©2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
2016 IEEE Intelligent Vehicles Symposium (IV), pp. 304-309, 19-22 June 2016.

Functional System Architectures towards Fully Automated Driving

Ömer Şahin Taş, Florian Kuhnt, J. Marius Zöllner and Christoph Stiller

***Abstract*— The functional system architecture of an automated vehicle plays a crucial role in the performance of the vehicle. When considered as a backbone, it does not only transmit information between distinct layers, but rather serves as a feedback mechanism coordinating the degradation between them and thereby regulates the behavior of the system against failures. Hence, the design of *robust* functional architectures is essential to cope with the uncertainties of the world.**

This paper summarizes existing system architectures and investigates them regarding their robustness against measurement inaccuracies, failures, and unexpected evolution of traffic situations. After illustrating their strengths and deficiencies, we derive the requirements and propose a structure for future, robust system architectures.

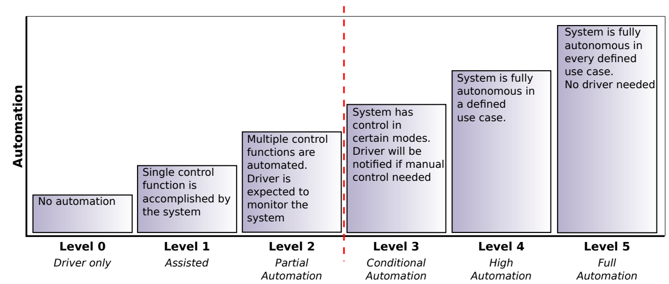

Fig. 1. The taxonomoy made by the Society of Automotive Engineers (SAE) [2]. The SAE carved out automated driving in six levels. The red line illustrates where robustness starts to play a vital role on automated driving.

I. Introduction

The past decade has witnessed tremendous progress in the field of automated driving. Although there has been noteworthy work on autonomous vehicles before, the competition structure of the DARPA challenges boosted their development. The first prototypical vehicles that showed up could hardly recover from failures or deal with situations where driving skills such as tightly merging into traffic were required. Through the experience gained, it become feasible to leave the artificial environment of the challenges and cope with the unpredictable nature of the real world by utilizing cost effective sensors and fault tolerant systems.

An extensive and up to date review on the evolution of intelligent vehicles is given in [1]. Briefly seen from the classification perspective and by adopting the taxonomy of the Society of Automotive Engineers (SAE) (see Figure 1) [2], the focus of research has shifted to the realization of highly and even fully automated driving. Experience shows that both of these levels of automation require highly robust driving systems and there are still challenging use cases that must be mastered [3]. In fact, mastering these is not only substantial for the SAE level 4 and 5 automation, but relatedly on the rate of triggering driver take-over requests in case of level 3, *conditional automation*. Increasing the automation level requires remarkable enhancements on the existing perception, prediction and planning algorithms but also on the system architectures [3]. Although the former have been an active field of research, focus on the system architecture has been limited so far. However, in our point of view, system architecture is a key element of fault detection and its design has therefore a massive influence on the overall robustness and reliability of an autonomous vehicle. Developing such a system architecture is one of the main objectives of the pan-European RobustSENSE project [4].

In this paper, instead of evaluating and comparing individual algorithms on perception, prediction, and planning, we deal with robustness and reliability by focusing on the architectural design of an autonomous vehicle and emphasize its influence on degraded operation modes in case of sensor failures, misdetections, and unexpected evolution of traffic situations. We compare existing architectures and derive their strengths and deficiencies. We do this by inspecting the state of the art of autonomous vehicles. Thereby we reveal the obscure progress in the system architectures and shed light on the requirements towards fully automated driving. We restrict our analysis to system architectures and refrain from focusing on network communication between single modules. Besides, although the v2x communication is a vast area of research and has tremendous potential to increase robustness, we also leave its integration unanalyzed. The reason for that is, from the architectural perspective, it shares the common properties of an array of redundant sensors. What might be an interesting consideration are the effects of information delay, and particularly, discrepancy, which are already studied in [5], [6], but lie out of the scope of this paper.

