Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Stefan Buschmann

Analyse redundanter Kommunikationsarchitekturen im Automobil

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Stefan Buschmann

Analyse redundanter Kommunikationsarchitekturen im Automobil

Masterarbeit eingereicht im Rahmen der Masterprüfung

im Studiengang Master of Science Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Franz Korf
Zweitgutachter: Prof. Dr. Bettina Buth

Eingereicht am: 21. März 2017

Stefan Buschmann

Thema der Arbeit
Analyse redundanter Kommunikationsarchitekturen im Automobil

Stichworte
Redundanz, Fahrzeugnetze, Fehlerbaumanalyse, ISO 26262, Echtzeit-Ethernet, Bussysteme, Analyse

Kurzzusammenfassung
Steigende Anforderungen an die Kommunikationssysteme im Automobil erfordern die Entwicklung und Einführung neuer Kommunikationstechnologien, da vorhandene Bussysteme diese nicht mehr erfüllen können. Um sicherheitskritische Funktionen realisieren zu können bedarf es einer sicheren und zuverlässigen Kommunikation. Mögliche Kandidaten sind um Echtzeitfähigkeit weiterentwickelte Ethernet-Technologien wie Time-Triggered Ethernet oder Audio Video Bridging. Um diese Netzwerke weiter abzusichern bietet sich der Einsatz von Redundanz an. Damit ist es möglich das System noch robuster gegen Fehler und Störungen zu machen. Um diese Redundanzkonzepte zu analysieren bietet sich die Fehlerbaumanalyse an.
Stefan Buschmann

Title of the paper
Analysis of redundant automotive communication architectures

Keywords
redundancy, in-car networks, fault tree analysis, ISO 26262, Real-Time-Ethernet, bussystems, analysis

Abstract
Increasing demands on communication systems in the automotive industry require the development and introduction of new communication technologies, as existing bus systems can no longer meet these demands. In order to realize safety-critical functions, a reliable communication is required. Potential candidates are real-time Ethernet technologies like Time-Triggered-Ethernet or Audio Video Bridging. In order to further secure these networks, the use of redundancy is an option. This makes it possible to secure the system even more against faults. In order to analyze these redundancy concepts, the fault tree analysis can be used.

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

1 Einleitung & Motivation

Fahrassistenzsysteme und Entertainmentanwendungen finden Einzug in immer mehr Fahrzeuge. Es müssen Audio- und Videodaten, Messwerte von RADAR (Radio Detection And Ranging) und LIDAR (Light detection and ranging) Sensoren sowie Kameradaten übertragen werden. Dies stellt enorme Anforderungen an die Kommunikationssysteme. Bussysteme, die derzeit im Auto zum Einsatz kommen, können diese nicht erfüllen. Entweder steht nicht genügend Bandbreite zur Verfügung oder die rechtzeitige Übermittlung kritischer Daten für sicherheitskritische Anwendungen kann nicht garantiert werden.

Hinzu kommt, dass die unterschiedlichen Technologien, die in einem Auto eingesetzt werden (z. B. CAN, LIN, MOST und FlexRay), ein heterogenes, schwer überschaubares System darstellen (vgl. Zimmermann und Schidgall, 2014, S. 9). Ein möglicher Ansatzpunkt wäre der Einsatz von, auf Ethernet basierender, Technologien. Da das Standard-Ethernet für echtzeitfähigen Datenverkehr nicht geeignet ist, gibt es einige Weiterentwicklungen, wie zum Beispiel Time-Triggered Ethernet oder der IEEE Standard Audio-/Video-Bridging, welche genau das ermöglichen.

Um als Kommunikationstechnologie für sicherheitskritische Anwendungen, wie beispielsweise Steer-By-Wire, im Auto eingesetzt werden zu können, müssen alle verwendeten Komponenten gewisse Voraussetzungen erfüllen. Die ISO 26262 stellt diese. Sie befasst sich mit sicherheitsrelevanten elektronischen Systemen in PWK. Jede Funktion, die im Rahmen dieser Norm implmentiert wird, kann in eine von fünf Klassen eingeteilt werden. Je nachdem, wie häufig die Funktion zum Einsatz kommt, wie kontrollierbar sie im Fehlerfall ist und welche Gefahren dadurch entstehen können, erfolgt die Einteilung in eines der *Automotive Safety Integrity Level* (ASIL).

Erfüllt eine Kommunikationsarchitektur diese Anforderungen des ASIL nicht, kann dies eventuell durch redundante Mittel kompensiert werden. Redundanz ist in der Flugzeugindustrie weit verbreitet. Für die Realisierung von Flugkontrollsystemen wird mindestens eine dreifache Redundanz eingesetzt (vgl. Sghairi u. a., 2008). Zudem befasst sich die Norm DIN EN 62439 mit hochverfügbaren Kommunikationsnetzwerken in der Industrie. In dieser werden unter

anderem einige auf Ethernet basierende Protokolle definiert, die das redundante Übertragen von Nachrichten realisieren.

Als geeignetes Analyseverfahren für die Netzwerke, das auch in der Automobilindustrie eingesetzt wird (vgl. Schilling, 2009, S. 1), bietet sich die Fehlerbaumanalyse an. Mit logischen Gates und Ereignissen, die mögliche Fehler in einem Netzwerk darstellen können, lassen sich Bäume generieren, mit denen die Ausfallwahrscheinlichkeiten der Systeme berechnet werden können. Da sich ein redundantes System im Laufe der Zeit durch den Ausfall redundanter Komponenten verändert, muss dies auch in den Fehlerbäumen abgebildet werden können. Da statische Fehlerbäume dazu nicht in der Lage sind, kommen dynamische Fehlerbäume zum Einsatz (vgl. Dugan u. a., 1992). Mit ihnen ist es möglich Abhängigkeiten und Redundanzen in den Bäumen darzustellen.

Der weitere Aufbau dieser Arbeit gestaltet sich wie folgt: In Kapitel 2 werden im Hinblick auf diese Arbeit verwandte Arbeiten vorgestellt. Kapitel 3 befasst sich mit der Kommunikation im Auto und gibt eine Übersicht über aktuelle Bussysteme. Weiterhin wird das Thema Ethernet im Automobil betrachtet und relevante Weiterentwicklungen vom Standard-Ethernet erläutert. Zum Schluss des Kapitels wird auf die Anforderungen der Autoindustrie und auf die ISO 26262 eingegangen. Das Kapitel 4 behandelt das Thema Redundanz. Es geht um die Klassifizierung von Anlagen und um die verschiedenen Redundanzkonzepte. Im nachfolgenden Kapitel 5 folgt dann die Erläuterung der statischen und dynamischen Fehlerbaumanalysen. Wie unterschiedliche Fehlerszenarien einer Kommunikationsarchitektur in Fehlerbäumen dargestellt werden können, wird in Kapitel 6 dargestellt. In Kapitel 7 werden Redundanzkonzepte zu einem bereits vorhandenen Netzwerk entwickelt und in Fehlerbäumen abgebildet, um anschließend in Kapitel 8 analysiert und bewertet zu werden. Das letzte Kapitel 9 fasst schließlich diese Arbeit zusammen.

2 Related work

Redundante Lösungen zur Steigerung der Zuverlässigkeit werden schon seit geraumer Zeit in unterschiedlichen Bereichen verwendet. So spielt ihr Einsatz in der Flugzeugbranche schon lange eine wichtige Rolle aber auch bei Industrieanlagen sind sie ein wichtiger Bestandteil. Die Fehlerbaumanalyse ist schon seit vielen Jahren ein wichtiges Verfahren um Anlagen auf ihre Ausfallwahrscheinlichkeiten zu untersuchen. Sie wurden stetig weiterentwickelt und finden weiterhin Verwendung.

2.1 Redundanz im Flugzeug

Schon vor einigen Jahrzehnten, befassten sich die Wissenschaftler mit dem Einsatz redundanter Elemente und wie diese sich auf die Fehlerwahrscheinlichkeit auswirken (vgl. Nalos und Schulz, 1965). Die Zuverlässigkeit unterschiedlicher Avioniksysteme wurde dabei auch mit den hierdurch entstehenden Kosten gegenübergestellt. Aktuelle Arbeiten wie zum Beispiel (Gohil u. a., 2011) befassen sich mit unterschiedlichen Redundanzkonzepten, ihren Vor- und Nachteilen und wie sie sich für den Einsatz im Flugzeug eignen.

Die Realisierung der Flugkontrollsysteme (engl. Flight Control System (FCS)) sind auch heute noch Thema in diversen wissenschaftlichen Arbeiten. Seit dem Airbus A310 haben Fly-By-Wire Systeme auch in der zivilen Luftfahrt Einzug gefunden. Die Arbeit (Sghairi u. a., 2008) behandelt unter anderem die unterschiedlichen Konzepte für die FCS von Airbus und Boeing. Beide Unternehmen setzen ein hohes Maß an Redundanz ein, um die Systeme so sicher wie möglich zu gestalten.

Airbus Design Das FCS eines Airbus besteht aus mehreren primären und sekundären Flight Control Computern (FCC). Jeder FCC besteht aus zwei voneinander unabhängigen Einheiten. Jede Einheit unterscheidet sich in Hard- und Software, wobei für die Programmierung unterschiedliche Entwicklerteams und unterschiedliche Programmiersprachen eingesetzt wurden.

Sollte sich das Ergebnis der Einheiten in einem FCC unterscheiden, wird ein anderer FCC genutzt. Zusätzlich kommunizieren die unterschiedlichen FCCs untereinander um Fehler erkennen zu können. Ebenso ist jeder Computer mit unterschiedlichen Sensoren und Aktoren

verbunden, um den Ausfall dieser Komponenten kompensieren zu können. Daraus ergibt sich ein stark verzweigtes Netzwerk, wie in Abbildung 2.1 zu sehen ist.

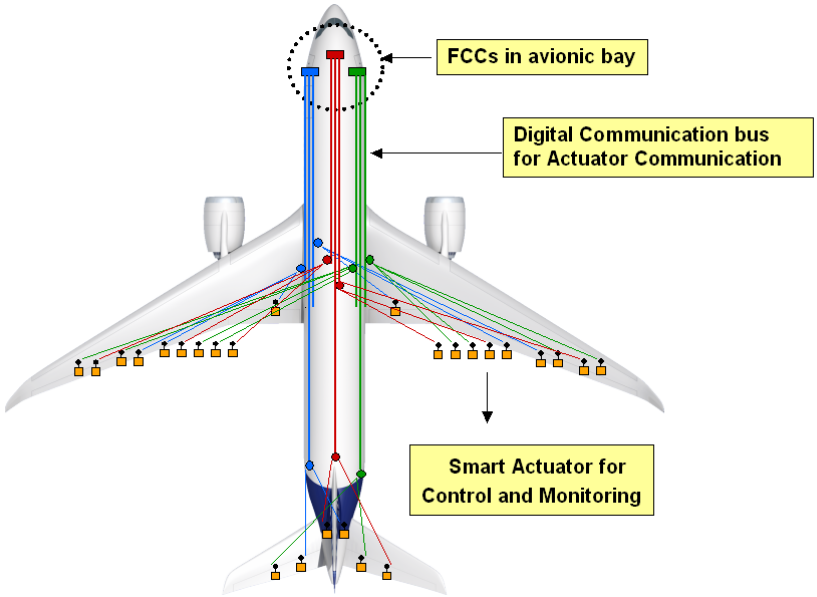


Abbildung 2.1: Beispiel einer Airbus FCS Architektur (Quelle: Sghairi u. a. (2008))

Boeing Design Das Primary Flight Control System (PFCS) von Boeing (siehe Abbildung 2.2) setzt sich aus drei identischen Primary Flight Computern (PFC) und vier weiteren Computern, den Actuator Control Electronics (ACE), zusammen. Die PFCs leiten die Befehle an die ACEs weiter, die dann wiederum die Ansteuerung der Aktuatoren übernehmen. Zusätzlich besteht jeder PFC aus drei verschiedenen Einheiten und kann außerdem mit jedem ACE kommunizieren. Auch dieses Netzwerk zeichnet sich durch ein starkes Maß an Redundanz aus.

2.2 Redundanz in der Industrie

Auch für industrielle Anlagen, wie beispielsweise Kraftwerke, Pumpensysteme oder automatisierte Anlagen sind redundante Lösungen unverzichtbar. Je nach Klassifikation der jeweiligen Anlage können unterschiedliche Anforderungen an das System und dessen Redundanz gestellt werden (vgl. Kirrmann und Dzung, 2006). Auch in dieser Arbeit werden verschiedene Konzepte vorgestellt wie unter anderem auch die Kommunikation redundant gestaltet werden kann. Speziell werden auch die Umschaltzeiten berücksichtigt, die ein Netzwerk braucht um den Fehler

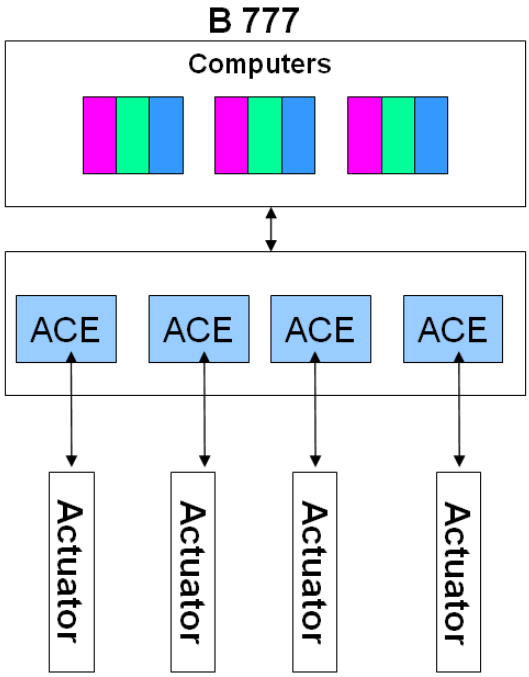


Abbildung 2.2: Übersicht über Boeing FCS Architektur (Quelle: Sghairi u. a. (2008))

zu erkennen und wieder in einen funktionstüchtigen Zustand zu kommen, um Anforderungen der kritischen Systeme erfüllen zu können.

Goraj und Harada (2012) stellen Möglichkeiten zur Umsetzung von Technologien vor, mit denen redundante Kommunikation verwirklicht werden kann. Hier werden unterschiedliche Protokolle für den Einsatz in Schaltanlagen untersucht und miteinander verglichen. Die Kommunikation in einem Pumpenkontrollsystem wird in Firoozshahi (2010) beschrieben. Durch die Kommunikation im Ring sowie redundanter Hardware wird dabei die Sicherheit des Systems erhöht.

Bei redundanten Elementen kann es zu einem Entscheidungsproblem kommen, wenn es mehrere aktive Komponenten für die gleiche Aufgabe gibt, da alle ein Ergebnis liefern (vgl. Karimi u. a., 2010). Läuft alles einwandfrei, ist dies bei allen identisch. Aber in einem Fehlerfall liegen eventuell mehrere unterschiedliche Ergebnisse vor und es muss entschieden werden, welches genutzt werden soll. Die Autoren stellen hierzu einen Algorithmus vor, mit dem ein Voter diese Entscheidung treffen kann.

2.3 Fehlerbaumanalyse

Die Fehlerbaumanalyse wird seit vielen Jahren in der Industrie eingesetzt um Systeme zu analysieren und deren Ausfallwahrscheinlichkeiten zu bestimmen. Die NASA brachte bereits 1981 die erste Version des *Fault Tree Handbook* heraus. 21 Jahre später wurde eine neue Auflage veröffentlicht und wird immer noch als eine der wichtigsten Techniken in der Bewertung von Zuverlässigkeiten bezeichnet (vgl. Vesely u. a., 2002). Wissenschaftliche Arbeiten auf diesem Gebiet befassen sich oft mit effektiveren Berechnungsmethoden der Bäume (vgl. Thums, 2004; Rao u. a., 2009; Merle u. a., 2009, 2010) aber auch mit der Anpassung und Optimierung für dedizierte Anwendungsgebiete (vgl. Esteve u. a., 2012; Yuyan u. a., 2015; Nassaj und Barabady, 2016). ,

3 Kommunikation im Automobil

Dieses Kapitel befasst sich mit unterschiedlichen Kommunikationstechnologien im Auto und mit Anforderungen der Automobilindustrie. Im ersten Abschnitt wird ein Überblick über aktuelle Bussysteme und deren Eigenschaften gegeben. Aus diesen unterschiedlichen Technologien entsteht innerhalb des Fahrzeugs ein heterogenes Netzwerk, da für die unterschiedlichen Anforderungen der Anwendungen unterschiedliche Lösungen zum Einsatz kommen. Dies führt dazu, dass ein komplexes, schwer zu beherrschendes Netzwerk entsteht (vgl. Zimmermann und Schidgall, 2014, S. 9). Aufgrund dessen wird anschließend ein Blick auf mögliche zukünftige Ethernet-Lösungen als Kommunikationstechnologie im Auto geworfen. Der letzte Abschnitt des Kapitels befasst sich mit Anforderungen, die sowohl durch die Automobilindustrie als auch durch die Norm ISO 26262 gestellt werden.

