A Fail-safe Architecture for Automated Driving

Sebastian vom Dorff[1,3], Bert Böddeker[2], Maximilian Kneissl[1], Martin Fränzle[3]

***Abstract*—The development of autonomous vehicles has gained a rapid pace. Along with the promising possibilities of such automated systems, the question of how to ensure their safety arises. With increasing levels of automation the need for fail-operational systems, not relying on a back-up driver, poses new challenges in system design. In this paper we propose a lightweight architecture addressing the challenge of a verifiable, fail-safe safety implementation for trajectory planning. It offers a distributed design and the ability to comply with the requirements of ISO26262, while avoiding an overly redundant set-up. Furthermore, we show an example with low-level prediction models applied to a real world situation.**

I. Introduction

The trend to automated driving keeps evolving and climbs the next levels on the SAE automation scale [1]. While many current systems are classified as level 1 and 2, first level 3 applications emerge and the logical next step is a transition to level 4 and upwards [2]. The essential change in progressing to these higher levels of automation is the absence of the driver from the control loop.

Along with this disruptive change, a new safety challenge emerges, the safety of the intended functionality. In contrast to the established functional safety, it considers the safety of a functionality in absence of any malfunction. While first attempts to standardize methodologies for this task have been made, the ISO PAS 21448 only considers automation up to level 2 [3]. The question how to ensure safe behavior of an automated system that acts without a human supervisor remains unsolved. AI based algorithms, many of them based on artificial neural networks, are hard to prove to be safe. By feeding collected data to the neural network, it begins to mimic the same behavior as given in the training data. It is neither possible to guarantee safe training data with reasonable effort as shown in [4], [5], nor is it possible to prove the resulting learned behavior being free from glitches [6].

Rule-based approaches can help creating a safety framework by formalizing traffic rules for machines. The challenges here occur when rules leave space for subjective interpretation and context sensitive reasoning is required. Especially urban scenarios, such as parking grounds, present a particular challenge since many rule based assumptions are not valid anymore on closed parking grounds. The right of way boils down to mutual respect and basically every action of other traffic participants has to be expected.

Even though the task of functional safety is understood and widely standardized in its methodologies (as in the ISO 26262), the complexity that comes along with the new driverless control loop increases drastically [7]. Until today, most systems just had to be fail-silent, handing over control to the driver in the event of failure. From level 4 on, the system needs to be fail-operational, at least being able to reach a safe state without external help.

The combination of these new challenges in the automotive sector underline the necessity to think of new, tailored solutions, exploiting the advantages of the specific domain — in contrast to other branches e.g. aerospace — whilst keeping in mind the peculiarity of autonomous systems. Our approach contributes a system architecture that provides a framework to tackle these challenges by natively supporting a policy for safe behavior and offering a modular, redundancy-free design that can be distributed, thereby keeping the implementation customizable.

II. Related work

Designing fail-operational systems while fulfilling automotive requirements of low price, energy consumption and build space is challenging. While applications in the aerospace sector demand a high integrity level, these systems still rely on a human fall-back for decision tasks and feature highly redundant designs contradicting the before mentioned requirements [8].

Triple-modular redundancy seems to be a unified solution for fail-operational system, but creates a significant overhead due to the threefold implementation. Resulting in at least tripling the costs, energy consumption, and build space requirements, an actuator-monitor based design is more desirable as mentioned in [9].

Classical actuator-monitor approaches are fail-silent systems, thus requiring a back-up driver to take over in case of a system fault. They are not fail-operational and therefore not fail-safe in a level 4 application.

Additionally to these functional safety requirements, the safety of the intended functionality is another task. The specification and implementation of requirements is a challenge on its own [10].

This work has been conducted within the ENABLE-S3 project that has received funding from the ECSEL Joint Undertaking under grant agreement No 692455. This joint undertaking receives support from the European Union's HORIZON 2020 research and innovation programme and Austria, Denmark, Germany, Finland, Czech Republic, Italy, Spain, Portugal, Poland, Ireland, Belgium, France, Netherlands, United Kingdom, Slovakia, Norway.