The rest of the paper is structured as follows: In Section II, we consider the autonomous vehicle as a cyber-physical system and then examine the layers and components of it. In Section III, we first define robustness and reliability and then recapitulate the basic categorization made in autonomous vehicle architectures. Then in subsection III-B, we inspect the architecture of various autonomous vehicles. In Section IV, we summarize the results of the comparison and then outline the requirements for realizing robust system architectures for highly automated driving. Finally in Section V, we provide an outlook for further studies for improving robustness.

The authors are with FZI Research Center for Information Technology, Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germany {tas, kuhnt, zoellner, stiller}@fzi.de

II. The Automated Vehicle as a Cognitive System

An automated vehicle can be seen as a cognitive technical system. A cognitive system mainly follows the concept of a *rational agent*: observes its environment, computes a decision and performs actions on its environment (see Fig. 2) [7, p. 35].

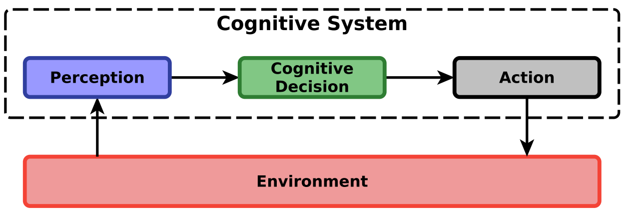

Fig. 2. A cognitive system interacting with its environment. The agent consists mainly of three components: perception, cognitive decision, and action module [7, p. 35].

An automated vehicle, also comprises mainly of those components. A deeper classification is possible, but when typical components are considered, its localization, obstacle detection, fusion and situation prediction components can be grouped under the *perception* module, behavior and motion planning under the *cognitive decision* module, and the control and actuation components under the *action* module. Very similar classifications are made in [8], [9], [10], whereas the latter furthermore sums up the functions of these components. Note that, in robotics, these modules are interchangeably called *perception*, *planning* and *control*.

III. System Architectures for Fully Automated Driving

In an automated vehicle, the arrangement of the modules stated in the previous section plays a crucial role on the *robust* and *reliable* operation of the vehicle. We define robustness as the faculty to adapt and operate properly under a large diversity of operating conditions (*e.g.* weather, illumination, traffic *etc.*), whereas we define reliability as the shortest time for a given system to fail or malfunction.

In this section, we first recapitulate the topological distinction made in the system architectures and then delve into the architectural details of the state of the art vehicles.

A. Types of Architectures

From the topological perspective, we distinguish between two distinct types of functional architectures in automated vehicles [11]:

- centralized architecture
- distributed architecture

The *centralized architecture* corresponds to the most simple form of an autonomous vehicle architecture. In this type, all of the perception, decision making, and planning are realized inside a single computing unit. Hence, this arrangement saves excessive networking and thereby avoids potential information delay and loss. Nevertheless, as clarified in [12], this simple structure has significant drawbacks. Firstly, collecting all of the computational burden into a single unit requires high computing power. Secondly, as the functions are not separated, it is not possible to develop and test the submodules simultaneously. A further problem may arise from the wire harnesses, which become more sensitive to noise as a result of the longer wiring [12]. A more crucial problem results from the inability to reliably detect and diagnose the failures and switch to a backup module.

The second type is called the *distributed architecture*. In this variant, in contrary to the former one, the individual modules are implemented inside distinct computing units. By means of this, one may experience performance loss seen from an information-theoretic perspective. However, the stated problems of the centralized architecture are resolved. In exchange, the communication must be secured against network problems [11]. For this demand, a time-triggered communication protocol, for example, depending on the specific application field, the FlexRay or the Automotive Data and Time-triggered Framework (ADTF) emerge as a solution [12], [13]. Note that the term distinct *computing unit* does not necessarily correspond to distinct physical computers. The distinct computation nodes of an architecture will also create a distributed framework of processes, as in the case of the Robot Operating System (ROS). This software framework has also found practical application in autonomous driving [14]. It should however be noted that ROS is event-triggered and is not a real-time system. Hence, it may be subject to information loss.

From our point of view and as also pointed out in the *Engineering Requirements of System Architectures* subsection of [15], an architecture should be able to contain fault detection and diagnosis components that reliably self-monitor its system status. As will be revealed in the following, centralized architectures could hardly find application in state of the art autonomous vehicles. Along with the forestated reasons, we do not expect this to change in fully automated vehicle architectures.

