Chapter 8
Design and Safety Analysis of a Drive-by-Wire Vehicle

Peter Bergmiller

8.1 Increasing Complexity in Modern Vehicles

In the Federal Republic of Germany, more than 700,000 people were employed in the automotive industry in 2010. According to the German Federal Transport Authority (Kraftfahrt-Bundesamt), the industry branch generated an overall turnover of about 315 Billion Euros. Therewith, the automotive industry contributes hugely to the national output of Germany and thus is vital for the further economic success of the country (Legler et al. 2009). At the same time, the complexity of modern vehicles is continuously increasing and vehicle development is becoming more and more challenging. Additional and more complex functionalities from different domains (ergonomics, entertainment, etc.) are demanded by the customer (Sangiovanni-Vincentelli 2007). Adaptations and extensions of the electrics and electronics (EE) of the vehicle, especially the integration and interaction of previously independent functions, significantly contribute to meeting these demands (Arbitmann et al. 2011; Sinha 2011; Pruckner et al. 2012). Thus, the number of interconnections and interdependencies within the EE system rapidly increases on functional and hardware level (Schäuffele and Zurawka 2004). This furthermore pushes complexity.

In parallel, operational safety of modern vehicles has to be maintained, or better, improved. This generates strongly conflicting goals between additional functionalities and proof of sufficiently safe operation of these increasingly complex systems. With electric vehicles joining the market, meeting both targets becomes even more challenging. Depending on the drive train structure, electric vehicles can provide powerful means to intervene into vehicle handling, e.g., due to torque vectoring.[1]

[1] Torque vectoring refers to an approach where individual wheels of a vehicle are driven with individual drive torques. When driving the wheels at one side of the vehicle with a different torque than the wheels of the opposing side, an additional yaw moment is generated. For further information including evaluation of safety criticality see, e.g., Euchler et al. (2010).

P. Bergmiller
Institute of Control Engineering,Technische Universität Branuschweig, Hans-Sommer-Str. 66, D-38106 Braunschweig, Germany
e-mail: p.bergmiller@tu-bs.de

M. Maurer and H. Winner (eds.), *Automotive Systems Engineering*,
DOI: 10.1007/978-3-642-36455-6_8,

These means are controlled by the EE system, and thus a failure of the EE system can result in fatal crashes that are hardly avoidable even for a skilled driver. Consequently, profound further development in the field of vehicle electronics is crucial not only for the German automotive industry to maintain its leading global position based on innovative functionalities, but also to make driving safer. This necessity is strongly supported by a technical survey issued and funded by the Federal Ministry of Economics and Technology in Germany (BMWi) (Bernard et al. 2010). The survey especially demands fundamental reconsideration of the established electronics architecture in series vehicles in terms of performance, reduction of complexity and functional safety. This is regarded as a key factor for further technical progress in the field of vehicle electronics and economic success. Turning from the classical perception of the vehicle as a number of interconnected parts to a more holistic functional perception of the overall vehicle can be a way to achieve this goal. This approach is followed at the Institute of Control Engineering at TU Braunschweig (Maurer 2010) and also slowly starting to be adopted by companies (Abele 2012; Papadopoulos et al. 2001).

To support this development, this contribution introduces (a) a flexible experimental vehicle (MOBILE) for fundamental investigations in the field of vehicle electronics that is built up at TU Braunschweig; and (b) an hierarchical approach for focused evaluation of functional safety of vehicles with a high degree of functional redundancy, functional integration, and complexity due to by-wire control.[2] For MOBILE, especially the functional and hardware/software architecture of the drive-by-wire system are introduced. The applicability of the hierarchical approach is demonstrated by analyzing MOBILE in terms of functional safety. Thereby, a simplified hazard analysis according to ISO 26262[3] that is based on expected use cases of MOBILE delivers the safety goals[4] that have to be met by the design of the drive-by-wire system.

8.2 The Experimental Vehicle MOBILE

The experimental vehicle MOBILE is custom built by the Institute of Control Engineering and the Institute of Engineering Design at TU Braunschweig. The intended purpose of the vehicle is to serve as a tool for a variety of future research projects on vehicle dynamics and mechanical or electric/electronic components. Still, the vehicle itself is also subject to research activities. The highly safety critical drive-by-wire system in combination with a high degree of functional integration require novel approaches in system design. Resulting, costs for hardware units in the vehicle can

[2] In this context, "by-wire" control means that actuators in the vehicle are controlled purely electronically without any mechanical or hydraulic linkage between the actuator and the driver. MOBILE implements by-wire control for braking, steering, and the drive motors.

[3] ISO 26262: Road Vehicles—Functional Safety, edition 2011.

[4] According to ISO 26262, a safety goal is a "top level safety requirement as a result of the hazard analysis and risk assessment" (ISO 26262-1:2011, p. 14).

be reduced, and benefit for the driver is increased due to synergies between different functionalities. This contribution presents the basic actuator set-up of MOBILE and details the architecture of the part of the EE system needed for vehicle control. Other parts as the extended Human-Machine-Interface (displays, inputs other than brake and gas pedal), the battery management, the cooling system, and the knowledge management are neglected. These aspects are investigated and implemented in the MOBILE project but are not covered in this contribution. Accordingly, the following sections introduce the architecture[5] of the vehicle control function stepwise: Starting from general requirements, the architecture is detailed on different hierarchical layers and from different views.[6] Figure 8.1 outlines the steps for definition of the architecture. Thereby, requirements and constraints (step 1 in Fig. 8.1) guide the derivation of the basic mechanical and actuator set-up of the vehicle (step 2 in Fig. 8.1). Following, the steps 3 to 5 in Fig. 8.1 iteratively derive the functional/software and hardware architecture of the EE system on all hierarchical layers. Functional and software architecture partially merge at higher hierarchical layers. Thus, no dedicated software views will be presented on the individual layers. Figure 8.2 introduces the referenced hierarchical layers.[7] The classification into result layer, detailed layers, and assumed inputs given in the figure will be explained in Sect. 8.3. Step 6 in Fig. 8.1 briefly introduces some additional aspects from a software view. Finally, the overall system is evaluated with regard to goal achievement (Step 7 in Fig. 8.1).

8.2.1 Requirements, Constraints, and Principles

The definition of requirements constitutes the first step of product development. Analogously, some core-requirements (Table 8.1) and constraints (Table 8.2) guided the development of MOBILE. Additionally, further influences impact decisions on architecture and system development. In general, Maier and Rechtin (2009) distinguish "normative (solution based), rational (method based), participative (stakeholder based) and heuristic (lessons-learned)" (Maier and Rechtin 2009, p. 1) methodologies. To cover important normative and heuristic influences, Sect. 8.2.3 provides a summarized state-of-the-art on structures of drive-by-wire systems, used mechanisms, and best-practices. In Sect. 8.3, a novel method to analyze system safety

[5] ISO/IEC 42010:2007 defines architecture as follows: "The fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution".

[6] The architecture of a system can be described from different views depending on the goal of the description. Examples can be business views, process views, but also functional or hardware views (Masak 2010). For some of these views guidelines for standardized diagrams exist, e.g., UML for software systems (Starke 2008).

[7] In the following, hierarchical layers are always referred to as "X level" or "X layer" (with X standing for vehicle, system, etc.) to clearly distinguish between referencing of a layer and the general use of the words system, component, or element to refer to certain elements independent from layers.

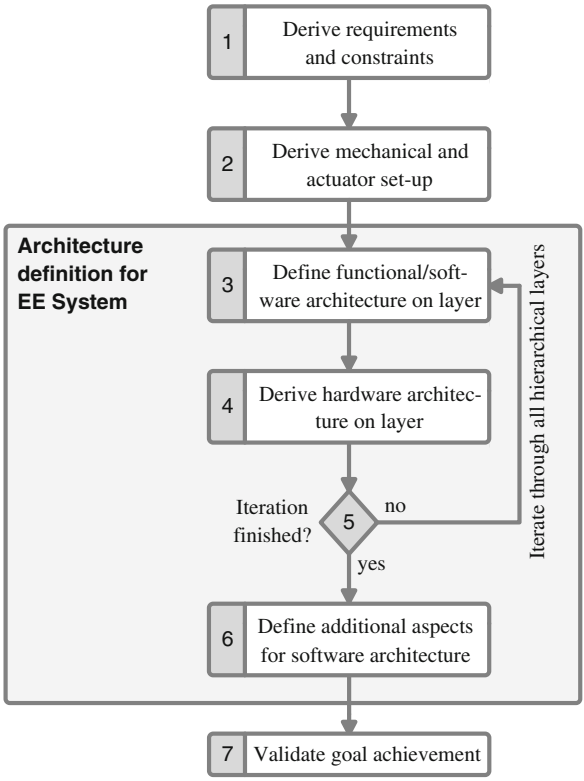
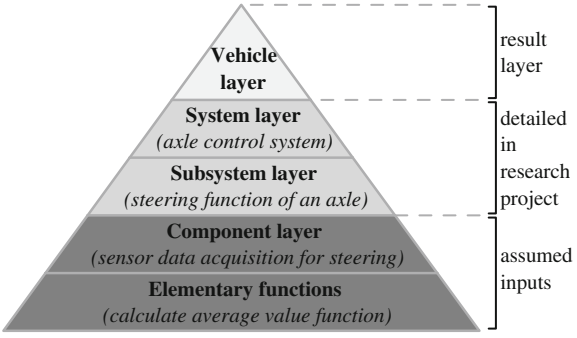


Fig. 8.1 Architecture definition process for MOBILE



(....): Example of functions at given hierarchical layer

Fig. 8.2 Hierarchical layers for architecture definition

Table 8.1 List of requirements

Providing a flexible experimental vehicle	
Requ. 1	The vehicle has to feature a *high degree of mechanical and electric/electronic modularity* that allows easy exchange of hardware components for research and testing. At the same time, the base configuration of hardware and software of the vehicle has to be powerful to support various and highly dynamic driving experiments.
Requ. 2	The software running on the electronic control units shall be easily accessible and exchangeable. The vehicle can be seen as an "*open-source vehicle*" that is readily available for any research tasks. Compatibility with the graphical programming environment Simulink of Mathworks is desired to support code re-use and shorten training periods.
Guaranteeing sufficiently safe testing	
Requ. 3	Although the experimental vehicle is only operated on test tracks, the vehicle should *fulfill basic safety requirements* and tolerate one independent fault[a] with a given probability. Details will be given in Sect. 8.3.1.
Requ. 4	The degree of hardware redundancies for safety purposes shall be reduced. In turn, the safety concept shall *exploit functional redundancies* between different types of actuators that are available in the vehicle.

[a] A fault is an "abnormal condition that can cause an element or item to fail" (ISO26262-1:2011, p. 7).

Table 8.2 List of constraints

Const. 1	MOBILE is a university only project and thus benefits from graduate and undergraduate students writing their theses on individual development tasks. Accordingly, these work packages have to be clearly defined and proper documentation plays a huge role in the project.
Const. 2	All tasks worked on by the project partners have to stem from the according core fields of research, i.e., vehicle electronics or design of mechanical parts. Other parts have to be sourced externally, e.g., actuators.
Const. 3	The project is subject to strict financial limits. Resulting, mostly "off-the-shelf" components have to be relied on.

is presented (rational methodology). Participative aspects are not detailed in this contribution.

8.2.2 Mechanical and Actuator Set-Up of MOBILE

The mechanical and actuator set-up is especially driven by the requirements on modularity and universal applicability of the vehicle *(Requ. 1)*. Accordingly, MOBILE was designed as a full electric vehicle with by-wire control for propulsion, braking, and steering:

The *electric drive concept* contributes to a powerful base configuration and ensures flexibility in the longitudinal behavior of the vehicle. The research project

InDrive demonstrated that the longitudinal behavior of a target vehicle can be simulated by a powerful carrier vehicle given little latencies in traction control (Cornelsen et al. 2011). The electric drive concept of MOBILE with a peak power of about 100 kW per wheel and 400 kW total can fulfill these requirements. Additionally, independent driving of each wheel allows yaw control via torque vectoring. The benefits and risks of such systems for vehicle handling are, e.g., evaluated by Euchler et al. (2010); Piyabongkarn et al. (2007), or Rohe (2012). Furthermore, the electric components in combination with by-wire control enable good modularity. Combined with appropriate mechanical design, drive units for an axle can be removed or replaced easily.

Four-wheel steering furthermore extends the fields of application of the vehicle. In general, the steering system can be implemented as a rack actuating type, a tie-rod actuating type or a knuckle actuating type (Park et al. 2005). In order to be able to individually steer each wheel and based on the components available on the market, the tie-rod actuating type was implemented. Thus, different steering geometries and steering concepts can be emulated by simple software modifications. In terms of performance, Wilwert et al. (2005) consider a ±40 degree steering angle per wheel as desirable for a drive-by-wire system at a steering rate of up to 40 degrees per second (see also Heiner and Thurner (1998) from the view of an OEM[8]). This steering angle approximates typical characteristics of steering systems in non-by-wire series vehicles with front wheel steering (Pfeffer and Harrer 2011). For MOBILE, each individual steering system features an adjusting range of approx. ±43 degrees and a steering rate of 130 degrees per second at nominal load. Thus, also highly dynamic maneuvers are possible.

The electro-mechanical braking system of MOBILE is designed to outperform most hydraulic brake systems in terms of reaction times. This allows precise slip control and research towards seamless integration of recuperative and mechanical braking for optimized recuperation (Pruckner et al. 2012). Additionally, the braking system renders hydraulic components in the vehicle unnecessary and thus does little impact vehicle package. The individual brake units at each wheel were designed by Vienna Engineering to ensure a 1 g deceleration of MOBILE at a maximal weight of 2.100 kg including passengers. First tests with the braking system on an experimental set-up indicate that the brake system will outperform these requirements. The safe state of each individual brake in the project MOBILE is defined as a state without any brake torque as also preferred in literature (Johannessen et al. 2004; Sinha 2011).

The vehicle is powered by a *modular power supply* consisting of two independent units providing 300, 48, and 12 V each. Currently, the main source of power of each unit is based on lead-acid batteries resulting in the given voltage level of approximately 300 V. In future, this battery pack is planed to be exchanged by lithium-ion batteries with a pack voltage of 400 V and higher energy density. 48 V and 12 V are mandatory to supply the externally sourced actuators and vehicle electronics (48 V for steering, 12 V for braking and vehicle electronics, *Const. 3*). The low voltage circuits are supplied by the main battery pack via DC-DC converters and buffer batteries

[8] Original Equipment Manufacturer, e.g., BMW, AUDI, Toyota for the automotive industry

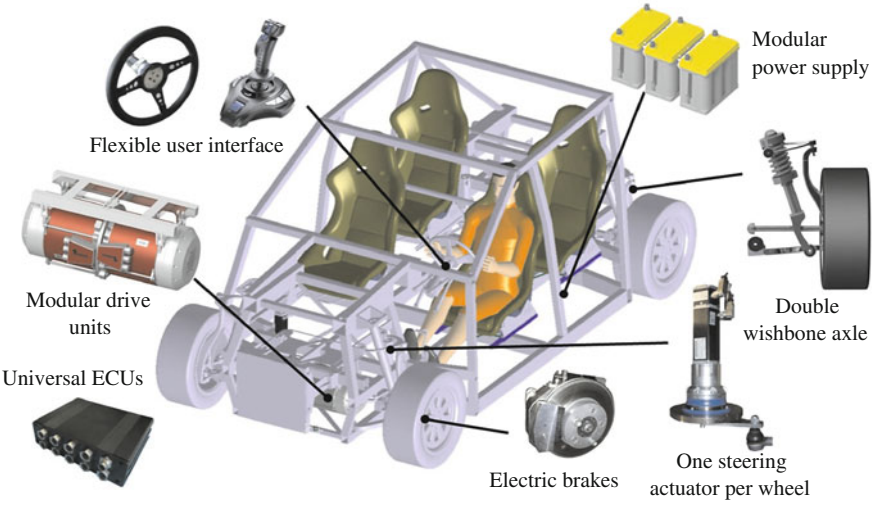


Fig. 8.3 Actuator equipment and mechanical set-up of MOBILE; ECU: Electronic Control Unit

with low capacity. Two independent power supply units reduce the overall failure[9] rate (*Requ. 3*) and limit the required currents per battery pack at peak load. Steering and braking actuators at diagonal positions in the vehicle are connected to a common power supply. Resulting, vehicle handling is less effected in case of failure of one power supply. This corresponds to the design of braking systems in series vehicles (ECE R13[10]) and is frequently replicated for brake-by-wire systems proposed in literature (Rieth 2012; Papadopoulos et al. 2001). The powerless steering actuators are back-drivable and thus allow to be moved by torque at the wheels applied by the drive motors given a suitable axle geometry (Dominguez-garcia et al. 2004).

The by-wire architecture allows flexible design of the *user interface*. All input devices can be exchanged on demand. In a base configuration, a force-feedback steering wheel, a force-feedback brake pedal, and a gas pedal are available to the driver. Also, these units provide feedback on the road surface and the current driving condition. A flexible touch-screen based visualization allows easy access to all measurements taken in the vehicle (Bergmiller et al. 2011a).

To conclude the introduction of the mechanical and actuator set-up of MOBILE (step 2 in Fig. 8.1), Fig. 8.3 summarizes the equipment of MOBILE and provides an overview of the mechanical set-up. Further details on the components can be found in Bergmiller and Maurer (2012). Summarized, the actuator set-up facilitates high flexibility of the vehicle. At the same time, it provides a high degree of functional redundancy between different types of actuators, which can be exploited by novel approaches to achieve functional safety.

---

[9] ISO 26262 defines failure as the "termination of the ability of an element to perform a function as required" (ISO26262-1:2011, p. 7).

[10] United Nations Economic Commission for Europe: Brake System Homologation.

8.2.3 Related Work for EE Systems of Drive-by-Wire Vehicles

This section provides an overview of the state-of-the-art for the design of safety critical by-wire systems. The gathered information on system structures and common practices serve as important input for the architecture derivation of MOBILE in steps 3 to 6 given in Fig. 8.1. Figure 8.4 proposes a generic view on by-wire systems as perceived by the author. The following section will first explain the basic structure of the figure and then outline the state-of-the-art for individual key aspects. The numbers given in the figure serve as a reference in the following paragraphs.

❶ Most by-wire systems for vehicles investigated in research, e.g., by Armbruster (2009), Heiner and Thurner (1998), Sinha (2011), Wilwert et al. (2005) or Zuo et al. (2005), split the system in two physically separated sections as generalized in Fig. 8.4. One section contains the user interface and consequently acquires data from the user, the other section controls the main vehicle actuators. Depending on the investigated system, the actuators are either steering actuators, brake actuators, other actuators as, e.g., for control of vertical dynamics, or combinations of these. Accordingly, the user input devices change. Input devices can feature actuators to provide additional feedback on the road surface and the driving situation to the driver.

