zu konstruieren. Auf jeder Ebene dieses Baums berechnen wir den mittleren Repräsentanten und legen einen kubischen Spline hindurch. Der Benchmark berechnet dann eine Genauigkeit (Acc.), Falsch-Positiv- (FP) sowie Falsch-Negativ-Rate (FN).

Methode	Acc	FP	FN	fps
LineCNN [9]	.969	.044	.020	17
PINet [10]	.958	.059	.033	40
LaneATT [12]	.967	.036	.018	250
ResNet-18 [11]	.961	.019 ²	.040 ²	312
YOLinO	.942	.188	.076	187

Tabelle 1: Unser Ansatz im Vergleich zu ausgewählten Arbeiten, die auf dem TuSimple Datensatz präsentiert wurden. Obwohl YOLinO nicht auf diese spezielle Aufgabe zugeschnitten ist, erreichen wir vergleichbare Ergebnisse. Im Hinblick auf die Inferenzzeit ist YOLinO unter den besten Beiträgen.

In [23] wurde eine ausführliche Parameterstudie präsentiert. Es kann gezeigt werden, dass auch mit 187 fps hohe Detektionsgenauigkeiten erreicht werden können. Auf dem *TuSimple* Datensatz [25] erreicht YOLinO State-of-the-Art-Ergebnisse (siehe Tabelle 1), ist aber gleichzeitig generalisierbar auf Anwendungen im Innenstadtbereich und insbesondere kreuzende Fahrstreifen. Auf dem Argoverse-Datensatz [21] zeigen wir Ergebnisse für eine *Mittellinienschätzung* mit einer Präzision von 41 %, Recall von 69 %, was zu einem F1-Wert von 52 % führt. Diese Art der Detektion wurde bisher in keiner anderen Arbeit gezeigt. Auch in einem *Luftbilddatensatz* [26] ist die Detektion und Klassifikation von Markierungen möglich. Hier erreichen wir einen F1-Wert 89 %, einen Recall von 90 % eine Präzision von 88 %. Beispiele sind in den Abbildungen 3, 4 und 5 dargestellt.

Der präsentierte Ansatz ermöglicht die Detektion von Kartenelementen, die als Linienzüge dargestellt sind. YOLinO approximiert beliebige linienförmige Darstellungen stückweise linear und prädiziert pro Zelle den Teil des Linienzuges, der in diese Zelle fällt. Dadurch wird eine dichte Darstellung ermöglicht, die keine Doppeldeutigkeit in der Zuordnung mit sich bringt und beliebige Linien approximieren kann. Im Gegensatz zu verwandten Arbeiten ermöglicht YOLinO, erstmals echtzeitfähig Kreuzungsgeometrien in Innenstädten zu beschreiben.

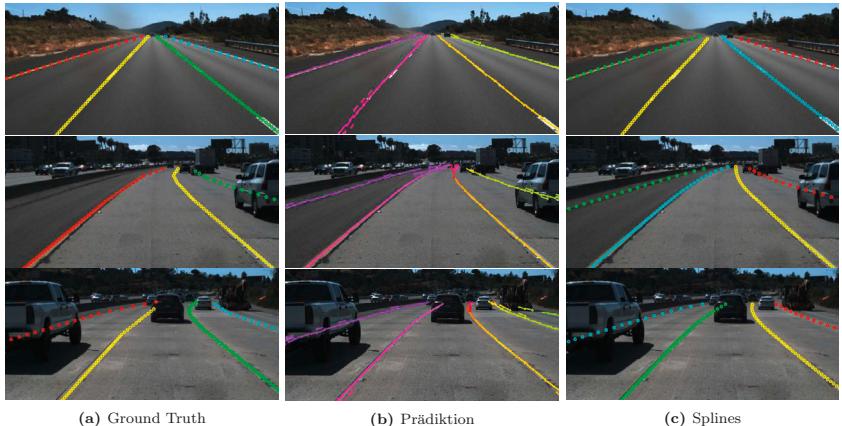


Abbildung 3: Ergebnisse auf dem TuSimple-Datensatz [25]. a) Ground Truth, b) Die Prädiktion nach der NMS. c) Gesampelte Splines, die in die YOLinO Prädiktion geschätzt wurden. Die Farben in (a) und (c) bezeichnen Instanzen. In (b) visualisiert die Farbe die Orientierung der prädizierten Liniensegmente.



Abbildung 4: Ergebnisse der Mittellinienschätzung auf dem Argoverse-Datensatz [21]. Die Farben zeigen hier die Orientierung der Liniensegmente und somit die Fahrtrichtung der Linien an. Besonders für Mittellinien ist es wichtig, dass YOLinO mehrere Linien pro Zelle schätzen und somit Kreuzungen repräsentieren kann.



Abbildung 5: Ergebnisse der Markierungsdetektion und -klassifikation in Luftbildaufnahmen [26]. Luftbilder: © Stadt Karlsruhe | Liegenschaftsamt

Literatur

- [1] F. Poggenhans, J.-H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, and M. Mayr, “Lanelet2: A high-definition map framework for the future of automated driving,” in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1672–1679, 2018.
- [2] J. Hawke, R. Shen, C. Gurau, S. Sharma, D. Reda, N. Nikolov, P. Mazur, S. Micklēthwaite, N. Griffiths, A. Shah, and A. Kendall, “Urban Driving with Conditional Imitation Learning,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 251–257, 2020.
- [3] G. L. Oliveira, W. Burgard, and T. Brox, “Efficient deep models for monocular road segmentation,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4885–4891, 2016.
- [4] A. Meyer, N. O. Salscheider, P. F. Orzechowski, and C. Stiller, “Deep semantic lane segmentation for mapless driving,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 869–875, IEEE, 2018.
- [5] D. Barnes, W. Maddern, and I. Posner, “Find your own way: Weakly-supervised segmentation of path proposals for urban autonomy,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 203–210, 2017.
- [6] S. M. Azimi, P. Fischer, M. Körner, and P. Reinartz, “Aerial LaneNet: Lane-Marking Semantic Segmentation in Aerial Imagery Using Wavelet-Enhanced Cost-Sensitive Symmetric Fully Convolutional Neural Networks,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, pp. 2920–2938, May 2019.
- [7] M. Ghafoorian, C. Nugteren, N. Baka, O. Booij, and M. Hofmann, “EL-GAN: Embedding Loss Driven Generative Adversarial Networks for Lane Detection,” in *The European Conference on Computer Vision (ECCV)*, pp. 256–272, 2019.
- [8] M. Bai, G. Mattyus, N. Homayounfar, S. Wang, S. K. Lakshmikanth, and R. Urtsasun, “Deep Multi-Sensor Lane Detection,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3102–3109, 2018.
- [9] X. Li, J. Li, X. Hu, and J. Yang, “Line-CNN: End-to-end traffic line detection with line proposal unit,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 248–258, 2020.
- [10] Y. Ko, Y. Lee, S. Azam, F. Munir, M. Jeon, and W. Pedrycz, “Key points estimation and point instance segmentation approach for lane detection,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2021.
- [11] Z. Qin, H. Wang, and X. Li, “Ultra fast structure-aware deep lane detection,” in *The European Conference on Computer Vision (ECCV)*, vol. 16, pp. 276–291, Springer, 2020.

-
- [12] L. Tabelini, R. Berriel, T. M. Paixão, C. Badue, A. F. De Souza, and T. Olivera-Santos, “Keep your eyes on the lane: Attention-guided lane detection,” *arXiv 2010.12035*, 2020.
 - [13] N. Garnett, R. Cohen, T. Pe'er, R. Lahav, and D. Levi, “3D-LaneNet: End-to-end 3D multiple lane detection,” in *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2921–2930, 2019.
 - [14] T. Suleymanov, P. Amayo, and P. Newman, “Inferring road boundaries through and despite traffic,” in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 409–416, 2018.
 - [15] Y. Huang, S. Chen, Y. Chen, Z. Jian, and N. Zheng, “Spatial-Temporal Based Lane Detection Using Deep Learning,” in *Artificial Intelligence Applications and Innovations, IFIP Advances in Information and Communication Technology*, pp. 143–154, 2018.
 - [16] N. Homayounfar, W.-C. Ma, J. Liang, X. Wu, J. Fan, and R. Urtasun, “DAGMapper: Learning to Map by Discovering Lane Topology,” in *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2911–2920, 2019.
 - [17] J. Liang, N. Homayounfar, W.-C. Ma, S. Wang, and R. Urtasun, “Convolutional Recurrent Network for Road Boundary Extraction,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9512–9521, 2019.
 - [18] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, “3D Traffic Scene Understanding From Movable Platforms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, pp. 1012–1025, May 2014.
 - [19] A. Meyer, J. Walter, M. Lauer, and C. Stiller, “Anytime Lane-Level Intersection Estimation Based on Trajectories of Other Traffic Participants,” in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2019.
 - [20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.
 - [21] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, “Argoverse: 3D Tracking and Forecasting With Rich Maps,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8748–8757, 2019.
 - [22] J. Redmon and A. Farhadi, “YOLO9000,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7263–7271, 2017.
 - [23] A. Meyer, P. Skudlik, J.-H. Pauls, and C. Stiller, “Yolino: Generic single shot polyline detection in real time,” in *IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 2916–2925, 2021.

- [24] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, “A density-based algorithm for discovering clusters in large spatial databases with noise.,” in *kdd*, vol. 96, pp. 226–231, 1996.
- [25] TuSimple, “Lane Detection Challenge.” abgerufen am 12.10.2021 von https://github.com/TuSimple/tusimple-benchmark/tree/master/doc/lane_detection, 2017.
- [26] J.-H. Pauls, T. Strauss, C. Hasberg, M. Lauer, and C. Stiller, “Can We Trust Our Maps?,” in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2639–2644, 2018.

Understanding the Domain Gap in LiDAR Object Detection Networks

Jasmine Richter Florian Faion Di Feng Paul Benedikt Becker
Piotr Sielecki Claudio Gläser *

Abstract: In order to make autonomous driving a reality, artificial neural networks have to work reliably in the open-world. However, the open-world is vast and continuously changing, so it is not technically feasible to collect and annotate training datasets which accurately represent this domain. Therefore, there are always domain gaps between training datasets and the open-world which must be understood. In this work, we investigate the domain gaps between high-resolution and low-resolution LiDAR sensors in object detection networks. Using a unique dataset, which enables us to study sensor resolution domain gaps independent of other effects, we show two distinct domain gaps - an inference domain gap and a training domain gap. The inference domain gap is characterised by a strong dependence on the number of LiDAR points per object, while the training gap shows no such dependence. These findings show that different approaches are required to close these inference and training domain gaps.

Keywords: neural networks, domain adaptation, object detection, LiDAR

1 Introduction

The open-world in which we live is diverse and constantly changing, so autonomous vehicles have to work robustly in frequently changing weather conditions [1], light conditions [2], and in different locations such as cities, highways and rural areas. In addition to this, the hardware used in autonomous vehicles is continuously being improved and updated. This trend is clearly observable in LiDAR sensors, which are continually being manufactured with higher and higher resolutions [3]. Collecting and labelling datasets covering the long-tail of open-world scenarios for every new sensor generation is prohibitively time-consuming and expensive [4]. Therefore, it is important to understand the domain gaps between datasets recorded with LiDARs of different resolutions - the sensor-to-sensor domain gaps - so that these domain gaps can be closed and datasets and models can be reused and improved.

Although domain adaptation has been widely studied [5, 6, 7, 8, 9, 10], to the best of our knowledge, this is the first time that the distinct inference and training domain gaps have been identified and investigated in LiDAR object detection networks. In this work, the inference domain gap refers to the domain gap observed when the network is trained with data from a particular LiDAR and then evaluated on data from the same LiDAR and a different LiDAR. Whereas the training domain gap refers to the domain

*Robert Bosch GmbH (e-mail: firstname.lastname@bosch.com).

gap observed when two networks are trained, the first with data from one LiDAR and the second with data from another LiDAR, and then both evaluated on data from one of those LiDARs. This distinction is only identifiable because, unlike other autonomous driving datasets [11, 12], our dataset enables us to study the sensor-to-sensor domain gaps directly and independently of other common domain gaps, such as from different weather conditions, light conditions and locations.

2 Dataset

In order to isolate the sensor-to-sensor domain gaps from other domain gaps, we used a unique dataset in which LiDAR data from high-resolution Hesai Pandar64P sensors and low-resolution Velodyne VLP-32C sensors was recorded simultaneously. This dataset was collected on a drive of 7,800 km, visiting 33 cities in 6 European countries. The route was chosen to maximise scenario variety and includes highway, rural and urban driving in diverse environmental conditions including day and night, as well as clear and adverse weather. For this work, we used around 13,100 single frames containing around 432,000 labelled bounding boxes in total.

The high-resolution and low-resolution LiDAR sensors were mounted in an alternating pattern on the roof of the car, as shown in Figure 1. These alternating mounting positions ensure that both the high-resolution and low-resolution sensors cover the full 360° field-of-view. The different sensor resolutions and fields-of-view cause an uneven distribution of points to be received from the objects in the dataset, with slightly more points received from the low-resolution sensors in the bottom-left and top-right of the BEV (where points are received from two low-resolution sensors and one high-resolution sensor) and significantly more points received from the high-resolution sensors in the top-left and bottom-right of the BEV (where points are received from two high-resolution sensors and one low-resolution sensor).

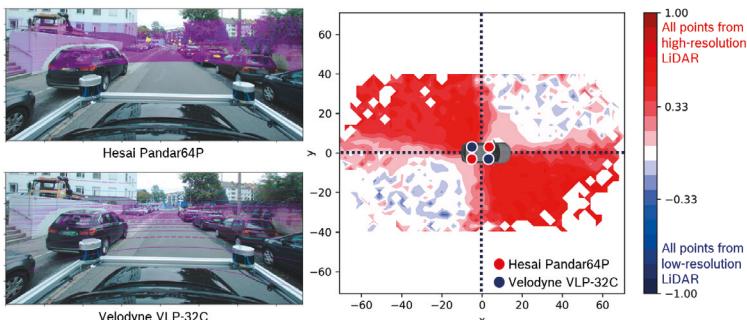


Figure 1: Configuration of the LiDAR sensors. Left: LiDAR points overlaid on camera images for the high-resolution Hesai Pandar64P sensors and low-resolution Velodyne VLP-32C sensors respectively. Right: Relative number of points received per object from the high-resolution Hesai Pandar64P sensors and low-resolution Velodyne VLP-32C sensors as a function of (x, y) location in the bird's eye view (BEV).

3 Experiments

Our experiments were carried out on an object detection network based on the PIXOR architecture [13]. Identical randomly initialised weights and hyperparameters were used for all trainings. First, we quantitatively compared relative average precisions. Then we explored how relative average precision varies spatially over the BEV. Finally, we investigated the relationship between recall and the average number of points received per object.

3.1 Relative average precision

In order to quantify the distinct inference and training sensor-to-sensor domain gaps, we trained and evaluated using all sensor combinations. The relative average precisions (APs) of these trainings and evaluations are shown in Table 1. The relative difference in the APs indicate a fundamental dissimilarity between the inference domain gap and the training domain gap.

Table 1: Relative APs of object detection networks trained and evaluated on different combinations of high-resolution and low-resolution LiDAR sensors. Higher is better. Top: Inference domain gap. Bottom: Training domain gap.

Trained on	Evaluated on	Relative AP	Trained on	Evaluated on	Relative AP
High-res	High-res	+25%	Low-res	High-res	+11%
High-res	Low-res	0%	Low-res	Low-res	0%
Trained on	Evaluated on	Relative AP	Trained on	Evaluated on	Relative AP
High-res	High-res	+7%	High-res	Low-res	0%
Low-res	High-res	0%	Low-res	Low-res	+6%

3.1.1 Inference domain gap

For the inference domain gap (trained on high-resolution and evaluated on high-resolution and low-resolution, or trained on low-resolution and evaluated on high-resolution and low-resolution), the relative AP is higher when the evaluation is carried out using the high-resolution LiDAR data, independent of whether the object detection network was trained with high-resolution or low-resolution LiDAR data. This shows that using a higher-resolution LiDAR sensor at inference time can improve the performance of the object detection network, even if the network was trained with data from a different LiDAR sensor.

3.1.2 Training domain gap

In contrast to the inference domain gap, for the training domain gap (trained on high-resolution and low-resolution, then evaluated on either high-resolution or low-resolution), the relative AP is higher when the evaluation is carried out using LiDAR data from the

same sensor that was used for training. In this case, there is no evidence that using a higher-resolution LiDAR sensor at training time leads to any improvement in the object detection network performance. In fact, it actually results in worse performance. The object detection network performs best when it is trained using data from the same LiDAR sensor that is used for the evaluation.

3.2 Spatial variation of relative average precision

In order to further understand the domain gaps observed in the relative APs, we explored how relative AP varies spatially over the BEV. Figure 2 shows the relative AP of all objects as a function of (x, y) location in the BEV, for the networks trained on high-resolution and low-resolution LiDAR data respectively. While Figure 3 shows the relative AP of all objects as a function of (x, y) location in the BEV, for the networks evaluated on high-resolution and low-resolution LiDAR data respectively.

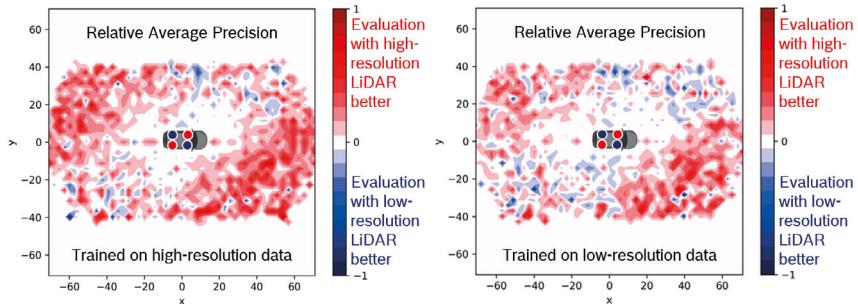


Figure 2: Relative AP of all objects as a function of (x, y) location in the BEV. Left: Trained on high-resolution LiDAR data. Right: Trained on low-resolution LiDAR data.

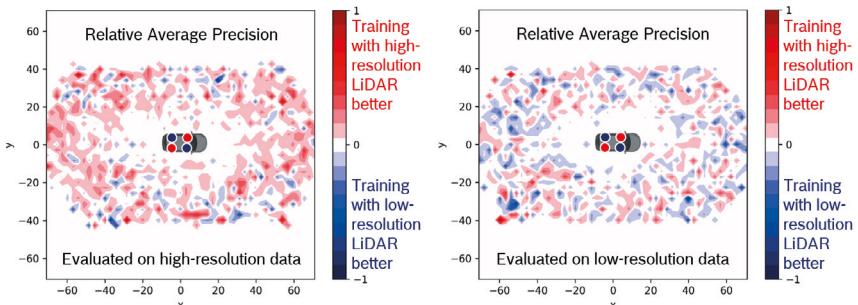


Figure 3: Relative AP of all objects as a function of (x, y) location in the BEV. Left: Evaluated on high-resolution LiDAR data. Right: Evaluated on low-resolution LiDAR data.

3.2.1 Inference domain gap

For the inference domain gap, independent of whether the network was trained on high-resolution or low-resolution LiDAR data, the relative AP is not evenly distributed over the entire BEV. Instead, for both plots in Figure 2, the relative AP is higher in the top-left and bottom-right of the BEV. This distribution of relative AP looks very similar to the distribution of relative points per object shown in Figure 1. Again, this indicates that no matter which LiDAR data is used for training, the performance of the object detection network can be improved at inference time by using a higher-resolution LiDAR sensor at inference time.

3.2.2 Training domain gap

In contrast to the inference domain gap, for the training domain gap, the relative AP is evenly distributed over the entire BEV for both plots in Figure 3 and there are no specific regions where the relative AP is higher or lower. Instead, when the networks trained on high-resolution and low-resolution data are evaluated on high-resolution data, the relative AP is generally higher for the object detection network trained on high-resolution data. The same is true when the network trained on low-resolution data is evaluated on low-resolution data. Again, this indicates that using a higher-resolution LiDAR at training time does not improve the performance of the object detection network. Instead, training with data from the same LiDAR sensor that is used for evaluation leads to the best object detection network performance.

3.3 Relationship between recall and average number of points received per object

Since both the relative APs and spatial variation of relative APs indicate a relationship between the number of points received from an object and the performance of the object detection network, we further explored this relationship by investigating how the recall varies with the average number of points received per object. Using recall, which is defined as the proportion of true positives discovered from the labelled real-world objects, enables us to understand how many points on average are required to correctly identify real-world objects. Such a metric is important for autonomous driving, because failing to identify real-world objects could pose a significant safety risk.

The relationships between recall and the average number of points received per object for training and evaluation on all sensor combinations is shown in Figure 4. For all training and evaluation combinations, the recall clearly increases as the average number of points received per object increases. This trend is especially noticeable in the region where less than 100 points are received on average per object. In general, this shows that the more points the object detection network receives from an object, the more likely it is that the object detection network is able to distinguish the object from the background.

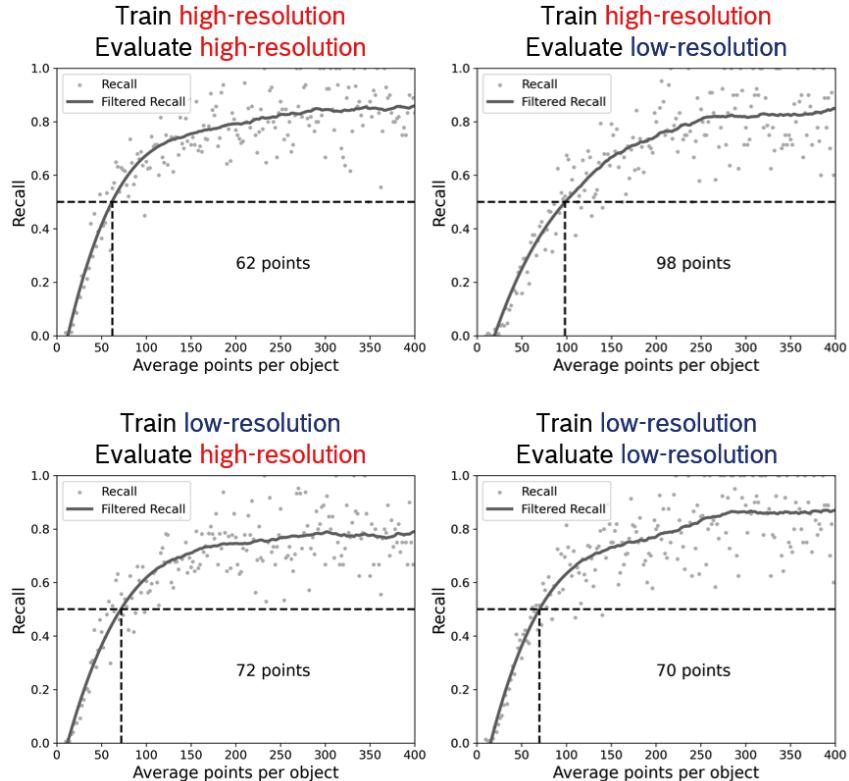


Figure 4: Recall versus average number of points received per object for training and evaluation on all sensor combinations. A recall of 1 means that all real-world objects were detected, while a recall of 0 means that no real-world objects were detected. All plots were filtered with a Savitzky-Golay filter in order to estimate the average number of points received per object required to achieve a recall of 0.5, which is indicated by the dashed black lines.

Each plot also shows the average number of points received per object required to achieve a recall of 0.5 for the given training and evaluation LiDAR sensor combination. These values are summarised in Table 2 for the inference domain gap and the training domain gap.

Table 2: Average number of points received per object required to achieve a recall of 0.5 for object detection networks trained and evaluated on different combinations of high-resolution and low-resolution LiDAR sensors. Lower is better. Top: Inference domain gap. Bottom: Training domain gap.

Trained on	Evaluated on	Points	Trained on	Evaluated on	Points
High-res	High-res	62	Low-res	High-res	72
High-res	Low-res	98	Low-res	Low-res	70
Trained on	Evaluated on	Points	Trained on	Evaluated on	Points
High-res	High-res	62	High-res	Low-res	98
Low-res	High-res	72	Low-res	Low-res	70

3.3.1 Inference domain gap

At first, the average number of points required to achieve a recall of 0.5 seems to contradict the relative APs in Table 1. For the model trained on high-resolution LiDAR data, 36 fewer points per object are required on average to achieve a recall of 0.5 when evaluating with the high-resolution instead of the low-resolution LiDAR sensor. However, for the model trained on low-resolution LiDAR data, two additional points per object are required on average to achieve a recall of 0.5 when evaluating with the high-resolution instead of the low-resolution LiDAR sensor, even though the relative AP in Table 1 is higher when evaluating with the high-resolution instead of the low-resolution LiDAR sensor. This indicates that it is not just the average number of points received per object that is important, but also whether these points were recorded with the high-resolution or low-resolution LiDAR sensor. At inference time, the object detection network can detect objects with fewer points when it was trained on data from the same LiDAR sensor. This again suggests that the inference gap can not only be closed, but even reversed by using a high-resolution LiDAR sensor at inference time, because of the additional points received per object. These additional points can be observed by plotting the average number of points received per real-world object versus the average distance of the real-world objects from the LiDAR sensor for the high-resolution and low-resolution LiDAR sensors, as shown in Figure 5. This observation explains why the combination of training and evaluating on high-resolution LiDAR data results in the largest performance improvement (relative AP of +25% shown in Table 1), because the object detection network requires fewer points on average per object to be able to detect real-world objects and the LiDAR data contains more points on average per object because of the higher-resolution.

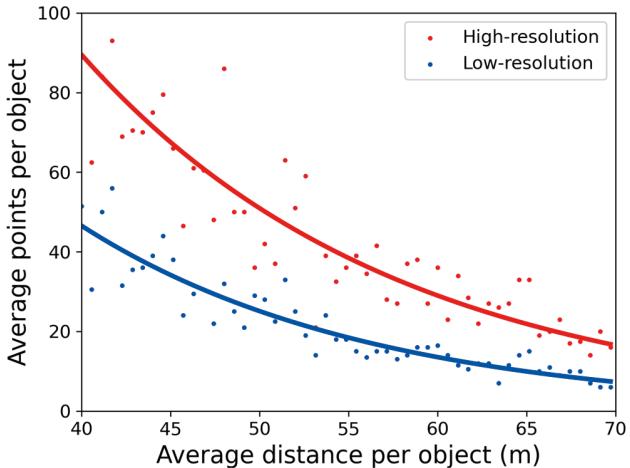


Figure 5: Average number of points received per real-world object versus average distance to the LiDAR sensor per real-world object, fitted with an exponential function. On average, more points are received per object from the high-resolution LiDAR sensor (red) than from the low-resolution LiDAR sensor (blue) and the number of points received per object decreases as the distance to the LiDAR sensor increases.

3.3.2 Training domain gap

For the training domain gap, when evaluating on high-resolution LiDAR data, 10 fewer points per object are required on average when the object detection network is trained on high-resolution rather than low-resolution LiDAR data. When evaluating on low-resolution LiDAR data, 18 fewer points per object are required on average when the object detection network is trained on low-resolution rather than high-resolution LiDAR data. This is consistent with the relative APs in Table 1, which show that training the object detection network on data from the same LiDAR sensor results in the best performance at inference time. This again indicates that the training gap cannot be closed by increasing the average number of points per object by using a higher-resolution LiDAR sensor at training time. Instead, the performance drops when a different LiDAR is used for training, independent of whether the LiDAR used for training is higher-resolution or lower-resolution than the LiDAR used at inference time.

4 Conclusions

In this work, we investigated sensor-to-sensor domain gaps in LiDAR object detection networks. Using a unique dataset, which enabled us to study the domain gaps independent of other effects, we showed that there are actually two distinct domain gaps - the inference domain gap and the training domain gap. The inference domain gap varies strongly with the average number of points per object at inference time and our experiments show that

increasing the average number of points per object by using a higher-resolution LiDAR sensor at inference time, can improve the performance of the object detection network, independent of which LiDAR data was used for training. This indicates that it is better to use a high-resolution LiDAR for inference, even if the network has been trained with data from a low-resolution LiDAR. In contrast, the training domain gap does not vary with the number of points per object. Our experiments show that adding more points to the objects at training time, by training with a higher-resolution LiDAR sensor, does not improve performance, but actually worsens the performance. In this case, if inference will be carried out with a low-resolution LiDAR, it is better to train with data from the same low-resolution LiDAR, even if a high-resolution LiDAR is available. These important findings show that different strategies are required to close these distinct inference and training domain gaps.

In the future, the same analysis could be applied to other sensor modalities [14], such as radar [15]. It could also be applied to other domain gaps, such as weather domain gaps, since it is known that significantly fewer LiDAR points are typically received in rainy weather than in dry weather [10]. In fact, our initial experiments on weather domain gaps indicate very similar behaviour to the sensor-to-sensor domain gaps, since we also observe a distinct inference domain gap which varies strongly with the average number of points per object at inference time and a distinct training domain gap which does not vary with the number of points per object. Again, this highlights the importance of using different strategies to close these distinct inference and training domain gaps.

Acknowledgements

The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Energy (BMWi) within the project “KI Delta Learning” (Förderkennzeichen 19A19013D). The authors would like to thank the consortium for the successful cooperation.

References

- [1] M. Hnewa and H. Radha, “Object detection under rainy conditions for autonomous vehicles: A review of state-of-the-art and emerging techniques,” *IEEE Signal Processing Magazine*, 2021.
- [2] A. Singha and M. K. Bhowmik, “Moving object detection in night time: A survey,” *2nd International Conference on Innovations in Electronics, Signal Processing and Communication (IESC)*, 2019.
- [3] X. Wang, H. Pan, K. Guo, X. Yang, and S. Luo, “The evolution of lidar and its application in high precision measurement,” *IOP Conference Series: Earth and Environmental Science*, 2020.
- [4] J. Li, Y. Wong, Q. Zhao, and M. S. Kankanhalli, “Learning to learn from noisy labeled data,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

- [5] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” *Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML)*, 2015.
- [6] L. Triess, M. Dreissig, C. Rist, and J. M. Zöllner, “A survey on deep domain adaptation for LiDAR perception,” *Proceedings of the IEEE Intelligent Vehicles Symposium (IV) Workshops*, 2021.
- [7] S. Wirges, S. Ding, and C. Stiller, “Single-stage object detection from top-view grid maps on custom sensor setups,” *IEEE Intelligent Vehicles Symposium (IV)*, 2020.
- [8] C. Rist, M. Enzweiler, and D. Gavrila, “Cross-sensor deep domain adaptation for lidar detection and segmentation,” *IEEE Intelligent Vehicles Symposium (IV)*, 2019.
- [9] C. Qin, H. You, L. Wang, C.-C. J. Kuo, and Y. Fu, “Pointdan: A multi-scale 3d domain adaption network for point cloud representation,” *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.
- [10] Q. Xu, Y. Zhou, W. Wang, C. R. Qi, and D. Anguelov, “SPG: Unsupervised domain adaptation for 3d object detection via semantic point generation,” *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.
- [11] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *International Journal of Robotics Research (IJRR)*, 2013.
- [12] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liou, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, “nuscenes: A multimodal dataset for autonomous driving,” *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [13] B. Yang, W. Luo, and R. Urtasun, “Pixor: Real-time 3d object detection from point clouds,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [14] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer, “Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges,” *IEEE Transactions on Intelligent Transportation Systems*, 2020.
- [15] N. Scheiner, F. Kraus, N. Appenrodt, J. Dickmann, and B. Sick, “Object detection for automotive radar point clouds - a comparison,” *AI Perspectives*, 2021.

General Panoptics: Combining Semantic Segmentation and Classical Methods for a Fast LiDAR Panoptic Segmentation

Lukas Beer and Hans-Joachim Wünsche*

Abstract: Detailed knowledge about the environment is a prerequisite for autonomous driving. One main question in this context is: Which objects are around me? The answer to this question contains two main parts: getting semantic and advanced geometric information. Thus, we need classified, individual instances with a unique ID, together with further information of the surrounding, such as a classification of the ground. This combination is generally known as panoptic segmentation. In this paper, we present a novel general approach for this task. Instead of focusing on an end-to-end network or a combination of object detection and semantic segmentation, our method combines semantic segmentation with a non-learning-based clustering. Both information sources are combined using a Neighborhood-Related Activation Function (NeRAF). This allows a general panoptic segmentation of everything in the surrounding - no matter of its class. We show that the proposed method can run with up to 45 fps on a mobile computer.

Keywords: Clustering, LiDAR, Panoptic Segmentation, Semantic Segmentation

1 Introduction

Since [1] recently introduced the task of panoptic segmentation (PS), it got more and more into focus of current research. Originally defined for images, PS combines semantic segmentation (assign a class to each pixel) and instance segmentation (detect and segment each instance). PS splits classes in two categories: countable objects (*things*) and amorphous regions (*stuff*). Not only limited to images, this task is also a hot topic with 3D point clouds. As a result, we receive individual, classified LiDAR instances.

These instances have several applications: besides the awareness of other traffic participants for motion planning, many of the current localization and SLAM algorithm need semantic landmarks for a precise positioning [2–4]. Nevertheless, most of the *thing* classes are considered to be traffic participants in recent works. Static objects such as trees, bushes, buildings and poles are ignored [5; 6] or only non-natural static objects (buildings, poles, signs) are seen as *things* [7]. To the best of our knowledge, static but natural objects, such as trunks, are not present in so far published LiDAR PS datasets.

In this work, we want to tackle this issue. Contrary to many other approaches [5; 6; 8–11], our method does not fully rely on deep learning. Instead, we combine the non-learning

*This research paper is funded by dtec.bw – Digitalization and Technology Research Center of the Bundeswehr [project MORE – Munich Mobility Research Campus].

The authors are with the Institute for Autonomous Systems Technology (TAS), Department of Aerospace Engineering, Universität der Bundeswehr München, Neubiberg, Germany. Lukas.Beer@unibw.de



(a) Instance output, each instance has a different color.
(b) Semantic output, each class has a different color.

Figure 1: Result of our approach: LiDAR scan from the KITTI dataset with semantic information (1a) and instance information (1b). A video of our approach can be seen on <https://www.mucar3.de/fas2022-general-panoptics>.

based clustering approach from [12] with the semantic segmentation network from [13]. Therefore, we do only need training data for the semantic segmentation.

Both parts are combined using a novel Neighborhood-Related Activation Function (NeRAF) in combination with a consensus-driven fusion as proposed in [8]. The activation function takes instance ID changes, label changes and the depth difference of two neighboring points into account and decides whether they belong to the same semantically or geometrically consistent object or not. In combination with a look-up table for defining the relations, we can simply tune the weights of each component in the activation function, depending on the classes of two neighboring points. With the help of NeRAF, we receive separated instances. For classifying them, we perform the consensus-driven fusion. The output of our approach can be seen in Fig. 1. Even geometrically close objects like fences and bushes can be split robustly.

Due to the high efficiency of all three parts (clustering, semantic segmentation and fusion) our approach runs with up to 45 fps on a consumer graphics card (NVIDIA GeForce GTX 1060).

Compared to previous approaches, the main advantages of this method are the generalized PS, which does not need dedicated PS training data together with the high efficiency while retaining adequate accuracy.

After we introduced PS in Section 1, we start in Section 2 with a short summary of related work in the field of PS. In Section 3 we explain our approach in detail. A quantitative and a qualitative evaluation in Section 4 is followed by a conclusion in Section 5.

2 Related Work

One of the first main contributions to LiDAR PS was [5]: the authors present a dataset for PS in addition to two baseline methods. Each of their approaches uses a state-of-the-art semantic segmentation (KPConv [14] and RangeNet++ [15]) in combination with a state-of-the-art object detector (PointPillar [16]). Having one detection and one segmentation part, they fuse both information afterward. All points inside one bounding box of the object detector are seen as one instance. Using oriented bounding boxes minimizes the error of misclassifying nearby objects. For detecting *things*, they need one object-detector per class.

The Dynamic Shifting Network (DS-Net) [8] combines semantic segmentation with a learning-based clustering. Grid-level features are extracted with the help of cylinder convolutions. Based on those features, two branches perform different tasks: the semantic branch connects Multi-Layer Perceptrons to the cylinder convolution to perform semantic segmentation, while the instance branch uses center regressions for preparing the *things*' points for further clustering. A so-called dynamic shifting (DS) shifts the regressed center to the correct cluster center. In the end, a consensus-driven fusion is applied to merge instances with the semantic segmentation. This consensus-driven fusion is a majority voting – for each predicted instance, the most appeared semantic label of its points is used for the whole instance.

In contrast to proposal-based approaches, the Panoptic-PolarNet of [11] is based on a single network. This deep network consists of four main components: a so-called Polar BEV encoder that encodes the raw point cloud into a 2D representation, a shared encoder-decoder network, two heads for semantic and instance segmentation and finally a fusion step.

One of the fastest PS algorithms is Panoster [9]. The authors use an end-to-end network that consists of a shared encoder and two decoupled, symmetric decoders - one for the instance segmentation and one for the semantic segmentation. They fuse the semantic and the instance information using a mask. Their approach is highly efficient and can run with 58 fps.

3 Proposed Approach

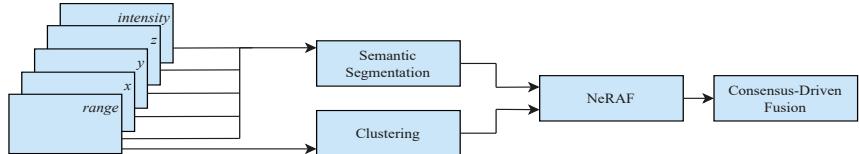


Figure 2: Overview of our approach. With an image-projection as input, a clustering and a semantic segmentation are performed. Afterwards, the activation function generates object instances which are classified by the consensus-driven fusion.

Assuming already preprocessed data, the current approach basically consists of three parts:

1. Semantic Segmentation
2. Clustering
3. Label Fusion

The label fusion itself consists again of two parts: generating instances with the NeRAF and classifying those instances using the consensus-driven fusion. The workflow of our approach can be seen in Fig. 2.