B. State-of-the-art Autonomous Vehicle Architectures

We start our analysis with *Boss*, the winner of the DARPA Urban Challenge [16], [17]. Like any other autonomous vehicle that took part in the challenge, Boss was tailored for the needs of the Urban Challenge. Hence, its architecture was shaped accordingly.

Boss was comprised of four main modules: perception, mission planning, behavioral executive and motion planning (see Figure 3). The perception module was responsible for fusing sensor data, object tracking, localization and static obstacle map generation. The raw sensor data was processed in the sensor specific submodule first. The measurements and the extracted features were then sent to the fusion layer and a set of object hypotheses and validated features were created. To improve the quality of the observation, these were also fed back to the distinct sensor layers. Subsequently, estimated object states were generated from the most probable observation. After information about the environment was gathered, the mission planner computed the costs of the checkpoints using the Route Network Definition File (RNDF) and returned an optimal route. The route was then converted into a sequence of abstract maneuver decisions, such as lane driving, intersection handling, and zone maneuvering. The motion planner then computed a set of candidate trajectories and the best one was selected for execution.

A remarkable property of Boss was its progress monitoring system embedded inside its behavioral executive. This system continuously monitored the progress and functioned like an exception handler: whenever its mission was repeatedly obstructed, it defined *recovery levels* associated with various maneuvers to circumvent the situation. The system also did not directly consider the obstacle data at the selection of a recovery goal. This made the system reliable in situations where transient environmental effects or perception artifacts occurred [17]. Besides Boss, the vehicle *AnnieWAY*, which was among the finalists, did also use such a progress monitoring system [18]. But the system of Boss could also perform hot and cold standbys, making it fault tolerant [19].

In 2013, the team that developed Boss presented a new autonomous vehicle. Instead of being equipped with expensive sensors and looking like a prototype vehicle, as Boss did, this vehicle is close to series production. When considered from the architecture and self monitoring perspective, this is a descendant of the former vehicle [20]. Later on the group stated that the demand on the motion planner to simultaneously regard distinct objectives is challenging, and the motion planner lacks to have a direct feedback from the vehicle controllers [21]. In the same work, they addressed this issue by proposing to evaluate the controller directives inside the motion planning layer, as shown in Figure 3. In our point of view, this is an important step towards increasing robustness. Nevertheless, the behavior planner still does not have direct information on the execution-quality of the planned trajectory. Evaluation of such an information, along with the availability of a fall-back strategy for the given trajectory, could increase the robustness of the behavior and trajectory planning layers. Furthermore, the system is still ineligible to find a reasonable decision in case of conflicting perception information.

In the same year team AnnieWAY, formed by the members of the Karlsruhe Institute of Technology (KIT) and the FZI Research Center for Information Technology, in collaboration with Daimler AG presented the autonomous vehicle *Bertha*, known for its successful completion of the Bertha Benz Memorial Route [23]. As can be inferred from the system overview paper, the primary goal in this project was to realize autonomous driving with cost effective sensors in a close to serial production vehicle, rather than ensuring robust and reliable operation. The perception system therefore consisted solely of video cameras, automotive grade radars and low-cost GNSS. For low-cost yet precise localization, it utilized vision based localization and thereby preserved its precision even in areas with deterioration effects due to satellite occlusion and multipath propagation [24]. The

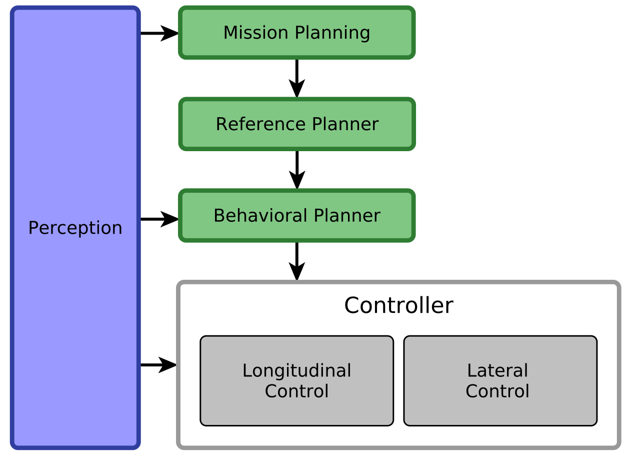

Fig. 3. The *behavioral planning* framework. In this architecture, the reference planner ignores the dynamic obstacles in the surrounding and creates a path. The behavioral planner then considers the interaction between vehicles and evaluates and selects the best control inputs for separated multiple controllers, which fundamentally consist of longitudinal and lateral controllers [21].