All components are controlled by ECUs that are mounted close to the relevant actuators or sensors. Other designs that wire all components directly to one central controller as presented by Park et al. (2005) or several other research vehicles with

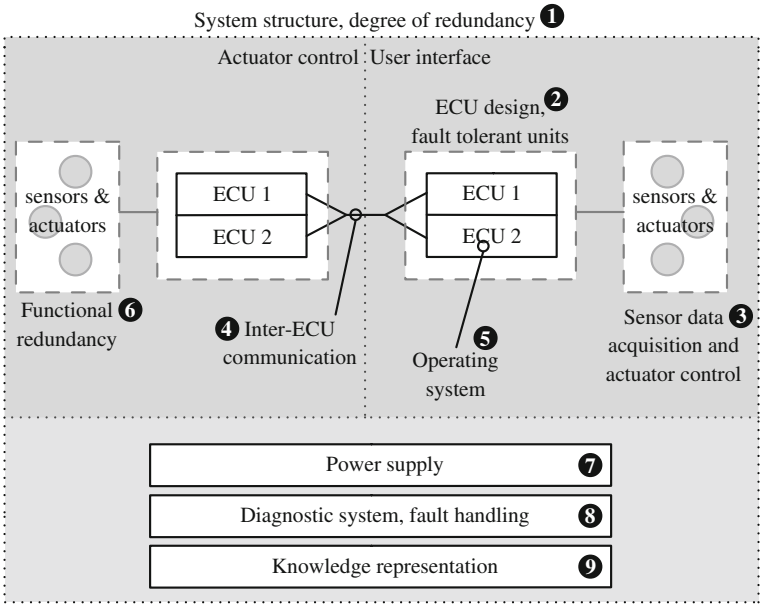


Fig. 8.4 Structure of a generic by-wire system as referenced in the state-of-the-art section; ECU: Electronic Control Unit

no focus on functional safety of the EE system are not further regarded for wiring effort, EMI,[11] and modularity reasons. Also, solutions with mechanical/hydraulic fall-back layer are not considered (Zuo et al. 2005). Each safety critical ECU is usually available redundantly as no single unit can—so far—achieve the required failure rates for automotive drive-by-wire systems. In general, the degree of hardware redundancy is kept as low as possible due to production costs. Two redundant components for one task in combination with a sufficiently powerful diagnostic and decision unit and fail-silent behavior of each component are assumed to be able to fulfill the required failure rates (Wilwert et al. 2005; Miller 2007). Several systems featuring this structure can be found in literature as, e.g., introduced by Hasan and Anwar (2008), Armbruster (2009), Sinha (2011), or Wilwert et al. (2005).[12] Such a combination of two (or more) ECUs is then regarded as fault-tolerant unit[13] ❷. If all units are permanently turned on, the system is denoted as operating in hot-standby mode (Neudörfer 2011). This shortens take-over times in case of failures of the primary unit as the secondary unit does not have to boot or initialize.

In some cases, redundant ECUs are replaced by a single ECU featuring a multi-core architecture and according board design in combination with special mechanisms to allow executing multiple safety critical functions independently on this platform. According strategies are, e.g., outlined in the RECOMP project (Motruk et al. 2012) funded by the Federal Ministry of Education and Research (BMBF) or by Philipps (2012). Amongst others, a dedicated software functions and storage management is required to proof independence of functions with regard to common cause failure.[14] Also on chip level, cost efficient means to detect and correct transient faults are investigated (Touloupis et al. 2005).

A further approach to reduction of hardware redundancies can be a network centric architecture, where nodes monitor each other mutually. In case of a failure of a single ECU, the other ECUs detect the failure and continue accordingly adapted operation. As a result, the number of ECUs can be reduced, but the complexity of each individual device increases (Kelling and Heck 2002; Johannessen et al. 2002; Sakurai et al. 2008). Resulting effects on costs have to be evaluated for each specific system. Especially, brake-by-wire can benefit from network centric architectures, as several redundant actuators exist that can be equipped with independent ECUs. Also, the vehicle can still be decelerated with two or three brakes available. For the project MOBILE, a combination of a network centric approach and classical redundancies seems reasonable.

The redundancy strategy is extended for sensors and actuators ❸. On the sensor side, typically at least triple hardware redundancy is applied to acquire multiple

[11] Electromagnetic interference.

[12] Note: For fly-by-wire systems at least quadruple redundancy for military aircrafts and higher degrees of redundancy for civil aviation are required (Collinson 1999).

[13] In a fault tolerant unit, a defined number of faults does not lead to a failure of the overall unit, e.g.,Wilwert et al. (2005).

[14] A common cause failure is a "failure of two or more elements of an item resulting from a single specific event or root cause" (ISO26262-1:2011, p.3).

sensor signals and perform majority voting to determine faulty measurements as, e.g., demonstrated by Bertacchini et al. (2005). Other approaches rely on analytical redundancy that replaces one or two sensors by software algorithms that derive additional "virtual sensor data" or diagnostic residuals for the investigated signal by comparison with other sensor data available in the system (Anwar and Niu 2010; Gadda et al. 2007; He et al. 2010; Kim et al. 2010). Also, fault tolerance strategies are frequently implemented mechanically and electronically within the sensors or actuators (Dilger et al. 2004). Especially, model based diagnostic algorithms can assist identification and treatment of faults already within the actuators (Isermann and Beck 2011; Muenchhof et al. 2009). Resulting, the required number of physically separated redundant units can be reduced. For steering, mostly two redundant actuators are available at one axle (Muenchhof et al. 2009; Wilwert et al. 2005; Zhen et al. 2005; Zuo et al. 2005). For the braking system, each wheel features an individual actuator (Isermann et al. 2001; Papadopoulos et al. 2001; Reichel and Armbruster 2011). The feedback actuators at the user input devices are mostly also classified as safety critical. Still, the criticality is lower than the one of the actuators at the wheels. Depending on the investigated system, developers implement these actuators as single actuators (Anwar and Niu 2010), redundantly (Wilwert et al. 2005) or provide mechanical back-up feedback (Pruckner et al. 2012; Zuo et al. 2005).

The system design according to the so far outlined guidelines ensures proper operation of the sensors and actuators and supports safe execution of the application algorithms on computational platforms. Still, a main issue in by-wire Systems is the communication between physically separated ECUs ❹. The according data bus systems have to be available at least in single redundancy, including physical separation in wiring as, e.g., outlined by Wilwert et al. (2005). Some research projects also propose more than two physically independent channels, e.g., Sinha (2011) and Sundar and Plunkett (2006). Additionally, the overall network has to feature given and precise timings to ensure that lost or delayed messages can be detected and a maximal round trip time can be guaranteed (Heiner and Thurner 1998). Wilwert et al. (2005) derive a maximal acceptable end-to-end response time for driver inputs to the steering actuators of 17.6 ms.[15] Beyond this limit, the vehicle becomes unstable. In applications, exclusively time-triggered data bus systems are relied on, e.g., TTCAN (He et al. 2010), TTP/C (Papadopoulos et al. 2001; Blanc et al. 2009), or FlexRay (Sinha 2011; Sundar and Plunkett 2006; Waraus 2009). Some research projects also investigate the applicability of Ethernet in combination with a time triggered extension (Müller et al. 2011). These bus systems provide precise and deterministic communication timing at the price of less flexibility for spontaneous adaptations during the design process (Mishra and Naik 2005). Then, the challenge arises to synchronize the user application with the network timings in order to ensure defined latencies and synchronization throughout the network (Sundar and Plunkett 2006). Different operating systems ❺ support this task as a modified OSEK (Sakurai et al. 2008), OSEK Time, FTCom (Wilwert et al. 2005), or recently also AUTOSAR (Mitzlaff et al. 2010;

[15] For comparison: In aviation, sensors are sampled about 100 times per second which roughly equals the minimal demands in the automotive field (data for A320, Collinson (1999)).

Tucci-Piergiovanni et al. 2011). Still, the applicability of each operating system has to be confirmed individually depending on the required precision of timings and the available computational resources provided by the network nodes.

Assuming a proper interaction and operation of the individual components within the vehicle was established, functional redundancies ❻ between different actuators and especially different types of actuators (steering, drive, brake) to achieve the overall safety goals can be exploited. Thereby, hardly any research projects are available that exploit these redundancies for a proof of safety in accordance with ISO 26262, but several projects investigate possible functional redundancies and cross-couplings between the individual actuators from a view of vehicle dynamics. Thereby, simulation or experimental vehicles serve as test beds (Arbitmann et al. 2011; Dominguezgarcia et al. 2004; Euchler et al. 2010; Hayama et al. 2008; Johannessen et al. 2002; Reinold et al. 2010). Further contributions from research groups in the field of vehicle dynamics, e.g., the groups of Gerdes at Stanford University and Trächtler at Universität Paderborn, exploit the capabilities of highly flexible experimental vehicles to make driving itself safer. Thereby, safety of the drive-by-wire system is not focused. These research results serve as a guideline to what vehicle control algorithms are already available or can be expected to be available in the near future. Accordingly, the design of the EE system of MOBILE was influenced.

To ensure the operation of the overall system, a fault tolerant power supply unit ❼ is mandatory. Typically, redundant systems with mutual isolation are implemented (Abele 2008; Kelling and Heck 2002; Sieglin 2009). The GM vehicle Sequel implements triple redundancy, and additional means to reconfigure the power supply in case of failure (Sundar and Plunkett 2006). Such central reconfiguration units for power supply systems are useful for failure compensation and frequently relied on- (Armbruster et al. 2006; Sundar and Plunkett 2006). But, they may also turn out as a weak spot of an architecture due to their huge impact on the system in case of failure. Typically, the power supply provides 12V to steering and braking actuators, unless the vehicle weight requires higher voltage levels ($\geq 42\,\mathrm{V}$) to cover the increased power demands (Wilwert et al. 2005; Sundar and Plunkett 2006).

Based on the components introduced so far, the by-wire system is operable. It should be able to maintain at least degraded operation after any first fault as demanded according to the state-of-the-art (Armbruster et al. 2006; Pruckner et al. 2012) and typical demands for licensing of vehicles (ECE R13). Still, a powerful diagnostic system ❽ has to be provided in order to ensure the detection of faults and to provide the basis for appropriate fault handling. As given above, most components already provide means to diagnose proper operation. In combination with network overarching monitoring mechanisms for timings and interfaces, a huge data base on the current state of the vehicle is available. To derive according actions from this data, different approaches, mostly relying on heuristics and probabilistic interpretations, were developed (Bergmiller et al. 2011b; Isermann and Beck 2011; Muenchhof et al. 2009; Schwall and Gerdes 2002). A challenge for these algorithms are the short execution times that have to be guaranteed. Additionally, the behavior of the diagnostic and action derivation system has to be predictable which renders most machine learning approaches unsuitable. Typically, the vehicle is regarded as non self-healing.

Thus, repair of defective components is not performed online. In aviation, self-restart of components are taken into account (Schroer 2008) to improve system safety. This idea is also investigated theoretically for the automotive domain (Pimentel 2003) but hardly followed for safety critical systems in real vehicles. In the project MOBILE, the idea is adapted for real application in the experimental vehicle (Bergmiller et al. 2011b).

Finally, the information on the overall system acquired by the diagnostics system can be put in relation and integrated in a "knowledge base" ❾. This knowledge[16] includes relevant information on the current capabilities of the vehicle and according maneuvers that can be executed. For autonomous vehicles, such investigations have already been made by Maurer (2000) and Siedersberger (2003). With increasing capabilities of the vehicle, the part focusing on the ego vehicle should be further emphasized and extended. The challenge becomes even bigger when different modules can be combined flexibly within one vehicle. This is so far not targeted sufficiently by research projects but investigated in the MOBILE project. Therefore, a flexible ability based self representation is implemented.

Conclusion and differentiation of MOBILE: Multiple research groups are actively working in the drive-by-wire field, but hardly any group targets the overall system including steering, braking and the propulsion function from a functional safety view. Usually, only one system is investigated, and mostly these systems are implemented on a test-rack or in combination with simulators and not in real vehicles. Some research groups as the ones of Gerdes and Trächtler (Beal and Gerdes 2010; Gadda et al. 2007; Trächtler and Niewels 2006; Reinold et al. 2010) have built up vehicles with extended by-wire functionality. Still, the safety of the onboard EE system is not investigated in detail for these vehicles. These groups focus on vehicle dynamics. Research results in this field are taken as boundary conditions for identification of possible functional redundancies in MOBILE.

Johannessen (2001) presents a modified Sirius vehicle (*SIRIUS 2001*) with individual steering and braking of each wheel and time-triggered communication for a network centric approach to safety. To some extent, also cross compensations between actuators to control the vehicle, e.g., steering by differential braking, are considered. The vehicle is based on a conventional propulsion system, and the wheel brakes are hydraulic but controlled by electrical pumps to generate pressure. Also, an analysis of the vehicle failure rate is performed resulting in a failure rate of 5.74 EE−8 catastrophic failures per hour but neglecting the power supply system. In a follow up project (FAR project) a model vehicle with four wheel steering, individual braking and four wheel drive was built up Johannessen et al. (2004b). Some results from the Sirius vehicle can be adopted and taken as reference for MOBILE. Still, flexibility requirements for tooling and full consideration of all components of the vehicle including power supply and an propulsion system capable of torque vectoring will require adaptations and extensions for MOBILE. Especially, the strongly

[16] Knowledge denotes the "awareness, consciousness, or familiarity gained by experience or learning" (Collins 2010). In the project MOBILE, the "self-awareness" of the vehicle is considered. The "experience" is provided at design time based on experiments or statistics.

network centric approach reduces flexibility of the Sirius vehicle for usage as a development tool.

The European project *SPARC* (Armbruster 2009; Reichel and Armbruster 2011; Sieglin 2009) stands out by thoroughly investigating a full by-wire concept for application in different vehicle classes (trucks and cars). The project presents a full drive-by-wire architecture that is applied to different experimental vehicles. The vehicles feature one steerable axle and a brake-by-wire system. The by-wire system includes redundant actuators and a degradation approach to handle faults. Still, the system architecture requires the memory in the network to be available in quadruple redundancy as all nodes have to be able to perform all computational tasks. Compared to this project, MOBILE features higher flexibility due to the given actuator set-up. Also, MOBILE targets an even lower degree of hardware redundancy of actuators and controllers by exploiting the functional redundancies in the vehicle instead.

In general, the design of MOBILE focuses the usage as an experimental platform and serves as a proposal for future series vehicles given the further progress of research in the given fields. This clearly distinguishes MOBILE from approaches for current series vehicles. Summarizing, the design of MOBILE, as will be presented in the next section, exploits several principles outlined above: Any ECUs, actuators, and data bus connections for one task should be available at most twice. Lower degrees of redundancy should be implemented if functional redundancies can be exploited. Sensors are implemented in triple redundancy. Later on, analytical redundancy can easily replace existing sensors. Safety critical components should be kept independent from each other wherever possible such that the overall vehicle can tolerate one independent fault and guarantee an emergency run interval. Communication in the vehicle will be time triggered and allow precise synchronization of applications throughout the network. Force Feedback is not regarded as a highly safety critical function in MOBILE as a skilled test driver is driving the vehicle. Resulting, the according actuators are not implemented redundantly, but mechanical open-loop feedback will ensure controllability.

8.2.4 Hierarchically Structured Architecture Derivation

This section presents the architecture of the EE system of MOBILE. The architecture is intended as a template for by-wire vehicles with high need for flexibility and safety at low costs. As will be outlined at the end of this section, basic ideas could also be transfered for cost efficient proof of functional safety in series vehicles. As mentioned, the architecture is derived top-down along the hierarchical layers introduced in Fig. 8.2. On each layer, at first the functional architecture is presented, then a suitable hardware architecture is derived that allows to execute all needed functions and fulfills requirements on modularity and functional safety (steps 3–5 in Fig. 8.1). The requirements given in Sect. 8.2.1, the hardware set-up introduced in Sect. 8.2.2, and already existing research results (Sect. 8.2.3) guide the architecture derivation.

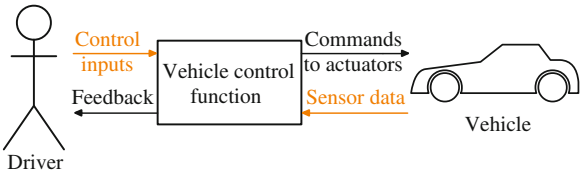


Fig. 8.5 Functional architecture of MOBILE on "vehicle layer"

8.2.4.1 Vehicle Layer

The fundamental functional architecture on "vehicle layer" is simple: It consists only of the vehicle control function if other comfort or add-on functions are neglected. The function acquires data from the driver (control inputs) and accordingly controls the actuators of the vehicle (commands to actuators). Vice versa, sensor data is gathered to execute the vehicle control function and to provide feedback to the driver. Figure 8.5 outlines these basic dependencies in the style of a UML context diagram. The hardware architecture is structured similarly to the functional set-up. It consists of the part of the EE system concerned with vehicle control. All other parts that are not relevant for the basic driving function are neglected. Individual hardware components are not distinguished at this layer.

8.2.4.2 System Layer

The "system layer" splits the vehicle control function into the most important components. Figure 8.6 shows the according functional view. Thereby, the special purpose of the vehicle as a development tool and the intention to exploit functional redundancies are already regarded:

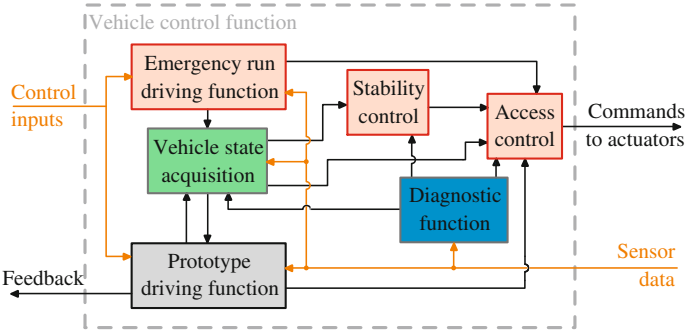


Fig. 8.6 Functional architecture of MOBILE on "system layer"

A *prototype driving function* that is realized by software/hardware under development controls the vehicle actuators in experimental mode. When implementing this prototype function, the developer should be provided full access to the vehicle including all sensor data and access to all actuators (*Requ. 1/2*). The prototype driving function allows to provide feedback to the driver via interface devices as the steering wheel or the brake pedal. The functionality implemented by the prototype driving function can hugely vary, e.g., four-wheel steering vs. steering of only one axle with adjustable steering ratio.