[1]S. vom Dorff and M. Kneissl are with the Corporate R&D department of DENSO Automotive Deutschland GmbH, Freisinger Str. 21-23, 85386 Eching, Germany {s.vomdorff,m.kneissl}@denso-auto.de

[2]B. Böddeker is with Autonomous Intelligent Driving GmbH, , Ungererstr. 69, 80805 München, Germany bert.boeddeker@aid-driving.eu

[3]M. Fränzle and S. vom Dorff are with the Carl von Ossietzky University, Department of Computing Science, 26111 Oldenburg, Germany fraenzle@informatik.uni-oldenburg.de

The validation task in classical means by statistical evaluation was shown to be unfeasible and requires on-road testing in ranges of billion of miles [4], [5], [11].

Within this field of problems, the verification is another issue. Machine learning based approaches appear as black boxes that can only be tested statistically as well, with exempt of extracting partial information and thereby transforming elements into white-box algorithms [6].

A counter measure to this issue is the usage of rule-based approaches such as [11] and [12]. While it is still challenging to validate the given rules against real world requirements, expert knowledge and human experience can be used to increase reliability of these rules. An attempt to apply the classical methods, such as "Failure Mode and Effects Analysis (FMEA)", from ISO26262 to this new field has been made in the ENABLE-S3 project, resulting in a range of scenario based safety goals for the automated valet parking use-case [13]. Besides, the verification of the implementation becomes easier by using sophisticated test-systems, also being researched within that project.

A third issue is the extrapolation of data. The more sophisticated prediction models get, the more uncertainty is included if they are correct. The survey in [14] displays the trade-off in this discipline.

The work of [15] provides a likelihood distribution, providing a reliable superset of the actual movement of other traffic participants, while reducing the total set by removing the unthinkable if not impossible elements. This extends the purely formalized, rule-based design described in [12] to a more naturalistic level of traffic participants which are more prone to move within less strict patterns. These approaches have been formally validated and shown to be feasible for certain traffic situations as shown in [16].

III. Baseline Approaches

In the following subsections we give a brief overview of two existing approaches to underline the difference to our approach.

A. Emergency Braking

The simplest approach to cope with unsafe states is to trigger an emergency full-stop braking maneuver. While this method would be sufficient for the majority of critical events, it could still lead to plenty hazardous situations. The first problem is given by the fact that the stopping distance is covered blindly. It is calculated by

$$d_{stop} = t_{dead} * v_{vehicle} + \frac{1}{2} * v_{vehicle}^2 / a_{decel}, \qquad (1)$$

where t_{dead} is the dead time until the actual braking is performed, $v_{vehicle}$ is the vehicle's initial speed and a_{decel} is the maximum vehicle's deceleration.

Adding a fault detection time of $50\ ms$ and a delay time of $150\ ms$ from [17] leads to a dead time of $200\ ms$ in total. An average deceleration for an emergency braking maneuver of $8.6\ m/s^2$ can be assumed from [18] and a speed of $50\ km/h$ as the most common speed limit in European cities. With 1

Fig. 1. Emergency braking might alter the trajectory in a dangerous way.

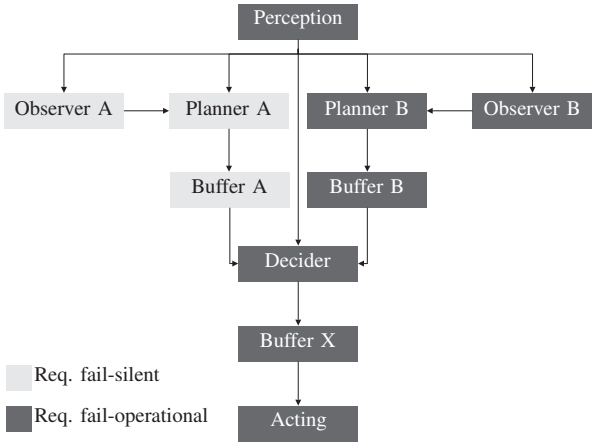

Fig. 2. A potential architecture for an approach with alternative trajectories.

this calculates to about $14\ m$ of blindly covered distance. This is not acceptable for fail-safe behavior.