system did not have any fall-back mechanisms for degraded operation and misdetections. The perception was restricted to vision and radar sensors, whereas the side perception was solely realized with the latter one. From the same paper, it can further be inferred that there was no direct information flow from the trajectory planner back to the decision making module.

At the same time in Italy, the follow up group that took part in Urban Challenge under the team name *TerraMax* demonstrated the vehicle platform *BRAiVE* and around one year later the platform *Deeva* [9], [25]. The system architectures of both vehicles separate the main three layers, namely the perception, planning, and control, from each other. The uncertainties at each layer therefore grow incrementally, which could lead to overly conservative behavior at the end. Furthermore, as already mentioned in the referred paper, the architectures of these platforms do not have any module for failure handling. Like Bertha, robustness was not a primary matter of concern.

A similar competition to the DARPA Urban Challenge was performed in Korea, Asia. The autonomous vehicle *A1*, that won the 2012 Autonomous Vehicle Competition was able to complete all of the missions such as handling moving vehicles and pedestrians, understanding traffic lights etc. [12], [22]. The team first tried to use a centralized architecture, however encountered the problems stated in Section III-A and therefore switched to a distributed architecture. The system architecture of A1 is presented in Figure 4. As shown in the figure, the only measure for ensuring safe operation is the system management module. This module observes the life signal of the individual modules and in case the health status of a critical module fails, it backs up the failed module. This module prevented some accidents caused by software errors [22].

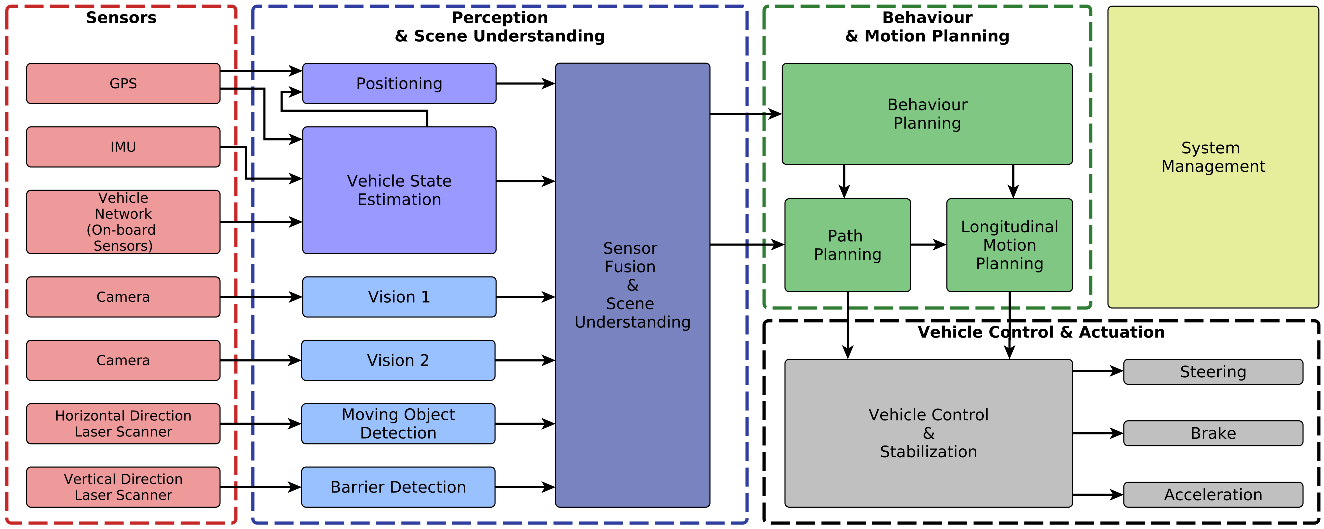

Fig. 4. The information flow between the modules of the automated vehicle A1, the vehicle that won the Korean Autonomous Vehicle Competition [12], [22]. The system management module detects the failures and in case backs up the failed unit. Therefore, the presented architecture can be regarded as the most basic structure to maintain robustness.