As a basis for the prototype driving function and to provide data to other units in the vehicle, the *vehicle state acquisition* function gathers all available sensor data related to vehicle dynamics and derives the current vehicle state. Thus, the vehicle state acquisition contributes significantly to the tooling character of the vehicle (*Requ. 1*). The vehicle state acquisition gathers its data throughout the network. Depending on the state of the according nodes, the vehicle state acquisition can rely on data from the emergency run or the prototype driving function.

Jointly, the prototype driving function and the vehicle state acquisition function allow the control of the vehicle. Further components are added to ensure safe driving. This includes an *emergency driving function*, a *stability control* module, and the *access control*. The emergency driving function provides basic control of actuators in a "fall-back" manner. Thereby, the actuators are operated in their most basic mode of operation with minimal usage of extra sensors. No cross-couplings or dependencies with other functions exist. Still, the emergency driving function provides the driver full access to steering actuators, brakes, and drive motors (*Requ. 3*).

As MOBILE is only equipped with one actuator for steering, braking, and drive at each wheel, functional redundancies have to be exploited for safety. Accordingly, the stability control system operates, on the one side, as a conventional stability control system known from series vehicles. On the other side, it compensates the failure of a single actuator by adapting its control strategy (*Requ. 4*). To detect each failure state, the stability control relies on the data provided by the *diagnostic system*.

The diagnostic system monitors the current state of the vehicle from an electric/electronic point of view. Faults within components and resulting possible failures are indicated to all nodes of the network.

Finally, the access control determines which driving function may control the actuators: the prototype driving function, the emergency run driving function or the stability control in case of actuator failures (*Requ. 3*). Thereby, the stability control re-uses low-level basic actuator access implemented both by the emergency driving function and the prototype driving function to control the vehicle.

The hardware units have to provide the basis for execution of the different functions outlined in the functional architecture. For the hardware architecture, the principle to keep components independent and the requirements for modularity of the vehicle are regarded. Figure 8.7 outlines the resulting architecture on system level. The vehicle state acquisition function and the stability control function are implemented on independent hardware units (*inertial measurement system* and *stability control system*) with no redundancy. A failure of one of these units can not induce a total system failure if fail silent behavior is ensured. At least, the emergency driving

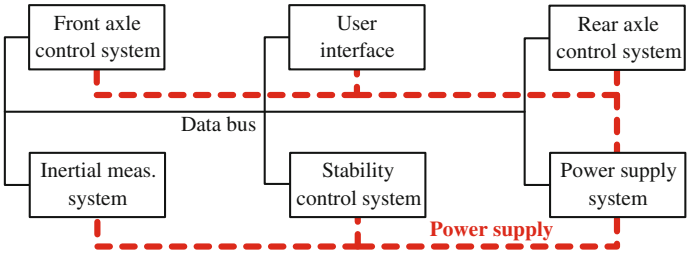


Fig. 8.7 Hardware architecture of MOBILE on "system layer"

function will be available. The driving functions (emergency and prototype) are distributed over three control systems. The *front axle control system* controls the actuators of the front axle and pre-processes all sensor data acquired at the axle. This contributes to fulfill the modularity requirements (*Requ. 1*). For modifications to the front axle, only the according control system has to be adapted. The rear axle control system is structured analogously. The *user interface* acquires the user inputs and provides the data to all other units via a *data bus* backbone. An additional simulation system can be added to the hardware set-up for more complex calculations required by prototype functions. As the simulation computer can always be disconnected from the network in case of failure, it is not further regarded in the hardware architecture. A *power supply system* provides the required electrical energy to all mentioned systems. The intelligent power supply system contributes to the desired fail-silent behavior of each individual system. More details will become obvious on lower hierarchical layers.

Figure 8.8 illustrates a merged view on hardware and functional aspects. It becomes obvious that the driving function and especially the diagnostic and safety related functions are distributed throughout the whole network. This supports the modularity concept as, e.g., axles can easily be exchanged as low-level tasks are kept local. Also, it reduces the probability of failure of the vehicle control function. In case of failure of a component, the system maintains at least a degraded mode of operation.

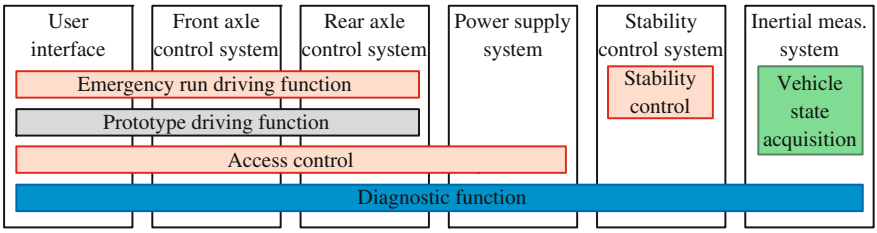


Fig. 8.8 Association of functional elements to hardware units on "system layer"

8.2.4.3 Subsystem Layer

Further detailing of the functional architecture on subsystem level reveals important functional modules and their interaction (Fig. 8.9). For the driving functions, *data acquisition*, *data processing*, and *actuator control* are distinguished. If necessary, the access control function blocks actuator access rather than starting or stopping the execution of a function. Thus, data acquisition and processing is performed in a hot-standby manner ensuring short switching times. The vehicle state acquisition is split into *inertial measurement* with an according sensor platform, *sensor data fusion* and *state estimation*. The sensor data fusion combines the inertial measurements with classical sensor data as wheel speeds or steering angles. Depending on the mode of operation, data from classical sensors is acquired from the emergency run function or the prototype driving function. The state estimation then takes the gathered and fusioned data as a starting point to estimate unmeasurable values as the side slip angle.

The stability control generates a *reference behavior* of the vehicle based on internal models of desired vehicle dynamics. If deviations in vehicle behavior from the reference given by the models are detected, the stability control modifies the actuator commands by the driver to ensure safe driving. Depending on the task of the stability control different reference models are referred to. For classical stability control, a model approximates the stable behavior of the real vehicle in order to identify critical deviations in vehicle behavior. Additionally, a simple front wheel steering vehicle model with limited dynamics resembles the fail-safe behavior of the vehicle in case of an actuator failure. Thus, the vehicle control system has to guarantee this minimal performance of the real vehicle even in case of a defined number of failures of actu-

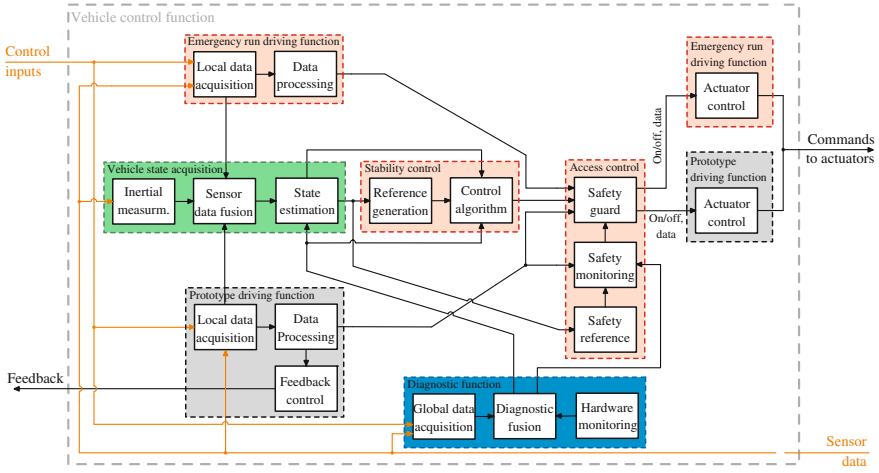


Fig. 8.9 Functional architecture of MOBILE on "subsystem level"

ators. This serves as a basis to describe the safe state of the vehicle in a later safety evaluation.

With the increased level of detail on subsystem layer, the information sources for the diagnostic system become obvious. The system extracts necessary information from *hardware monitoring* algorithms and the *globally available data* in the vehicle. Hardware monitoring relies on diagnostics as referenced in the state of the art (Sect. 8.2.3). The global data acquisition takes into account the driver commands and the reaction of the vehicle to these commands. If deviations between desired and actual vehicle handling become significant, the diagnostic system identifies possible failures. Still, the diagnostic unit has a more hardware and EE system focused perception of the vehicle compared to a stability control system. More details on the diagnostic algorithms can be found in Bergmiller et al. (2011b).

The access control is split into *safety reference* generation, *safety monitoring*, and *safety guard*. These units re-configure the system if a driving function fails. Basically, the operation is similar to the one of a stability control: A safety reference describes the desired state of the EE system. The safety monitoring detects failures in the system behavior by comparison to this reference and commands system reconfiguration if necessary.

The hardware architecture on subsystem layer is outlined in Fig. 8.10. If only one failure has to be tolerated, the stability control and the inertial measurement unit need not be fault-tolerant and thus are implemented on only one controller each (*inertial measurement controller* and *stability controller*). Still, each unit is powered by both power supply lanes. This is necessary, as a total loss of one power line leads

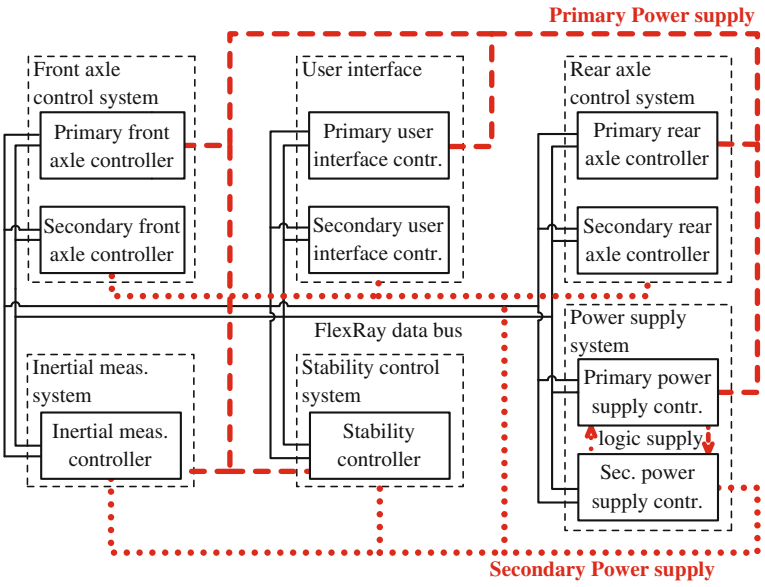


Fig. 8.10 Hardware architecture of MOBILE on "subsystem layer"

to a loss of all connected controllers, two diagonal brakes, propulsion at one axle, and two diagonal steering motors. Resulting, a stability control system is needed to maintain at least emergency operation of the vehicle. If the stability control and state acquisition would be powered by one lane, only the failure of one power lane would be manageable. The safety analysis of MOBILE showed that a total failure of one power supply lane is not possible due to a single point fault, but still a failure of the 12 V lane is possible and stability control is useful to compensate the resulting loss of two brakes and two drive motors. If the stability control itself fails, power can be cut by the power supply system and fail-silent behavior is achieved.

A failure of all other systems would lead to a full or partial loss of control of the vehicle. Thus, these systems are set up as fault tolerant units consisting of two redundant controllers each. The two controllers within a fault tolerant unit are powered by different power supplies that are controlled by individual *power supply controllers*. Resulting fail-silent behavior can be assured for each component. Communication between the modules is performed via a fault tolerant and time triggered FlexRay data bus with physically separately wired redundant channels. The data bus communication is designed to support in-cycle response and a cycle time of 4 ms. According to the information given in Sect. 8.2.3 and experiences in the project MOBILE, this timing suffices for stable operation of the vehicle and to implement high performance control algorithms for vehicle dynamics. Redundant information that is transmitted within one cycle is distributed equally over the communication cycle to reduce the impact of burst errors. Within each fault tolerant unit, sensors for basic actuator control or to acquire driver inputs are available in triple redundancy for majority voting, while actuators are only available once for each task. The wiring of the sensors and actuators to the axle controllers is done according to the requirements of the components available on the market. Mostly, CAN-bus connections implementing a CANopen protocol are relied on. Furthermore some digital and analog signals of sensors are evaluated and directly connected to the axle controllers. Within each axle, the allocation of sensor signals and actuator commands to bus systems ensures that the stability control unit can continue to control the axle such that at least neutral behavior with regard to vehicle dynamics can be achieved in case of a failure of a bus connection. E.g., the drive motor of one wheel is connected to a different bus than the braking unit of this wheel. Thus, in case of failure of one of the systems, the wheel can still be decelerated to some extent—either by recuperative braking or by mechanical braking. For MOBILE, research is ongoing to furthermore clarify potentials and limitations of control algorithms to handle actuator failures (Goldschmidt 2012; Lieberam 2011; Töpler 2010).

To avoid loss of control due to loss of power, the power supply system is set up redundantly. The power supply controllers contribute to the desired fail-silent behavior of all components in the vehicle. If a controller is classified as defective, the power supply for the controller can be cut. Thus, fail-silent behavior can be enforced externally if internal mechanisms fail. Within the power supply system, the two power supply controllers supply the logic part of each other. As a result, a malfunctioning power supply controller can be switched off by the partner controller. The safe state of the controllers is to supply all connected controllers in case of unpowered control

logic. Using this configuration, all reasonable failure scenarios of the power supply can be handled in cooperation with decentralized safety guards executed on each node.

Again, the hardware and functional view are merged to determine the allocation of functions to hardware components (Fig. 8.11). It becomes obvious that the front and rear axle modules are set up analogously. The controllers within each according fault tolerant unit perform different tasks. The primary controller is intended to execute the prototype software under development and control all actuators including feedback generation. The primary controllers do not significantly contribute to monitoring of safe vehicle operation. Only the hardware monitoring algorithms provide feedback on the state of the node via the network. Additionally, a simple safety guard can block the boot of the node if an according command is received from the power control unit. In case of a necessary intervention of the stability control based on the primary controllers, base access to the actuators is granted. These algorithms are not visible to the user and are executed in the background. This allows the developer to act almost unrestricted by the safety concept.

The main diagnostic functions are implemented on the secondary controller. These controllers are operated in hot-standby manner to take over vehicle control if required. While not performing vehicle control, the available resources are used to perform sophisticated diagnostic algorithms. These algorithms continuously compare the behavior of the primary controller to a safe reference. Furthermore, all inputs from

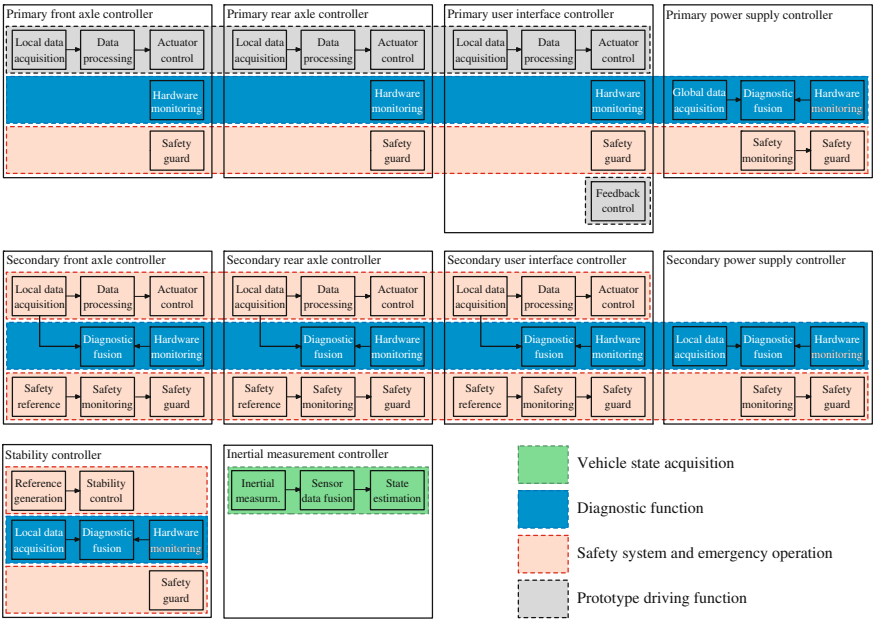


Fig. 8.11 Association of functional elements to hardware units on "subsystem layer"

the hardware monitoring of both primary and secondary controller are regarded. In case of failure of one node, the secondary controller communicates the failure state to the power supply controllers—either by a dedicated message or by falling silent. The power supply controllers can then command one node to transition into fail-silent mode and the other to maintain or take over the driving task. Also, a defective node can simply be disconnected from the power supply. It is important to note, that the secondary controller continually has to identify the basic behavior of the primary controller in order to achieve smooth taking over after a failure. Thus, the steering ratio is adapted online. Still, the adaptation is performed within strict bounds to avoid adaptation to erroneous behavior. If the secondary node is subject to failure, a restart or reinitialization can be triggered to repair the system. For the primary node, such measures are not applicable as the prototype software might go through unintended initialization routines while driving.

Obviously, the secondary controller represents a possible source of single point failures as it is mainly responsible for the diagnostic tasks and thus the decision making within one axle. This challenge is addressed by the self-monitoring of each node and monitoring of each node by the remaining network. The distributed diagnostic system on all nodes monitors proper operation of all other nodes by an Alive Network Management Vector based mechanism similar to the one implemented by TTP/C on bus controller level or (Sakurai et al. 2008) as an extension to OSEK. The alive monitoring is implemented on application layer and allows to derive the operability status of each node within the network on application level. Thus, advantages of a network centric architecture are exploited.

As already indicated, the power supply controllers play an important role for the central coordination of the safety concept. Basically, the access control is mainly implemented on the power supply controllers. To prevent scenarios where a power supply controller generates single point of failure scenarios, the power supply nodes monitor each other intensely and include information from the network wide alive monitoring. Additionally, the individual nodes in the axle modules perform consistency checks between the commands of the two power supply units. For the first version of the safety concept, the overall logic is based on the assumption that only one independent fault has to be tolerated. All multi-point faults are assumed to lead to a loss of vehicle control.

With the presented merged view on subsystem layer, the introduction of the hardware and functional architecture concludes. The following hierarchical layers (component and elementary layer) feature further increasing levels of detail and focus on individual functions and their allocation to hardware parts within one controller. To some extent, these investigations are made within the project MOBILE during layout of the electronic components and software implementation. Still, several hardware parts are sourced externally and no detailed information is available.