3.1 Aktuelle Bussysteme im Auto

In diesem Abschnitt Bussysteme mit ihren Eigenschaften und Einsatzgebieten vorgestellt, die häufig ein Fahrzeugen zum Einsatz kommen.

3.1.1 Controller Area Network - CAN

CAN ist das am häufigsten im Auto eingesetzte Kommunikationssystem. Es wurde bereits 1983 von Bosch entwickelt und kam 1991 das erste Mal in einem Fahrzeug zum Einsatz. Mit einer Datenrate von 1 MBit/s und 0 bis 8 Byte Daten pro Frame eignet es sich nicht für große Datenmengen, wie sie beispielsweise bei Multimediaanwendungen anfallen.

In einem CAN-Netz funktioniert die komplette Kommunikation ereignisgesteuert. Das heißt sobald die Leitung nicht belegt ist und ein Knoten senden möchte, kann dieser das auch jederzeit tun. Sollten zufällig zwei Teilnehmer gleichzeitig anfangen zu senden, darf der Knoten mit der höher priorisierten Nachricht den Bus für sich beanspruchen. Erkannt wird das gleichzeitige Senden durch die bitweise Arbitrierung. Mithilfe des Identifiers (ID), der gleich am Anfang des Frames übertragen wird, kann die Priorität der Nachricht festgelegt werden. Je niedriger die ID, desto höher ist die Priorität. Starten also zwei Knoten gleichzeitig die Übertragung senden sie

nach und nach die Bits der ID an den Bus. Da ein 0-Signal dominant gegenüber einem 1-Signal ist, erkennt der Sender der niedrig priorisierten Nachricht, dass gleichzeitig noch ein weiterer Knoten sendet und beendet seine Übertragung. Sobald der Bus wieder frei ist, wird eine neue Übertragung gestartet.

3.1.2 Local Interconnect Network - LIN

Der LIN-Bus wurde als kostengünstigere Alternative zu CAN entwickelt. Mit einer maximalen Bandbreite von 20 KBit/s ist es für Sensor-Aktor-Anwendungen gedacht, bei denen nur wenig Daten anfallen. Genau wie bei CAN können pro Nachricht maximal 8 Byte Daten verschickt werden.

Ein LIN-Netz besteht aus bis zu 16 Knoten (Slaves). Hinzu kommt ein Master, der die komplette Kommunikation auf dem Bus organisiert, indem er periodisch zu festen Zeitpunkten den Header einer Nachricht sendet. Der für diesen Header zuständige Slave sendet dann die Datenbytes. Der Master fungiert im Normalfall auch gleichzeitig als Gateway zu einem anderen Bussystem.

3.1.3 Media Oriented Systems Transport - MOST

Da er als Kommunikationssystem für Infotainment-Anwendungen konzipiert wurde, bietet der MOST-Bus eine hohe Datenrate von bis zu 150 MBit/s. Meist erfolgt die Kommunikation über Lichtwellenleiter in einer Ring-Topologie. Hier synchronisieren sich die maximal 64 Teilnehmer auf einen Master, der zusätzlich die Generierung der Nachrichten übernimmt.

Die Steuergeräte haben die Möglichkeit die bereits generierten Nachrichten mit ihren Daten zu füllen. Hierzu wird das Datenfeld in einen flexiblen synchronen und einen flexiblen asynchronen Bereich aufgeteilt. Im synchronen Teil können die Anwendungen exklusiven Zugriff auf den benötigten Bereich anfordern und so gewährleisten, dass beispielsweise für die Übertragung einer Audio-CD genügend Bandbreite zur Verfügung steht. In dem Datenfeld bleibt nur so viel Platz für asynchrone Nachrichten, wie der synchrone Teil zulässt. Wird das komplette Datenfeld von Anwendungen für synchrone Daten reserviert, bleibt kein Platz mehr für asynchrone Daten. Steht aber noch Bandbreite zur Verfügung, können asynchrone Daten, gegebenenfalls auch auf mehrere Nachrichten aufgeteilt, verschickt werden.

3.1.4 FlexRay

Die Entwicklung von FlexRay begann im Jahr 2000 und wurde 2010 als Standard für Automobile veröffentlicht. Ein starker Fokus lag darauf ein System zu entwickeln, das hohe Anforderungen

an die Sicherheit erfüllt, um unter anderem für X-By-Wire-Anwendungen genutzt werden zu können. Pro Kanal können bis zu 10 MBit/s übertragen werden und pro Nachricht steht ein maximal 254 Byte großes Datenfeld zur Verfügung. Den Knoten stehen zwei Kanäle gleichzeitig zur Verfügung. Diese beiden Kanäle können entweder genutzt werden um mehr Daten gleichzeitig übertragen zu können (unterschiedliche Daten auf beiden Kanälen) oder um die Daten redundant zu übermitteln (mehr Sicherheit).

Die Kommunikation erfolgt in fest definierten Kommunikationszyklen. Am Ende von jeden Zyklus wird die globale Zeit unter allen Teilnehmern synchronisiert. Das ist essentiell für das Time Division Multiple Access (TDMA) Verfahren mit dem die Kommunikation organisiert wird. Der Zyklus ist in zwei Teile (statisches und dynamisches Segment) unterteilt, in denen die Teilnehmer Daten senden können. Das statische Segment ist in feste Zeitschlitze (Slots) aufgeteilt, in denen eine Botschaft komplett übertragen werden kann. Der Zugriff wird schon vor dem Systemstart fest eingeteilt, sodass nur ein Knoten in genau diesem Slot senden darf.

Das dynamische Segment ist ebenfalls in Zeitschlitze unterteilt. Diese sind jedoch kleiner als die im statischen Segment und es ist nicht möglich innerhalb der kurzen Zeit eine komplette Nachricht zu übermitteln. Auch in diesem Segment werden die jeweiligen Zeitschlitze den Knoten vor Systemstart zugeteilt. Sendet ein Knoten nun in seinem Slot eine Nachricht, verschieben sich die Zeitpunkte für die Zeitschlitze der anderen Knoten entsprechend nach hinten. Dies kann zur Folge haben, dass Daten im dynamischen Segment nicht innerhalb eines Zyklus verschickt werden können und eine erneute Übertragung im nächsten Durchlauf versucht werden muss.

3.2 Ethernet im Automobil

Durch seine weite Verbreitung und die hohen Stückzahlen stellt Ethernet eine mögliche Alternative zu vorhandenen Kommunikationstechnologien im Auto dar. Mit derzeit bis zu 100 GBit/s kann es die steigenden Anforderungen an die Bandbreite kann es erfüllen, da es sich in dieser Hinsicht stetig weiterentwickelt. Tatsächlich wird Ethernet bereits im Auto beispielsweise als Diagnoseschnittstelle eingesetzt. Standard-Ethernet-Komponenten eigenen sich aber im Hinblick auf die EMV-, Temperatur-, Schüttel- und Feuchtebelastung nicht für den Einsatz in einem Auto. Diesem Problem hat sich die Firma Broadcom angenommen und mit *BroadR-Reach* PHY-Bausteine entwickelt, die über eine Zwei-Draht-Leitung im Voll-Duplex-Betrieb Daten übertragen können und gleichzeitig die Anforderungen durch die Automobilindustrie erfüllen (vgl. Zimmermann und Schidgall, 2014, S.141 - 142).

Anforderungen an die Latenz für sicherheitskritische Anwendungen lassen ich mit dem Standard-Ethernet allerdings nicht zuverlässig realisieren, da sich nicht bestimmen lässt wann eine Nachricht beim Empfänger ankommt. In dem Ethernet-Standard IEE802.3 (IEEE 802.3, 2015) ist nicht geregelt in welcher Reihenfolge Nachrichten in einem Switch weitergeleitet werden. Es könnte also passieren, dass eine Nachricht von anderen Nachrichten, die den gleichen Weg nehmen, stark verzögert werden.

Es existieren Weiterentwicklungen vom Standard-Ethernet, die darauf abzielen den Datenverkehr besser zu lenken und Echtzeitanforderungen zu erfüllen. Einige dieser Konzepte werden in diesem Abschnitt erläutert:

IEEE 802.1Q (siehe IEEE 802.1 TSN Task Group) erweitert Ethernet um virtuelle Netzwerke (VLAN) und Prioritäten. Der Ethernet-Header wird hierzu um 4 Byte erweitert. 12 Bit hiervon stehen für die VLAN Identifier (VID) zur Verfügung und ermöglichen somit, dass theoretisch 4096 unterschiedliche VLANs in einem Netzwerk existieren können. Da die IDs „0" und „4095" reserviert sind, sind es im Endeffekt aber nur 4094. Geräte, die einem VLAN zugeordnet sind, können auch nur mit anderen Geräten im selben VLAN kommunizieren. So ist es möglich physisch zusammenhängende Netzwerke logisch voneinander zu trennen.

Als weiteres Feature ermöglicht der Standard die Priorisierung von Nachrichten. Hierzu werden 3 Bit des Headers genutzt, was dazu führt, dass acht unterschiedliche Prioritäten konfiguriert werden können. Für jede Priorität gibt es in den Switches und Endknoten separate Buffer, die es ermöglichen hoch priorisierte Nachrichten bevorzugt verarbeiten zu können. So kann ein gewisses Sendeverhalten aber nach wie vor keine genauen Zeiten vorausgesagt werden, da Nachrichten von anderen Nachrichten mit gleicher Priorität (z. B. während eines Bursts) verzögert werden können.

Audio-/Video-Bridging (AVB) wird in dem Standard IEEE 802.1BA (2011) definiert und ermöglicht das Versenden von Daten mit einer garantierten maximalen Latenz. AVB ist darauf ausgelegt Datenströme per Multicast an mehrere Empfänger zu versenden. Die benötigte Bandbreite muss vorher mit Hilfe des Stream Reservation Protocols (SRP) (siehe IEEE 802.1Qat, 2006) reserviert werden. Die Datenströme können in zwei unterschiedlichen Paketklassen versendet werden:

- Klasse A garantiert eine maximale Latenz von 2 ms über bis zu 7 Hops
- Klasse B garantiert eine maximale Latenz von 50 ms über bis zu 7 Hops

Das tatsächliche Versenden erfolgt mit dem Credit Based Shaper Verfahren (CBS) (siehe IEEE 802.1Qav, 2009)). Das sieht vor, dass an jedem Port des Senders und der Bridges für jede Paketklasse ein *Credit* geführt wird. Hat dieser einen Wert von 0 oder ist positiv, darf ein Paket dieser Klasse über den Port verschickt werden. Während der Übertragung sinkt dieser *Credit* und führt dazu, dass nachfolgende Pakete der Klasse eventuell verzögert werden. Im Laufe der Zeit steigt der Wert wieder, bis er nicht mehr negativ ist. Mit Hilfe des CBS ist es möglich die Nachrichten einer Klasse gleichmäßig zu verteilen und so die reservierten Bandbreiten einzuhalten.

Avionics Full DupleX Switched Ethernet (AFDX) (siehe ARINC 664, 2009) wurde für sicherheitskritische Systeme in Flugzeugen konzipiert und wird aktuell auch in diesen eingesetzt (vgl. Brajou und Ricco, 2004; Schneele und Geyer, 2012). Die sicherheitskritische Kommunikation wird durch virtuelle Links realisiert. Dies sind statisch konfigurierte Pfade von einer Quelle zu einem oder mehreren Empfängern. Die Rate mit der Nachrichten über diesen virtuellen Link gesendet werden können, wird durch Traffic Shaper begrenzt. Zwischen zwei Paketen muss mindestens eine bestimmte Zeit liegen, die sogenannte Bandwidth Allocation Gap (BAG).

Time-Triggered Ethernet (TTE) (siehe AS 6802, 2016) erweitert das Ethernet nach IEEE 802.3 unter anderem um ein TDMA Verfahren, welches es ermöglicht Daten zu definierten Zeitpunkten über das Netzwerk zu schicken. Hierzu ist es notwendig, dass alle TTE-Komponenten über eine synchronisierte Zeitbasis verfügen.

Die Kommunikation in einem TTE-Netzwerk erfolgt über insgesamt drei Nachrichtenklassen:

- **Time-Triggered Nachrichten** bilden die erste Klasse der echtzeitfähigen Kommunikation und haben die höchste Priorität im Netzwerk. Für sie kommt das TDMA Verfahren zum Einsatz. Die genauen Sende- und Empfangszeiten dieser Nachrichten werden schon vor dem Systemstart festgelegt. Für jede Komponente auf dem Pfad ist genau definiert von wann bis wann die Nachricht ankommt und in welchem Zeitraum sie weitergeleitet werden muss. So ist es möglich, dass andere Nachrichten, die über den gleichen Port verschickt werden sollen, zurückgehalten werden um den höher priorisierten Datenverkehr nicht zu verzögern.
- **Rate-Constrained Nachrichten**, die zweite echtzeitfähige Nachrichtenklasse, entsprechen genau dem Prinzip von AFDX. Nur Time-Triggered Nachrichten können diese verzögern.

- **Best-Effort Nachrichten** sind identisch mit normalen Standard-Ethernet Nachrichten. Sie haben in einem TTE-Netzwerk die niedrigste Priorität und werden von den beiden echtzeitfähigen Nachrichtenklassen verdrängt.

Time Sensitive Networking (TSN) ist der Nachfolger von AVB und befindet sich derzeit noch in der Entwicklung durch die IEEE - Time-Sensitive Networking Task Group. Als Ziele werden angegeben, dass die Nachrichtenübertragung mit einer festen Latenz, niedrigem Jitter und mit geringem Paketverlust ermöglich werden soll (vgl. TSN TG). Konkrete Standards werden in einer Präsentation von der ISPCS 2015 in Beijing (vgl. Teener, 2015) genannt. Einige davon sind:

- Identifikation und Replikation von Frames für redundante Übertragungen (siehe IEEE 802.1CB, 2013)
- Weiterentwicklung des SRP (siehe IEEE 802.1Qcc, 2013)
- Scheduled Traffic mit Time Aware Shaper (siehe IEEE 802.1Qbv, 2012)
- Preemption, damit kritische Daten nicht so lange verzögert werden (siehe IEEE 802.1Qbu, 2012)

3.3 Anforderungen

Anforderungen an die Kommunikationssysteme werden zum einen von der Autoindustrie gestellt aber entstehen auch durch Vorgaben aus der ISO Norm 26262.

3.3.1 Anforderungen der Automobilindustrie

Die Autoherstellen möchten ihren Kunden natürlich ein Produkt bieten, welches einwandfrei funktioniert, um diese möglichst zufrieden zu stellen. Ein zuverlässiges Kommunikationssystem stellt hierfür eine gute Grundlage dar. Als weitere zentrale Anforderung ist die Wirtschaftlichkeit zu nennen. Durch die hohen Stückzahlen sind die Kosten jeder verbauten Komponente ein zentrales Kriterium. Zudem kann hier die Wartbarkeit des Systems angeführt werden, da andererseits hohe Reparaturkosten anfallen können.

3.3.2 ISO 26262

Die Grundlage der ISO 26262 (Road vehicles - Functional safety) bildet die Norm IEC 61508 (Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer

elektronischer Systeme). Sie ist speziell auf die Anforderungen der sicherheitskritischen Systeme im Auto ausgerichtet und dient als Grundlage für die Lebenszyklusphasen von PKW bis zu 3,5 Tonnen. Die Phasen reichen von der Konzeptphase, in der speziell Gefährdungen und Risiken behandelt werden, über die Produktentwicklung auf System-, Hardware- und Softwareebene bis hin zur Produktion und zum Betrieb. Einen wichtigen Punkt in der Norm nehmen die Automotive Safety Integrity Level (ASIL) ein, die im Folgenden erläutert werden.

