Proceedings of the 33rd European Safety and Reliability Conference (ESREL 2023)
Edited by Mário P. Brito, Terje Aven, Piero Baraldi, Marko Čepin and Enrico Zio

Published by Research Publishing, Singapore.
doi: 10.3850/978-981-18-8071-1_P348-cd

Overall Markov diagram design and simulation example for scalable safety analysis of autonomous vehicles

Ivo Häring, Sunil Kumar Reddy Mopuru[1], Teo Puig Walz, Mayur Dhanani[2], Nikhilesh Sandela[3], Jörg Finger, Georg Vogelbacher, Fabian Höflinger, Aishvarya Kumar Jain, Alexander Richter

Fraunhofer EMI, Germany. E-mail: {ivo.haering; sunil.kumar.reddy.mopuru; teo.puig.walz; mayur.dhanani; nikhilesh.sandela; joerg.finger; georg.vogelbacher; aishvarya.kumar.jain; alexander.richter}@emi.fraunhofer.de
[1]Master student Embedded Systems Design, Hochschule Bremerhaven
[2]Master student Renewable Energy Systems, Nordhausen University of Applied Science
[3]Master student Digital Engineering, Otto-von-Guericke University Magdeburg

Konstantin Kirchheim

Otto-von-Guericke University. Email: konstantin.kirchheim@ovgu.de

Markov models are a promising tool regarding the assessment of availability, safety, security, and reliability of autonomous driving functions. The paper addresses challenges regarding the overall system functional and static modeling and related overall Markov diagram design options. To this end, the model space is presented, extending the main functions consisting of extended sensory system, decision and control, and vehicle platform manipulation. Sample transition models from literature are used. It is shown how to color-label overall Markov system product states in terms of the level of their criticality, independent of the multiplicity of failures. This is used to model the effect of structural and functional redundancies, e.g., of redundant sensors and sensors of different technology. The modeling approach allows to compare the effect of redundancy options on a systemic level, as well as to identify the need for further aggregation or subdivision of Markov states or refinement of the transition modeling and simulation approach. For instance, in terms of providing statistical assessment of historic events or by using simulation results of specific autonomous driving scenarios, e.g., interaction with vulnerable road users in case of darkness, bad weather, and partial sensor degradation. The paper presents Markov modeling results with a focus on modeling of redundancies of sensors.

Keywords: Functional and structural architecture, Markov model, safety and reliability analysis, fail-operational, fail-safe, redundancy, autonomous driving function.

1. Introduction

Autonomous vehicle (AV) architectures including interfaces with the environment are increasingly dominated by software functions. Architectures rely on redundant self-monitoring subsystems. Sensor technology types, their capabilities and multiplicities are increasing, including camera (Weber et al. 2023), lidar (Abbasi et al. 2023), radar (Zhou et al. 2022), and ultrasound or even audio (Furletov et al. 2022).

To understand which subsystems are relevant for providing an autonomous driving (AD) function, within classical system analysis, a stepwise approach is conducted, e.g. consisting of the following tabular and matrix analysis steps:

(i) System structural analysis to identify subsystems. (ii) Generation of subsystem dependency matrix (design structure matrix, DSM) (Felgen et al. 2005) to determine subsystems that are related with each other. (iii) System functional analysis to identify functions and subfunctions on system level (see e.g. short overview in (Halbe 2021)). (iv) Generation of function dependency matrix to determine subsystems that are related with each other. (v) Matrices to determine which subsystems are needed for which function. (vi) Matrices to determine which subfunctions are needed for system functions.

The question that arises is which subsystems and subfunctions should be considered for modern autonomous systems. The present paper will provide a structural-functional analysis of modern autonomous driving architectures by reviewing recent publications and conducting mainly steps (i) and (iii). Also, steps (iv) and (v) will be conducted to identify the main dependencies of functions on system level on subsystems and subfunctions. When compared to the systemic functional resilience approach in (Fehling-Kaschek et al. 2019) (Häring et al. 2021), the presented steps are closer to classical system design analysis approaches as provided in (Browning 2001) (Eickhoff 2009) (Browning 2016) (Häring 2021).