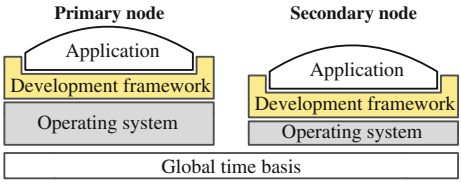


Fig. 8.12 Time synchronization and software framework on the network nodes

8.2.4.4 Software View

Complementing the architecture introduction, a brief look at a software view of the system is taken (Step 6 in Fig. 8.1). As mentioned, the software view of the system on the individual layers widely correlates with the introduced functional architecture. After all, each elementary function can be implemented as a software function provided by an object or a dedicated stand-alone function. Still, the overall software structure "orthogonally" to the demonstrated application layer was not yet regarded. This structure includes the layered approach from hardware abstraction to the application software modules and the organization of their interaction. For the project MOBILE, Fig. 8.12 provides a simplified overview of the structure on each node. Each node runs a custom written operating system that fulfills the requirements for task scheduling, minimal resource consumption, and latencies while featuring the required flexibility and allowing full access to all components. Also, it ensures that the node synchronizes itself to the global time basis based on the FlexRay bus clock. Resulting, all actions within the individual nodes can be synchronized at a precision of microseconds if necessary. This way, also in-cycle response and just-in-time data processing and transmission can be realized. The operating system interfaces with the development framework provided by Mathworks Embedded Coder. The application modules are then integrated as Simulink blocks that are written in C/C++ code or modeled graphically with Simulink. This strongly facilitates code reuse and reduces coding errors due to graphical programming. The application modules can—if required—be executed on any node within the network due to the common interfacing. To reduce common cause failures, primary and secondary node run different operating systems. Summarized, the presented layered approach allows flexible exchange and reuse of software modules while at the same time hard real-time requirements can be met on microcontroller hardware (*Requ. 2*).

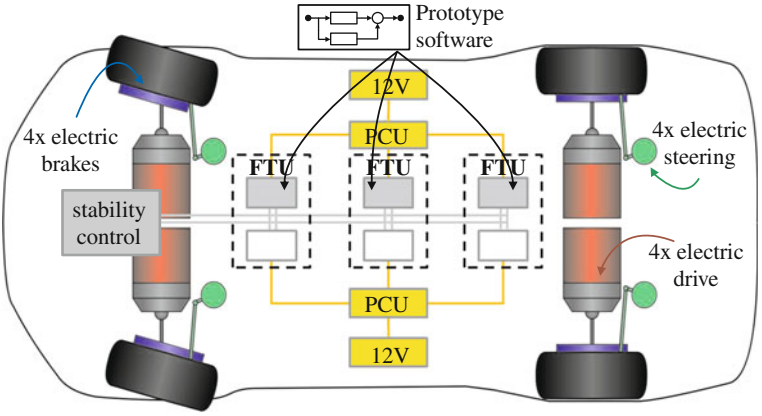
8.2.5 Summary and Criticism of the Presented Architecture

In the previous sections, a by-wire architecture for an experimental vehicle was derived in a top-down manner as far as possible within a research project. The architecture bridges the gaps (a) between flexibility and safety by network based monitoring combined with degradation concepts and (b) between safety and costs

for hardware redundancy by exploiting functional redundancies. Thus, the architecture can fulfill the initially set requirements on flexibility, safety, and reduction of hardware redundancies (step 7 in Fig. 8.1). Several key aspects distinguish it from other approaches:

- The vehicle control function is implemented as a highly integrated system of all driving functions keeping required hardware redundancies low (Sect. 8.2.3) while allowing to fully exploit cross couplings between different functions. Comparable systems with focus on safety and a similar actuator set up are not found in literature.
- The architecture strongly emphasizes the importance of distributed execution of safety critical diagnostics on application level to achieve low failure rates while keeping the degree of redundancy low.
- System degradation including exploitation of functional redundancies is an indispensable part of the architectural approach. Online reconfiguration and "online repair" of components, e.g., by reinitializing components is possible. Still, repair mechanisms are not yet regarded for safety analysis.
- All actuators installed in the vehicle are used to generate novel functionalities while also contributing to system safety. Redundant actuators are economized.
- A cost efficient mean to achieve fail-silent behavior of the individual network nodes is implemented based on joint action of the power supply controllers and decentralized safety guards.
- The architecture supports flexible development of prototype software on the primary controllers with little restrictions due to safety mechanisms. Sophisticated monitoring algorithms help to keep safety mechanisms out of the application software and perform external monitoring. This approach might also be extended to complex functionalities in series vehicles. Then, not the complex function itself, but the external monitoring system has to comply with the given safety requirements. If done properly, this external safety guard can be structured generically and be re-used for different versions of the complex function.
- A model of the intended vehicle handling clearly defines the emergency run of the vehicle. This model serves as a baseline for benchmarking stability control algorithms but also for functional safety analysis. Resulting, well-defined requirements for proof of functional safety are defined moving away from less meaningful requirements on the behavior of individual components as used so far.

The architecture is based on important assumptions that are partially still subject to research. It is assumed that the FlexRay system with redundant channels suffices to fulfill the automotive safety requirements. Still, detailed safety investigations for data bus systems are ongoing. If required, the data bus system could be exchanged or extended. Moreover, the stability control to exploit functional redundancies represents the key for the reduction in redundant actuators and thus the key to one of the main benefits of the presented architecture. Based on the results of multiple research projects in vehicle dynamics and investigations in the project MOBILE, it seems reasonable that the stability control will be able to handle failures of individual actuators. Still, so far hardly any quantitative investigation of cross-compensations between dif-



PCU: Power control unit; **FTU**: Fault tolerant unit

Fig. 8.13 Simplified architectural overview of MOBILE

ferent types of actuators under varying environmental conditions are available in the research community.

Summarized, the architecture enables construction of a powerful development platform. On the one side, novel applications for highly flexible vehicles can be evaluated during real test runs. On the other side, new means to ensure safety based on functional redundancy and vehicle stability control can be developed and verified. Concluding architecture derivation, Fig. 8.13 summarizes important aspects of the vehicle architecture merging mechanical, hardware and software views.

8.3 Functional Safety Evaluation

> A person who, intentionally or negligently, unlawfully injures the life, body, health, freedom, property or another right of another person is liable to make compensation to the other party for the damage arising from this.
>
> German Civil Code[17]

The above excerpt of the German Civil Code transfered to the automotive industry stresses the duty of any car manufacturer and engineer to ensure that its products are designed according to the state-of-the-art in terms of safety. If a vehicle demonstrably fails due to negligent design, the responsible person can be held liable for any consequences and thus has to provide according compensation. As mentioned in Sect. 8.1, the proof of a state-of-the-art design of modern vehicles on functional

[17] Official Translation by the Langenscheidt Translation Service of the German Civil Code (BGB) §823 in the version of its promulgation from 2nd of January 2002, last amended by statute of 28th of September 2009.

level is becoming more and more challenging for the car manufacturer. To give a guideline for proper design and safety validation of new vehicles on functional level, the ISO 26262 for functional safety in vehicles was derived from the more general IEC 61508[18] for functional safety in electronic safety-related systems. As a result, the ISO 26262 also serves as benchmark for the design of a vehicle and the design process in case of law suits. One key aspect of the ISO 26262 is the hazard analysis and the resulting classification of the derived safety goals in terms of ASILs.[19] Amongst others, these levels determine upper thresholds for the acceptable failure rate of the investigated function. Especially, in the field of electric vehicles or by-wire approaches, ISO 26262 opposes high demands on newly developed safety critical systems. These systems do not have a long lasting history with associated statistics based on millions of sold vehicles to rely on, but safety of the overall system has to be proven in full compliance with ISO 26262.

This section introduces an approach for hierarchical safety analysis of the "driving functionality" provided by a drive-by-wire vehicle with close functional couplings between individual units. Thereby, especially functional cross compensations between highly safety critical systems that are traditionally investigated separately (e.g. braking, steering, and drive system) are exploited for proof of functional safety. MOBILE serves as suitable demonstration platform as it features a high degree of functional redundancies (Sect. 8.2). In series vehicles, such redundancies are increasingly being introduced. Possible configurations include independently controllable rear axle steering (Pruckner et al. 2012) or superimposed steering systems at the front (Pfeffer and Harrer 2011; Pruckner et al. 2012) and full or hybrid electric drive train structures that allow torque vectoring. Still, cross-actuator functional redundancies are a topic of research and not yet investigated for series vehicles in the context of verification of functional safety.

8.3.1 Summary of Results of the Hazard Analysis

For the vehicle control function of the experimental vehicle MOBILE, a hazard analysis was performed based on ISO 26262. The procedure given in ISO 26262 is adapted to suit the evaluation of an experimental vehicle with high functional integration that is developed from scratch. Thereby, it is argued that (a) there is a strong relation between the evolving system architecture and the hazard analysis that requires iterative re-evaluation of hazards. Additionally, the approach proposed by ISO 26262 is (b) adapted to suite the special conditions of operation of the experimental vehicle. E.g., special circumstances due to operation on a closed off test track with well known environment are taken into account during safety analysis.

[18] IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems, edition 2.0.

[19] Automotive Safety Integrity Levels (ISO26262-1:2011, p. 2).

Table 8.3 Assumptions for operation of the experimental vehicle on the test track

Assumption 1	A **skilled test driver** is driving the vehicle. The driver is capable of handling critical driving situations on a high friction surface if sufficient means to control the vehicle are provided.
Assumption 2	Test runs are only executed in **good weather** (dry road, no rain).
Assumption 3	The driver wears a **protective suit** as, e.g., used in formula one vehicles.
Assumption 4	High speed tests are only performed on a **wide open testing ground** that allows the vehicle to come to a safe halt even if the braking system fails. Thereby, it is assumed that the drive motors can be deactivated by the driver.
Assumption 5	Critical sections of the test track denote sections where the closest **buildings or obstacles** are located at a **minimal distance** of 6 m orthogonally to the track.
Assumption 6	The experimental vehicle features an **emergency-off system** that allows the driver to cut the power of the drive motors at any time.
Assumption 7	For critical sections of the test track, a **speed limit** of 10 m/s is set that has to be obeyed by the test driver. Combined with Assumptions 5 and 6, a worst case impact speed into obstacles in case of a failure of approx. 13.9 m/s (50 km/h) results.[a]

[a]This speed was determined based on simulation experiments with a double track vehicle model, as, e.g., described by von Vietinghoff (2008), assuming different steering concepts, distances to obstacles ranging from 6 m to 10 m orthogonally to the track, a high friction surface, unintended acceleration of the drive motors, and a reaction time of the driver to hit the emergency off of 0.6 s. This reaction time corresponds to typical reaction times of well-trained average drivers, e.g., for emergency braking (Mclaughlin 2007; Mehmood and Easa 2009). The given maximal impact speed was determined for a scenario where the steering angle at the front and rear axle were set to 0.35 rad and 0.09 rad in equal directions.

To start with, Table 8.3 provides an excerpt of important assumptions that were made for the operation on the test track. The given hazard analysis is only valid as long as the vehicle is operated under these conditions. The safe state[20] of the vehicle in case of any failure is defined as follows:

> The vehicle continues to provide basic means for vehicle control to the driver until the vehicle can be safely halted.

Thereby, a linear vehicle dynamics model with front wheel steering and drive that is evaluated online defines the required minimum performance of MOBILE in case of a failure. This emergency operation has to be maintained for a given time interval. More details will be given in Sect. 8.4.

Now, Hazards while driving MOBILE on critical sections of the test track are identified. Therefore, the approach given in ISO 26262 for series vehicles is followed. The results are given as hazards that are evaluated in terms of ASILs[21] depending

[20] ISO 26262 defines the safe state as "the operating mode of an item without an unreasonable level or risk" (ISO26262-1:2011, p.14), while risk refers to the "combination of the probability of occurrence of harm and the severity of that harm" (ISO26262-1:2011, p.13).

[21] The Automotive Safety and Integrity Levels (ASILs) are used to classify hazards according to ISO 26262. ASIL levels range from A (least stringent) to D (most stringent).

Table 8.4 Excerpt of the modified hazard and risk assessment according to ISO 26262

Hazard	Severity	S[a]	Probability of exposure	E[b]	Controllability	C[c]	ASIL
Unintended acceleration leading to a crash	Survival of the driver is uncertain as collisions at high speed are possible.	**S3**	Frequent operation of the vehicle at locations where unintended acceleration can cause collisions	**E4**	A skilled test driver can simply control the vehicle by applying the emergency-off system and/or brakes.	**C1**	**B**
Deviation from the yaw rate reference intended by the driver leading to a crash	Light and moderate injuries are likely at low speeds ($\leq$10 m/s) for a driver wearing a protective suite.	**S1**	Frequent operation of the vehicle at locations where deviation from the yaw rate reference can cause collisions	**E4**	At the given low speed, a skilled test driver can normally control the vehicle by braking.	**C3**	**B**

[a] The levels S0 to S3 classify the severity of an accident. S0 denotes lowest and S3 highest severity
[b] The levels E0 to E4 classify the exposure. E0 denotes lowest and E4 highest exposure
[c]The levels C0 to C3 classify the controllability. C0 denotes best and C3 worst controllability

on the expected controllability,[22] severity,[23] and exposure.[24] Table 8.4 outlines two exemplary hazards that will be used as a reference in the following.

These hazards are then associated to safety goals. Safety goals again serve to derive technical and functional safety requirements for system components. Each safety requirement inherits the ASIL classification from the safety goal and thus from the identified hazards unless ASIL decomposition[25] is applied. If this is done for a highly integrated drive-by-wire system as introduced for MOBILE (see Sect. 8.2), one notes that all requirements on vehicle level that are not associated to the emergency-off system have to be associated to the overall vehicle control function. An association of safety requirements to clearly separated sub-functions may be possible in the functional architecture but is no longer useful if the hardware architecture is taken into account. E.g., one hardware unit contributes to braking, steering, and propulsion function. Figure 8.14 illustrates such a system structure on vehicle level for an

[22] Controllability refers to the "ability to avoid a specified harm or damage through the timely intervention of the persons involved, possibly with support from external measures" (ISO26262-1:2011, p. 4).

[23] In this context, the severity gives an "estimate of the extent of harm to one or more individuals that can occur in a potentially hazardous situation" (ISO26262-1:2011, p. 16).

[24] Exposure classifies the frequency of being in a "an operational situation that can be hazardous if coincident with [the currently investigated] failure" (ISO26262-1:2011, p. 6).

[25] According to ISO 26262 ASIL decomposition denotes the "apportioning of safety requirements redundantly to sufficiently independent elements" (ISO26262-1::2011, p.2).

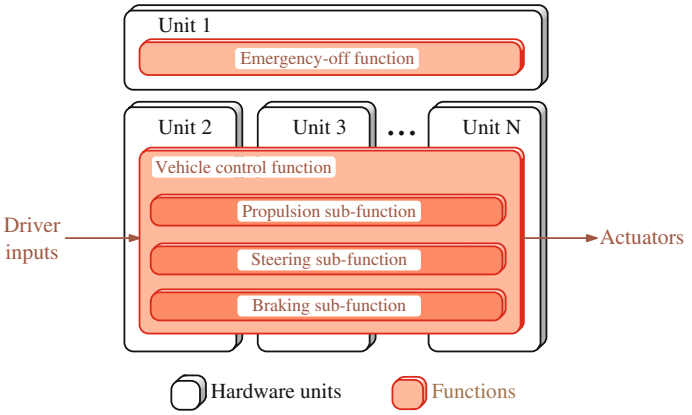


Fig. 8.14 Highly integrated demonstration system

experimental vehicle like MOBILE with emergency-off system and a highly integrated vehicle control system based on drive-by-wire. Thereby, a functional view with hardware units in the background is chosen. The given system consists of several individual hardware units that are combined to fulfill the overall task. A similar tendency towards integration of multiple safety critical functions on one control unit and within one mechatronic components can be seen in modern hybrid electric and electric vehicles: BMW proposes an "integrated chassis management" that closely links different control functions for longitudinal, lateral and vertical dynamics. Additionally, functions provided by one system are re-used by other systems, e.g., a head unit provides data that is used by several other systems as the active steering controller or the Dynamic Stability Control (Koehn et al. 2006; Smakman et al. 2008). Freitag and Kuhn (2012) go even further and replace conventional brakes at the rear axle with an in-wheel motor that drives and brakes the wheels exclusively. Thus, borders between different functionalities and classically separated systems start to blur also in series vehicles. A safety evaluation yields that the failure of one unit may lead to loss or degradation of multiple functions.

Transfered to the simplified system structure given in Fig. 8.14, the sub-functions merge into the overall vehicle control function. Thus, all safety requirements would have to be assigned to the one vehicle control function and the overall underlying hardware consisting of several hardware units. According to ISO 26262, the system would then be associated the highest safety requirements opposed on one of the executed functions. Resulting, safety evaluation according to ISO 26262 presented so far can lead to lower safety requirements on the overall system than intended: If one unit executes multiple functions of lower safety criticality that directly or indirectly effect vehicle handling, the overall criticality of this unit has to be higher than the one of each individual function. E.g., if the vehicle control function in a drive-by-wire vehicle fails, the driver has no means to intervene into vehicle control anymore. This renders the previous hazard analysis wrong and requires adaptation.

Thus, this contribution proposes that all hazards have to be re-evaluated iteratively based on knowledge about the evolving system architecture. Table 8.4 provides the results of this re-evaluation of the two example hazards. Thereby, only hazards are effected that require intervention by the driver by controlling sub-functions of the vehicle control function such as steering or braking. With the re-evaluation finished, the highest ASIL of one of the functional components of a unit can be assigned to the overall unit. For MOBILE, this yields a ASIL B classification of the vehicle control function that is taken as a reference for the safety evaluation introduced in the following sections. In this case, the re-evaluation did not increase safety requirements due to the independent emergency-off system and the well-defined environmental conditions. Still, the re-evaluation is necessary if the system architecture undergoes significant changes during development. Then, changes in hazard classification can occur that are not foreseeable at the beginning of the development project.

Note: It is important to note that the comparatively low safety classification of the experimental vehicle is based on the assumptions of a skilled test driver wearing a protective suit, the well known environment, and the emergency-off system. The emergency-off system features a serial redundancy structure of two emergency-off switches. The system is kept as simple as possible. Thus, it is very likely to be available to the driver in case of failures.

Remark for series vehicles: The hazard analysis for the steering or braking sub-functions would obviously yield an ASIL D classification for series vehicles. E.g., Richter and Köhnen (2012) and Sinha (2011) perform an analysis of these functions for electric vehicles with by-wire design confirming this result. Thus, the correct ASIL D classification of the overall system would already result without re-evaluation. This is intuitively comprehensible as ASIL D already represents the highest possible safety classification. Still, the need for re-evaluation pointed out in this contribution can be transfered to other highly integrated sub-systems in vehicles that execute several functions with lower ASIL levels.

