2017 IEEE International Conference on Intelligent Transportation Systems, 16-19 October 2017.

Automated Vehicle System Architecture with Performance Assessment

Ömer Şahin Taş*[1], Stefan Hörmann[2], Bernd Schäufele[3], and Florian Kuhnt[1]

***Abstract*— This paper proposes a reference architecture to increase reliability and robustness of an automated vehicle. The architecture exploits the benefits arising from the interdependencies of the system and provides self awareness. *Performance Assessment* units attached to subsystems quantify the reliability of their operation and return performance values. The *Environment Condition Assessment*, which is another important novelty of the architecture, informs augmented sensors on current sensing conditions. Utilizing environment conditions and performance values for subsequent centralized integrity checks allow algorithms to adapt to current driving conditions and thereby to increase their robustness. We demonstrate the benefit of the approach with the example of false positive object detection and tracking, where the detection of a ghost object is resolved in centralized performance assessment using a Bayesian network.**

***Index Terms*— System architecture, performance assessment, integrity monitoring, self awareness, fully automated driving, self driving vehicle, robust, reliable, RobustSENSE.**

I. Introduction

The evolution of active safety and driver assistance applications are going to a direction where the on-board vehicle computers can take over control of the vehicle from a human driver in increasingly more situations [1]. The main challenge, which still remains, is preserving reliability and robustness of the perception systems in all possible outdoor conditions, and the ability to react appropriately to the unexpected behavior of other traffic participants. In order to address these issues, several components have to be integrated in an automated driving architecture [2]. The two most important ones are the consequent consideration of uncertainties in completely probabilistic processing and the introduction of system-wide performance awareness. These two approaches are the focus of the pan-European project *RobustSENSE* [3] to improve robustness of advanced driver assistance systems and automated driving in all weather and driving conditions.

In this paper we present the RobustSENSE system architecture, which exploits redundant sensor information from a multi-sensor platform and thereby addresses system deactivation arising from a single sensor malfunction or degradation. The coordination among the individual sensors and subsystems is managed by performance assessment modules that monitor system performance during the vehicle's operation and return metrics that quantify the performance of the subsystems (cf. Fig. 1). This allows the subsystems to observe overall system performance and to review the reliability of their own output. Such an architecture enables switching to degraded operation modes and thereby allows to maintain continuous operation.

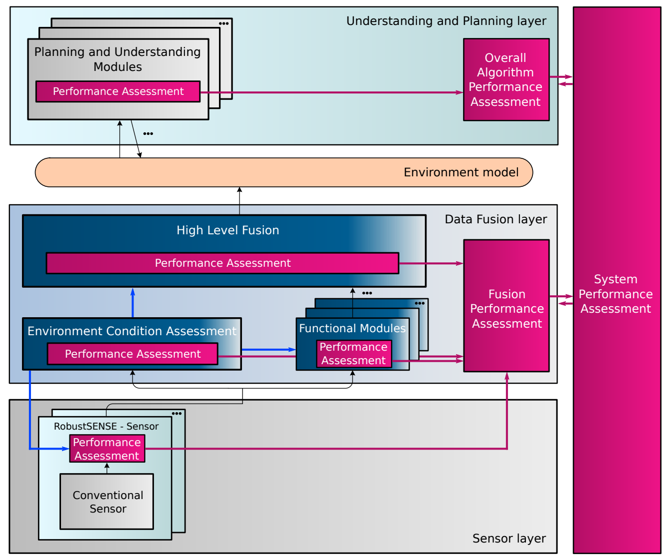

Fig. 1: An automated vehicle system architecture that benefits from redundancy, interdependencies and disparity through the utilization of performance assessment modules and an Environment Condition Assessment module.

The rest of the paper is structured as follows: In Section II we give an overview of related work. Subsequently, in Section III, we introduce the novel features of the RobustSENSE system architecture. Once the introduction of the architecture is complete, we continue with Section IV in which we present performance metrics defined for the performance assessment of our algorithms and afterwards demonstrate the success of the proposed approach. For the demonstration, we artificially inject a ghost object due to clutter and investigate the results on ghost object probability delivered by our Bayesian network. In Section V, we give a detailed outlook to highlight the impact to future research. Section VI concludes the paper by summarizing key elements and the benefits of the proposed architecture.

II. Related Work

Fault diagnosis and system monitoring are both well developed topics in engineering [4], [5]. Although our previous work [2], which reviews the requirements for realizing fully automated driving, has proven that the state of the art automated vehicles utilizing health monitoring systems can

*Corresponding author: tas@fzi.de, [1]FZI Research Center for Information Technology at the Karlsruhe Institute of Technology, Karlsruhe, Germany, [2]University of Ulm, Institute for Measurement, Control and Microtechnology, Ulm, Germany, [3]Fraunhofer Institute for Open Communication Technologies (FOKUS), Berlin, Germany.

deal with faults, a thorough application of system monitoring and performance assessment on automated vehicles remains undone.