A further challenge is the qualitative assessment of safety of autonomous driving functions assuming that it is known which subsystems and functions contribute. A natural transition from knowing which subsystems constitute to a system or which subfunctions constitute to a system for an assessment on system level is to design a product state space and to verify for which combinations of subsystem states the states of the overall system become critical. For instance, if sufficient sensory subsystems fail it can be concluded that a given autonomous driving function will fail. The system behavior with dependence on the subsystems or subfunctions can be summarized in a system truth table (see e.g. (Rahmat et al. 2006) (Thornton et al. 2014)). The present paper will provide a truth table for the sensing system of an autonomous driving system.

For quantitative assessment an option is to use Markov modelling and use the truth table for applying a color-labeling, i.e. labeling of all overall state space elements on system level. The challenge is the state-space explosion issue and hence exponentially increasing computation times. This holds true even more than in the case of product state identification of autonomous functions and their color-labeling. The latter can be conducted using automatic application of rules. The present paper will explore if for simple exponential transition models, state space explosion is already a challenge for the modelling of autonomous driving functions.

To this end, an autonomous driving sensing subsystem will be simulated without environmental information. Based on this example a tabular estimate will be given on the number of states necessary on system level to model autonomous driving functions using extended Markov models including propagation of environmental information (scenario or setting information) as proposed in (Häring et al. 2022).

The paper is organized as follows. Section 2 gives further background on methodology to determine a representative functional-structural architecture of autonomous vehicles and in an informed way a Markov diagram from this information as well as its numerical simulation. Section 3 presents and discusses the functional-structural architecture proposed. Section 4 provides the Markov diagram and a truth table excerpt of a sample sensing system of an autonomous vehicle and sample simulation results along with estimates on necessary states for overall abstract extended Markov autonomous vehicle modelling and simulation. Section 5 offers a conclusion and an overview of future directions.

2. Methodology

Based on the functional structural analysis, within a classical Markov approach, the next steps are, e.g., (vii) Definition of product state space, i.e. subsystem identification and definition of their states. Development of a truth table on system level. (viii) Markov diagram generation and Markov state transition identification (allowed transitions). (ix) Transition modelling and quantification. (x) Quantitative Markov model generation (initial conditions and equation systems). (xi) Numerical state space propagation and computation of safety and reliability quantities. (xii) Visualization, evaluation, and assessment. (xiii) Identification of improvement options.

The paper mainly contributes to the steps (i) and (iii) by presenting semi-formal diagrams, and to steps (vii) to (xiii) by presenting and explaining the used Markov diagram structure, the transition models used, and the equation system. For the latter steps, mainly results are presented and discussed in terms of Markov state space-time histories and probability distributions of states at given assessment times. Note that the transitions from steps (v) and (vi) to steps (vii) to (ix) are not formalized.

The scalability of the approach is argued in terms of how to generalize the example and use a tabular assessment of state space increase estimates.

3. Functional-structural architecture of AVs

3.1. *Comparison of AD architectures in literature*

ISO 26262 (Functional Safety – Road vehicles) (Kirovskii and Gorelov 2019) demands a quantitative assessment of vehicle safety. To enable such assessments, it is necessary to understand the architecture of the vehicle in full detail. Therefore, AV functional architecture is a central point of discussion in many studies and standards.

For instance, Novickis et al. (2020) provide a viable functional architecture for an example of AV functions that include lane keeping, parking assistance, or autopilot driving mode on the highway. This architecture also shows which sensors would be necessary for each of these functions. With a little less detail in the relations between sensors and functions, Munir et al. (2018) also present an AV architecture with a similar classification of high-level functionality (i.e. sensors, perception, planning, and control). In this case, the system is further divided into two main parts: software and hardware. The software part includes the control block with the localization, detection, motion, and mission planning modules, while the hardware part includes all the actuators.