8.3.2 Hierarchical Approach to Safety Analysis

The previous section demonstrated requirements on functional safety for highly integrated vehicle systems. Although, integrated drive-by-wire systems with multiple actuators and without mechanical or hydraulic fall-back layer as referenced in the exemplary hazard analysis are not yet available in series vehicles, the current tendencies in evolution of EE systems indicate similar challenges. High integration of functionalities promises enhanced customer benefit while limiting production costs, and by-wire control can serve as an enabling technology for further progress (Pruckner et al. 2012). The safety evaluation of such systems becomes time consuming and prone to errors (Papadopoulos et al. 2001). To address this challenge, the following

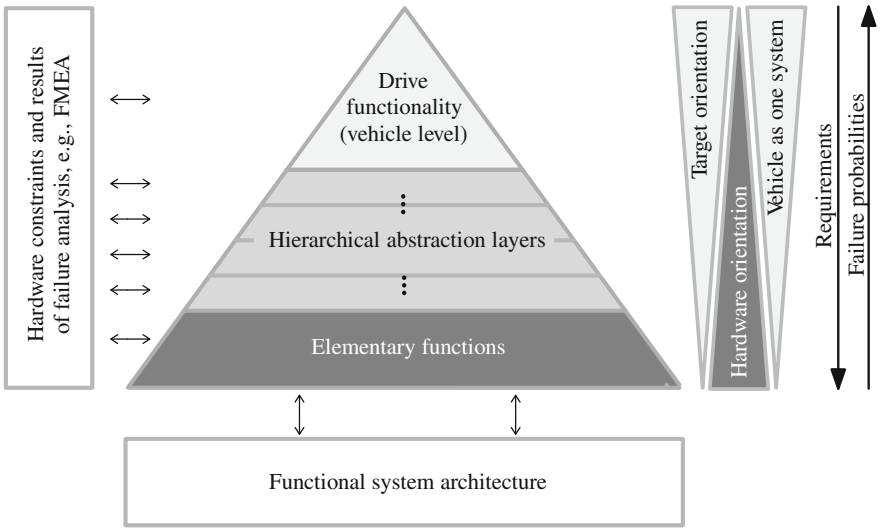


Fig. 8.15 Hierarchical approach to safety analysis

section introduces a tailored method[26] for hierarchical safety analysis that extends the existing approaches by several points: The hierarchical approach especially focuses on highly integrated systems with a *high degree of functional redundancies* and exploits these redundancies for safety purposes. These redundancies are currently hardly addressed at all for quantitative safety assessment. To reduce work effort for the analysis, the presented method introduces virtual systems and generalized failure states that allow to *focus the analysis on necessary components* by front loading knowledge on dependencies in the system. The hierarchical approach results in a *failure rate for the overall system* including the associated *emergency operation interval*. Knowledge about the available emergency operation interval is vital to ensure a safe stop of the vehicle and can possibly be exploited to define a limp-home mode. To assess the performance of the distributed diagnostic algorithms in the vehicle, the approach furthermore provides the diagnostic coverage of a globally operating *virtual diagnostic unit*. This diagnostic coverage can guide further development of the local monitoring algorithms or can be used for safety evaluation while at the same time encouraging "vehicle level thinking". To demonstrate applicability of the method, a prototype tool environment was set up, and the design of MOBILE was evaluated.

Figure 8.15 illustrates the perception of the investigated system adopted by the hierarchical approach. The safety analysis is performed on different hierarchical levels for the overall vehicle and its components. Quantitative results and failure probabilities are propagated bottom-up, while dependencies of components and

[26] A method is "a way of proceeding or doing something, esp a systematic or regular one" Collins (2010). During a development process, (multiple) methods can be applied to achieve necessary results (Hammerschall 2008).

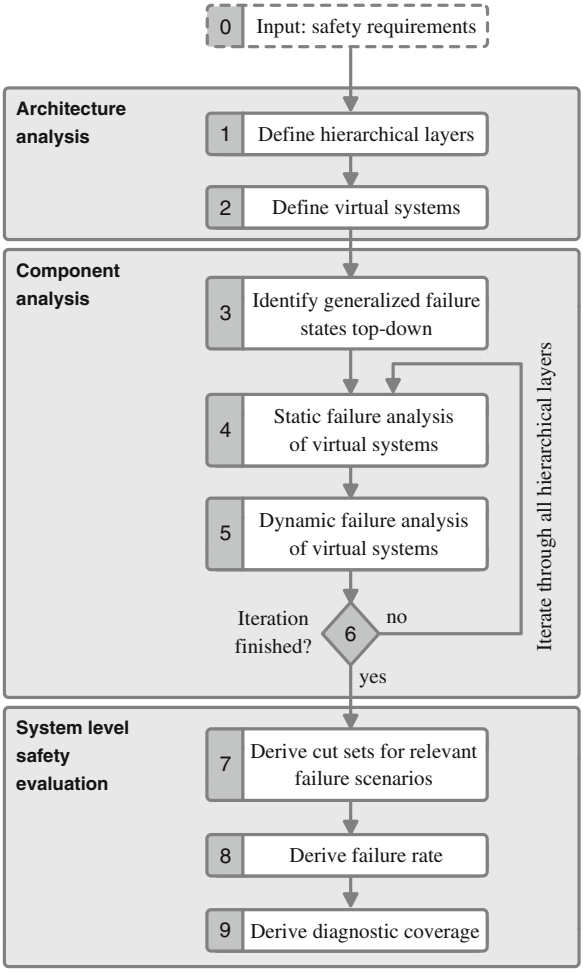


Fig. 8.16 Important steps of the hierarchical approach to safety analysis

requirements are forwarded top-down. The functional structure of the investigated vehicle and the constraints due to the hardware and software architecture are regarded on all hierarchical layers. Thereby, the hardware influence diminishes with increasing hierarchical level, but therefore the required understanding of the overall vehicle by the person in charge strongly increases. Both, profound knowledge about interconnection of vehicle components and knowledge about vehicle dynamics are highly relevant on higher hierarchical levels.

Figure 8.16 outlines the required steps for the hierarchical approach. In general, the process starts with a targeted analysis of the vehicle architecture then investigates relevant components and finally performs the evaluation of functional safety of the overall system. The following sections detail the individual steps and provide according related work.

8.3.2.1 Step 1: Define Hierarchical Layers

To begin with, the architecture of the driving system of the investigated vehicle is analyzed. As a first step, hierarchical layers are defined to support handling of complexity and to serve as a basis for the stepwise safety evaluation. The number of investigated levels varies depending on the investigated system. Layers have to be detailed up to the "result layer". Result layer denotes the hierarchical layer where the investigated unit is located. Thus, the result layer contains the unit that shall be classified as "o.k." or "not o.k.". For this unit, failure rates have to be given and evaluated according to the ASIL classification. For analysis of the vehicle control function, the vehicle layer has to be set as result layer (Fig. 8.2).

Hierarchical layering of systems is frequently applied in research and industry to handle complexity of automotive systems. E.g., Abele (2012) defines "vehicle level", "system and subsystem level" for hierarchical derivation of safety requirements for subfunctions and components of a single ECU in an hybrid electric vehicle. Similarly, Papadopoulos et al. (2001) identify the need for a hierarchically structured approach for safety analysis at the example of a brake-by-wire system.

8.3.2.2 Step 2: Define Virtual Systems

To reduce work effort and focus the analysis process, a novel approach based on virtual systems is introduced. For the safety analysis, it is vital to determine which subsystems are subject to common cause failures, and which ones can be assumed to be independent. Thereby, reasonable splitting of the overall system into subsystems with regard to the safety goals reduces complexity and work effort for the safety analysis. Resulting, within each hierarchical layer independent virtual systems with clearly documented interfaces are defined (Fig. 8.17). For independence of systems, power supply units are of particular importance. The failure of a power supply connected to several other units can obviously cause temporary or partial loss of all supplied units due to over- or undervoltage. This needs not be regarded as a form of dependence at this point, but will be handled later on. If a dependence is assumed, the granularity of the safety analysis is reduced and the safety evaluation becomes more pessimistic unless more effort will be taken in step 3 (explanation given there).

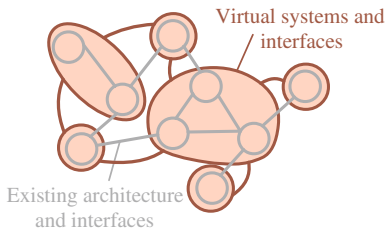


Fig. 8.17 Virtual systems introduced on one hierarchical level

The same strategy can be followed if other units for some reason have significant but similar effect on several other units.

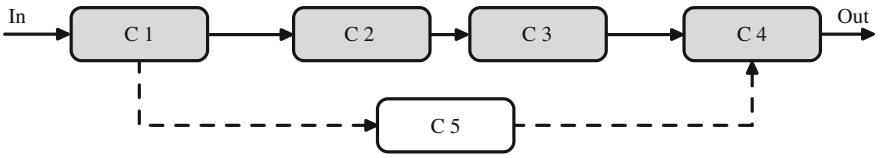
The definition of virtual systems is challenging as the developer requires profound knowledge of the functional, software, and hardware architecture at the given hierarchical level. Still, all following steps of the analysis can then be performed based on the architecture of virtual systems without having to regard other architectural perspectives. The virtual systems ensure linkage between relevant views on the architecture, and overall complexity is reduced by front loading this knowledge. Typically, the definition of virtual systems will be performed middle-out. On the level of control units, independence between different units can be determined easier. Starting from there, the investigations are continued up- and downwards. Thereby, the boundaries of the virtual systems have to remain consistent throughout all hierarchical layers. A system on a lower layer must only be contained in one system at the next higher hierarchical layer.

If on a higher hierarchical layer (unless result layer) a split into independent virtual systems is not possible, this indicates that too many hierarchical layers were introduced or a weakness of the chosen system architecture was identified. The consequences will become obvious in step 7 and will be discussed in more detail there.

8.3.2.3 Step 3: Identify Generalized Failure States Top-Down

Starting with Step 3, failure modes of the virtual systems identified during the architecture analysis are investigated. As a basis for the analysis, the hierarchical approach introduces generalized failure states for the virtual systems defined on each hierarchical layer. These states abstract information on the current failure state of the virtual system. Thereby, only information needed by other systems on the same and especially on higher hierarchical layers is included. This significantly reduces work effort for later quantitative system safety evaluation when compared to existing approaches as, e.g., the HiP-HOPS approach proposed by Papadopoulos et al. (2001). Figure 8.18 provides an example for generalized failure states of a simple system. For the hierarchical approach, these generalized states serve as a well-defined and well-documented interface between experts or suppliers working on different subsystems throughout hierarchical layers. After definition of the generalized failure states, an expert working on a component can focus on a locally well-defined work package, while vehicle level effects are implicitly taken care of.

Still, the definition of the generalized failure states is a challenging task and requires cooperation among experts. It mainly follows two strategies: On the one side, the requirements from a higher layer have to be propagated top-down. This ensures that each state provides sufficient information to an expert working on the higher layer. The expert can then evaluate the overall system based on the pool of generalized failure states from the next lower layer. On the other side, the structure of the investigated virtual system influences the definition of the generalized failure states. Therefore, a rough understanding of the systems purpose and behavior in case



Failure of	Generalized failure state
C1 | C4 | (C5 & (C2 | C3))	Total loss of system
C2 | C3	Emergency operation
No failure | C5	Regular operation

Full operation: ⟶, Emergency operation: ⇢, Logic "or": |, Logic "and": **&**, Component x: Cx

Fig. 8.18 Simplified example of a definition of generalized failure states

of failure is required. As a result, the generalized failure states of a virtual system have to be defined iteratively in cooperation between the affected experts. Typically, the necessary states should be pre-defined top-down to ensure target orientation and then be detailed by an expert for the investigated layer.

After definition of the failure states, each state is assigned a severity top-down-judging from the effect of the failure on the overall system. The original severity on vehicle level stems from the hazard and risk analysis performed according to ISO 26262. The assignment of severity levels is a vital input for the failure analysis of the subsystem detailed in the next step.

To reduce complexity, generalized failure states have to be targeted at safety goals. Unnecessary states, especially in lower layers, increase work effort for the analysis process. Also, too many states for a component can indicate insufficient granularity during definition of virtual systems. “Global failures” that effect several units in a similar way, as a loss of power, should be treated within a generalized failure state of the responsible unit. This procedure is well suited to, e.g., describe the effects of a loss of a central power supply unit.

In literature, the method of introducing generalized failure states is regarded to some extent so far: Sinha (2011) defines generalized failure states for a braking system regarding one hierarchical layer. The states are not exploited for linking systems or to hierarchically propagate severity levels. An application of generalized failure states to a more complex system is outlined by Rehage et al. (2005). The introduced states are identical for all systems (“active”, “isolated”, “active-hot”, “passive-warm”, “passive-cold”) and applicable for aerospace systems with multiple parallel redundancy but are hardly compliant with the requirements to exploit functional redundancies in this automotive project.

8.3.2.4 Step 4: Static Failure Analysis of Virtual Systems

With regard to the generalized failure states, a more detailed analysis of the virtual systems is required. To start with, step 4 performs a "static" failure analysis without taking any timely effects into account. This represents the first step of a bottom-up iteration via steps 4, 5, and 6 (Fig. 8.16).

The failure analysis in step 4 targets at internal failures of virtual systems. On higher levels, the analysis is supported by the results from lower layers, as the investigated system consists of a defined number of already analyzed systems with associated generalized failure states. Different methods support the failure analysis: ISO 26262-4 suggests deductive, e.g., Fault Tree Analysis (FTA) as well as inductive analysis approaches, e.g., Failure Mode and Effect Analysis (FMEA). Details on FTA, FMEA, and further methods are, e.g, given by Rausand and Hoyland (2009) or Löw et al. (2010). This work mainly relies on a slightly modified FMEA that includes possible ways to diagnose and handle failures and a simplified FMEDA (Failure Modes, Effects and Diagnostic Coverage Analysis) to determine quantitative data. For FMEDA, failure rates of software components (control and monitoring algorithms) are included. This extends the classical approach for failure analysis given in ISO 26262 that exclusively refers to hardware components for failure rates. To some extent algorithms are regarded by the demanded diagnostic coverage achieved by certain failure detection mechanisms. In general, software is considered to comply to ASIL requirements if the software was developed according to the guidelines given in the standard. Software is not investigated quantitatively. This approach may be valid for series vehicles with profoundly developed software components and little dependence on external influences. Still, the failure rate given for the vehicle according to ISO 26262 assumes perfect software and is after all only valid for the hardware set-up. For the experimental vehicle, the failure rate of software components has to be considered for two main reasons: Firstly, parts of the software running on the vehicle are prototypical and feature failure rates that are several orders of magnitude higher than the ones of hardware components. These failure rates have to be estimated roughly based on experiences made in previous research projects. Secondly, algorithms as for vehicle stability control are a vital part of the safety concept. These algorithms can not be expected to operate properly under all environmental conditions or for all input configurations. The system, by its design, may just not be able to handle some rarely occurring situations. A failure rate has to be assigned that most likely has to be derived from statistical data acquired with a similar system under similar conditions of operation of the vehicle.

The failure analysis performed during the hierarchical approach profits from the option to apply methods as the FMEA locally: Usually, FMEA has to include failures that globally effect vehicle control, making the evaluation challenging for an expert for the local component. The hierarchical approach allows to evaluate the effects of a failure with regard to the generalized failure states that were defined for the component. The severity needed for the FMEA is then based on the top-down propagated severity associated to the generalized failure state in step 3. As a result, the global context is taken care of. Again, the allocation of tasks to local experts is supported.

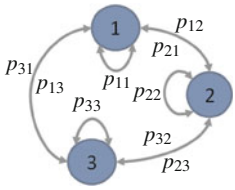


Fig. 8.19 Markov-Chain with three states

Additionally, the linkage of the severity analyses for different components via the top-down propagation supports comparability between results.

8.3.2.5 Step 5: Dynamic Failure Analysis of Virtual Systems

Following, for each available failure state of a system the probability of the system being in that state has to be derived. Therefore, an approach based on first order Markov-Chains is taken. This procedure has already been suggested by Tkachev for the general "analysis of systems with complex structure" in 1983 (Tkachev 1983) and was also followed by Zuo et al. (2005) to perform "quantitative reliability analysis [...] of steer-by-wire system[s]" in the automotive domain. ISO 26262-4 references Markov modeling in general as a valid way to analyze system design, too. A simple first order Markov Chain with three states is given in Fig. 8.19. The p_{ij} resemble the transition probabilities from state i into state j. According to Köhler and Broy (1983) the p_{ij} can be defined as:

$$p_{ij} = P(X_{t+1} = s_j | X_t = s_i). \tag{8.1}$$

Thereby, X_t defines the system state at the time step t and s_i resembles the feature vector characterizing the system state X within state i. As can be seen, the transition probabilities from one state into the other state at a given point in time only depend on the system state at the previous time step (Markov Property for first order Markov Chains). This is an important aspect for failure analysis as the history leading to a certain system state does not have to be known. All background knowledge has to be modeled by the structure of the Markov Chain. For failure analysis, the transition probabilities p_{ij} resemble failure rates λ of parts of a virtual system. These failure rates are derived from elementary hardware or software components on lower layers and are, by means of the hierarchical approach, propagated to any higher hierarchical layer similar to the approach presented by Papadopoulos et al. (2001). For the presented method up to two independent faults occurring one after the other are regarded for definition of the Markov Chain. This ensures that both the reaction of the system to the first fault and the mode of operation afterwards can be evaluated in step 6. Further consecutive faults are not regarded (see also recommendation of ISO 26262-5, Annex C), which reduces work effort. Failure rates and failure states of externally supplied components are directly fed into the system at the appropriate hierarchical layer.

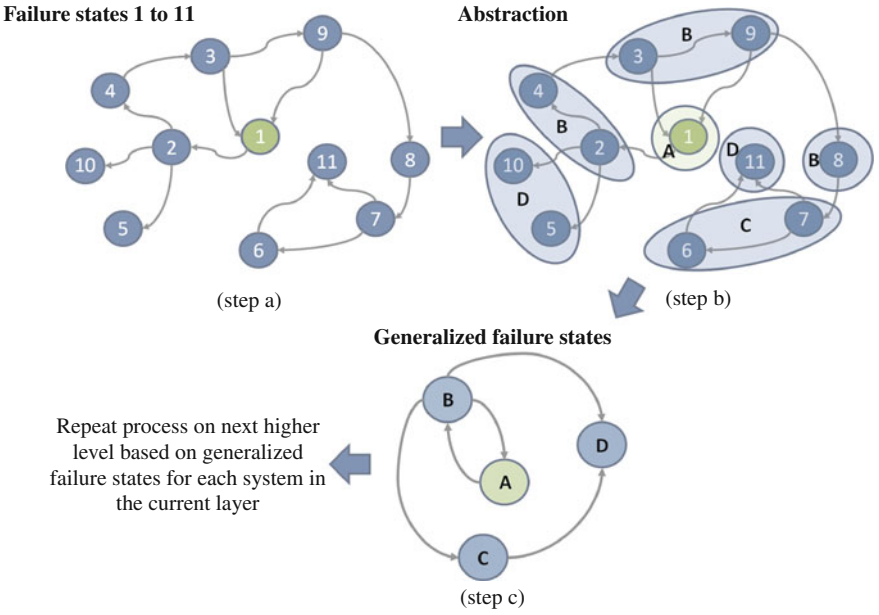


Fig. 8.20 Markov Chain and generalized failure states A to D for a system with failure states 1–11

As all failure rates that are associated to hardware components vary over lifetime due to aging, the Markov chain is not homogeneous and therefore is solved iteratively. The change of failure rates over lifetime of the vehicle is modeled separately. Ideally, the aging models rely on statistical data from systems already in the market. Otherwise, typical aging curves for components can be taken from literature.