One of the first representatives of mode switching systems was implemented in the VaMoRs-P [6]. The switching rules, however, were implemented as behavior selection rather than performance degradation. The notion of *situational awareness* was investigated by Albus et al. for task-based behavioral decisions [7]. In another work, they also presented metrics and performance measures for intelligent ground vehicles [8]. However, the metrics they presented are for testing the overall performance capabilities of the end system – or product.

In the DARPA Urban Challenge, the performance monitoring system implemented in the winner vehicle, *Boss*, monitored the progress in its mission. If the mission was repeatedly obstructed, it issued recoveries [9], [10]. Another advanced monitoring module was designed for *Shelley*. The module was completely separated from the rest of the system and executed three different kinds of stops in case of inconsistencies [11]. The winner of the Korean 2012 Autonomous Driving Challenge, *A1*, utilized also a very simple system management module, but this rescued the vehicle from many failures and contributed significantly to the success of the vehicle [12].

Another state of the art automated vehicle, *Jack*, performs probabilistic reasoning and deals with the discrepancies in its perception system. Furthermore, the vehicle can switch into degraded operation modes [13]. However, the automated vehicle lacks system-wide situational awareness. The authors that developed the system architecture of Jack propose an implementation independent functional system architecture for automated driving in their very recent publication [14]. However, as the paper inspects architecture from the functional perspective, conclusions on robustness and reliability can be made to a limited extent.

Literature, unfortunately lacks an architectural consideration of module based and system wide performance assessment for automated vehicles.

III. Novel Features of RobustSENSE System Architecture

In a layer-function based classification, current architectures typically distinguish between a *sensor* layer, a *perception and scene understanding* layer, and a *planning* layer [2], [15], [16]. By slightly diverging from this taxonomy, we divide the system architecture into four different layers and classify scene understanding and situation prediction modules within the understanding and planning layer.

A main feature to maintain robustness and reliability is to realize performance awareness. The RobustSENSE architecture evaluates sensors' performance under consideration of environmental conditions and associates the performance values of thereupon building subsystems by a system-wide performance assessment.

The layers and their colors in the figures throughout this paper are:

A. Sensor layer (grey)
B. Data Fusion layer (dark blue)
C. Understanding and Planning layer (light blue)
D. System Performance Assessment layer (magenta).

The layers strictly follow the information flow – all but one: the System Performance Assessment is a horizontal task that enables overall system monitoring and draws conclusions by observing the orthogonal and independent information flow of performance values.

A. Sensor Layer and RobustSENSE-Sensor

The Sensor layer is the lowest layer of the architecture. Through this layer, the system retrieves the information required for the basic environment perception.

Conventional sensors (dashed boxes in Fig. 2) deliver transduced, partly denoised and smoothed outputs and are unaware of the current environment conditions. However, in particular on an outdoor moving platform, varying sensing conditions highly influence the measurement. Therefore, we define the RobustSENSE-Sensor, which continuously evaluates the reliability and confidence of the measurement data under consideration of environment conditions. The information on environment conditions is delivered by the *Environment Condition Assessment* presented in the next subsection.

As shown in Fig. 2, a RobustSENSE-Sensor is set up by a performance assessment module attached onto a conventional sensor. Sensor models, possibly conditioned on the environment, are usually confidential and not published by manufacturers. Therefore, the environment conditions are fed back into the RobustSENSE-Sensor. The resulting sensor yields the ordinary output data augmented with probabilistic assessment such as uncertainty of the measurement, or the clutter probability in the field of view. Examples are given in Section V.

B. Data Fusion Layer and Environment Condition Assessment

The Data Fusion layer contains functional modules and a high level fusion module for data fusion, and a newly introduced Environment Condition Assessment module (cf. Fig. 3). During processing, input data is propagated and consequently an environment model containing probabilistic features, e.g. existence probability or state variance of objects and their relations, is built.

The functional modules have access to the whole sensor data stream and the main aim is to combine strengths of different types of sensors, e.g. the velocity measurement of radars and angular resolution of lasers. Functional modules tackle specific tasks like ego motion estimation with a strict no-feedback structure, to avoid self-sustaining trends.

In High-Level Fusion, on the other hand, the estimates of the functional modules can be optimized using data from all other functional modules, e.g. by performing ego-motion and localization compensation. The module also resolves ambiguities originating from functional modules and puts data in relation.