Additionally, a braking maneuver can cause instabilities in driving dynamics and alter the intended trajectory. This behavior is depicted in Fig. 1. While Trajectory 1 has been judged to be safe, an occurring hazard prevents Trajectory 2 from being executed. Due to the physical limitations of tire friction, the vehicle will leave the driveable area and either leave the road or steer into oncoming traffic.

B. Alternative Trajectory Planning

A more sophisticated approach shown in [16], creates an alternative trajectory in case the first intended path appears to be not feasible. This approach shows benefits especially in high speed applications, i.e. highway scenarios. A significant disadvantage of this approach arises from the architectural implementation.

Fig. 2 pictures an architecture for such an approach according to our interpretation. It shows the necessity of at least doubling all major components for the planning phase. In addition to that, the decider judging if a secondary trajectory "B" shall be triggered is a single point of failure. Furthermore, the whole chain from perception to acting needs to be designed fail-operational as it is mandatory to provide a new trajectory and decide about its usage at any given point in time. This requires a diverse and redundant design, therefore interfering with the before mentioned demands in automotive design.

IV. Lightweight Approach

In this section we introduce our new architecture. By using model predictive control (MPC) it enhances the emergency

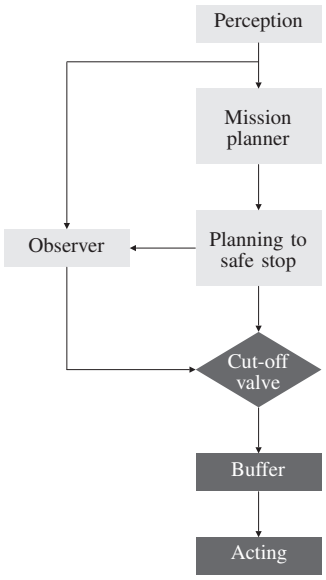

Fig. 3. The general architecture of the lightweight concept. Only darker colored modules need to be fail-operational.

braking approach baseline to mitigate its shortcomings while sustaining its simplicity.

A. General architecture

The general architecture shown in Fig. 3 consists of a system decomposed into separate functional blocks.

First of all, the environment data has to be acquired. The concept is agnostic to the implementation of the environment perception. This follows the hierarchical paradigm of sensing, planning, and acting in robotics [19]. The environment data, assumed as ground-truth, are then transferred to a mission planner and the safety observer. The mission planner creates the macroscopic path from the starting point to the designated destination. The following trajectory to safe stop planner includes the speed profile and takes environment data into account. The provided mission plan is chopped into short-time sections with a suitable speed profile, always followed by a full-stop maneuver. The resulting short-time trajectories are then passed on to the trajectory buffer and the safety observer. The safety observer checks whether the trajectory provided by the planner is free from unreasonable risk until the full-stop. Therefore it has to extrapolate the current snapshot of environment data with suitable models, like movement models of other traffic participants, and a policy what is considered safe behavior. The prediction of the environment and the effects of the own action within the near future is a major difference to a simple emergency braking approach. If the trajectory section is judged to be safe, it is passed on to the buffer and will be executed by the underlying actuators, i.e., the vehicle control. In case the safety observer assesses the provided trajectory as unsafe, it will disable the forward path to the buffer, thus preventing the trajectory being transmitted. In that case, the buffer runs empty, the vehicle will perform the last operations stored in the last successful iteration and finally comes to a stop which has previously been judged safe. The system is therefore fail-operational for the time needed to reach a minimal-risk state. Redundancies are avoided by predicting a short time horizon and therefore bridging the small gap to a safe stop with pre-emptive actionplans that are inherently executed in case of system failure or non-compliance with behavior requirements.