3.1 Preprocessing

Having a 3D point cloud as input, we need to preprocess the incoming data first. Thus, all 3D points (x, y, z) with their euclidean norm $d = \sqrt{x^2 + y^2 + z^2}$ are projected to a set of 2D image coordinates (u, v) :

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} W \left[\frac{1}{2} [1 - \arctan(y, x) \pi^{-1}] \right] \\ H \left[1 - \frac{\left(\arcsin\left(\frac{z}{d}\right) + \text{fov}_{\text{down}} \right)}{\text{fov}_{\text{up}} + \text{fov}_{\text{down}}} \right] \end{pmatrix}. \quad (1)$$

H and W represent the height and the width of the resulting image. fov_{up} and fov_{down} represent the field of view of the sensor. We can see the resulting output as an image with five channels: $\{d, x, y, z, \text{intensity}\}$. If two 3D points lie on the same image coordinate, we choose the one with the smaller euclidean norm (range).

3.2 Semantic Segmentation

Generally, any semantic segmentation could be used for our approach. Nevertheless, we focus on the capability of a high-speed PS. That is why we choose the more efficient variant ("tiny") of the 3D-MiniNet [13] as the segmentation part of our PS approach. We do not perform the K-Nearest-Neighbors post-processing. Further, we keep the proposed settings for training and data augmentation. For a detailed description of the 3D-MiniNet, we refer to [13].

3.3 Clustering

Due to the focus on efficiency, we choose the clustering algorithm from [12]. This approach takes a range image as input and computes instances based on an angle criterion. Therefore, we can use the same range image as in our semantic segmentation part to compute separated instances. We do not need to perform any other preprocessing, due to the shared image projection. Contrary to object detection algorithms which are based on deep learning and run on the GPU, this algorithm runs efficiently on the CPU. While the GPU is busy performing the semantic segmentation, the clusters can be calculated on the CPU. For further information about the clustering, we refer to [12].

3.4 Label Fusion

The core of our approach is the NeRAF. Neither the semantic segmentation nor the clustering result in 3D points yet. So far, the outcome of each part is a matrix which contains class IDs and instance IDs. These two matrices share the same image projection, and thus also the underlying 3D points. We combine these matrices using the NeRAF: In combination with a breadth-first-search (BFS), such as it is used in the labeling part of [12], we iterate through the matrices and compare the range, the class and the instance ID of two pixels. Further, we ignore ground points – we fully trust on the semantic segmentation for classifying ground. We start with a new instance, and NeRAF decides if two neighboring pixels belong to the same instance. Having two neighboring pixels p_x, p_{x+1} with the classes c_x, c_{x+1} , the instance IDs i_x, i_{x+1} and the ranges d_x, d_{x+1} , we define the varying of the class as v_c , the varying of the instance IDs as v_i and the difference between

two depths as δd . For improving the robustness, we do not compare the direct classes c_x and c_{x+1} . Instead we compare the predicted class of the current instance c_i with c_{x+1} . We define NeRAF as

$$\text{NeRAF}(p_x, p_{x+1}) = \frac{w_c v_c + w_i v_i + w_\delta (1 - \frac{\min(|\delta d|, \delta d_{max})}{\delta d_{max}})}{w_c + w_i + w_\delta} \quad (2)$$

with w_c, w_i, w_δ as the three weights for v_c, v_i and δd . v_c and v_i have two possible states: 0 if there is a class / instance ID change and 1 if they are equal.

NeRAF results in a value between 0 and 1. We set the decision boundary to 0.5. If the output is greater than that, we add pixel p_{x+1} to the instance. As it can be seen in Eq. (2), we use weights to control each component of NeRAF. Therefore, we can add common a-priori knowledge about possible neighborhood relations. We define three set of weights (w_c, w_i, w_δ):

Set 1. The main focus lies on v_i and δd .

Set 2. This is the counterpart to set 1: v_c is dominant.

Set 3. Here, we set equal weights. Due to the close relation between δd and v_i , we decrease w_δ .

Now, those three sets can handle several different scenarios, depending on the prior knowledge. The scenarios are:

1. very close, but different classes,
2. same class,
3. normal distance between objects.

As an example: It is common sense, that vegetation and trunks are very close or even overlapping, and thus would fall into scenario 1. Contrary to the pure clustering, the semantic segmentation can handle the separation of those two classes. That would be an ideal use case for set 2. Having two neighboring pixel with the same class, we focus v_i and δd (set 1). Neighborhood relations of objects which should not be too close to each other (e.g. vegetation and cars, buildings and trunks) will be handled in set 3. The scenarios are stored in a look-up table. Each combination of classes receives one integer value, which represents the corresponding scenario. Depending on the scenario, NeRAF can switch between the three different weight sets.

Nevertheless, NeRAF just helps to decide if two neighboring points belong to the same object. The class of those instances is still ambiguous: Each point of each instance can have a different class. As proposed in [8], we use a consensus-driven fusion module. This simple fusion strategy is a majority vote: the whole new instance gets that class ID, which occurs most. As a result, we receive a set of pixels of the projected objects which are either geometrically and / or semantically separated.

3.5 Backprojection

The backprojection from images to point clouds is kept simple. For each point in the 3D point cloud, we get the information on which pixel the point should be projected. Due to the discretization, two or more points might share one pixel. In Section 3.1, we projected only the closest points. Thus, for receiving the class and instance information per point, we add a distance threshold. If the range differs more than 30 cm between the current point and the depth on its position on the range image, the class and the instance ID of this point remain undefined.

4 Experiments

We evaluate our approach using different settings to observe the dependencies between our PS and the semantic input:

- 512*i*: Projection size: 64×512 px.
- 2048: Projection size: 64×2048 px.
- 2048*f*: Projection size: 64×2048 px, "full" 3D-MiniNet.

Note that 2048*f* refers to the non-tiny version of the 3D-MiniNet.

For each setting, we train a network for 500 epochs on the KITTI data for semantic segmentation [17; 18]. As proposed in [17], sequences 00-10 were used for training, except sequence 08 which is used as validation set. For testing and evaluating, a notebook was used with an NVIDIA GTX 1060 as GPU and an i7-8750H CPU with 2.20 GHz. The clustering and the semantic segmentation run in parallel, everything else runs on a single core. We implement our approach in C++ inside the Robot Operating System (ROS) framework [19] and use the TensorRT [20] library to speed up the inference.

4.1 Evaluation Metric

As proposed in [1], we evaluate our approach using the Panoptic Quality (PQ), the Segmentation Qualitiu (SQ) and the Recognition Quality (RQ) which are calculated across all classes. The scores are between 0 and 100%, whereby 100% denotes a perfect PS. We further evaluate those three metrics separately on *things* PQth, SQth, RQth and *stuff* PQst, SQst, RQst. Moreover, PQ[†] defines the PQ metric, while swapping the PQ of *stuff* classes with its IoU. Even though our approach itself does not distinguish between *stuff* and *things* by its class, we keep the *stuff/things* setting from [5] due to the lack of static or natural *things* in the evaluation data. Setting the estimated instance ID of those classes would cause a misleading result. Therefore, we follow [5] and set the instance ID of the *stuff* classes (for the evaluation) to zero.

4.2 Quantitative Results

4.2.1 Panoptic Quality

We evaluate our approach using the SemanticKITTI Benchmark for PS. We compare our results to the results of the two baseline methods in [17], LPSAD [10], the Multi-Object Panoptic Tracking of [21], Panoster [9], the Panoptic-PolarNet [11] and the DS-Net [8]. The results can be found in Table 1.

Generally, our approach receives a PQ score between 24.5 and 39.5%, depending on the network. Therefore, our method reaches similar results to [10] and [15] in combination with [16]. Nevertheless, the accuracy of our approach lies under most of the state-of-the-art methods. One main reason is the trade-off between performance and quality: The better the overall performance of the semantic segmentation (e.g. 2048*f*), the better the result of the PS. Nevertheless, this also leads to an extended computation time. Moreover, our approach is the only general approach: while others do need the strict definition of *stuff* and *things* and dedicated training data for the instance segmentation, our algorithm results in PS of each object – no matter of its class.

Table 1: Comparison of LiDAR panoptic segmentation performance on the SemanticKITTI test set. All scores are in [%]. Note different GPUs:

[†]NVIDIA RTX 2080ti, ^{*}NVIDIA Quadro P6000, ⁺NVIDIA GTX 1060, [#]NVIDIA GTX 1080, ^ounknown

Method	PQ	PQ [†]	SQ	RQ	PQ Th	SQ Th	RQ Th	PQ St	SQ St	RQ St	mIoU	t [ms]
R.N. [15] + P.P. [16] [†]	37.1	45.9	75.9	47.0	20.2	75.2	25.2	49.3	76.5	62.8	52.4	40
KPC [14] + P.P. [16] [†]	44.5	52.5	80.0	54.4	32.7	81.5	38.7	53.1	79.0	65.9	58.8	51
LPSAD [10] [*]	38.0	47.0	76.5	48.2	25.6	76.8	31.8	47.1	76.2	60.1	50.9	8
Panoster [9] [#]	52.7	59.9	80.7	64.1	49.4	83.3	58.5	55.1	78.8	68.2	59.9	1
Panoptic-PolarNet [11] ⁺	54.1	60.7	65.0	81.4	53.3	60.6	87.2	54.8	68.1	77.2	59.5	23
DS-Net [8] ^o	57.7	63.4	68.0	77.6	61.8	68.8	78.2	54.8	67.3	77.1	63.5	53
ours (512) ⁺	27.0	36.9	72.0	36.5	15.9	74.3	20.9	35.0	70.3	47.8	40.8	2
ours (2048) ⁺	34.3	42.8	74.1	45.4	23.4	76.7	29.8	42.2	75.1	56.7	45.1	10
ours (2048f) ⁺	39.6	47.6	74.8	52.0	30.9	78.4	38.5	45.9	72.1	61.8	51.1	1

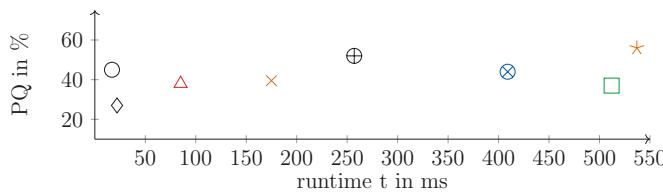
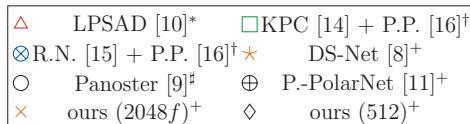


Figure 3: Runtime of several approaches. As already in Table 1, note the different GPUs.

4.2.2 Runtime Analysis

The runtime of the different approaches can be seen in Table 1. In our calculation, we exclude the spherical projection and the backprojection: Everything else, including further preprocessing, copying, and moving data, is included.

Generally, our approach reaches a mean runtime of 22 to 175 ms, depending on the setting. As one would expect, a four times smaller input (64×512 vs. 64×2048) results in a roughly four times lower computation time. Comparing our approach with others is rather difficult due to different hardware or non-public code. Nevertheless, in Fig. 3 we can see the runtime and PQ of several methods and their hardware used. Even though a low-level consumer graphics card was used, we can clearly see that our approach is up to 26 times faster than other approaches. Nevertheless, we can not reach the 17 ms from [9], conducted on an NVIDIA GTX 1080. Comparing the full point cloud (64×2048), the approach from [10] beats our runtime, too. According to the authors, they used an NVIDIA Quadro P6000.

Even though we use a highly efficient network, the segmentation still has the highest impact on the computation time: With a full range-image (64×2048), the fusion step takes 15 ms. Copying the data in combination with further preprocessing takes 40 ms. The clustering does not impact the computation time because it is always faster than the semantic segmentation. 42 to 69 % of the computation time is used for the inference.

In summary, even with a low-level graphics card, our approach runs faster than most of the other approaches, despite their superior hardware.

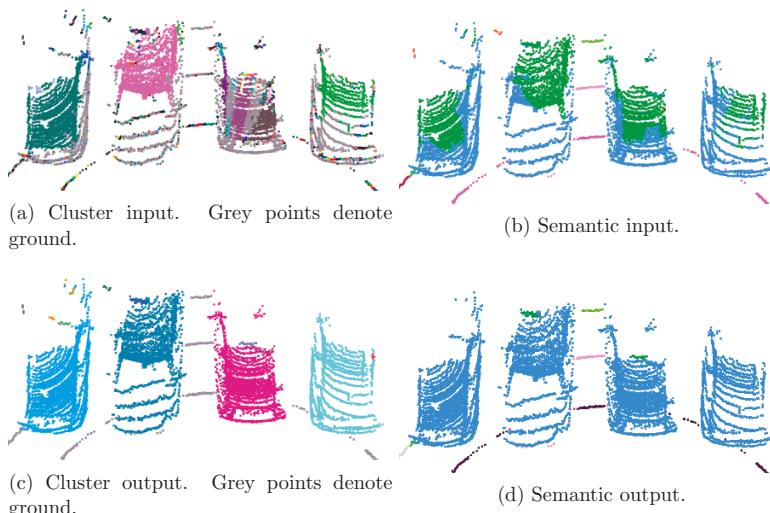


Figure 4: In- and outputs of our method. While Fig. 4a shows the cluster input and Fig. 4b shows the semantic input, Fig. 4c shows the resulting cluster and Fig. 4d shows the resulting class.

4.3 Qualitative Results

Even though other purely deep learning based methods perform well in their evaluation sequences, our approach has one major advantage: through the combination of learning and non-learning based methods, it takes the benefits of both worlds. Hence, the output of our PS is generally stable in unknown areas and different sensors. Fig. 4 shows an example using a Velodyne VLS-128 LiDAR. Further, we added extra noise to the point cloud in order to enlarge the error. Even though neither the semantic segmentation nor the clustering result in a good solution, the outcome of the combination becomes more coherent.

5 Conclusion

In this work, we show a general approach for PS. Contrary to other methods, it is not fully based on deep learning and combines semantic segmentation with clustering using a neighborhood-related activation function (NeRAF). Hence, it generates classified instances, no matter of its class. Depending on the incoming semantic segmentation, our approach can reach the accuracy of current state-of-the-art. Moreover, every single part of our approach is highly efficient. This makes the PS run with up to 45 fps on a consumer graphics card. Future work should keep a focus on the combination of non-learning based clustering and semantic segmentation for PS and evaluate different methods for fusing both parts. Further, NeRAF should be evaluated further, using advanced networks and different clustering methods.

References

- [1] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, “Panoptic Segmentation,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, Jan. 2019.
- [2] B. Cao, C.-N. Ritter, D. Göhring, and R. Rojas, “Accurate Localization of Autonomous Vehicles Based on Pattern Matching and Graph-Based Optimization in Urban Environments,” in *Proc. IEEE Intelligent Transportation Syst. Conf. (ITSC)*, 2020.
- [3] A. Schaefer, D. Buescher, J. Vertens, L. Luft, and W. Burgard, “Long-Term Urban Vehicle Localization Using Pole Landmarks Extracted from 3-D Lidar Scans,” in *European Conference on Mobile Robots (ECMR)*, Prague, Czech Republic, Sep. 2019.
- [4] P. Burger, B. Naujoks, and H.-J. Wuensche, “Map-Aware SLAM with Sparse Map Features,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Syst. (IROS)*, Macau, China, 2019.
- [5] J. Behley, A. Milioto, and C. Stachniss, “A Benchmark for LiDAR-based Panoptic Segmentation based on KITTI,” *arXiv preprint arXiv:2003.02371*, 2020.
- [6] K. Sirohi, R. Mohan, D. Büscher, W. Burgard, and A. Valada, “EfficientLPS: Efficient LiDAR Panoptic Segmentation,” *arXiv preprint arXiv:2102.08009*, 2021.

- [7] J. Xie, M. Kiefel, M.-T. Sun, and A. Geiger, “Semantic Instance Annotation of Street Scenes by 3D to 2D Label Transfer,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, 2016.
- [8] F. Hong, H. Zhou, X. Zhu, H. Li, and Z. Liu, “LiDAR-Based Panoptic Segmentation via Dynamic Shifting Network,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, Jun. 2021.
- [9] S. Gasperini, M.-A. N. Mahani, A. Marcos-Ramiro, N. Navab, and F. Tombari, “Panoster: End-to-end Panoptic Segmentation of LiDAR Point Clouds,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, 2021.
- [10] A. Milioto, J. Behley, C. McCool, and C. Stachniss, “LiDAR Panoptic Segmentation for Autonomous Driving,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Syst. (IROS)*, 2020.
- [11] Z. Zhou, Y. Zhang, and H. Foroosh, “Panoptic-PolarNet: Proposal-Free LiDAR Point Cloud Panoptic Segmentation,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, Jun. 2021.
- [12] I. Bogoslavskyi and C. Stachniss, “Fast Range Image-Based Segmentation of Sparse 3D Laser Scans for Online Operation,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Syst. (IROS)*, 2016.
- [13] I. Alonso, L. Riazuelo, L. Montesano, and A. C. Murillo, “3D-MiniNet: Learning a 2D Representation from Point Clouds for Fast and Efficient 3D LIDAR Semantic Segmentation,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, 2020.
- [14] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, “KPConv: Flexible and Deformable Convolution for Point Clouds,” in *Proc. IEEE Int. Conf. Comput. Vision (ICCV)*, 2019.
- [15] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, “RangeNet++: Fast and Accurate LiDAR Semantic Segmentation ,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Syst. (IROS)*, 2019.
- [16] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, “PointPillars: Fast Encoders for Object Detection from Point Clouds,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, 2019.
- [17] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, “SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences,” in *Proc. IEEE Int. Conf. Comput. Vision (ICCV)*, 2019.
- [18] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, 2012.
- [19] Stanford Artificial Intelligence Laboratory et al., “Robotic Operating System.” [Online]. Available: <https://www.ros.org>
- [20] NVIDIA, “TensorRT.” [Online]. Available: <https://developer.nvidia.com/tensorrt>
- [21] J. V. Hurtado, R. Mohan, W. Burgard, and A. Valada, “MOPT: Multi-Object Panoptic Tracking,” *Proc. IEEE Conf. Comput. Vision and Pattern Recognition Workshops (CVPRW)*, 2020.

A Safety-Certified Automotive SDK to Enable Software-Defined Vehicles

Jan Becker*

Zusammenfassung: Automated driving, connected vehicles, e-mobility, shared mobility—all mobility disruptors rely on software but lack a unified software platform is preventing cross-domain software development. In the meantime, the vehicle compute and network architectures are moving to centralized high-performance computers, but the software implementation lags behind the hardware architecture. We have introduced Apex.OS, the first mobility software platform that can truly integrate across all in-vehicle domains. A primary vehicle operating system, robust and flexible enough to cover major systems throughout the vehicle and the cloud, enables user-focused development, just like iOS and Android SDK (software development kit) software development kit do so for embedded devices. This paper describes our development of an automotive SDK capable of covering all automotive software domains and already certified to ISO 26262 ASIL D. It addresses software development of driver assistance systems, automated driving, other safety-critical automotive systems, as well as applications for smart machines and IoT. **Schlüsselwörter:** vehicle OS, SDK, Apex.OS, vehicle software architecture.

1 Introduction

The automotive industry is moving toward new generations of the E/E-architecture based on centralized high-performance computers to support the megatrends towards electric, self-driving, shared, and connected vehicles. But the vehicle software architecture is lagging behind the hardware architecture. Today, automotive OEMs are stitching together disconnected software components to build proprietary platforms [8]. In the past, in a distributed E/E-Architecture, distributed and independently functioning electronic control units (ECUs) communicate with each other over CAN bus-based communication. For today's architecture, a central gateway was introduced to enable some cross-domain communication. Currently under development is the fourth generation of the E/E-architecture, which will see the introduction of domain controllers. These are computing platforms centralizing computing resources within one domain, such as the powertrain, chassis control, in-vehicle-infotainment, or ADAS and vehicle automation. This enables cost optimization, e.g., through consolidation of software and reduction of wiring harness cost but requires much higher bandwidth network communication and increases the requirements for safety and security. The fifth generation, currently being developed, will then see the introduction of centralized computing infrastructure, in which all major functionality is implemented on one high-performance computing platform, satellite sensors

*Jan Becker is CEO and Co-Founder of Apex.AI, Inc., 844 E Charleston Rd, Palo Alto, CA 94303, USA (e-mail: jan@apex.ai) as well as Lecturer at Stanford University.

are connected via a high-speed ethernet network and separation of domains is implemented in software. Overall, the architecture is rapidly moving from a complex system architecture with simple software to a simple system architecture with complex software. If done incorrectly, this will pose incredible challenges and risks, such as exploding software complexity, increased cybersecurity risks, incoherent cross-domain implementation, diverging non-consistent tooling, and so on. If done properly, this opens the opportunity to introduce modern software development processes and paradigms, a clean and scalable software architecture, cross-domain consistency and tooling, limited controllable cybersecurity risks, over-the-air update-ability, just to name a few.

2 How to get to software-defined vehicles?

The challenges of bringing about the Software-Defined Vehicle (SDV) are manifold and are resulting in widespread changes rippling throughout the automotive industry. The Software-Defined Vehicle is changing technology and toolchains. It is causing changes to the V-cycle development model's execution, openness, and iteration speed. It is changing organizational structures within companies. And it is changing the nature of OEM - supplier relationships.

Let's take a look at commercialization first. Automakers started by selling vehicles, then parts and accessories became an additional source of income after the initial vehicle purchase. Then came financing of sales and leases. Building Internet connectivity into vehicles opened additional potential revenue streams and kicked off the automakers' efforts to build their digital own ecosystems and services with the vehicle in the center of these ecosystems—but the world evolved differently. People still order their coffee through an app on their phone and not through the onboard UI and manufacturer ecosystem. The digital ecosystem and services are in the center and extend through apps to various devices at the edge. The vehicle has become one of many devices in these ecosystems, such as the phone, the pad, the home, health devices, etc, and services are consistently offered on just the few most successful digital ecosystems such as Apple or Android.

Automakers are determined to get there with large internal reorganizations and tremendous investments in their software abilities. *Software as product differentiator* is one of the factors driving OEMs to Software-Defined Vehicle and bringing more of their software development in-house.

3 Software-defined phones—what did we learn?

The market for operating systems is typically a winner-takes-all market. The industry segment does not matter, and that is why the automotive industry can learn a lot from what the mobile phone industry went through in the mid-2000s. Until then, a mobile phone's software was loaded onto the device at the end of the production line. The software was only updated if a significant bug was identified.

The introduction of the iPhone in 2007 popularized the separation of applications from the base software and hardware, turning the mobile phone into a software-defined device. It became a general-purpose platform upon which manufacturers could create an operating system with developer SDK and deliver updates over-the-air (OTA), enabling

developers to develop and deliver new applications. This drove the exponential growth of the smartphone market and compute performance of these devices. The two winners were Apple's iOS, which was first-to-market with superior user experience, and Android, which has grown to a 70% market share.

Previous platforms were swept aside. What led to the dominance of Android? Android is an open architecture with an SDK, is customizable and brandable, which saves the OEMs development cost and provides a rich application ecosystem that no single manufacturer (besides Apple) can achieve on their own.

4 Why a common OS and developer SDK?

Why is a common operating system (OS) with a software development kit (SDK) for mobility platforms relevant? Why shouldn't every manufacturer develop their own? Vehicles are becoming devices in a larger mobility ecosystem and are thereby competing with the larger mobile device market. Ecosystems derive value primarily through their size, so the larger an ecosystem, the more value it contains. Software developers and suppliers need to focus, so they are drawn to the most significant markets.

need graphic

Nevertheless, many automotive manufacturers started in-house initiatives to develop their own operating systems. Their goals are ambitious; various companies want to integrate their OS systems into their products as early as 2025. However, the time frame seems much too tight. Going into production in just three years with functions based on the operating system developed in-house is overambitious, especially for companies that are not software companies. Last but not least, the safety and security certification of the new software already takes up to three years. But what if the basis for such an operating system with developer SDK already existed? Could that speed development?

5 An approach to a common OS and developer SDK

In this paper, we aim to describe a world as well as a path to this world in which software throughout the whole vehicle is truly integrated end-to-end. A primary vehicle operating system on the outside—but really an SDK (software development kit) in the sense of Android SDK or iOS SDK on the inside—to abstract complexity away from the developers, provide common and open APIs as well as robust and flexible enough to cover major systems throughout the vehicle. What is needed to get to software-defined vehicles? We argue that the following requirements need to be fulfilled:

1. A standardized software architecture with open APIs to enable mutually compatible solutions ideally across many manufacturers, suppliers, and academia.
2. An awesome developer experience to enable developer productivity—based on the understanding that the quality of the developer experience is directly related to their productivity.
3. A software architecture that scales to massive software systems.
4. A software implementation based on modern software engineering practices.

5. Abstraction of the complexity of all underlying hardware and software.
6. All of the above with deterministic, real-time execution, and with automotive functional safety certification.

We draw an analogy to the field of robotics, which encountered similar challenges (with some exceptions w.r.t determinism and real-time) ten years ago and addressed the first five challenges by developing ROS (the Robot Operating System) as an open-source software framework. The authors are not aware of a single automotive company that is not or has not been using components from ROS or its rich software ecosystem in their development. But the first generation of ROS had its significant deficiencies. This has now changed: In 2018, the 2nd generation ROS 2 was released with improvements based on the lessons learned of using ROS 1 in hundreds of robotic and automotive prototypes over ten years. ROS 2 is based on a proven architecture that enables scaling to complex software systems. It is also based on a proven and standardized middleware instead of a homegrown data communication approach. In addition, with the release of Apex.OS (Apex.AI's proprietary fork of ROS 2), we addressed real-time and deterministic execution by rewriting the underlying implementation based on the same open APIs. And we developed a process to take this implementation through functional safety certification in record time. Here, we will describe the advantages of working with open APIs and an architecture that has been proven in use by tens of thousands of developers and in thousands of robots and automated vehicles. We will close by outlining a software development process that allows taking open-source software through functional safety certification into production vehicles.

6 Lessons Learned from Robotics

A robotics framework is a collection of software tools, libraries, conventions, and APIs made to simplify the task of developing software for a complex multi-sensor, multi-actuator system, like a vehicle. A well-implemented robotics framework is implemented with the separation of concerns in mind. These concerns are introduced in the paper by Vanthienen as 5Cs [14]: configuration, coordination, composition, communication, and computation. Writing the software this way allows for maximum reusability, testability, reliability, portability, maintainability, extensibility of components. In most cases, the use of a robotics framework dictates the general architectural principles of the developed software. For example, if the software is centralized or decentralized, real-time or non-real-time, etc. A key component is the middleware, which is the glue that holds together the numerous components of a robotics framework. The most basic task of the middleware is to provide the communications infrastructure between software nodes in an autonomous vehicle. The typical use case for a robotics framework is to provide the essential interfaces between high-level (software) and low-level (hardware) components of the system. These interfaces and components consist of various operating system (OS) specific drivers that would take a single developer a significant amount of time to develop. The software framework should abstract away ECUs, OSes, physical transport layers, and interface to the components needed for offline (e.g., mapping, simulation, testing) as well as online operations (user applications).

6.1 ROS 1

Various efforts at Stanford University in the mid-2000s involving integrative, embodied AI, such as the STanford AI Robot (STAIR) and the Personal Robots (PR) program, created in-house prototypes of flexible, dynamic software systems intended for robotics use—which they named ROS[10]. In 2007, Willow Garage, a nearby visionary robotics incubator, provided significant resources to extend these concepts much further and create well-tested implementations. The effort was boosted by countless researchers who contributed their time and expertise to both the core ROS ideas and to its fundamental software packages. Throughout, the software was developed in the open using the permissive BSD open-source license and gradually has become a widely used platform in the robotics research community.

From the start, ROS was developed at multiple institutions and for multiple robots, including many institutions that received PR2 robots from Willow Garage. Although it would have been far simpler for all contributors to place their code on the same servers, over the years, the "federated" model has emerged as one of the great strengths of the ROS ecosystem. Any group can start their own ROS code repository on their own servers, and they maintain full ownership and control of it. They don't need anyone's permission. If they choose to make their repository publicly available, they can receive the recognition and credit they deserve for their achievements and benefit from specific technical feedback and improvements like all open-source software projects.

The ROS ecosystem now consists of tens of thousands of users worldwide, working in domains ranging from tabletop hobby projects to vehicles to large industrial automation systems. Despite its large success, ROS 1 did not find its path to automotive production programs because of the following reasons: low code quality, non-standard, non-automotive middleware, lack of real-time capabilities, no nodes with managed lifecycle, no security, lack of testing and documentation, no support for automotive ECUs.

6.2 ROS 2

To address the aforementioned shortcomings, the Open Source Robotics Foundation (OSRF) started a new project in 2013, ROS 2. Open Robotics considered autonomous vehicles as well as the following use cases before starting development for ROS 2:

- Small, embedded platforms: ROS 2 has a smaller and more optimized codebase compared to ROS 1, which can run on aarch64 computer architectures.
- Real-time systems: ROS 2 communication mechanism, memory management, and threading model allow for it to be real-time upon hardening.
- Production environments: Coupled with real-time constructs, ROS 2 can run on an RTOS, such as QNX.
- Prescribed patterns for building and structuring systems: While ROS 2 has the underlying flexibility that is the hallmark of ROS 1, ROS 2 provides patterns and supporting tools for features such as life cycle management and static configurations for deployment.

Now let's have a look at how ROS 2 fulfills the promises given in the introduction of this article.

7 A standardized software architecture with open APIs

Developing a standardized software architecture that can satisfy applications ranging from an embedded microcontroller running on a drone or radar sensor to the centralized high-performance ECU for the autonomous car is no easy task. Many different aspects such as a hardware abstraction layer, an OS abstraction layer, a runtime layer, support for various programming languages, non-functional performance, security, safety, software updates, tools for the development, debugging, recording and replay, visualization, simulation, tools for the continuous integration and continuous deployment, interfaces to the legacy systems (such as AUTOSAR Classic), execution management in the user applications, time synchronization, support for hardware acceleration, model-based development, and more have to be considered. To integrate these aspects into one coherent framework, the following design decisions were followed in the design of ROS 2:

- Adopt positive lessons learned from the extensive deployment of ROS 1
- Apply negative lessons learned from the extensive deployment of ROS 1
- Use hourglass design to keep the core of ROS 2 as a single code base
- Use industry-proven middleware and focus on the layers above the middleware
- Newly developed mechanisms for security, safety, software updates, interfaces to the legacy systems (such as AUTOSAR Classic), execution management in the user applications, time synchronization, support for hardware acceleration, model-based development

With these aspects in mind, a new architecture shown was created for ROS 2. The largest change from ROS 1 to ROS 2 is that the underlying communication layer was implemented in a plug-in-like interface for many types of middleware solutions and also provides tier 1 support for DDS implementations. On top of the middleware layer is a middleware abstraction layer called the ROS middleware API, which is used to prevent a DDS vendor lock-in. Implementation of the middleware layer for example, SOME/IP is with this design rather trivial. On top of this layer are the ROS 2 client libraries written in C++ and Python. Using these client libraries, developers can write application code. With this so-called hourglass design, the thin client libraries always wrap around the same code base for operating system and communication.

One of the concepts that proved itself the most in ROS 1 was the concept of a node as a central logical unit. The node component brings together different publisher and subscriber patterns, threading, queues, and the execution model—all abstracted behind the intuitive set of APIs. A node can subscribe to one or more data streams (called topics), the data is then aggregated and processed in a single-threaded or parallel loop, and the result is published on another topic. The node can also be parametrized, composed into one or more processes, and it also comes with its own lifecycle for starting and stopping. Nodes also emit diagnostics information such as logging, heartbeat, and node state information.

One of the missing concepts in ROS 1 was the concept of executor, which was added in ROS 2. The executor provides spin functions and then coordinates the nodes and

callback groups by looking for available work and completing it, based on the threading or concurrency scheme provided by the subclass implementation. An example of available work is executing a subscription callback or a timer callback. The executor structure allows for a decoupling of the communication graph and the execution model—a feature much needed when building real-time and deterministic systems.

ROS 1 provided standardized messages [1] for diagnostics, geometric primitives, robot navigation, and common sensors such as laser range finders, cameras, and point clouds. This allows for having the agreed-upon contract between different software and hardware modules and in turn allows for great flexibility and reusability of developed software components. These messages were also adopted in ROS 2 but implemented with the OMG IDL language, which enables e.g., message versioning and message forward and backward compatibility.

ROS 1 had very high non-functional performance characteristics, especially in terms of CPU usage and high bandwidth which allowed the developers to cope with the throughputs of up to 6 Gb/s of data as produced by for example an autonomous driving car. In ROS 2 this had to be preserved and even further enhanced by the introduction of e.g., zero-copy transport mechanisms such as iceoryx [2]. These are just a few examples of proven in the field which could be improved because ROS 1 is an open-source framework [11] and because there is a large community of users constantly running into the edge cases and the best software is developed through the coding iterations and rather than through implementation solely based on specifications.

7.1 An awesome developer experience

ROS community has developed a various set of desktop and cloud tools that allow the users to introspect and debug their systems, record, replay, and visualize the data, and emulate and simulate their systems. For the full set of tools, we would like to direct the user to the dedicated webpage [13]. For ROS 2, we would like to single out two particular tools that contributed to the improved developer experience and productivity.

The ADE Development Environment [1] is a modular Docker-based tool to ensure that all developers in a project have a common, consistent development environment. ADE creates a Docker container from a base image and mounts additional read-only volumes in /opt. The base image provides development tools (e.g., vim, udpreplay, etc.), and the volumes provide additional development tools (e.g., IDEs, large third-party libraries) or released software versions. Furthermore, ADE enables easy switching between versions of the images and volumes.

Arguably the most significant effort required for the production deployment of autonomous vehicles (AVs) is the proof that the system is safe for homologation. To provide such proof, a holistic, software-first, approach to testing for autonomous vehicles must be available. In other words, we need a framework that enables writing of the relevant tests and supports extracting the proofs. To develop such a framework, it is important to look at the automotive V-model and to understand what kind of tests one is expected to write: Unit tests, component tests, integration tests, sub-system tests, system tests, non-functional tests, acceptance tests. Furthermore, it is important to understand how input data is generated for these tests: Manually created, manually generated, simulation generated, vehicle recordings, and in what kind of environments the tests are executed: Software-in-the-loop (SIL), Hardware-in-the-loop (HIL), Vehicle-in-the-loop (VIL). Lastly,

it is important to understand what kind of key performance indicators are to be extracted from the tests.

There are plenty of tools available for unit and component testing (e.g., [4]) but upon extensive research, we were unable to find a framework that would satisfy other types of tests, cover all other input data and environments and that would clearly distinguish between the following phases: test configuration, test setup, test running and results logging and extraction. We thus wrote a tool in the open source called launch testing [5], which is a framework for the integration, system and sub-system, non-functional as well as acceptance testing of applications written with ROS 2 and Apex.OS.

7.2 A software architecture that scales to massive systems

By "Software architecture that scales" we refer to scaling with the number of developers, scaling in terms of the non-functional performance, scaling in terms of the integration of third-party automotive components, and scaling in terms of centralized and decentralized applications. ROS (and thereby Apex.OS) is the most field-proven framework worldwide in terms of user adoption. Per the last ROS community metric report citeROS-metrics, there are more than 200.000 known users of ROS worldwide, and there are presumably significantly more unknown developers in corporate development. Key enablers for this are open APIs, an ecosystem of tools, best development practices, and tight integration with various CI/CD systems.

The ROS and Apex.OS non-functional performance has been described in detail [9] with respect to the efficiency of abstraction layers and the efficiency of the transport protocols resulting in the acceptable CPU utilization, memory consumption, latency, page faults, context switches, and other such system metrics. As explained in [10], it is also important to have a properly designed tool for unbiased measurements and a methodology that mimics that of the actual application.

7.3 A software implementation based on modern software engineering practices

To have a robust and maintainable codebase one needs to use a) well-defined but friendly software development process and b) state-of-the-art software development tools. Such SW development process encompasses all the steps from the input requirements all the way to the CI/CD. The keys to the success of such a development process are:

- An integrated development environment, which at Apex.AI is centered around Gitlab, Gitlab CI and docker.
- An integrated IDE, which at Apex.AI is Clion. Clion provides all of the state-of-the-art features such as code completion, debugging but also integration of external tools such as e.g., gtest, valgrind, different build tools, doxygen, tool for code test coverage.
- The steps implementing the development process must be fast to allow quick coding iterations.

- The local development environment and the CI/CD must be equivalent to be able to reproduce CI failures.
- Integrations with the 3rd party tools, such as the JAMA requirements management tool, must be custom tailored to the particular team to allow for the quick fixes and extensions.
- The main code repository should be as monolithic as possible and all of the development artifacts (design documents, code, tests, documentation, etc) should be as co-located as possible.

8 From ROS 2 to Apex.OS Cert–ISO 26262 [5] ASIL D certification of an open-source codebase

Apex.OS Cert is based on our fork of ROS 2 Dashing. Starting with a technical safety concept (TSC) for Apex.OS, which defined an assumed use case and ODD (operation design domain). For the first iteration of Apex.OS Cert assumed use case consisted of a rather simple scenario where one or more publishers communicate data to one or more subscribers. Of the 5Cs [14] of a typical robotics framework (communication, coordination, computation, composition, and configuration), hazards derived were restricted to data communication. This reduced the scope of Apex.OS Cert safety concept but at the same time was sufficient to meet customer requirements at the time. The next step was to determine the boundary of Apex.OS Cert. We decided to restrict the boundary to the source code and to add libraries to the scope in future releases of Apex.OS Cert. At this point, we wrote an initial safety case of Apex.OS Cert. The Apex.OS Cert safety case is an internal document that lays out what are our safety objectives and what artifacts are generated. An safety audit resulted in tacit approval for both the aforementioned TSC and safety case. While defining the safety case, we also inventoried all the development tools that were required to run TCL (Tool Confidence Level) reports (ISO 26262-8:2018 [6]) and planned our processes in FSLC (functional safety lifecycle) such as our change management, safety culture, code review enforcements and so on. These plans were also captured in functional safety management (FSM) artifacts as dictated in ISO 26262-2:2018.