Die ASIL dienen dazu die Funktionen im Auto in fünf Klassen einzuteilen:

- QM: keine speziellen Sicherheitsanforderungen
- ASIL A bis D: Funktionen benötigen spezielle Maßnahmen
  - ASIL A: niedrigste Anforderungen
  - ASIL D: höchste Anforderungen

Zur Einteilung in die jeweiligen Klassen müssen die Funktionen in den folgenden Punkten bewertet werden:

- Häufigkeit mit der die Funktion genutzt wird (engl. Exposure *E*)
  - *E0*: sehr selten
  - bis *E4*: oft
- Beherrschbarkeit durch den Fahrer bei einer Fehlfunktion (engl. Controllability *C*)
  - *C0*: sehr leicht
  - bis *C3*: sehr schwer oder gar nicht beherrschbar
- Schwere des möglichen Schadens (engl. Severity *S*)
  - *S0*: keine Verletzung
  - bis *S3*: Lebensgefährliche Verletzung

Je nachdem in welche Bereiche die Funktion eingeteilt werden kann, ergibt sich daraus das ASIL. Tabelle 3.1 zeigt die Verteilung der Klassen mit den unterschiedlichen Kombinationen der Kriterien. ASDIL D kommt beispielsweise ausschließlich bei häufig verwendeten, nicht kontrollierbaren und lebensgefährlichen Funktionen zum Einsatz. Die ASIL stellen ein geeignetes Mittel dar um als Bewertungsgrundlage eines Systems zu dienen. Ist die Ausfallwahrscheinlichkeit höher als die durch die Klassifizierung ermittelten Anforderungen, muss das System überarbeitet werden oder die Lösung ist für diese Funktion nicht einsetzbar. In Leu u. a. (2015)

Tabelle 3.1: Verteilung der ASIL

		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	ASIL A
	E4	QM	ASIL A	ASIL B
S2	E1	QM	QM	QM
	E2	QM	QM	ASIL A
	E3	QM	ASIL A	ASIL B
	E4	ASIL A	ASIL B	ASIL C
S3	E1	QM	QM	ASIL A
	E2	QM	ASIL A	ASIL B
	E3	ASIL A	ASIL B	ASIL C
	E4	ASIL B	ASIL C	ASIL D

werden die Grenzen für die maximale Ausfallwahrscheinlichkeit der Level B, C und D wie folgt angegeben:

- ASIL B = $1 \cdot 10^{-7}$
- ASIL C = $1 \cdot 10^{-7}$
- ASIL D = $1 \cdot 10^{-8}$

4 Redundanz

Aus der DIN 40041 zum Thema Redundanz:

> „Vorhandensein von mehr funktionsfähigen Mitteln in einer Einheit, als für die Erfüllung der geforderten Funktion notwendig sind." (vgl. DIN 40041, 1990)

Da sicherheitskritische Systeme, wie sie im Flugzeug, Auto oder in einem Atomkraftwerk vorkommen, Anforderungen aus Normen wie der IEC 61508 oder der ISO 26262 erfüllen müssen, spielt das Thema Redundanz eine wichtige Rolle. Mit den richtigen Mitteln lassen sich die Systeme so absichern, dass ganze Teile der Anlage ausfallen können, ohne dass es zu einer Katastrophe kommt.

Aber Redundanz geht auch immer mit höheren Kosten und neuen Anforderungen einher. Deswegen gilt es für ein System das passende Konzept zu finden. Das folgende Kapitel befasst sich mit den unterschiedlichen Anforderungen, die ein System stellen kann und vermittelt Konzepte, wie Redundanz realisiert werden kann.

4.1 Klassifikation und Anforderungen

Vor dem Design eines redundanten Systems muss geklärt werden, welche Anforderungen die sicherheitskritischen Funktionen stellen. In der Arbeit (Kirrmann und Dzung, 2006) werden die jeweiligen Anlagen in drei Klassen unterteilt und somit die Anforderungen an das zugrundeliegende redundante Netzwerk definiert. Die Klassen basieren auf der maximalen Reaktionszeit, die von den sicherheitskritischen Funktionen vorausgesetzt wird. Die Zeit, die ein System benötigt in einem Fehlerfall wieder einsatzbereit zu sein, spielt hier also eine große Rolle. Die Klassen werden wie folgt definiert:

All-round plants sind Anlagen mit weichen Echtzeitanforderungen, bei denen eine Reaktionszeit von unter zwei Sekunden ausreichend ist.

Benign plants sind Anlagen mit höheren Echtzeitanforderungen und erwarten eine maximale Reaktionszeit von unter 50 ms. Als Beispiele werden hier die chemische Industrie und Kraftwerke genannt.

Critical plants sind Anlagen, bei denen harte Echtzeitanforderungen gestellt werden. Die Netzwerke dieser Systeme müssen innerhalb von maximal 2 ms wieder funktionstüchtig sein. Diese höchsten Anforderungen gibt es bei der Robotersteuerung oder X-By-Wire Anwendungen in der Automobilindustrie.

Es gibt noch weitere Anforderungen, die die Art und das Ausmaß der Redundanz beeinflussen. Wenn das System einen teilweisen Systemausfall tolerieren kann, können eventuell andere Komponenten die Aufgaben des ausgefallenen Elements übernehmen. Dann gibt es noch eine Reihe an ökonomischen Gesichtspunkten. Der erste Punkt wären die direkten Kosten durch die Redundanz. Wird mehr Hardware verbaut, fallen auch höhere Kosten an. Außerdem sinkt bei einer höheren Anzahl an Komponenten die durchschnittliche Zeit bis zur nächsten Reparatur, da selbst nicht aktiv genutzte Elemente ausfallen können. Das Verhalten des Systems während eines Fehlers ist noch ein weiterer Punkt. Hier ist die Frage, ob das System im Fehlerfall in einen sicheren Zustand überführt werden kann.

4.2 Redundanzkonzepte

Redundanz kann durch einige unterschiedliche Herangehensweisen realisiert werden. Das einfache Vorhandensein mehrerer Elemente führt im Normalfall nicht zum gewünschten Erfolg, da das System auch lernen muss mit den zusätzlichen Mitteln umzugehen. In diesem Abschnitt werden unterschiedliche Konzepte vorgestellt, wie redundante Systeme aussehen können. Unterteilt werden sie in Redundanz in Netzwerken durch unterschiedliche Topologiekonzepte, Hardwareredundanz und Softwareredundanz.

4.2.1 Redundanz durch Topologiekonzepte

Um Redundanz durch die Topologie zu ermöglichen verfügen die Netzwerke über alternative Routen, die bei einem Fehlerfall weiterhin eine funktionstüchtige Kommunikation sicherstellen sollen. Solche Netzwerke können auf unterschiedlichste Art und Weise aufgebaut werden. Aus den verschiedenen Konzepten ergeben sich auch verschiedene Eigenschaften, wie etwa die Dauer der Umschaltzeiten.

Die DIN EN 62439 beschreibt eine Menge an unterschiedlichen Protokollen um hochverfügbare Kommunikationsnetzwerke für die Industrie zu definieren. Alle in dieser Norm enthaltenen Protokolle basieren auf der Ethernet-Technologie IEEE 802.3 und eine Auswahl daraus wird unter anderem in diesem Abschnitt vorgestellt.

(Rapid) Spanning Tree Protocol ((R)STP) Mit dem IEEE Standard 802.1D von 1990 wurde unter anderem das STP standardisiert. Der Hintergrund war, dass in einem LAN keine Schleifen existieren sollen, damit Nachrichten nicht unerwünscht in dem Netzwerk kreisen. Deswegen muss sichergestellt werden, dass bei mehreren Verbindungen zwischen zwei Netzwerkteilnehmern, nur eine Leitung genutzt wird. Diese Aufgabe übernimmt das Spanning Tree Protocol. Im ersten Schritt wird unter allen Switches und Bridges des Netzwerks die Root Bridge gewählt. Von dort aus werden die Pfade festgelegt, die zwischen den Teilnehmern aktiv sind. Enthält das Netzwerk redundante Pfade, darf nur einer davon aktiv bleiben. Fällt einer der Pfade aus, müssen die kompletten Pfade neu berechnet werden. Dies kann durch die maximalen Zeiten für einige Zustände bis zu 30 Sekunden in Anspruch nehmen. Das Rapid Spanning Tree Protocol hingegen nutzt die bisherige Konfiguration erstmal weiter, bis die neuen alternativen Pfade berechnet wurden und diese genutzt werden. So lässt sich die Ausfallzeit auf unter eine Sekunde reduzieren.

Media Redundancy Protocol (MRP) Bei dem Media Redundancy Protocol (vgl. DIN EN 62439-2, 2010) ist das Netzwerk in einer Ringtopologie aufgebaut. Ein Knoten in diesem Ring ist der *Media Redundancy Manager* (MRM). Dieser hat die Aufgabe den Ring zu überwachen und auf eventuelle Fehler zu reagieren. Der MRM schickt regelmäßig Pakete in beide Richtungen des Ringes und erwartet deren Ankunft auf dem jeweils anderen Port. Empfängt er beide Pakete, liegt kein Problem auf dem Netzwerk vor und er blockiert einen Port, sodass über diesem Pfad keine reguläre Kommunikation stattfindet.

Sobald ein Link oder ein Knoten im Ring ausfällt, wird der Ring neu konfiguriert. Über den zuvor blockierten Port werden jetzt auch die regulären Datenpakete weitergeleitet und die Kommunikation zwischen den funktionierenden Elementen ist weiter sichergestellt. Je nach Konfiguration des Netzwerks sind Umschaltzeiten von 10 bis 500 ms möglich.

Parallel Redundancy Protocol (PRP) Bei diesem Protokoll wird in der (DIN EN 62439-3, 2013) beschrieben. Das gesamte System ist in zwei voneinander unabhängige Netzwerke aufgeteilt und bietet den Knoten die Möglichkeit parallel über beide zu kommunizieren. So können Fehler in dem einen Netzwerk durch die redundante Nachricht kompensiert werden.

Die Knoten, die an beide Netzwerke angeschlossen sind, werden als *Doubly Attached Nodes* (DAN) bezeichnet. Es ist aber auch möglich *Singly Attached Nodes* (SAN), die lediglich an ein Netzwerk angeschlossen sind, in das System zu integrieren. Diese Knoten können aber auch nur mit anderen Netzwerkteilnehmern kommunizieren, die an das gleiche Netzwerk angeschlossen sind.

Um einem DAN zu ermöglichen Duplikate zu erkennen, werden die Frames um einen *Redundancy Control Trailer* erweitert. Dieser enthält unter anderem eine Sequenznummer, die bei jedem neuen Frame hochgezählt wird. Mit dieser Nummer sowie der MAC-Adresse des Senders ist es dem Empfänger möglich ein Duplikat zu identifizieren. Die Nachricht, welche später eintrifft, wird verworfen.

High Availability Seamless Redundancy (HSR) Auch bei dem Konzept der HSR (vgl. DIN EN 62439-3, 2013) wird das Netzwerk, wie beim MRP, als Ring aufgebaut. Anders als bei STP oder MRP wird hier aber keine Leitung deaktiviert. Die Daten werden über beide Ports gleichzeitig in beide Richtungen des Rings gesendet. Hier werden sie von Knoten zu Knoten weitergeleitet, entweder bis das Paket sein Ziel erreicht und der Empfänger das Paket nicht mehr weiterleitet (im Fall von Unicast) oder bis es wieder an seinem Ausgangspunkt angekommen ist. Genau wie beim PRP können duplizierte Frames anhand ihrer Sequenznummer und der MAC-Adresse des Senders identifiziert werden und auch hier werden die später ankommenden Duplikate verworfen.

Der Vorteil dieses Verfahrens ist, dass bei einem Fehler auf dem einen Pfad die Daten vom anderen Pfad noch ankommen. Es fallen also keinerlei Umschaltzeiten an. Es wird allerdings auch doppelt so viele Nachrichten erzeugt, was das Netzwerk mehr belastet.

4.2.2 Hardwareredundanz

Das Erweitern eines Systems um redundante Hardware hat zum Ziel dieses vor Ausfällen einzelner Komponenten zu schützen. Das Design der redundanten Elemente kann je nach Anforderungen an das System unterschiedlich gestaltet werden. Mögliche Ansatzpunkte sind die Anzahl der zusätzlichen Komponenten, die Entscheidung, ob es sich um aktive oder um inaktive Standby-Elemente handelt, oder wie genau die unterschiedlichen Ergebnisse überprüft werden sollen. Ein weiterer wichtiger Punkt ist, dass man sich der zusätzlichen Probleme und den Anforderungen durch die zusätzliche Hardware bewusst sein muss.

Zusätzliche Probleme und Anforderungen

Wie zuvor erwähnt, bringt Redundanz zusätzliche Anforderungen und Probleme mit sich. Liegen beispielsweise Ergebnisse von mehreren redundanten Elementen vor, muss eines ausgewählt werden. Solange alle Komponenten einwandfrei arbeiten, ist das kein Problem aber sobald ein falsches Ergebnis vorliegt, muss eine Entscheidung getroffen werden, welches weitergeleitet werden soll.

Ein ähnliches Problem stellt das Erkennen von Fehlern dar. Enthält die redundante Lösung inaktive Elemente, die beim Ausfall des aktiven Elements einspringen, muss dieser Zeitpunkt unter Umständen möglichst schnell erkannt werden, damit die Umschaltzeiten so gering wie möglich gehalten werden können.

Abhängigkeiten zwischen den redundanten Elementen können dazu führen, dass die Änderungen am System keine großen Auswirkungen auf die gesamte Ausfallwahrscheinlichkeit haben. Beziehen die Komponenten den Strom aus der gleichen Quelle, sind sie den gleichen äußeren Einflüssen ausgesetzt oder kommunizieren sie über den gleichen fehleranfällig Pfad? All das wären Möglichkeiten, die die Vorteile der redundanten Hardware egalisieren könnten.

Ein weiterer Punkt, der beachtet werden muss ist, dass durch zusätzliche Hardware auch die Anforderungen an das gesamte System steigen. Es wird mehr Strom benötigt, es fällt zusätzliches Gewicht an und auch der Bedarf an Platz steigt.

Statische Redundanz

Als statische Redundanz oder auch funktionsbeteiligte Redundanz werden die Konzepte bezeichnet, bei denen alle redundanten Elemente jederzeit aktiv an der entsprechenden Funktion arbeiten. Die Ergebnisse aller Module werden auch bei der Ergebnisfindung berücksichtigt, was dazu führt, dass irgendeine Art von Entscheidung getroffen werden muss, sollten unterschiedliche Ergebnisse vorliegen. In (Storey, 1996, S. 132 - 136) werden einige unterschiedlich komplexe Arten vorgestellt:

Triple modular redundancy (TMR) ist ein statisches Konzept, das das System gegen ein fehlerhaftes Modul schützt. Hier empfangen drei Module die Eingangsdaten, zum Beispiel von einem Sensor, und ein Voter entscheidet welches Ergebnis weitergeleitet wird (siehe Abbildung 4.1). Ein einfaches Verfahren wäre hier alle drei Ergebnisse miteinander zu vergleichen und per Mehrheitsentscheid ein möglicherweise falsches Ergebnis zu überstimmen. Sobald aber zwei Module nicht richtig arbeiten, funktioniert das Konzept nicht mehr. Auch Probleme beim Input (z. B. falsche Sensordaten), beim Voter oder beim Output können zu falschen Ergebnissen führen, da sie einen Single Point of Failure (SPOF) darstellen.

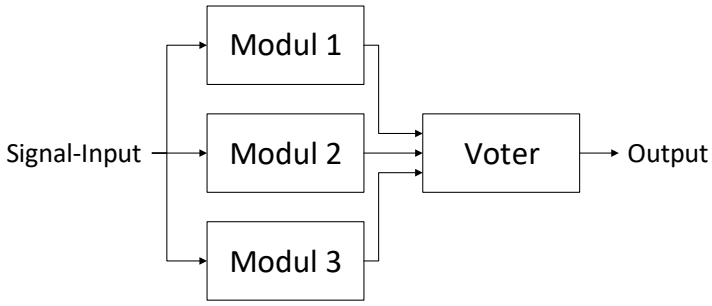


Abbildung 4.1: Einfache TMR

Um dem Problem der SPOFs entgegenzuwirken, kann die Redundanz auch erweitert werden. In dem Beispiel in Abbildung 4.2 bekommt jedes Modul die Daten von einem anderen Sensor und auch für den Voter gibt es drei Instanzen, von der jede seinen eigenen Output hat. In diesem System würde allerdings die Wahrscheinlichkeit steigen, dass die Module aufgrund unterschiedlicher Sensordaten auch unterschiedliche Ergebnisse liefern. Dadurch wären die Voter wiederum stärker gefordert auf solche Situationen reagieren zu können.