TABLE I
VALID DECOMPOSITIONS OF ASIL C.

Target level	First decomposition component	Second decomposition component
ASIL C	ASIL A(C)	ASIL B(C)
ASIL C	ASIL C(C)	QM(C)

B. ASIL decomposition

The architecture consists of a number of independent elements. This allows the required ASIL (Automotive Safety Integrity Levels) to be decomposed and to assign the resulting (decomposed) ASILs to the independent components [7]. ASILs range from QM as lowest, followed by ASIL A up to ASIL D, the highest level. The levels differ for various safety goals. As an example the top level safety goal "The vehicle shall only follow trajectories that do not hit pedestrians. <ASIL C>" from [13] can be used for the automated valet parking.

To comply with the requirements of ISO26262 the ASIL has to be fulfilled by the whole functional chain of the system. Therefore the environment data has to be ASIL C in the beginning. Since the trajectory planning to a safe stop on the one side and the safety observer on the other side are functionally independent and diversely implemented, the safety goal can be decomposed. Valid decompositions can be seen in Table I. The safety observer is meant to be the simpler implementation that only has to focus on safety. The trajectory planner in comparison has to cope with more complex tasks. Due to this circumstances, it makes sense to assign the higher ASIL to the safety observer. A potential decomposition could be done in the following way: "The trajectory planning shall only plan trajectories that do not collide with pedestrians <QM(C)>" "The safety observer shall reject trajectories that collide with pedestrians <ASIL C(C)>"

Another advantage of this architecture is the immanent availability of valid, safe trajectories within the buffer. Therefore, there is no availability requirement for the perception, only for a fault detection that indicates to the safety observer to reject new trajectories. This removes the requirement of fail-operational design from several modules and communication paths. This is depicted by a lighter coloring of the modules in Fig. 3. These modules only have to be fail-silent, thus only having to ensure no faulty data is transmitted. Only the darker colored modules have to be fail-operational and thoroughly available also in cases of malfunctioning.

C. Overlapping trajectories

To keep this architecture running, a receding horizon control is necessary [20]. Otherwise the vehicle would stop after a

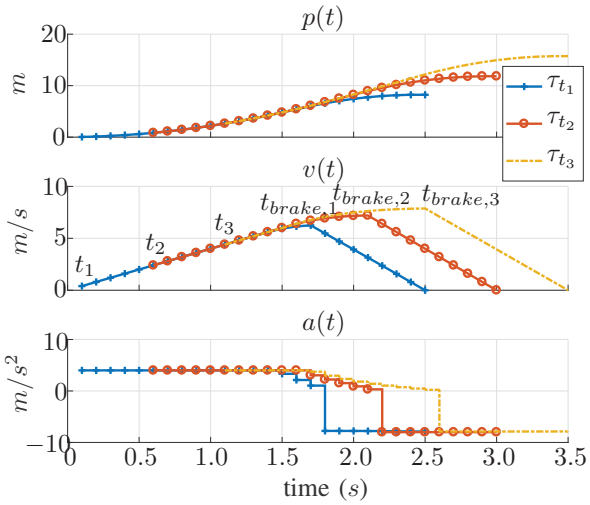

Fig. 4. Short, overlapping receding horizon trajectories form a steady long-term path until the buffer runs empty.

few seconds. To ensure a smooth transition, at least one new trajectory should be provided by the trajectory planner while the vehicle is still executing the last valid trajectory and has not yet reached the transition to the full stop. This enables the system to keep operating even if there is one trajectory found to be unsafe. In that case, the unsafe trajectory can be skipped while the following one might be safe again and will be written to the buffer before the braking phase starts.