After proceeding as outlined, a Markov chain results for each system at a hierarchical layer (Fig. 8.20 step a) with associated failure rates for state transitions. Figure 8.20 depicts only one way transitions as the system is assumed to be *not self-healing* for most failures. Thus, a re-transition from one failure state into a state with less failures is not possible unless, e.g., by a system restart leading through the "ok" state "1". The states of the detailed Markov chain will typically not yet resemble the generalized failure states of the system. The abstraction process is shown in steps b and c of Fig. 8.20. Thereby, states of the initially detailed Markov Chain are associated to the generalized failure states defined in previous steps. The failure rates for transitions between the generalized failure states are calculated from the underlying Markov chain by means of conditional probabilities and by summing up relevant transition probabilities.

8.3.2.6 Step 6: Finish Iteration through hierarchical layers

The evaluation outlined in the steps 4 and 5 is repeated until all hierarchical layers up to the result layer have been analyzed. In the result layer, the final classification of failure effects on the overall system has to be performed. For highly integrated driving systems, the developer has to asses the "vehicle control function" including steering, braking, and propulsion with regard to the effects of failures. The abstracted failure states of the "system layer" serve as input for the evaluation. Thereby, only single and double point faults are regarded:

At first, the developer one by one rates the effect on vehicle handling if a virtual system transitions into an abstracted failure state. The safe state for the vehicle as, e.g., defined for MOBILE in Sect. 8.3.1, serves as reference for minimal acceptable handling characteristics of the vehicle. In a second step, the developer assess the effects of an additional failure within the already faulty system or any other second virtual system. One by one, each virtual system is set to be in one of its failure states. Then, the possible transitions into generalized failure states of this and any of the remaining systems are investigated. Further state transitions do not have to be investigated. Assuming small failure rates, the probability of occurrence of even two independent faults within a given short time interval leading to various system failures is several orders of magnitude smaller than the probability of a single fault. In literature, it is demanded that only one independent fault has to be tolerated by the vehicle control system (Armbruster et al. 2006; Johannessen et al. 2002; X-by-Wire Project 1998). This is also backed up by the current legislative demands, e.g., for approval of the braking system for public traffic (ECE R13). For the safety analysis, it is assumed that more than two independent faults will always lead to a loss of vehicle control, which makes the analysis results slightly more pessimistic. The evaluation of the second fault is needed to determine, whether a sufficient emergency operation interval can be guaranteed to get the vehicle to a safe halt after a first failure occurred. Step 8 details the according calculations.

The above evaluations require profound knowledge of vehicle dynamics. Intense investigation of both the capabilities of actuators relevant for vehicle handling and the associated control algorithms is vital. Several research groups worldwide are investigating these effects and intense research focuses on the novel opportunities in electric or drive-by-wire vehicles. Some examples were given in Sect. 8.2.3.

8.3.2.7 Step 7: Derive Cut Sets for Relevant Failure Scenarios

Starting with step 7, the tool environment, set up for the hierarchical approach, automatically evaluates the so far gathered information numerically. The following two sub-steps are performed in step 7:

1. *Split system into an operational and a faulty part*: Each of the critical failure scenarios based on single or double point faults that were identified for "result layer" implicitly splits the system into an operational part and a faulty part that

is non-operational. The latter part causes the system to fail. Thus, the probability of failure has to be determined for this part. Prior to this, the system split has to be performed throughout all hierarchical layers associating each virtual system to one or the other part. Due to the hierarchical analysis performed so far, the linkage between the virtual systems is known within and between layers by means of the generalized failure states. Thus, the faulty system part can be defined automatically.

2. *Identify cut sets*[27] *for the relevant system part:* The algorithm can now identify all relevant faults and fault combinations that lead to a failure of the investigated part of the system. As already mentioned, only combinations of up to two faults are regarded in this analysis. Resembling the notion in reliability engineering, the algorithm identifies cut sets that lead to the relevant failure state. Thereby, each cut set is determined on a hierarchical layer that provides a complete set of quantitative data to evaluate the effected virtual systems. Typically, this will be the lowest hierarchical layer. Still, this approach also allows to easily integrate third party supplier components that are not detailed to lower hierarchical layers but provide failure rates at higher layers. If needed, the algorithm precisely indicates missing failure rates that have to be provided. Thus, only failure rates that are directly relevant for safety analysis are required. As generation of this quantitative data is costly especially for novel components, the front loading performed by the hierarchical approach due to virtual systems can reduce costs when compared to other approaches that start from a full set of quantitative data (Papadopoulos et al. 2001).

   Each identified cut set is then referenced according to the notion introduced in Fig. 8.21. The superscript for the layer l and the cut set counter i uniquely identify a cut set. The indices s and d are supplementary to highlight whether the cut set is based on a single or double point fault. They constitute redundant information for better readability and later reference. If one of the additional indices or superindices is not given, this references a number of cut sets with all valid combinations for the omitted indices. E.g., C^l resembles all cut sets of the hierarchical layer l and C alone resembles all cut sets on all layers.

   Figure 8.22 illustrates a generic system with a system split derived from a critical scenario determined by the developer. Each system is associated the generalized states “o.k.” or “not o.k.”. The later state is marked by the hatching. Four exemplary cut sets including according deduction paths from layer 2 are given. Each cut set consists of up to two independent faulty units. One cut set belongs to an externally supplied component that is not detailed to the lowest layer. To support processing of the derived data, each cut set has to be unique, and double fault based cut sets have to be pairwise disjoint with single fault based ones:

[27] “A cut set refers to the group of those elements or units which will make the system fail if their failure occurs. The minimum number of such units form the minimal cut set” (Verma and Ajit 2010, p. 85].)

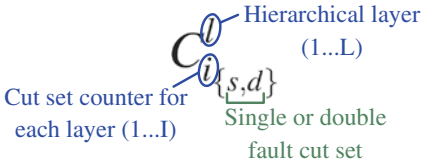


Fig. 8.21 Nomenclature for cut sets

$$C^{l}_{i_s} \cap C^{g}_{j_d} = \emptyset \quad \forall \quad \text{valid combinations of } i, j, l, g \tag{8.2}$$
$$C^{l}_{i_{\{s/d\}}} \neq C^{g}_{j_{\{s/d\}}} \quad \forall \quad i \neq j \wedge l \neq g$$

Thereby, g and j denote values of the cut set counter and the hierarchical layer analogously to i and l. The proposed algorithm ensures disjoint cut sets by the structured segmentation of the system down to any hierarchical layer.

After completion of the above given two sub-steps of step 7, a list of cut sets exists for each failure scenario identified for the "result layer". According to the best knowledge of the developers performing the safety analysis, the combination of these cut sets then forms a minimal cut set for the overall system with the limitation that only up to two independent faults are regarded.

8.3.2.8 Step 8: Derive Failure Rate

Step 8 derives the failure rate of the overall system from the failure rates of components that are associated to identified cut sets. Therefore, two main tasks have to be addressed: At first, the probability of failure of the overall system due to one single or double fault based cut set is calculated. Therefore, both the mission time and the emergency operation interval of the vehicle are regarded. The calculations

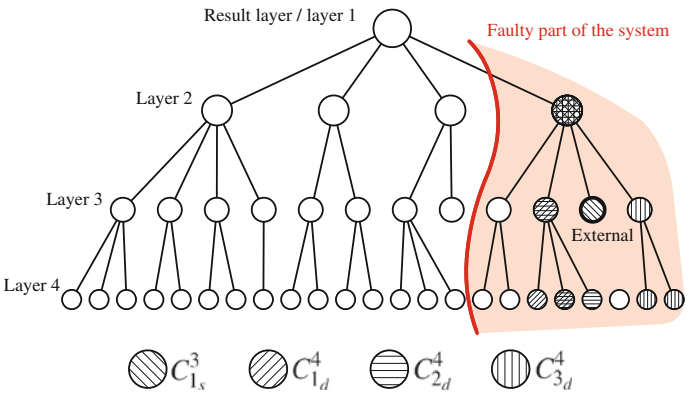


Fig. 8.22 Example cut sets for the faulty part of an example system

assume that aging of the vehicle is negligible during a single mission, and thus the failure rates are approximately constant. Secondly, the failure rates derived for the cut sets have to be combined throughout all hierarchical layers. This produces the overall failure rate of the vehicle. Iterative execution of the calculation in a time loop with modified failure rates regards aging effects. The following sub-steps result:

1. *System failure due to one single fault based cut set*: To start with, the failure rate of the overall system due to one single fault based cut set is determined. For a single fault based cut set, a failure of the system represents a single Markov transition. Accordingly, the failure probability per mission can be derived from the failure rate λ_i^l associated to the i-th single fault based cut set on layer l and the mission time T_M:
$$P(C_{i_s}^l) = 1 - e^{-\lambda_i^l \cdot T_M} \approx \lambda_i^l \cdot T_M \quad \text{for small lambdas.} \tag{8.3}$$

2. *System failure due to one double fault based cut set*: For calculation of the failure rates due to double faults, an approach presented by Sieglin (2009) is adopted. Sieglin (2009) derived a formula to calculate the failure probability of a power supply system consisting of two independent units with failure rates λ_1 and λ_2 and a diagnostic unit. The structure of this duplex system is given in Fig. 8.23. The probability p_{fail} of a system failure due to failure of both independent units can be calculated as:

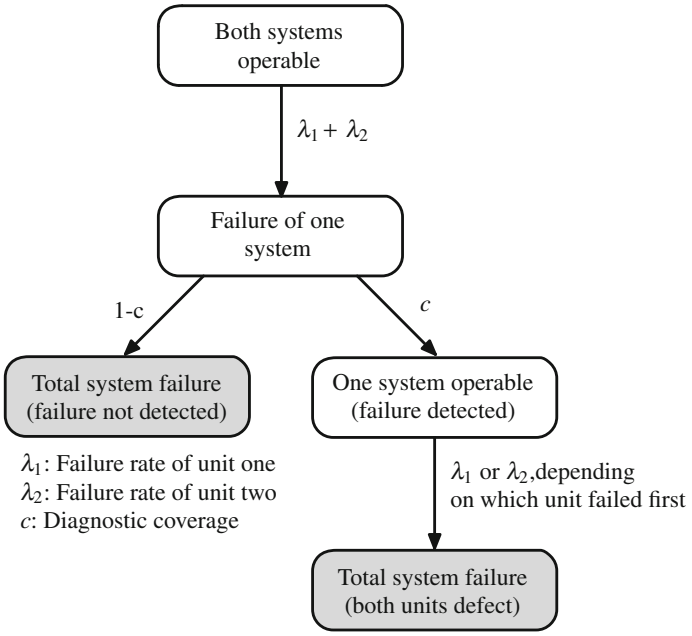


Fig. 8.23 State transitions in case of failure for a duplex system with diagnostic unit; figure similar to Sieglin (2009)

$$p_{\text{fail}} = 2\lambda_1\lambda_2(T_M T_{SS} - \frac{1}{2}T_{SS}^2), \tag{8.4}$$

with T_{SS} resembling the emergency operation interval. The formula can only be applied if three assumptions are valid: (a) At the start of a mission, all systems have to be in ok state. This can, e.g., be ensured by a detailed self check. (b) The exponential function $f(\Delta t) = 1 - e^{-\lambda \Delta t}$ can be approximated by the linear function $f(\Delta t) = \lambda \Delta t$ for small products of failure rates λ and time intervals Δt. And (c), the failure rates do not change depending on the order of occurrence of the two failures.

For the hierarchical approach the assumptions (a) and (b) are valid. Assumption (c) is mostly fulfilled due to the definition of independent virtual systems. If this assumption is not valid, the developer can easily introduce different failure rates for different orders of failures as the two failure scenarios with one failure occurring before the other one are treated independently during system analysis anyways. Thus, the formula can directly be used for most cases:

$$P(C_{i_d}^l) = 2\lambda_{i_1}^l \lambda_{i_2}^l (T_M T_{SS} - \frac{1}{2}T_{SS}^2). \tag{8.5}$$

Thereby, $\lambda_{i_1}^l$ and $\lambda_{i_2}^l$ resemble the failure rates of the first and second fault associated to the double fault based cut set with counter value i on the hierarchical layer l. The factor two ensures that both scenarios with one of the units failing first are already included. If for some failure combinations only one order of failure occurrence is possible or failure rates change for different orders, the evaluation algorithm accordingly neglects the factor 2 and treats both cases independently. Thus, the presented approach provides intrinsic means to handle timely effects. Other approaches have to make special extensions to handle these aspects as introduced by Mahmud et al. (2010) or Walker and Papadopoulos (2009) by means of temporal fault trees.

3. *Combination of all failure probabilities throughout all hierarchical layers*: To combine the failure probabilities throughout all hierarchical layers, a bottom up approach is followed. Starting with the lowest layer, the failure probabilities are combined. Then, the process is repeated on the next higher layer for the so far untreated cut sets. Thereby, the failure probabilities of the lower layer can simply be added due to independence of the cut sets before propagating the result to the next higher layer.

   The combined failure probability of multiple cut sets can be calculated as given in reliability engineering (Verma and Ajit 2010, p. 22). Transfered to the notion of the hierarchical approach, the following formula results for the failure probability within one hierarchical layer:

$$
\begin{aligned}
P(C^l) = (+1)\cdot & \sum_{i=0}^{I} P(C^l_{i_{\{s/d\}}}) \\
(-1)\cdot & \sum_{i=0}^{i=I-1} \sum_{j=i+1}^{j=I} P(C^l_{i_{\{s/d\}}} \cap C^l_{j_{\{s/d\}}}) && \} \, \mathrm{OM}(\lambda^3) \approx 0 \\
(+1)\cdot & \sum_{i=0}^{i=I-2} \sum_{j=i+1}^{j=I-1} \sum_{k=j+1}^{k=I} P(C^l_{i_{\{s/d\}}} \cap C^l_{j_{\{s/d\}}} \cap C^l_{k_{\{s/d\}}}) && \} \, \mathrm{OM}(\lambda^4) \approx 0 \\
& \cdots \\
(-1)^{I+1}\cdot & P(C^l_{1_{\{s/d\}}} \cap C^l_{2_{\{s/d\}}} \cap \ldots \cap C^l_{I_{\{s/d\}}}) && \} \mathrm{OM}(\lambda^{I+1}) \approx 0.
\end{aligned}
\tag{8.6}
$$

Thereby, I denotes the number of cut sets within one hierarchical layer, j and k are helping variables defined based on i. OM$(\cdot)$ qualitatively resembles the order of magnitude of the computational terms in terms of a typical failure rate λ. The orders of magnitude are derived from Eqs. 8.3, 8.5, and the assumptions given in Eq. 8.2. Thus, cuts between a single and any double point fault based cut set do not exist. Resulting, only double fault based cut sets have to be regarded in line two and the following lines of Eq. 8.6. The cuts are calculated based on conditional probabilities resulting in the given order of magnitude in terms of λ. For example, the probability of a cut between two cut sets with identifiers i and j of the same hierarchical layer with one common fault within each cut set is determined as:

$$
P(C^l_{i_d} \cap C^l_{j_d}) = P(C^l_{i_d}) \cdot P(C^l_{j_d} | C^l_{i_d}) \propto \lambda^l_{i_1} \cdot \lambda^l_{i_2} \cdot \lambda^l_{j_2} \quad \} \, \mathrm{OM}(\lambda^3) \tag{8.7}
$$

Thereby, the faulty unit that is part of both cut sets is associated the failure rates $\lambda^l_{i_1}$ and $\lambda^l_{j_1}$, respectively. Based on the assumption of small lambdas, all terms with an OM higher than λ^2 are neglected.

Concluding step 8, the failure rates of the overall system per mission F results by bottom-up addition of all $P(C^l)$ derived for each hierarchical layer:

$$
F = P(C) = \sum_{l=1}^{l=L} P(C^l). \tag{8.8}
$$

Resulting, compliance with the given ASIL requirements can be verified.

8.3.2.9 Step 9: Derive Diagnostic Coverage

Finally, the diagnostic coverage (DC) on vehicle level can be estimated based on the results of the hierarchical approach. As defined for an FMEDA, the diagnostic coverage is calculated as (Löw et al. 2010):

$$DC = \frac{\lambda_{dd}}{\lambda_{dd} + \lambda_{du}}. \tag{8.9}$$

λ_{dd} and λ_{du} denote the cumulative probabilities of occurence of any dangerous failure that is detected (dd) or remains undetected (du).

This approach is now transfered to vehicle level. Thereby, it is important to note that the highly integrated system does not feature one separate diagnostic unit as, e.g., introduced by Sieglin (2009) and indicated in Fig. 8.23. The functionality of the diagnostic unit is distributed throughout the whole network. If furthermore functional redundancies between different actuators are regarded, inevitably the classical perception of the diagnostic unit has to be modified towards a globally operating system. The system may include complex knowledge about vehicle dynamics and the overall vehicle network.

The basic idea of the calculation is as follows: Due to targeted abstraction based on the generalized failure states, the hierarchical approach only regards dangerous faults. Furthermore, it is assumed that all safety critical functions are executed at least on redundant units that are insusceptible to common cause failures as otherwise the high safety requirements can not be met anyways. Thus, any functional unit can be seen as being structured similar to Fig. 8.23. Any single point failure leading to a failure of the overall system then must be caused by a failure of the diagnostic system and must contribute to λ_{du}. Resulting, the diagnostic coverage DC for a system that only has to tolerate one independent fault can be calculated according to Eq. 8.9 by relating faults that lead to an immediate failure of the system (λ_{du}) and faults that allow the vehicle to transition into its safe state (λ_{dd}).

The diagnostic coverage calculated as outlined then indicates the performance of the distributed diagnostic algorithms and signals whether the failure rate of the system is driven by high failure rates of individual units or missing quality of the diagnostic algorithms. On demand, automatic hints can be generated for which functions/units the performance of the distributed diagnostic algorithms is the lowest. Then, experts for the local system can derive according solutions.

8.3.3 Criticism of the Hierarchical Approach

This section provides a summarized criticism of the hierarchical approach to review the fields of application, novel contributions, limitations, and further usage of the generated data.