9 Challenges

Several technical challenges were encountered during this process, including the following. Achieving 100% structural coverage: ROS 2 packages came with low MC/DC coverage, and we selected a modern and certified coverage tool to help us achieve complete coverage. ROS 2 is written in modern C++ with heavily templated constructs. Available coverage tools failed to process complex C++ constructs such as back-to-back lambda functions. We worked with our tool vendors to identify solutions. Furthermore, defensive coding methodologies can make it difficult to reach conditional statements that are gated by external libraries over which we have little control. We reverted to GoogleMock [3] techniques in such cases. Checking for runtime memory allocations: Applicable tools for checking runtime memory allocations and blocking calls are not available to our

knowledge. We repurposed LTTng [7], the Linux tracing tool next gen, to accomplish this task. Automation was difficult, and issues such as instrumenting code had to be done manually, which is undesirable for obvious reasons. Making Apex.OS runtime execution deterministic: ROS 2 uses standard containers which allocate memory explicitly. We reverted to our own containers for string, vector, and map/set to not only make Apex.OS Cert runtime memory static but also make the execution as deterministic as possible. Development tool classification and qualifications: ROS 2 uses external development tools for activities such as parsing/configuring (e.g., YAML) and code generation (e.g., rosidl_generator_cpp). TCL reporting led to TCL2 (medium confidence) and consequently the tools had to be qualified. Qualification of external tools is challenging since access to source code and test base is often not available. Qualifying a large open-source tool such as YAML parser (that also includes a library) is prohibitively resource-intensive and time-consuming. We reverted to proprietary solutions for each purpose.

10 Summary and Outlook

We have been able to show that certifying an open-source project to the highest level of integrity is possible with the right expertise and strategic choices of scope and tooling. Apex.OS Cert provides manufacturers and suppliers a fast track and scalable process to build a safe automotive software stack. Our customers have shown applicability to a wide range of automotive applications not limited to autonomous driving. We have been able to accomplish development and certification in a very short amount of time based on our prior expertise in ROS and through reuse of a proven architecture and tooling. For future work, Apex.AI will not only expand the technical safety concept of Apex.OS but also certify libraries including the important transport layer (middleware) libraries. Apex.OS is also in the process of being integrated with AUTOSAR Adaptive and SOME/IP.

References

- [1] *ADE environment*, <https://ade-cli.readthedocs.io/en/latest/>, last accessed 2021/05/08.
- [2] *Eclipse iceoryx*, <https://github.com/eclipse-iceoryx/iceoryx>, last accessed 2021/05/08.
- [3] *GoogleTest Mocking (gMock) Framework*, <https://github.com/google/googletest/tree/master/gmock>, last accessed 2021/05/08.
- [4] *GoogleTest Framework*, <https://github.com/google/googletest>, last accessed 2021/05/08.
- [5] *ISO 2626 Road vehicles–Functional safety*, <https://www.iso.org/obp/ui/#iso:std:iso:26262:-9:ed-2:v1:en>, last accessed 2021/05/08.
- [6] *ROS launch-testing*, https://index.ros.org/p/launch_testing/, last accessed 2021/05/08.

- [7] *LTTng tracing framework for Linux*, <https://lttng.org/>, last accessed 2021/05/08.
- [8] McKinsey & Company, *Mapping the automotive software and electronics landscape through 2030*, <https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/mapping-the-automotive-software-and-electronics-landscape-through-2030>, last accessed 2021/05/08.
- [9] Pemmaiah, A., Pangercic, D., Aggarwal, D., Neumann, K., Marcey, K., *Performance Testing in ROS 2*, <https://www.apex.ai/post/performance-testing-in-ros-2>, last accessed 2021/05/08.
- [10] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A., *ROS an open source operating system*, In: ICRA workshop on open source software 2009, <http://robotics.stanford.edu/~ang/papers/icraoss09-ROS.pdf>.
- [11] *ROS Community Metrics Report*, <http://download.ros.org/downloads/metrics/metrics-report-2020-07.pdf>, last accessed 2021/05/08.
- [12] *ROS core stacks*, <https://github.com/ros>, last accessed 2021/05/08.
- [13] *ROS tools*, <http://wiki.ros.org/Tools>, last accessed 2021/05/08.
- [14] Vanthienen, D., Klotzbücher, M., Bruyninckx H. *The 5C-based architectural Composition Pattern: lessons learned from redeveloping the iTaSC framework for constraint-based robot programming*. In: 9th Journal of Software Engineering for Robotics, pp. 17-35. <https://lirias.kuleuven.be/1748457>.

Towards «type approval» of automated vehicles: Means of safety validation and simulation-based methods, in particular

Wolfgang Kröger* and Ali Ayoub†

Summary: Within the “type approval” system”, mandatory in Europe, a progressive approach is evolving allowing for virtual testing as part of a generic assessment method including the demonstration of the automated vehicles’ capability to cope with the most critical scenarios. Besides questions on completeness, parameter space explosion emerges as a main problem. Advanced methods such as data-driven probabilistic frameworks based on PCE and statistical learning techniques are proposed. Finally, issues of complexity of fully automated and connected vehicles are addressed, calling for a “systems approach” and methods which capture the entire system and go beyond traditional methods of reliability theory.

Keywords: automated vehicles, complex system, safety validation, simulation-based methods

1 Introduction

Automated vehicles (AVs) and automated driving systems, respectively, rely on machine learning (ML) during different driving phases and for different functions. Therefore, while recognizing patterns and interpreting sensor and image data, the ML driving algorithms learn and get better. Hence, reliability and safety, which are driven by software, are not static but change, hopefully increasing continuously with modifications and updates during operation. However, the question remains when these systems are sufficiently safe and mature to be released to public use and how to guide and organize the safety validation and certification process.

Already existing methods of safety validation of automated vehicles can be systematically categorized into physical track and real-world testing, simulations, extreme value theory, formal verification, and scenario-based testing (Junietz et al., 2018). Each of these methods has its pros, cons, basic assumptions, and limitations.

* Swiss Federal Institute of Technology Zurich (ETH) & Swiss Academy of Engineering Sciences (SATW), email: kroeger@ethz.ch.

† Nuclear Science and Engineering Department, Massachusetts Institute of Technology (MIT), email: aliayoub@mit.edu.

2 Safety requirements and validation

The question “How safe is safe (or just good) enough?” is crucial and well known from other domains. Commonly agreed, even mandatory reliability and safety targets do not exist yet for (cooperative and) automated vehicles¹.

We distinguish three levels of abstraction and related targets. First, at the highest level, self-driving vehicles should better perform than vehicles driven by “attentive” humans with state-of-the-art assistant systems, which realistically appears hard to achieve. Targets or thresholds may remain in qualitative forms or exist in quantitative forms for which accident/collision-free driving per distance (km), time between crashes, or compliance with risk curves and associated tolerability lines are proposed as metrics/substitutes. The draft of the EC Implementing Regulation suggests as indicative target, *hazardous errors from the vehicle equipped with Automated Driving Systems (ADS) should be at most of the rate of 10^{-9} per hour, derived from the minimum endogenous mortality risk*². The UN-ECE working party on functional requirements FRAV proposed 10^{-8} per hour for accidents with fatalities and 10^{-7} per hour for accidents with light or severe injuries, *aiming to achieve a neutral or positive risk balance compared to human driving* (FRAV, 2020).

Second, at the system level, the manufacturer has to demonstrate by a robust design and validation process that the system (here specified for Automated Lane Keeping System (ALKS)) is free of “unreasonable” risks for the driver, passengers and other road users and compliance with road traffic rules is ensured (UNECE, 2021).

Third, at components and subsystems level, safety must be ensured by compliance with standards, such as ISO 262 62: 2018 for development of safety-critical functions and devices and ISO PAS 21448:2019 for demonstrating safety of the intended functionality (SOTIF), geared to identify real world scenarios.

Test-driving in real-world traffic environment appears to be the most logical way to validate the safety of AVs and to evaluate and improve systems’ performance while allowing to take the complexity of the entire system into account. However, this might be dangerous, not scalable, and turned out to be impossible/very inefficient proposition as the needed vehicle kilometers to be driven -- to meet most relevant traffic situations -- are huge (billions) and the time needed would be by far too long (see also Kalra and Paddock, 2016). Therefore, various ways out of this dilemma have been identified and pursued.

As to the certification process two paradigms have been pursued in the past and become apparent today: on the one hand “self-certification” or “self-assessment” in the USA which encourages test-driving as much as possible and encourages industry to validate internally that designs meet best practice standards and a set of regulatory requirements, then leaving the remaining risk to automotive industry. The process is supported by a voluntary guidance document for industries and authorities with 10 priority safety design elements for consideration, such as *system safety*,

¹ RAND Corp. has carried out a comprehensive survey and developed *safety as a measurement, safety as a process and safety as a threshold* as categories of approaches for assessing AV safety (Blumenthal et al., 2020).

² Commission Implementing Regulation (EU) .../... on uniform procedures and technical specifications for the type-approval of motor vehicles with regards to their automated driving system (ADS), 2021, draft for discussions

operational design domain, object and event detection and response, fallback (minimal risk condition) validation methods, human machine interface, vehicle cybersecurity, crashworthiness, post-crash ADS behaviour and data recording, released by the Department of Transportation (USDOT/NHTSA, 2018).

On the other hand, the “type approval system” is mandatory for cars since 1998 in the European Union (EU) and adopted by other non-EU states. It operates with fully harmonized requirements, valid across borders, and heavily relies on international, notably UNECE³ regulations for technical rules while single State authority will finally decide on the certification. An evolving “progressive approach” supports large-scale testing but does not solely rely on physical test-driving and allows for virtual testing methods. It encompasses guidelines under the EU exemption procedure (EC Directive, 2007 Art. 20) and provides a new legal framework (EU Regulation, 2019), applicable from July 2022. Following the expressed need, a new generic assessment method on automated driving has been developed which is based on three pillars: (1) audit of the manufacturer design/development process, (2) confirmation of the audit/minimum performance in normal and emergency conditions before market placement through testing and (3) confirmation of the audit after release through continual feedback from the operational experience. Testing includes the demonstration of basic driving capabilities (on public road, on test-track) and of the ability to cope with main critical scenarios (by desktop simulation/on test-track).

Critical scenarios are defined as a sequence or combination of situations for assessing the functional requirements for automated vehicles and involve a wide range of elements such as roadway layouts, interaction with different types of road users and objects as well as environmental conditions. They are classified as *logical, functional and concrete scenarios* with decreasing level of abstraction. Based on intensive use of available and specifically generated data as well as on results of theoretical studies, a large set of adequate and representative critical scenarios is fleshed out and agreed upon by key actors against which automated vehicles have to be tested. They should be transferred to a scenario database as a common framework for manufacturers and authorities, subject to continuous update (see Fig. 1 for illustration).

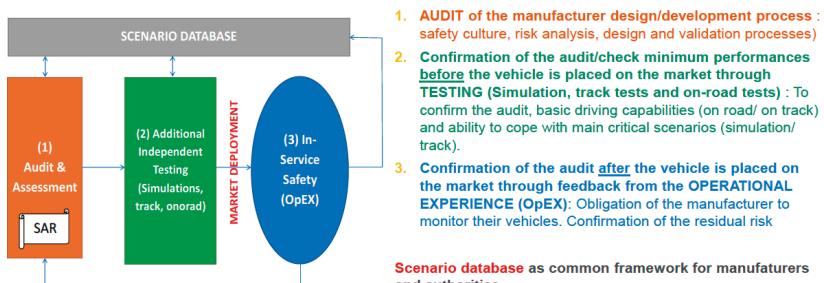


Fig. 1 Three pillars of the EU generic assessment method on automated driving (SAR stands for safety analysis report)

³ United Nations Economic Commission for Europe

The application should start with “easier cases”, focussed on Level 3 up to Level 4 automated vehicles, and advanced assistance systems in particular, as building blocks towards entire autonomous vehicles.

To provide technical regulations and uniform provisions concerning the approval of automated vehicles the UNECE has set up a working party on “Automated/autonomous and Connected Vehicles” (GRVA) with a set of informal working groups as the group on Functional Safety of Automated and Autonomous Vehicles (FRAV) and on Validation Method for Automated Driving (VMAD) and the task Force on Cyber Security (OTA) and Software Updates (VMAD). In accordance with the Framework Document on the Safety of Automated Vehicles⁴, the GRVA has proposed a regulation concerning the approval of vehicles with regard to Automated Lane Keeping Systems (ALKS) as first regulatory step for an automated driving system (ADS) in traffic and innovative provision aimed at addressing the complexity related to the evaluation of the system safety (UNECE/TRANS/WP.29 2020/8). The original text limits the operational speed to a maximum of 60 km/h, and further to passenger cars and activation of the ALKS under certain conditions on roads where pedestrians and cyclists are prohibited and divided lanes are physically separated. The general requirements relate to the system safety and fail-safe response; the possibility for the driver to override the system must be ensured at any time.

To support the process and to provide input to EC and UNECE working groups, a cooperative project between the EC DG GROW and the EC Joint Research Centre has been established. Their achievements include the draft of a voluntary Safety Guide with format and content of the information document for AV type approval, to be submitted by manufacturers and general procedures for different kinds of testing as well as contributions to ALKS regulation. This regulation should be expanded to Automated Lane Changing Systems (ALCS), motorway applications beyond 60 km/h maximum speed, and from passenger cars to other applications (e.g., valet parking, robot taxi/shuttles). Ongoing project work is focussed on testing concepts of general validity and for specific use-cases (e.g. highway chauffeur), on (quantitative) safety targets/thresholds, on analytical safety envelopes to define preventable accidents in traffic scenarios and on operational feedback recording and storage systems is ongoing and results are coming next.

As mentioned before, the services of the European Commission have expressed their views in a document with detailed annexes on uniform procedures and technical specifications for the type-approval of motor vehicles with regards to their ADS (see footnote ²). Nevertheless, the whole design-simulation-test-redesign-certification procedure is still not established, neither by industry nor the regulator. Most recently, the working group VMAD of the working party on automated and connected vehicles (GRVA), established by the UNECE in mid 2018, proposed a validation framework called New Assessment/Test Methods for AV (NATM) to foster ongoing innovation in the automotive industry. The framework (see Fig. 2) is based on several pillars and five validation methods including a catalogue of critical scenarios and simulations. The testing might follow a logical sequence from simulation (based on use of various simulation toolkits), to testing on dedicated tracks and then real-world testing.

⁴ UNECE/TRANS/WP.29/2019/34

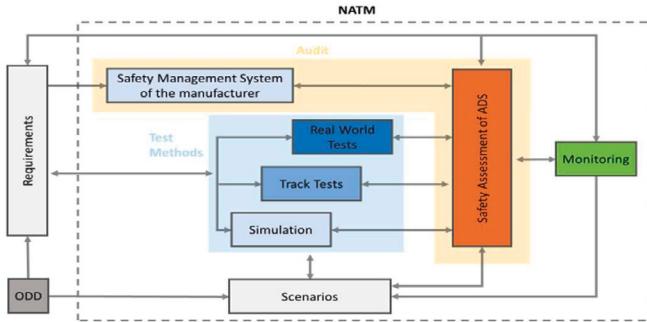


Fig. 2 Multi-pillar validation framework NATM and its integration with FRAV functional safety requirements (GRVA 2021), ODD stands for operational design domain

Harmonization of regulations is necessary and an ongoing effort at the UN level, in particular at the UNECE's Forum for Harmonization of Vehicle Regulation including Japan and China, among others.

3 Simulation-based methods and ways to reduce the parameter space

Simulation-based (or scenario-based) methods are a proposed alternative to the statistical approach of real-world testing and is favored as part of the type-approval process to validate automated driving functions. It is by far a much faster method of assessing advanced driver-assistance systems (ADAS) performance and safety where these systems have driven almost an order of magnitude more miles in simulation than in real-world testing. Based on the assumption that a large portion of road scenarios are uncritical in reality, it is proposed to identify “critical scenarios” out of a large set of developed scenarios and to expose single vehicles, equipped with automated systems, to exclusively critical scenarios to check their performance under “real world conditions” and on test benches in particular.

One problem with simulation-based methods is undermining the complexity of the real world and its uncertainties. This casts questions on coverage and completeness of scenarios driven in simulation (Kröger and Ayoub, 2021). More concretely, a good question to be answered in this direction is: how many miles driven in simulation equate to a mile in the real world?

Another main problem which emerges in the scenario-based approach is the parameter-space explosion. More abstractly, if the driving scenario has n parameters: P_1, P_2, \dots, P_n , and each parameter has m_i assumed discrete realizations, then the overall number of possible parameter combinations (scenarios) grows exponentially as m^n . For example, the cut-in scenario alone results in an order of 10^{23} parameter combinations or concrete scenarios (Amersbach and

Wimmer, 2019). This clearly indicates that a brute force method of trying all the scenarios doesn't work and even focusing on critical scenarios results in a huge number which still seems to exceed the existing capacity so far.

Novel approaches have been investigated to reduce the parameter space of critical scenarios and thus the test coverage significantly. Various attempts/proposals deserve attention. (Koné et al., 2020) proposed a hazardous behavior criterion with five severity classes for evaluation of scenarios identified by assuming functional insufficiencies. (Weber et al., 2020) proposed a simulation-based statistical approach to derive concrete scenarios for highly automated driving functions (SAE level 3 and higher) with a takeover process prompted by the vehicle. The methodology extends the framework of (Hallerbach et al., 2018) for the derivation of logical scenarios and encompasses the statistical evaluation and discretization of influence parameters identified by the traffic simulation package, their application to functional layers of the decomposed automated driving function, and finally a deterministic variation of previously discretized parameters which define a concrete scenario. The application to *cut-in* and *traffic-jam dissolution* functional scenarios showed a significant reduction of the parameter space.

Within a student's projects at ETH Zurich, a data-driven probabilistic framework was proposed, based on Stochastic Spectral Methods, namely Polynomial Chaos Expansion (PCE). Relying on both virtual and physical simulations of the autonomous vehicle system (considered as a black box), a dataset of input/ output pairs is used to train a metamodel, which surrogates an unknown, generically nonlinear criticality function, mapping the input scenarios to a risk metric. This so-called criticality function quantifies the severity of the input scenarios by weighting various Safety Performance Indicators obtained through experiments, and ultimately can help in reducing the input space dimensionality by identifying the most influential input parameters.

Another proposed methodology -- within a collaborative student's project between ETH Zürich and American University Beirut (AUB) -- was to use statistical learning techniques to reduce the dimensionality of the exploding parameter space. One prominent example is using Linear Discriminant Analysis (LDA) to project the initial huge input parameter space into a lower dimensional one, spanned by the most affecting parameters. The basic idea behind LDA is to find a linear combination of the input features which results in the maximal separation of different classes into distinct clusters. In this case, the input features are the parameters of the scenario, and the different classes represent varying levels of crash criticality (a binary class critical/non-critical is used for simplicity). Using this linear combination of features, the higher dimensional data can be projected into a much lower dimensional space without losing a lot of information. By looking at the Euclidean components of the projected vectors, one can determine which features have more weight affecting the output-label distribution, thus, giving an idea of which feature bears more importance regarding the criticality of the crash.

Finally, (Zanella, Shehab and Ayoub, 2021) proposed a practical algorithmic framework combining surrogate modeling and importance analysis in series. It starts with the identification of the input parameters (scenarios) along with their probability distributions based on historical data and experts' knowledge. From the constructed scenario space, a set of input scenarios is created using Latin hypercube sampling, which are then fed to a self-driving simulator, CARLA (Dosovitskiy et al., 2017), outputting various metrics of criticality. At this stage, an importance analysis method, based on Linear Discriminant Analysis (LDA) or Morris elementary effects, is

implemented on this input-output mapping dataset to determine which parameters are the most influential. The original input dataset is then refined by refining the selected most influential parameters, resulting in an expanded dataset that now includes scenarios sensitive to the most important parameters, which are then fed again into CARLA to calculate their criticality values. The researchers then propose to use this refined input-output mapping to surrogate the criticality function using PCE to reduce the computational burden of evaluating new input scenarios. After the surrogate model is generated, an optimization routine onto the fully differentiable surrogate criticality surface finds the most severe regions (critical regions) as the portions of the hypersurface where the criticality is above a certain threshold, called the criticality limit – defined by the user. The critical scenarios are defined then as the n-dimensional algebraic vectors containing the coordinates on the criticality surface identifying the parameter combination where the criticality function is higher than the criticality limit. Finally, they proposed to introduce a resampling strategy that refines the already built surrogate by increasing the chance that critical scenarios are sampled during the training phase by performing an informed sampling circumscribed within the already identified critical regions.

4 Complex system-of-systems and challenges to methods

Some expect highly and, notably, fully automated (SAE level 4 and level 5) vehicles, connected to other vehicles (V2V) and infrastructure (V2X) to evolve into a “complex system” or even into a “system-of-systems” rather than just into a “complicated system”. This distinction, with associated elements and attributes as well as challenges to methods, is considered worth to be carved out.

The term “complexity” is not well defined. However, it is commonly agreed that complexity is something with parts interacting with each other’s in multiple ways, culminating in a higher order of emergence greater than the sum of its parts. According to (Aven et al., 2015), complexity is when “it is not possible to establish an accurate prediction model of system behavior based on knowing the specific functions and states of its individual components”.

Characteristics of complex systems versus complicated systems are highlighted as follows, see (Kröger and Nan, 2019) for more details:

- Both system types entail a huge number of highly connected components, for complicated systems event frequency-consequence curves tend to follow a normal distribution while such curves for complex systems tend to have fat tails and follow a power law distribution.
- Rules of interaction between the components of complex systems may change over time and may not be well understood, while components of complicated systems have well-defined roles and are governed by prescribed interactions.
- Complex systems are more open, respond to external conditions and evolve, interact with their environment; structures do not remain closed and stable over time and the range of responses to changes in their environment is not limited, all in contrast to complicated systems.
- Complex systems tend to show high dynamic, emerging, and non-linear behavior, as well as sudden regime shifts; behaviors are not fully predictable, opposite to complicated systems.

- The overall behavior of complex systems cannot be described in terms of building blocks or by the sum of its parts as in complicated systems.

Despite the early stage of development, we conclude courageously that attributes and behaviors of complex systems may fully apply to envisaged coordinated, highly, or notably fully automated vehicles and associated mobility concepts. In this respect, they are considered similar to large cyber-physical networks of critical infrastructure systems such as the power grid. Thus, methods are needed for the proof of reliability and safety which are capable of mapping and analyzing the system as a whole entity, calling for a “systems approach”. Some adapted traditional methods based on “reliability theory” and thus on decomposition, like deductive Fault Tree Analysis (FTA), and causal chains, like inductive Failure Mode and Effects Analysis (FMEA) or Event Tree Analysis (ETA), which are successfully applied in other industrial sectors, may prove insufficient alone for safety validation of coordinated autonomous vehicles.

The traditional methods based on reliability theory have also been criticized because they focus on hardware component failures and do not sufficiently consider software failures and human interactions, in addition to not considering the system as a whole entity. STPA (Systems-Theoretic Process Analysis) has been developed (Leveson, 2011) to overcome these limitations in terms of identifying design errors, flawed requirements, human factors implications, software failures and unsafe and unintended component interaction failures. STPA uses a “feedback loop safety control structure” to identify unsafe scenarios and develops a detailed set of safety constraints/requirements and has been applied to various development aspects of autonomous vehicles. Various attempts have been made to compare STPA results with other methods (Kröger and Ayoub, 2021), as well as to combine them.

5 Short outlook

Vehicles of different degree of automation are under massive development and testing or even close to deployment. Certification requirements and rules are in the process of being structured at international and national level, with validation of sufficient functional and operational vehicle safety as well as the elimination of unreasonable risks as key elements. Adequate modelling and testing methods for different phases of development and safety validation are advancing and under early case applications, currently focused on advanced assistance systems. These efforts seem to be lagging the development of adequate methods for reliability/safety validation, at least for highly to fully automated cars.

Note: The first author is engaged in corresponding type-approval process activities at European level and ongoing methodological developments at academic level; progress achieved, and further results gained will be presented at the workshop.

References

- [1] C. Amersbach, H. Winner, Functional decomposition – A contribution to overcome parameter space explosion during interactions of highly automated driving, *Traffic Injury Prevention*, volume 20 (1), 2019
- [2] T. Aven, Y. Ben-Haim, H. B. Andersen, T. Cox, E. L. Drogueut, M. Greenberg, S. Guikema, W. Kröger, O. Renn, K. M. Thompson, *SRA Glossary*, Council of the Society of Risk Analysis (SRA), 2015
- [3] M. S. Blumenthal, L. Fraade-Blanar, R. Best, J. L. Irwin, *Safe enough*, RAND Corporation, 2020
- [4] DIRECTIVE 2007/46/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on Establishing a framework for the approval of motor vehicles and their trailers, and of systems, component and separate technical units intended for such vehicles, Sept. 2007
- [5] FRAV, Common safety requirements of autonomous vehicles, FRAV-02-05/Rev. 2, Jan. 2020
- [6] GRVA, New assessment/test method for automated driving (NATM), WP.29-183-05, March 2021.
- [7] A. Dosovitskiy, G. Ros, F. Codevilla, A. López and V. Koltun, "CARLA: An open urban driving simulator", Conference on Robot Learning (CoRL), 2017.
- [8] S. Hallerbach, Y. Xia, U. Eberle, F. Köster, Simulation-based identification of critical scenarios for cooperative and automated vehicles, *SAE Technical Papers*, April 2018
- [9] P. Junietz, W. Wachenfeld, K. Klonecki, H. Winner, Evaluation of different approaches to address safety validation of automated driving, 21st International Conference on Intelligent Transportation Systems (ITSC), 2018
- [10] N. Kalra, S. M. Paddock, How many miles of driving would it take to demonstrate autonomous vehicle reliability? RAND Corporation, 2016
- [11] T. Koné, E. Levrat, E. Bonjour, F. Mayer, S. Géronimi, Safety assessment of scenarios for the simulation-based validation process of AV with regards to its functional insufficiencies, Proc. of ESREL2020-PSAM15 conference, Venice, 2020
- [12] W. Kröger, C. Nan, Power systems in transition – Dealing with complexity, in C. Büscher, J. Schippl, P. Sumpf (editors), *Energy as a Sociotechnical Problem*, Routledge, 2019
- [13] W. Kröger, A. Ayoub, Autonomous driving: A survey with focus on reliability and risk issues, *Environment Systems and Decisions*, 2021 (invited article, under review)
- [14] N. G. Leveson, *Engineering a safer world: Systems thinking applied to safety (engineering systems)*, MIT Press Cambridge, 2011
- [15] REGULATION (EU) 2019/2144 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on Type-approval requirements for motor vehicles and their trailers, and systems, components and separate technical units intended for such vehicles, as regards their general safety and the protection of vehicle occupants and vulnerable road users, applicable from July 2022

- [16] UNECE, Regulation No. 157 on Uniform provisions concerning the approval of vehicles with regards to Automated Lane Keeping Systems, March 2021.
- [17] USDOT/NHTSA, Automated driving systems 2.0, 2018.
- [18] N. Weber, D. Frerichs, U. Eberle, A simulation-based, statistical approach for the derivation of concrete scenarios for the release of highly automated driving functions, AmE, GMM-Fachbericht, VDE, 2020.
- [19] M. Zanella, M. L. Shehab, A. Ayoub, Autonomous driving safety validation: A method to overcome parameter space explosion using Importance Analysis, Surrogate Modeling, and informed sampling, Technical paper, SATW Document (under review), 2021

Inferenz von Verhaltensattributen der Verhaltenssemantischen Szeneriebeschreibung für die Entwicklung automatisierter Fahrfunktionen

Felix Glatzki* und Hermann Winner†

Zusammenfassung: Ein entscheidender Schritt für den Sicherheitsvalidierungsprozess von hochautomatisierten Fahrzeugen (HAF) ist die Definition der Operational Design Domain (ODD). Die Verhaltenssemantische Szeneriebeschreibung ermöglicht eine explizite Beschreibung der verhaltensrelevanten Information einer statischen Verkehrsumgebung. Diese Information ist als Teil der ODD wertvoll für die Spezifikation des Systems. Wir zeigen wie die Verhaltensattribute basierend auf der vorliegenden Szenerie anhand von Regeln bestimmt werden. In einem Anwendungsbeispiel werden diese Regeln zur automatisierten Ableitung der Attribute implementiert. Dies ermöglicht die universelle Erfassung der Verhaltensregeln von Szenerien als Basis zur Charakterisierung der ODD und als Grundlage für die Bewertung der Verkehrsregelkonformität von HAF.

Schlüsselwörter: Automatisiertes Fahren, Operational Design Domain, Szeneriebeschreibung, Regelkonformität

1 Einleitung

Die Sicherheitsfreigabe von hochautomatisierten Fahrzeugen (HAF) ist immer noch eine ungelöste Herausforderung. Aktuelle Ansätze zum Nachweis der Sicherheit solcher Systeme erfordern aufgrund des Fehlens einer geschlossenen funktionalen Spezifikation einen wirtschaftlich nicht vertretbaren Aufwand [1]. Neben der Forderung nach Kollisionsfreiheit muss das Verhalten eines automatisierten Fahrzeugs verkehrsregelkonform sein. Hierfür ist es notwendig abzugrenzen, welches Verhalten regelkonform ist und welches nicht. Grundsätzlich werden Verkehrsregeln durch eine Verkehrsordnung vorgegeben (in Deutschland durch die Straßenverkehrsordnung (StVO)). Abhängig von der örtlichen Szenerie (Definition nach [2][3]) werden die vorgegebenen Regeln durch vorliegende Elemente (bspw. Verkehrszeichen) instanziert und beschränken das Fahrzeugverhalten in dieser Szenerie. Im Betrieb muss ein automatisiertes Fahrzeug die Verhaltenseinschränkungen innerhalb eines gewissen Horizonts nach vorne entlang der geplanten Route kennen und verstehen, um die Verkehrsregeln zu befolgen. Die Beschreibung der Szenerie, so wie sie ein Mensch wahrnimmt, führt zu einer hohen Komplexität hinsichtlich der beschriebenen Elemente, während

* Felix Glatzki ist Wissenschaftlicher Mitarbeiter am Fachgebiet Fahrzeugtechnik der TU Darmstadt, Otto-Berndt-Str. 2, 64287 Darmstadt (e-mail: felix.glatzki@tu-darmstadt.de).

† Hermann Winner ist Professor für Fahrzeugtechnik an der TU Darmstadt, Otto-Berndt-Str. 2, 64287 Darmstadt (e-mail: hermann.winner@tu-darmstadt.de).

der daraus resultierende Informationsgehalt hinsichtlich der Verhaltenseinschränkungen für ein Fahrzeug erheblich geringer ist. Dieses Wissen ist dann nur implizit in der Beschreibung der Elemente vorhanden und muss daraus abgeleitet werden.

Glatzki et al. [4] stellen Verhaltensattribute für eine Verhaltenssemantische Szeneriebeschreibung (Behavior Semantic Scenery Description, BSSD) vor. Diese Verhaltensattribute stellen die Grenzen des legalen Verhaltens explizit in semantischer Form zur Verfügung und knüpfen die Verhaltensregeln direkt an die Szenerie. Durch den Abstraktheitsgrad der Attribute ist die Anwendung nicht nur auf lokale Szenerien beschränkt, sondern ermöglicht eine Anwendung in verschiedenen, vielfältigen Verkehrsbereichen. In Tab. 1 werden alle für den weiteren Verlauf dieses Beitrages wichtigen Begrifflichkeiten aus der Veröffentlichung aufgelistet und ihre Bedeutung dargelegt. Für eine detaillierte Erklärung ist auf [4] verwiesen.

Die explizite Verfügbarkeit der Verhaltensgrenzen durch BSSD, ermöglicht verschiedene Anwendungsfälle. Zunächst dient es als Basis zur Charakterisierung und Auswahl der Operational Design Domain (ODD) [5] für HAF, da hiermit Verhaltensbeschränkungen konkret vorliegen und somit die benötigten Fähigkeiten eines Fahrzeugs in der Entwicklung damit abgeglichen und ein- bzw. ausgeschlossen werden können. Darauf basierend besteht die Möglichkeit, systematisch Anforderungen an HAF ausgehend von der ODD abzuleiten, wofür Lippert und Winner [6] eine Methodik vorstellen. Denkbar ist auch ein Einsatz als Vorgabe für Verhaltensplanungsalgorithmen, die durch die universelle Darstellung der Verhaltensgrenzen in Gebieten mit unterschiedlichen Regelwerken eingesetzt werden können, ohne große Anpassungen vorzunehmen. Zuletzt besteht das Ziel, die mit den Verhaltensattributen verbundenen Regeln zu formalisieren und als Testkriterien für die Verkehrsregelkonformität zu nutzen.

Gerade für die Anwendung auf große Streckennetze bzw. den Einsatz innerhalb von Verhaltensplanungsalgorithmen, ist die automatisierte Ableitung der Verhaltensattribute von hoher Bedeutung. Hierdurch können große Datenmengen in kurzer Zeit unter Ausschluss von menschlichen Fehlern bearbeitet werden. In dieser Veröffentlichung legen wir die Regeln zur Ableitung der Verhaltensattribute dar. Durch Hinterlegung der Regeln innerhalb der Verhaltenssemantischen Szeneriebeschreibung von gesamten Streckennetzen (Verhaltenskarte), ist die benötigte Kenntnis der Verhaltenseinschränkungen entlang der geplanten Route explizit verfügbar. Innerhalb von Beispielen zeigen wir die automatisierte Ableitung der Verhaltensattribute.

Tabelle 1: Glossar der Begrifflichkeiten.

Begriff	Bedeutung
Szenerie	Räumlich stationäre Aspekte der Fahrzeugumgebung [16,17]
Verhaltensraum	Abgegrenzte Menge gesetzlich möglicher Verhaltensweisen
Verhaltensattribute	Dimensionen des Verhaltensraumes zur Beschreibung der Verhaltensregeln
Atomarer Verhaltensraum	Straßensegment, innerhalb dessen sich die Verhaltensattribute nicht ändern
Indikationselemente	Elemente der Szenerie, die eine Regel anzeigen/vorgeben

2 Stand der Forschung

Verwandte Arbeiten zeigen, dass Ontologien [7,8,9,10] ein häufig verwendeter Ansatz zur Modellierung von Szenerien mit ihren zugehörigen Elementen sind. Ontologien werden bislang aber unseres Wissens nach nicht für die Modellierung von Verhaltensregeln verwendet. Eine weitere Form, die Szenerie zu beschreiben, ist die Nutzung von hochgenauen Kartenformaten [11,12]. Diese Karten sind im Hinblick auf das geforderte Verhalten nur wenig abstrahiert und stellen meist die Elemente der Verkehrsumgebung detailliert nach. Dies führt zu großen Datenmengen mit geringem Informationsgehalt bezüglich der automatisierten Fahraufgabe. Lanelet2 [12] zeigt allerdings vielversprechende Ansätze, da die Szenerie u.a. auf regulatorische Elemente abstrahiert wird. Diese regulatorischen Elemente bieten die Möglichkeit, Verkehrsregeln zu hinterlegen. Jedoch wird die explizite Verhaltensanforderung nicht angegeben. Beispielsweise verweist das Regelement „Lichtsignalanlage“ nur auf die jeweilige Lichtsignalanlage (LSA) und die korrespondierende Haltelinie. Es sagt nicht explizit aus, dass es nicht erlaubt ist, bei einer Rotphase über die Haltelinie zu fahren. Butz et al. [13] stellen einen Ansatz zur Abstraktion von Szenerien und Situationen hinsichtlich erforderlichem Systemverhalten vor. Die vorgestellten *zone graphs* sind spezifisch für einzelne Szenerien und somit nicht universell anwendbar. Zudem wird nicht nur das rein gesetzlich vorgegebene Verhalten beschrieben, sondern auch daraus resultierende Fähigkeiten und Verhaltensweisen abgeleitet. Auch die Regeln zur Ableitung der zone graphs bleiben offen.

Die Ableitung von Verhaltensanforderungen für HAF in der Literatur basiert auf formalisierten Verkehrsregeln [14,15], Regeln des gesunden Menschenverstands [16] oder Sicherheitsanalysen [17]. Zusätzlich kann Expertenwissen genutzt werden, um die resultierenden Verhaltenssätze zu ergänzen. Neben der Anforderungsgerstellung werden Verhaltenssätze einschließlich Verkehrsregeln auch zur Überprüfung der Sicherheit von Trajektorien, die von Verhaltensplanern generiert werden, verwendet [18]. In Vorarbeiten [4] wurde ein Ansatz beschrieben, der Informationen über die Verhaltensregeln aus der Szenerie extrahiert, indem sie in Form von Verhaltensattributen als Dimensionen des Verhaltensraumes beschrieben werden. Dieser Ansatz wird als Grundaufbau für diese Veröffentlichung verwendet und daher im nächsten Abschnitt dargestellt.

3 Verhaltensattribute

Für das weitere Verständnis werden in diesem Abschnitt die einzelnen Verhaltensattribute, die einen atomaren Verhaltensraum aufspannen, kurz erläutert. In einer parallelen Arbeit entwickeln wir eine gesamtheitliche Beschreibung, die auch die Verknüpfungen zwischen den einzelnen atomaren Verhaltensräumen abbildet (Nachbarn, Vorgänger/Nachfolger). Diese Verknüpfungen sind notwendig, um zu identifizieren, welche Verhaltensbeschränkungen beim Eintritt in die Segmente entlang der geplanten Route vorliegen, um entsprechend das Verhalten rechtzeitig anzupassen. Beispielweise muss die Fahrzeuggeschwindigkeit schon bei Erreichen eines Geschwindigkeitslimits herabgesenkt sein.

Geschwindigkeitslimit

Dieses Attribut definiert die *zulässige Höchstgeschwindigkeit* innerhalb des atomaren Verhaltensraums. Ein Fahrzeug, das sich in diesem Raum bewegt, darf nicht schneller als diese Geschwindigkeit fahren. Zusätzlich zur Geschwindigkeitsbegrenzung selbst kann es eine *zeitliche Einschränkung* und *andere Bedingungen* geben, die für die Gültigkeit des Limits vorliegen müssen.

Begrenzungen

Dieses Attribut beschreibt die zugehörigen Verhaltensregeln für das Überschreiten der Grenzen eines atomaren Verhaltensraums. Das Überschreiten von Grenzen kann *erlaubt*, *bedingt erlaubt*, *verboten* oder *nicht möglich* sein. Es wird zwischen Längs- und Querbegrenzungen unterschieden.

Reservierung

Die Reservierung legt die Bedingungen fest, die erfüllt sein müssen, um einen Raum aus Vorrangicht zu betreten und/oder darin zu bleiben. Jeder atomare Verhaltensraum ist für eine bestimmte (oder mehrere) *Klasse(n)* von *Verkehrsteilnehmern* reserviert. Diese Klasse von Verkehrsteilnehmern darf innerhalb dieses Raumes nicht durch andere Verkehrsteilnehmer behindert werden. Die *Reservierungsart* definiert in Bezug auf die Fahrtrichtung, für welche Verkehrsteilnehmerklasse der zugehörige Szenerieabschnitt reserviert ist. Eine Reservierungsart kann *eigenreserviert* (für die betrachtete Verkehrsteilnehmerklasse), *fremdreserviert* (für andere Verkehrsteilnehmerklassen) oder *gleichreserviert* (wenn es keine eindeutige Vorrangregel gibt) sein. Ist die Reservierung nicht vom Typ *eigen-reserviert*, kennzeichnet ein *Link* die Bereiche, aus denen die reservierungsberechtigten Verkehrsteilnehmer kommen können.