Along the same lines as the previously mentioned papers, Serban *et al.* (2018) propose an AV architecture with a different level classification of functions. In their approach, they distinguish between functional components, class of components, and sub-class of components. It provides high-resolution data management systems consisting of different elements like sensor abstraction, data management, actuator, and safety management. In addition, data management support functions are given (e.g. map database). This provides input for the main layer of the proposed functional and structural architectures.

Often papers that cover AD functional and structural architecture have a focus on the assessment of the reliability, safety, and security of AV. In those cases, they show how each component affects the overall safety of the vehicle. An example of such an approach is Bhavsar et al. (2017), who conduct risk analyses by risk identification, risk estimation, and evaluation of a fault tree model.

Sagar Behere and Törngren (2015) provide a functional and structural data flow architecture with a high level of detail of how vehicle movements are defined through trajectory execution with the help of decision and control and other functions like sensing, localization, semantic understanding, and sensor fusion.

The advantages of utilizing a multi-class driverless vehicle fleet for mobility on demand have been discussed and demonstrated through successful experimental operations in a proposed functional architecture in the paper by Pendleton et al. (2016).

Functional architecture of autonomous driving with a focus on different functional sub-systems and how the data is transferred between these systems is explained in detail by Ahangar et al. (2021).

Novickis et al. (2020) proposed a functional architecture for autonomous driving vehicles with perception, sensors with localization of the world model, and how the vehicle platform manipulation is achieved by trajectory generation.

In the paper by Tas et al. (2016), a robust structure for future system architectures is derived by summarizing the existing system architectures from the current literature and investigating them with regards to their robustness against measurement inaccuracies, failures, and unexpected evolution of traffic situations.

A holistic architecture for autonomous on-road motor vehicles is proposed by Matthaei and Maurer (2017), who extend existing architectures by systematic integration of external data, such as map data and V2X (vehicle to everything) information. The consideration of bidirectional communication also allows the implementation of automated map updates.

Shah (2019) describes an award-winning functional architecture of AV according to Autonomous Vehicle Competition in Korea. In this architecture, a basic layout of all the critical sensors and systems leading to the actuation and movement of the vehicle is shown.

In Table 1 a summary of the reviewed publications is presented with the components or functions that each study includes.

3.2. Functional and structural architecture concept

According to (Collin et al. 2020), functional architecture is a system of interconnected tasks that communicate through messages. It can be represented as a graph, with tasks as nodes, and messages as edges. The design of this architecture can be informed by using techniques from network analysis, such as functional dependency network analysis. Thus, functional architecture is a design pattern that emphasizes the separation of concerns into independent functions. In functional architecture, the application logic is organized around functions that perform specific tasks, rather than components that represent each task and their interactions.

Table 1. Main functions and subsystems of autonomous driving architectures.

Function or component	**Munir et al. 2018**	**Serban et al. 2018**	**Behere and Törngren 2015**	**Pendleton et al. 2016**	**Ahangar et al. 2021**	**Novickis et al. 2020**	**Tas et al. 2016**	**Bhavsar et al. 2017**	**Matthaei and Maurer 2017**	**Shah 2019**	**Sum**
Perception	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**<u>10</u>**
Sensors	X	X	X	X	X	X	X	X	X	X	**10**
Sensor data fusion		X	X	X	X	X	X	X	X	X	**9**
Localization	X	X		X	X	X	X	X	X	X	**9**
Semantic understanding		X	X	X	X	X	X		X	X	**8**
Detection		X	X	X	X	X	X		X		**7**
Decision and control	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**<u>10</u>**
World model		X	X	X		X	X	X	X		**7**
Trajectory generation	X	X	X	X	X	X	X	X	X		**9**
Safety management		X	X		X			X	X		**5**
Vehicle platform manipulation	**X**	**X**	**X**	**X**	**X**	**X**	**X**		**X**	**X**	**<u>9</u>**
Trajectory execution	X	X	X	X	X	X	X	X		X	**9**
Platform stabilization		X	X	X		X		X	X		**6**
Vehicle control	X	X	X	X	X	X	X	X	X	X	**10**
Actuation	X	X	X	X	X	X	X		X	X	**9**
Hardware components (steering, brake, acceleration, etc.)	X	X	X	X	X	X	X		X	X	**9**