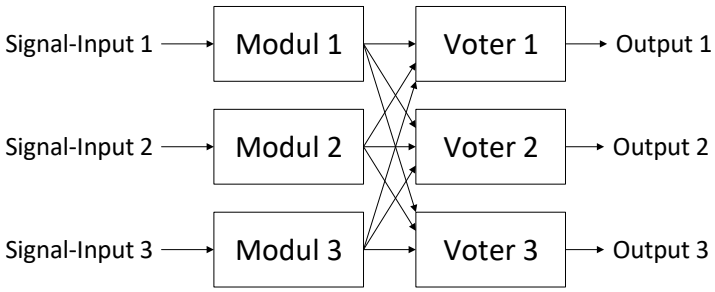


Abbildung 4.2: TMR mit dreifachem Input und drei Votern

N-modular redundancy (NMR) kann als Erweiterung der TMR gesehen werden. Es ist möglich nicht nur drei redundante Module ins System einzubinden sondern je nach Bedarf auch mehrere (siehe Abbildung 4.3). Mit fünf Modulen ließen sich beispielsweise zwei gleichzeitige Fehler kompensieren. Aber es gilt zu bedenken, dass mehr Komponenten auch wieder in mehr Kosten, in einen höheren Energieverbrauch oder auch in eine höhere Komplexität resultieren.

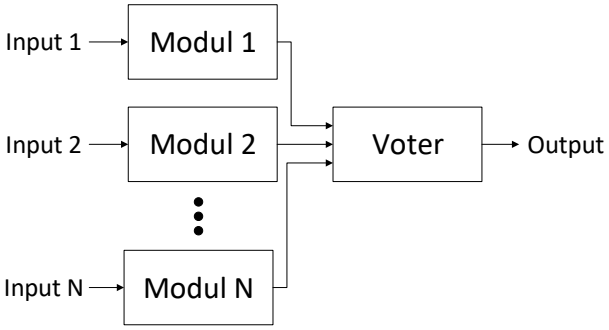


Abbildung 4.3: NMR mit beliebig vielen Modulen

Dynamische Redundanz

Die dynamische Redundanz, auch nicht funktionsbeteiligte Redundanz, umfasst Systeme, deren redundante Module nicht aktiv am Prozess teilnehmen. Sie werden erst aktiviert, sobald bei der derzeit aktiven Komponente ein Fehler vorliegt. Dies erfordert, dass es ein weiteres Modul zu Fehlererkennung gibt und zusätzlich einen Mechanismus, der zwischen den aktiven Komponenten wechseln kann. Ein solcher Aufbau wird in Abbildung 4.4 gezeigt. Hier werden die Ergebnisse der Module sowohl an die Fehlererkennung als auch Richtung Output an den Switch weitergeleitet. Sobald ein Fehler erkannt wird, kann der Switch benachrichtigt werden und das Umschalten von einem Modul auf das andere eingeleitet werden.

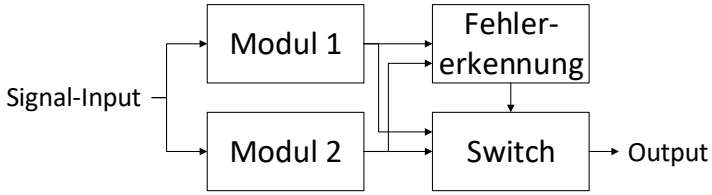


Abbildung 4.4: Dynamische Redundanz mit einem Spare

Die Realisierung eines Systems mit dynamischer Redundanz ist potentiell die günstigere Lösung im Vergleich zur einer statischen Lösung. Wo bei der TMR mindestens drei Module benötigt werden, kann hier schon mit zwei Modulen ein redundantes System geschaffen werden. Außerdem können sich die redundanten Alternativen (Spares) in einem inaktiven Zustand befinden, in dem sie weniger Strom verbrauchen und gleichzeitig auch der Verschleiß gering gehalten werden kann. Dies ist ein weiterer Vorteil gegenüber der statischen Redundanz. Allerdings muss das System auch temporäre Fehler tolerieren, denn das Erkennen des Fehlers,

das Umschalten auf ein neues Modul und das Eintreffen des neuen Ergebnisses benötigt eine gewisse Zeit. Diese Zeit hängt auch davon ab, wie die redundanten Module konfiguriert wurden. Sie können in drei Kategorien eingeteilt werden:

Kalte Spares befinden sich in einem komplett inaktiven Zustand und müssen in einem Fehlerfall erst gestartet werden. Bis dahin verbrauchen die Module keinen Strom und auch der Verschleiß ist sehr gering. Das System muss im Falle eines Fehlers aber warten bis das Modul gestartet und initialisiert ist.

Warme Spares sind bereits gestartet und warten auf ihren Einsatz. Wird ein Fehler bei der aktiven Komponente entdeckt und schaltet das System auf den Spare um, beginnt er mit der Arbeit und kann anschließend Ergebnisse liefern. Warme Spares haben im Vergleich zu Kalten eine geringere Umschaltzeit, da sie nicht erst gestartet werden müssen. Allerdings startet das Modul erst mit der Arbeit sobald es aktiviert wird, was zu Wartezeiten führt.

Heiße Spares sind ebenfalls bereits gestartet, arbeiten aber, im Gegensatz zu warmen Spares, bereits aktiv an ihrer jeweiligen Funktion. Ihre Ergebnisse werden lediglich nicht genutzt. Dies hat den Vorteil im Fehlerfall sehr schnell auf das Ergebnis des Spares zugreifen und so die Umschaltzeiten gering halten zu können. Der Vorteil des geringeren Stromverbrauchs und des reduzierten Verschleißes ist bei dieser Variante allerdings nicht mehr gegeben.

4.2.3 Softwareredundanz

Neben der Hardware kann auch die Software für Fehler verantwortlich sein. Softwarefehler resultieren meist aus Designfehlern oder falschen Implementierungen (vgl. Pullum, 2001, S. 18). Daraus folgt, dass der redundante Einsatz identischer Programme den Großteil der Fehler nicht beheben wird. Um die Gefahr derartiger Fehler zu reduzieren, gibt es verschiedene Techniken die Software widerstandsfähiger zu machen.

N-version programming (NVP) steht für den Ansatz für die gleiche Funktion unterschiedliche Softwarelösungen zu erstellen. Als Voraussetzung für jede Version dient beispielsweise eine Spezifikation, aufgrund derer verschiedene Entwicklerteams, im Optimalfall mit unterschiedlichen Programmiersprachen, unabhängig voneinander die Software entwickeln. Genau wie schon bei der NMR ist es denkbar eine gewisse Anzahl unterschiedlicher Versionen zu betreiben um so möglichst einfach Fehler zu erkennen. Allerdings gilt es zu beachten, dass

redundante Software hohe Entwicklungskosten mit sich bringt. Auch ist der Aufwand diese zu Warten deutlich größer als bei einer einzigen Version.

Zusätzlich zu den Softwarelösungen kommt auch noch hinzu, dass, wie schon bei der NMR, eine Entscheidung getroffen werden muss, welches Ergebnis von welcher Software das richtige ist. Auch hier muss also ein Votingelement entwickelt werden, das diese Aufgabe übernimmt. Die nächste Frage, die sich hieraus ergibt, ist, wie die Software in das System eingebunden wird:

- Läuft alles auf der gleichen Hardware?
- Gibt es für jede Software ein eigenes Hardwareelement?
- Wo befindet sich der Voter?

In (Pullum, 2001, S. 19) werden unterschiedliche Varianten vorgestellt. Die erste Möglichkeit wäre die komplette Software auf der gleichen Hardware laufen zu lassen (siehe Abbildung 4.5). Hierdurch wären die Anforderungen an die Hardware besonders hoch, da sie einen SPOF darstellt und außerdem auch die benötigte Leistung, beispielsweise die der CPU, deutlich steigt. Die zweite Möglichkeit ist es die einzelnen Softwarelösungen auch auf unterschiedliche Hardware auszulagern (siehe Abbildung 4.6 (a) und (b)). Hier kann noch entschieden werden, ob der Voter auf einer eigenen Hardware läuft (b) oder auf der Hardware eines Softwaremoduls (a).

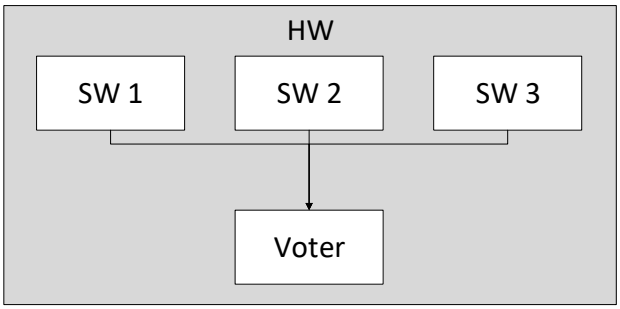
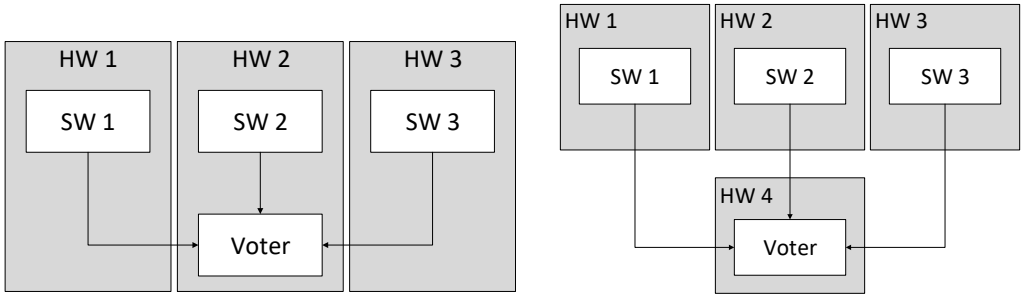


Abbildung 4.5: Softwareredundanz auf einer Hardware



(a) Voter integriert

(b) Voter auf eigener Hardware

Abbildung 4.6: Softwareredundanz auf verteilter Hardware

5 Fehlerbaumanalyse

Die DIN 25424 (vgl. DIN 25424-1, 1981; DIN 25424-2, 1990) befasst sich mit den Zielen, dem Aufbau und der Verwendung von Fehlerbäumen. Außerdem werden Berechnungen vorgestellt, mit deren Hilfe etwa Zuverlässigkeitskenngrößen wie z. B. die Ausfallhäufigkeit, die Eintrittshäufigkeit eines unerwünschten Ereignisses oder die Wahrscheinlichkeit der Nichtverfügbarkeit des Systems zu einem bestimmten Zeitpunkt berechnet werden. Auch die wesentlich neuere DIN 61025 aus dem Jahr 2007 (vgl. DIN EN 61025, 2007) deckt diese Themenbereiche ab, ist aber in allen Bereichen detaillierter und behandelt auch neue Themen. Hierzu zählen beispielsweise dynamische Gates oder die genaue Herangehensweise an die Konstruktion eines Fehlerbaums.

In diesem Kapitel werden die Funktionen und die Berechnungen sowohl von den statischen als auch den dynamischen Elementen erläutert. Zusätzlich werden anhand von Beispielen die unterschiedlichen Schritte zur Berechnung gezeigt.

5.1 Statische Fehlerbaumanalyse

Um ein System anhand eines Fehlerbaums zu analysieren, wird zuerst ein Top-Ereignis (auch Top-Event) definiert. Dieser unerwünschte Fall stellt das Ereignis dar, das anhand der Fehlerbaumanalyse untersucht werden soll. Anschließend wird das zu untersuchende System abgebildet. Alle Ereignisse und Elemente, die Auswirkungen auf das Top-Ereignis haben, werden mit ihren entsprechenden Eigenschaften im Baum angeordnet. Zur Darstellung logischer Zusammenhänge der einzelnen Komponenten stehen AND-, OR- und NOT-Gates zur Verfügung.

Wurde ein System entsprechend gestaltet, kann dieses analysiert werden. Eine solche Analyse liefert mehrere Ergebnisse, die zur Beurteilung des Baums herangezogen werden. Zuerst wäre da die Eintrittswahrscheinlichkeit des Top-Ereignisses. Die ergibt sich aus der entsprechenden Kombination der einzelnen Ausfallwahrscheinlichkeiten aller Events (siehe Abschnitt 5.1.2). Es handelt sich hierbei also um die Ausfallwahrscheinlichkeit des gesamten Systems. Außerdem erhält der Benutzer Informationen darüber, welche Ausfallkombinationen zum Eintreten des Top-Events führen. Man kann mit diesen Kombinationen erkennen, welche Ereignisse gleichzeitig eintreten müssen um einen Fehler zu produzieren. Zu jeder Ausfallkombination kann

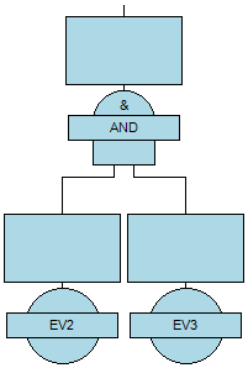


Abbildung 5.1: AND-Gate in einem Fehlerbaum

Tabelle 5.1: Funktionstabelle AND-Gate

Event 1	Event 2	Ausgang
1	1	1
1	0	0
0	1	0
0	0	0

die jeweilige Wahrscheinlichkeit ermittelt werden, mit der diese eintreten wird. So können die kritischen Bereiche des Systems ermittelt werden und gegebenenfalls besser abgesichert werden. Die Minimal Cut Sets bilden hier eine spezielle Gruppe der Ausfallkombinationen, bei denen besonders wenig Events eintreten müssen um zum Fehler zu führen. Diese können tendenziell als kritischere Bereiche des Systems bezeichnet werden.

5.1.1 Elemente

Die statischen Fehlerbäume setzen sich aus dem unerwünschten TOP-Ereignis, verschiedenen logischen Gattern und diversen Events, die zu dem unerwünschten Ereignis führen, zusammen. Aus der Kombination dieser Elemente lassen sich die Fehlerbäume konstruieren um ein System zu untersuchen.

TOP-Ereignis Der Ausgang des gesamten Fehlerbaums ist das unerwünschte Ereignis, das untersucht werden soll.

Event Ein Event repräsentiert den Ausfall oder einen Fehler eines Elements innerhalb des gesamten Systems. Tritt ein Event ein, kann dieses dazu beitragen, dass das TOP-Ereignis ebenfalls eintritt. Hierbei kann es sich um Soft- oder Hardwareausfällte handeln aber auch um Faktoren wie menschliches Versagen oder äußere Einflüsse.

AND-Gate Damit dieses Gate (Abbildung 5.1) zu einem Fehler führt, müssen alle darunter liegenden Events oder Gates fehlerhaft sein. Die Funktionstabelle 5.1 zeigt die Ausgänge für zwei eingehende Ereignisse.

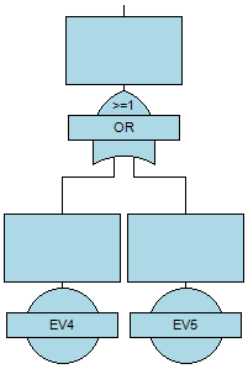


Abbildung 5.2: OR-Gate in einem Fehlerbaum

Tabelle 5.2: Funktionstabelle OR-Gate

Event 1	Event 2	Ausgang
1	1	1
1	0	1
0	1	1
0	0	0

OR-Gate Sobald eins der unter diesem Gate (Abbildung 5.2) liegenden Events oder Gates fehlerhaft ist, schaltet dieses Gatter. Die Funktionstabelle 5.2 zeigt die Ausgänge für zwei eingehende Ereignisse.

5.1.2 Berechnung

Dieser Abschnitt befasst sich mit der Berechnung der AND- und OR-Gates sowie dem Umgang mit identischen Events in unterschiedlichen Teilen des Fehlerbaums.