Überholen

Dieses Attribut definiert die Erlaubnis, andere Verkehrsteilnehmer zu überholen. Ein Überholvorgang findet statt, wenn zwei Verkehrsteilnehmer auf derselben Fahrbahn mit unterschiedlichen Geschwindigkeiten in dieselbe Richtung fahren und das schnellere Fahrzeug das langsamere Fahrzeug überholt. Dabei spielt es keine Rolle, ob sich die beiden Fahrzeuge im gleichen Fahrstreifen befinden oder nicht. Das Überholen kann *verboten* oder *erlaubt* sein.

4 Vorgehen zur Ableitung der Verhaltensattribute

In Abschnitt 1 haben wir das Ziel einer *Verhaltenskarte* eines Streckennetzes in Form der Verhaltenssemantischen Szeneriebeschreibung motiviert. Diese Verhaltenskarte dient sowohl zur Charakterisierung und Spezifizierung der ODD als auch zur Nutzung als Vorgabe der Verhaltengrenzen im Fahrzeug. Hochgenaue Karten [11][12] liefern ein detailliertes Abbild der Szenerie. Sie enthalten sowohl die Straßengeometrie, Markierungen, Verkehrszeichen als auch sonstige Infrastruktur eines Verkehrsbereichs. Sie stellen somit die Quelle der notwendigen Information zur Ableitung der Verhaltensattribute dar. Zur Ableitung der Verhaltensattribute muss ein Streckennetz, das in Form einer hochgenauen Karte vorliegt, zunächst in Abschnitte entsprechend der *atomaren Verhaltensräume* unterteilt werden. Die

atomaren Verhaltensräume werden durch die vier Attribute *Geschwindigkeitslimit*, *Begrenzung*, *Reservierung* und *Überholen* aufgespannt. Die atomaren Verhaltensräume sind dementsprechend einzelnen Szenerieabschnitten zugewiesen, die wiederum *Indikationselemente* enthalten, welche die im Abschnitt gültigen Regeln instanziiieren und vorgeben. Abb. 1 fasst diesen Rahmen für die Ableitung der Verhaltensattribute in Form eines Entity-Relationship-Modells zusammen.

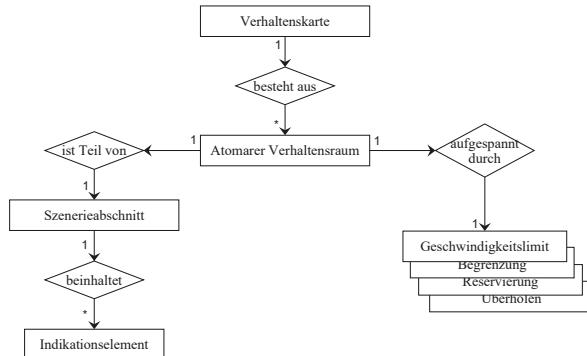


Abbildung 1: Entity-Relationship-Modell Verhaltenskarte und atomarer Verhaltensraum.

Innerhalb der atomaren Verhaltensräume werden die Verhaltensattribute aus den vorhandenen Indikationselementen abgeleitet. In diesem Abschnitt wird das Vorgehen zur Bestimmung der Verhaltensattribute innerhalb eines atomaren Verhaltensraumes vorgestellt. Dies bildet die Grundlage zur Implementierung einer automatisierten Ableitung von Verhaltensattributen auf Basis von Kartendaten. Abb. 2 zeigt die Zusammenhänge zwischen den einzelnen Entitäten, die zur Bestimmung der Verhaltensattribute führen.

Die Grundlage für die Ableitung der Verhaltensattribute bilden die *Indikationselemente*, die innerhalb des betrachteten *Szenerieabschnitts* liegen. Innerhalb eines Abschnitts kann eine beliebige Anzahl an Indikationselementen vorhanden sein. Mit ihnen sind direkt *Verhaltenswirkungen* verbunden. Eine Verhaltenswirkung beschreibt die von dem Indikationselement ausgehende Verhaltenseinschränkung auf das Fahrzeug. Somit bedient eine Verhaltenswirkung eines der Verhaltensattribute. Ein Stoppsschild hat beispielsweise die Verhaltenswirkung, dass an der korrespondierenden Haltelinie gehalten werden muss, bevor in den Abschnitt eingefahren werden darf und bedient damit das Begrenzungsattribut. Dabei kann ein Indikationselement mehrere Verhaltenswirkungen hervorrufen. Im Rahmen der Analyse der *Straßenverkehrsordnung* wurden alle Verkehrszeichen auf entsprechende Auswirkungen geprüft und die entsprechende Verhaltenswirkung mit ihnen verknüpft. Neben den

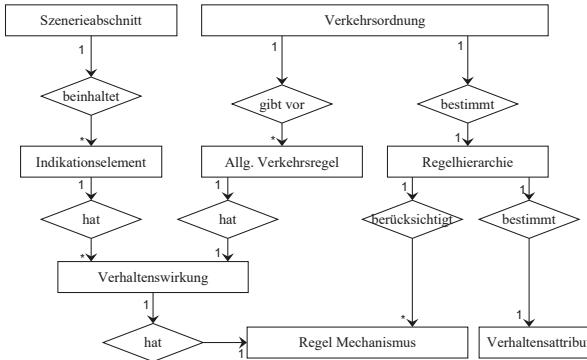


Abbildung 2: Entity-Relationship-Modell zur Bestimmung der Verhaltensattribute.

durch Indikationselemente hervorgerufenen Verhaltenswirkungen, können Verhaltenswirkungen auch durch *allgemeingültige Verkehrsregeln* hervorgerufen werden. Dies sind Verkehrsregeln, die auch ohne ein vorhandenes Indikationselement gültig sind. Beispielsweise gilt in Deutschland an Kreuzungen ohne weitere Verkehrszeichen die „Rechts-vor-Links“-Regel. Diese Regeln wurden ebenfalls aus der Straßenverkehrsordnung extrahiert und mit ihrer Verhaltenswirkung verbunden.

Nun liegen die Verhaltenswirkungen einzelner Indikationselemente sowie der allgemeinen Verkehrsregeln innerhalb des Abschnitts des atomaren Verhaltensraumes vor. Diese konkurrieren jedoch untereinander, da wie zuvor beschrieben, zum Beispiel einzelne Indikationselemente dafür sorgen können, dass Verkehrsregeln, die ohne Verkehrszeichen gültig sind, überschrieben werden. Ein Verkehrszeichen „Vorfahrtsstraße“ überschreibt beispielsweise die Regel „Rechts-vor-Links“. Dies macht eine *Hierarchie* für die gefundenen Verhaltenswirkungen notwendig. Mit dieser Hierarchie wird bestimmt, welche der konkurrierenden Auswirkungen gültig ist und daraus die entsprechenden *Verhaltensattribute* bestimmt, die den Verhaltensraum aufspannen.

Hierfür wird zunächst darauf eingegangen, nach welchen Mechanismen Regeln im deutschen Straßenverkehr hervorgerufen werden. Poggenhans [19] identifiziert hierfür vier unterschiedliche Mechanismen, die abgewandelt für diese Veröffentlichung übernommen werden. Die *Zonen-Regelung* definiert Zonen, innerhalb der bestimmte Verkehrsregeln gelten. Die Zonen sind so lange gültig, bis sie durch ein zugehöriges Verkehrszeichen aufgehoben werden. An *Straßen oder Strecken gebundene Verkehrsregeln* enden sobald diese verlassen wird oder sie durch ein entsprechendes Zeichen aufgehoben werden. Die von Poggenhans identifizierten Mechanismen „Ampeln“ und „Vorfahrtregelungen“ werden im Mechanismus *Lokale Regeln* zusammengefasst. Diese Regeln wirken auf einen lokal beschränkten Bereich. Im Gegensatz zu Poggenhans beschränken sich in dieser Betrachtung die lokalen Regeln nicht nur auf Lichtsignalanlagen, da auch andere Verkehrszeichen eine lokale Verhaltensauswirkung haben können (z.B. zwingt ein Stoppschild zum Halten an der Haltelinie). Da einige Vorfahrtregelungen ebenfalls in einem lokal beschränkten Bereich

gültig sind, werden sie ebenfalls mit in diesen Mechanismus aufgenommen. Des Weiteren werden auch Vorrangregelungen aufgenommen und somit nicht nur Konfliktsituationen mit anderen Fahrzeugen betrachtet, sondern ebenfalls mit Fußgängern und anderen Verkehrsteilnehmern. Bei gleichzeitigem Vorhandensein von verschiedenen Verhaltenswirkungen, die das gleiche Verhaltensattribut bedienen, gilt als erstes die lokale Regelung, dann die an Strecken gebundene Regelung, dann die Zonenregelung und zuletzt die allgemeinen Verkehrsregelungen. Innerhalb der lokalen Regelungen ergibt sich nochmal eine untergeordnete Hierarchie. Hierbei gilt, dass die Regelungen durch Weisungen von Polizeibeamten vor Lichtzeichen und Lichtzeichen vor Verkehrszeichen gelten.

An Verzweigungen (bspw. Kreuzungen) ist aufgrund der Überlappung verschiedener Fahrstreifen eine weitere Logik notwendig, um auf die in dem Reservierungsattribut abgelegten Vorrangregeln zu schließen. An Kreuzungen ohne Lichtsignalanlagen besteht die Möglichkeit der „Rechts-vor-Links“-Regel auf Basis der allgemeinen Verkehrsregeln oder es wird durch Indikationselemente eine bevorrechtigte Straße vorgegeben (dies beinhaltet abknickende Vorfahrtstraßen). Exemplarisch ist in Abb. 3 eine fünf-armige Kreuzung für beide Fälle dargestellt.

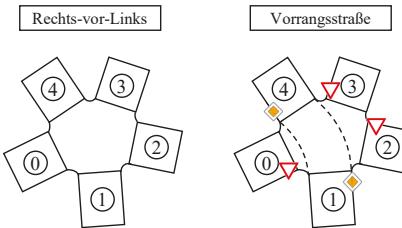


Abbildung 3: Exemplarische Kreuzungen zur Vorrangableitung.

Zur Ableitung der Vorrangregeln werden die Kreuzungsarme auf Basis ihrer Ausrichtung gegen den Uhrzeigersinn nummeriert, um die topologische Anordnung abzubilden. Die Nummerierung kann an einem beliebigen Arm beginnen. Eine weitere geometrische Bestimmung ist nicht notwendig. Zusätzlich werden alle möglichen Verbindungen in Form von richtungsabhängigen Tupeln hinterlegt. Eine Verbindung von 1 nach 4 wird durch das Tupel (1,4) beschrieben. Grundsätzlich muss nur den Verbindungen mit geometrischer Überlappung potenziell Vorrang gewährt werden. An einer „Rechts-vor-Links“-Kreuzung muss allen Eingängen, die innerhalb dieses Tupels liegen, Vorrang gewährt werden. Für die Verbindung (1,4) sind das beispielsweise die Eingänge 2 und 3. An der Kreuzung mit Vorrangstraße muss zwischen drei Fällen unterschieden werden.

1. Entlang der Vorrangstraße muss keinem anderen Eingang Vorrang gewährt werden ((1,4) und (4,1))
2. Bei Einfahrt über eine bevorrechtigte Straße, jedoch Ausfahrt in eine nachrangige Straße, muss bevorrechtigten Eingängen innerhalb des Tupels Vorrang gewährt werden. Beispielsweise muss die Verbindung (1,0) dem Eingang 4 Vorrang gewähren
3. Bei Einfahrt über und Ausfahrt in eine nachrangige Straße muss den Eingängen der Vorrangstraße sowie Eingängen innerhalb des Tupels Vorrang gewährt werden. Beispielsweise muss die Verbindung (2,0) den Eingängen 1, 3 und 4 Vorrang gewähren.

5 Implementierung und Anwendungsbeispiel

In diesem Abschnitt wird die Implementierung des zuvor beschriebenen Vorgehens zur Ableitung der Verhaltensattribute anhand von einer Beispieldatenbasis vorgestellt. Hierfür wurde eine OpenDRIVE Karten erstellt, die als Basis für die Ableitung der Attribute genutzt wird. Bisher ist die Umsetzung auf einstreifige Straßen je Fahrtrichtung beschränkt. In dem betrachteten Beispiel wird eine abknickende Vorfahrt betrachtet. Zudem werden Zonenregelungen über Ortseingangsschilder sowie Streckenregelungen über Geschwindigkeitsbegrenzungszeichen eingeführt. Eine Übersicht über die Beispieldatenbasis findet sich in Abb. 4.

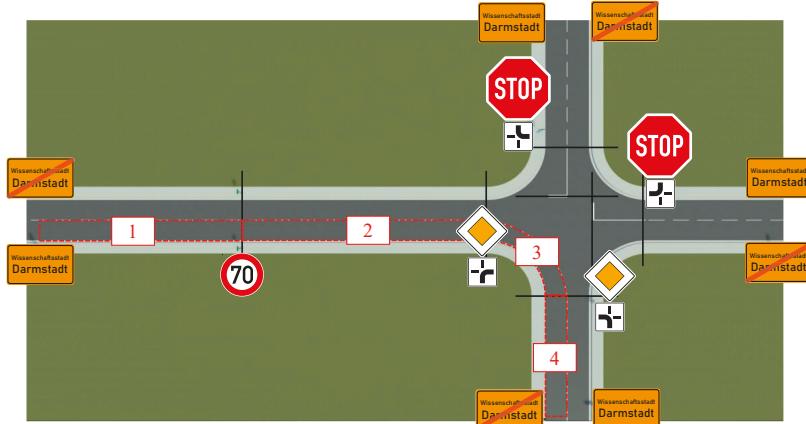


Abbildung 4: Beispieldatenbasis zur Anwendung der Ableitung von Verhaltensattributen.
Eigene Darstellung basierend auf Visualisierung der OpenDRIVE Karte mit MathWorks® RoadRunner

Die Implementierung gliedert sich in die in Abb. 5 gezeigten Schritte. Das Einlesen der Karte zur Nutzung mit Python erfolgt mit Hilfe des OpenDRIVE Parser aus dem Common-Road Scenario Designer der TUM [21]. Nachfolgend werden die weiteren Schritte der Implementierung erläutert.

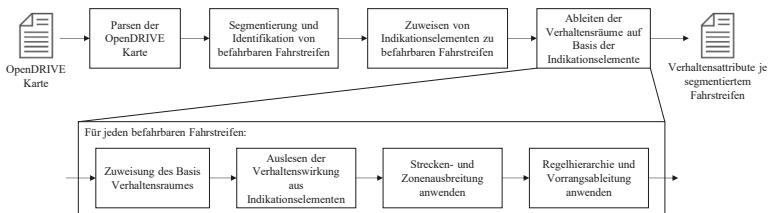


Abbildung 5: Implementierung des Vorgehens zur Ableitung der Verhaltensattribute.

Segmentierung und Identifikation von befahrbaren Fahrstreifen

Als erstes werden auf Basis des Fahrstreifentyps aus OpenDRIVE befahrbare Fahrstreifen identifiziert. Bereiche außerhalb des regulären Bewegungsbereiches erhalten keine Verhaltensraumbeschreibung, da ein Fahrzeug sich dort nicht aufhalten soll. Die Attribute könnten jedoch hier genauso verwendet werden. Wie in Abschnitt 3 erläutert, werden die Verhaltensattributen in Form von atomaren Verhaltensräumen einzelnen Szenerieabschnitten zugewiesen. Dies bedeutet, dass die befahrbaren Fahrstreifen in einzelne Segmente unterteilt werden. Dies geschieht mit Hilfe eines Tools, dass bei jedem Element, das eine Verhaltenseinschränkung hervorruft, eine Segmentierung vornimmt. Bei einer Analyse des Verkehrsraums wurden folgende Elemente identifiziert:

- Straßentopologie (Anzahl und Anordnung der Fahrstreifen)
- Kreuzungen und Verzweigungen (als Auslöser für Vorrangregeln)
- Fahrbahn-/streifenbegrenzungen (Fahrstreifenmarkierung, Bordstein, ...)
- Verkehrszeichen (inkl. Fahrbahnmarkierungen)
- Statische undurchfahrbare Objekte (Leitplanke, Schranke, ...)

Hierbei besteht die Möglichkeit, dass aufgrund aufeinanderfolgender Elemente mit gleicher Verhaltenswirkung eine Segmentierung vorgenommen wird, die in identischen aufeinanderfolgenden Verhaltensräumen resultiert. In zukünftigen Arbeiten wird eine Logik implementiert, die diese Segmente wieder zusammenfügt, somit ist die genaue Kenntnis der Segmentierung vor dem Ableiten der Verhaltensattribute nicht notwendig. Bei Bedarf kann aktuell eine manuelle Korrektur dieser Segmentierung erfolgen. Bei der Segmentierung ist der Top Down Aufbau des OpenDRIVE Formats ein Vorteil, da die Zuweisung und Verbindung von Straßen bereits vorhanden und hinterlegt sind. Es ergibt sich die in Abb. 3 durch schwarze Querstriche gezeigte Segmentierung.

An Verzweigungen bspw. Kreuzungen wird grundsätzlich auf Basis der verschiedenen möglichen Pfade segmentiert (hier: links abbiegen, rechts abbiegen, kreuzen). Zusätzlich gibt es im vorliegenden Beispiel eine Segmentierung auf Basis der Haltelinie, da hierdurch das longitudinale Begrenzungsattribut beeinflusst wird. Ansonsten erfolgen Segmentierungen auf Basis sich ändernder Geschwindigkeitsvorgaben.

Zuweisen von Indikationselementen

Im nächsten Schritt werden die Indikationselemente den einzelnen Fahrstreifen-Segmenten zugewiesen. Hierfür wird die Ausrichtung der Verkehrszeichen berücksichtigt. Im vorliegenden Beispiel werden die Verkehrszeichen den entsprechenden Segmenten zugewiesen. Die Hinterlegung von Indikationselementen in der Top Down Struktur von OpenDRIVE stellt hierbei eine Herausforderung dar. Zum Beispiel werden Fahrbahnmarkierung in einer anderen Form und an anderer Stelle abgelegt als Bordsteine, obwohl beides Begrenzungen eines Fahrstreifens darstellen. Bei der Implementierung muss also auf korrekte Zuweisung der Elemente besonders geachtet werden.

Ableiten der Verhaltensräume

Als erstes bekommt jedes Fahrstreifensegment einen Basis-Verhaltensraum zugewiesen. Dieser Verhaltensraum deckt die allgemeinen Verkehrsregeln (ohne jegliche Indikationselemente) ab. Anschließend werden die Verhaltenswirkungen inklusive Regelmechanismus

aus der erstellten Datenbank der Indikationselemente ausgelesen. Da die Indikationselemente von Zonen-Regelungen und an Strecken gebundenen Regelungen nicht zwangsläufig in jedem Segment wiederholt werden, ergibt sich die Notwendigkeit, Informationen über die Verhaltenswirkungen und deren Gültigkeitsmechanismus von vorherigen Verhaltensräumen zu kennen. Hierfür wird die Strecken- und Zonenausbreitung auf Basis der Indikationselemente im Streckennetz durchgeführt.

Abschließend werden auf Basis der Regelhierarchie die Verhaltensattribute bestimmt und ausgegeben. Hier werden aus Platzgründen beispielhaft die rot hervorgehobenen Fahrstreifensegmente (1, 2, 3 und 4) betrachtet. Für die Zonenausbreitung wird jedem Segment, das ein zonenauslösendes Indikationselement (hier: Ortseingangsschild) beinhaltet, die entsprechende Verhaltenswirkung und Richtung der Zone zugewiesen. Nun werden auf Basis der OpenDRIVE-Informationen die Nachfolger durchlaufen bis ein zonenbeendendes Zeichen (hier: Ortsausgangsschild) erreicht wird. Alle durchlaufenden Segmente erhalten die Verhaltenswirkung der Zone (Geschwindigkeitslimit 50 km/h). Da Fahrstreifensegment 1 noch vor der Streckenbegrenzung liegt, erhält es das Geschwindigkeitsattribut 50 km/h. Analog werden für die streckenbezogenen Regeln (hier: Geschwindigkeitslimit 70 km/h) die startenden Indikationselemente identifiziert. Der Unterschied ist nun, dass an Kreuzungen nicht in alle nachfolgenden Richtungen die Fahrstreifen durchlaufen werden, sondern nur entlang der Routenregelung. An der abknickenden Vorfahrt wird dementsprechend nach rechts abknickend (Fahrstreifensegment 2) fortgesetzt, bis das streckenbeendende Ortsausgangsschild erreicht wird. Dementsprechend erhalten Fahrstreifensegment 2, 3 und 4 das Geschwindigkeitsattribut 70 km/h, da die Streckenregelung einen höheren Regelrang als die Zonenregelung hat. Abschließend überschreibt die Vorrangregelung durch die abknickende Vorfahrt die Vorrangregelung des Basis-Verhaltensraumes. Hierbei wird die in Abschnitt 4 beschriebene Logik angewendet. Durch die abknickende Vorfahrt resultiert ein eigenreservierter Bereich in Fahrstreifensegment 3. Nachfolgend sind in Tab. 2 die resultierenden Verhaltensräume aus der automatisierten Ableitung der vier Fahrstreifensegmente aufgelistet:

Tabelle 2: Resultierende Verhaltensattribute für die vier Fahrstreifensegmente.

Attribut	Fahrstreifen-Segment 1	Fahrstreifen-Segment 2	Fahrstreifen-Segment 3	Fahrstreifen-Segment 4
Geschwindigkeitslimit	Max: 50 km/h	Max: 70 km/h	Max: 70 km/h	Max: 70 km/h
Begrenzung	<i>Long:</i> allowed <i>Links:</i> allowed <i>Rechts:</i> verboten			
Reservierung	<i>Typ:</i> eigenreserviert	<i>Typ:</i> eigenreserviert	<i>Typ:</i> eigenreserviert	<i>Typ:</i> eigenreserviert
Überholen	<i>Erlaubnis:</i> ja	<i>Erlaubnis:</i> ja	<i>Erlaubnis:</i> ja	<i>Erlaubnis:</i> ja

6 Fazit und Ausblick

In dieser Veröffentlichung werden die Regeln zur Ableitung von Verhaltensattributen vorgestellt und angewendet. Dies ermöglicht, den atomaren Verhaltensraum der Szenerie automatisiert auf Basis der vorhandenen Indikationselemente zu bestimmen. Innerhalb von einem Anwendungsbeispiel wird die praktische Umsetzung des Ansatzes gezeigt. In Zukunft wollen wir die Anwendung auf mehrstreifige Straßenlayouts erweitern. Die automatisierte Ableitung von Verhaltensattributen bildet die Grundlage für die Anwendung der BSSD auf gesamte Streckennetze. Hierfür wurde die Anforderung identifiziert, die Zusammenhänge zwischen verschiedenen atomaren Verhaltensräumen erfassen und darstellen zu können. Derzeit arbeiten wir an einer Methode zur Beschreibung ganzer Verkehrsnetze. Diese Beschreibung ist in der Lage, die identifizierte Anforderung zu erfüllen und somit eine vollständige Repräsentation des Verhaltensraums für ein ganzes Verkehrsgebiet zu schaffen. In Zukunft planen wir mit Hilfe einer Formalisierung der Verhaltensattribute die Verkehrsregelkonformität unabhängig vom Einsatzgebiet zu spezifizieren und somit Testkriterien für das Verhalten von HAF abzuleiten. Diese Kriterien werden innerhalb einer Testsuite zur Prüfung der Verkehrsregelkonformität abgetestet, um die Fähigkeit eines HAF Verkehrsregeln einzuhalten, zu bewerten. Der Vorteil besteht hierbei in der universellen Anwendbarkeit der Testkriterien, unabhängig von der Ausprägung der Indikationselemente des Einsatzgebietes.

7 Danksagung

F. G. dankt Continental für die gute Zusammenarbeit und Unterstützung im Rahmen von PRORETA 5, einem gemeinsamen Forschungsprojekt der Universität Bremen, der Technischen Universität Darmstadt, der TU Iasi und Continental zur Erforschung von Zukunftskonzepten für autonome Fahrsysteme.

Wir danken Moritz Berghöfer für die Umsetzung des Tools zur Segmentierung des Streckennetzes als Basis für die implementierten Methoden.

Literatur

- [1] W. Wachenfeld und H. Winner, "The Release of Autonomous Vehicles", in *Autonomous Driving: Technical, Legal and Social Aspects*, M. Maurer, B. Lenz, H. Winner, and J. C. Gerdes, Eds., s.l.: Springer, 2016, S. 425-449.
- [2] S. Geyer, M. Baltzer, B. Franz, S. Hakuli, M. Kauer, M. Kienle, S. Meier, T. Weißgerber, K. Bengler, R. Bruder, F. Flemisch, H. Winner, "Concept and development of a unified ontology for generating test and use-case catalogues for assisted and automated vehicle guidance", in *IET Intelligent Transport Systems*, vol. 8, no. 3, pp. 183–189, 2014, doi: 10.1049/iet-its.2012.0188.
- [3] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, und M. Maurer, "Definition der Begriffe Szene, Situation und Szenario für das automatisierte Fahren", in *Workshop Fahrerassistenzsysteme und automatisiertes Fahren*, Bd. 10. Walting im Altmühlthal, 2015.

- [4] F. Glatzki, M. Lippert und H. Winner, "Behavioral Attributes for a Behavior-Semantic Scenery Description (BSSD) for the Development of Automated Driving Functions", in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 667-672, doi: 10.1109/ITSC48978.2021.9564892.
- [5] Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, J3016, SAE International (SAE), 2021.
- [6] M. Lippert und H. Winner, "How to Derive Behavioral Requirements for Automated Driving from a Behavior-Semantic Scenery Description", in *Workshop Fahrerassistenzsysteme und automatisiertes Fahren*. Bd. 14. Berkheim: Uni-DAS e.V., 2022.
- [7] G. Bagschik, T. Menzel, und M. Maurer, "Ontology based Scene Creation for the Development of Automated Vehicles", in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Changshu, 2018, pp. 1813-1820.
- [8] S. Ulbrich, T. Nothdurft, M. Maurer, und P. Hecker, "Graph-based context representation, environment modeling and information aggregation for automated driving," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, MI, USA, 2014, pp. 541-547.
- [9] M. Buechel, G. Hinz, F. Ruehl, H. Schroth, C. Gyoeri, und A. Knoll, "Ontology-based traffic scene modeling, traffic regulations dependent situational awareness and decision-making for automated vehicles", in *28th IEEE Intelligent Vehicles Symposium: June 11-14, 2017*, Redondo Beach, California, USA, Los Angeles, CA, USA, 2017, pp. 1471-1476.
- [10] R. Regele, "Using Ontology-Based Traffic Models for More Efficient Decision Making of Autonomous Vehicles", in *Fourth International Conference on Autonomic and Autonomous Systems*, 2008, Gosier, Guadeloupe, 2008, S. 94-99.
- [11] ASAM OpenDRIVE: Offene dynamische Straßeninformationen für die Fahrzeugumgebung: ASAM, 2020. Accessed: Nov. 6 2020. [Online]. Verfügbar: <https://www.asam.net/standards/detail/opendrive/>.
- [12] F. Poggenhans, J.-H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, M. Mayr, "Lanelet2: A high-definition map framework for the future of automated driving", in *2018 IEEE Intelligent Transportation Systems Conference*, Maui, HI, 2018, pp. 1672-1679.
- [13] M. Butz, C. Heinemann, M. Herrmann, J. Oehlerking, M. Rittel, N. Schalm, D. Ziegenbein, "SOCA: Domain Analysis for Highly Automated Driving Systems," 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), 2020, pp. 1-6, doi: 10.1109/ITSC45102.2020.9294438.
- [14] K. Esterle, L. Gressenbuch, und A. Knoll, "Formalizing Traffic Rules for Machine Interpretability", in *2020 IEEE 3rd Connected and Automated Vehicles Symposium (CAVS)*, Victoria, BC, Canada, Nov. 2020 - Dec. 2020, pp. 1-7.
- [15] S. Maierhofer, A.-K. Rettinger, E. C. Mayer, und M. Althoff, "Formalization of Interstate Traffic Rules in Temporal Logic", in *2020 IEEE Intelligent Vehicles Symposium*, pp. 752-759.
- [16] S. Shalev-Shwartz, S. Shammah, und A. Shashua, "On a Formal Model of Safe and Scalable Self-driving Cars," Aug. 2017. [Online]. Verfügbar: <http://arxiv.org/pdf/1708.06374v6.pdf>.

- [17] T. Stolte, G. Bagschik, A. Reschka und M. Maurer, "Hazard analysis and risk assessment for an automated unmanned protective vehicle", in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1848-1855, doi: 10.1109/IVS.2017.7995974.
- [18] C. Pek, S. Manzinger, M. Koschi, und M. Althoff, "Using online verification to prevent autonomous vehicles from causing accidents", in *Nat Mach Intell*, vol. 2, no. 9, pp. 518-528, 2020, doi: 10.1038/s42256-020-0225-y.
- [19] F. Poggenhans, "Generierung hochdetaillierter Karten für das automatisierte Fahren", Dissertation, KIT Karlsruhe Institut für Technologie, Institut für Mess- und Regelungstechnik, 2019.
- [20] S. Maierhofer, M. Klischat und M. Althoff, "CommonRoad Scenario Designer: An Open-Source Toolbox for Map Conversion and Scenario Creation for Autonomous Vehicles," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 3176-3182, doi: 10.1109/ITSC48978.2021.9564885

Manöverbasierte Lenkgesten für das teil- und hoch-automatisierte Fahren: Ergebnisse und Ausblick

Frank Flemisch¹, Nicolas Herzberger² und Marcel Usai³

Zusammenfassung: Trotz großen technologischen Fortschritts im Bereich des hochautomatisierten Fahrens gibt es Forschungsbedarf insbesondere zu einer intuitiven Interaktion. Das von Reiter und Pferd inspirierte Projekt Vorreiter widmet sich diesem Problem, indem es intuitive Lenkgesten an einem berührungssempfindlichen Lenkrad ermöglicht, die von der Automatisierung ausgeführte Manöver einleiten. Die Gesten folgen einem universellen Designansatz (Design for All / Universal Design), der alle Fahrerinnen und Fahrer, auch Fahranfänger und Fahrerinnen und Fahrer mit Behinderungen unterstützt. Der Beitrag konzentriert sich auf die Gesamtevaluation des Konzepts in einem Fahrsimulator und präsentiert neue Daten, insbesondere zum Vergleich von Dreh-/Drück- und Streichgesten.

Schlüsselwörter: Automobile User Interfaces, Fahrzeugautomation, Gestenbasierte Interaktion, Interaktionskonzepte

1 Allgemeines Design: Von der H-Metapher zu Lenkgesten

Ausgehend von den drei Ebenen einer Fahraufgabe (Stabilisierung, Bahnführung und Navigation [1]) beschreiben Flemisch et al. [2], wie die Fahrzeugsteuerung zwischen Fahrerin bzw. Fahrer und einer kooperativen Automatisierung aufgeteilt werden kann. Winner et al. [3] beschreiben die Idee des manöverbasierten Fahrens, um die Ausführung der Fahrzeugstabilisierung von der Fahrerin auf ein Automatisierungssystem zu verlagern und der Fahrerin gleichzeitig Zugriff auf die Manöverkontrolle dieses Systems zu gewähren. Diese Konzepte der Bahnhörführungsebenen können um die kooperative Ebene erweitert werden [4]. Mit der zunehmenden Zusammenarbeit zwischen Fahrerin und Automatisierungssystem wächst der Bedarf an alternativen Steuerungen. Ein Teil des Stands der Technik wurde von einer biologisch inspirierten Design-Metapher beeinflusst, die das automatisierte Fahren mit

¹ Frank Flemisch ist Leiter des Lehr- und Forschungsgebietes Systemergonomie am Institut für Arbeitswissenschaft der RWTH Aachen University, Eifelschornsteinstr. 18, 52062 Aachen (f.flemisch@iaw.rwth-aachen.de), und Abteilungsleiter für balanced Human System Integration am Fraunhofer FKIE Wachtberg bei Bonn (frank.flemisch@fkie.fraunhofer.de)

² Nicolas Herzberger ist Gruppenleiter Systemergonomie am Institut für Arbeitswissenschaft der RWTH Aachen University, Eifelschornsteinstr. 18, 52062 Aachen (n.herzberger@iaw.rwth-aachen.de)

³ Marcel Usai ist wissenschaftlicher Mitarbeiter des Lehr- und Forschungsgebietes Systemergonomie am Institut für Arbeitswissenschaft der RWTH Aachen University, Eifelschornsteinstr. 18, 52062 Aachen (m.usai@iaw.rwth-aachen.de)

der Beziehung zwischen Reiter und Pferd vergleicht. Diese H(orse)-Metapher wurde ursprünglich von [5] beschrieben und in eine multimodale Bedienweise für teil- und hochautomatisierte Fahrzeuge umgesetzt [6].

Ein von der H-Metapher beschriebener Schlüssel zur Kontrollierbarkeit von teil- und hochautomatisierten Fahrzeugen ist der fluide Wechsel der Kontrollverteilung, aus der die späteren Levels of Vehicle Automation der BASt [7] und SAE [8] wurden. Ein weiterer Schlüssel zur Kontrolle von Pferden besteht darin, dass die **kontinuierliche Interaktion**, z. B. durch Halten eines Zügels, durch **diskrete Interaktion**, z. B. an den Steigbügeln oder Zügen ergänzt wird, um komplexe Verhaltenssequenzen zu starten, zu modifizieren oder zu stoppen. In Fahrzeugen werden diese Sequenzen als Manöver bezeichnet und können durch eine ähnliche, diskrete Interaktion mit der Hand oder dem Fuß eingeleitet werden, kombiniert mit anderen visuellen oder akustischen Interaktionen: Eine Kombination, die wir Manövergesten nennen (Schema siehe Abbildung 1, Umsetzung Abbildung 4). Es wurde deutlich, dass die Interaktion mit größerer Wahrscheinlichkeit erfolgreich sein wird, wenn:

1. Das Co-System/Automatisierung mögliche Manöver anbietet, z.B. indem es diese auf einem Head-Down- oder Head-Up-Display anzeigt,
2. die Fahrerin eines dieser Manöver mit einer Lenkgeste einleitet und dann
3. die Automatisierung das Manöver ausführt, wobei
4. das Manöver von der Fahrerin durch haptische Interaktion gespürt, beeinflusst oder unterbrochen werden kann.

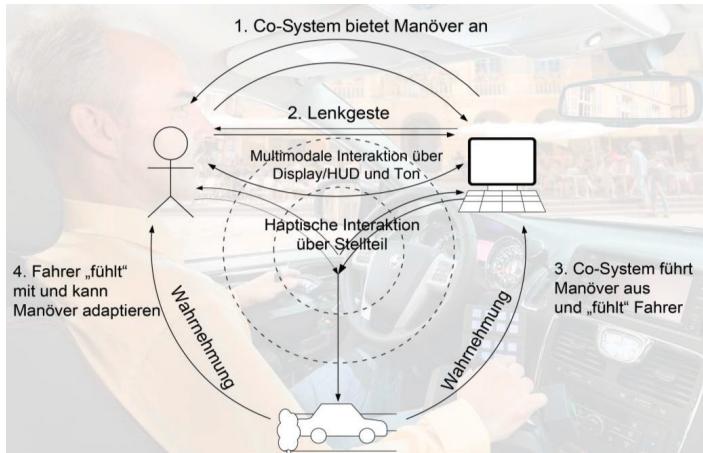


Abbildung 1: Interaktionsmuster „Manövergesten“: Gemeinsame und kooperative Kontrolle in einem Mensch-Automatisierungssystem (erweitert aus [9]).

Die hier beschriebene Studie sowie die Erarbeitung der Lenkgesten wurden im Rahmen eines BMBF-Projektes „Vorreiter“ durchgeführt. Ein detaillierter Überblick über das gesamte Projekt Vorreiter findet sich in [10]. Die abschließende Auswertung sowie die Ergebnisse werden hier jedoch erstmalig deutschsprachig veröffentlicht. Im Folgenden werden die abschließende Evaluation und die wichtigsten Ergebnisse skizziert.

2 Implementierung der Manövergesten

Basierend auf erarbeiteten Use Cases für manöverbasiertes Fahren wurden zunächst die elementarsten Manöver in einem Katalog zusammengefasst. Abbildung 2 links zeigt eine Überlagerung aller Möglichkeiten. Es kristallisieren sich vier Hauptrichtungen (nach rechts, nach links, nach vorne, sowie nach hinten) heraus, welchen situationsabhängig das passende Manöver zugeordnet werden kann. Somit kann über eine abstrakte Geste an oder auf einem Stellteil das passende Manöver ausgewählt werden. Diese Reduzierung der Eingabemöglichkeiten soll die Nutzung erleichtern und Fehleranfälligkeit reduzieren. Die Gesten folgen der Idee einer pars pro toto-Bewegung des Fahrzeugs (siehe Abbildung 2 Mitte). Aufbauend auf diesem abstrakten Gestenset wurden zwei Gestensets für die Anwendung an Lenkrad (Streichgesten) bzw. Lenkrad und Pedale (Dreh-/Drückgesten) konzipiert und in den IAW-Fahrtsimulator implementiert (siehe Abbildung 2 rechts).