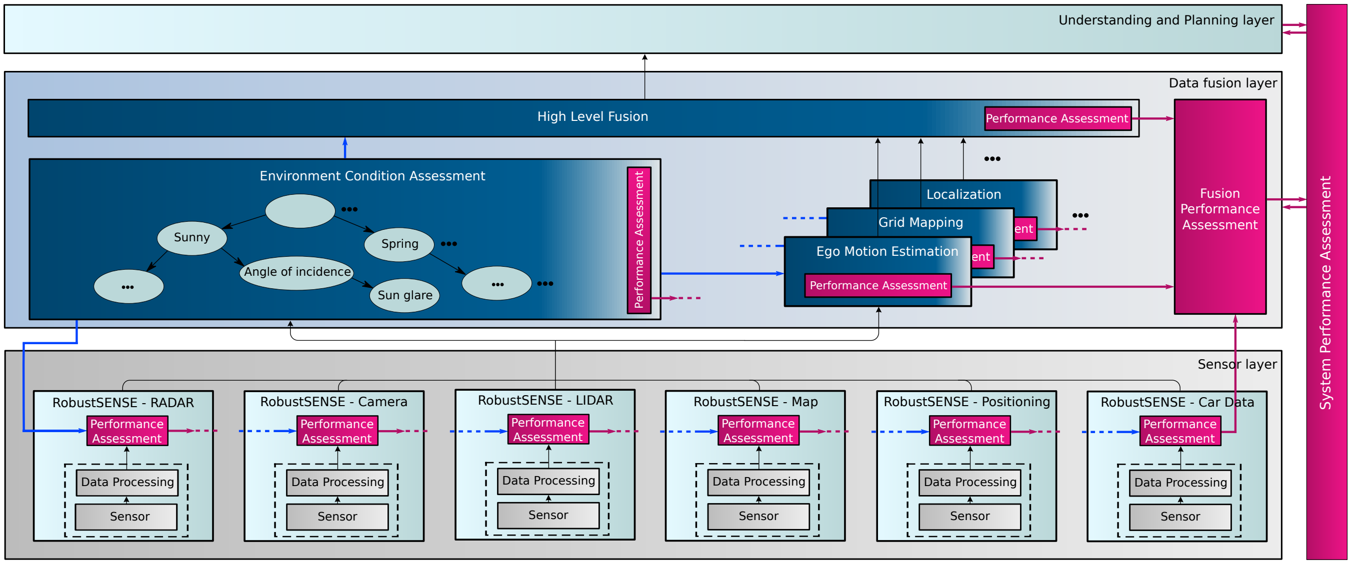

Fig. 2: A system architecture that is capable of assessing the performance of the Sensor layer. The performance assessment modules attached to individual sensors receive information on the environmental conditions from the Environment Condition Assessment module and send status information to the Fusion Performance Assessment module.

The RobustSENSE strategy of being adaptable to varying weather and light conditions is addressed by the Environmental Condition Assessment module. The RobustSENSE-Sensors are informed by this module to consider current conditions in the measurement models. This way, the sensor performance data can be adjusted, e.g. by increasing clutter probability, or the sensor can switch the operation mode to maintain its performance.

The Fusion Performance Assessment module receives its input by the performance assessment modules of sensors, the Environment Condition Assessment, individual functional modules, and the High Level Fusion. The module evaluates the validity of the environment model in a holistic manner to express the trustworthiness of the resulting environment model. The assessment is based on the confidence, existence probability and consistency of the fused data. Statistical tests comparing predicted measurements with actually obtained measurements can be used for module consistency assessment [17].

C. Understanding and Planning Layer

The Understanding and Planning layer consists of Scene Understanding, Situation Prediction, Behavioral Planning and Trajectory Planning modules that are mainly processed sequentially and thereby enrich the fused environment model by additional interpretations and conclusions (cf. Fig. 4). The final result is a trajectory decision that is transmitted to the vehicle controllers.

The Scene Understanding module provides situational awareness by identifying relations and interactions among traffic participants and traffic infrastructure [18]. Interrelating object motion models, which include discrete behavior classes such as braking in front of an intersection or driving right through, are estimated using the relational information.