In contrast, the physical architecture includes components such as processors and data buses. The structural data flow architecture focuses more on each component and how the data is transferred between them for the system to function automatically.

A similar approach is to use system functional trees and system structural trees as e.g. proposed within standards like IEC 61508, ISO 26262 and SOTIF ISO/PAS 21448 for documenting sufficient system knowledge before providing a generic functional and structural architecture (Kirovskii and Gorelov 2019).

3.3. *Generic functional-structural architecture*

Based on Table 1 the primary functional elements of the autonomous driving system's motion control component are divided into three major categories. This agrees with the main functions used in (Behere and Törngren 2015) as well as all relevant papers identified (Ahangar et al. 2021; Shah 2019; Serban et al. 2018).

These groups are: **Perception** of the outside environment or context in which the vehicle operates with external and internal sensors, including interfaces and generalized sensors like cameras, lidar, radar, and GPS. **Decision and control** of the vehicle's long- and short-term trajectory planning based on the perceived external environment. **Vehicle platform manipulation**, primarily through the actuation of the ego vehicle to attain the desired motion.

In Fig. 1, sub-functions of these main functions are given which cover current modern, and expectable future designs by comparing the most recent publications. This architecture is a synthesis of the different approaches analyzed in section 3.1 and aims to provide a comprehensive representation of functional architectures presented in the research papers. It contains the highest resolution proposed to be used in building the Markov models. This architecture focuses mainly on the functions and sub-functionalities of AVs.

The structural architecture is presented in Fig. 2. It focuses more on the structural resolution of data flow showing the main subsystems and components and how the information is transferred along the components.

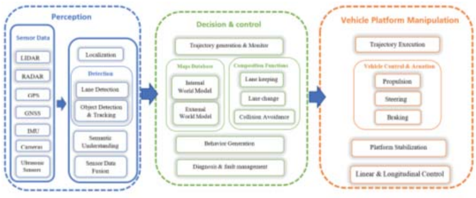

Fig. 1. Overall functional architecture of the autonomous driving vehicle.

This allows to identify which sub-systems and components are used to realize system functions. For instance, as Table 1 shows, functionality perception has been part of the AV architecture in the papers (Ahangar et al. 2021; Behere and Törngren 2015; Bhavsar et al. 2017; Pendleton et al. 2016; Serban et al. 2018). In a similar way, it is also possible to trace which functional architecture has been already used in literature and also where further information can be found.

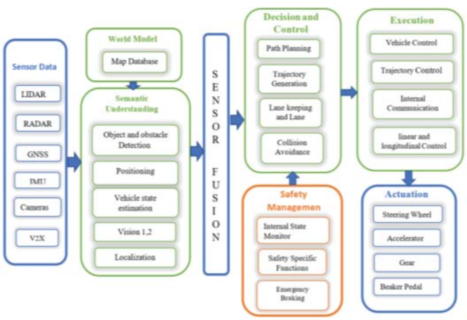

Fig. 2. Proposed structural data flow architecture of the autonomous driving vehicle.

4. Markov modelling and simulation of AV sensor system

4.1. *Scaling and color-labeling Markov models*

A 32-state prototype Markov model for the sensing subsystem can be developed comprised of five generalized sensors including the camera system, radar system, lidar system, GPS, and V2X communication system. If for each of the 5 sensor systems only two states are allowed, namely operational and failed, the state space cardinality is given by $2^5 = 32$.