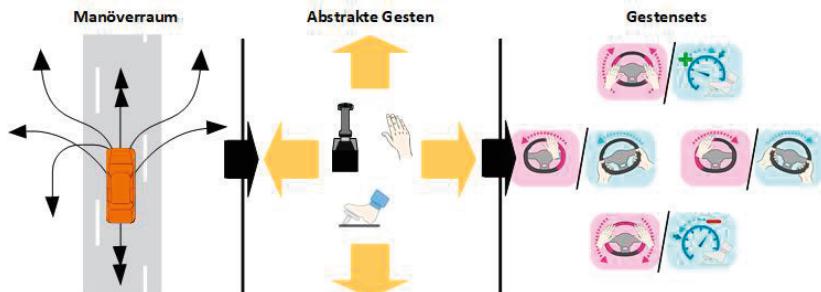


Abbildung 2: Darstellung möglicher ausführbarer Manöver (links); Mögliche Richtungen zur Ausführung von Gesten auf einem Stellteil (Mitte); abgeleitete Gestensets: Dreh-/Drückgesten (Blau) an Lenkrad und Pedale, sowie Streichgesten (Magenta) auf dem Lenkrad (rechts). [10]

Als Dreh-/Drückgesten werden kleine Impulse an Lenkrad oder Pedale erkannt. Für die Erkennung von Streichgesten wurde von der Valeo Schalter und Sensoren GmbH ein berührungssensitives Lenkrad aufgebaut, auf dem der Lenkkranz in zehn Zonen plus eine Zone auf der Rückseite aufgeteilt ist. Die Gesten werden durch Überstreichen von mehreren dieser Zonen in die entsprechende Richtung (nach rechts-links) mit einer Hand bzw. mit zwei Händen (nach vorne/hinten) ausgelöst (siehe Abbildung 3). Um gegen ungewollte Auslösungen abzusichern und gleichzeitig das Systemverhalten der Fahrzeugautomation transparent gestalten zu können, wurden drei Stufen der Erkennung eingeführt: armed (Geste ist angedeutet), interlocked (Geste ist erkannt) und executed (Geste ist erkannt

und bestätigt). Eine Geste kann jederzeit abgebrochen werden oder durch Eingabe einer Geste in die Gegenrichtung eine anschließende Ausführung rückgängig gemacht werden.

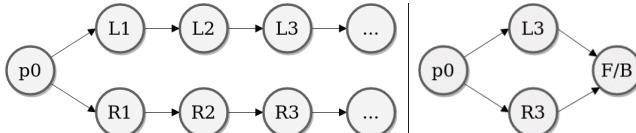


Abbildung 3: Zustandsautomat zur Erkennung von Streichgesten nach rechts/links (links); sowie zur Erkennung von Streichgesten nach vorne/hinten (rechts in Kombination mit links). [11]

Die Gestensets wurden für SAE Level 2 und 3/4 [10] implementiert. Abbildung 4 zeigt ein Foto während der Ausführung einer Streichgeste nach links im SAE Level 2. Aufgrund des doch fundamentalen Unterschieds in der Rolle der Fahrerin im entsprechenden automatisierten Fahrmodus unterscheiden sich hier auch die Verhalten der Automationssysteme, sobald eine Geste erkannt wurde: Im SAE Level 2 prüft ein Mediations-Untersystem [12] zunächst kontextabhängig, welches Manöver zur erkannten Geste passt und anschließend, ob das Manöver Teil der vorgeschlagenen, d.h. von der Automation ausführbaren Manöver ist. Ist dies der Fall, so wird das Manöver ausgeführt. Andernfalls erfolgt eine Rückmeldung über die Nichtverfügbarkeit des Manövers. Im SAE Level 3/4 muss die Automation zusätzlich noch prüfen, inwieweit die Ausführung des Manövers innerhalb der aktuellen Verkehrssituation sicher und legal ist. Erst dann wird ein Manöver ausgeführt. Die Fahrerin hat zwar im SAE Level 3/4 keine Autorität oder Kontrolle über die Kontrollebene der Fahraufgabe, kann jedoch auf der Führungsebene Einfluss nehmen, indem sie Vorschläge für Manöver macht, ohne bereit für eine Übernahme sein zu müssen. Sie kann jederzeit auch den Automationsmodus wechseln und damit die direkte Kontrolle übernehmen.



Abbildung 4: Manöverbasierte Interaktion für Vorreiter im IAW-Fahrtsimulator. Das Vorreiter-Lenkrad von Valeo gibt während einer Gesteneingabe visuelles Feedback über die überstrichenen Zonen. Im HUD ist eine Geschwindigkeitsanzeige sowie die Rückmeldung über eine begonnene Gestenerkennung.

3 Gesamtevaluation im Fahrsimulator

3.1 Methodik

Die Studie wurde im statischen Fahrsimulator des IAW der RWTH Aachen University (siehe Abbildung 4) durchgeführt. Als Fahrsimulationsssoftware wurde SILAB 6.0 verwendet. Die Fahrzeugsteuerung erfolgte mittels des kapazitiven Lenkrads von Valeo sowie mittels der vom IAW entwickelten Gestenerkennungssoftware.

Um alle zuvor entwickelten Gesten evaluieren zu können, wurden acht Fahrszenarien implementiert, die diese Eingaben erforderten: Landstraße mit anschließender T-Kreuzung, Autobahnauffahrt, Autobahn, Baustelle, Autobahnausfahrt, erneut Landstraße mit T-Kreuzung und abschließend ein Parkplatz. Dabei wurden fünf unterschiedliche Bedingungen in einem 2x2+1 Design verglichen: Manuelle Fahrt (SAE Level 0, Baseline) sowie Dreh-/Drückgesten versus Streichgesten, jeweils in SAE Level 2 als auch in SAE Level 3/4. Die Probanden wurden instruiert, den Verkehrsschildern zu einem Point-of-Interest zu folgen, was Eingriffe in die Fahraufgabe erforderte, da die automatisierte Fahrfunktion auf der kürzesten Strecke zum Zielpunkt fuhr und der Point-of-Interest einen Umweg bedeutete. Die einzelnen Fahrten dauerten jeweils ca. 4 Minuten.

An der Studie nahmen $N = 26$ Versuchspersonen teil (26,9 % weiblich, 73,1 % männlich). Voraussetzung für die Teilnahme war ein gültiger Führerschein sowie keine unkorrigierten Sehbeeinträchtigungen. Das Alter der Teilnehmenden lag zwischen 19 und 64 Jahren ($M = 28,96$ Jahre, $SD = 13,25$ Jahre)

Jeder Versuchsdurchlauf begann mit einer Einführungsfahrt in SAE Level 0, in der sich die Teilnehmenden mit der Steuerung des Fahrsimulators vertraut machen konnten. Anschließend folgten die fünf Testblöcke. Diese begannen stets mit einem Naive Run (ohne vorherige Erläuterung der jeweiligen Gestensteuerung), auf den ein kurzer Fragebogen folgte. Anschließend wurde den Teilnehmenden das jeweilige Konzept sowie die Bedienung detailliert erläutert und etwaige Fragen beantwortet. Darauf folgte ein Trained Run⁴ sowie ein dazugehöriger Fragebogen. Die Naive Runs wurden durchgeführt, um sowohl die intuitive Verständlichkeit als auch die Erlernbarkeit zu überprüfen. Nach Abschluss der Testblöcke folgte ein finaler Fragebogen. Jeder Durchlauf dauerte ca. 4 Stunden und die Versuchspersonen erhielten eine Probandenvergütung.

3.2 Ergebnisse

Um einen Überblick über die Leistungsfähigkeit der implementierten Gesten zu erhalten, wurde die Erkennungsrate der einzelnen Gestenaktivierungen ausgewertet (siehe Abbildung 5).

⁴ Ein sogenannter „Trained Run“ ist ein Testlauf nach einer Einweisung oder ausreichendem Training der Teilnehmenden. Im vorliegenden Fall erhielten die Teilnehmenden vor dem Trained Run eine Einweisung in den Umgang mit den Gesten.

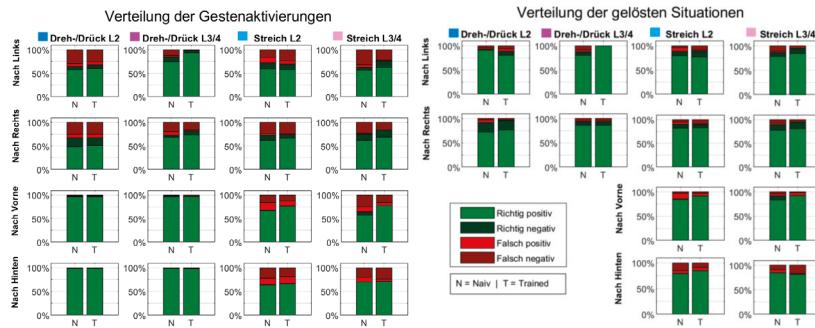


Abbildung 5: Verteilung von Gesteaktivierungen und Situationslösungen. Aufgezeichnet wurde die Reaktion der Ko-Automation auf menschliche Eingaben und ihre beobachtete Klassifizierung (basierend auf [11]).

Auf der linken Seite von Abbildung 5 ist die Verteilung der Gesteaktivierungen dargestellt. Alle Versuche der Teilnehmenden, eine Geste zu aktivieren, wurden gezählt und es wurde nach der Signalerkennungstheorie (Signal Detection Theorie, z.B. [13]) nicht nur ausgewertet, ob die Gesten erfolgreich erkannt und auch aktiviert wurden (richtig positiv), oder ob sie erkannt oder nicht erkannt, aber zu Recht nicht umgesetzt wurden, weil sie von der Automatisierung als unsicher eingestuft wurden (richtig negativ). Ebenso wurden Gesteneingaben erfasst, die erkannt, aber offensichtlich von der Fahrerin nicht beabsichtigt bzw. von ihr andere Manöver beabsichtigt waren (falsch positiv), oder nicht erkannt oder abgelehnt wurden, obwohl das entsprechende Manöver möglich gewesen wäre (falsch negativ).

Die Daten zeigen, dass die Mehrheit der Reaktionen auf Gesteneingaben richtig positiv oder richtig negativ waren. Die richtig negativ-Reaktionen lagen bei dieser fragilen Risikobilanz zwischen falsch positiv und falsch negativ auf der sicheren Seite. Eine Ausnahme bildeten die lateralen Streichgesten in Level 2, die eine relativ hohe Rate an falsch positiv Antworten aufwiesen. Hier hatten Teilnehmende Schwierigkeiten mit der Doppelbelegung des Lenkrads, das sowohl als Eingabegerät für die Fahrzeugstabilisierung als auch als Gesteneingabegerät diente.

Abbildung 5 rechts zeigt die Auswertung, ob die Fahrsituation insgesamt gelöst wurde, unabhängig davon, wie viele Versuche einzelner Lenkgesten unternommen wurden. Eine Situation wurde immer dann beendet, wenn eine gegebene Gelegenheit, z.B. zum Abbiegen, verstrichen ist, das gewünschte Manöver ausgeführt wurde oder die Fahrerin das Interesse an der Einleitung eines Manövers verlor. Hier zeigen die Daten, dass mehr als 75 % aller Situationen in jedem Fall so gelöst wurden, wie sie hätten gelöst werden müssen.

Abbildung 6 stellt einen Auszug aus der abschließenden subjektiven Bewertung dar. Rechts sind die Ergebnisse der NASA TLX-Bewertungen dargestellt. Dabei zeigt sich,

dass die getesteten Gestenkonzepte sowohl hinsichtlich mentaler und körperlicher Beanspruchung als auch hinsichtlich Anstrengung positiver bewertet wurden als die Baseline. Für die Faktoren zeitliche Beanspruchung sowie Frustrationsniveau zeigt sich jedoch, dass die Implementierung der SAE Level 2 Streichgesten noch Entwicklungspotential bietet – hier interferierte die Hands-On-Pflicht mit den intendierten Gesteneingaben, was bisweilen zu nicht erkannten oder fälschlich interpretierten Eingaben führte. Für die Gesamtbewertung lässt sich hervorheben, dass zwei der Steuerungskonzepte (Dreh-/Drückgesten und Streichgesten in Level 3/4) von den Versuchspersonen besser als die Baseline bewertet wurden. Der rechte Teil von Abbildung 6 zeigt oben die Ergebnisse der System Usability Scale (SUS) sowie unten eine Legende. Hier zeigt sich, dass die Baseline gut ($M = 81.6$), die Dreh-/Drückgesten in Level 2 ok ($M = 70.0$) und in Level 3/4 ebenfalls gut ($M = 79.2$) bewertet wurden. Die Streichgesten wurden in Level 2 schlecht ($M = 51.6$) und in Level 3/4 gut ($M = 76.8$) bewertet.

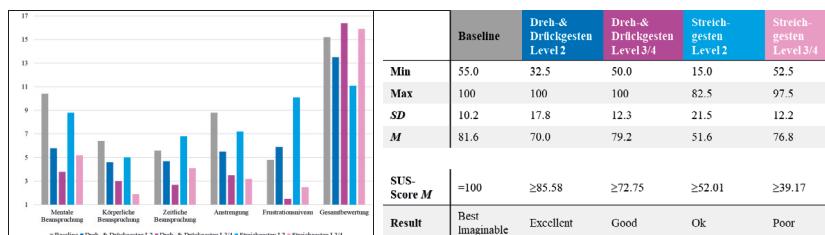


Abbildung 6 Ergebnisse der subjektiven Bewertungen: Links: Ergebnisse des NASA TLX Rechts: Ergebnisse der SUS.

Neben den Bewertungen mittels NASA TLX und SUS hatten die Teilnehmenden im Rahmen der abschließenden Gesamtbewertung die Gelegenheit anzugeben, welches der Systeme von ihnen bevorzugt wurde. Dabei waren Mehrfachnennungen möglich. Die Ergebnisse dieser Bewertungen sind in Abbildung 7 dargestellt. Es zeigte sich, dass in jedem der Automatisierungsgrade mehr Teilnehmende Dreh-/Drückgesten (Level 4: 16, Level 2: 2 Nennungen) als Streichgesten (Level 4: 8, Level 2: 1 Nennungen) bevorzugten. Die Baseline (manuelles Fahren) erhielt 10 Nennungen.

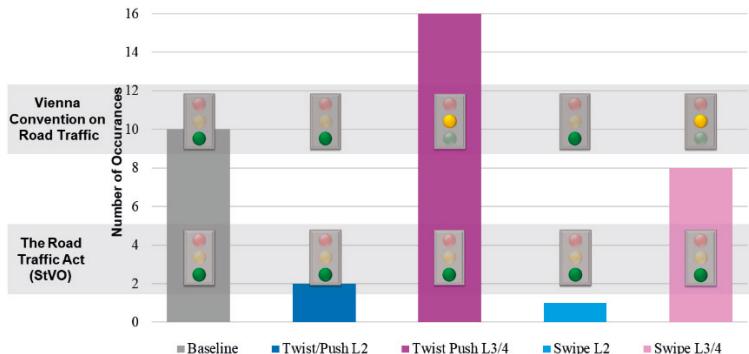


Abbildung 7. Subjektive Präferenzen und Einschätzungen zur Zulassbarkeit der Kontrollgesten.

In der Abbildung sind zusätzlich die Einschätzungen zur Zulassbarkeit der Gestenkonzepte nach Straßenverkehrsordnung (StVO) bzw. Vienna Convention on Road Traffic dargestellt. Hierzu wurden die Gestensteuerungskonzepte im Rahmen des Vorreiter-Projekts durch die Hochschule für Wirtschaft und Recht (HWR) in Berlin evaluiert.

3.3 Diskussion

Die Ergebnisse deuten darauf hin, dass Gestensteuerungskonzepte und manöverbasierte Systeme in Zukunft eine echte Alternative darstellen könnten. Dennoch zeigen die implementierten SAE-Level-2-Systeme Schwierigkeiten: Eine Herausforderung war die Tatsache, dass mindestens eine Hand am Lenkrad vorgegeben war (Hands-on). Dies führte bisweilen zu einer Vermischung der Manövergesten und Stabilisierungsaufgaben. Die subjektiven Präferenzen zeigen auch, dass die Fahrer entweder manuelles oder hochautomatisiertes Fahren bevorzugten. Die Dreh-/Drückgesten in Kombination mit Level 3/4 wurden sogar gegenüber der Basisvariante bevorzugt. Diese Ergebnisse sind einerseits recht vielversprechend, andererseits wurden noch Verbesserungs- und Verfeinerungsmöglichkeiten bei der Benutzerfreundlichkeit identifiziert. Darüber hinaus sollte erwähnt werden, dass die vorliegende Studie ein Test in einem statischen Simulator war, mit der dynamische Rückkopplungseffekte nicht simuliert werden konnten. Hier könnte eine Studie im realen Verkehr wichtige zusätzliche Ergebnisse liefern. Zu diesem Zweck wurde im Rahmen von Vorreiter ein Testfahrzeug konzipiert und aufgebaut. Dieses Fahrzeug kann als Wizard-of-Oz System [10] eingesetzt werden, bei dem die Automatisierung von einer menschlichen Fahrerin hinter einem Halbspiegel nachgebildet wird.

4 Fazit und Ausblick

Ausgangspunkt des BMBF-Projektes Vorreiter waren große technische Fortschritte im teil- und hochautomatisierten Fahren, die gleichzeitig wieder Herausforderungen für eine möglichst intuitive Einbeziehung des Menschen bedeuten. Die Designmetapher „Reiter-Pferd“ liefert, wie bereits vorher zu Automationsgraden und Transitionen, auch hierzu wertvolle Inspiration hin zu Manövergesten. Manövergesten wurden insgesamt im Vergleich zum manuellen Fahren sehr gut akzeptiert. In einem direkten Vergleich von Dreh-Drück und Streichgesten sind die noch etwas durchwachsenen Ergebnisse der Streichgesten für uns Hinweise darauf, dass weniger das Konzept als vielmehr die hier getestete Implementierung der Streichgesten noch aufwändiger und fehleranfälliger ist als für die Dreh-Drück-Gesten. Das muss nicht immer so bleiben: Unsere Versuche ergaben wertvolle Hinweise darauf, wie man die für eine neuartige und für viele Nutzer faszinierende Bedienweise bereits ermutigende Annahmequote weiter steigern kann. Streichen oder Drehen/Drücken auf einem Lenkrad oder einem alternativen Stellteil sind vielversprechende Kandidaten für eine solche Manöver-Interaktion, aber nicht die einzigen: Eine Kombination mit Blicken, Handgesten und/oder Sprachkommandos könnte eine sinnvolle Erweiterung sein, nicht nur für Fahrerinnen und Fahrer mit Behinderungen. Das frühe Einbeziehen von Fahrerinnen und Fahrenden mit und ohne Behinderungen empfanden die Teilnehmenden und wir als sehr inspirierend, und empfehlen das uneingeschränkt für zukünftige Forschungsprojekte. Teil- und Hochautomatisierung, insbesondere als kooperative Fahrzeugführung ist nun nach von der H-Metapher ausgehenden 20 Jahren Forschung vieler Kolleginnen und Kollegen an einem Punkt, an dem einerseits erste Systeme bereits in Serie verfügbar sind, z.B. als SAE Level 3 Staupilot, und andererseits noch entscheidende Aspekte wie z.B. Migrationsfähigkeit in einem heterogenen Verkehrssystem der Zukunft noch besser erforscht werden müssen, um das Potential dieser faszinierenden Technologie auch für alle Menschen nutzbar und sicher zu gestalten.

Literatur

- [1] Donges, E. (1982). Aspekte der aktiven Sicherheit bei der Führung von Personenkraftwagen. *Automob-Ind*, 27(2).
- [2] Flemisch, F. O., Kelsch, J., Schieben, A., & Schindler, J. (2006). Stücke des Puzzles hochautomatisiertes Fahren: H-Metapher und H-Mode. Zwischenbericht 2006.
- [3] Winner, H., Heuss, O., & Landau, K. (2005, March). X-by-wire Betätigungslemente-Überblick und Ausblick. In Darmstädter Kolloquium Mensch und Fahrzeug. Cockpits für Straßenfahrzeuge der Zukunft (pp. 79-115). Stuttgart: Ergonomia.
- [4] Flemisch, F., Abbink, D. A., Itoh, M., Pacaux-Lemoine, M. P., & Weßel, G. (2019). Joining the blunt and the pointy end of the spear: towards a common framework of joint action, human–machine cooperation, cooperative guidance and control, shared, traded and supervisory control. *Cognition, Technology & Work*, 21(4), 555-568.

- [5] Flemisch, F. O., Adams, C. A., Conway, S. R., Goodrich, K. H., Palmer, M. T., & Schutte, P. C. (2003). The H-Metaphor as a guideline for vehicle automation and interaction.
- [6] Altendorf, E.; Baltzer, M.; Heesen, M.; Kienle, M.; Weißgerber, T.; Flemisch, F.: H-Mode, a Haptic-Multimodal Interaction Concept for Cooperative Guidance and Control of Partially and Highly Automated Vehicles; Winner et. al. Eds.: Handbook of Driver Assistance Systems; Springer, 2015.
- [7] Gasser, T. M., Arzt, C., Ayoubi, M., Bartels, A., Bürkle, L., Eier, J., Flemisch, F., Häcker, D., Hesse, T., Huber, W., Lotz, C., Maurer, M., Ruth-Schumacher, S., Schwarz, J., und Vogt, W. (2012). Rechtsfolgen zunehmender Fahrzeugautomatisierung - Gemeinsamer Schlussbericht der Projektgruppe. Fahrzeugtechnik F 83, Bundesanstalt für Straßenwesen (bast).
- [8] SAE Levels of Driving Automation; SAE International Standard J3016, 2018.
- [9] Donges, E.: Supporting Drivers by Chassis Control Systems. Seminar Smart Vehicles, TNO Delft NL, 13 - 16. In: Pauwelussen, J. P., and Pacejka, H. B. (Ed): Smart Vehicles. Swets & Zeitlinger B. V., Lisse, the Netherlands, 1995 (ISBN 90 265 1456 5), February 1995.
- [10] Frank Flemisch, Frederik Diederichs, Ronald Meyer, Nicolas Herzberger, Ralph Baier, Eugen Altendorf, Julia Spies, Marcel Usai, Vera Kaim, Bernhard Doentgen, Anja Valeria Bopp-Bertenbreiter, Harald Widlroither, Simone Ruth-Schumacher, Clemens Arzt, Evin Bozbayir, Sven Bischoff, Daniel Diers, Reto Wechner, Anna Sommer, Emre Aydin, Verena Kaschub, Tobias Kiefer, Katharina Hottelart, Patrice Reilhac, Gina Weßel, and Frank Kaiser: Vorreiter, Manoeuvre-Based Steering Gestures for Partially and Highly Automated Driving; In Meixner, G.: Smart Automotive Mobility; Springer 2020.
- [11] Usai, M., Meyer, R., Baier, R., Herzberger, N., Lebold, K., & Flemisch, F. (2021, February). System Architecture for Gesture Control of Maneuvers in Automated Driving. In International Conference on Intelligent Human Systems Integration (pp. 65-71). Springer, Cham.
- [12] Baltzer, M., Flemisch, F., Altendorf, E., Meier, S. (2014) Mediating the interaction between human and automation during the arbitration processes in cooperative guidance and control of highly automated vehicles. In: Ahram, T., Karwowski, W., Marek, T. (eds) Proceedings of the 5th international conference on applied human factors and ergonomics AHFE 2014, Krakow, July 2014
- [13] Wickens, C. D., Helton, W. S., Hollands, J. G., & Banbury, S. (2021). Engineering psychology and human performance. Routledge.

Longitudinal Acceleration during Lane Changes - A Human-Centered Investigation for Automated Driving

Johannes Ossig* Stephanie Cramer[†] Klaus Bengler*

Abstract:

Lane changes represent central driving maneuvers on highways and are frequently linked to acceleration maneuvers. For automated driving, previous studies have addressed the issue of appropriate longitudinal accelerations for vehicle occupants. However, these investigations only considered pure longitudinal acceleration maneuvers and have neglected potential influence of lane changes on driving experience. For this reason, this paper presents an evaluation of longitudinal accelerations during non-automated and automated lane changes and compares the results with previous studies. Based on this, the usefulness of further human-centered research on longitudinal accelerations during automated lane changes is discussed and recommendations for a future study are proposed.

Keywords: Automated Driving, Human Factors, Lane Changes, Vehicle Dynamics

1 Introduction

Appropriately designed automated driving styles contribute to increase driving comfort [1] and the general acceptance of automated driving [2]. Besides the tactical decision-making process and question of when which driving maneuver is performed automatically, the operational performance of these maneuvers plays an important role. In particular, longitudinal dynamic parameters, such as longitudinal acceleration, have a strong influence on driving comfort during automated driving [3], which is why a human-centered parameterization is essential in this case. According to a summary of studies on non-automated and automated driving [4], longitudinal accelerations up to 0.9 m/s^2 are considered *cautious*, up to 1.47 m/s^2 are considered *normal*, and up to 3.07 m/s^2 are considered *dynamic*. Further studies on automated driving indicate that longitudinal acceleration profiles should be designed symmetrically [5] and that a longitudinal acceleration of 1.5 m/s^2 should not be exceeded in low-speed zones [6]. In comparison, a relevant standard [7] defines a maximum longitudinal acceleration of 2.0 m/s^2 for adaptive cruise control (ACC) systems and thus, according to the Society of Automotive Engineers (SAE) [8], for the first level of automated driving.

In the aforementioned literature and in consideration of further studies, acceleration maneuvers are mostly investigated independent of lane changes. However, especially

*Chair of Ergonomics, Technical University of Munich, 85748 Garching (e-mail addresses: johannes.ossig@tum.de, bengler@tum.de).

[†]Predevelopment of Automated Driving Functions, AUDI AG, 85045 Ingolstadt (e-mail address: stephanie.cramer@audi.de).

on highways, lane changes represent main driving maneuvers, which are characterized by interaction with other vehicles [9], a relatively high safety risk [10], a combination of longitudinal and lateral dynamics [11], and resulting quiet complex driving situations [11, 12]. For these reasons, acceleration maneuvers during lane changes are not necessarily comparable to ordinary acceleration maneuvers without lane changes. Therefore, the research question of how longitudinal acceleration during automated lane changes should be designed cannot be answered in detail at the current time. However, it is known that an acceleration maneuver starting shortly before the actual lane change can contribute to the predictability of automated passing maneuvers and the automation system's behavior [13].

In this paper, we investigate on the basis of recorded non-automated and automated lane changes the longitudinal acceleration during passing maneuvers on German highways. Based on this, appropriate maximum longitudinal accelerations during automated lane changes are estimated and recommendations for future human-centered studies are derived.

2 Methodology

This paper focuses on non-automated and automated so-called *tactical lane changes* to the left, which serve to increase speed and to initiate passing maneuvers [14]. Accordingly, the following investigations only take into account lane changes with a lower speed at the beginning of the driving maneuver than the maximum permitted speed in the respective highway section and, if available, than the intended target speed of the automation system. For the estimation of a comfortable longitudinal dynamic and, more specifically, an appropriate maximum longitudinal acceleration during automated lane changes, a three-step methodology was defined:

1. Automated lane changes that were performed under real-world conditions in our last study are evaluated and respective assessments of participants on the longitudinal dynamic are summarized. Since the overarching objective of this study was to identify appropriate moments for automated lane changes [15, 16], the number of assessments of the longitudinal dynamic is severely limited, despite more than 750 automatically performed lane changes to the left.
2. Due to the limited number of available assessments of the longitudinal dynamic during automated lane changes, we also investigate non-automated lane changes using the highD dataset [17]. Similar to the previous evaluation of automated lane changes, the focus lies in this second step on the maximum longitudinal acceleration during passing maneuvers and, moreover, on the potential influence of the traffic scenario.
3. Finally and in consideration of relevant literature, the conducted evaluations of non-automated and automated lane changes are discussed and recommendations for further human-centered research on longitudinal dynamics during automated lane changes are given.

3 Evaluation of Automated Lane Changes

This data evaluation is based on our on-road vehicle study ($N = 60$) focusing appropriate moments for conditionally automated lane changes on highways, which was briefly described in a short paper [15]. Since the main results of this study about appropriate moments for automated lane changes [16] are not relevant for the research issue stated in Section 1, the following explanations concentrate on the test vehicle's behavior planning and the resulting longitudinal dynamic.

The maximum longitudinal acceleration profile was defined on the basis of previous research [13] and is depicted in a simplified form in Fig. 1. Each automated acceleration maneuver was determined individually depending on, among other things, this profile and the difference between the initial speed at the beginning of the lane change and the target speed of the automation system. The target speed of the automation system corresponded to 120 km/h or, if available, to the speed of a slower preceding vehicle. This approach for determining the longitudinal acceleration by means of the speed difference is included in several traffic models [18] and did not distinguish between acceleration maneuvers during lane changes and single-lane acceleration maneuvers in this study.

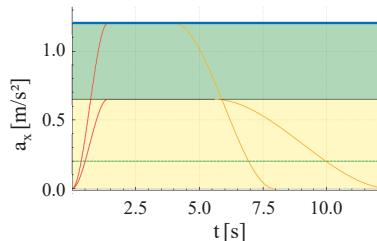


Figure 1: The maximum longitudinal acceleration profile was characterized in our last study by the parameters $a_{x,\max} = 1.2 \text{ m/s}^2$ (blue line) and $j_{x,\max} = 1.3 \text{ m/s}^3$ (upper red line).

Since the aim of the aforementioned study was the identification of appropriate moments for automated lane changes [15, 16], no separate questionnaires about the longitudinal dynamic during passing maneuvers were used. However, the participants (P) had the opportunity to add qualitative comments on the assessment of their general sense of well-being and perceived discomfort, which can be used to investigate the suitability of the longitudinal dynamic. Several participants described the experienced longitudinal dynamic during lane changes qualitatively as too low (e.g. P5, P11, P19, P28, P39) and blocking rearward traffic in the target lane was mentioned several times as main reason for the perceived discomfort (e.g. P5, P19, P28, P39). For example, one participant (P19) described that such moderate lane changes would slow down rearward traffic and another participant (P39) suggested that rearward traffic would even expect a faster acceleration behavior of passing vehicles. Furthermore, another participant (P30) described that an additional benefit of higher longitudinal accelerations is the possible avoidance of safety-related aborts of automated lane changes that have already started.

In summary, various participants noticed the low longitudinal acceleration during automated lane changes in this study in a negative way and considered it uncooperative for rearward vehicles in the target lane. This influence of rearward traffic is examined in more detail in the following evaluation of non-automated lane changes, which represents the second step of this work (see Section 2).

4 Evaluation of Non-Automated Lane Changes

In addition to evaluating automated lane changes, we examined the longitudinal acceleration during non-automated lane changes in greater detail. For this purpose, the highD dataset [17] was used, which contains traffic records of several approximately 400 m long three-lane highway sections. The evaluation took into account lane changes to the left that were carried out in 38 highway sections with a maximum permitted speed of 120 km/h. This speed value corresponds to the target speed of the automation system described in Section 3 and thus contributes to the comparability of both data evaluations. As another prerequisite for consideration in this evaluation, the lane changes have to be completely included in the dataset and in the recorded highway sections. For this reason, referring to a previous data evaluation [15], each lane change was replaced by an individual logistic function (1) and the positions of the beginning and completion of the maneuvers were subsequently estimated. In this function, x_a represents the x-coordinate at which the lane markings are crossed by the respective recorded vehicle trajectory.

$$y(x) = \frac{G \cdot e^{k(x-x_a)}}{1 + e^{k(x-x_a)}} \quad \text{with} \quad y'(x_a) = \frac{G \cdot k}{4} \quad \text{and} \quad G = 3.75 \text{ m} \quad (1)$$

The beginnings of the lane changes were defined in accordance with the previous data evaluation [15] at the x-coordinates where the respective logistic functions reach 3% of the presumed lane width. In comparison, the completions of the lane changes were defined at the x-coordinates where the respective logistic functions reach 97% of the lane width. Fig. 2 shows as an example the first twelve of approximately 700 lane changes

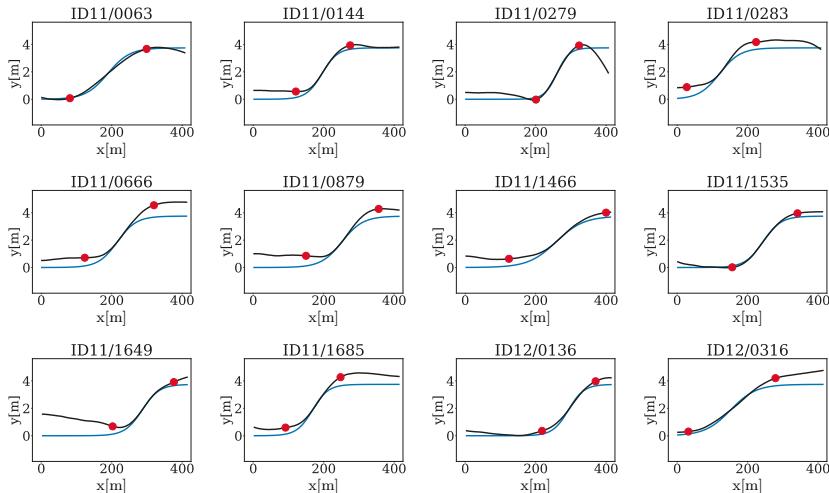


Figure 2: This figure shows the estimated positions of the beginnings and completions (red dots) of the first twelve non-automated lane changes considered in the data evaluation.

considered in this evaluation (black lines) with the logistic functions (blue lines) and the estimated positions of the beginnings and completions from a bird's eye view. Despite minor differences between recorded lane changes and the respective logistic functions, the calculated positions of the beginnings and completions are plausible and appear sufficient for our purpose.

For each lane change considered in this evaluation, the maximum longitudinal acceleration $\dot{a}_{x,\max}$ in the recorded highway section was identified and marked in relation to the difference Δv_x between the initial speed at the beginning of the lane change and the maximum permitted speed of 120 km/h in Fig. 3. In this process, a distinction was made between lane changes without and with a rearward vehicle in the target lane as well as between lane changes from the right lane to the center lane and from the center lane to the left lane. The relationship between the relative speed Δv_x and the maximum longitudinal acceleration was subsequently evaluated and with the help of linear regression functions (black lines) presented in Fig. 3. It can be seen that the longitudinal acceleration during lane changes from the center lane to the left lane are higher than the longitudinal acceleration during lane changes from the right lane to the center lane. This circumstance can be explained by the heterogeneous speed distribution on German highways [19] and the obligation to drive on the right-hand side of the road. Furthermore, it emerges that the longitudinal acceleration increases with the relative speed Δv_x and is slightly higher for lane changes with a rearward vehicle in the target lane ($\tilde{\dot{a}}_{x,\max} = 0.67 \text{ m/s}^2$) compared to lane changes without a rearward vehicle in the target lane ($\tilde{\dot{a}}_{x,\max} = 0.60 \text{ m/s}^2$).

In addition to Fig. 3, Fig. 4 shows lane changes with a rearward vehicle in the target lane including respective time-to-collision (TTC_x) values of the beginnings of the lane changes. The depicted gray surface represents the predictions resulting from a polynomial regression model. Here, an influence of the TTC_x on the maximum longitudinal acceleration during non-automated lane changes could not be identified.

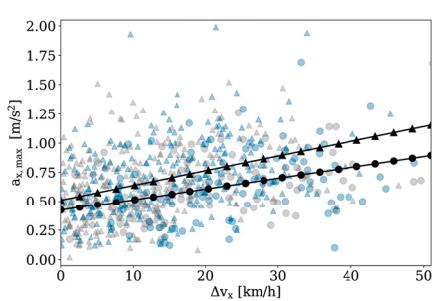


Figure 3: Maximum longitudinal accelerations during non-automated lane changes ($N = 713$) depending on the relative speed Δv_x . Dot symbols represent lane changes from the right to the center lane. Triangle symbols represent lane changes from the center lane to the left lane. Blue symbols represent lane changes with a rearward vehicle in the target lane. Gray symbols represent lane changes without a rearward vehicle in the target lane.

In summary, it can be stated that, from a descriptive point of view, the longitudinal dynamic during non-automated lane changes depends on the existence of a rearward vehicle in the target lane, the position of the lane change, and especially the relative speed Δv_x . At a high relative speed of 50 km/h, the maximum longitudinal acceleration during the driving maneuvers is approximately 1.2 m/s^2 on average. However, it should also be pointed out that there is a large scatter of maximum longitudinal accelerations

during non-automated lane changes, which indicates additional influencing factors, such as individual human driving styles or personalities, and deserves further discussion.

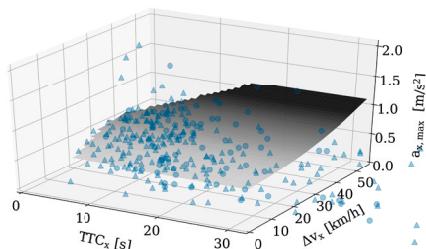


Figure 4: Maximum longitudinal accelerations during non-automated lane changes ($n_1 = 334$) with a rearward vehicle in the target lane depending on the relative speed Δv_x and TTC_x . Dot symbols represent lane changes from the right to the center lane. Triangle symbols represent lane changes from the center lane to the left lane.

5 Discussion and Conclusion

The longitudinal acceleration during automated lane changes assessed as too low in Section 3 and the longitudinal acceleration during non-automated lane changes investigated in Section 4 differ only to a minor extent ($a_{x,\max} \approx 1.2 \text{ m/s}^2$). The assumption that the longitudinal acceleration during automated lane changes was designed too low in our last study is consequently not supported by the data evaluation of non-automated driving. However, we have to emphasize that the vehicle trajectories of the highD dataset [17] are only available in highway sections of about 400 m (see Fig. 2). It is conceivable that the actual maximum longitudinal accelerations are achieved before or after the evaluated non-automated lane changes and thus do not lie within the recorded highway sections. As a consequence, the actual maximum longitudinal accelerations would exceed the values presented in Section 4. In addition, various studies (e.g. [2, 6]) indicate that automated driving styles should not necessarily correspond to human driving styles. For this reason, the high degree of agreement between the automated longitudinal dynamics and non-automated longitudinal dynamics does not mean that increasing acceleration would not optimize driving comfort for automated driving. Taking into account previous research on automated driving, the qualitative participant opinions mentioned in Section 3, and the influence of rearward vehicles in the target lane on the longitudinal acceleration during non-automated lane changes (see Section 4), a maximum longitudinal acceleration during automated lane changes higher than 1.2 m/s^2 appears for individual traffic scenarios reasonable. Such scenarios include, for example, changes to faster lanes or lane changes in front of faster road users.