Fields of application: In modern vehicles, borders between individual safety critical functionalities as steering, braking and propulsion start to vanish. This becomes even more challenging if full-drive-by-wire vehicles are regarded. To limit additional costs for such systems due to numerous redundant hardware parts, integration of functionalities is unavoidable. Additionally, functional redundancies between different types of actuators have not yet been exploited for functional safety. Judging from the promising research results in the field of vehicle dynamics, these functional

redundancies flanked by degradation concepts may hugely support cost-effective integration of highly safety critical EE systems into modern vehicles. Dedicated methods for safety analysis of such systems are missing. The hierarchical approach targets to close this rising gap.

Novel contribution: Summarized, some key aspects distinguish the hierarchical approach from other approaches:

- The hierarchical approach presents a *targeted way to system safety evaluation*. Thereby, front loading of knowledge about dependencies and critical states (virtual systems and generalized failure states) ensures that only relevant systems and faults have to be regarded in the later process. A goal-oriented split of the system in safety relevant components—away from the traditional domain oriented splits—is performed. Resulting, a high degree of abstraction can be achieved while still maintaining mathematical linkage for quantitative evaluation and proper documentation. Thus, especially the effort to derive quantitative failure rates can be reduced compared to other approaches (Papadopoulos et al. 2001).
- The hierarchical approach provides a tailored *structure function*[28] of the investigated system that neglects unnecessary components. The structure function can be visualized in different ways or be reused for further efficient analysis of the system as, e.g., shown by Adachi et al. (2011), Herath et al. (2007), Rehage et al. (2005), or Sinha (2011).
- The tool chain implemented in the project highlights components with the highest impact on failure rates but also *indicates the contribution to the safety concept* on each hierarchical layer. Thus, the approach supports system level thinking and encourages failure handling on all hierarchical layers.
- The hierarchical approach provides a failure rate for the overall system taking into account a configurable *emergency operation interval* and *failure rate estimates of dedicated software* functions. Additionally, the *diagnostic coverage* at vehicle level is approximated based on single and double fault based cut sets and the according probabilities of failure.
- Due to the dedicated re-partitioning of the system into safety relevant components, new safety concepts can become obvious that are not supported by domain oriented thinking. Analogously, *functional redundancies* can be exploited for safety evaluation. On top level, these redundancies can be integrated into the safety concept intuitively. On the basis of the resulting system structure, algorithms as presented by Herath et al. (2007) could be used for optimized allocation of failure rates or failure detection mechanisms within components with regard to overall system failure rate and costs.

Limitations: The hierarchical approach contributes when evaluating a system in terms of safety. Naturally, there are clear limitations of application.

1. As already pointed out, the hierarchical approach is neither a process model nor a method that primarily supports the development of a product. The hierarchical

[28] The structure function defines the "dependence of the system state on the state of its components" (Gertsbakh (2000), p.1).

approach supports evaluation of an already drafted vehicle architecture. Still, iterative application can support system development.

2. Additionally, the strong tailoring of the hierarchical approach on safety evaluation on the one side reduces work effort, but on the other side may also be unsuitable for extended investigation of a system, e.g., for a full investigation of reliability measures. If the hierarchical approach is extended to also investigate according scenarios, work effort approaches the one of already existing methods. The other way round, if generalization during the hierarchical approach is overdone to further decrease work effort, safety estimates may become more pessimistic. Still, results from common methods as FMEA support the developer to set the abstraction level appropriately.
3. The hierarchical approach is focused on quantitative evaluation of failure rates but does not evaluate fulfillment of process requirements opposed by ISO 26262. Process requirements derived from ASIL levels are a vital aspect for safety evaluation (Palin et al. 2011) and significantly contribute to development costs. The hierarchical approach could contribute to reduction of these local requirements by identification of local redundancies that could then allow ASIL decomposition. Still, handling of functional redundancies across different types of actuators in term of ASIL classification is a so far completely un-investigated topic in research and development.
4. Until now, the hierarchical approach focuses only on the basic vehicle control functions. Other functions provided by the human machine interface are not regarded. Still, this contribution holds the view that none of these aspects is relevant if the driver can no longer control the main actuators of the vehicle. Thus, the vehicle control system forms the basis for any other applications and should be treated separately. This perception is, e.g., backed by the generic safety life cycle for intelligent transport systems, especially Driver Assistance Systems, outlined by Carsten and Nilsson (2001).

Outlook: The information on system architecture and dependencies gathered by the hierarchical approach could be further exploited for online knowledge representation in the vehicle. Based on the known dependencies among systems and faults, the diagnostic heuristics could be (automatically) complemented to specifically take into account interactions on vehicle level. Multiple possible applications are thinkable that could make use of the well structured information on the system architecture from a safety point of view. Especially, systems dealing with self representation and online failure handling by degradation are possible fields of application and are investigated in the project MOBILE.

8.4 Safety Evaluation of MOBILE with the Hierarchical Approach

This section evaluates the safety of the vehicle control function of MOBILE using the hierarchical approach introduced in the previous section. As MOBILE is a primarily student driven university project, some restrictions have to be regarded:

- Up to 20 students were working on parts of the vehicle in parallel. Each student works on a specialized field on a low hierarchical level. The students are assumed to be the experts for a specific field. The work on higher levels is mostly done by members of the scientific staff.
- Failure rates are not available for all parts of MOBILE. For these parts typical values were derived from literature. Failure rates of software under development is roughly estimated based on previous in-field experience.
- Aging effects are approximated by typical bath-tub curves given in literature as, e.g., by (Reif 2009, p. 261) or (Verma and Ajit, 2010, p. 2). Thereby, one observes slightly higher failure rates of hardware components in early phases of the part's life time and a significant increase towards the end of the life time.
- Only the hierarchical layers "vehicle", "system" and "subsystem" have so far been taken into account (see also marks in Fig. 8.2). Some individual components of the underlying hierarchical layers are currently being investigated and results are fed back to the evaluation of MOBILE.
- Processes performed during development of MOBILE do not comply to the requirements imposed by ISO 26262.
- As the construction of MOBILE is not yet finished, only qualitative results are given that origin from quantitative but not yet complete data.

Still, the safety evaluation of MOBILE demonstrates the applicability and benefits of the hierarchical approach. Quantitative figures give a rough impression of the safety level. Additionally, relative changes in failure rates after modifications to components or the architecture of electronics can be observed. Working with a group of students showed that the hierarchical approach supports splitting of the complex vehicle design task into a number of smaller work packages that are easier to handle. At the same time, the system context is kept available and traceable for all developers.

8.4.1 Assumptions for the Safety Analysis of MOBILE

As indicated in Sect. 8.3, top level assumptions on the mode of operation of MOBILE are required to evaluate the functional safety of MOBILE: The *mission time* of MOBILE is limited to 30 min. After 30 min, the lead-acid drive batteries are assumed to be emptied anyways or the test driver is expected to have a break. In case of a failure, the emergency operation interval that has to be guaranteed is set to 30 s. This time span suffices to get MOBILE to a safe halt even if the failure occurred

while driving at MOBILEs top speed of approx. 160 km/h (44 m/s). It is assumed that *only one independent fault* has to be tolerated. For MOBILE, one "point in time" is defined as a 4 ms time slot. This slot length is derived from the cycle time of the FlexRay network in MOBILE that facilitates synchronization of all network nodes and precise triggering of the diagnostic algorithms. If two faults occur within a 4 ms time slot, they are treated as a double fault at one point in time. A similar assumption for small diagnostic time intervals is, e.g., made by Sieglin (2009).

8.4.2 Evaluation of Complexity of the Hierarchical Approach

On "system layer" of MOBILE, eight units were defined: front and rear axle control system consisting of the according FTUs, user input control system (also embodied by the according FTU), two power supply systems, emergency off systems for front and rear drive motors and the stability control system (Table 8.5). Due to the design of MOBILE, these systems can be regarded as insusceptible to common cause failures—except loss of power. If cross couplings between the systems exist, the couplings are assumed to be irrelevant during the emergency operation interval of 30 s, e.g., low voltage buffer batteries can compensate the loss of charging power due to failure of the high voltage system. In particular, this independence of the elements at "system level" led to the definition of the virtual systems as given and not to the classical system partitioning into braking, drive and steering system. For each of the chosen virtual systems, 2 to 9 generalized failure states, not including the "ok"/"no failure present" state, were defined. Resulting, 31 failure states have to be evaluated on vehicle layer. Thereby, the controllability of the vehicle has to be evaluated after occurrence of a given first and second system failure, summing up to 702 state transitions. Especially, for the second faults, several transitions need not be regarded as they do not furthermore impact the controllability of the vehicle. Additionally, several transitions are identical for more than one system and thus only have to be considered once. For each failure scenario, the state of the vehicle is well defined as the generalized failure states are part of a first order Markov Chain. Thus, all relevant information is contained in the state descriptions and no knowledge about the failure history is needed. Given the knowledge about the effects of the system failures on vehicle dynamics, it takes the developer approximately an hour to go through all states and define the according consequences. Table 8.5 shows a simplified classification for MOBILE after the first failure for each system. Within the tool environment, the classification is done graphically based on Excel tables by color highlighting. For the evaluation of MOBILE on vehicle level, several experiments with a 1:5 scale vehicle were performed to estimate the effect of actuator or power supply failures on the controllability of the vehicle (Töpler 2010; Lieberam 2011; Goldschmidt 2012). Additionally research results of other groups were taken into account to fully exploit functional redundancies. Still, the classification at vehicle level is a challenging and not fully solved task from a scientific point of view but easy to handle formally, which allows the researcher to focus on his main tasks.

Table 8.5 Graphically assisted failure classification at vehicle level

system	generalized failure states/effect of first system failure				
FAC_Sys	destabilizing	neutral	ok		
RAC_Sys	destabilizing	neutral	ok		
EOffVA_Sys	defect off	defect on	ok		
EOffRA_Sys	defect off	defect on	ok		
ESup1_Sys	all off	12V off	48V off	HV off	... ok
ESup2_Sys	all off	12V off	48V off	HV off	... ok
SC_Sys	destabilizing	off	ok		
UI_Sys	defect	only loss of braking	only loss of steering	ok	

key:
FAC_Sys / RAC_Sys: front/rear axle control system
EOffVA_Sys / EOffRA_Sys: Emergency off system for front/rear axle
ESup1_Sys / ESup2_Sys: power supply 1/2
SC_Sys: stability control system
UI_Sys: user interfacing system
vehicle operable after failure of system: yes, no, no failure

The failure states on "system layer" are derived from approximately 60 failure states on "subsystem layer". In average, on system level approx. 100 state transitions have to be evaluated per system. Thereby, the behavior of the system for all "first faults" has to be considered. Additionally, selected "second faults" have to be investigated. Second faults that have to be regarded are identified automatically top down from "vehicle level". The number of state transitions that have to be investigated by the developer serves as an estimate for work load and complexity. If compared to "vehicle level" and depending on the individual system, the individual researcher on system level has to evaluate a similar amount of relevant combinations.

On lower levels (component and elementary) the number of total failure states furthermore increases but again can be handled due to the partitioning into virtual systems and allocation of tasks to local experts. Third party components can easily be integrated at any hierarchical level. Within the project MOBILE several such components exist (steering motors, drive motors, etc.).

As mentioned, the evaluation process in the project MOBILE is supported by an Excel Sheet. Necessary calculations and the linking between hierarchical layers are automatically derived from "graphical" inputs of the user (compare Table 8.5). As the input tables are continuously being updated during the development process, the current state of the vehicle with regard to safety as well as the most critical components are known at any point in time. The generalized failure states including proper documentation support transparency and long time usability of the results of the safety analysis. These state descriptions also form the basis for discussions between experts in different fields and on different hierarchical levels. A further extension of the tool environment to automatically link graphical architecture descriptions (fault

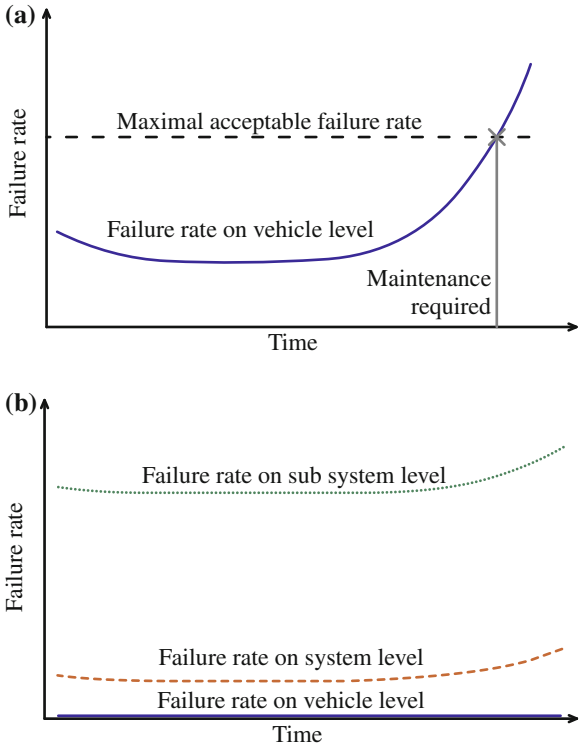


Fig. 8.24 Qualitative failure rates on "vehicle layer" (**a**) and for comparison on "vehicle", "system" and "subsystem layer" (**b**) over lifetime of the vehicle

trees, reliability block diagrams) or descriptions of state transitions (Markov chains) with the inputs in the Excel environment would be useful. Currently, these steps are performed manually, which is acceptable for the scope and scale of the project.

Summarized, the analysis results for MOBILE can serve as a well documented and tailored safety report and support continuous monitoring during development. The tailoring of the analysis by front loading knowledge on dependencies lowers work effort compared to other hierarchically structured approaches.

8.4.3 System Monitoring and Failure Rates

Figure 8.24a illustrates the failure rates of MOBILE at vehicle level over lifetime. Thereby, the failure rate was calculated using the approach detailed in Sect. 8.3 for several points in time. The curvy form of the graph with high increase in failure rates towards the end of the vehicle lifetime results from the assumed aging of hardware parts. As introduced in Sect. 8.3.2.4, software parts that feature a high

probability of failure are also taken into account—differently from the approach in ISO 26262. Of course, these failure rates are highly volatile, but are several orders of magnitude higher than the failure rates of the underlying hardware and thus have to be considered. Of course, software components are unconcerned by aging.

Figure 8.24b visualizes the huge benefit for failure compensation in the vehicle by considering interactions at "system" and "vehicle layer". E.g., the curve for "system layer" considers only cross-compensations between different systems up to "subsystem level" and so on. On higher layers, these cross-compensations are more and more due to functional redundancies. Thus, a highly flexible vehicle as MOBILE especially profits. Analogously, the efficiency of the diagnostic coverage over lifetime is automatically derived from the gathered data.

Tendencies show, that the proposed integrated safety concept relying on functional redundancies can increase functional safety while also maximizing the functional benefit from additional actuators and limiting system costs due to reduction in required hardware redundancy. Still, final results can only be provided after the vehicle has been completed, and further experiments can be conducted.

8.4.4 Conclusion

This contribution introduces a novel system architecture for an experimental drive-by-wire vehicle with high functional integration and over-actuation. For this vehicle, a system architecture is derived top-down driven by according requirements. Especially, the top-down partitioning of the system can reveal novel structures also for series vehicles. Resulting, a system structure that exploits functional redundancies instead of hardware redundancies for safety purposes is presented. Exploiting functional redundancies necessitates a clearer definition of the safe state of the vehicle compared to typical part-oriented safe-state assumptions. For MOBILE, a model of the desired minimal vehicle dynamics is used. Consequently, control algorithms for vehicle dynamics play an important role in the proposed safety concept, and assessment of quality of these algorithms has to become more quantitative.

To evaluate the safety of complex and integrated systems as proposed for MOBILE, a hierarchical approach to safety analysis is introduced. The approach complements already existing means for safety evaluation by taking a holistic view of the overall vehicle. It especially focuses on the targeted evaluation of highly integrated systems that provide functional redundancies. Therefore, virtual systems and generalized failure states support early reduction of the number of faults that have to be analyzed quantitatively. At the same time, system partitioning promotes allocation of work packages to developers that are best suitable. As given, the proposed approach features some restrictions and potential for further development. Especially, questions related to a development process in industry as intellectual property, responsibilities, or process management are not regarded in this contribution.

Future work will focus on completion of the safety evaluation of MOBILE. Starting from there, the further usage of the structured information on the system architec-

ture for online self-representation of the vehicle and diagnostics will be investigated. Another important topic of future work will be the ongoing evaluation of control algorithms for vehicle dynamics to exploit functional redundancies between different types of actuators by coordinated control of remaining actuators. In parallel, analysis of critical components of the EE system will go on with regard to functional safety and failure rates.