Autonomous race cars, due to their high dynamics, require a more sophisticated monitoring system. Such a system is implemented in the autonomous race car *Shelley* [26]. It watches other system modules and in case of any error condition, such as error states, information in undesirable range, or logical inaccuracies, it performed a controlled stop along the path. As the vehicle is designed to drive in obstacle free environments, in case of a controlled stop the system only endeavors to keep the vehicle inside the road boundaries until standstill. A unique feature of this monitoring system is that it is not only separated from the other modules by its hardware, but also in its programming language: the monitoring system is implemented in a different language than the other modules. This reduces the likelihood of including implementation errors of the high level controller inside the monitoring system. The monitoring system is also divided into submodules, so that a master module can detect failures in its associated submodules. Such a measure certainly increases the reliability of the monitoring system further.

Another state of the art robust system is the autonomous vehicle of the Ulm University [27]. This vehicle differs from the aforementioned ones in its fusion algorithm. The vehicle is set up with a multi-sensor setup that comprises different types of sensors. A great portion of the environment of the vehicle is covered with at least two different types of sensors. Such a redundant setup meets the fundamental requirement for maintaining fault tolerance in a perception system. However, this is not sufficient as the information from those distinct sensors must also be fused accordingly. The vehicle uses a probabilistic centralized sensor fusion system. The overall architecture is presented in Figure 5. As presented in the figure, the fusion takes place in three different layers: in grid mapping, in localization and in object tracking. The results are then fused into an environment model. At this phase the conflicting perception information is resolved. Such a probabilistic perception module hinders information loss in the fusion layers and therefore yields more reliable results. A fusion architecture of this type is also robust against the changes in the system. Even when a sensor fails to operate, the fusion is performed with the rest of the perception system.

The probabilistic fusion approach of the Ulm University clearly improves the quality of the perception. By integrating the uncertainties into the object prediction, the vehicle motion can be adapted to harsh environmental conditions. This will allow the autonomous vehicle to drive in degraded states. On the other hand, the architecture lacks a system monitor and hence is not able to issue recovery goals. Furthermore, the localization system is highly sensitive to the actuality and accuracy of the enhanced digital map.

A further state of the art autonomous vehicle, *Jack* has been presented in 2015 by Audi AG [28]. In order to perform probabilistic reasoning from perception uncertainties the system utilizes Bayesian networks and an unscented variance transform. The integration of the uncertainties into the behavior layer increases reliability and robustness of the automated driving system. The system architecture allows redundant information flows which can eventually be fused and compared: one type of information is gathered from global localization and mapping, and the other one from the environmental perception. If information discrepancies between the both sources occur, the vehicle switches itself into a safe state [29]. By separating the source of scene modeling information and monitoring the discrepancies between them, fault statuses are deduced in this architecture.

IV. Conclusions on Implemented Architectures

In this paper we have inspected several successful and popular autonomous vehicles. We used redesigned graphics