For future human-centered investigations of automated lane changes and in addition to the already known driving style with a maximum longitudinal acceleration of 1.2 m/s^2 described in Section 3, we recommend a driving style with a maximum longitudinal acceleration of 2.0 m/s^2 . The latter value corresponds to the upper limit of adaptive cruise control systems [7] and is located in the lower range of a *dynamic* acceleration [4]. We hypothesize that such a dynamic acceleration during automated passing maneuvers reduce the perceived discomfort induced by blocking other traffic participants and consequently optimize the driving experience. For this reason, both mentioned driving styles were im-

plemented in a prototypical vehicle, verified in a pre-study with experts under real-world conditions and will form the basis for a future study on the human-centered design of automated lane changes. With respect to higher levels of automated driving, which allow the driver to be distracted from traffic by non-driving related tasks [8], and the resulting impact on the human perception of driving dynamics [20], non-driving related tasks will be considered in our following study.

6 Acknowledgment

This research was supported by AUDI AG.

References

- [1] T. Gasser et al., *Bericht zum Forschungsbedarf: Runder Tisch Automatisiertes Fahren – AG Forschung*. 2015.
- [2] F. Hartwich, M. Beggiato, and J. F. Krems, “Driving comfort, enjoyment and acceptance of automated driving – effects of drivers’ age and driving style familiarity,” *Ergonomics*, vol. 61, no. 8, pp. 1017–1032, 2018.
- [3] S. Scherer, A. Dettmann, F. Hartwich, T. Pech, A. C. Bullinger, and G. Wanielik, “How the driver wants to be driven - Modelling driving styles in highly automated driving,” in *Tagungsband 7. Tagung Fahrerassistenz*, 2015.
- [4] I. Bae, J. Moon, and J. Seo, “Toward a Comfortable Driving Experience for a Self-Driving Shuttle Bus,” *Electronics*, vol. 8, no. 9, p. 943, 2019.
- [5] H. Bellem, B. Thiel, M. Schrauf, and J. F. Krems, “Comfort in automated driving: An analysis of preferences for different automated driving styles and their dependence on personality traits,” *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 55, pp. 90–100, 2018.
- [6] S. Scherer, D. Schubert, A. Dettmann, F. Hartwich, and A. C. Bullinger, “Wie will der ‘Fahrer’ automatisiert gefahren werden? Überprüfung verschiedener Fahrstile hinsichtlich des Komforterlebens,” in *VDI/VW-Gemeinschaftstagung Fahrerassistenzsysteme und automatisiertes Fahren*, 2016.
- [7] ISO, “Intelligent transport systems - Adaptive cruise control systems - Performance requirements and test procedures, 15622:2010,” 2010.
- [8] SAE, “Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, J3016_201806,” 2018.
- [9] C. Oh, J.-h. Choi, and S. Park, “In-depth understanding of lane changing interactions for in-vehicle driving assistance systems,” *International Journal of Automotive Technology*, vol. 18, no. 2, pp. 357–363, 2017.

- [10] A. Bartels, S. Steinmeyer, S. Brosig, and C. Spichalsky, “Fahrstreifenwechselassistenz,” in *Handbuch Fahrerassistenzsysteme* (H. Winner, S. Hakuli, and G. Wolf, eds.), pp. 562–571, Wiesbaden, Germany: Vieweg+Teubner Verlag, 2009.
- [11] K. Sun, X. Zhao, and X. Wu, “A cooperative lane change model for connected and autonomous vehicles on two lanes highway by considering the traffic efficiency on both lanes,” *Transportation Research Interdisciplinary Perspectives*, vol. 9, p. 100310, 2021.
- [12] L. Weber, *Driver Modeling and Simulation of Lane Change Situations: Influence of Different Rear View Mirror Types on Gap Acceptance Behavior*. Ph.D. Thesis, University of Oldenburg, Oldenburg, Germany, 2017.
- [13] A. Lange, *Gestaltung der Fahrdynamik beim Fahrstreifenwechselmanöver als Rückmeldung für den Fahrer beim automatisierten Fahren*. Ph.D. Thesis, Technical University of Munich, München, Germany, 2018.
- [14] J. Erdmann, “SUMO’s Lane-Changing Model,” in *Modeling Mobility with Open Data* (M. Behrisch and M. Weber, eds.), pp. 105–123, Cham, Switzerland: Springer International Publishing, 2015.
- [15] J. Ossig and S. Cramer, “Tactical Decisions for Lane Changes or Lane Following? Development of a Study Design for Automated Driving,” in *International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI)*, pp. 23–26, 2020.
- [16] J. Ossig, S. Cramer, A. Eckl, and K. Bengler, “Tactical Decisions for Lane Changes or Lane Following: Assessment of Automated Driving Styles under Real-World Conditions,” *IEEE Transactions on Intelligent Vehicles*, in press.
- [17] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, “The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems,” in *International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2118–2125, 2018.
- [18] A. Kesting, M. Treiber, and D. Helbing, “General Lane-Changing Model MOBIL for Car-Following Models,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1999, no. 1, pp. 86–94, 2007.
- [19] J. Kienzle, *Analyse von Einzelfahrzeugdaten: Verkehr verstehen*. Diploma Thesis, University of Stuttgart, Stuttgart, Germany, 2001.
- [20] M. Festner, *Objektivierte Bewertung des Fahrstils auf Basis der Komfortwahrnehmung bei hochautomatisiertem Fahren in Abhängigkeit fahrfremder Tätigkeiten*. Ph.D. Thesis, University of Duisburg-Essen, Duisburg, Germany, 2019.

Controllability of lateral drift failures while driving with SAE Level 2 Advanced Driver Assistance Systems

Norbert Schneider, Lenne Ahrens and Achim Pruksch*

Abstract: SAE Level 2 systems, which allow the driver to drive hands-off while monitoring visual attention, raise the question whether the driver is still able to fulfill his allocated responsibility (object and event detection and response). We developed a setup to evaluate the driver's ability to detect and respond to silent lateral system failures. The results indicate that an attentive driver can perceive and handle lateral system failures, but a cognitive misattribution of the systems capabilities might occur even for attentive drivers. This differentiation of causal factors allows developers to focus on adequate countermeasures to increase safety of SAE Level 2 systems.

Keywords: controllability, SAE Level 2, safety, SOTIF

1 Introduction

SAE Level 2 systems, which can continuously take over lateral and longitudinal control while the driver is still responsible for the dynamic driving task and the supervision of the function, are becoming more prevalent in series vehicles. This responsibility is summarized under the term "object and event detection and response" [1]. Various studies raise concerns about the ability of the driver to fulfill this responsibility and provide a fallback in case of system errors, especially when the system is used for an extended time [2] [3]. A potential cause could be a decrease in driver vigilance associated with the monotonous monitoring task (see [4]), but additional cognitive and behavioral aspects e.g., the development of automation complacency [5] and the potential for misuse in the form of non-driving related tasks (NDRT) should be considered as well. Currently, the UNECE R 79 allows only SAE Level 2 systems designed for hands-on use, which include hands-on monitoring [6]. However, system concepts in the U.S., China and Japan also include SAE Level 2 systems which allow the driver to take his hands off the steering wheel (e.g., Cadillac Super Cruise, Ford Blue Cruise, Nissan Pro Pilot). To ensure that the driver is still fulfilling his responsibility according to the SAE Level 2, a driver monitoring camera is used, which monitors the driver's visual attention. From a human factors perspective, this design decision brings these SAE Level 2 systems closer to a SAE Level 3 system, although the basic requirements towards the driver remain the same as for a SAE Level 2 hands-on system. However, only a few studies analyzed the driver's ability to perceive and compensate silent failures when driving with combined longitudinal and

*Cross-Domain Computing Solutions, Human Centered Advanced Development (XC/ENA1), Robert Bosch GmbH, Postfach 13 55, 74003 Heilbronn, Germany (norbert.schneider4@de.bosch.com, lenne.ahrens@de.bosch.com, achim.pruksch@de.bosch.com)

lateral control (e.g., [7] [8] [9] [10]). Commonly accepted controllability thresholds for lateral failures have been derived almost exclusively from manual driving [11]. This raises the question, whether results can be transferred to systems with continuous lateral and longitudinal guidance where the driver only has a monitoring task. Most publicly available studies combining longitudinal and lateral control have been conducted with SAE Level 3 systems in mind. These SAE Level 3 systems are assumed to provide a notification to the driver in case of a system failure or hazardous situation. However, it is a common misconception that an SAE Level 3 system will provide a take-over notification in all cases [12]. Nonetheless, it is assumed to be unfeasible for SAE Level 3 systems to not provide a notification, especially if they allow the driver to attend to non-driving related activities. This might also transfer to SAE Level 2 systems which allow the driver to take his/her hands off the wheel (see [9]). The available literature offers only limited insights regarding the capabilities of drivers to handle silent system failures, which require action within a short time-frame, when driving hands-off. This limited focus of human factors research on take-over scenarios in the automotive domain and its drawbacks have also been addressed by [13]. Previous studies looking at the controllability of system failures while driving hands-on can only provide general boundaries for the driver's capability to handle system failures even when driving hands-off. However, the results cannot be directly transferred to hands-free driving because the haptic feedback loop, which allows for very fast reflex-like counter-reactions, is missing. Internal studies indicate that the established test method for controllability, which evaluates whether drivers can stay in a driving lane with limited width, that drivers are not able to control limited failure dynamics while driving hands-off. Further, explorative studies revealed that this could be a consequence of the experimental setup, which utilized traffic cones positioned at the lane boundaries to increase pressure for the driver to act. Therefore, we created a new experimental setup to analyze the drivers' ability to handle lateral system failures while driving with continuous longitudinal and lateral control. The experimental setup allows lane deviations and lane departures without resulting in an objective fail (in contrast to the traffic cone setup) and provides a clear and adaptable pressure to act for the driver. In addition, it can be used with naïve participants at higher vehicle speeds (up to 120 km/h) to provide a clear pass/fail criterion without compromising safety; a clear improvement compared to method proposed by [14].

One goal of this study was to assess the viability of the proposed experimental setup. The other goal was to evaluate whether attentive drivers, driving hands-off with combined longitudinal and lateral control by a SAE Level 2 system, can handle silent lateral failures with limited failure dynamics.

2 Methods

2.1 Setup

The study was conducted on the TRIWO test track in Pferdsfeld, Germany. A round course with a length of 5km was used for the tests, including a straight highway section with 3 driving lanes and an additional part with two driving lanes and variable speed limits (Figure 1).

To simulate a SAE Level 2 capable vehicle with hands-free driving abilities, we used

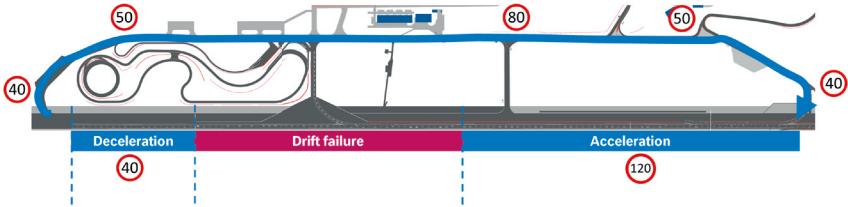


Figure 1: Round course for testing on the test track in Pferdsfeld.

a functional prototype vehicle from the Robert Bosch GmbH. Basis for this functional prototype was a VW Golf VII equipped with a DGPS system (iMAR iTraceRT-F402/7), programmable cluster display, capacitive steering wheel, driver monitoring camera, dual pedal controls for the safety driver, control switches for the experimenter and data recording equipment. A safety co-driver (passenger seat) and an experimenter (rear seat) accompanied the participant.

The prototype SAE Level 2 hands-free system included longitudinal and lateral guidance, intelligent speed adaption for curves and on/off-ramps to the highway section and automatic lane change capabilities. Longitudinal and lateral control could be activated simultaneously by pressing a button on the steering wheel. Before transitioning to hands-free driving, the driver had to drive hands-on for a limited time and was informed about the availability of the hands-free mode. To activate hands-free driving the driver had to take his hands off the steering wheel. The system could always be overruled or deactivated by turning the steering wheel (lateral guidance temporarily deactivated), applying the brake pedal, or press a button on the steering wheel (longitudinal and lateral control deactivated). All driving modes were displayed via a prototype human machine interface (HMI) in the programmable cluster display. The prototype HMI included a multi-stage warning concept (cf. [15]), which visually and acoustically warned the driver after 3 seconds eyes-off road (EOR) time (stage 1), followed by a visual-acoustic warning after additional 5 seconds, asking the driver to put his hands on the steering wheel (stage 2). Finally the driver is asked to take over manual control while the vehicle is decelerating, and visual and acoustic warnings are presented continuously (stage 3).

2.2 Test scenario

The test scenarios were set up on the highway section of the round course and consisted of 5 obstacles, positioned on left and right side of the driving lane (Figure 2). In both test scenarios a lateral drift with a defined maximum lateral velocity (0.2 m/s vs. 0.6 m/s) was triggered at pre-defined GPS positions after the experimenter enabled the failure activation via the safety control switch. The GPS position for the trigger was chosen based on the distance to the second obstacle on right side, the lateral drift velocity, the targeted overlap with the obstacle (25%) and the test speed of 120 km/h. This resulted in two different GPS trigger positions, depending on the maximum lateral drift velocity ($TTC = 2.8$ seconds for a drift velocity of 0.6 m/s and $TTC = 5.8$ seconds for 0.2 m/s). The lateral drift was initiated without any further notifications to the driver or changes in the HMI. In addition, the build-up of the lateral drift did not occur abruptly but was

build-up gradually to avoid noticeable lateral jerks of the vehicle. In both scenarios a crash with the objects would occur if the driver did not react e.g., by steering back into the driving lane.

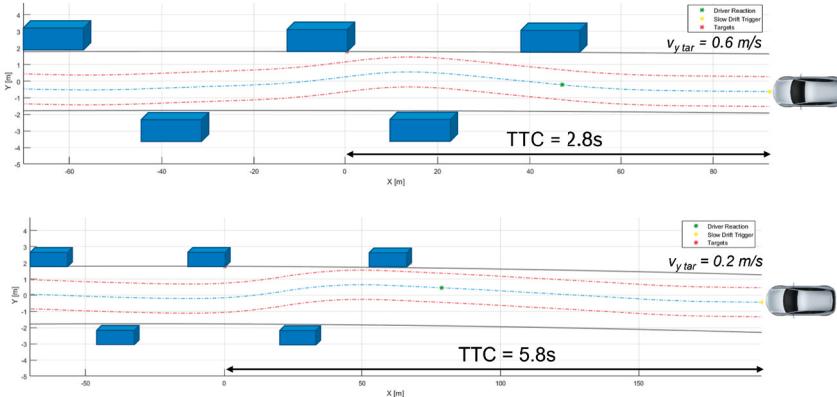


Figure 2: Schematic illustration of the test scenario setup. The trigger point was adapted based on lateral drift velocity which resulted in different TTCs.

2.3 Procedure

To conceal the true purpose of the study, participants were told that the test focused on user acceptance of a novel automation system, which allowed them to drive partially automated while taking their hands off the steering wheel under special conditions. Before the testing each participant received written information about the system's capabilities and potential system boundaries. This included the explicit information about the necessity for the driver to take over control of the vehicle e.g., if the system departed from the current driving lane. Afterwards the participants started with manual drive to familiarize themselves with the test track and test vehicle. The manual drive was continued until participants reported that they were familiar with the vehicle and the test track (5-10 minutes). After completing the manual drive, combined longitudinal and lateral control were activated for the first time but restricted to hands-on driving only. The participants were asked to explore the system and the experimenter noted their comments. During this exploration phase, the participants were instructed to try out all options to activate/deactivate and overrule the system (e.g., pressing the button on the steering wheel, applying the accelerator, applying the brake pedal, turning the steering wheel). After a short pause and filling out a questionnaire about the hands-on system, the hands-free system was activated, and participants were asked again to explore the system. Participants were instructed to overrule the system including putting the hands on the steering wheel to adjust the lane position. If participants asked about their responsibility, we reminded them verbally that they are responsible for the driving task and encouraged to take over control whenever they wanted. Before activating the GPS trigger and initiating

the lateral drift towards the obstacle, the experimenter made sure that the participant was attentive but not in a hyper-vigilant state (e.g., constantly trying to take over the system without an apparent reason, not taking the hands off the steering wheel on the straight section of the highway, keeping the hands up and close to the steering wheel). After the test a semi-structured interview was conducted, and participants filled out additional questionnaires.

2.4 Sample

In total $N=61$ (40 male / 21 female) participants between 20 and 72 years of age and an average annual mileage of 19.778 km took part in the study. Out of the $N=61$ participants only $N=54$ filled out the demographic questionnaire completely. 26% ($N=14$) reported to be experienced with adaptive cruise control (ACC), 30% ($N=19$) reported to be experienced with lane keeping assistance and 4% ($N=2$) reported to be experienced with assistance systems with combined longitudinal and lateral guidance.

3 Experimental design and data analysis

3.1 Experimental design

The experiment was conducted as a between-subject design with the experimental factor lateral drift velocity (0.2 m/s vs. 0.6 m/s), meaning that each participant only experienced one drift scenario (either with 0.2 m/s or with 0.6 m/s). To fulfill the requirements according to ISO 26262 and the Response Code of Practice regarding required sample sizes for a controllability assessment of C2, (C2 = 90% of drivers are able to control the failure) a valid sample size of at least $N=20$ per experimental group was targeted. In total, $N=61$ took part in the study. $N=7$ cases had to be rated as invalid either due to the participants having the hands at the steering wheel ($N=3$) or missing data recordings ($N=4$). No crash was observed in any of the invalidated cases. For the analyses $N=34$ participants were considered with a lateral drift velocity of 0.6 m/s and $N=20$ with a lateral drift velocity of 0.2 m/s.

3.2 Data analysis

Data analysis was based on the number of passed/failed test cases. In addition, vehicle dynamic parameters, video recordings of the driver, lane position and distance to the targets and questionnaire and interview data were considered for descriptive and exploratory analysis. Perception time and hands-on time were estimated based on video labeling; the steering reaction time was estimated on the first increase of steering torque after the failure was triggered. Time to collision (TTC) to the obstacle and time to lane crossing (TLC) were estimated based on the vehicle dynamics parameters and the GPS position of the vehicle, the lane markings, and the target.

4 Results

Four participants failed to control the situation. Three with a drift velocity of 0.6 m/s and one with a drift velocity of 0.2 m/s. All other participants ($N=50$) were able to avoid a collision with the target and control the vehicle safely after the evasive maneuver.

4.1 Driver behavior in failed cases

A detailed video analysis revealed the same behavioral pattern in all failed cases (see Figure 3). At first, the participants moved their hands close to the steering wheel (Figure 3, middle), indicating that they perceived the lateral drift and considered taking over the control of the vehicle. This initial reaction was followed by putting the hands down again (Figure 3, right).



Figure 3: Sequence of driver reactions in the cases with observed collisions. Approaching obstacle with hands on lap (left), hands close to steering wheel (middle), hands put down shortly before collision (right).

4.2 Reaction and perception time

The pattern observed in the failed cases is also reflected in the perception time (see Figure 4). Based on the video labeling, every participant perceived the lateral failure before the crash (range between 1,05 and 1,59 s for a drift velocity of 0.6 m/s and up to 4.16 s for a drift velocity of 0.2 m/s). Although participants showed a steering reaction at a different time to collision (TTC) depending on the drift velocity, the steering reaction occurred at a similar time to lane crossing (TLC). However, not all participants responded before departing from the lane. In some cases departing from the lane was observed although no collision occurred with the target (see Figure 5). In addition, the semi-structured interview revealed the following exemplary responses from the participants to the question why they did not take over control in the situation:¹

- "I trusted the vehicle and hoped, that it would react." (crashed participant 1)
- "I expected the system to intervene." (crashed participant 2)
- "The system did not tell me to take over." (crashed participant 3)

¹similar responses have been reported by [16]

- "I perceived the system to be autonomous, because I could drive without any input." (crashed participant 4).

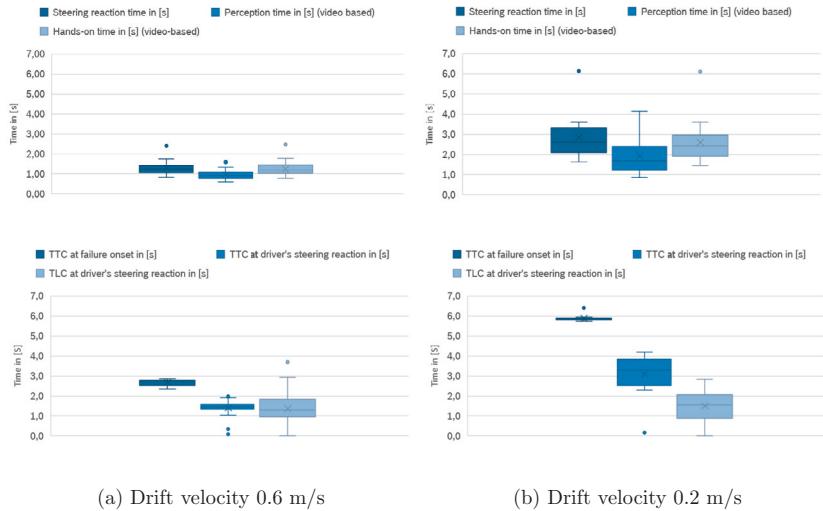


Figure 4: Steering reaction time, perception time and hands-on time (top) and TTC at failure onset as well as TTC and TLC at the time of the steering reaction (bottom) for both test cases.

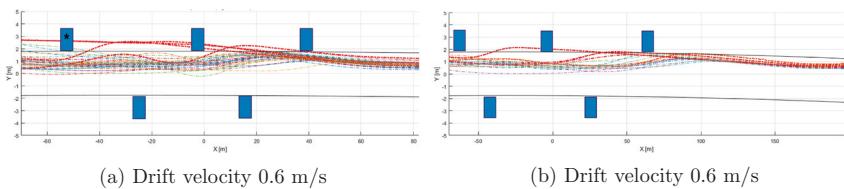


Figure 5: Lateral position of the right corner of the car for all test-cases. Red lines indicate cases in which the vehicle crossed the lane boundary.

5 Summary and discussion

The main goal of this study was to evaluate whether an attentive driver (who is using the SAE Level 2 system with hands-free driving capabilities according to the driver manual) can perceive lateral drift failures and avoid a collision without losing control of the vehicle. The study was conducted on a closed test track with the help of a prototype vehicle and a SAE Level 2 hands-free system with a gaze-based driver monitoring system and a staged warning design.

The results indicate that all participants were able to perceive the lateral drift failure (irrespective of its dynamics) in time and even before departing from the driving lane. However, four participants did not intervene which led to a collision. The observed behavior as well the verbal responses of participants indicate that this behavioral pattern is mainly a result of an overtrust in the system's capabilities, despite explicit information that they are required to take over control in cases where the vehicle is departing from the lane. Interestingly, the participants reacted at a comparable time to lane crossing (TLC) irrespective of the initial drift velocity. This could indicate that the TLC is a relevant cue for participants to intervene although not all participants, which avoided a collision, stayed within the lane boundaries. This strengthens our initial assumption that departing from the lane does not necessarily indicate that drivers cannot react adequately. However, this also indicates that a reaction might only occur if an imminent hazard is perceived by the driver. Although this finding seems to be trivial it highlights the necessity to ensure an adequate situational awareness of the driver when driving with an SAE Level 2 hands-free system.

Based on these results, we conclude that the problem is not the driver's ability to handle limited lateral failure dynamics when driving hands-free but rather a cognitive misattribution of the system's capability which is built up by experience of the system and user expectations. This phenomenon has been described before as "automation expectation mismatch" [17], "mode confusion" or "automation complacency" [5] and several causal factors are discussed in the literature (e.g., perceived system performance or the out-of-the-loop concept [18]). Although the solution to this problem might not be trivial, being able to differentiate between "driver ability" and "cognitive misattribution" is important for the safety evaluation of SAE Level 2 hands-free systems. The driver's ability to handle system failures is an integral part of the functional safety evaluation of driver assistance systems (e.g., for the assessment of controllability). If a failure is rated to be uncontrollable, one possible solution is to limit the failure dynamics. Our study clearly indicates that limiting the failure dynamics of a lateral drift does not solve the problem (that drivers will depart from the lane in case of lateral failure or even collide with an obstacle in the neighboring lane). However, it also demonstrates that attentive drivers can perceive and compensate lateral failures with limited dynamics. Therefore, we assume that the analyzed lateral failures are controllable for at least 90% of the population (corresponding to a controllability level of C2) if the driver is visually attentive and has adequate situational awareness. Nonetheless, the shown cognitive misattribution of the system's capabilities must be addressed. This problem is covered in the ISO 21448 [19] and could be attributed to the aspect foreseeable misuse. Due to this differentiation, it is possible to focus on adequate counter measures e.g., better driver training and information about the system's capabilities [20], and the driver's responsibilities when driving with a hands-free system.

It has yet to be clarified whether the same behavior would have been observed outside of this experimental context. Although similar results can be observed in test track studies [9] [10] [21] [17] as well as simulator studies [8] the question remains: Would the participants have behaved differently on-road? Due to the safety constraints, a trade-off between a realistic hazard scenario and the safety of the participants had to be accepted for the experimental setup. Naturalistic driving studies and field observations with already available SAE Level 2 hands-free capable vehicles could provide further information regarding whether or not drivers show similar behavioral patterns on the road.

References

- [1] SAE On-Road Automated Vehicle Standards Committee. *J3016 Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. Warrandale, USA, Apr. 2021. doi: https://doi.org/10.4271/J3016_202104.
- [2] Victoria A. Banks, Alexander Eriksson, Jim O'Donoghue, and Neville A. Stanton. "Is partially automated driving a bad idea? Observations from an on-road study". In: *Applied Ergonomics* 68 (2018), pp. 138–145. ISSN: 0003-6870. doi: <https://doi.org/10.1016/j.apergo.2017.11.010>. URL: <https://www.sciencedirect.com/science/article/pii/S0003687017302594>.
- [3] Alexander Frey. "Müdigkeit und Vigilanz in einer automatisierten Realfahrt". In: *Mensch-Maschine-Mobilität 2019*. Der (Mit-)Fahrer im 21. Jahrhundert! Vol. 2360. Düsseldorf: VDI Verlag, 2019, pp. 121–132. doi: <https://doi.org/10.51202/9783181023600-121>.
- [4] Robert Molloy and Raja Parasuraman. "Monitoring an automated system for a single failure: Vigilance and task complexity effects". In: *Human Factors* 38.2 (1996), pp. 311–322.
- [5] Raja Parasuraman and Dietrich H Manzey. "Complacency and bias in human use of automation: An attentional integration". In: *Human factors* 52.3 (2010), pp. 381–410.
- [6] United Nations. *Regulation No 79 of the Economic Commission for Europe of the United Nations Uniform provisions concerning the approval of vehicles with regard to steering equipment*. Standard. UN/ECE, Oct. 2018. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=uriserv:OJ.L_.2018.318.01.0001.01.ENG.
- [7] Giulio Bianchi Piccinini, Esko Lehtonen, Fabio Forcolin, Johan Engström, Deike Albers, Gustav Markkula, Johan Ledin, and Jesper Sandin. "How do drivers respond to silent automation failures? Driving simulator study and comparison of computational driver braking models". In: *Human factors* 20 (7 May 2019), pp. 1212–1229. doi: <10.1177/0018720819875347>.
- [8] Laura Wörns. "Analysis of drivers' reaction to automation failures in a curve scenario". Master Thesis. Sweden: Chalmers University of Technology / Department of Mechanics and Maritime Sciences, 2018. URL: <https://odr.chalmers.se/handle/20.500.12380/254961>.

- [9] Myra Blanco, Jon Atwood, Holland Vasquez, Tammy Trimble, Vikki Fitchett, Joshua Radlbeck, Gregory Fitch, Sheldon Russell, Charles Green, Brian Cullinane, and Justin Morgan. *Human Factors Evaluation of Level 2 and Level 3 Automated Driving Concepts*. Report No. DOT HS 812 182. Washington, DC: National Highway Traffic Safety Administration, Aug. 2015. DOI: 10.13140/RG.2.1.1874.7361. URL: https://www.nhtsa.gov/sites/nhtsa.gov/files/812182_humanfactorseval-1213-automdrivingconcepts.pdf.
- [10] Robert E. Llaneras, Brad R. Cannon, and Charles A. Green. “Strategies to Assist Drivers in Remaining Attentive While Under Partially Automated Driving: Verification of HumanMachine Interface Concepts”. In: *Transportation Research Record* 2663.1 (2017), pp. 20–26. DOI: 10.3141/2663-03. eprint: <https://doi.org/10.3141/2663-03>. URL: <https://doi.org/10.3141/2663-03>.
- [11] Alexandra Neukum and Hans Peter Krüger. “Fahrerreaktionen bei Lenksystemstörung: Untersuchungsmethodik und Bewertungskriterien”. In: *VDI Berichte* 1791 (2003), pp. 297–318.
- [12] Philip Koopman. *SAE J3016 User Guide*. Carnegie Mellon University. Sept. 2021. URL: <https://users.ece.cmu.edu/~koopman/j3016/>.
- [13] Joost de Winter, Neville Stanton, and Yke Bauke Eisma. “Is the take-over paradigm a mere convenience?” In: *Transportation Research Interdisciplinary Perspectives* 10 (2021), p. 100370. ISSN: 2590-1982. DOI: <https://doi.org/10.1016/j.trip.2021.100370>. URL: <https://www.sciencedirect.com/science/article/pii/S2590198221000774>.
- [14] Nadja Schömig, Katharina Wiedemann, Ruth Julier, Alexandra Neukum, Andre Wiggerich, and Heike Hoffmann. *Methoden für die Bewertung der Mensch-Maschine-Interaktion beim teilautomatisierten Fahren*. 2021. ISBN: 978-3-95606-591-0.
- [15] Alexandra S. Mueller, Ian J. Reagan, and Jessica B. Cicchino. “Addressing Driver Disengagement and Proper System Use: Human Factors Recommendations for Level 2 Driving Automation Design”. In: *Journal of Cognitive Engineering and Decision Making* 15.1 (2021), pp. 3–27. DOI: 10.1177/1555343420983126. eprint: <https://doi.org/10.1177/1555343420983126>. URL: <https://doi.org/10.1177/1555343420983126>.
- [16] Pär Gustavsson, Trent Victor, Joel Johansson, Emma Tivesten, Regina Johansson, and Mikael Ljung Aust. “What were they thinking? Subjective experiences associated with automation expectation mismatch”. In: Oct. 2018.
- [17] Trent W. Victor, Emma Tivesten, Pär Gustavsson, Joel Johansson, Fredrik Sangberg, and Mikael Ljung Aust. “Automation Expectation Mismatch: Incorrect Prediction Despite Eyes on Threat and Hands on Wheel”. In: *Human Factors* 60.8 (2018). PMID: 30096002, pp. 1095–1116. DOI: 10.1177/0018720818788164. eprint: <https://doi.org/10.1177/0018720818788164>. URL: <https://doi.org/10.1177/0018720818788164>.

- [18] Natasha Merat, Bobbie Seppelt, Tyron Louw, Johan Engström, John Lee, Emma Johansson, Charles Green, Satoshi Kitazaki, Chris Monk, Makoto Itoh, Daniel McGehee, Takashi Sunda, Kiyozumi Unoura, Trent Victor, Anna Schieben, and Andreas Keinath. "The Out-of-the-Loop concept in automated driving: proposed definition, measures and implications". In: *Cognition, Technology Work* 21 (Feb. 2019). DOI: 10.1007/s10111-018-0525-8.
- [19] International Organization for Standardization. *ISO/PAS 21448:2019 Road vehicles Safety of the intended functionality*. Standard. Geneva, CH: International Organization for Standardization, Jan. 2019.
- [20] Annika Boos, Birte Emmermann, Bianca Biebl, Anna Feldhütter, Martin Fröhlich, and Klaus Bengler. "Information Depth in a Video Tutorial on the Intended Use of Automated Driving". In: May 2021, pp. 575–582. ISBN: 978-3-030-74607-0. DOI: 10.1007/978-3-030-74608-7_70.
- [21] Linda Pipkorn, Trent Victor, Marco Dozza, and Emma Tivesten. "Driver conflict response during supervised automation: Do hands on wheel matter?" In: *Transportation Research Part F: Traffic Psychology and Behaviour* 76 (Jan. 2021), pp. 14–25. DOI: 10.1016/j.trf.2020.10.001.

OMEGAFormat: A Comprehensive Format of Traffic Recordings for Scenario Extraction

Maike Scholtes*, Michael Schuldes*, Hendrik Weber*, Nicolas Wagener*,
Michael Hoss*, and Lutz Eckstein*

Abstract: The most crucial aspects influencing the behavior of Automated Vehicles (AVs) are static environment, dynamic objects and weather. Traffic recordings that include these aspects allow the derivation of comprehensive test scenarios of AVs. However, existing recording descriptions are either not designed for test scenario derivation or use separate formats for the mentioned aspects. Therefore, in this paper, we present an hdf5-based recording format that unifies the data basis for scenario derivation by covering all layers of the 6-Layer Model (6LM). By open-sourcing the format specification along with a library[†] and converters and enrichers[‡], we hope to facilitate research on scenario generation considering all relevant aspects of traffic.

Keywords: automated driving, environment description, scenario-based validation, traffic recording

1 Introduction

For the verification and validation of Automated Driving Function (ADF) the scenario-based approached is favored. Driving randomly in real-world traffic to achieve a safety assurance is not feasible [1]. Therefore, the scenario-based approach focuses on investigation of situations that are in some way challenging for the ADF. To find such scenarios, two methods are known: knowledge-based and data-driven [2]. The format developed in this paper was mostly developed to facilitate the data-driven approach. However, the two approaches are interrelated, meaning that it can also contribute to the knowledge-based approach.

A scenario-based validation and testing workflow was developed in the research project PEGASUS [3]. Different sources of data, such as field test- or naturalistic driving data and data from accident databases, can function as input information [4]. The available data can be stored in a database and subsequently analyzed to derive logical scenarios [5] and transform those into sets of test cases. The format developed in this work comes into play at the beginning of the workflow when gathering data. A wide variety of input data sources are possible. However, those data sources all provide different information

*The authors are or were with the research area Vehicle Intelligence & Automated Driving, Institute for Automotive Engineering, RWTH Aachen University, 52074 Aachen, Germany email: {schuldes, etc., eckstein}@ika.rwth-aachen.de.

[†]Format and library: https://github.com/ika-rwth-aachen/omega_format/tree/master/omega_format.

[‡]Data converters: [www.github.com/ika-rwth-aachen/vvm-rec-converters](https://github.com/ika-rwth-aachen/vvm-rec-converters).

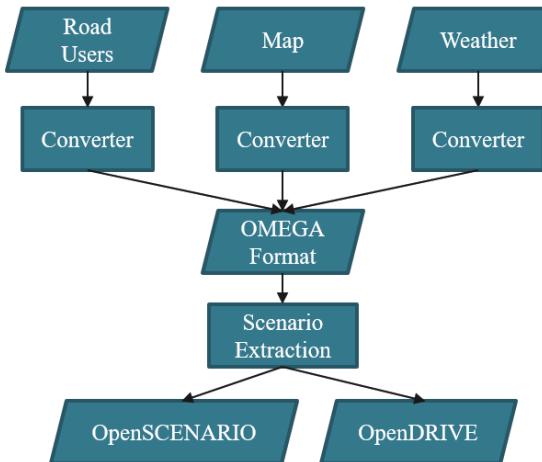


Figure 1: The OMEGA format harmonizes the data representation of traffic recordings such that approaches for scenario extraction can rely on this defined input format.

in different formats. In order to establish a uniform data processing chain, a harmonized data format is needed [4].

Currently the existing formats and data as well as the processing algorithms and concepts mostly consider information on road users [6] [7]. To the best of the knowledge of the authors, there is no format available that covers all six layers of the 6LM for environment description [8]. The model is commonly used to provide a structured description and categorization of the complex and open real-world design domain. The format developed in this work is not only important in terms of harmonization, but also due to it covering all six layers of the 6LM. This is essential since relevant scenarios that need to be considered in the validation process can also be triggered through properties of the road network [9] or weather data.

The latter is also of utmost interest when assessing perception data gathered by an ADF [10]. The traffic data recorded can not only be used as an input source for the scenario-based approach focusing on planning and acting of an ADF, but also as reference data that can be compared to recorded perception/sensor data.

Our contributions in this paper are:

- We propose an open data format including tooling that systematically covers all six environment description layers [8]. -> Section 3.2, 3.3, 3.4 and 3.5
- We provide research on the entities relevant for description and data analysis. At the same time, the goal is that those factors stay retrievable. Furthermore, common classifications of factors are provided. This is also of use for knowledge-based scenario approaches that focus on ontology creation [11]. -> Section 3

- The format stays compatible with existing formats that only cover certain aspects of the 6LM. There are converters from other formats into our holistic format available.
-> Section 4
- The format provides a clever inclusion of temporary aspects in the map description. Therefore, it can be easily reused for recordings at the same locations even if temporary aspects are present. -> Section 3.2
- Our format provides a description through polylines that stays understandable, interpretable and is flexible. For instance, repeating objects can be easily described without having to provide information on each individual object. -> Section 3.1 and 3.2
- We introduce one consistent time stamp for easy comparison. This is especially handy when using the proposed format in combination with perception recordings.
-> Section 3.1 and 6

2 Related Work & Methods

In general, there is a wide variety of different data containers and encodings available. They were developed for different use cases such as pure data exchange or semantic recording and feature different compression rates. Some formats, also used in the automotive context, are .csv, .mat, .json, .xml, .hdf5 or rosbags. In [12], a more detailed analysis of different file formats can be found.

Various existing formats addressing individual layers of the 6LM were analyzed during the development of the OMEGA format. In the automotive context, divers partly standardized formats exist that use the different data containers. Coming from the area of simulation, the formats OpenDRIVE [13] and OpenSCENARIO [14] are used that are both based on an xml schema. Furthermore, Lanelet2 [15] is a map format often used in the scientific context for trajectory planning of an ADF. All the different formats require different libraries and tools to view them. Some of which are not free even though the formats are generally open source. Other research projects like PEGASUS [3] and L3Pilot [12] have developed their own formats to save traffic recordings in the .mat or .hdf5 schema. However, those formats were developed with the use case highway in mind and mostly focus on road user information. The aim of the format described in this work is to cover all layers of the 6LM.

To meet this requirement also existing taxonomies and ontology schemas for ADF were studied. Those feature schemas to describe the real-world conceptually rather than through data. For this work specifically a taxonomy by NHTSA [16] and a world model ontology were analyzed [17], [18].

The developed OMEGA format is designed to be applicable for different modes of data recording. Traffic recordings can for instance be acquired by a sensor equipped vehicle or infrastructure or through drones. The OMEGA format can be used for the different data sources or a fusion of such [19]. This facilitates the processing of the different data sources to extract scenarios and implement a validation process for ADFs.