Gates

AND Berechnung der Nichtverfügbarkeit:

$$U = \prod_{i=1}^{n} U_i \tag{5.1}$$

OR Berechnung der Nichtverfügbarkeit:

$$U = 1 - \prod_{i=1}^{n} (1 - U_i) \tag{5.2}$$

Disjunktes Zerlegen

Bei der Beschreibung eines Systemausfalls kann es sein, dass das selbe Ereignis in unterschiedlichen Teilen des Fehlerbaums zum Eintreten des Top-Events beiträgt. Wird dieser Umstand bei der Berechnung ignoriert, fließt die Ausfallwahrscheinlichkeit des Ereignisses öfter in das Gesamtergebnis ein. Dies führt zu einem ungenauen Ergebnis. Um das zu verhindern,

wird in der DIN 61025 (vgl. DIN EN 61025, 2007, S. 44) gezeigt, wie der zu einem Fehlerbaum aufgestellte Ausdruck disjunkt zerlegt werden kann.

Der durch boolesche Terme definierte Ausdruck wird nach und nach zerlegt, indem alle einzelnen Terme miteinander verglichen und gegebenenfalls angepasst werden. Zu Beginn werden die zu vergleichenden Terme daraufhin untersucht, ob irgendeine Variable des einen Terms komplementär im anderen Term vorhanden ist. Sollte dies der Fall sein, sind diese beiden Terme bereits disjunkt und müssen nicht weiter betrachtet werden. Trifft es nicht zu, werden alle Variablen aus dem ersten Term, die nicht im Zweiten vorkommen, genommen und der zweite Term durch Terme mit den jeweiligen komplementären Variablen ersetzt. Diese beiden Terme sind nun disjunkt.

Anschließend wird der erste Term mit dem nächsten Term des Ausdrucks verglichen. Auch die werden wie zuvor beschrieben auf das Vorhandensein eines komplementären Elements überprüft und entsprechend angepasst. Ebenso wird mit allen weiteren Termen verfahren, bis der erste Term mit allen anderen verglichen wurde. Bei dem hieraus entstandenen Ausdruck wird nun der zweite Term dem Dritten gegenübergestellt. Anschließend mit dem Vierten und so weiter, bis jeder Term mit jedem verglichen wurde.

Die folgenden Beispiele zeigen unterschiedliche Möglichkeiten, bezüglich des disjunkten Zerlegens:

Beispiel 1 Gegeben sei folgender Ausdruck:

$BSP1 = ab + \overline{b}c$

Betrachtet man diese beiden Terme, erkennt man, dass die Variable b in beiden Termen in komplementärer Form vorhanden ist. Beim disjunkten Zerlegen muss somit kein Term ersetzt werden.

Beispiel 2 Gegeben sei folgender Ausdruck:

$BSP2 = ab + bc$

In diesem Beispiel ist keine komplementäre Variable bezüglich des erstens Terms im zweiten Term. Der zweite Term muss also ersetzt werden. Jetzt gilt es alle Variablen aus dem ersten Term, die nicht im zweiten Term vorkommen, komplementär in den zweiten Term einzufügen und diesen damit zu ersetzen. In diesem Fall ist das Variable a und es ergibt sich folgender neuer Ausdruck:

$BSP2_1 = ab + \overline{a}bc$

Beispiel 3 Gegeben sei folgender Ausdruck:

$BSP3 = ab + cd$

In diesem Beispiel ist keine komplementäre Variable bezüglich des erstens Terms im zweiten Term. Der zweite Term muss also ersetzt werden. Jetzt gilt es alle Variablen aus dem ersten Term, die nicht im zweiten Term vorkommen, komplementär in den zweiten Term einzufügen und diesen damit zu ersetzen. In diesem Fall sind das die Variablen a und b. Für jede Variable wird ein neuer Term in den Ausdruck eingefügt und es ergibt sich folgender neuer Ausdruck:

$BSP3_1 = ab + \overline{a}cd + a\overline{b}cd$

5.1.3 Berechnung eines statischen Fehlerbaums

In diesem Abschnitt wird die Berechnung eines statischen Fehlerbaums näher betrachtet. Neben der Berechnung der einzelnen Gates liegt der Fokus auf dem disjunkten Zerlegen des Baums um ein exaktes Ergebnis zu erhalten. Der Aufbau des Fehlerbaums ist in Abbildung 5.3 dargestellt.

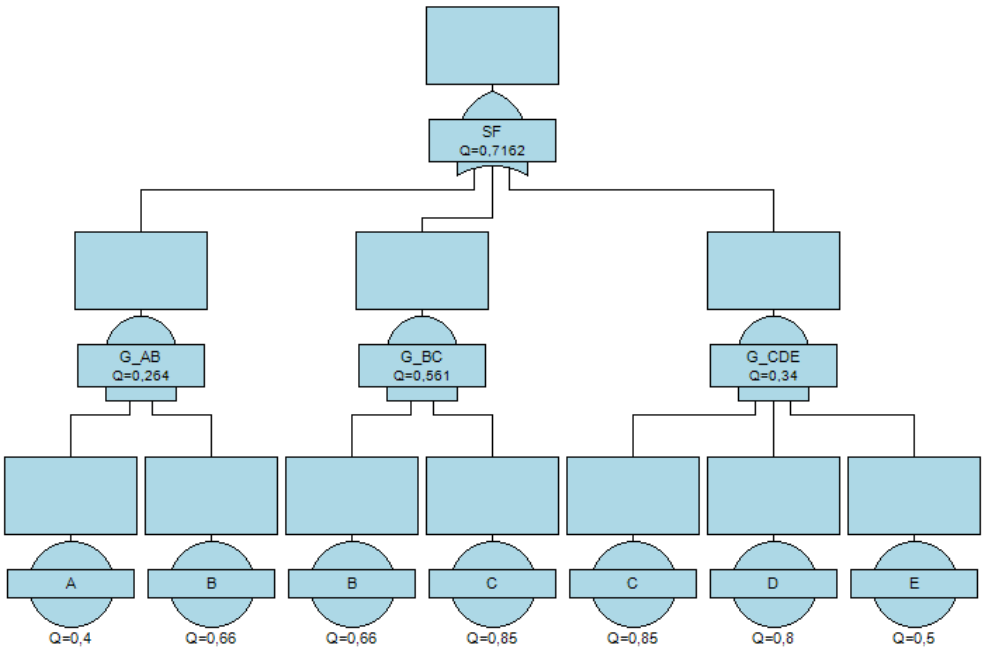


Abbildung 5.3: Beispielhafter Fehlerbaum

Folgende Ausfallwahrscheinlichkeiten sind für die jeweiligen Events gegeben:

$$F_a = 0,4; \quad F_b = 0,66; \quad F_c = 0,85; \quad F_d = 0,8; \quad F_e = 0,5$$

Um die Ausdrücke übersichtlicher zu halten, werden die Ausfallwahrscheinlichkeiten folgendermaßen abgekürzt:

$$F_a = a; \quad F_b = b; \quad F_c = c; \quad F_d = d; \quad F_e = e$$

Die Berechnung der drei AND-Gates erfolgt mit Hilfe der Formel 5.1:

$$G\_AB = ab = 0,4 \cdot 0,66 = 0,264$$
$$G\_BC = bc = 0,66 \cdot 0,85 = 0,561$$
$$G\_CDE = cde = 0,85 \cdot 0,8 \cdot 0,5 = 0,34$$

Da sich die beiden Ereignisse B und C jeweils unter zwei AND-Gates befinden, muss der Ausdruck zu dem Fehlerbaum, wie im vorherigen Abschnitt beschrieben, disjunkt zerlegt werden. Hierzu wird zur Berechnung des Systemausfalls (system failure) der Ausdruck SF passend zum Fehlerbaum aufgestellt:

$$SF = ab + bc + cde$$

Ohne weitere Anpassungen und ohne Berücksichtigung der Formel 5.2 für OR-Gates würde der Ausdruck nur eine sehr grobe Annährung, mit einer Wahrscheinlichkeit von über eins, an das tatsächliche Ergebnis liefern:

$$SF = G\_AB + G\_BC + G\_CDE = 0,264 + 0,561 + 0,34 = 1,165$$

Wird die Formel genutzt und nur das disjunkte Zerlegen außen vor gelassen erhält man schon ein deutlich besseres Ergebnis:

$$SF = 1 - (1 - G\_AB)(1 - G\_BC)(1 - G\_CDE) = 1 - 0,736 \cdot 0,439 \cdot 0,66 = 0,7868$$

Zum disjunkten Zerlegen wird der Ausdruck SF als Ursprung genommen. Im ersten Schritt werden die beiden Terme ab und bc verglichen. Da weder a noch b im zweiten Term komplementär vorhanden sind, muss dieser ersetzt werden. Dazu wir das Komplement aller Attribute, die nicht im zweiten Term vorkommen, diesem hinzugefügt. In diesem Fall ergibt sich also ein neuer zweiter Term $\overline{a}bc$. Im Abschluss wird der erste Term mit dem Dritten verglichen und da auch dieser kein komplementäres Attribut enthält, wird er ebenfalls ersetzt. In diesem Fall durch $\overline{a}cde$ und $a\overline{b}cde$ und es ergibt sich somit der neue Ausdruck SF_1:

$$SF_1 = ab + \overline{a}bc + \overline{a}cde + a\overline{b}cde$$

Im zweiten Schritt wird der neue zweite Term mit dem dritten und vierten Term verglichen. Da im Dritten keines der Attribute komplementär vorkommt und b gar nicht ergibt sich der neue Term $\overline{ba}cde$. Beim Vergleich mit dem vierten Term gibt es sowohl für $\overline{a}$ als auch für b ein komplementäres Attribut. Somit muss dieser Term nicht ersetzt werden und es ergibt sich der neue Ausdruck SF_2:

$$SF_2 = ab + \overline{a}bc + \overline{ba}cde + a\overline{b}cde$$

Beim letzten Vergleich zwischen dem dritten und vierten Term muss aufgrund des Attributs a keine Anpassung vorgenommen werden. Somit ist der neue Ausdruck identisch mit dem vorherigen:

$$SF_3 = ab + \overline{a}bc + \overline{ba}cde + a\overline{b}cde$$

In diese Formel können nun die Werte für die Ausfallwahrscheinlichkeiten eingesetzt werden und die Wahrscheinlichkeit für einen Systemausfall berechnet werden:

$$\begin{aligned} SF_3 &= 0,4 \cdot 0,66 + 0,6 \cdot 0,66 \cdot 0,85 \\ &\quad + 0,34 \cdot 0,6 \cdot 0,85 \cdot 0,8 \cdot 0,5 \\ &\quad + 0,4 \cdot 0,34 \cdot 0,85 \cdot 0,8 \cdot 0,5 \\ SF_3 &= 0,264 + 0,3366 + 0,06936 + 0,04624 \\ SF_3 &= 0,7162 \end{aligned}$$

Dies ist das genaue Ergebnis des Fehlerbaums und diese Art der Berechnung sollte für alle Fehlerbäume genutzt werden. Diese Methode wird auch von den verschiedensten Softwarelösungen für Fehlerbäume genutzt.

5.2 Dynamsiche Fehlerbaumanalyse

Mit statischen Fehlerbäumen können sich verändernde Systeme nur schwer und nicht vollständig abbilden lassen. Kann ein System auf gewisse Fehler reagieren und so die Funktionalität weiterhin gewährleisten, muss auch der dazugehörige Fehlerbaum entsprechend auf den Fehler reagieren können. Für diesen Zweck wurden die dynamischen Fehlerbäume entwickelt (vgl. Dugan u. a., 1992). Um Konzepte wie Redundanz, Abhängigkeit zwischen Ereignissen oder komplexe Fehlerbehebung zu realisieren, wurden zusätzliche dynamische Gates eingeführt.

5.2.1 Elemente

Die dynamischen Fehlerbäume ermöglichen den Einsatz vier weiterer Gates. Jedes dieser Gates wird unter den folgenden Punkten erläutert.

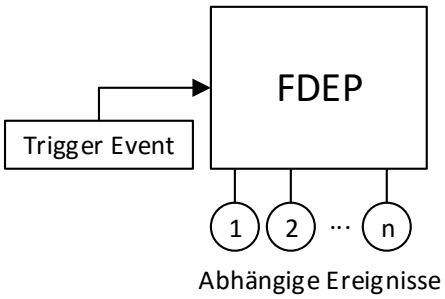
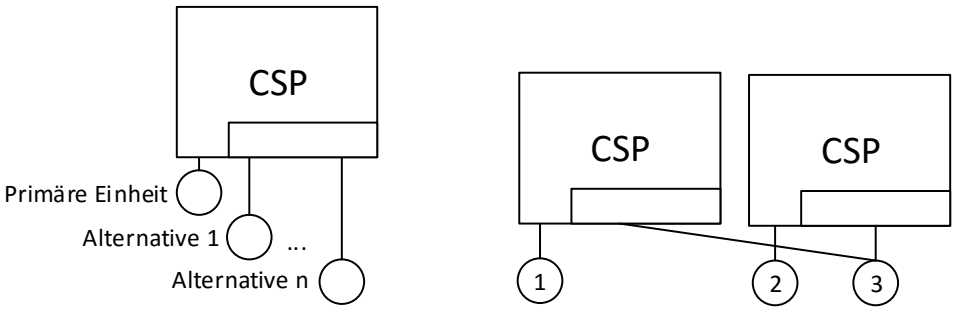


Abbildung 5.4: Funtional-Dependency Gate mit Trigger Event und abhängigen Ereignissen

Functional-Dependency Gate (FDEP) Bei dem FDEP Gate (Abbildung 5.4) führt ein bestimmtes Ereignis, das Trigger Event, zum Eintreten einer Anzahl an abhängigen Ereignissen.



(a) Gate mit mehreren Alternativen (b) Eine redundante Ressource für zwei Gates

Abbildung 5.5: Cold-Spare Gate mit primärem und redundanten Ereignissen

Cold-Spare Gate (CSP) Mit dem Cold-Spare Gate (Abbildung 5.5 (a)) lassen sich redundante Systeme beschreiben. Es verfügt über eine primäre Ressource und über weitere Alternativen, die bei einem Fehler der Haupteinheit für diese einspringen können. Mit diesem Gate können kalte, warme und heiße Redundanzlösungen dargestellt werden. Außerdem kann ein redundantes

Event als Alternative für mehrere primäre Einheiten dienen (siehe Abbildung 5.5 (b)). In diesem Beispiel kann Element „3" sowohl als Ersatz für „1" als auch für „2" genutzt werden. Es gilt aber zu beachten, dass es nicht beide primären Einheiten gleichzeitig ersetzen kann. Es dürfte also maximal eins der drei Events eintreten.

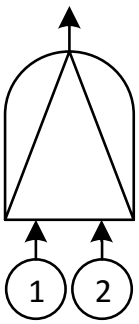


Abbildung 5.6: Priority-AND Gate mit zwei eingehenden Ereignissen

Priority-AND Gate (PAND) Zum Schalten eines PAND Gate (Abbildung 5.6) kommt es nicht nur wenn alle darunter liegenden Ereignisse oder Gates schalten, sondern auch nur dann, wenn diese in einer festgelegten Reihenfolge eintreten. In der Abbildung müsste also zuerst Event „1" und anschließend Event „2" einen Fehler produzieren. Eine andere Reihenfolge hätte keine Auswirkungen auf das Gate.

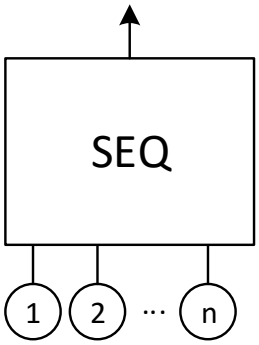


Abbildung 5.7: Sequence-Enforcing Gate mit eingehenden Ereignissen

Sequence-Enforcing Gate (SEQ) Genau wie bei dem PAND Gate hat ein SEQ Gate (Abbildung 5.7) eine Anzahl an Ereignissen, die alle in einer bestimmten Reihenfolge schalten müssen, damit das Gate schaltet. Der Unterschied besteht aber darin, dass die Events auch nur in dieser Reihenfolge auftreten können, während sie bei einem PAND Gate auch möglich ist, dass sie in zufälliger Reihenfolge eintreten.