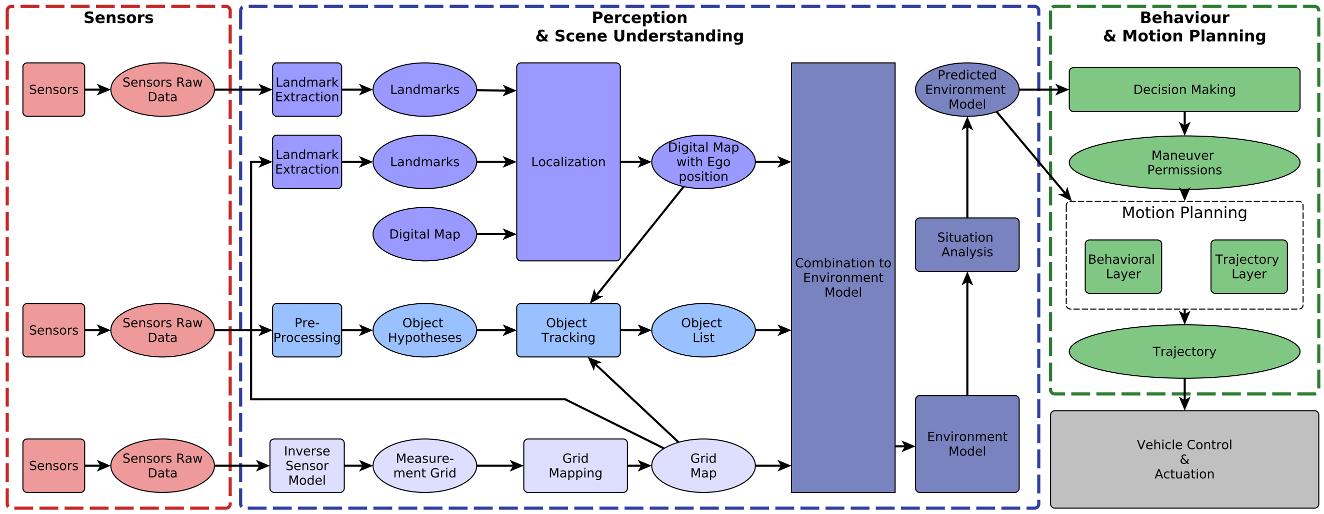

Fig. 5. The functional system architecture diagram of Ulm University's autonomous vehicle [27]. The independent modules in the above referred publication are merged into a single diagram. Thereby an overview of the entire architecture is achieved. In this architecture, the fusion takes place in several layers of the perception system while retaining the confidence level of the sensor information.

from papers and brought them into unified representations to highlight similarities and differences in their functional architectures. Our goal was to reinvoke interest in functional system architectures. We have revealed that the vehicles that could deal with faults in the system have subsequently proven to exhibit renowned performance. It can clearly be deduced that in the development of highly automated vehicles robust and reliable operation is increasingly becoming a matter of interest and the endeavor towards realizing fully autonomous driving will bring further advancements in functional system architectures along with the progress in the individual algorithms.

In Section III-B we started our analysis with Boss. In its perception system, results of the fusion were sent back to the sensor specific raw data processing module. This had improved its perception accuracy. Furthermore, it employed a complex recovery module to handle freezes. These measures made the system fault tolerant and contributed to the success of the vehicle. The same group later on proposed to evaluate possible control inputs inside the motion planning layer. This can be seen as a feedback from controllers to the motion planner serving to improve reliability.

We subsequently inspected two vehicles with close to serial production sensors developed by different research groups. Because of their specific use, both systems lack degraded operation modules. Afterwards we inspected the Korean A1. Even though it had a very simple fault diagnosis and recovery system, it played a vital role on the success of the vehicle. We then considered an autonomous race car traversing very sharp turns at the limits of handling. As such a vehicle had very high demands, we saw a very elaborated monitoring system that could even take over the vehicle control, integrated in the architecture. Next, we inspected the autonomous vehicle of Ulm University. Its fusion architecture enabled robust and reliable perception. However, it required very accurate and reliable high precision map data and the system was not robust against the changes in it. We eventually reviewed Jack. Compared to the vehicle of Ulm, it was robust against the uncertainties in the measurement system. Furthermore, it could reason about the discrepancies between the map-based localization information and the perception information.

The number of inspected vehicles can surely be extended. We tried to inspect different types of vehicles with different types of architectures from all over the world. When we summarize their most commendable features, they:

- are built on a distributed architecture
- utilize monitoring system that supervises the entire system and in case of any failure, releases recovery instructions and switches to degraded operation modes
- cover the vehicle surrounding with redundant and complementary types of sensors
- consider uncertainty in sensor information and the propagation of it
- use feedback from the controller to the motion planning module
- benefit from the separation and redundancy of modules
- allow graceful degradation of software functions
- can compare sensor information with map-based data and deduce results about system status

V. Outlook for Robust and Reliable System Architectures

The state of the art autonomous vehicles already utilize the measures summarized in the previous section. However, the number of disengagements are still too high to drive fully automated [30]. As reasoned out before, functional system architectures will therefore be one main focus for improving robustness and reliability.