However, the multiplicity (number of sensors) of each sensor technology and their perception direction is not considered. A minimum extension is to consider 3 forward and backward sensor systems for each technology. This results in a state space of $2^2 2^2 2^2 2^1 2^1 = 256$.

In a similar way, other subsystem redundancies can be considered, e.g. further resolving the forward sensor system of a given technology. Furthermore, more states per subsystem or subfunction could be considered than only operation and failed, e.g. adding degraded operational.

The color-labeling of system states needs to consider all subsystem or subfunction states. Based on the discrete finite product state, the labeling is conducted, e.g. (0,1; 0,1; 1,1; 1; 1) could label that the forward camera and radar system is degraded in forward direction, and only the lidar sensor system is fully available. This can be interpreted as fail operaional on system level.

4.2. *256-state Markov Sensor model*

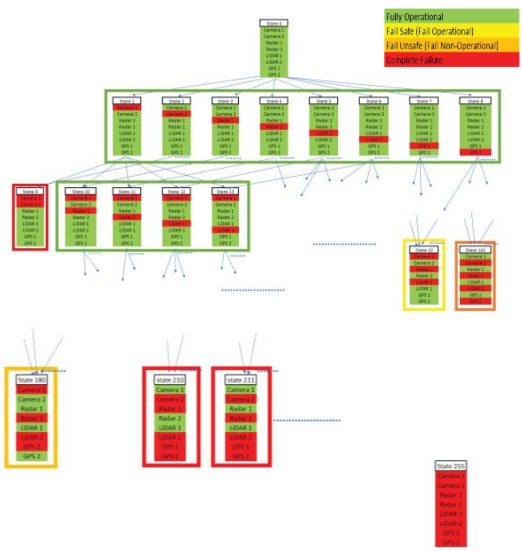

Fig. 3. Complex 256-state Markov model for sensor systems with added redundancy.

The 256-state Markov model in Fig. 3 represents various failure scenarios for a system comprised of 8 different components: Camera-1/-2, Radar-1/-2, LiDAR-1/-2, and GPS-1/-2. The model is relatively large, comprised of $2^2 2^2 2^2 2^2 = 256$ states, each representing to a specific failure scenario. The color-labeling of system states is indicated in Fig. 3.

The model accounts for single, double, triple, and multiple component failures. The model also includes absorbing states (red) when a set of sensors of the same type is failed. This double sensor failure leads to the complete system failure, from which it cannot recover. Hence, e.g. state 255 is never reached.

Note that each Markov transition only implies the failure or repair of a single component, e.g. Camera-1.

5.3. *Quantification and time-propagation of the model*

For quantification of the model, the initial conditions are to start at time $t = t_0 = 0$ h in state 0 (all subsystems operational) of Fig. 3. Table 2 gives sample transition rates based on the sample rates used in (Häring et al. 2022). This allows to determine the constant transition rate matrix (Q-matrix) for a continuous finite state space Markov model of the form $\frac{d}{dt}\overrightarrow{P(t)} = Q\,\overrightarrow{P(t)}$, where $\overrightarrow{P(t)} = \left(P_0(t), P_1(t), \dots, P_M(t)\right)^T$ is the time depend state vector and $M = 256 - 1$, see e.g. (Rausand 2013). The time propagation is conducted using adaptive Python scipy.odeint integration routine until $T = 50{,}000\,h$ with an integration relative failure tolerance of $1.5 * 10^{-8}$.

Table 2. Sample exponential failure and repair rates of sensor model.