3 The OMEGA Format

The OMEGA data format uses the hdf5 format [20] to store the data. The hdf5 data model utilizes *groups* and *datasets* to store data objects. Furthermore, attributes are used to describe certain objects to which they are attached. They contain metadata of those objects. The documentation available at [21] provides further information how the individual signals are stored. In general, the hdf5 file format was chosen as it proved to be useful in other public research projects [12]. The main advantage of the format is its portability across different platforms. The format can be used in different programming languages such as Python or C++ and is also supported by newer versions of MATLAB. Investigations regarding the memory consumption and achievable compression can be found in [12].

Section 2 has introduced different formats currently used in the area of validation and verification for ADFs. Those formats usually depict only certain aspects of the 6LM [8]. For instance, formats like OpenDRIVE or Lanelet2 concentrate on the description of the road network (Layers 1 to 3) while OpenSCENARIO depicts dynamic information (Layers 4 to 6). The format introduced at hand aims at addressing all layers of the 6LM in one format with one consistent time stamp (see Section 3.1). Furthermore, data campaigns in larger research projects with different entities tend to produce several different recording formats. For the analysis, however, it is very useful to establish one format. Obviously, when deriving the format, it should be considered which signals and information are needed for the applications planned so that all of those can be retrieved by all entities recording data. However, when designing the format at hand, it was also considered which signals are retrievable in some way in the first place, since some information might be nice to have, but impossible to gather without large efforts.

The following sections will give an overview on the different signals used in the format. For extensive details and technical information the reader is referred to the Github repository [21] and its corresponding documentation.

3.1 Basics

At the top level, the format contains some descriptive meta information in form of hdf5 attributes. Those information lets the user know which format version the file is in and which partner recorded the data at which time of the day (e.g. needed to enrich weather information). Furthermore, a unique recording number is set. This is helpful to link the data to other information such as perception data also captured during the recording (see Sec. 6 for more details) or reference videos. In contrast to formats used in PEGASUS or L3Pilot, which are designed for highway applications, the format at hand uses an absolute coordinate system. This has the clear advantage that no predefined ego vehicle is existent. It can be imagined as a birds eye perspective, such as received by using drones [6]. The absolute coordinate system is located at a certain point given in lateral and longitudinal coordinates. The information on this point is also provided in the top-level attributes. All positional information is given relatively in respect to that point. This is legit as the format is designed mainly for urban applications, such as recordings at an intersection, and not for hundreds of kilometers along a highway. Furthermore, for later analysis purposes, it is important to know if the recording features natural behavior of all traffic

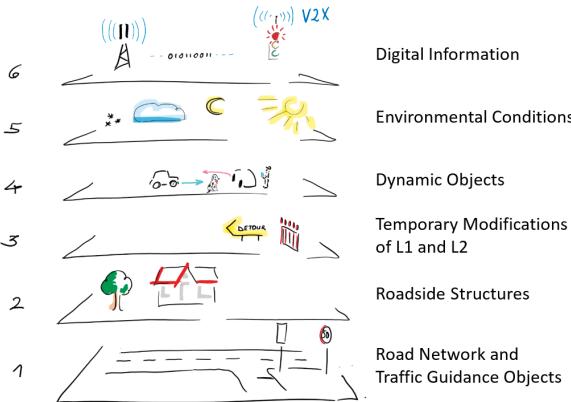


Figure 2: Different layers of the 6LM also covered in the format of this work.

participants (i.e. is not staged) and natural exposure (i.e. no preselection has been made on the basis of various criteria before providing the data). This information is also stored in the attributes.

One special characteristic of the format is the synchronized time stamp for all recorded time-dependent data. All data that is to be recorded in the format and consists of a time series, such as trajectory data of an object or weather data, uses the same time stamp vector that is contained in the format only once. I.e., the x th entry in the object information, such as longitudinal position, corresponds to the x th entry in the time stamp vector. This enables synchrony of the data already by design.

In general, the format uses so called polylines, i.e. a set of different points with x, y and z coordinates, to describe the road outline, trajectories, etc. This is in contrast to the design chosen in OpenDRIVE which uses euler spirals, etc., but is better suited for the use case. Polylines can easily be extracted from real world data, which is the primary use case of this format and for which already fitted spirals, and polynomials are not given. Above all, it stays easily readable and also complex shapes, e.g., for different buildings, can be described by a polygon course.

In the following, the information is structured by the different layers of the 6LM. Since map data, information on dynamic objects and weather data usually comes from different sources, separate fields are provided to note down the version of the corresponding converters used. This allows for traceability if changes are made to a certain converter used by a certain data provider.

3.2 Map Data (Layers 1,2,3 of the 6LM)

The map data includes information that is static for the duration of the recording. This comprises the road network and traffic regulating objects as well as roadside structures (Layer 1 and 2 of the 6LM). Furthermore, the elements described here can experience

temporary changes (Layer 3 of the 6LM), such as during road work. In such cases, existing information can be flexibly modified and new elements can be added noting that they are temporary. This is advantageous if generating several recordings at the same intersection with road work being present for some of those recordings. The existing road description can stay unaltered with the temporary modifications being stacked on top. All information on Layer 1 to 3 entities of the 6LM is included in the *road* group of the format.

Wherever possible, the format sticks to existing norms of road construction [22] to simplify the description. Furthermore, it uses official type catalogs, such as the traffic sign catalog of the Bundesanstalt für Straßenwesen (BASt) [23]. Nevertheless, for some signals, such as condition of the road surface or marking, human judgment needs to be applied.

For some repetitive and extended structures, the format decreases the complexity of their description by providing specific boundary types. For instance, guard rails or reflector posts do not need to be modeled separately, but the boundary type of a lane can be set to the specific type. If the type is present, it is assumed that the corresponding element is extended along the lane without describing each entity individually. Furthermore, to decrease the complexity even further, the format separates the description of the geometrical border of a lane and its actual boundary type. This allows for the use of several boundaries. Therefore, a lane can feature a solid lane marking and a guard rail without having to describe these via separate lanes.

The map data of the format is compatible to OpenDRIVE and Lanelet2, but is more closely orientated towards Lanelet2. For instance, the lanes of a road are described via polylines that depict their borders rather than using one center line, center line offsets and lane widths such as in OpenDRIVE. Each road in the format at hand features borders, signs, lateral markings (e.g. stop line), road objects (e.g. a traffic island), structural objects (e.g. a building) and lanes. The lanes themselves then contain information on their boundary, flat markings (pictograms such as a bicycle sign) and their surface. Since the variety of structural objects that can be encountered in an urban setting can be enormous, the format currently sticks to the most common urban objects and to objects that are expected to impose challenges on the ADF (see also Sec. 6). However, it is generally possible to extend the format and describe all structures using polygon courses.

3.3 Dynamic Objects (Layer 4 of the 6LM)

The hdf5 groups *roadUser* and *miscObject* contain information on the dynamic objects (Layer 4 of the 6LM) that are generally able to move (even though they do not necessarily have to). It needs to be stressed that the format does not only concentrate on road users, but also on miscellaneous objects, such as animals, play equipment, etc.

In general, each road user or misc object features a type and a subtype, trajectory data containing position, velocity, etc. for the different time stamps and information on its bounding box. In case the data is recorded by one of the road users, e.g., in case of a measurement vehicle recording reference and/or sensor data, this can also be set as a flag for the corresponding object. The flag will be very important for later analysis, especially if sensor data was recorded along with the traffic recording (see also Sec. 6).

The number of different road user types can be very massive. For the individual road user types, the format sticks closely to the ones chosen in the nuscenes dataset [24]

including a few simplifications and adaptations. The classifications are chosen in such a way that they are reasonable and support later data evaluation. Again, for details the reader is referred to the Github repository [21] and its documentation, however, some important aspects are pointed out here. In general, all objects that are coupled, but move individually, such as trailers and pulling entities, are described individually in order for the bounding boxes to be correct in case of turning maneuvers, etc. Furthermore, subtypes that can be provided for the different types of road users are of interest. In general, each road user type can be assigned the subtype *emergency* or *construction*. Choosing this procedure, a car as well as a pedestrian can be marked as a first responder (e.g. police car and police officer) without having to introduce this individually for all types. Moreover, rideable objects such as bicycles, motorcycles and all kinds of personal mobility devices can be equipped with a subtype noting if the object has a rider or not. This can be helpful in the subsequent analysis to differentiate between parked rideable objects and such in use.

3.4 Environmental Data (Layer 5 of the 6LM)

All environmental data, weather data and weather related road conditions, are contained in the hdf5 group *weather* of the format. Such data can generally either be measured at time of recording or can be enriched at a later point in time. Each weather signal contained in the format includes a *source* information. In case of later enrichment, the Climate Data Center of the Deutscher Wetterdienst (DWD) [25] can be useful. It provides various weather information in different update intervals reaching from once per minute (precipitation) to once per hour. The format uses the same consistent time stamp described above and covers precipitation information, visibility, cloudiness, solar information, temperature, wind, air pressure and humidity as well as weather related road conditions. For the latter research results of countries/regions with more extreme weather were used [26]. Besides noting down the actual amount of a weather entity, such as precipitation amount or wind speed, also a classification is made. The classifications are based on existing scales [27] and weather warning stages [28].

3.5 Status of Traffic Guidance Objects (Layer 6 of the 6LM)

Layer 6 of the 6LM focuses on communication and cooperation aspects on the basis of digital data. It is expected to increase in importance with more propagated implementation of Vehicle-to-Vehicle (V2V) and Vehicle-to-Everything (V2X) concepts [29]. Layer 6 information for the format at hand comprises the status of traffic lights and switchable traffic signs. Note that the position and type of a traffic light / switchable sign is part of the map data (Layer 1). Only the time variable status of such signs is depicted here. Furthermore, the format covers the status of flashing lights or barriers, etc., if desired. Again, the holistic time stamp vector is used.

4 Using the format

The published library on Github [21] does not only provide the format and documentation on the choice of signals themselves, but also the complete data structure in Python and C++. This allows to easily read and write the corresponding hdf5 files. Furthermore,

enums are provided for the different types allowing for an easy mapping. Along with those functionalities comes a tool to visualize the data in the OMEGA format. For an easier analysis and visualization of large recordings, recordings can be separated into individual snippets featuring the different road users as ego vehicle. The visualizer reveals information on all different layers of the 6LM contained in the OMEGA format. I.e., it can also be used to visualize a specific road network and its regulatory elements. Moreover, the data in the OMEGA format can be verified and sanity checks are provided.

The library is published under MIT license that is available in the repository. Adaptations to the format can be made and we encourage users to provide adaptations that could be useful for others via merge requests.

Furthermore, various converters exist to convert existing data into the OMEGA format and use its functionalities. This is the case for OpenDRIVE, Lanelet2 and data from the inD dataset [6]. Additionally, the data can be enriched with weather data from the Climate Data Center of the Deutsche Wetterdienst [25]. Moreover, data from different input sources can be merged together into one OMEGA file.

5 Discussion

As stated above, the format covers all layers of the 6LM for environment description and is, therefore, of great value for the tool chain of scenario-based testing. The level of detail is chosen in such a way that the information should stay retrievable either at time of measurement or through data enrichment. However, depending on the recording method chosen, some signals might not be retrievable even though they would be useful for the analysis. Moreover, it needs to be kept in mind that the format and its choice of coordinate system is designed for the use in urban surroundings, e.g., at intersections. It would be usable, but is not designed for locally extended drives, such as, for instance, on a highway. Furthermore, the enrichment of the map data with traffic signs and markings, etc. as well as the status of the traffic lights often requires manual labor or even an additional observer at time of recording. Due to the format being developed in a German research project, it is currently designed for German roads. While it is extendable to other countries, this would require some major updates in terms of classification of the different signals.

We are aware that, through the choice of polylines, depending on the step size, e.g., map data is not as exact as it can be when using the OpenDRIVE format. However, focusing on real world data, extracting polylines has the benefit that they are always available and the format is purposely designed in such a way to simplify the description. A simplification is for instance also achieved when describing reoccurring objects like reflector posts or lanes that feature two boundaries (e.g. gurad rail and solid line). The format and the signals that are depicted in it are under continuous development and are, therefore, still undergoing changes to improve the format. The user should be aware of this when using the format. Updates will be provided with change logs and compatibility should be given.

6 Conclusion & Future Work

The format is currently used in a processing chain for validation of ADFs that is under development. The processing chain implements the data-driven scenario approach and gathers traffic recordings from multiple input sources in the harmonized OMEGA format in a database. The metrics applied in the database provide a scenario extraction that detects different scenario types [7]. The concept aims at deriving logical scenarios that can subsequently be transformed into test cases. Those can, e.g., be provided in OpenX formats for the later use in simulation. The input provided by the OMEGA format is also utilized to detect scenario semantics via so called General Descriptive Entity Attributes (GDEAs). This, e.g., includes the detection of parsing events or occlusions.

Furthermore, for the analysis of perception / sensor data a coordinated approach is chosen. Currently a format is under development to cover perception data from different sensor modalities. Through the concerted approach, the recordings in the OMEGA format can be used as reference data and can be compared to the synchronized perception data. This extends the chosen approach for scenario-based safety assurance to perception information, which is a challenging future topic of concern.

7 Acknowledgement

The research leading to these results is funded by the Federal Ministry for Economic Affairs and Energy within the project “VVM - Verification and Validation Methods for Automated Vehicles Level 4 and 5”. The authors would like to thank the consortium for the successful cooperation.

We would like to thank our project partners from the project VVM - a project of the PEGASUS family - for the fruitful discussions about this data format. We also thank Simon Schäfer, Nico Neumann, Joschka Theissen, Stefan Hausmann, Ernesto Walter and Jonas Füllmar for their help with the implementation of the data converters and with mapping the example sites.

References

- [1] W. Wachenfeld and H. Winner, “The release of autonomous vehicles”, in *Autonomous Driving: Technical, Legal and Social Aspects*, M. Maurer, J. C. Gerdes, B. Lenz, and H. Winner, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, ch. The Challenge of Releasing Fully-Automated Vehicles for Production (the “Approval-Trap”), pp. 425–449.
- [2] D. Nalic, M. Baeumler, T. Mihaly, M. Lehmann, A. Eichberger, and S. Bernsteiner, “Scenario based testing of automated driving systems: A literature survey”, *FISITA World Congress*, 2020.
- [3] Pegasus, *PEGASUS Method - An Overview*, 2019. [Online]. Available: <https://www.pegasusprojekt.de/files/tmp1/Pegasus-Abschlussveranstaltung/PEGASUS-Gesamtmethoden.pdf> (visited on Apr. 20, 2020).

- [4] A. Puetz, A. Zlocki, J. Bock, and L. Eckstein, “System validation of highly automated vehicles with a database of relevant traffic scenarios”, in *12th ITS European Congress*, 2017.
- [5] G. Bagschik, T. Menzel, A. Reschka, and M. Maurer, “Szenarien fuer Entwicklung, Absicherung und Test von automatisierten Fahrzeugen”, in *Workshop Fahrerassistenzsysteme*, Uni-DAS e.V., Ed., 2017, pp. 125–135.
- [6] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, “The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections”, in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1929–1934.
- [7] H. Weber, J. Bock, J. Klimke, *et al.*, “A framework for definition of logical scenarios for safety assurance of automated driving”, *Traffic injury prevention*, vol. 20, no. sup1, pp. 65–70, 2019.
- [8] M. Scholtes, L. Westhofen, L. R. Turner, *et al.*, “6-layer model for a structured description and categorization of urban traffic and environment”, *IEEE Access*, no. 9, pp. 59 131–59 147, 2021.
- [9] J. Hiller, F. Müller, and L. Eckstein, “Aggregation of road characteristics from online maps and evaluation of datasets”, in *2021 IEEE Intelligent Vehicle Symposium (IV)*, 2021.
- [10] T. Goelles, B. Schlager, and S. Muckenhuber, “Fault detection, isolation, identification and recovery (fdiir) methods for automotive perception sensors including a detailed literature survey for lidar”, *Sensors (Basel, Switzerland)*, vol. 20, no. 13, 2020.
- [11] G. Bagschik, T. Menzel, and M. Maurer, “Ontology based scene creation for the development of automated vehicles”, in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018.
- [12] J. Hiller, E. Svanberg, S. Koskinen, F. Bellotti, and N. Osman, “The l3pilot common data format - enabling efficient automated driving data analysis”, Jun. 2019.
- [13] Association for Standardization of Automation and Measuring Systems, *Opendrive*, 2021. [Online]. Available: <https://www.asam.net/standards/detail/opendrive/> (visited on Apr. 21, 2021).
- [14] Association for Standardization of Automation and Measuring Systems, *Openscenario*, 2021. [Online]. Available: <https://www.asam.net/standards/detail/openscenario/> (visited on Apr. 21, 2021).
- [15] F. Poggenhans, J.-H. Pauls, J. Janosovits, *et al.*, “Lanelet2: A high-definition map framework for the future of automated driving”, *21st Int. Conf. on Intelligent Transportation Systems (ITSC)*, pp. 1672–1679, 2018.
- [16] National Highway Traffic Safety Administration and U.S. Department of Transportation, *A framework for automated driving system testable cases and scenarios: Dot hs 812 623*, 2018.
- [17] K. Czarnecki, “Operational world model ontology - part 1 road structure”, 2018.
- [18] K. Czarnecki, “Operational world model ontology - part 2 road users and environmental conditions”, 2018.

- [19] M. Bäumler, L. Dziuba-Kaiser, Z. Yin, M. Lehmann, and G. Prokop, “Use information you have never observed together: Data fusion as a major step towards realistic test scenarios”, in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–8.
- [20] The HDF Group, *Welcome to the hdf5 support page!*, 2021. [Online]. Available: <https://portal.hdfgroup.org/display/HDF5/HDF5> (visited on Jul. 6, 2021).
- [21] Institute for Automotive Engineering (ika), RWTH Aachen University, *Omega format, github repository*, 2021. [Online]. Available: https://github.com/ika-rwth-aachen/omega_format/tree/master/omega_format (visited on Aug. 6, 2021).
- [22] H. Natzschka, *Straßenausstattung: Entwurf und Bautechnik*. Stuttgart: Teubner, 1997.
- [23] Bundesanstalt für Straßenwesen, *Fachthemen – Verkehrstechnik - Verkehrszeichen und Symbole*, 2021. [Online]. Available: https://www.bast.de/BAST_2017/DE/Verkehrstechnik/Fachthemen/v1-verkehrszeichen/vz-start.html (visited on Apr. 20, 2021).
- [24] H. Caesar, V. Bankiti, A. H. Lang, *et al.*, *Nuscenes: A multimodal dataset for autonomous driving*, 2019. [Online]. Available: <http://arxiv.org/pdf/1903.11027v5.pdf>.
- [25] Deutscher Wetterdienst, *CDC (climate data center)*, 2021. [Online]. Available: https://www.dwd.de/DE/klimaumwelt/cdc/cdc_node.html (visited on Apr. 20, 2021).
- [26] L. Fu, L. Thakali, T. J. Kwon, and T. Usman, “A risk-based approach to winter road surface condition classification”, *Canadian Journal of Civil Engineering*, vol. 44, no. 3, pp. 182–191, 2017.
- [27] Deutscher Wetterdienst, *Wetter- und Klimalexikon*, 2021. [Online]. Available: https://www.dwd.de/DE/service/lexikon/lexikon_node.html (visited on Jul. 15, 2021).
- [28] Deutscher Wetterdienst, *Warnkriterien*, 2020. [Online]. Available: https://www.dwd.de/DE/wetter/warnungen_aktuell/kriterien/warnkriterien.html (visited on Dec. 23, 2020).
- [29] K. Lemmer, *Neue Automobilität II: Kooperativer Straßenverkehr und Intelligente Verkehrssteuerung für die Mobilität der Zukunft (Acatech STUDIE)*, Munich, Germany: UTZ Verlag GmbH, 2019.

Scenario-based X-in-the-Loop Test for Development of Driving Automation

Felix Reisgys*, Johannes Plaum*, Andreas Schwarzhaupt* and Eric Sax**

Abstract: Even though scenario-based X-in-the-Loop testing has been evaluated in multiple research projects, there are still few application examples in series development projects. Based on current challenges in ADAS/AD testing, five requirements for scenario-based XiL testing are derived. Subsequently, we define and discuss the following three use cases along a reference ADAS/AD development process, each supplemented by respective example applications: Automated concept evaluation during Software (SW) Requirements Analysis, support of SW Unit Verification and automated validation in SW Qualification Test. All three use cases can contribute significantly to requirement fulfillment. This especially applies to use case 3.

Keywords: Driving Automation, Scenario-based testing, V&V, XiL

1 Introduction

Validation & Verification (V&V) of Advanced Driver Assistance Systems (ADAS) and Highly Automated Driving (HAD) has become more and more challenging due to the increase of testing width and depth as driving automation advances. Consequently, it is crucial to adjust and extend the existing state-of-the-art V&V processes. [1] [2] A promising way currently discussed in multiple research projects [3] [4] [5] is to combine so-called scenario-based testing with X-in-the-Loop (XiL) environments. However, only few industry applications have been presented so far. We will define a set of requirements for new test approaches and discuss the potential of scenario-based XiL test utilizing three possible use cases.

2 State of the Art

2.1 Existing Development Process Models

Current ADAS development projects deploy systems engineering and SW development process models which can be combined and adjusted depending on project needs.

The **Stage Gate Model** divides the project in several stages and connects them via gates. A stage contains activities to produce deliverables and gather information to reduce project risk. A gate represents the end of the previous stage and the beginning of

*Felix Reisgys, Johannes Plaum and Andreas Schwarzhaupt are with Daimler Truck AG (e-mail: {felix_manuel.reisgys, johannes.plaum, andreas.schwarzhaupt}@daimlertruck.com).

**Eric Sax is with Institut für Technik der Informationsverarbeitung at Karlsruher Institut für Technologie (e-mail: eric.sax@kit.edu).

the next stage and requires a decision on whether the project will be continued, paused or terminated. [6] Development processes in automotive industry may contain overlaps between stages [7] and involve samples (A-sample to D-sample) to represent gates with increasing system maturity [8].

The **V-Model** describes a well-established development process. In its initial publication, the V-Model 97 was composed of submodels project management, software development, quality assurance and configuration management [9]. Its major principle is to divide the system into multiple subsystems which are developed in parallel. Subsystem testing and integration are performed to verify that the system fulfills all requirements. The V-Process referenced in other publications such as ISO 26262 [10] and Automotive SPICE [11] is described in the submodel software development.

Agile Software Development follows an iterative and flexible process to increase efficiency and manage uncertainties better by leveraging developer competencies [12]. The Agile Manifesto published in 2001 includes four values and twelve leading principles [13]. Among different methods, Scrum has become the most commonly used process [14].

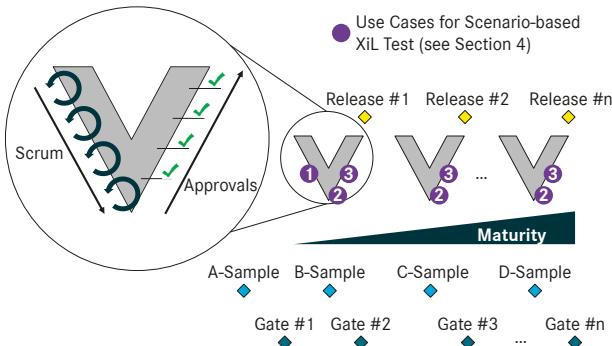


Figure 1: Custom ADAS Development Process

By combining all three models, the **custom industry process for ADAS development** as displayed in Figure 1 is derived and shall serve as reference for the following sections. On the highest level, a Stage Gate Model based process defines maturity gates for system releases. Each release is following the design and test process as defined by the V-Model. On the working level, implementation of SW units ("SW component that is not subdivided into other components" [15]) are performed based on a Scrum process.

2.2 Scenario-based Testing

As an approach applied in research projects such as PEGASUS [3], VVM [4] or SET Level [5], scenario-based testing has gained increasing relevance. Its general concept is to use a scenario as part of a test case description in order to aggregate redundant test inputs into equivalence classes [16]. A scenario is defined as a "temporal development between several scenes in a sequence of scenes" [17], where a scene is a snapshot of the ego vehicle's environment including static and dynamic objects [17]. The information aggregated in

a scenario can be structured as defined in the six-layer model: street layer (1), traffic infrastructure (2), temporal modification of layers 1 and 2 (3), movable objects (4), environment conditions (5) and digital information (6) [18]. Scenarios can be categorized as follows [19]:

- **Functional scenarios:** A consistent natural language scenario description of entities and their relations on a semantic level with a use-case dependent varying degree of detail (e.g. movement of vehicles, road topology). Descriptions may be supplemented by illustrations.
- **Logical scenarios:** A formal scenario description providing scenario parameter ranges in a state space to represent entities and their relations (e.g. velocity, distance). It is optionally possible to define distributions and constraints for parameters (e.g. $v_1 > v_2$).
- **Concrete scenarios:** A formal scenario description based on a logical scenario, additionally providing explicit scenario parameter values (e.g. $v_1 = 10m/s$).

Test cases to be reused consistently in every release (see Figure 1) can be composed from a scenario and respective evaluation criteria as well as requirements for test execution [17]. A potential approach for scenario-based testing is to generate concrete test cases from a logical scenario description by applying a parameter variation [3].

2.3 X-in-the-Loop Testing

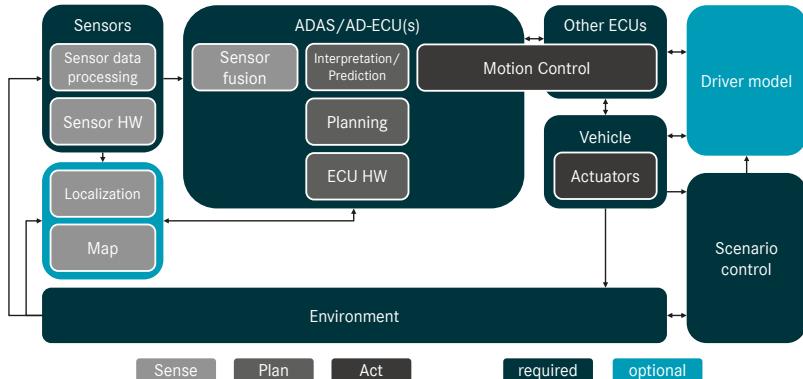


Figure 2: XiL Structure for Driving Automation Systems

By definition, X-in-the-Loop test environments stimulate a System under Test (SUT) by processing system outputs in a feedback loop [20]. They can be distinguished from open loop environments which do not contain a closed feedback loop [21]. The "X" refers to the SUT representation which can be a model, a SW or hardware (HW) for example [20]. A XiL environment for ADAS/AD testing can be structured as illustrated

in Figure 2. A driver model is only required up to SAE Level 3 to represent driver control acitivity, however is not part of the SUT. Depending on the representation (simulated, emulated or real) [22] of the systems interfacing with the SUT, the following XiL variants are commonly used [20]:

- Model-in-the-Loop (MiL): A fully virtual environment to test SUT simulation models created from a model-based design process.
- Software-in-the-Loop (SiL): A fully virtual environment to test SUT SW code.
- Hardware-in-the-Loop (HiL): A partly virtual environment to test a SUT on target HW (ECU).

3 Requirements for Future Test Methods

Even though real-world testing capacities are limited due to economical and practical reasons, it still plays a major role in the development process [10]. Nevertheless, there are multiple trends in testing of driving automation leading to a growth of the overall testing scope:

- Both new driver assistance systems and the extension of their sensor suite lead to a higher variance of scenarios to be tested [2]. Consequently, testing width increases.
- In addition, customer expectations in regards to quality impose additional requirements in the testing depth of each system.
- All systems assigned to SAE Level 3 and above need to be tested in scenarios where (temporarily) no driver is present as fall-back for the system [23]. This increases both testing width and depth.

Given the challenges, future testing methods need to meet the following requirements:

1. **Reduction of real-world testing share:** Since real-world testing capacities are limited, their overall share of all testing efforts should be reduced.
2. **Increase of test coverage:** The number of known and untested scenarios should be quantified and minimized.
3. **Independence of test validity:** Whenever simulation is used, it has to be ensured that its validity is evaluated and considered in the overall test statement.
4. **Identification of new scenarios:** The share of unknown scenarios should be minimized by extending the existing database by new logical scenarios.
5. **Efficient scenario prioritization:** Test plans and the overall test statement should consider the relevance of different scenarios.

The combination of scenario-based testing and XiL environments allows to collect and aggregate scenarios from sources such as expert knowledge, driving data or accident databases [24] in a structured way to test them in different test environments. This especially facilitates reproducibility and consistency of test execution on different platforms. Furthermore, it is possible to scale up the number of test cases and eliminate redundant scenarios, leading to an increased test efficiency.

4 Example Use Cases for Scenario-based XiL-Test

In this section, we will introduce three potential use cases to deploy scenario-based XiL testing in a development process for all degrees of driving automation. Each use case includes an example application in an industry reference development project and can be allocated to the example development process as indicated by the numbers in Figure 1. Use cases 2 and 3 can be repeated for every release. Within scenario-based XiL testing, the representation of the SUT and its environments may vary [22]. Based on the XiL structure in Figure 2, Table 1 provides an overview of the subsystem representation in each use case. Furthermore, Table 1 indicates which subsystems can be part of the SUT depending on the use case.

Subsystems	Use Cases	1 SW Requirements Analysis	2 SW Unit Verification	3 SW Qualification Test
Sensor data processing		simulated	simulated	simulated or real
Sensor HW		simulated	simulated	simulated or real
Localization		simulated	simulated	simulated or real
Map		simulated	simulated	simulated or real
Sensor fusion		simulated	simulated or real	real
Interpretation/ Prediction		simulated	real	real
Planning		simulated	real	real
ECU HW		simulated	simulated or real	simulated or real
Motion Control		simulated	real	real
Actuators		simulated	simulated	simulated

SUT optional SUT not SUT

Table 1: XiL SUT Representation by Use Case

4.1 Use Case 1: Software Requirements Analysis

Scenario-based XiL test can already support the definition of system requirements and the feasibility evaluation of a specific solution as long as there is no series SUT SW or HW available.

4.1.1 Overview

In this use case, varying concepts of the SUT based on a behavioral model are evaluated for different logical scenarios derived from the intended Operational Design Domain (ODD). Both concept parameters and ODD can be adjusted during the requirements analysis process. The SUT is represented by simulation models to be integrated into MiL or SiL environments. Scenarios can be reused along the whole subsequent development

process [25]. Overall, this use case represents an addition to existing processes in the SW Requirements Analysis.

4.1.2 Example Application

The example application supports the model-based specification of a Moving Off Information System (MOIS) on SW module (multiple units) level. A MOIS is a driver warning system for Vulnerable Road Users (VRUs) at low vehicle speeds [26]. The MOIS function concept is implemented as a Functional Mock-up Unit (FMU) to be integrated as SUT into a SiL environment containing surrounding AUTOSAR SW components (SWC) of the target ECU as well as an environment and sensor simulation. Technically, there is no feedback loop in this example application as the SUT is a warning-only system.

Scenario-based testing is used here to vary parameters of the MOIS simulation model and consequently optimize the SUT specification. In addition, it is possible to perform error injections to sensor signals to evaluate the robustness of the concept. Finally, a back-to-back proving ground test can be used to validate the SiL results and yields data for repeatable open-loop tests. Pass-fail criteria (e.g. MOIS information/warning issued) allow a concept evaluation for each logical scenario.

4.2 Use Case 2: Software Unit Verification

SW Unit Verification does not necessarily require execution of all possible test cases after each unit change by the developer, but rather focus on specific aspects of the system behavior to ensure fast verification of units.

4.2.1 Overview

Depending on whether a single or multiple SW units are tested, the SUT may vary in this use case (see Table 1). Since scenario-based testing focuses on the functional system behavior, MiL or SiL environments are sufficient in most cases, while HiL is generally feasible as well if HW samples are available. Parameters can be varied both manually or by an automated algorithm.

4.2.2 Example Application

The example application deploys a SiL test environment for the SW Unit Verification of an Advanced Emergency Braking System (AEBS) which is integrated on a dedicated ADAS ECU using AUTOSAR. Consequently, the SUT is represented by those SWCs required for the AEBS functionality. This includes interpretation and planning algorithms, as well as sensor fusion and motion control. All remaining parts of the ECU such as AUTOSAR base SW and HW are represented by simplified simulation models. The same applies to environment perception sensors, other ECUs and vehicle actuators.

Parameter variation to generate concrete scenarios is conducted via a Python script. Developers can define and adjust parameter sets manually and deterministically repeat identical scenarios multiple times as back-to-back tests. Inputs by the ego vehicle driver and other objects are fed into the simulation loop by Python modules. Test evaluation by the developer is supported by criticality metrics (e.g. Time-to-Collision) or pass/fail criteria (e.g. collision yes/no), both depending on the respective SUT and scenario.

4.3 Use Case 3: Software Qualification Test

SW Qualification Test targets an increased testing width and depth compared to use case 2. The goal is a comprehensive V&V of the SW modules in the relevant logical scenarios through a parameter variation that covers the possible parameter space.

4.3.1 Overview

A typical SUT to test an ADAS or HAD system includes subsystems sensor fusion, interpretation, prediction, planning and motion control SW (see Table 1). SiL environments leverage high scalability of scenario numbers to test a real SW module, while MiL is relevant to test single SW units. HiL environments are deployed to test the SW integrated on ECU(s), optionally even in combination with real sensor HW. A major approach for scenario-based XiL in SW Qualification Tests is parameter variation to generate concrete scenarios [27]. Furthermore, SW Qualification Tests can also be executed as regression tests in a continuous testing and integration workflow to track SUT performance.

4.3.2 Example Application

The application example deploys scenario-based testing for a risk evaluation for the release of an HAD system, as introduced in [28]. Parameter variation is applied using sampling algorithms from reliability analysis and parameter distributions derived from real world data. For the safety assessment of the concrete scenarios, different criticality metrics such as TTC or Time Headway (THW) are combined into an overall scenario criticality score.

As an example, an entering highway scenario of a heavy duty vehicle is implemented in a Python environment to generate concrete scenarios. Sampling and analysis are performed in an optimization SW. In addition to the safety metrics, additional observers (e.g. "merge performed") provide a qualitative overview of vehicle behavior in different parameter regions. The two evaluation levels enable the identification of potentially hazardous scenarios occurring in the logical scenarios when testing new functions.

5 Discussion

The contribution of each use case towards the fulfillment of the requirements defined in Section 3, is discussed in this section and summarized in Table 2.

Regarding the first requirement, both use case 1 and 2 are only used as a supplement to existing methods with low impact on the share of real-world testing. In contrast, the SW Qualification Test use case is supposed to reduce the scope of real-world testing by focusing on those scenarios with high relevance.

Through the continuous use of test cases along the development process, all use cases contribute to the second requirement. Especially the systematic test case generation in use cases 1 and 3 provides high potential to improve test coverage, while the manual scenario selection process in use case 2 leads to a "medium" rating.

Both use cases 1 and 3 are highly dependent on XiL validity [29]. This is especially critical for use case 1 as there is no real SUT test data available for XiL validation and reliability of simulation results (e.g. criticality metrics) may be limited. SW Unit

Verification does not impose high XiL validity demands as it is executed via white box tests focusing on specific SUT aspects.

Use cases 1 and 3 have a high potential to fulfill the fourth requirement. The combination with established processes like a hazard and risk analysis in use case 1 and the integration of data-driven approaches for use case 3 enable the enhancement of a scenario database [16]. In contrast, use case 2 concentrates on a fixed set of scenarios.

Especially in use cases 1 and 3, e.g. coverage-based [30] or criticality-based [28] selections facilitate efficient scenario prioritization to meet requirement 5. SW Unit Verification can also make use of these prioritization methods, however typically involves a manual selection which limits its potential [31].

Overall, all use cases are beneficial for development of both ADAS and highly automated driving. While use case 1 and 2 can be considered as an addition to existing approaches, use case 3 offers significant potential to meet the requirements defined in Section 3 (see Table 2), especially for SAE level 3 and above. The ability to manage growing ODDs as well as the consistency and traceability of scenarios along all use cases and the whole development process are major benefits of scenario-based testing.

Requirements	Use Cases	1 SW Requirements Analysis	2 SW Unit Verification	3 SW Qualification Test
1 Reduction of real-world testing share		low	low	high
2 Increase of test coverage		high	medium	high
3 Independence of test validity		low	high	medium
4 Identification of new scenarios		high	low	high
5 Efficient scenario prioritization		high	medium	high

Table 2: Degree of Requirement Fulfillment

6 Summary and Outlook

The main contribution of this work is the systematic integration of three use cases for scenario-based XiL testing into the development process of driving automation: SW Requirements Analysis, SW Unit Verification and SW Qualification Test. Each use case includes a concrete industry application example and is evaluated against previously defined requirements. Overall, all use cases can be regarded as a reasonable supplement for existing development processes. Especially the use of scenario-based XiL testing for SW Qualification Test is considered to meet the requirements to a high degree.

In future work, scenario-based XiL testing should be evaluated for additional use cases such as system application and vehicle integration testing to further facilitate continuous use of scenarios along the development process. Moreover, it is possible to derive multiple variants for each use case.