Future work will integrate the existing solution approaches into a unified functional system architecture. We expect every module to employ various metrics that quantify its performance status and thereby give feedback to the vehicle to switch into degraded operation modes. Such an advanced monitoring system can further be used to activate alternative data processing algorithms to adapt the system to changed environments or sensor performances. During such an algorithm or operation mode adaptation, transition times must be limited and safety must be guaranteed [31]. Metrics that reflect confidence, existence probabilities, and consistencies of the perception and fusion modules will allow a better long term prediction about the evolvement of the environment and increase robustness in case of temporary sensor degradation. This will allow the trajectory planner to adapt its velocity according to the guarantees of the perception system to meet safe and acceptable minima and generate safe but not overly conservative trajectories. Such a monitoring system will bring the individual modules in harmony and will improve the system robustness and reliability.

Acknowledgements

The research leading to these results has received funding from the European Union under the H2020 EU.2.1.1.7. ECSEL Programme, as part of the RobustSENSE project, contract number 661933. Responsibility for the information and views set out in this publication lies entirely with the authors. The authors would like to thank all partners within RobustSENSE for their cooperation and valuable contribution.

References

[1] K. Bengler, K. Dietmayer, B. Färber, M. Maurer, C. Stiller, and H. Winner, "Three decades of driver assistance systems: Review and future perspectives," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 4, pp. 6–22, 2014.

[2] SAE, "Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems," SAE, 2014.

[3] O. Pink, J. Becker, and S. Kammel, "Automated driving on public roads: Experiences in real traffic," *it-Information Technology*, vol. 57, no. 4, pp. 223–230, 2015.

[4] "RobustSENSE – Robust and Reliable Environment Sensing and Situation Prediction for Advanced Driver Assistance Systems and Automated Driving – Project," 2016, [Retrieved: January 22, 2016]. [Online]. Available: http://www.robustsense.eu

[5] S. Öncü, N. van de Wouw, W. Heemels, and H. Nijmeijer, "String stability of interconnected vehicles under communication constraints," in *IEEE Annu. Conf. on Decision and Control (CDC)*, 2012, pp. 2459–2464.

[6] P. Fernandes and U. Nunes, "Platooning with IVC-enabled autonomous vehicles: Strategies to mitigate communication delays, improve safety and traffic flow," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 91–106, 2012.

[7] S. Russell and P. Norvig, *Artificial intelligence: a modern approach*, 1995.

[8] C. Berger and B. Rumpe, "Autonomous Driving-5 Years after the Urban Challenge: The Anticipatory Vehicle as a Cyber-Physical System," *CoRR*, vol. abs/1409.0413, 2014.

[9] A. Broggi, P. Cerri, S. Debattisti, M. C. Laghi, P. Medici, D. Molinari, M. Panciroli, and A. Prioletti, "Proud—public road urban driverless-car test," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3508–3519, 2015.

[10] S. Behere and M. Törngren, "A functional architecture for autonomous driving," in *Proceedings of the First International Workshop on Automotive Software Architecture*. ACM, 2015, pp. 3–10.

[11] U. Nunes, J. A. Fonseca, L. Almeida, R. Araújo, and R. Maia, "Using distributed systems in real-time control of autonomous vehicles," *Robotica*, vol. 21, no. 03, pp. 271–281, 2003.

[12] K. Jo, J. Kim, D. Kim, C. Jang, and M. Sunwoo, "Development of Autonomous Car–Part I: distributed system architecture and development process," *IEEE Trans. Ind. Electronics*, vol. 61, no. 12, pp. 7131–7140, 2014.

[13] R. Schabenberger, "ADTF: Framework for driver assistance and safety systems," *VDI BERICHTE*, vol. 2000, p. 701, 2007.