Fig. 4 shows a simplified representation of this concept by means of an example. The overall path, expressed by the enveloping curve over $\mathcal{T} = \{\tau_{t_1}, \tau_{t_2}, \tau_{t_3}\}$, is the macroscopic goal of the autonomous system. This path is chopped into short-time sections τ_{tn}, which eventually come to a safe stop. This can be seen from the horizontal lines of $p(t)$ implying a stationary vehicle. While τ_{t_1} is still executed and the vehicle is in motion, τ_{t_2} overlaps steadily. Thus, the vehicle continues its journey along the designated path. The same holds for τ_{t_3}. In case that τ_{t_2} would have been rejected, τ_{t_3} would overlap the first trajectory just in time to mitigate a stop of the car. A potential algorithm fulfilling the requirements of this approach is presented in [21]. Therein, the authors propose a MPC methodology, which computes trajectories coming to a full-stop of the vehicle at the end of its prediction horizon. In order to move the braking maneuver to the latest possible point on the trajectory a backward reach-set method is applied. This has the goal of achieving driving maneuvers without applying the braking action as a newly computed trajectory is expected before.

V. Prediction over short time horizons

Automated systems can have a significantly shorter reaction time than human drivers. Considering the situation of emergency braking, Table II compares the time needed from the initial event to a full stop for humans and automated vehicles [17], [22].

TABLE II
Dead times of human drivers and automated vehicles.

Task	Human $[s]$	Automated Vehicle $[s]$
Perception	0,1	0,05
Reaction	1	
Planning		0,2
Foot switch	0,3	
Brake application	0,2	0,1
Total time until braking	1,6	0,35

The dead times of automated systems are significantly shorter compared to human drivers and provide the opportunity to drive with much smaller prediction horizons.

For a simple prediction model, that extrapolates a position coordinate linearly into two dimensions over time, a short time horizon reduces the uncertainty of the actual position. In everyday life, humans tend to acquire information about intention and look at gestures to assume what another person is going to do next. A person waiting at a traffic light would be considered as trying to cross the road. This can be derived by the environment (the traffic light) and the orientation of the person (looking to the street). For the latter part, the task of determining the orientation is easy to humans but challenging for computers. The same holds true for realizing if a pedestrian is distracted while walking, as it is crucial to estimate if this pedestrian will actually interact with the environment or not. Since these intention and gesture-based models currently still lack the required confidence in their correctness, physical models should be favored [14].

To exploit the two above-mentioned observations, one of the key elements of the proposed architecture is its focus on short time horizons. Since the dead times are rather short, a comparably pessimistic, i.e., overly safe, prediction can be made without sacrificing performance, as it will still be rather precise. This eases the need for sophisticated prediction models of other traffic agents.

To underline the idea behind the concept in an example calculation, a very simple, physics-based model is used. In this example, pedestrians and localization are considered as given ground-truth data. Since there is no further knowledge about the orientation, intention and current action, a potential omnidirectional movement for pedestrians is assumed. The only boundary to this movement is the maximum speed considered as reasonable in the current environment. As a result, pedestrians are projected as circles on a 2D map, with radii growing by $speed * time$.

For this quantitative example, we are considering the boundary conditions, based on a hurrying pedestrian at 7 km/h, braking performance on dry asphalt (8 m/s^2) and the dead times from Table II adding up to 350 ms [18]. To demonstrate the impact of the trajectory duration, three values (500 ms, 1 s, and 2 s) are shown. A description of the used variables can be found in Table III.

Total time of the vehicle's short-time trajectory:

$$t_{trajectory} = t_{drive} + t_{dead} + v_{vehicle}/a_{decel}. \quad (2)$$

TABLE III
VARIABLES USED WITHIN CALCULATIONS.