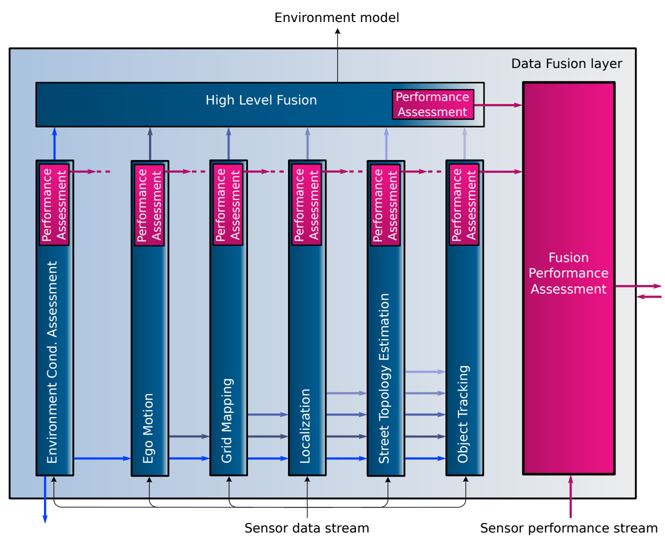

Fig. 3: Overview of an exemplary Data Fusion process chain. A functional module can contribute to other modules, as long as it is on a lower hierarchical level. This restriction is done to prevent inner feedback loops. The order and the components of the process chain can be changed.

The refined environment model for the current scene serves as a basis for situation prediction.

The Situation Prediction module utilizes the dynamic environment, street topology as well as legal information like speed limits or rules to give way from the environment model and couple these with interpreted relations. Compared to other modules, this module plays a more important role on the overall performance assessment. If there are clear contradictions between the predictions of the objects, or if the predictions are not consistent with the previous results, the whole automated driving system is switched into a degraded operation mode.

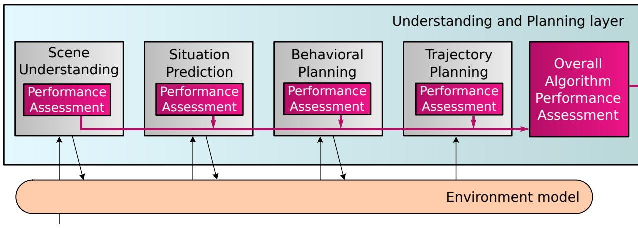

Fig. 4: The Understanding and Planning layer consists of Scene Understanding, Situation Prediction, Behavioral Planning and Trajectory Planning modules which process and enrich the environment model.

The Behavior Planning is responsible for creating a set of behavior options and choosing one according to the acceptable uncertainties, system performance status and postulations of the current traffic situation. It serves as a *master* module guiding the trajectory planner into a certain solution-space. The variation in number of alternative behaviors and their regarding costs can for example be utilized as a metric by the performance assessment module.

The Trajectory Planning module calculates safe trajectories using the confidence values of the environment model and choosing the confidence bounds according to the output of the System Performance Assessment. It employs metrics reflecting how hard it is to find an acceptable solution. If numerical methods are utilized for trajectory planning, metrics on convergence properties can be utilized. In case of sampling-based planning algorithms, number of admissible samples, or for example in case of RRT* [19] the number of parent node changes after rewiring, can be used as a metric instead. By receiving input from the central System Performance Assessment module and the uncertainties of the environment model, the planner adapts its confidence bounds.

D. System Performance Assessment

The System Performance Assessment module continuously evaluates the performance status of the whole vehicle by processing its inputs from the Data Fusion and Understanding and Planning layers to create an awareness of the system's performance and to be able to react by either function degradation or component adaptation. Decision making algorithms utilize the information about the source of degradation and return evaluate system degradation for the current situation of the environment.

The Fusion Performance Assessment module and the Overall Algorithm Performance Assessment module deliver an abstract, qualitative understanding of the layers' performance. Thus, only a small set of performance measures have to be evaluated in the System Performance Assessment module. Performance values like the detection of failure cases can be inferred in different ways, e.g. *rule-based* as described in our previous work for small-scale vehicles [20] or *probabilistically* as will be described in Section IV.

IV. Implementation and Experiments

In our experiments we tackle the scenario of a ghost object originated from clutter in a sensor, e.g. due to bad weather conditions. Fig. 5 illustrates the considered car following scenario. An automated vehicle follows two leading vehicles, while a slower third object, the ghost object, is falsely detected in between the leading vehicles. In our experiments vehicle 1 hits the ghost object but drives through it without any physical impact to its trajectory. Before the crash, the

Fig. 5: The considered scenario: A ghost object appears in a car following situation, between vehicle 1 and vehicle 2.

ego vehicle should be aware that the scene is inconsistent because of the abnormal behavior of vehicle 1 considering the ghost object - in the best case already inferring a high probability of having a ghost object. As soon as vehicle 1 drives through the ghost object the ego vehicle should be completely aware of that object being a ghost object.