[14] M. Aeberhard, T. Kühbeck, B. Seidl, M. Friedl, J. Thomas, and O. Scheickl, "Automated Driving with ROS at BMW," ROSCon 2015 Hamburg, Germany. [Retrieved: January 20, 2016]. [Online]. Available: http://roscon.ros.org/2015/presentations/ROSCon-Automated-Driving.pdf

[15] J. Mårtensson, A. Alam, S. Behere, M. Khan, J. Kjellberg, K.-Y. Liang, H. Pettersson, D. Sundman *et al.*, "The development of a cooperative heavy-duty vehicle for the GCDC 2011: Team Scoop," *IEEE Trans. Intell. Trans. Syst.*, vol. 13, no. 3, pp. 1033–1049, 2012.

[16] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[17] C. R. Baker, D. I. Ferguson, and J. M. Dolan, "Robust mission execution for autonomous urban driving," *Robotics Institute*, p. 178, 2008.

[18] S. Kammel, J. Ziegler, B. Pitzer, M. Werling, T. Gindele, D. Jagzent, J. Schröder, M. Thuy, M. Goebl, F. v. Hundelshausen *et al.*, "Team AnnieWAY's autonomous system for the 2007 DARPA Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 615–639, 2008.

[19] J. Kim, G. Bhatia, R. Rajkumar, and M. Jochim, "SAFER: System-level Architecture for Failure Evasion in Real-time Applications," in *Proc. IEEE 33rd Real-Time Systems Symp.*, 2012, pp. 227–236.

[20] J. Wei, J. M. Snider, J. Kim, J. M. Dolan, R. Rajkumar, and B. Litkouhi, "Towards a viable autonomous driving research platform," in *IEEE Proc. Intell. Veh. Symp.*, 2013, pp. 763–770.

[21] J. Wei, J. M. Snider, T. Gu, J. M. Dolan, and B. Litkouhi, "A behavioral planning framework for autonomous driving," in *IEEE Proc. Intell. Veh. Symp.*, 2014, pp. 458–464.

[22] K. Jo, J. Kim, D. Kim, C. Jang, and M. Sunwoo, "Development of Autonomous Car–Part II: A case study on the implementation of an autonomous driving system based on distributed architecture," *IEEE Trans. Ind. Electronics*, vol. 62, no. 8, pp. 5119–5132.

[23] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller *et al.*, "Making Bertha drive — An autonomous journey on a historic route," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 2, pp. 8–20, 2014.

[24] H. Lategahn and C. Stiller, "Vision-only localization," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 3, pp. 1246–1257, 2014.

[25] A. Broggi, S. Debattisti, P. Grisleri, and M. Panciroli, "The Deeva autonomous vehicle platform," in *IEEE Proc. Intell. Veh. Symp.*, 2015, pp. 692–699.

[26] J. Funke, P. Theodosis, R. Hindiyeh, G. Stanek, K. Kritatakirana, C. Gerdes, D. Langer, M. Hernandez, B. Müller-Bessler, and B. Huhnke, "Up to the limits: Autonomous Audi TTS," in *IEEE Proc. Intell. Veh. Symp.*, 2012, pp. 541–547.

[27] F. Kunz, D. Nuss, J. Wiest, H. Deusch, S. Reuter, F. Gritschneder, A. Scheel, M. Stubler, M. Bach, P. Hatzelmann *et al.*, "Autonomous driving at Ulm University: A modular, robust, and sensor-independent fusion approach," in *IEEE Proc. Intell. Veh. Symp.*, 2015, pp. 666–673.

[28] S. Ulbrich and M. Maurer, "Situation Assessment in Tactical Lane Change Behavior Planning for Automated Vehicles," in *IEEE Proc. Intell. Trans. Syst. Conf.*, 2015, pp. 975–981.

[29] R. Matthaei and M. Maurer, "Autonomous driving–a top-down-approach," *at-Automatisierungstechnik*, vol. 63, no. 3, pp. 155–167, 2015.

[30] "Google Self-Driving Car Testing Report on Disengagements of Autonomous Mode December 2015," [Retrieved: January 15, 2016]. [Online]. Available: "http://static.googleusercontent.com/media/www.google.com/us//selfdrivingcar/files/reports/report-annual-15.pdf"

[31] J. Becker and M. Helmle, "Architecture and System Safety Requirements for Automated Driving," in *Road Vehicle Automation 2*. Springer, 2015, pp. 37–48.