Failure rates (λ)	
Camera 1	$10\ h^{-1}$
Camera 2	$0.58 \cdot 10^{-6} h^{-1}$
LiDAR 1	$10\ h^{-1}$
LiDAR 2	$0.752 \cdot 10^{-6} h^{-1}$
Radar 1	$1.59 \cdot 10^{-6} h^{-1}$
Radar 2	$1.59 \cdot 10^{-6} h^{-1}$
GPS 1	$6.70 \cdot 10^{-6} h^{-1}$
GPS 2	$6.70 \cdot 10^{-6} h^{-1}$
Repair rates (μ)	
Same for all components	$1 * 10^{-3}\ h^{-1}$

5.4. *Reliability and safety quantities results discussion*

The model can be used to analyze the probability of being in a Markov sate at any given time (Fig. 4.A), to determine the dominating (absorbing) asymptotic states (Fig. 4.B, Fig. 4.D), and to determine the overall system reliability (Fig. 4.C).

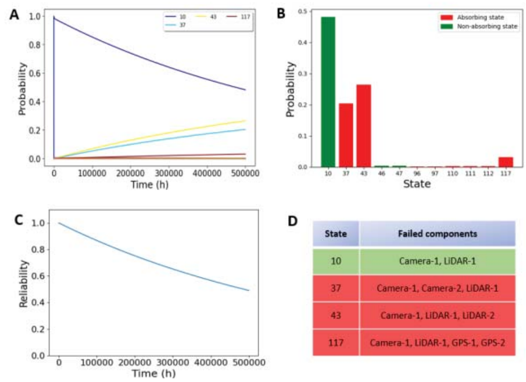

Fig. 4. Simulation results for the 256-sate redundant Markov model. A. Time evolution of the state probability distribution. B. State distribution at the end of the simulation. C. Reliability of the system over time. D. Most dominant states at the end of the simulation and their respective failed components.

It can be used to assess the impact of individual component failures on the system performance. In fact, the very high failure rates assumed for Camera-1 and Lidar-1 in Table 2, e.g. due to dirt affecting some front sensors, result in the results of Fig. 4. It shows that the nearest natural absorbing state is reached, e.g. additional failure of Camera-2 or Lidar-2, in case of pre-damage, degradation or weak initial performance sub-systems of Camera-1 and LiDAR-1.

The approach allows for the identification of critical components and the development of strategies to mitigate the propagation of failures, e.g. by increasing multiplicity of sensors that are not sensitive to identified critical failure root cause, e.g. rain or snowfall.

5. Conclusions

The paper demonstrated the scalability of the Markov modeling of autonomous vehicles if only the resolution of vehicle technical system is considered (sensors, hardware, (embedded) software, interfaces, actuators, and chassis). This can be inferred since as described for the 32-state (see section 4.1) and 256-state model (see section 4.1) in a similar way 4-state (e.g. using the 3 overall Markov state color-labeling of Fig. 3) and 8-state (e.g. using the 3 main functions on system level introduced in Fig. 1) and of course higher order state models can be constructed based on the structural-functional models introduced (Fig. 1 and Fig. 2).

The models can be understood as refinements (higher resolution) or abstractions (lower resolution) of each other showing in a non-formal way the scalability of the approach. Regarding control of sate-space explosion Table 3 gives an estimation of the maximum state space dimension needed when using the structural-functional model given in Fig. 1 at different resolutions.

Table 3. Sample Markov state space dimensions.

Type of subsystems/ -functions	Subs./ -funct.	State reso-lution	Markov sates
Main functions: Perception, decision and control, Actuation	3	2 to 4	8 to 64
Consideration of on average ca. 3 subsystems/ -functions per main function	9	2 to 4	512 to 262,144
Consideration of on average ca. 5 subsystems/ -functions per main function	15	2 to 4	32,768 to $1.\cdot 10^9$
Maximum resolution of Fig.1 in total 26 subsystems/-functions	26	2 to 4	$7.\cdot 10^7$ to $4.\cdot 10^{15}$

Hence state explosion is an issue (e.g. when considering that Lenovo Thinkpad laptop with Intel i7-4600 CPU and 8GB RAM. ca. 10,000 states are a limit when asking for simulation time of ca. 1 h) even if, as in the present case for the 256-state model, (a) the simulation does not need to consider non-reachable states, which reduces the total number of states that need to be propagated, (b) minimum state resolution of 2 is used, and (c) non-technical subsystems or subfunctions are not considered, e.g. driver, environment, other (vulnerable) road users.