5.2.2 Berechnung

Da die dynamischen Gates sich nicht durch einfache Formeln berechnen lassen, kann hier auf den Einsatz von Markov-Ketten zurückgegriffen werden. Mit ihrer Hilfe kann die Wahrscheinlichkeit berechnet werden, mit der das Gate schaltet und somit einen Fehler produziert. Es kann kein generell gültiger Aufbau einer Markov-Kette für ein bestimmtes Gate definiert werden, da sich die Ketten im Normalfall mehrere Gates gleichzeitig abbilden. Nachfolgend werden die Prozessdiagramme der Ketten an Beispielen für PAND und CSP Gates dargestellt.

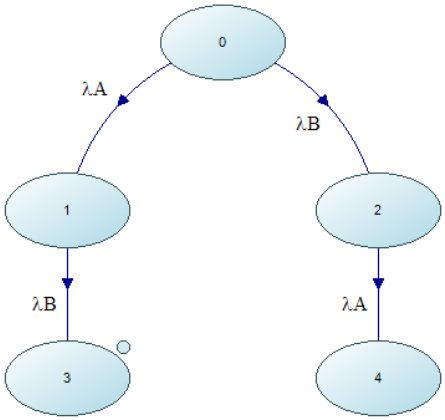
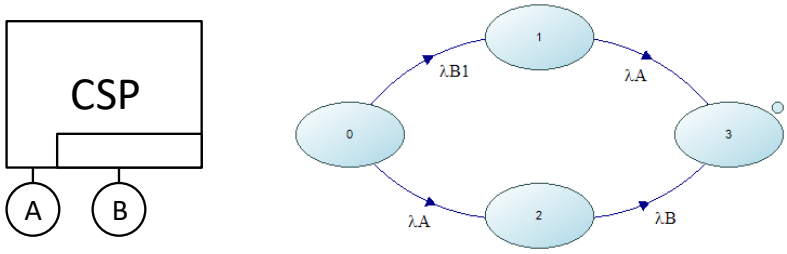


Abbildung 5.8: Prozessdiagramm zur Markov-Kette eines PAND Gates

PAND Abbildung 5.8 zeigt das Prozessdiagramm der Markov-Kette eines PAND Gates mit den beiden Ereignissen A und B. In diesem Fall muss erst das Ereignis A eintreten und danach das Ereignis B. Von dem Ausgangszustand „0" aus gelangt das Gate mit der Transition λA in den Zustand „1" bzw. mit der Transition λB in den Zustand „2". Von dem jeweils erreichten Zustand führt nun die entsprechend andere Transition in den nächsten Zustand. Wurde der Endzustand „3" erreicht, führt dies zum Schalten des Gates, da erst Ereignis A und anschließend

Ereignis B eingetreten ist. Befindet sich das System in einem der anderen Zustände, führt dies zu keinem Fehler.



(a) CSP-Gate (b) Prozessdiagramm zur Markov-Kette eines CSP Gates

Abbildung 5.9: CSP-Gate mit Prozessdiagramm

CSP Bei diesem CSP-Gate verfügt das Ereignis A über einen Spare B (siehe Abbildung 5.9 (a)). Bei der Redundanz handelt es sich um eine warme Redundanz, was bedeutet, dass Spare B im inaktiven Zustand eine geringere Ausfallwahrscheinlichkeit besitzt. In dem Prozessdiagramm (siehe Abbildung 5.9 (b)) steht die Transition λA für einen Fehler von Ereignis A. Für B existieren die zwei Transitionen λB1 und λB. Befindet sich der Spare im inaktiven Zustand gilt die Transition λB1 mit einer geringeren Ausfallwahrscheinlichkeit. Kommt es beim Ereignis A zu einem Fehler, schaltet die λA Transition und Spare B ist nun die aktive Komponente. Ab diesem Zeitpunkt gilt für den Spare nicht mehr λB1 sondern λB mit einer höheren Ausfallwahrscheinlichkeit. Sobald der Zustand „3" des Prozessdiagramms erreicht wird, handelt es sich um einen Fehler und das Gate schaltet.

Die dynamischen Elemente können mit Hilfe der Markov-Ketten analysiert werden. Somit können auch Systeme untersucht werden, die sich beispielsweise durch redundante Komponente während der Analyse verändern. Der Einsatz eines Markov-Modells führt aber auch zu anspruchsvolleren Berechnungen (vgl. Vesely u. a., 2002, S. 106) und selbst kleine Systeme können zu großen Markov-Ketten führen. Da die Ketten mit einer steigenden Anzahl an Ereignissen sehr stark wachsen, werden effektivere Methoden entwickelt um die dynamischen Elemente zu lösen (vgl. Rao u. a., 2009). Im Laufe dieser Arbeit werden weiterhin die Markov-Ketten verwendet, da die sich die dynamischen Elemente in den zu untersuchenden Bäumen in kleinen Subtrees befinden, die maximal 5 Ereignisse unter sich vereinen. Die daraus resultierenden Prozessdiagramme bleiben also auch überschaubar.

Bei dem Einsatz in dynamischen Fehlerbäumen kann der Aufwand für die Analyse verringert werden, indem die dynamischen Elemente des Baums separat berechnet und die Ergebnisse anschließend in einen klassischen Fehlerbaum übertragen werden (vgl. Vesely u. a., 2002, S. 107).

5.3 Entwicklung eines Fehlerbaums

Die Entwicklung eines Fehlerbaums kann in fünf Schritte unterteilt werden:

Festlegen des Top-Events: Das eindeutige Formulieren und das genaue Abstecken der Grenzen des Systems muss als erstes erfolgen. Hieraus ergibt sich dann das Top-Events.

Ereignisse ermitteln: Im nächsten Schritt gilt es die konkreten Ereignisse zu ermitteln, die zum Ausfall der Funktionen des Systems führen können. Diese können von Fehlern in der Hard- und Software über Leistungsgrenzen der verbauten Komponenten bis hin zu menschlichem Versagen reichen.

Konstruktion des Baums Die Konstruktion des Fehlerbaums erfolgt von oben nach unten. Nach dem Top-Event folgen die Funktionen, die bei einem Ausfall zum Versagen des Systems führen. Diese werden durch geeignete Gatter dargestellt und haben wiederum Eingaben, die zu ihrem Eintreten führen. Je nachdem wie tief die Analyse des Systems reichen soll, wird die Entwicklung der Funktionen so fortgeführt. Beispielsweise kann ein elektronisches Gerät einfach definiert werden in dem eine Wahrscheinlichkeit angegeben wird mit der es ausfällt oder der Baum wird weiter fortgesetzt bis die genauen Ausfallursachen ermittelt wurden. Die Tiefe der Analyse wird zusammen mit der Definition des Top-Events festgelegt.

Analyse des Baums Sobald die Konstruktion abgeschlossen ist, kann mir der Analyse begonnen werden. Mit den Berechnungsmethoden kann die gesamte Ausfallwahrscheinlichkeit des Systems ermittelt werden. Hier sind vor allem die Ereignisse und Ereigniskombinationen zu ermitteln, die einen hohen Einfluss auf das gesamte System haben. Verbesserungen bei diesen Ereignissen haben letztendlich das größte Potential zur Verbesserung des untersuchten Top-Events beizutragen.

Erweitern des Systems Im Laufe der Entwicklung eines Systems fallen regelmäßig Änderungen und Anpassungen an. Der Fehlerbaum muss hier ständig auf dem aktuellen Stand gehalten werden um immer einen Überblick über das System zu haben. Vor allem nachdem

Probleme durch die Analyse identifiziert werden konnten und eine Lösung entwickelt wurde, kann so der Effekt der Anpassungen ermittelt werden.

5.4 Toolgestützte Generierung und Analyse

Es gibt einige Softwarelösungen zum Analysieren von Fehlerbäumen. Eine von ihnen ist in das Softwarepaket „Reliability Workbench" (RWB) von Isograph integriert. Die Software bietet eine Vielzahl an Funktionen bezüglich Zuverlässigkeit und Sicherheit. Eine davon ist das Modul FaultTree+, welches die Möglichkeit bietet sowohl statische als auch dynamische Elemente in Fehlerbäumen zu untersuchen.

Soll ein System analysiert werden, kann in der RWB ein Projekt angelegt werden. Für ein solches Projekt können mehrere Fehlerbäume, Ereignisse, Fehlermodelle und Markov-Ketten angelegt werden. Die Events können ihren entsprechenden Gates zugeordnet werden und entweder direkt mit einer Ausfallwahrscheinlichkeit konfiguriert werden oder ihnen wird ein Fehlermodell zugeordnet. Fehlermodelle bieten sich an, wenn es im System mehrere Einheiten des gleichen Typs mit den gleichen Wahrscheinlichkeiten gibt. So muss nicht jedes Ereignis einzeln konfiguriert werden und etwaige Änderungen können schnell an allen Einheiten gleichzeitig vorgenommen werden.

Die gleichen Events können auch in unterschiedlichen Fehlerbäumen eingesetzt werden. Soll das gleiche System in unterschiedlichen Varianten getestet werden, reicht es die Stellen im Baum auszutauschen, in denen sich die Lösungen unterscheiden. Werden Änderungen vorgenommen, die alle Lösungen gleichermaßen betreffen, muss auch hier nicht jeder Fehlerbaum einzeln angepasst werden.

Die Gatter der dynamischen Fehlerbäume kann der Anwender mit Markov-Ketten realisieren. Hier können die unterschiedlichen Zustände mit Transitionen verbunden werden, die mit gewissen Wahrscheinlichkeiten zum nächsten Zustand führen. Die Ergebnisse der Markov-Ketten können ebenfalls als Fehlermodell für die Ereignisse im Fehlerbaum genutzt werden.

Die fertigen Fehlerbäume werden von der Software automatisch berechnet. Sie liefert nicht nur die Eintrittswahrscheinlichkeit des Top-Ereignisses, sondern auch die jedes einzelnen Gates. Hinzu kommen noch die Minimal Cut Sets inklusive deren Wahrscheinlichkeiten.

6 Fehlerklassen in Fehlerbäumen

Während der Kommunikation in einem Netzwerk können eine Vielzahl an Fehlern auftreten. Diese können ganz unterschiedlicher Natur sein. Hardwareelemente können ausfallen, fehlerhafte Software liefert falsche Ergebnisse oder äußere Einflüsse beeinträchtigen wichtige Funktionen. Wird ein Kommunikationsnetzwerk in einem Fehlerbaum abgebildet, gilt es möglichst alle Gefahren und Risiken zu identifizieren und diese in den Baum einzubringen.

Je nach Netzwerktopologie und der verwendeten Technologien existieren unterschiedliche Fehlerquellen. Beispielsweise entstehen, wie schon im Kapitel 4 erwähnt, durch redundante Komponenten im Netzwerk zusätzliche Fehlerquellen. Dies kann der Umgang mit Replikaten, welche über zwei verschiedene Pfade gesendet wurden, sein oder bei einer dynamischen Redundanz das Umschalten auf die Standby-Komponente.

Je nach eingesetzter Kommunikationstechnologie spielen zudem unterschiedliche Fehlerquellen eine Rolle. Während in einem Standard-Ethernet-Netzwerk ein Burst wichtige Nachrichten entscheidend verzögern könnte, hätten die Nachrichten auf eine echtzeitfähige TT-Übertragung keinerlei Einfluss. In einem TTE-Netzwerk könnte es im Gegensatz zu Problemen bei der Uhrensynchronisation kommen.

Bei dem Design eines sicherheitskritischen Systems, wird in der ISO 26262 während der Konzeptphase gefordert, dass Gefährdungen und Risiken betrachtet werden (siehe Abschnitt 3.3.2). Dies sollte stets mit Experten auf den jeweiligen Gebieten geschehen um das System möglichst vollständig zu erfassen.

Dieses Kapitel umfasst einige grundlegende Fehlerszenarien, um einen Überblick über mögliche Fehler zu schaffen. Bei dem Design eines realen Systems muss an dieser Stelle angesetzt und die jeweiligen Szenarien noch weiter vertieft werden.

6.1 Allgemeine Darstellung von Fehlern

Viele Fehler, wie z. B. defekte Komponenten oder die Gefahr einer fehlerhaften Übertragung, lassen sich in einem Fehlerbaum in der Regel durch einfache Ereignisse darstellen. Sie lassen sich durch eine bestimmte Eintrittswahrscheinlichkeit definieren und fließen somit in die

Berechnung der Ausfallwahrscheinlichkeit des Systems ein. Bei komplexeren Fehlerszenarien wie beispielsweise Abhängigkeiten zwischen den Ereignissen oder Fehler auf die das System in irgendeiner Weise reagiert, sind eventuell dynamische Fehlerbaumelemente nötig, um diese in dem Baum abzubilden.

6.2 Fehlerszenarien

Die Fehlerszenarien, welche in der kommenden Analyse der Redundanzkonzepte eingesetzt werden, sind in vier Bereiche unterteilt:

- Hardwarefehler
- Fehler durch Redundanz
- Fehler durch Kommunikationstechnologie
- Netzwerkfehler

6.2.1 Hardwarefehler

Unter diesen Punkt fallen die Ausfälle der einzelnen Netzwerkkomponenten. Konkret sind das Ausfälle der ECUs und Switches sowie defekte Kabelverbindungen zwischen den Netzwerkteilnehmern. Weiterhin werden in diesem Bereich des Fehlerbaums redundante Mittel im Netzwerk untergebracht. Wie diese genau integriert werden, hängt von der Art der Redundanz ab. Während redundante Pfade unter einem AND-Gate zusammengeführt werden können, ist für Elemente mit dynamische Redundanz ein CSP-Gate vonnöten.

6.2.2 Fehler durch Redundanz

Wie schon im Kapitel 4 erwähnt wurde, kommen durch den Einsatz redundanter Mittel auch neue Risiken und Fehlerquellen in ein Netzwerk. Existieren in einem System beispielsweise redundante Sensoren, ist es gut möglich, dass diese leicht unterschiedliche Ergebnisse liefern, da Messungen zu unterschiedlichen Zeitpunkten durchgeführt werden (vgl. Storey, 1996, S.133). Das Problem des Identifizierens und der Bearbeitung duplizierter Nachrichten wurde bereits behandelt und wird auch in den Fehlerbäumen der Redundanzkonzepte berücksichtigt. In Spitzer (2006), S. 4-14, werden noch weitere Probleme redundanter Systeme genannt:

Abhängigkeiten zwischen redundanten Elementen bestehen beispielsweise, wenn sie mit der selben Stromversorgung verbunden sind oder die gleiche fehlerhafte Hardware verbaut wurde. Tritt ein solcher Fall auf, kann durch die Redundanz keine zusätzliche Sicherheit erreicht werden.

Load sharing failure sind ein Problem, wenn Aufgaben einer defekten Komponente von anderen zusätzlich übernommen werden. Als Beispiel wird hier ein ausgefallener Motor eines Flugzeugs aufgeführt, wodurch die anderen Motoren mehr leisten müssen und somit stärker beansprucht werden.

Erhöhter Stromverbrauch muss berücksichtigt werden, wenn redundante Hardware zum Einsatz kommt.

6.2.3 Fehler durch Kommunikationstechnologie

Je nach Kommunikationstechnologie, die zum Einsatz kommt, sind unterschiedliche Fehlerszenarien möglich. Kommt eine Technologie zum Einsatz, die mit Hilfe einer synchronen Zeitbasis den kritischen Datenverkehr organisiert, wie z. B. TTE oder FlexRay, können Probleme bei der Uhrensynchronisation die Kommunikation so beeinflussen, dass kritischer Datenverkehr nicht rechtzeitig am Ziel eintrifft.

Bei einem Netzwerk ohne echtzeitfähigen Datenverkehr spielen hingegen Probleme eine Rolle, die bei TTE keinen Einfluss haben. Wird das Netzwerk beispielsweise von einem *Babbling Idiot* mit Nachrichten überflutet, kann dieser ein komplettes Ethernet-LAN negativ beeinflussen, indem Nachrichten stark verzögert werden oder durch überfüllte Nachrichtenbuffer in einem Switch ganz verloren gehen. Der höher priorisierte Datenverkehr bei TTE würde davon wiederum nicht behindert werden.

6.2.4 Netzwerkfehler

Paketverlust Geht ein Paket verloren, kann dies eventuell zu Problemen führen. Ein solcher Fehler wird im Fehlerbaum als einfaches nicht permanentes Ereignis dargestellt. In Bezug auf dynamische Fehlerbäume sind solche Fehler interessant, wenn Abhängigkeiten zwischen mehreren solcher Ereignisse bestehen.