Name	Variable	Unit
Vehicle dead time	t_{dead}	s
Planned undecelerated driving time	t_{drive}	s
Total time of the short-time trajectory	$t_{trajectory}$	s
Total length of the short-time trajectory	$d_{trajectory}$	m
Pedestrian movement radius	$d_{pedestrian}$	m
Vehicle speed	$v_{vehicle}$	$\frac{m}{s}$
Pedestrian speed	$v_{pedestrian}$	$\frac{m}{s}$
Vehicle deceleration	a_{decel}	$\frac{m}{s^2}$

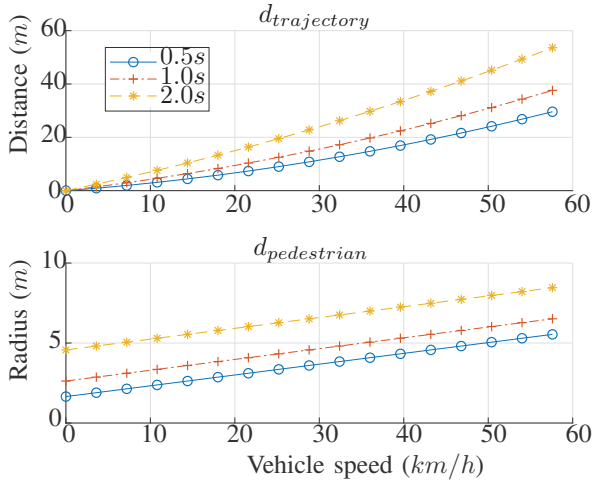

Fig. 5. Distance covered by vehicle and pedestrian movement radius with different trajectory durations t_{drive}.

Total distance of the vehicle following the trajectory, including the full-stop maneuver:

$$d_{trajectory} = (t_{dead} + t_{drive}) * v_{vehicle} + \frac{1}{2} * v_{vehicle}^2 / a_{decel}. \quad (3)$$

The pedestrian movement radius:

$$d_{pedestrian} = t_{trajectory} * v_{pedestrian}. \quad (4)$$

Fig. 5 shows how the total distance that is covered by the trajectories grows over speed with different trajectory planning horizons in comparison with the potential movement radius of a pedestrian.

Resulting from the two independent movements of the vehicle and the pedestrian, a potentially hazardous zone can be drawn. Since the movements are not depending on each other, the only common variable is the time. To draw the final zone of interest, the coordinates of the boundary are calculated by superimposing the two movements over the time horizon of interest, given by the trajectory length including the full-stop maneuver.

Fig. 6 shows a projection of this resulting area in a real parking lot. Even with this rather primitive model it is possible to keep the necessary safety distances.

VI. Interpretation of Results

The above-mentioned examples show that the duration of a vehicle trajectory significantly influences its safety margins.

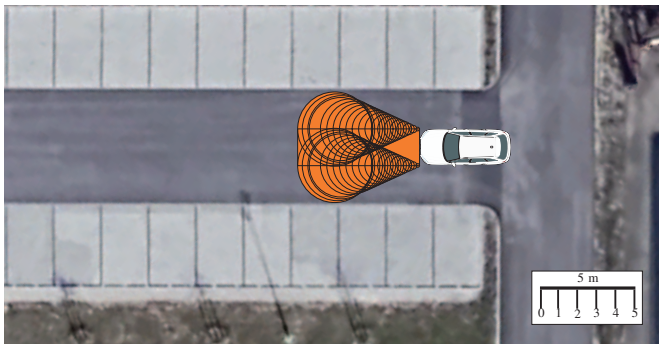

Fig. 6. The resulting hazardous zone by superimposing the movements projected on a real parking lot. Circle pairs represent accumulated time steps.

Consequently, a shorter trajectory is beneficial to performance. While the stopping corridor grows in a quadratic manner, the uncertainty radius of the pedestrian only grows linearly to the vehicle's speed. That means the potentially dangerous zones become elongated by higher speeds but grows linear in each direction by the amount of dead time and speed.