To exemplarily implement this desired system behavior using the proposed system architecture, we put a focus on the following components: Sensor Performance Assessment, Fusion Performance Assessment, Scene Understanding Performance Assessment, and Trajectory Planning Performance Assessment. The performance measures are passed through the different performance assessment modules and a probability for having a ghost object is derived.

For this experiment we recorded trajectories of objects in car following scenarios using radar sensors. For recording we used the automated vehicle of Ulm University [21], equipped with a differential global positioning system and two radar systems, LRR3 and ARS300. In particular, in a car following scenario, the radar sensors' capabilities to detect occluded cars is an advantage compared to lidar sensors. The LRR3 radar field of view covers up to $250\,\mathrm{m}$ with an opening angle of $30°$ at $12.5\,\mathrm{Hz}$. The used ARS300 provides radar units for near distant (up to $60\,\mathrm{m}$) and far distant (up to $200\,\mathrm{m}$) objects. The opening angles are $56°$ (near) and $17°$ (far) with a $15\,\mathrm{Hz}$ update rate. The recorded data was filtered in offline post processing by removing clutter, labeling target measurements to objects and fitting trajectories in the noisy measurements. The post processed trajectories are considered as ground truth for a simulated Sensor and Data Fusion layers.

A. Simulated Sensor and Fusion

In the experiment, the perception stage sensor clutter probability and tracked object uncertainty were simulated. The sensor clutter probability was always set high because of the harsh weather conditions in the scene. For testing the ghost object detection we artificially modified the environment model.

In the first case, due to propagation of sensor clutter probability, the ghost object has low existence probability.

This is the way how a good probabilistic implementation should handle sensor uncertainties and thus suppress falsely detected ghost objects in subsequent understanding and planning modules.

Nevertheless improbable events can still occur and thus it can happen that a lot of false detections result in a high existence probability of the ghost object. Thus, in the second case, we simulate a high existence probability for the ghost object, causing the subsequent modules to consider it in processing. Probabilistic modelling alone cannot solve this and an additional system wide performance assessment is necessary to detect the algorithm failure.

B. *Particle based Scene Understanding and Prediction*

We use a particle based scene understanding and situation prediction algorithm in these experiments [22]. In scene understanding, the behavior defining parameters of a car following model, the Intelligent Driver Model [23], are estimated for tracked objects. In the subsequent situation prediction, the particles are forward propagated using the same model and a noise process. Performance assessment is related to the resampling process, in the parameter estimator. Since resampling is performed at every update step, every particle has the same weight $\frac{1}{\mathrm{N}}$. The importance of a particle after resampling is drawn by the number of duplications. Ideally, every particle has the same importance, leading to a high effective number of particles. However, if the filter diverges and the underlaid model does not fit to the real observation, the number of unique particles decreases. We consider a model to be suitable when the ratio between unique particles $\hat{\mathrm{N}}$ and the total number of particles N is greater than 0.5 and define a metric for model suitability with

$$\eta_{\mathrm{IDM}} = \min\left(1, \frac{2\hat{\mathrm{N}}}{\mathrm{N}}\right). \tag{1}$$

C. *Trajectory Planning*

For trajectory planning we use a local-continuous optimization based planner, as presented in [24]. The planning problem is modeled as a quadratic objective function with nonlinear constraints. The metrics we define in the following are mainly for a numerical solution based approach. However, the approach we present can be adapted to other algorithms. We set the metrics so that they are normalized and 1 reflects a good performance, whereas 0 reflects bad performance. We choose the metrics as simple as possible and try to avoid applying transformations.

The first metric we define is based on the final cost of the planner. The value of cost reflects the *quality* of the planned trajectory. The cost value is typically low if the planner converges to a local minimum. However, in some cases even though the algorithm has converged, the cost function can still have high values due to environment conditions. In order to separate the plausible bounds from unacceptable values, we first clip the values and then choose the Sigmoid function as a feature

$$\eta_{\mathrm{cost}} = 1 - \frac{1}{1 + e^{c_{\mathrm{ref}} - c}}, \tag{2}$$

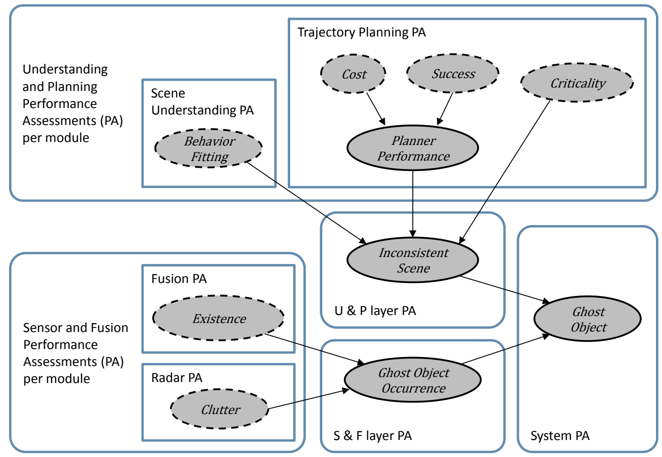

Fig. 6: The performance assessment is realized as a discrete Bayesian network combining the four exemplary performance measures to an overall estimation of the ghost object probability.

where c represents the current cost and c_{ref} the origin value.