Verfälschte Daten Die Nachricht wird zwar übertragen aber der Inhalt der Daten ist verfälscht. Wie auch beim Paketverlust kann dieser Fehler durch ein einfaches nicht permanentes Ereignis abgebildet werden. Auch hier werden die dynamischen Elemente relevant, sobald

Abhängigkeiten bestehen. Zusätzlich könnte bei redundanten Nachrichten aber auch das richtige Ergebnis von einer anderen Quelle beim Empfänger ankommen. Dies würde ein weiteres Szenario darstellen, das mit Fehlerbäumen untersucht werden kann.

Bursts Ein Burst liegt dann vor, wenn innerhalb kürzester Zeit eine große Menge an Daten direkt hintereinander über das Netzwerk verschickt werden. Dies kann dazu führen, dass andere Nachrichten verzögert werden.

Äußere Einflüsse

> „Many safety-critical systems are required to function in environments that would be described as 'hostile'." (Storey, 1996, S. 200)

Zu dieser Aussage werden dies Umgebungen zusätzlich noch charakterisiert:

- Temperaturschwankungen
- Feuchtigkeit
- Vibration
- Elektromagnetische Interferenzen

All diesen Bedingungen ist ein Auto ausgesetzt und da sie das System zusätzlich beanspruchen, werden sie ebenfalls in den Fehlerbäumen berücksichtigt.

7 Konstruktion von Redundanzkonzepten

Dieses Kapitel befasst sich mit der Konstruktion unterschiedlicher Redundanzkonzepte für ein bereits vorhandenes nicht redundantes Netzwerk. Das Netzwerk basiert auf dem eines BMW-Serienautos und wird in Abschnitt 7.1 beschrieben. In Abschnitt 7.2 werden die unterschiedlichen Konzepte vorgestellt und im letzten Abschnitt die resultierenden Fehlerbäumen abgebildet.

7.1 Netzwerkszenario

Als Grundlage für das Autonetz dient ein aus einem Serienauto von BMW abgeleitetes Netzwerk. Dieses wurde im Zusammenhang mit den wissenschaftlichen Arbeiten Steinbach u. a. (2012) und Steinbach u. a. (2015) entwickelt und für den Einsatz mit Time-Triggered Ethernet und AVB untersucht. Hierzu wurde ein Teil des ursprünglichen Netzwerks, bestehend aus Kommunikationstechnologien wie MOST, CAN oder FlexRay, durch die Echtzeit-Ethernet Technologien ersetzt.

Die meiste Bandbreite wird durch Multimediastreams verursacht. Gleichzeitig handelt es sich hierbei aber auch um die am niedrigsten priorisierten Nachrichten. Er besteht aus Video-, Audio- und TV-Daten. Mit etwas weniger Bandbreite aber einer höheren Priorität folgen die Kameradaten für die Fahrassistenzsysteme. Zudem werden noch hoch priorisierte Steuerdaten über das Netzwerk übertragen. Im Vergleich zu den anderen Daten zeichnen diese sich durch eine geringe Bandbreite und verhältnismäßig lange Zykluszeiten von fünf Millisekunden bis zu einer Sekunde aus.

Das neue Netzwerk besteht aus sieben Switches und 15 Hosts (siehe Abbildung 7.1). Diese sind in einer Baumstruktur angelegt und zu übertragende Daten müssen maximal vier Hops überwinden um zum Ziel zu gelangen.

7.1.1 Prioritäten der Nachrichten

In den wissenschaftlichen Arbeiten wurden den unterschiedlichen Nachrichtentypen Prioritäten für die Übertragung zugeordnet.

Bei der AVB-Konfiguration werden sowohl die Steuernachrichten, als auch die Kameradaten für die Fahrassistenzsysteme, in die Paketklasse A eingeteilt. Alle Multimediastreams werden unter der Klasse B versendet.

In einem TTE-Netzwerk werden 30 der 42 Steuerdatennachrichtenflüsse als Time-Triggered Nachrichten verschickt, die restlichen zwölf als Rate Constrained Nachrichten. Die Daten der Kameras sowie von den Multimediaanwendungen sind ebenfalls in die RC Klasse eingeteilt.

In Steinbach u. a. (2015) enthält das Netzwerk zusätzlich noch Best-Effort Cross-traffic, um die Auswirkungen des zusätzlichen Datenverkehrs auf AVB und TTE zu untersuchen.

7.1.2 Nachrichtenflüsse

Innerhalb des Netzwerks werden die Daten zwischen festgelegten Kommunikationspartnern ausgetauscht. Das heißt es kommt nicht vor, dass jeder Knoten mit jedem anderen Knoten Daten austauscht. Das hat zur Folge, dass es Bereiche im Netzwerk gibt, deren Anforderungen an die Ausfallsicherheit höher sind und andere Bereiche, die in diesem Aspekt vernachlässigt werden können.

Einige der Netzwerkteilnehmer sind beispielsweise ausschließlich für Aufgaben im Entertainmentbereich zuständig. So besteht zwischen den Elementen *TV*, *MM_Disk*, *Rear Seat Entertainment (RSE)* und *Audio_AMP* ein hoher Bedarf an Bandbreite um Video- und Audiodaten übertragen zu können. Gleichzeitig ist es für diese Anwendungen aber auch wichtig, dass die Daten rechtzeitig und zuverlässig beim Empfänger ankommen um eine einwandfreie Wiedergabe zu ermöglichen. Es handelt sich hierbei aber nicht um sicherheitskritische Anwendungen, da ein Ausfall oder eine vorübergehende Störung keinerlei Gefahren darstellen.

Anders sieht es bei Steuerdaten aus, die beispielsweise zwischen der *HeadUnit* und *DME1* ausgetauscht werden. Je nachdem um welche Daten es sich genau handelt, kann eine Fehlfunktion im schlimmsten Fall einen Unfall zur Folge haben und dabei das Leben der Beteiligten gefährden. Auch der Datenaustausch zwischen Fahrassistenzkameras (*DA_CAM*) und dem *Display1* kann als kritisch betrachtet werden, da eine Bildstörung im falschen Moment ebenfalls zu einem Unfall führen könnte. Diese Bereiche gilt es zu untersuchen und gegebenenfalls an die entsprechenden Anforderungen anzupassen.

7.2 Redundanzkonzepte

Für das zuvor beschriebene Szenario wurden zwei Redundanzkonzepte entwickelt. Im Ersten wurde die Topologie dahingehend erweitert, dass ein zusätzlicher Kommunikationspfad eine

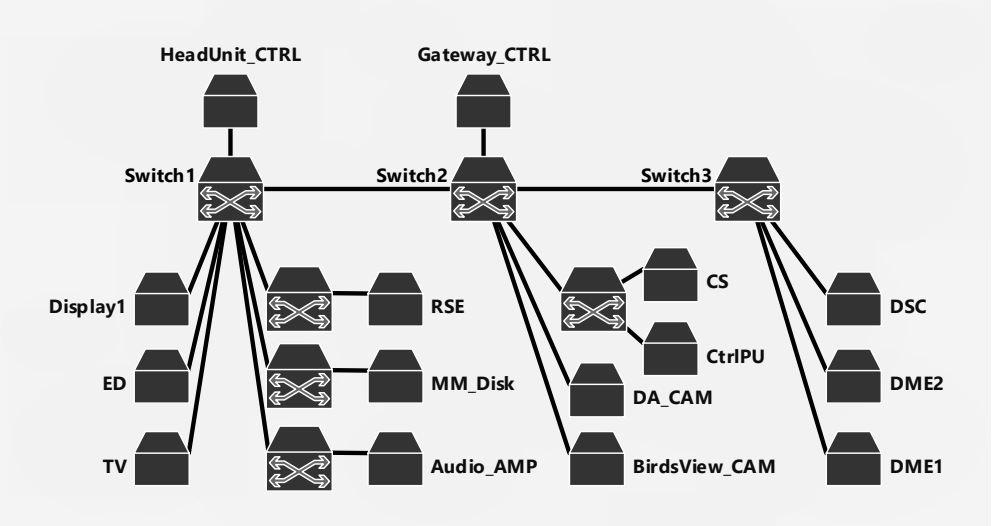


Abbildung 7.1: Baumstruktur des BMW-Netzes

alternative Route für die Daten zu Verfügung stellt. Im zweiten Konzept wurde ein Teil der Hardware redundant ins Netz eingebunden.

In diesen Konzepten liegt der Fokus auf der Kommunikation zwischen *DA_CAM* und *Display1*. Bei dem Austausch der Daten für die Fahrassistenzkameras handelt es sich um kritischen Datenverkehr und soll deswegen zusätzlich abgesichert werden. Es gibt in dem Netzwerk noch einige andere wichtige Kommunikationspfade zwischen anderen Knoten, die in einem fertigen Netzwerk auch mit redundanten Mitteln berücksichtigt werden müssen. Dies würde zu vielen komplexen Fehlerbäumen führen, die nicht im Rahmen dieser Arbeit betrachtet werden. Die Kommunikation zwischen den zuvor genannten Komponenten dient als anschauliches Beispiel für die Fehlerbaumanalyse.

7.2.1 Ringtopologie

Die Intention bei diesem Konzept ist durch möglichst geringen Aufwand und einem einfachen Aufbau des Netzes Redundanz zu schaffen und so das System vor Ausfällen und Fehlern zu schützen. Es orientiert sich an den im Abschnitt 4.2.1 vorgestellten Protokollen MRP und HSR. Die Switche des Netzwerks werden zu einem Ring verbunden und geben dem Knoten *DA_CAM* so die Möglichkeit seine Daten über zwei Pfade an das Display zu senden. Zusätzlich würde dieses Konzept es auch anderen Knoten ermöglichen ihre sicherheitskritischen Daten redundant zu versenden.

Sogar durch so geringe Änderungen kommen auf das System einige neue Herausforderungen zu. Zum einen wird das Netzwerk durch die zusätzlichen Daten stärker beansprucht. Die Hardware muss dazu in der Lage sein die Daten richtig über die unterschiedlichen Pfade zu verschicken und die duplizierten Nachrichten müssen an der richtigen Stelle identifiziert und bearbeitet werden. Zum Beispiel könnten die Duplikate bereits bei dem Switch vor dem Display erkannt und genau wie beim HSR eliminiert werden. Andererseits wäre es denkbar diese Aufgabe dem Empfänger zu überlassen um diesem zu ermöglichen zusätzliche Informationen aus den Daten zu gewinnen. Durch die neuen Funktionen und die zusätzliche Belastung des Netzwerks kommen neue Fehlerquellen in das System, die es gilt in die Fehlerbäume zu integrieren.

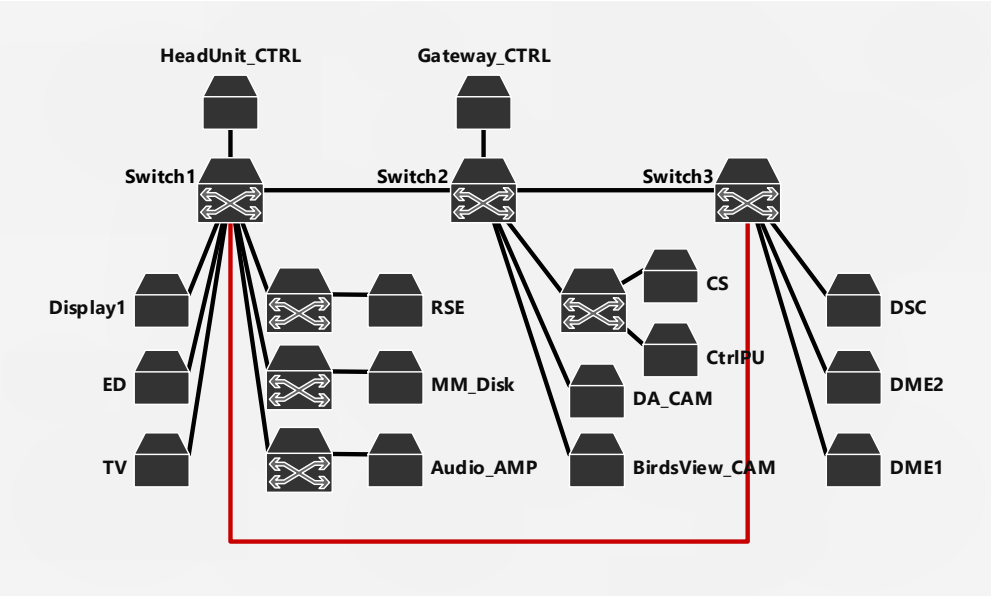


Abbildung 7.2: Baumstruktur des BMW-Netzes mit einem redundanten Pfad

7.2.2 Redundante Hardware

Das zweite Konzept sieht vor die Kommunikation durch zusätzliche Hardware abzusichern (siehe Abbildung 7.3). Neben einem zusätzlichen Switch zwischen *DA_CAM* und *Display1* gibt es für *DA_CAM* eine dynamische Redundanzlösung (siehe Abbildung 7.4). Dieses Konzept ermöglicht es die Zahl der SPOFs deutlich zu verringern und so eine gute Grundlage für einen fehlerfreien Datenaustausch zu schaffen. Durch die Kommunikation über *Switch1_1* werden die duplizierten Nachrichten über einen vom Rest des Netzwerks unabhängigen Teil gesendet. Dieses Prinzip baut auf dem im Abschnitt 4.2.1 vorgestellten PRP auf.

Der nun redundante Knoten *DA_CAM* verfügt über eine zusätzliche ECU, die sich im Standby-Modus befindet und bei Bedarf die Aufgabe des Knotens übernehmen kann. Der weitere Aufbau gleicht dem im Abschnitt 4.2.2 unter dem Punkt „Dynamische Redundanz" vorgestellten Beispiel und umfasst somit eine Komponente zur Fehlererkennung sowie einem Modul, das das Umschalten von einer ECU auf die andere übernimmt. Der einzige Unterschied ist, dass es zwei Ausgabeports gibt, über welche die Daten an beide Switches ausgeteilt werden können. Da die Kosten bei einem Auto möglichst gering gehalten werden müssen, stellt diese Alternative der Hardwareredundanz eine kostengünstige und gewichtsparende Methode, im Vergleich zu beispielsweise einer TMR, dar.

Die genaue Funktionsweise der einzelnen Module gestaltet sich so, dass im fehlerfreien Zustand *ECU1* seine Ergebnisse an die Fehlererkennung und an den internen Switch sendet. Der Switch übernimmt die Weiterleitung der Frames über seine beiden Ports an das Netzwerk. Die Fehlererkennung versucht unterdessen Fehler zu identifizieren um in einem solchen Fall das Umschalten auf *ECU2* zu veranlassen. Ist dieser Fall eingetreten, übernimmt diese die Aufgaben der ersten ECU. Weiterhin ist dieser Knoten so konzipiert, dass auch bei einem Ausfall der Fehlererkennung oder beim Versagen des Umschaltmechanismus Daten an das Netzwerk übermittelt werden. Sollte also *ECU1* noch laufen und dann die Fehlererkennung oder der Switch versagen, befindet sich der Knoten noch in einem funktionsfähigen Zustand. Wenn anschließend *ECU1* einen Fehler hat, kann nicht mehr auf die redundante Komponente umgeschaltet werden.

Mit dem Einsatz der dynamischen Redundanz und des zusätzlichen Pfades gehen auch bei diesem Konzept neue Probleme und Anforderungen einher. Fallen die Fehlererkennung oder der Switch in *DA_CAM* aus, kann die redundante ECU nicht mehr genutzt werden und es fallen in einem Fehlerfall Umschaltzeiten an, welche die Übertragung der Daten verzögern können. Zusätzlich müssen, wie schon bei der Ringtopologie, Duplikate erkannt und bearbeitet werden. Zudem gilt zu beachten, dass der Einsatz redundanter Hardware weitere Kosten verursacht.

7.3 Fehlerbäume

Die Fehlerbäume der drei Netzwerke können grob in zwei Bereiche aufgeteilt werden. Der erste beinhaltet die Hardwarefehler. Dazu zählen die Ausfälle von den ECUs, den Switches sowie den Verbindungen zwischen diesen. Hier werden zudem die redundanten Elemente eingebunden. Im zweiten Bereich der Bäume sind die jeweiligen Fehlerszenarien angesiedelt, wie sie in Kapitel 6 vorgestellt wurden.