References

- [1] W. Wachenfeld, *How Stochastic can Help to Introduce Automated Driving*. PhD Thesis, TU Darmstadt, 2017.
- [2] D. A. Weitzel, *Absicherungsstrategien für Fahrerassistenzsysteme mit Umfeldwahrnehmung: Bericht zum Forschungsprojekt FE 82.0546/2012*, vol. 98 of *Berichte der Bundesanstalt für Straßenwesen Fahrzeugtechnik*. Bremen: Fachverl. NW, 2014.
- [3] DLR, “PEGASUS Abschlussbericht Gesamtprojekt,” 2020.
- [4] European Center for Information and Communication Technologies – EICT GmbH, “VVM - Projekt,” 2021. <https://www.vvm-projekt.de/projekt>.
- [5] DLR, “SET Level - Projekt,” 2021. <https://setlevel.de/projekt>.
- [6] R. G. Cooper, “Perspective: The Stage-Gate Idea-to-Launch Process—Update, What’s New, and NexGen Systems,” *Journal of Product Innovation Management*, vol. 25, no. 3, pp. 213–232, 2008.
- [7] G. Hab and R. Wagner, *Projektmanagement in der Automobilindustrie*. Wiesbaden: Springer Fachmedien Wiesbaden, 2017.
- [8] E. Sax, ed., *Automatisiertes Testen Eingebetteter Systeme in der Automobilindustrie*. München: Hanser Verlag, 1. auflage ed., 2008.
- [9] T. Binder, D. Winkler, and S. Biffl, “Quo Vadis V-Modell,” 2006.
- [10] ISO, “ISO 26262:2018: Road vehicles - Functional safety,” 2018.
- [11] VDA QMC Working Group 13 / Automotive SIG, “Automotive SPICE: Process Reference Model Process Assessment Model,” 2017.
- [12] T. Dyba and T. Dingsoyr, “What Do We Know about Agile Software Development?,” *IEEE Software*, vol. 26, no. 5, pp. 6–9, 2009.
- [13] Beck et al., “Manifest für Agile Softwareentwicklung,” 2001. <https://agilemanifesto.org/iso/de manifesto.html>.
- [14] Digital.ai, “14th Annual State of Agile Report,” 2020.
- [15] ISO/IEC/IEEE, “ISO/IEC/IEEE 24765: International Standard - Systems and software engineering - Vocabulary,” 2017.
- [16] C. King, L. Ries, J. Langner, and E. Sax, “A Taxonomy and Survey on Validation Approaches for Automated Driving Systems,” in *2020 IEEE International Symposium on Systems Engineering (ISSE)*, pp. 1–8, IEEE, 2020.
- [17] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer, “Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving,” in *2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC 2015)*, (Piscataway, NJ), pp. 982–988, IEEE, 2015.

14. Workshop Fahrerassistenz und automatisiertes Fahren

- [18] J. Bock, R. Krajewski, L. Eckstein, J. Klimke, J. Sauerbier, and A. Zlocki, "Data Basis for Scenario-Based Validation of HAD on Highways," in *27th Aachen colloquium automobile and engine technology*, 2018.
- [19] G. Bagschik, T. Menzel, A. Reschka, and M. Maurer, "Szenarien für Entwicklung, Absicherung und Test von automatisierten Fahrzeugen," in *11. Workshop Fahrerassistenzsysteme und automatisiertes Fahren* (Uni-DAS e. V., ed.), 2017.
- [20] K. v. Neumann-Cosel, *Virtual Test Drive*. PhD Thesis, TU München, 2014.
- [21] J. Bach, *Methoden und Ansätze für die Entwicklung und den Test prädiktiver Fahrzeugreglungsfunktionen*. PhD Thesis, Karlsruher Institut für Technologie, 2018.
- [22] F. Schuldt, *Ein Beitrag für den methodischen Test von automatisierten Fahrfunktionen mit virtuellen Umgebungen*. PhD Thesis, TU Braunschweig, 2017.
- [23] SAE, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," 2018.
- [24] A. Pütz, A. Zlocki, J. Bock, and L. Eckstein, "System validation of highly automated vehicles with a database of relevant traffic scenarios," in *12th ITS European Congress*, 2017.
- [25] T. Braun, L. Ries, F. Körtke, L. Turner, S. Otten, and E. Sax, "Collection of Requirements and Model-based Approach for Scenario Description," in *Proceedings of the 7th International Conference on Vehicle Technology and Intelligent Transport Systems - Volume 1: VEHITS*, pp. 634–645, SciTePress, 2021.
- [26] United Nations, "Proposal for a new UN Regulation on uniform provisions concerning the approval of motor vehicles with regard to the Moving Off Information System for the Detection of Pedestrians and Cyclists (Moving Off Information Systems (MOIS)): ECE/TRANS/WP.29/2020/122," 2020.
- [27] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer, "Survey on Scenario-Based Safety Assessment of Automated Vehicles," *IEEE Access*, vol. 8, 2020.
- [28] M. Rasch, P. Ubben, T. Most, V. Bayer, and R. Niemeier, "Safety Assessment and Uncertainty Quantification of Automated Driver Assistance Systems using Stochastic Analysis Methods," in *NAFEMS World Congress 2019*, 2019.
- [29] F. Reisgys, M. Elgharbawy, A. Schwarzaupt, and E. Sax, "Argumentation on ADAS Simulation Validity using Aleatory and Epistemic Uncertainty Estimation," in *Proceedings of the Driving Simulation Conference 2021 Europe VR* (A. Kemeny, J.-R. Chardonnet, and F. Colombet, eds.), (Munich, Germany), pp. 25–32, 2021.
- [30] Foretellix Ltd., "M-SDL Specification," 2021.
- [31] D. Baumann, R. Pfeffer, and E. Sax, "Automatic generation of critical test cases for the development of highly automated driving functions," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, pp. 1–5, 2021.

How to Derive Behavioral Requirements for Automated Driving from a Behavior-Semantic Scenery Description

Moritz Lippert* und Hermann Winner†

Abstract: For development and safety validation of highly automated vehicles, there is currently no systematic approach for the derivation of requirements. However, deriving requirements directly from the specified operational design domain (ODD) seems promising. Therefore, this paper presents an approach for deriving behavioral requirements based on the scenery as the main component of the ODD. With the help of Behavior-Semantic Scenery Description (BSSD), lane-specific routes within the scenery are derived and the corresponding resulting behavioral demands are identified. These behavioral demands are then used for the specification of behavioral requirements. Finally, the entire approach is applied as an example for a selected real-world scenery section.

Key words: automated driving, behavioral requirements, ODD, requirement derivation

1 Introduction

Ensuring a safe intended functionality is essential from the beginning of the development of highly automated vehicles (HAV) that shall move safely through road traffic [1]. The intended functionality defines the desired vehicle behavior within the operational design domain (ODD), which specifies operating conditions for a specified operational area [2]. The ODD primarily constrains the vehicle environment, but may also define vehicle behavior and vehicle states [3]. The major part of the vehicle environment is the scenery, which mainly describes roads with associated traffic infrastructure [4]. In order to ensure safe and traffic rule compliant intended functionality within the ODD, the driving behavior of HAV must be defined accordingly within the specified scenery. In this context, the driving behavior is understood as externally observable behavior and will be referred to simply as observable behavior in the following. It represents the interaction of a vehicle with other traffic participants and the remaining environment [5]. Consequently, the observable behavior is the benchmark for evaluating safe and traffic rule compliant intended functionality. Therefore, for the successful development and safety validation of HAV, requirements for the observable behavior must first be identified and defined. In the following, these requirements are referred to as behavioral requirements.

*Moritz Lippert is research associate at the Institute of Automotive Engineering Darmstadt (FZD), TU Darmstadt, Otto-Berndt-Str. 2, 64287 Darmstadt (e-mail: moritz.lippert@tu-darmstadt.de).

†Hermann Winner is professor for Automotive Engineering at TU Darmstadt, Otto-Berndt-Str. 2, 64287 Darmstadt (e-mail: hermann.winner@tu-darmstadt.de).

2 Problem Statement

The HAV development process requires approaches and methods that first identify the constraints of the observable behavior and then define behavioral requirements. In addition to basic collision-preventive behavior, such as defined in the Responsibility-Sensitive Safety (RSS) [6], it is necessary to establish the scenery dependency of behavioral requirements. Glatzki et al. [7] present an approach to a Behavior-Semantic Scenery Description (BSSD) that represents behavioral constraints using behavioral attributes. Thereby, the behavioral demands are directly derived based on the scenery as a central component of the ODD. However, the derivation of behavioral requirements based on the behavioral demands remains unclear. The authors of this paper are not aware of any approach that presents a method for deriving those essential behavioral requirements systematically starting from the ODD or scenery. Different subdomains of the HAV development chain such as the randomized scenario generation for testing [8] or the derivation of safety goals based on different intermediate steps [9, 10] are addressed, but not systematically started and executed in a holistic context. Without this context, it is not clear, on the one hand, how exactly the intended functionality is defined on the basis of the ODD and, on the other hand, how the intended functionality is tested and validated in the development process. Based on a BSSD, we therefore present an approach for a systematic derivation of behavioral requirements for HAV to potentially overcome this root specification problem.

3 Fundamentals of the BSSD

In the following, the approach for BSSD using behavioral attributes according to Glatzki et al. [7] is described as a basis for further steps. BSSD describes the scenery with the help of behavioral demands limiting the vehicle behavior. In this process, scenery and applicable traffic rules are combined so that the corresponding behavioral demands are available for each part of the scenery. In this way, the complexity of a scenery is reduced to the behaviorally relevant information. The behavioral demands are mapped using four behavioral attributes. These four behavior dimensions span the behavior space as the delimited set of legally possible behaviors. The behavior space consists of at least one atomic behavior space, which usually represents one lane section of a scenery. It is thus the smallest possible behavior space in which the behavioral demand does not change. In the following, the atomic behavior space will be called behavior space for simplicity.

The four behavioral attributes are *speed*, *boundary*, *reservation* and *overtake*. The *speed attribute* contains all behavioral demands regarding the maximum or the minimum permissible speed. The *boundary attribute* limits the behavior space not only in terms of driving behavior but also geometrically. There is one longitudinal boundary specifying the crossing demand when entering the behavior space longitudinally and two lateral boundaries specifying the crossing demand when leaving it laterally. The *reservation attribute* contains behavioral demands with respect to staying in a behavior space. Here, the demands are linked to the priority rules that apply. In this way, for example, crosswalks or intersections can be represented with the appropriate traffic participants having priority. Finally, the *overtake attribute* describes the behavioral demands regarding overtaking maneuvers.

4 Preliminary Considerations

Before the BSSD can be used to systematically derive behavioral requirements for automated driving functions, the relationship between ODD, scenery and requirements must first be shown. The simplified UML class diagram in Figure 1 is used to illustrate this relationship. The starting point for the considerations is an automated vehicle with a defined observable behavior. This behavior only exists if the vehicle also exists, so it is modeled as a composition of the vehicle. Such a vehicle is operated in a defined ODD, which demands requirements for automated driving (AD) within it. During operation, the automated vehicle moves or drives within the scenery, which constitutes the main part of the ODD, but in principle also exists without ODD definition. It therefore represents an aggregation of the ODD. The scenery can basically be described by the BSSD with respect to the behavior-relevant information. It represents the behavioral demands defined by the scenery. Even without BSSD, these behavioral demands exist, but without explicit representation. Thus, the BSSD acts as a tool to explicitly represent the scenery-based behavioral demands. These can potentially be used to derive behavioral requirements. They are a central part of the overall requirements for automated driving and explicitly define the observable behavior of the automated vehicle under consideration.

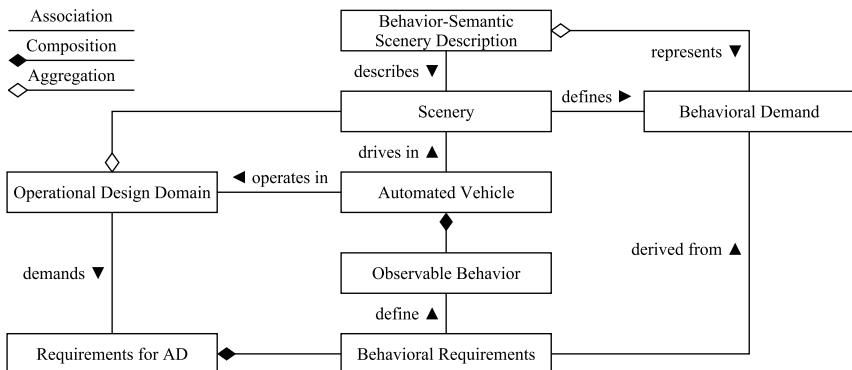


Figure 1: Relationship between ODD, scenery and requirements for AD

5 Systematic Derivation of Behavioral Requirements

Based on the identified relationship between scenery and requirements for automated driving, the behavioral requirements are derived using the BSSD. We first distinguish local from global behavioral requirements. *Global requirements* are requirements for the behavior of HAV that do not result from a specific scenery. They are scenery-independent, generally valid and have a global scope. These requirements essentially reflect the requirement of being collision-free, which is ideally satisfied always and everywhere. One approach to describing and formalizing collision avoidance requirements is RSS [6]. *Local behavioral requirements*, on the other hand, are requirements for the behavior of HAV

that result from a specific scenery. They are scenery-dependent, not generally valid, and have a local scope. These requirements result from a specific combination of scenery elements and traffic rules. Within a considered scenery or ODD, the same local behavioral requirements may apply in several places, but still the local reference remains, so that no globality can be attributed to these requirements. They are derived directly on the basis of the BSSD, in which the behavioral demands resulting from combinations of scenery elements and traffic rules are already present. In parallel, the global requirements are to be defined, which always apply everywhere. In case of conflicts of the required vehicle behavior from local and global requirements, the behavior has to be prioritized based on the global requirements. In the context of this work, only the *local behavioral requirements* are considered, which are simply called *behavioral requirements* in the following.

To systematically derive the behavioral requirements, the relevant behavioral demands of the BSSD are identified. It is not possible to speak of a general relevance, since the behavioral demands depend on the selected route within the scenery or ODD. Specifically, the behavioral demands are dependent on the behavior spaces that are part of the considered route. Since order and transitions of the navigated behavior spaces influence the resulting behavioral demands of the route, a concatenation of the behavior spaces must be considered. Thus, a directed analysis in form of a lane-specific route within the BSSD, that specifies a concatenation of behavior spaces, is always necessary to derive behavioral requirements. A part of these requirements results directly from the individual concatenated behavior spaces and another part results from the concatenation itself. Figure 2 shows these steps for a derivation of behavioral requirements in simplified form. The BSSD of a road network represents the behavior spaces and thus offers the possibility to concatenate these behavior spaces as in a lane-specific route. Within the concatenation, the transitions between the individual behavior spaces are defined. Depending on

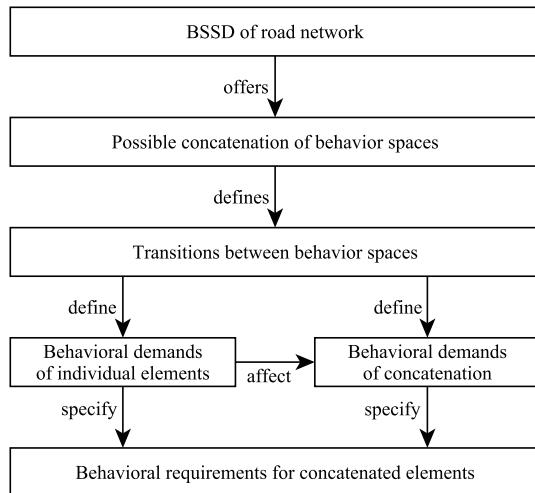


Figure 2: Derivation of behavioral requirements based on BSSD

the transitions, the relevant behavioral demands of the individual behavior spaces are defined. Additionally, the transitions between the individual behavioral demands result in behavioral demands of the concatenation. Together, the behavioral demands of the individual behavioral spaces and the concatenation specify the resulting behavioral requirements. In the following, the individual steps are explained in more detail. The BSSD and its derivation are taken as given in this work, but Glatzki and Winner [11] present an approach to derive the behavioral attributes.

5.1 Transitions within the BSSD

The transitions between the behavior spaces determine the relevance of the stored information. This means that depending on the type of transition, not every expression of a behavioral attribute has an influence on the resulting behavioral demands for the respective concatenation of behavior spaces. Before the relevance of the information is identified, the possible transitions within a BSSD must first be known.

Each behavior space is represented by the four behavioral attributes *speed*, *boundary*, *reservation*, and *overtake*. Only the *boundary attribute*, in addition to limiting the possible behavior of an HAV, also limits the physical dimensions of the behavior space itself. Consequently, this attribute gives the abstract behavior space its physical geometric shape. The behavior spaces can be simplified as rectangles as shown in Figure 3. For clarity, spacings are drawn between the individual behavior spaces that do not exist in a real scenery. In reality, the behavior spaces correspond to lane sections, for example. Due to the directionality of the behavioral demands, a driving direction must be assumed for the consideration of the behavioral spaces (here from left to right). Since only the geometric shape is relevant for a consideration of the possible transitions between several behavior spaces, only the *boundary attribute* is visualized accordingly. A behavior space is always bounded by two lateral boundaries (right and left) and a longitudinal boundary. The longitudinal boundary is located at the beginning of the behavior space according to the

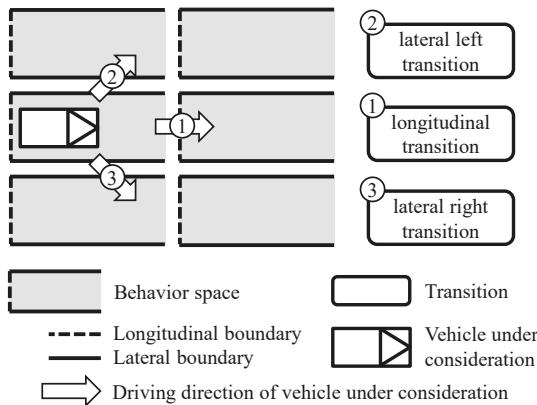


Figure 3: Transitions within BSSD

direction of travel, i.e. at the left edge of the behavior space in the figure shown.

In principle, there are only two states for a considered vehicle within the BSSD. The vehicle may or may not be completely within a behavior space. If the vehicle is in more than one behavior space, we speak of overlapping. Overlapping is always initiated by a transition, which describes the moment when the vehicle crosses a boundary of the behavior space. After a transition, a vehicle can basically maintain the state of overlapping, for example by driving in two lanes at the same time, or return completely to a single behavior space. Since a behavior space always consists of one longitudinal and two lateral boundaries, this results in three basic transitions: *longitudinal*, *lateral left* and *lateral right*.

In the figure, these transitions are shown in isolation. In reality, however, the transitions are not always isolated, since the vehicle has a spatial extension and cannot be modeled as a point. Considering a lane change, the vehicle could make an additional longitudinal transition during a lateral transition, resulting in overlapping four behavior spaces. In this case, there would actually be three boundaries involved: one lateral and two longitudinal. Furthermore, a behavior space could be geometrically very short or very narrow, so that a vehicle performs several lateral or longitudinal transitions simultaneously while passing through it (e.g., in the area of a crosswalk). This problem of overlapping more than two behavior spaces will be neglected in the context of this work, since it has no influence on the derivation mechanism of the behavioral requirements. Thus, an explanation of the rule mechanism for resolving these ambiguities is omitted. As a result, a vehicle is assumed to be a point for the consideration of transitions.

5.2 Behavioral Demands of Individual Elements

In order to obtain the relevant, applicable behavioral demands for a concatenation of behavior spaces, there must be a unique assignment of behavioral demands for each possible position of a vehicle within this concatenation. If a route is considered along concatenated behavior spaces, the vehicle necessarily passes through the associated transitions. Consequently, it is necessary to identify which behavioral demands of all concatenated behavior spaces are relevant and valid for this considered route. For identification, an arbitrary concatenation of behavior spaces, in the following also referred to as elements, is traversed. A necessary prerequisite is that the concatenated elements are direct neighbors, so that a direct transition is possible.

Let M_{BS} be the set of all behavior spaces within a BSSD. We define the concatenation of $n_C \in \mathbb{N}$ behavior spaces as the sequence $C = (E_i)_{i=1,2,\dots,n_C}$ while $E_i \in M_{BS}$ and $(E_j, E_{j+1})_{j=1,2,\dots,n_C-1}$ are pairs of direct neighbors with the transition $T_{j,j+1}$ (section 5.1) between them. Let D_i be the set of relevant behavioral demands of element E_i .

Figure 4 shows an example of a concatenation C with $n_C = 6$ elements, corresponding transitions $T_{j,j+1}$ and resulting relevant behavioral demands D_i of the individual elements to visualize the following considerations. The example section of a BSSD road network with 12 behavior spaces could represent a three-lane one way road. However, the concrete scenery or BSSD representation is not important for the considerations. Here, in particular, the relation of the different terms should become clear in order to build up an overall understanding.

Starting in the first element E_1 of concatenation C , the behavioral demands of the *speed*, *reservation*, and *overtake* attributes of this element are relevant. In general, the

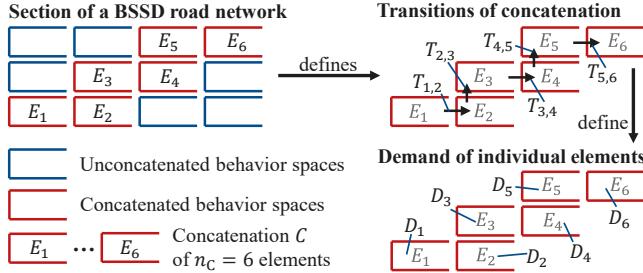


Figure 4: Visualization of an example BSSD road network section

first element is a special case because it is the only element of the concatenation that is not reached by a transition when traveling from first element E_1 to last element E_n with $n = n_C$ (E_6 in the example). For this reason, no behavioral demand results from a transition, so the behavioral demands of the *boundary attribute* are irrelevant.

Proceeding from the first element, the remaining elements are now traversed in sequence. In doing so, the relevant behavioral demands are always determined for the next element, since equivalently to real driving, the demands must be known before entering the next element. For all further elements E_{j+1} , the relevant behavioral demands D_{j+1} depend on the transition $T_{j,j+1}$. This dependency is based on the definition of the BSSD, which links crossing demands to the longitudinal entry and to the lateral exit of a behavior space (see Section 3). Accordingly, for a longitudinal transition from E_j to E_{j+1} (e.g., $T_{1,2}$ from E_1 to E_2), the longitudinal boundary of element E_{j+1} is relevant. For a lateral transition from E_j to E_{j+1} (e.g., $T_{2,3}$ from E_2 to E_3), in contrast, the lateral boundary of element E_j is relevant. Since a transition may occur to the right as well as to the left side, the two directions must be additionally distinguished accordingly. Regardless of the transition, the inherent attributes *speed*, *reservation* and *boundary* are relevant for each element E_{j+1} . Table 1 summarizes the relevant behavioral demands for all elements E_{j+1} .

Table 1: Relevant behavioral demands with respect to the transitions

Relevant Behavioral Demand D_{j+1} for Element E_{j+1} depending on the Transition $T_{j,j+1}$						
Behavioral Demand	$T_{j,j+1} = \text{longitudinal}$		$T_{j,j+1} = \text{lateral right}$		$T_{j,j+1} = \text{lateral left}$	
	E_j	E_{j+1}	E_j	E_{j+1}	E_j	E_{j+1}
Speed		x		x		x
Boundary	Longitudinal		x			
	Lateral Right			x		
	Lateral Left				x	
Reservation		x		x		x
Overtake		x		x		x

5.3 Behavioral Demands of Concatenation

With the help of the identified relevant behavioral demands for driving through a concatenation of behavior spaces, the derivation of the associated behavioral demands becomes possible. For each element E_i of a concatenation C , the behavioral demands for driving in this element depending on the entry transition are known. The behavioral demands can be semantically transformed in order to specify the associated behavioral requirements in a further step. But are these behavioral requirements alone sufficient? So far, only the resulting behavioral demands of individual behavior spaces have been considered. A simple example shows that the behavioral demands of each individual behavior space might not be sufficient in any case. We consider a transition from E_j to E_{j+1} . In this case, the behavioral demands change from D_j to D_{j+1} . Only the change in behavioral demands of the *speed attribute* is considered in this example. Given D_j demands a higher speed limit than D_{j+1} , a vehicle is allowed to drive faster in E_j than in E_{j+1} . However, the current consideration does not provide more information. A vehicle would exceed the speed limit when entering E_{j+1} due to kinematic dependencies, since it would have had to decelerate before entering this element in order to comply with this limit. Thus, it can be concluded that an additional demand for speed adjustment before entering the element is necessary in this case.

For the derivation of the behavioral requirements, therefore, an additional consideration of the concatenated sets of behavioral demands D_j and D_{j+1} is necessary. This concatenation may result in an additional set of behavioral demands $D_{j,j+1}$. The set of the total resulting behavioral demands for all successor elements E_{j+1} thus results in $D_{res,j+1} = D_{j+1} \cup D_{j,j+1}$. For the first Element E_1 of a concatenation, simply $D_{res,1} = D_1$ holds due to the lack of transition. Figure 5 shows the relationship of the different behavioral demands based on the introduced example of Figure 4. For clarity, the demands are shown directly in the elements of the concatenation to which they apply.

To determine the behavioral demands $D_{j,j+1}$, in particular, the kinematic dependencies of a moving vehicle must be considered. These dependencies lead to the fact that behavioral demands with respect to allowed speeds or accelerations are possible from all four behavioral attributes. For example, if a behavior space is externally-reserved due to the *reservation attribute*, a vehicle must give priority when entering this space. This behavioral demand directly affects the demanded speed profile when entering this space.

In order to derive the behavioral demands $D_{j,j+1}$ holistically, all possible combinations of the concatenated behavioral demands of D_j and D_{j+1} must be explored. Although the behavioral-level abstraction within the BSSD allows for a large reduction in the necessary information compared to other approaches, the goal is to minimize the parameter space for an analysis even in this combination approach. Since the BSSD classifies the behavioral demands into four independent behavioral attributes, there is no need to combine the

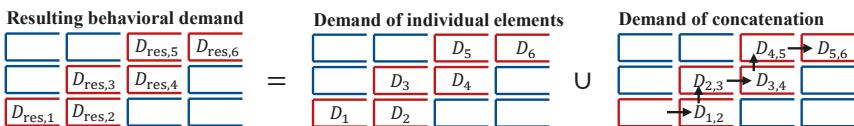


Figure 5: Visualization of relationship between behavioral demands $D_{res,i}$, D_i and $D_{j,j+1}$

attributes due to the independency among them. This means that combining behavioral attributes with different characteristics does not create new behavioral demands. For example, it has no influence on the behavioral demands with respect to the *speed attribute* whether a combination additionally changes the *reservation attribute*. As a result, a demand for a maximum allowed speed profile may arise in each case, but independently from each other. In case of different arising speed demands, a minimum prioritization depending on a ordinal scale would be possible. However, the scales of the different behavioral demands are not part of this work.

Due to the independency, the combination of the behavioral demands is done separately for each behavioral attribute. The behavioral attributes not considered are neglected. For each combination D_j and D_{j+1} it is checked whether further behavioral demands $D_{j,j+1}$ result from this specific concatenation. In the following, a specific combination of behavioral demands is considered for three behavioral attributes by way of example.

Speed: The example at the beginning of this section is considered again. Thus, D_j demands a higher speed limit than D_{j+1} , so that a vehicle must drive slower in E_{j+1} than in E_j . This results in the behavioral demand $D_{j,j+1}$, which demands a speed adjustment to the changed speed limit before entering E_{j+1} . In contrast, given D_j demands a lower speed limit than D_{j+1} would not result in a new behavioral demand $D_{j,j+1}$, since a vehicle is not legally forced to increase speed in the absence of a minimum speed constraint.

Reservation: let E_j be own-reserved and E_{j+1} be externally-reserved. Accordingly, D_{j+1} requires that certain traffic participants with certain directions of arrival who have reservation rights to the area E_{j+1} are not obstructed. Obviously, an obstruction can only occur if other traffic participants with reservation claim are actually nearby when entering this area. If this is the case, these traffic participants must therefore be guaranteed to continue their driving as unhindered as possible. For this purpose it is important to indicate to these traffic participants that the own waiting obligation is fulfilled (see also German road traffic regulations [12]). Another behavioral demand $D_{j,j+1}$ is therefore to indicate in advance that the respective traffic participants will be given priority.

Overtake: If D_j is a permission and D_{j+1} is a prohibition to overtake, then initially there is only the demand that overtaking is not allowed in E_{j+1} . But what if a vehicle starts an overtaking maneuver beforehand and does not finish it prior to entering E_{j+1} ? In this case, the behavioral demand would clearly be violated, so that another demand $D_{j,j+1}$ is needed. $D_{j,j+1}$ states that a potential overtaking maneuver is to be completed before entering area E_{j+1} .

5.4 Resulting Behavioral Requirements

After deriving the resulting behavioral demands $D_{\text{res},i}$ for all elements E_i of a concatenation C , the specification of the behavioral requirements is possible. The specification essentially consists of a semantic processing of the derived behavioral demands. This step is necessary in order to use the behavioral requirements in the context of automated driving. This means, on the one hand, that the behavioral requirements are used to precisely specify the observable behavior and, on the other hand, that the remaining requirements for the automated driving task are inferred based on the behavioral requirements. In particular, it must be ensured that the set of possible behaviors of a vehicle is restricted only as far as necessary. For this purpose, the requirements should be formulated as

negated as possible, since in this way, as with the behavior space itself, only the limit of the permitted behavior is defined.

The following simple example of an imprecise formulation shows an intervention in the behavioral planning of a vehicle that clearly goes beyond the definition of the behavioral limit. If the entry into a behavior space is controlled by a traffic light, then the crossing demand of the *boundary attribute* results in "no red light". Thus, the vehicle is not allowed to cross the associated boundary when the traffic signal is red. A restrictive requirement would be: *The vehicle shall stop at the boundary when the traffic light is red*. With this requirement, the vehicle is forced to stop, although it could approach the traffic light just as foresightedly in order not to have to stop. Further example requirements are additionally shown in the following section.

6 Real-World Application Example

To demonstrate the presented method, a real scenery from Darmstadt (Germany) is considered in Figure 6. The aerial view shows a T-intersection with a multi-lane one-way road running from left to right and a two-lane side road with two-way traffic. An abstract representation of the BSSD is shown as the second layer. Here, the dark frames show part of the behavior spaces on this scenery section, which are marked with capital letters. Since there is always one behavior space per direction of travel (even against the one-way street) and these can also overlap in intersection areas, not all behavior spaces are shown for clarity (including the behavior space of the restricted area). The present segmentation of the behavioral spaces is based on changes in the behavioral demands in the longitudinal direction. If there are changed behavioral demands due to the scenery, a new segment is created. The behavior spaces are present in the BSSD unconcatenated, so that initially only information about the relative position of the behavior spaces to each other is known. Therefore, a possible concatenation of the behavior spaces is represented as the third level. The concatenation follows the path drawn in blue, which can potentially be followed by a vehicle. Non-concatenated behavior spaces are shown slightly transparent compared to the concatenated ones.

Transitions $T_{i-1,i}$, behavioral demands of individual (D_i) and concatenated elements ($D_{i-1,i}$) as well as resulting requirements of E_i are shown in the attached table. Since one column is considered for each element E_i in the table, the relationship of neighboring elements $(E_j, E_{j+1})_{j=1,2,\dots,n_C-1}$ from previous sections is reformulated into the mathematical equivalent $(E_{i-1}, E_i)_{i=2,3,\dots,n_C}$ to ensure a formally correct representation. Consequently, transition $T_{i-1,i}$ and behavioral demands $D_{i-1,i}$ of concatenation are not defined for $i = 1$.

As shown in the figure, the concatenation $C = (I, J, K, H, E)$ is considered, which consists of $n_C = 5$ elements E_i with $i = 1, 2, \dots, n_C$ (number in white boxes corresponds to i). When this concatenation is followed, first a right turn is made coming from the minor road, and then a lane change to the left into the middle lane. According to these transitions, the behavioral demands D_i of the individual elements result. Along the concatenation, the demand of the *speed attribute* (S) does not change, so that a maximum allowed speed of 30 km/h applies to all elements E_i . In the first element $E_1 = I$ there is no behavioral demand based on the *boundary attribute* (B) because there is no transition. In the third element $E_3 = K$, there is a requirement that the vehicle stops before entering. The cause of this demand is a stop sign with associated stop line in the scenery. The

	E_i	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
	$T_{i-1,i}$					
	D_i	S: 30 km/h R: own O: yes	S: 30 km/h B: allowed R: own O: yes	S: 30 km/h B: stop R: ext. O: yes	S: 30 km/h B: allowed R: own O: yes	S: 30 km/h B: allowed R: own O: yes
	$D_{l-1,i}$	X	-	R: indication of giving priority	-	R: yield to TP with same reservation
Requirements of E_i	S: The vehicle shall not exceed a speed of 30 km/h.	X	X	X	X	X
	B: The vehicle shall stop at the longitudinal boundary before proceeding.			X		
	R: The vehicle shall not obstruct traffic participants with reservation entitlement for element E_i .			X		X
	R: The vehicle shall indicate in advance that it yields to traffic participants with reservation entitlement for element E_i .			X		X
	O: The vehicle shall not overtake.					

Figure 6: Behavioral Requirements of concatenated elements E_i based on behavioral demands of individual elements (D_i) and concatenation ($D_{i-1,i}$) depending on transitions $T_{i-1,i}$ (Aerial image © Orthophoto Vermessungsamt Darmstadt 2021)

other elements E_i have no demands concerning the entry (Crossing condition: allowed). Regarding the *reservation attribute* (R), there are no restrictive behavioral demands for the elements that are own-reserved (own), in these areas from an individual point of view no priority is to be given. Only element E_3 as representation of the intersection area has an external reservation (ext.), so that certain other traffic participants shall not be obstructed. For reasons of clarity, we do not specify the type of traffic participant and the direction of arrival in this example. Overtaking is allowed in every element E_i , so there is no restriction on behavior based on the *overtake attribute* (O).

From the transitions between the individual behavioral demands D_i , the behavioral demands $D_{i-1,i}$ from concatenation are derived. In element $E_3 = K$, an additional behavioral demand of the *reservation attribute* results. Accordingly, it must be indicated that priority is given to potentially occurring traffic participants who are entitled to reservation. In element $E_5 = E$, the *reservation attribute* additionally requires that traffic participants with the same reservation entitlement must not be obstructed. This means that a lane change from $E_4 = H$ to $E_5 = E$ shall only take place if traffic in the same direction of travel is not obstructed in the process. Due to this demand, the same requirement as in $D_{2,3}$ additionally applies, since it must also be indicated here that priority is given (not shown in table). For the purpose of clarity, we again omit a representation of the traffic participant type and direction of arrival of the traffic participants entitled to reservation.

For the resulting behavioral demands $D_{res,i} = D_i \cup D_{i-1,i}$ of the concatenated elements E_i , the behavioral requirements result as shown in the lower half of the figure. Although there is no restrictive requirement of the *overtake attribute* for the elements E_i , it is still instantiated for completeness. The distribution of requirements shows that the intersection entry and lane change have significantly more behavioral requirements than the

remaining elements of the concatenation. Consideration of traffic participant type and direction of arrival of the reservation-entitled traffic participants would further increase the complexity of the requirements.

7 Conclusion and Outlook

This work first identified the lack of a systematic and holistic derivation of requirements for HAV. An analysis demonstrated that the scenery as the main component of the ODD is a promising basis for such an approach. Here, the scenery is described using behavioral demands based on four behavioral attributes (BSSD) [7]. We identified possible transitions of HAV within the BSSD. They served as the basis for identifying the behavioral demands of the individual behavior spaces and of the concatenation itself, which were used to derive behavioral requirements. The entire approach was applied as an example for a selected real-world scenery section. We demonstrated that it is possible to derive behavioral requirements directly from the scenery considering a concatenation of behavior spaces representing a lane-specific route.

In further work, we will show the decomposition of the behavioral requirements to other functional levels of the automated driving task. Using these requirements based on concatenated behavior spaces, we will conceptualize and implement capability-based routing in order to archive a dynamic ODD. In this approach, driving capabilities will be matched with route section requirements. In order to achieve that, we also work on a holistic representation of the BSSD so that arbitrary road networks can be described and modeled in maps. The overall approach potentially enables a route-wise development and validation of HAV for a specific ODD and thus reduced validation effort.

8 Acknowledgement

This research is accomplished within the project “UNICARagil” (FKZ 16EMO0286). We acknowledge the financial support for the project by the Federal Ministry of Education and Research of Germany (BMBF).

References

- [1] "ISO/PAS-Norm 21448:2019 - Road vehicles - Safety of the intended functionality", International Organization for Standardization (ISO), 2019.
- [2] "Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems", J3016, SAE International (SAE), 2014.
- [3] K. Czarnecki, "Operational Design Domain for Automated Driving Systems - Taxonomy of Basic Terms", 2018.
- [4] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt and M. Maurer, "Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving", 2015 IEEE 18th International Conference on Intelligent Transportation Systems, 2015, pp. 982-988, doi: 10.1109/ITSC.2015.164.

- [5] M. Nolte et al., "Towards a skill- and ability-based development process for self-aware automated road vehicles", in IEEE ITSC 2017: 20th International Conference on Intelligent Transportation Systems: Mielparque Yokohama in Yokohama, Kanagawa, Japan, October 16-19, 2017, Piscataway, NJ: IEEE, 2017, pp. 1–6.
- [6] S. Shalev-Shwartz, S. Shammah and A. Shashua, "On a Formal Model of Safe and Scalable Self-driving Cars", arXiv preprint arXiv:1708.06374, 2017.
- [7] F. Glatzki, M. Lippert and H. Winner, "Behavioral Attributes for a Behavior-Semantic Scenery Description (BSSD) for the Development of Automated Driving Functions", 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), 2021, pp. 667-672, doi: 10.1109/ITSC48978.2021.9564892.
- [8] G. Bagschik, T. Menzel, C. Körner, and M. Maurer, "Wissensbasierte Szenariengenerierung für Betriebsszenarien auf deutschen Autobahnen", in *Workshop Fahrerassistenzsysteme und automatisiertes Fahren*, Vol. 12. Walting im Altmühltafel: Uni-DAS e.V., 2018.
- [9] T. Stolte, G. Bagschik, A. Reschka and M. Maurer, "Hazard analysis and risk assessment for an automated unmanned protective vehicle", 2017 IEEE Intelligent Vehicles Symposium (IV), 2017, pp. 1848-1855, doi: 10.1109/IVS.2017.7995974.
- [10] T. Stolte, G. Bagschik and M. Maurer, "Safety goals and functional safety requirements for actuation systems of automated vehicles", 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), 2016, pp. 2191-2198, doi: 10.1109/ITSC.2016.7795910.
- [11] F. Glatzki and H. Winner, "Inferenz von Verhaltensattributen der verhaltenssemantischen Szeneriebeschreibung für die Entwicklung automatisierter Fahrfunktionen", in *Workshop Fahrerassistenzsysteme und automatisiertes Fahren*. Vol. 14. Berkheim: Uni-DAS e.V., 2022.
- [12] "Straßenverkehrs-Ordnung: StVO", Bundesministerium der Justiz und für den Verbraucherschutz, 2013.

Uni-DAS

Herausgeber

Klaus Bengler

Klaus Dietmayer

Lutz Eckstein

Markus Maurer

Christoph Stiller

Hermann Winner