This allows concluding that controlled abstraction and refinement of the Markov model for autonomous driving functions is a key capability that needs to be employed, e.g. by focusing on the sensory system only as conducted in the present approach.

Next steps could include showing in a formal way the scalability by requiring more defined relations between resolution levels, e.g. more abstract system boundaries are never lifted when going to finer resolution, and quantification approaches for abstraction and refinement. Also, the relation between a functional-structural semi-formal (graphical) model and discrete product state space could be formalized, as well as the visualization of color-labeling rules, e.g. using SysML/UML.

Acknowledgement

All authors but C. K. acknowledge funding within the German BMWK project Real Driving Validation (RDV), Funding Number: 19A21051D.

References

1. Abbasi, R., A. K. Bashir, Hasan J. Alyamani, F. Amin, J. Doh, and J. Chen. 2023. Lidar Point Cloud Compression, Processing and Learning for Autonomous Driving. *IEEE Transactions on Intelligent Transportation Systems 2023.*
2. Ahangar, M. Nadeem, Qasim Z. Ahmed, Fahd A. Khan, and M. Hafeez. 2021. A Survey of Autonomous Vehicles: Enabling Communication Technologies and Challenges. *Sensors (Basel, Switzerland)* 21.
3. S. Behere, and M. Törngren. 2015. A Functional Architecture for Autonomous Driving. In *Proceedings of the First International Workshop on Automotive Software Architecture 2015.*
4. P. Bhavsar, P. Das, M. Paugh, K. Dey, and M. Chowdhury. 2017. Risk Analysis of Autonomous Vehicles in Mixed Traffic Streams. *Transportation Research Record: Journal of the Transportation Research Board* 2625.
5. Browning, T. R. 2001. Applying the design structure matrix to system decomposition and integration problems: a review and new directions. *IEEE Transactions on Engineering Management 2001*.
6. Browning, T. R. 2016. Design Structure Matrix Extensions and Innovations: A Survey and New Opportunities. *IEEE transactions on engineering management 2016*.
7. A. Collin, A. Siddiqi, Y. Imanishi, E. Rebentisch, T. Tanimichi, and O.L. Weck. 2020. Autonomous driving systems hardware and software architecture exploration: optimizing latency and cost under safety constraints. *Systems Engineering 2020.*
8. Eickhoff, Jens. 2009. *Simulating Spacecraft Systems. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg 2009*.
9. M. Fehling-Kaschek, K. Faist, N. Miller, J. Finger, I. Häring, M. Carli, F. Battisti, R. Makri, G. Celozzi, M. Belesioti, and E. Sfakianakis. 2019. A systematic tabular approach for risk and resilience assessment and Improvement in the telecommunication industry. In *Proceedings of the 29th European Safety and Reliability Conference ESREL 2019*.
10. L. Felgen, F. Deubzer, and U. Lindemann. 2005. Complexity management during the analysis of mechatronic systems. In *International Conference on Engineering Design ICED 2005*.