Another metric for evaluating the algorithmic performance of the planner is the ratio of number of cost minimizer iterations over total number of iterations

$$\eta_{\mathrm{succes}} = \frac{n_{\mathrm{minimizer}}}{n_{\mathrm{total}}}. \tag{3}$$

This metric when treated together with η_{cost} reflects whether the problem is ill-formed or not.

A further metric for evaluating the performance of the planner given the current situation is an analysis on required deceleration. Traffic participants are proven to behave cooperatively hence not requiring other participants to apply full braking [25]. An instantaneous requirement of hard braking, when considered together with clutter and existence probabilities, can be used as an indicator of ghost objects. Such an analysis is complementary to the cost returned by the planner. If hard braking is indispensable for collision avoidance, then the optimization based planner will not be able to deliver low cost values as the acceleration and jerk is penalized.

To determine the criticality, we utilize the naïve approach of finding the minimum required braking accelerations and normalize it to the maximum admissible value a_{max}

$$\eta_{\mathrm{criticality}} = 1 - \frac{a_{\mathrm{req}}}{a_{\mathrm{max}}}. \tag{4}$$

A more detailed criticality analysis can be done by incorporating sensor uncertainties [26], but this lies out of the focus of this paper.

D. *System Performance Assessment via discrete Bayesian Network*

As previously mentioned, inferring the overall system performance from qualitative discrete performance measurements can be realized in different ways. In this work, we