References

Abele, A.: Design and realization of an integrated safety concept based on an architecture model with the given example for the serial development of a powertrain control unit used in electric driven vehicle. In: Hybrid and Electric Vehicles, pp. 481–525. Braunschweig (2012)

Abele, M.: Modellierung und Bewertung hochzuverlässiger Energiebordnetz-Architekturen für sicherheitsrelevante Verbraucher in Kraftfahrzeugen. Ph.D. thesis, Universität Kassel, Kassel (2008)

Adachi, M., Papadopoulos, Y., Sharvia, S., Parker, D., Tohdo, T.: An approach to optimization of fault tolerant architectures using HiP-HOPS. Softw. Pract. Experience **41**(11), 1303–1327 (2011)

Anwar, S., Niu, W.: Analytical redundancy based predictive fault tolerant control of a steer-by-wire system using nonlinear observer. In: 2010 IEEE International Conference on Industrial Technology, pp. 477–482 (2010)

Arbitmann, M., Raste, T., Lauer, P., Kelling, E., Eckert, A., Rieth, P.E.: Motion Control—Zentraler Baustein zukünftiger funktional strukturierter Domänenarchitektur im Fahrzeug. In: AUTOREG 2011, pp. 375–387. Baden-Baden (2011)

Armbruster, M.: Eine fahrzeugübergreifende X-by-Wire Plattform zur Ausführung umfassender Fahr- und Assistenzfunktionen. Ph.D. thesis, Universität Stuttgart, München (2009)

Armbruster, M., Zimmer, E., Lehmann, M., Reichel, R., Sieglin, E., Spiegelberg, G., Sulzmann, A.: Affordable X-By-Wire technology based on an innovative scalable E/E platform-concept. In: IEEE 63rd Vehicular Technology Conference, pp. 3016–3020. Melbourne, Australia (2009)

Beal, C.E., Gerdes, J.C.: Experimental validation of a linear model predictive envelope controller in the presence of vehicle nonlinearities. In: 6th IFAC Symposium on Advances in Automotive Control. Munich (2010)

Bergmiller, P., Ibele, P., Maurer, M., Gerdes, J.C.: Development tool for dynamic drive control systems. ATZelektronik worldwide **2011–03**, 60–67 (2011)

Bergmiller, P., Maurer, M.: Flexible Versuchsträger als Testplattform für Antriebskonzepte in Elektrofahrzeugen. In: Schäfer, H. (ed.) 2012, Trends in der elektrischen Antriebstechnologie für Hybrid- und Elektrofahrzeuge, pp. 232–243. Expert Verlag, Renningen (2012)

Bergmiller, P., Maurer, M., Lichte, B.: Probabilistic Fault Detection and Handling Algorithm for Testing Stability Control Systems with a Drive-By-Wire Vehicle. In 2011 IEEE International Symposium on Intelligent Control (ISIC), pp. 601–606. Denver (CO), USA (2011b)

Bernard, M., Buckl, C., Döricht, V., Fehling, M., Fiege, L., von Grolmann, H., Ivandic, N., Janello, C., Klein, C., Kuhn, K.-J., Platzlaff, C., Riedl, B.C., Schätz, B., Stanek, C.: Abschlussbericht des vom Bundesministerium für Wirtschaft und Technologie geförderten Verbundvorhabens "eCar-IKT-Systemarchitektur für Elektromobilität". ForTISS GmbH, Garching (2010)

Bertacchini, A., Pavan, P., Tamagnini, L., Fergnani, L.: Control of brushless motor with hybrid redundancy for force feedback in steer-by-wire applications. In: 31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005, pp. 1407–1412. Raleigh, USA (2005)

Blanc, S., Bonastre, A., Gil, P.: Dependability assessment of by-wire control systems using fault injection. J. Syst. Archit. **55**(2), 102–113 (2009)

Carsten, O.M.J., Nilsson, L.: Safety assessment of driver assistance systems. Eur. J. Transp. Infrastruct. Res. **1**(3), 225–243 (2001)
Collins: Collins English Dictionary 30th Anniversary Edition, 10th edn. William Collins Sons & Co. Ltd, London (2010)
Collinson, R.: Fly-by-wire. Comput. Control Eng. J. **10**(4), 141 (1999)
Cornelsen, K., Jänsch, D., Gerson, S., Nietschke, W., Maurer, M., Canders, W. R., Schumacher, W., Meyer, H.: InDrive Simulator—Innovative Tool for Simulating and Designing Complex Drive Structures in Real Operation. In: Hybrid and Electric Vehicles, pp. 166–186. Braunschweig (2011)
Dilger, E., Karrelmeyer, R., Straube, B.: Fault tolerant mechatronics [automotive applications]. In: 10th IEEE International On-Line Testing Symposium, pp. 214–218. IEEE Computer Society (2004)
Dominguez-garcia, A.D., Kassakian, J.G., Schindall, J.E.: A Backup System for Automotive Steer-by-Wire, Actuated by Selective Braking. In: 35th Annual IEEE Power Electronics Specialists Conference, pp. 383–388. Aachen (2004)
Euchler, M., Bonitz, T., Mitte, D., Geyer, M.: Bewertung der Fahrsicherheit eines Elektrofahrzeugs bei stationärer Kreisfahrt. ATZ - Automobiltechnische Zeitschrift **2010–03**, 206–213 (2010)
Freitag, G., Kuhn, K.-J.: Hochintegrierter Antrieb: Radnabenantrieb ohne Reibbremse. In: Schäfer, H. (ed.) Trends in der elektrischen Antriebstechnologie für Hybrid- und Elektrofahrzeuge, pp. 73–83. Expert Verlag, Renningen (2012)
Gadda, C.D., Laws, S.M., Gerdes, J.C.: Generating diagnostic residuals for steer-by-wire vehicles. IEEE Trans. Control Syst. Technol. **15**(3), 529–540 (2007)
Gertsbakh, I.: Reliability Theory With Applications to Preventive Maintenance. Springer, Berlin (2000)
Goldschmidt, D.: Entwicklung eines fahrdynamischen Stabilitätsprogramms für ein Drive-by-Wire-Versuchsfahrzeug. Diplomarbeit, TU Braunschweig (2012)
Hammerschall, U.: Flexible Methodenintegration in anpassbare Vorgehensmodelle. Technische Universität München, Dissertation (2008)
Hasan, M.S., Anwar, S.: Sliding mode observer based predictive fault diagnosis of a steer-by-wire system. In: Proceedings of the 17th International Federation of Automatic Control World Congress, pp. 8534–8539. Seoul, Korea (2008)
Hayama, R., Higashi, M., Kawahara, S., Nakano, S., Kumamoto, H.: Fault tolerant architecture of yaw moment management with steer-by-wire, active braking and driving-torque distribution integrated control. SAE Automotive Electronics Series, 2008–01-01 (2008)
He, L., Zong, C., Wang, C.: A steering-by-wire fault-tolerance control strategy based on multi-dimension gauss hidden Markov model. In: International Conference on Intelligent Control and Information Processing, pp. 227–230. Dalian, China (2010)
Heiner, G., Thurner, T.: Time-triggered architecture for safety-related distributed real-time systems in transportation systems. In: Symposium, Twenty-Eighth Annual International symposium on Fault-Tolerant Computing, pp. 402–432. IEEE Computer Society, Washington, DC (1998)
Herath, I., Roberts, C., Arvanitis, T.N., Bold, A.: Satisfying design constraints for automotive safety-critical systems. SAE Automotive Electronics Series, 2007–01-14 (2007)
Isermann, R., Beck, M.: Modellbasierte Methoden zur Erhöhung der Verfügbarkeit und Sicherheit von Fahrwerkkomponenten. AUTOREG 2011, pp. 679–690 (2011)
Isermann, R., Schwarz, R., Stölzl, S.: Fault-tolerant drive-by-wire systems. IEEE Control Syst. Mag. **22**(5), 64–81 (2002)
Johannessen, P.:SIRIUS, : Technical Report 01. Department of Computer Engineering Chalmers University of Technology. Göteborg, Sweden (2001)
Johannessen, P., Ahlström, K., Torin, J.: Conceptual design of distributed by-wire systems. SAE Automotive Electronics Series, 2002–01-02 (2002)
Johannessen, P., Törner, F., Torin, J.: Actuator based hazard analysis for safety critical systems. In: Computer Safely Reliability Security, vol. 3219, pp. 130–141 (2004)
Johannessen, P., Törner, F., Torin, J.: Experiences from model based development of drive-by-wire control systems. In: Kleinjohann, B., Gao, G.R., Kopetz, H., Kleinjohann, L., Rettberg, A. (eds.)

Design Methods and Applications for Distributed Embedded Systems, pp. 103–112. Springer, Boston (2004)
Kelling, N.A., Heck, W.: The BRAKE project—centralized versus distributed redundancy for brake-by-wire systems. SAE Automotive Electronics Series, 2002–01-02 (2002)
Kim, M.H., Lee, S., Lee, K.C.: Kalman predictive redundancy system for fault tolerance of safety-critical systems. IEEE Trans. Industr. Inf. **6**(1), 46–53 (2010)
Koehn, P., Eckrich, M., Smakman, H., Schaffert, A.: Integrated chassis management : introduction into BMW's approach to ICM. SAE Technical Paper Series 1(1219), (2006)
Köhler, R., Broy, J.: Markov-Ketten und Autokorrelation in der Sprach- und Textanalyse. In: Köhler, R., Broy, J. (ed.) Glottometrika 5 Bochum (1983)
Legler, H., Gehrke, B., Krawczyk, O., Schasse, U., Rammer, C., Leheyda, N., Sofka, W.: Die Bedeutung der Automobilindustrie für die deutsche Volkswirtschaft im europäischen Kontext (2009)
Lieberam, J.: Entwicklung eines Softwaresystems zur Zustandserfassung und -regelung im Kraftfahrzeug. Diplomarbeit, TU Braunschweig (2011)
Löw, P., Pabst, R., Petry, E.: Funktionale Sicherheit in der Praxis, 1st edn. Heidelberg: dpunkt.verlag GmbH (2010)
Mahmud, N., Papadopoulos, Y., Walker, M.: A translation of state machines to temporal fault trees. In: 2010 International Conference on Dependable Systems and Networks Workshops, pp. 45–51. Chicago, USA (2010)
Maier, M.W., Rechtin, E.: The Art of Systems Architecting, 3rd edn. CRC Press Taylor & Francis Group, Boca Raton (2009)
Masak, D.: Der Architekturreview. Springer, Berlin (2010)
Maurer, M.: Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen. Dissertation, Universität der Bundeswehr München, Düsseldorf (2000)
Maurer, M.: Automotive systems engineering—a personal perspective. In: Maurer, M., Winner, H. (eds.) Automotive Systems Engineering. Springer, Heidelberg (2013)
McLaughlin, S.B.: Analytic assessment of collision avoidance systems and driver dynamic performance in rear-end crashes and near-crashes. Ph.D. thesis, Virginia Polytechnic Institute and State University, USA (2007)
Mehmood, A., Easa, S.M.: Modeling reaction time in car-following behaviour based on human factors. Int. J. Appl. Sci. Eng. Techn. **5**(14), 93–101 (2009)
Miller, P.: A Prototype distributed architecture for safety critical automotive systems. SAE Automotive Electronics Series, 2007–01-16 (2007)
Mishra, P.K., Naik, S.M.: Distributed control system development for flexray-based systems. SAE Automotive Electronics Series, 2005–01-12 (2005)
Mitzlaff, M., Lang, M., Kapitza, R., Schröder-Preikschat, W.: A membership service for a distributed, embedded system based on a time-triggered flexray network. In: 2010 European Dependable Computing Conference, pp. 155–162. Valencia, Spain (2010)
Motruk, B., Diemer, J., Ernst, R., Buchty, R., Berekovic, M.: IDAMC : A many-core platform with run-time monitoring for mixed-criticality. In: 14th International High Assurance Systems Engineering Symposium Omaha, USA (2012)
Muenchhof, M., Beck, M., Isermann, R.: Fault-tolerant actuators and drives—structures, fault detection principles and applications. Ann. Rev. Control **33**(2), 136–148 (2009)
Müller, K., Steinbach, T., Korf, F., Schmidt, T.C.: A real-time ethernet prototype platform for automotive applications. In: 2011 IEEE International Conference on Consumer Electronics - Berlin (ICCE-Berlin), pp. 221–225. Berlin (2011)
Neudörfer, A.: Konstruieren sicherheitsgerechter Produkte. Springer, Heidelberg (2011)
Palin, R., Ward, D., Habli, I., Rivett, R.: ISO 26262 safety cases: compliance and assurance. In: 6th IET International Conference on System Safety, pp. 1–6. Birmingham, UK (2011)
Papadopoulos, Y., McDermid, J., Sasse, R., Heiner, G.: Analysis and synthesis of the behaviour of complex programmable electronic systems in conditions of failure. Reliab. Eng. Syst. Saf. **71**(3), 229–247 (2001)

Park, T.-j., Han, C.-s., Lee, S.-h.: Development of the electronic control unit for the rack-actuating steer-by-wire using the hardware-in-the-loop simulation system. Mechatronics **15**(8), 899–918 (2005)

Pfeffer, P., Harrer, M.: Lenkungshandbuch. Wiesbaden: Vieweg+Teubner Verlag | Springer Fachmedien Wiesbaden GmbH (2011)

Philipps, J.: Kontrolle ist gut, Misstrauen ist besser: Funktionale Sicherheit für integrierte Softwarefunktionen. In: Schäfer, H. (ed.) Trends in der elektrischen Antriebstechnologie für Hybrid- und Elektrofahrzeuge, pp. 129–140. Expert Verlag, Renningen (2012)

Pimentel, J.: Safety-reliability of distributed embedded system fault tolerant units. In: IECON'03. 29th Annual Conference of the IEEE Industrial Electronics Society, pp. 945–950. Roanoke, USA (2003)

Piyabongkarn, D., Lew, J.Y., Rajamani, R., Grogg, J.A., Yuan, Q.: On the use of torque-biasing systems for electronic stability control: limitations and possibilities. IEEE Trans. Control Syst. Technol. **15**(3), 581–589 (2007)

Pruckner, A., Stroph, R., Pfeffer, P.: Drive-By-Wire. In: Eskandarian, A. (ed.) Handbook of Intelligent Vehicles, pp. 235–282. Springer, London (2012)

Rausand, M., Hoyland, A.: System reliability theory—models, statistical methods and applications. Wiley, Hoboken (2009)

Rehage, D., Carl, U.B., Vahl, A.: Redundancy management of fault tolerant aircraft system architectures—reliability synthesis and analysis of degraded system states. Aerosp. Sci. Technol. **9**(4), 337–347 (2005)

Reichel, R., Armbruster, M.: X-by-Wire Plattform—Konzept und Auslegung. at—Automatisierungstechnik **59**(9), 583–596 (2011)

Reif, K.: Automobilelektronik, Eine Einführung für Ingenieure, 3rd edn. Wiesbaden: Vieweg+Teubner GWV Fachverlage GmbH (2009)

Reinold, P., Nachtigal, V., Trächtler, A.: An advanced electric vehicle for development and test of new vehicle-dynamics control strategies. In: 6th IFAC Symposium Advances in Automotive Control. Munich (2010)

Richter, D., Köhnen, A.: Sicherheitsziele für zukünftige Elektro-Fahrzeuge: Sicherheitsarchitektur für den elektrischen Antrieb basierend auf den Anforderungen der ISO 26262. In: Schäfer, H. (ed.) Trends in der elektrischen Antriebstechnologie für Hybrid- und Elektrofahrzeuge, pp. 95–100. Expert Verlag, Renningen (2012)

Rieth, P.E.: Das mechatronische Fahrwerk der Zukunft. In H. Winner, S. Hakuli, & G. Wolf (eds., 2012), Handbuch Fahrerassistenzsysteme, pp. 626–631. Vieweg+Teubner Verlag | Springer Fachmedien Wiesbaden GmbH, Wiesbaden (2012)

Rohe, M.: Entwicklung der Gesamtfahrzeugstrategie eines E-Fahrzeugprototyps mit Torque Vectoring. In: Schäfer, H. (ed.), Trends in der elektrischen Antriebstechnologie für Hybrid- und Elektrofahrzeuge, pp. 101–111. Expert Verlag, Renningen (2012)

Sakurai, K., Matsubara, M., Hoshino, M.: Membership middleware for dependable and cost-effective X-by-wire systems. SAE Automotive Electronics Series, 2008–01-04, 1–9 (2008)

Sangiovanni-Vincentelli, A.: Quo Vadis, SLD? reasoning about the trends and challenges of system Level design. Proc. IEEE **95**(3), 467–506 (2007)

Schäuffele, J., Zurawka, T.: Automotive Software Engineering—Grundlagen, Prozesse, Methoden und Werkzeuge. Friedr. Vieweg & Sohn Verlag/GWV Fachverlage GmbH, Wiesbaden (2004)

Schroer, R.: Flight control goes digital [Part Two, NASA at 50]. IEEE Aerosp. Electron. Syst. Mag. Part Two **23**(10), 23–28 (2008)

Schwall, M.L., Gerdes, J.C.: A probabilistic approach to residual processing for vehicle fault detection. In: Proceedings of the 2002 American Control Conference, vol. 3, pp. 2552–2557 (2002)

Siedersberger, K.-H.: Komponenten zur automatischen Fahrzeugführung in sehenden (semi-), autonomen Fahrzeugen. Dissertation, Universität der Bundeswehr München (2003)

Sieglin, E.: Beitrag zur Energieversorgung eines innovativen Drive-by-wire-Fahrzeugkonzepts. Dissertation, Technische Universität Dresden, Renningen (2009)

Sinha, P.: Architectural design and reliability analysis of a fail-operational brake-by-wire system from ISO 26262 perspectives. Reliab. Eng. Syst. Saf. **96**(10), 1349–1359 (2011)
Smakman, H., Köhn, I.P., Vieler, D.H.: Integrated Chassis Management—ein Ansatz zur Strukturierung der Fahrdynamikregelsysteme. In: 17. Aachener Kolloquium Fahrzeug- und Motorentechnik, pp. 1–13 (2008)
Starke, G.: Effektive Software-Architekturen. Carl Hanser Verlag, Munich (2008)
Sundar, M., Plunkett, D.: Brake-by-wire, motivation and engineering—GM sequel. SAE Automotive Electronics Series, 2006–01-31 (2006)
Tkachev, O.A.: Application of Markov chains for the reliability analysis of systems with a complex structure. Cybern. Syst. Anal. **19**(5), 96–101 (1983)
Töpler, S.: Entwicklung eines Abgleichreglers für die Fahrzeug Längs- und Querdynamik. Diplomarbeit, TU Braunschweig (2010)
Touloupis, E., Flint, J.A., Chouliaras, V.A., Ward, D.D.: A fault-tolerant processor core architecture for safety-critical automotive applications. SAE Automotive Electronics Series, 2005–01-03 (2005)
Trächtler, A., Niewels, F. Integrierte Querdynamikregelung mit ESP, AFS und aktiven Fahrwerksystemen. In: Isermann, R. (ed.) Fahrdynamik-Regelung, pp. 237–251. Friedr. Vieweg & Sohn Verlag | GWV Fachverlage GmbH, Wiesbaden (2006)
Tucci-Piergiovanni, S., Mraidha, C., Wozniak, E., Lanusse, A., Gerard, S.: A UML model-based approach for replication assessment of AUTOSAR safety-critical applications. In: IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications, pp. 1176–1187. Changsha, China (2011)
Verma, A.K., Ajit, S.: Reliability and Safety Engineering. Springer, London (2010)
von Vietinghoff, A.: Nichtlineare Regelung von Kraftfahrzeugen in querdynamisch kritischen Fahrsituationen. Dissertation, Universität Karlsruhe (2008)
Walker, M., Papadopoulos, Y.: Qualitative temporal analysis: towards a full implementation of the fault tree handbook. Control Eng. Pract. **17**(10), 1115–1125 (2009)
Waraus, D.: Steer-by-wire system based on flexray protocol. In: Applied Electronics, pp. 269–272. Czech Republic, Pilsen (2009)
Wilwert, C., Navet, N., Song, Y.Q., Simonot-Lion, F.: Design of automotive X-by-wire systems. In: Zurawski, R. (ed.) The Industrial Communication Technology Handbook, pp. (29–1)–(29–34). CRC Press, Boca Raton (2005)
X-by-Wire Project (1998). Brite-EuRam 111 Program. X-By-Wire—safety related fault tolerant systems in vehicles, final report
Zhen, B., Altemare, C., Anwar, S.: Fault tolerant steer-by-wire road wheel control system. In: Proceedings of the 2005 American Control Conference, pp. 1619–1624. Portland, USA (2005)
Zuo, G., Kumamoto, H., Nishihara, O., Hayama, R., Nakano, S.: Quantitative reliability analysis of different design alternatives for steer-by-wire system. Reliab. Eng. Syst. Saf. **89**(3), 241–247 (2005)