11. Furletov, Yu. M., A. M. Ivanov, S. S. Shadrin, and M. A. Toporkov. 2022. Sound Source Direction of Arrival Estimation for Autonomous Driving Applications. In *Intelligent Technologies and Electronic Devices in Vehicle and Road Transport Complex 2022.*
12. Halbe, Johannes. 2021. Development of a Sustainable Systems Engineering Framework for the Design and Assessment of Sustainability Visions, *McGill University, Montreal 2021.*
13. I. Häring, 2021. Technical Safety and Reliability Methods for Resilience Engineering. In *Technical Safety, Reliability and Resilience: Methods and Processes*, *Springer 2021.*
14. I. Häring, M. Fehling-Kaschek, N. Miller, K. Faist, S. Ganter, K. Srivastava, A. Kumar Jain, G. Fischer, K. Fischer, J. Finger, A. Stolz, T. Leismann, S. Hiermaier, M. Carli, F. Battisti, R. Makri, G. Celozzi, M. Belesioti, E. Sfakianakis, E. Agrafioti, A. Chalkidou, G. Papadakis, C. Fuggini, F. Bolletta, A. Neri, G. Giunta, H. Scheithauer, F. Höflinger, D. Schott, C. Schindelhauer, S. Köhler, and I. Linkov. 2021 A performance-based tabular approach for joint systematic improvement of risk control and resilience applied to telecommunication grid, gas network, and ultrasound localization system. *Environment Systems and Decisions 2021.*
15. I. Häring, Y. Satsrisakul, J. Finger, G. Vogelbacher, C. Köpke, and F. Höflinger. 2022. Advanced Markov modeling and simulation for safety analysis of autonomous driving functions up to SAE 5. In *Proceedings of 32nd European Safety and Reliability Conference*.
16. Kirovskii, O. M., and V. A. Gorelov. 2019. Driver assistance systems: analysis, tests and the safety case. ISO 26262 and ISO PAS 21448. *IOP Conference Series: Materials Science and Engineering.*
17. M. Richard, and M. Maurer. 2017. *Functional System Architecture for an Autonomous on-Road Motor Vehicle: Springer 2017.*
18. M. Farzeen, S. Azam, M.I. Hussain, A.M. Sheri, and M. Jeon. 2018. Autonomous Vehicle. In *SSIP 2018*, 1–5, Prague, Czech Republic 2018.
19. R. Novickis, A. Levinskis, R. Kadikis, V. Fescenko, and K. Ozols. 2020. Functional Architecture for Autonomous Driving and its Implementation. In *2020 17th Biennial Baltic Electronics Conference (BEC)2020.*
20. S.D. Pendleton, H. Andersen, X. Shen, Y. Hong Eng, C. Zhang, H. Xun Kong, W. Kang Leong, M.H. Ang, and D. Rus. 2016. Multi-class autonomous vehicles for mobility-on-demand service. In *2016 IEEE/SICE International Symposium on System Integration 2016.*
21. M. Rahmat, S. Jovanovic, and Kwok Lo. 2006. Reliability Estimation of Uninterruptible Power Supply Systems: Boolean Truth Table Method. In *International Conference on Telecommunications Energy (INTELEC) 2006.*
22. M. Rausand. 2013. *Reliability of safety-critical systems: Theory and applications Springer.2013.* Hoboken, New Jersey: Wiley.
23. A.C. Serban, E. Poll, and J. Visser. 2018. A Standard Driven Software Architecture for Fully Autonomous Vehicles. In *IEEE International Conference on Software Architecture Companion (ICSA-C) 2018*.
24. S. Syed Asim. 2019. Safe-AV: A Fault Tolerant Safety Architecture for Autonomous Vehicles. In *Master Thesis, McMaster University, Hamilton, Ontario 2019.*
25. T. Omer Sahin, F. Kuhnt, J.M. Zollner, and C. Stiller. 2016. Functional system architectures towards fully automated driving. In *IEEE Symposium on Intelligent Vehicle 2016.*
26. Thornton, M. A., T. W. Manikas, S. A. Szygenda, and Shinobu Nagayama. 2014. System Probability Distribution Modeling Using MDDs. In *2014 IEEE 44th International Symposium on Multiple-Valued Logic 2014*.
27. W. Michael, T. Weiss, F. Gechter, and R. Kriesten. 2023. Approach for improved development of advanced driver assistance systems for future smart mobility concepts. In *Autonomous Intelligent Systems 2023*.
28. Z. Yi, L. Liu, H. Zhao, M. López-Benítez, L. Yu, and Y. Yue. 2022. Towards Deep Radar Perception for Autonomous Driving: Datasets, Methods, and Challenges. In *Sensors (Basel, Switzerland) 22.*