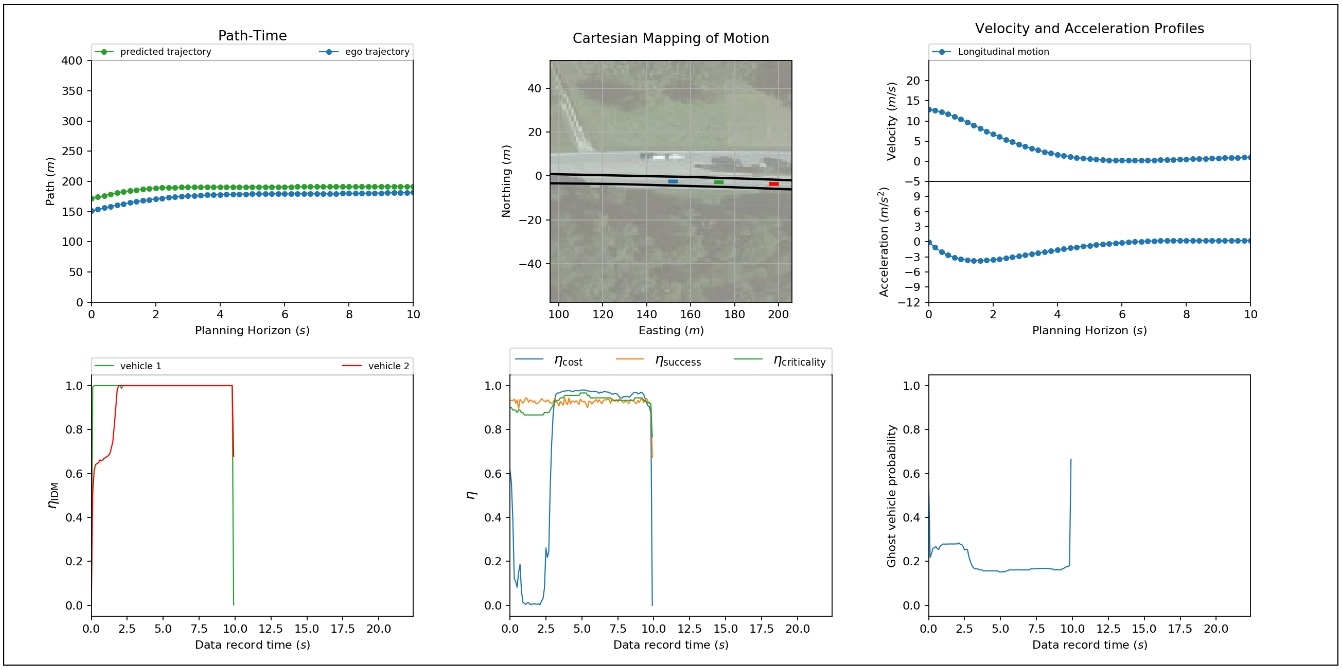

Fig. 7: Results of the online evaluation in the case of a falsely detected ghost object with high existence probability. The ghost object appears at $9.804\,\mathrm{s}$ resulting in a bad match of the behavior model for vehicle 1 (bottom left). The performance metrics of the planning algorithm indicate that there is an anomaly (bottom center). The performance assessment using a Bayesian network to combine all performance metrics infers a high ghost object probability (bottom right), and the vehicle 2 turns out to be a ghost object. The low performance values of behavior model at the very beginning are due to initialization. Because the existence probability is always set high, the ghost object probability is not less than 0.336 over the entire record.

chose a discrete Bayesian Network because of its ability to factorize a big, hardly modelable conditional dependency into sub problems with less complexity.

The graph in Fig. 6 describes the joint probability distribution as a factorization of local conditional probabilities using the hidden variables Ghost Object Occurrence, Planner Performance, Inconsistent Scene and Ghost Object. We modeled the conditional probabilities by expert knowledge, but also machine learning can be applied to sub problems to prevent wrong assumptions. Using the Bayesian network the a posteriori Ghost Object probability can be inferred given the per module performance metrics described in Section IV-A to IV-C. This Bayesian network is distributed into the three modules Fusion Performance Assessment, Overall Algorithm Performance Assessment and System Performance Assessment and thus directly reflects the factorization in the system architecture.

E. Results

The results are demonstrated in several sequences and visualized on a GUI (cf. Fig. 7). A video visualizing the results is also provided[1]. Scene understanding estimates behavior parameters of the first leading vehicle approaching the ghost object, resulting in low precision of the a posteriori distribution and consequently in a low value of the performance metric. The subsequent situation prediction provides results with high variance. Without system performance assessment, trajectory planning will evaluate a critical situation and will brake to reduce criticality. This will return low performance of the planner. However, this result is consistent with the low performance value of the scene understanding and hence indicate successful operation of the planner. By incorporating planning and understanding performance values to the clutter and existence probabilities, we achieved a unified assessment and resolved the use-case of confronting ghost objects.

V. Impact to Future Automated Driving

The holistic probabilistic processing with incorporated self-monitoring starts in the lowest layer, the Sensor layer. Since the performance of sensors highly depends on the environment, in particular the weather conditions, it is necessary to provide fused environment conditions from the sensor fusion layer to sensor post processing. The Environment Condition Assessment can for example refer to rain sensors, weather forecast and the relative sun position.

Since the sensor performance assessment is done within a RobustSENSE sensor, manufacturers do not have to publish sensor models, but can adapt the model to the current conditions. Examples for sensor model adjustments based on sensing conditions are utilized in lidar, radar or camera. Lidar sensor often suffer from rain and reflections on a wet road surface, increasing the clutter probability. The detection probability of static objects, as well as the existence

[1] `http://url.fzi.de/robustsense_pa`

probability of detected static objects largely decreases for single beam mono pulse radars if the sensor is itself still standing [27]. Cameras might even drop out when directly facing the sun (cf. Fig. 8). These effects can easily be taken into account with Environment Condition Assessment.

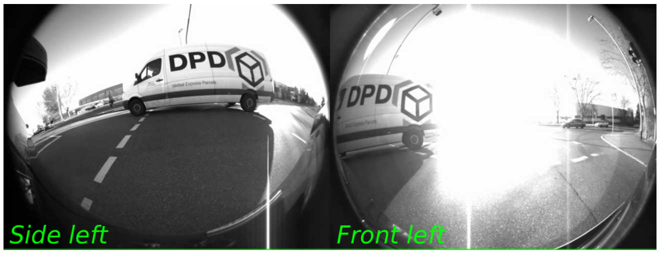

Fig. 8: An exemplary situation highlighting the benefits of sensor performance assessment. Object detection probability for the front left fish-eye camera highly decreases considering the relative position of the sun provided by Environment Condition Assessment.

Further processing can adapt to the augmented sensor information. Object tracking with multiple sensors, e.g., can provide an enriched existence probability based on detection probabilities and clutter probabilities of different sensors. Scene understanding, as well as situation prediction can run parallel relative motion hypothesis when the existence probability of an object is low. Consequently, the planning modules exploit this information for a decision or behavior adaption.

Finally, the consequent implementation of performance assessment will introduce performance awareness to autonomous systems making them able to react with function degradation or algorithm adaptation to handle more environment conditions and increase the robustness of automated driving.