On the other hand, a shorter trajectory tightens the time frame available for providing an alternative trajectory when the previous one has been rejected. To keep the vehicle in steady motion, t_{drive} should be at least twice as long as the time needed to acquire new data and calculate a new trajectory. In the given example that would be at least $500\ ms$. As a result, the variable for tweaking the system is not based on safety considerations but on performance and convenience. If the system fails to fulfill the desired function, it means that the performance and convenience requirements will not be reached but safety is still guaranteed. This approach makes it possible to optimize the planning algorithms without having to worry about possible impairments of safety. In other words, the approach guarantees the safety of the intended functionality.

VII. Discussion

A. Strengths at lower speeds

The proposed approach shows the advantages of planning on short time horizons especially at lower speeds. For applications such as automated valet parking, the very simple model of a omnidirectionally moving pedestrian that is only limited by a maximum speed is already sufficient. Considering the narrow spaces on parking grounds, the available free space is confined. Since the general speed limit on parking areas in most European countries is set at $10\ km/h$, the stopping time calculates to $0.7\ s$. Combining this with a trajectory of $0.5\ s$ to follow the initially planned path, the total radius of uncertainty adds up to $2,33\ m$. This distance covers a lot of scenarios in daily driving in parking lots. For critical cases, the speed can be reduced further, what again reduces the longitudinal and lateral distance needed, until the minimum distance that is calculated by $(t_{drive} + t_{dead}) * v_{pedestrian} = (0.5\ s + 0.35\ s) * 7\ km/h = 1.65\ m$. This potentially conflicting area becomes so small, that just one or two steps of the pedestrian might solve the conflicting situation. At an

increased speed of $50\ km/h$ the lateral distance needed grows to $5\ m$ on each side. Therefore more sophisticated prediction models could be necessary to cope with this situation.

B. Potential for distributed system implementation

The proposed approach allows to be implemented as a distributed design.

In a baseline scenario, all components would be located within the vehicle. Depending on the use-case, moving single components out of the car might inhabit benefits for the overall performance and engineering challenge. Taking the use-case of automated valet parking as an example, the sensing and trajectory planning could be done stationary within the parking grounds. The only elements that have to necessarily be allocated within the vehicle are the trajectory buffer and a rudimentary observer, rejecting outdated trajectories. As mentioned before, there is no availability requirement for the precedent elements, thus wireless communication would be possible. The advantage of being able to move the components out of the vehicle are diverse. For once, data can be collected and processed at a central location, e.g., camera images from different cameras within a parking garage can be combined. This can help avoiding blind spots in the perception range. The path and trajectory planning can be moved to a centralized station as well, improving the performance of vehicle coordination and traffic flow. The second huge benefit of this architecture is the simplification of the hardware used. Stationary hardware does not have to be designed to withstand harsh environmental conditions like in-vehicle components and does not add separate load to the vehicle in terms of space, weight and energy consumption.

VIII. Conclusion

We presented a way to address the challenges of functional safety and safety of the intended functionality with a combined method. To make use of complex methodologies such as artificial neural networks for the trajectory planning, a mixed system approach ensures safety by adding a supervising element that is more amenable to safety verification. The quantitative experiment shows that the architecture performs especially well under low speeds and with short-time trajectories. For higher speeds, a cascaded system can be considered, lowering the computation demand by defining the actual area of interest with a safety policy of low complexity. Based on the architectural design, an effective decomposition of the ASIL requirements is possible, ensuring safety in a mixed system environment. By exploiting the short time-horizons needed to come to a safe stop, a redundancy free design could be proposed. It lowers the complexity and demands on energy, space and money for realizing level 4 driving functions. Furthermore, the modular design enables a distributed setup of the different system components. The omission of fail-operational components enables to introduce wireless communications into this setup.

References

[1] On-Road Automated Driving (ORAD) committee, "J3016 - Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," SAE International, Standard. [Online]. Available: https://www.sae.org/content/j3016_201806

[2] M. A. Hörwick, "Sicherheitskonzept für hochautomatisierte Fahrerassistenzsysteme," Thesis, 2011.

[3] I. O. for Standardization, "ISO/PAS 21448:2019: Road vehicles - Safety of the intended functionality," ISO, Geneva, Switzerland, Standard, Jan. 2019.

[4] W. H. K. Wachenfeld, "How Stochastic can Help to Introduce Automated Driving," Thesis.

[5] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182–193, 2016.

[6] S. J. Oh, B. Schiele, and M. Fritz, "Towards reverse-engineering black-box neural networks," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, 2019, pp. 121–144.

[7] I. O. for Standardization, "ISO 26262: Road Vehicles : Functional Safety," ISO, Geneva, Switzerland, Standard, 2011.

[8] A. J. Kornecki and K. Hall, "Approaches to assure safety in fly-by-wire systems: Airbus vs. boeing," in *IASTED Conf. on Software Engineering and Applications*, 2004, pp. 471–476.

[9] P. Koopman and M. Wagner, "Challenges in autonomous vehicle testing and validation," *SAE International Journal of Transportation Safety*, vol. 4, no. 1, pp. 15–24, 2016.

[10] J. E. Stellet, M. R. Zofka, J. Schumacher, T. Schamm, F. Niewels, and J. M. Zöllner, "Testing of Advanced Driver Assistance Towards Automated Driving: A Survey and Taxonomy on Existing Approaches and Open Questions," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sep. 2015, pp. 1455–1462.

[11] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a Formal Model of Safe and Scalable Self-driving Cars," *arXiv:1708.06374 [cs, stat]*, Aug. 2017, arXiv: 1708.06374. [Online]. Available: http://arxiv.org/abs/1708.06374

[12] C. Pek, P. Zahn, and M. Althoff, "Verifying the safety of lane change maneuvers of self-driving vehicles based on formalized traffic rules," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1477–1483.

[13] V. Schönemann, B. Böddeker, H. Winner, T. Glock, S. Otten, E. Sax, G. Verhaeg, F. Tronci, and G. Garcia Padilla, "Scenario-based functional Safety for Automated Driving on the Example of Valet Parking," in *2018 Future of Information and Communication Conference (FICC)*, IEEE, Ed.

[14] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH journal*, vol. 1, no. 1, p. 1, 2014.

[15] M. Koschi, C. Pek, M. Beikirch, and M. Althoff, "Set-based prediction of pedestrians in urban environments considering formalized traffic rules," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2704–2711.

[16] C. Pek, M. Koschi, and M. Althoff, "An Online Verification Framework for Motion Planning of Self-driving Vehicles with Safety Guarantees," 2019.

[17] W. Spielmann and M. Reuter, "Bremswege als Vergleichsgröße zwischen Bremsanlagen," *ATZ*, vol. 104, no. 5, pp. 464–472, Jan. 2002.

[18] N. Kudarauskas, "Analysis of emergency braking of a vehicle," *Transport*, vol. 22, no. 3, pp. 154–159, 2007.

[19] R. R. Murphy, *Introduction to AI Robotics*, 1st ed. Cambridge, MA, USA: MIT Press, 2000.

[20] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.

[21] M. Kneissl, A. Molin, H. Esen, and S. Hirche, "A one-step feasible negotiation algorithm for distributed trajectory generation of autonomous vehicles," in *Proceedings of the Conference on Decision and Control (CDC)*, 2019.

[22] M. Powelleit, E. Muhrer, M. Vollrath, R. Henze, L. Liesner, and T. Pawellek, "Verhaltensbezogene Kennwerte zeitkritischer Fahrmanöver," *Berichte der Bundesanstalt für Strassenwesen - Fahrzeugtechnik (F)*, vol. 100, Jan. 2015.