VI. Conclusions and Future Work

In this paper we presented a guideline description for achieving robust and reliable operation of automated vehicles. We defined metrics and presented methods which can be utilized for performance evaluation. By means of these metrics, parallel to robustness and reliability of individual algorithms we introduced a system-wide performance awareness that relies on the interdependencies of the subsystems. Such a system maintains performance awareness and by providing feedback to the individual system modules allows the vehicle to adapt its function and algorithms to the current driving condition.

We carried out experiments on real trajectory data, where we demonstrate the benefits of our system with the very common scenario of false object detection due to clutter measurements in a car following scenario. Propagating probabilistic performance assessment from sensor to trajectory planning resolves the scenario where a radar sensor is aware of a high clutter probability. Using the introduced System Performance Assessment based on Bayesian inference, potential response of different submodules in understanding and planning layer are considered and we reliably resolved the situation of a falsely detected ghost object.

Our future work focuses on the implementation of the presented approach in the RobustSENSE automated vehicle demonstrators. Algorithms utilizing these metrics and making decisions under uncertain information received by the presented metrics will constitute the basis of our research. We will further focus on Environment Condition Assessment and utilize its output in the RobustSENSE-Sensor together with renowned automotive suppliers.

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] Ö. Ş. Taş, F. Kuhnt, J. M. Zöllner, and C. Stiller, "Functional System Architectures towards Fully Automated Driving," in *IEEE Proc. Intell. Veh. Symp.*, 2016, pp. 304–309.

[3] "RobustSENSE – Robust and Reliable Environment Sensing and Situation Prediction for Advanced Driver Assistance Systems and Automated Driving – Project," 2016, [Retrieved: March 3, 2017]. [Online]. Available: http://www.robustsense.eu

[4] G. Niu, *Data-Driven Technology for Engineering Systems Health Management*. Springer.

[5] D. Wang, M. Yu, C. B. Low, and S. Arogeti, *Model-based health monitoring of hybrid systems*. Springer, 2013.

[6] E. D. Dickmanns, R. Behringer, D. Dickmanns, T. Hildebrandt, M. Maurer, F. Thomanek, and J. Schiehlen, "The seeing passenger car'vamors-p'," in *Proc. IEEE Intell. Veh. Symp.*, 1994, pp. 68–73.

[7] J. S. Albus, "4d/rcs: A reference model architecture for intelligent unmanned ground vehicles," in *AeroSense 2002*. International Society for Optics and Photonics, 2002, pp. 303–310.

[8] ——, "Metrics and performance measures for intelligent unmanned ground vehicles," DTIC Document, Tech. Rep., 2002.

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

[10] 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.

[11] 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.

[12] 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.

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

[14] S. Ulbrich, A. Reschka, J. Rieken, S. Ernst, G. Bagschik, F. Dierkes, M. Nolte, and M. Maurer, "Towards a functional system architecture for automated vehicles," *arXiv preprint arXiv:1703.08557*, 2017.

[15] 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.

[16] 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.

[17] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[18] F. Kuhnt, J. Schulz, T. Schamm, and J. M. Zöllner, "Understanding Interactions between Traffic Participants based on Learned Behaviors," in *IEEE Proc. Intell. Veh. Symp.*, 2016.

[19] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the rrt," in *Proc. IEEE Int. Conf. Robot. and Autom.*, 2011, pp. 1478–1483.

[20] F. Kuhnt, M. Pfeiffer, P. Zimmer, D. Zimmerer, J. M. Gomer, V. Kaiser, R. Kohlhaas, and J. M. Zöllner, "Robust environment perception for the Audi Autonomous Driving Cup," in *Proc. IEEE Intell. Trans. Syst. Conf.*, Nov 2016, pp. 1424–1431.

[21] 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.

[22] S. Hoermann, D. Stumper, and K. Dietmayer, "Probabilistic long-term prediction for autonomous vehicles," in *IEEE Proc. Intell. Veh. Symp.*, 2017.

[23] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.

[24] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for Bertha — a local, continuous method," in *Proc. IEEE Intell. Veh. Symp.*, 2014, pp. 450–457.

[25] M. Treiber and A. Kesting, "Traffic flow dynamics," *Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg*, 2013.

[26] J. E. Stellet, P. Vogt, J. Schumacher, W. Branz, and J. M. Zöllner, "Analytical derivation of performance bounds of autonomous emergency brake systems," in *Proc. IEEE Intell. Veh. Symp.*, 2016, pp. 220–226.

[27] M. I. Skolnik, *Introduction to Radar Systems*, 3rd ed. New York: McGraw Hill Book Co., 2001.