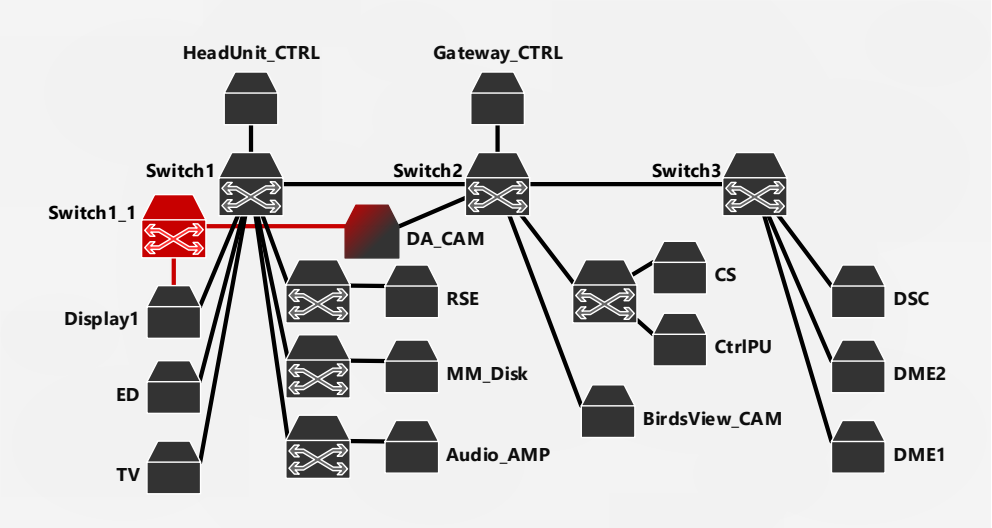


Abbildung 7.3: Baumstruktur des BMW-Netzes mit mehreren redundanten Hardwareelementen

Der folgende Abschnitt ist so aufgeteilt, dass die Fehlerbäume erst im Bezug auf ihre Hardwarefehler behandelt werden und anschließend die Bereiche der weiteren Fehlerszenarien, da die Bäume in diesem Bereich alle sehr ähnlich sind. Sämtliche Hardwarefehler sind so definiert, dass die entsprechende Komponente nicht mehr funktionsfähig ist und eine Reparatur erforderlich macht. Vorrübergehende Fehler oder Störungen befinden sich im anderen Teil des Fehlerbaums.

7.3.1 Ursprüngliches Netzwerk

Da das ursprüngliche Netzwerk über keinerlei Redundanz verfügt, handelt es sich um einen sehr flachen Baum (siehe Abbildung 7.5). Das liegt daran, dass jede Komponente auf dem Pfad von *DA_CAM* zu *Display1* ein SPOF ist und beim Ausfall eines Elements die Kommunikation sofort zum erliegen kommt.

Das Top-Event *FTA_ORIGINAL* ist also ein OR-Gate unter dem sich direkt alle Elemente des Pfades befinden:

- Die ECUs *DA_CAM* und *Display1*
- Die Switches *Switch1* und *Switch2*
- Die drei Verbindungen zwischen den ECUs und den Switches

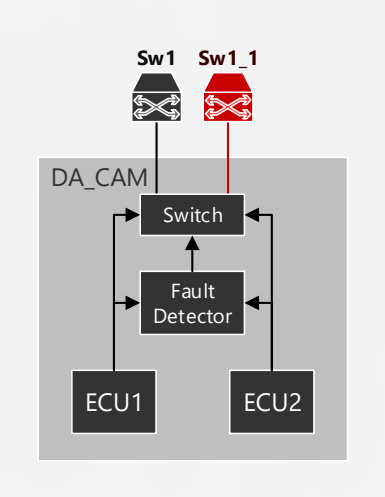


Abbildung 7.4: Redundanter Knoten DA_CAM

7.3.2 Ring Topologie

Die zusätzliche Leitung zwischen *Switch1* und *Switch3* hat zur Folge, dass die Daten über zwei Pfade an ihr Ziel gelangen können. Im Fehlerbaum macht sich dieser Umstand dadurch bemerkbar, indem sich die Ereignisse für die zwei Pfade unter dem AND-Gate *RED_RING* befinden (siehe Abbildung 7.6). Die anderen Ereignisse für die ECUs, die Switches und die Verbindungen von den ECUs zu den Switches befinden sich nach wie vor unter dem OR-Gate des Top-Events.

Ab dem zweiten Switch können die Daten entweder die direkte Verbindung zu *Switch1* nehmen oder den redundanten Pfad über *Switch3* nutzen. Im Fehlerbaum kann der kurze Weg einfach über das entsprechende Ereignis *SW1<->SW2* abgebildet werden, während für den längeren Weg ein weiteres OR-Gate eingefügt wurde, das die Ereignisse der Hardwarekomponenten zusammenfasst (siehe Abbildung 7.7).

7.3.3 Redundante Hardware

In dem zu diesem Netzwerk zugehörigen Fehlerbaum befindet sich lediglich das Display direkt unter dem Top-Event. Alle anderen Elemente sind in irgendeiner Weise redundant. Die komplette Kommunikation von *DA_CAM* zu *Display1* erfolgt über separate Pfade. Erst bei dem Display treffen die duplizierten Nachrichten wieder aufeinander. Im Fehlerbaum sind die Pfade, wie schon bei der Ringtopologie, unter einem AND-Gate voneinander getrennt. Der erste Pfad

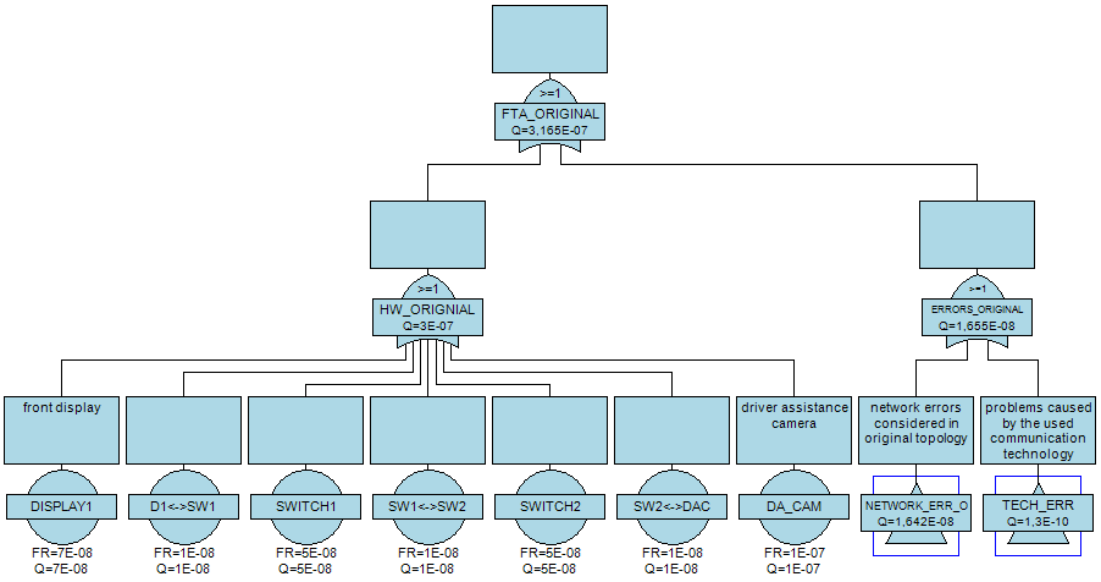


Abbildung 7.5: Fehlerbaum für das ursprüngliche Netzwerk

enthält die Ereignisse des ursprünglichen Pfades (über *Switch1* und *Switch2*) und der zweite Pfad die über den neuen *Switch1_1*.

Der redundante Knoten *DA_CAM* wird durch eine Markov-Kette im Baum abgebildet, da für die Modellierung des Knotens dynamische Gates zum Einsatz kommen. Dieser Subtree ist in Abbildung 7.9 zu sehen. Die beiden redundanten Elemente des Knotens *ECU1* und *ECU2* sind an einem CSP-Gate angeschlossen. So springt die zweite ECU ein sobald die Erste ausfällt. Da ein Fehler in dem Modul zur Fehlererkennung (*FE*) oder des internen Switches (*SW*) nicht zur Folge hat, dass gleich der komplette Knoten ausfällt, wird dieses Verhalten mit Hilfe eines PAND-Gates gelöst. Damit dieses Gate schaltet muss erst *FE* oder *SW* ausfallen (als OR-Gate in dem Baum integriert) und danach *ECU1*, da der Knoten in diesem Fall nicht mehr auf die noch funktionierende *ECU2* umschalten kann.

Der komplette Subtree muss in einer Markov-Kette abgebildet werden, da sich mit *ECU1* unter beiden Gates das selbe Ereignis befindet und sie somit voneinander abhängig sind. Diese Markov-Kette ist in Abbildung 7.10 dargestellt. Die Transitionen sind wie folgt definiert:

- $\lambda 1$: *ECU1* fällt aus
- $\lambda 2$: *ECU2* fällt aus
- $\lambda 3$: *FE* oder *SW* fallen aus

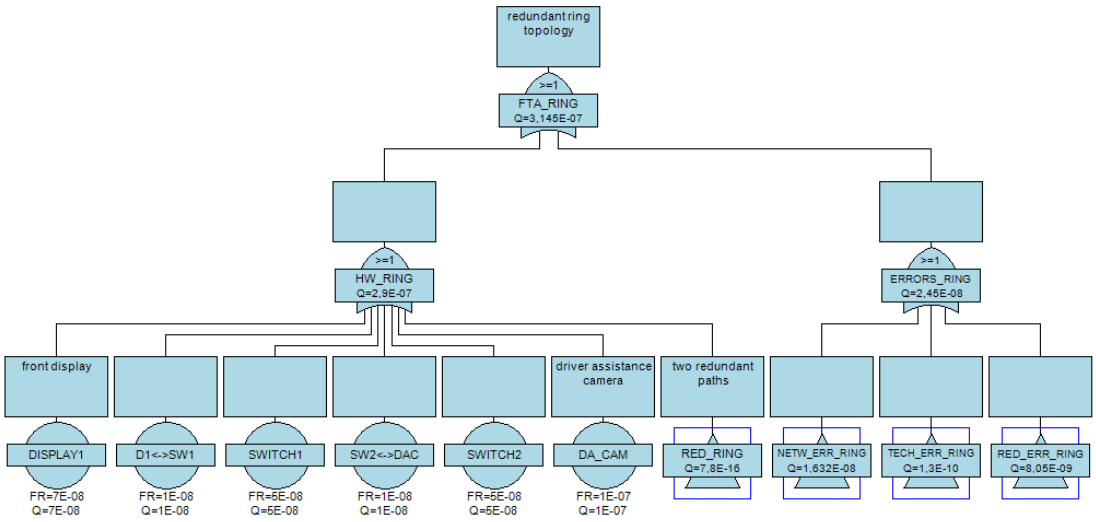


Abbildung 7.6: Fehlerbaum für die Ringtopologie

Das System startet im Zustand „0". Von hier aus führen die drei Transitionen in die Zustände „1", „2" und „3".

Zustand „1" bedeutet, dass die erste ECU ausgefallen ist und die Aufgaben von *ECU2* übernommen wurde. Die Transitionen $\lambda 2$ und $\lambda 3$ führen in die Zustände „4" bzw. „5". Zustand „4" bedeutet, dass beide ECUs ausgefallen sind und der komplette Knoten ausfällt. Im Zustand „5" funktioniert entweder die Fehlererkennung oder das Umschalten zwischen den redundanten Elementen nicht mehr. Da *ECU2* aber weiterhin seine Arbeit verrichten kann, ist das System noch funktionstüchtig. Erst mit der Transition $\lambda 2$ geht auch dieser Pfad der Kette in einen fehlerhaften Endzustand „6" über.

Zustand „2" bedeutet, dass die zweite ECU ausgefallen ist und die Aufgaben weiterhin von *ECU1* ausgeführt werden. Die Transitionen $\lambda 1$ und $\lambda 3$ führen in die Zustände „7" bzw. „8". Zustand „7" bedeutet, dass nun zusätzlich die erste ECU ausgefallen ist und somit der komplette Knoten ausfällt. Im Zustand „8" funktioniert entweder die Fehlererkennung oder das Umschalten zwischen den redundanten Elementen nicht mehr. Fällt nun auch *ECU1* mit der Transition $\lambda 1$ aus, geht das System in den fehlerhaften Endzustand „9" über.

Zustand „3" bedeutet, dass das Umschalten von *ECU1* auf *ECU2* nicht mehr möglich ist. Die Transitionen $\lambda 1$ und $\lambda 2$ führen in die Zustände „10" bzw. „11". Zustand „10" bedeutet, dass die erste ECU ausgefallen ist und da das Umschalten bzw. das Erkennen des Fehlers nicht mehr

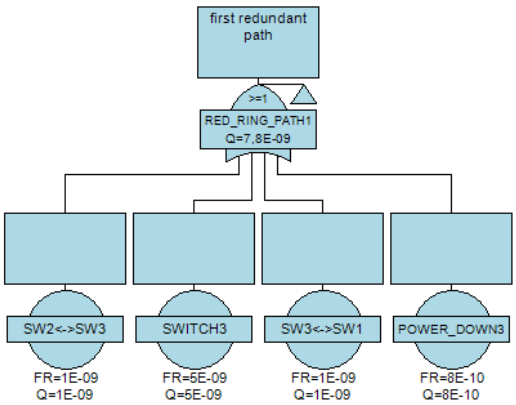


Abbildung 7.7: Subtree für einen redundanten Pfad

funktioniert, endet die Kette in diesem fehlerhaften Endzustand. Im Zustand „11" ist *ECU2* ausgefallen. Da *ECU1* aber noch funktionstüchtig ist, ist das System noch in einem sicheren Zustand. Schaltet von hier aus aber die Transition $\lambda 1$, fällt auch das letzte funktionierende Modul des Knotens aus und der fehlerhafte Endzustand „12" wird erreicht.

7.3.4 Subtrees für Fehlermodelle

Der Bereich für die unterschiedlichen Fehlermodelle ist bei allen drei Fehlerbäumen identisch aufgebaut. Es existiert für jede Fehlerkategorie ein Subtree, welcher die Ereignisse zu den unterschiedlichen Fehlerszenarien enthält. Alle Fehler können als einfache Ereignisse in die Bäume integriert werden. Wie in Kapitel 6 beschrieben, handelt es sich bei den Kategorien um:

- Fehler im Netzwerk
- Fehler durch Redundanz (außer beim ursprünglichen Netzwerk)
- Fehler durch die verwendete Technologie (in diesem Fall TTE)

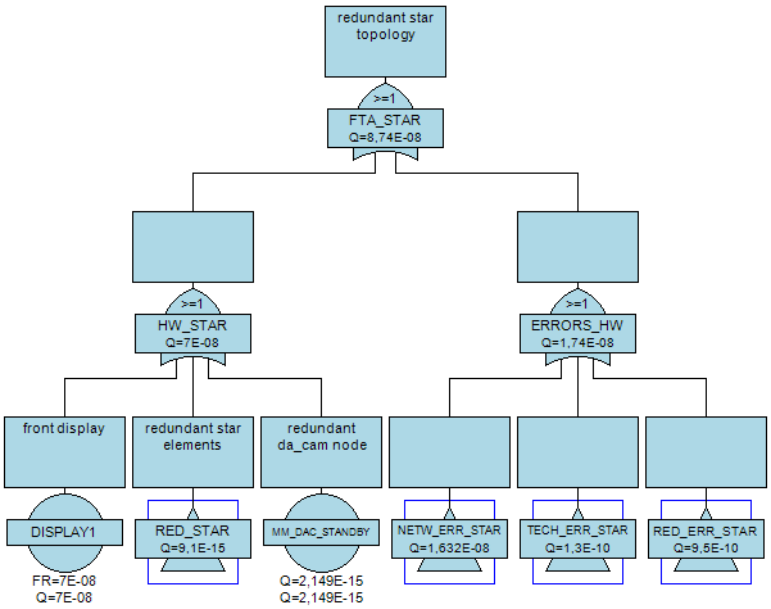


Abbildung 7.8: Fehlerbaum für das Konzept mit redundanter Hardware

Fehler im Netzwerk

Dieser Subtree gestaltet sich bei allen drei Bäumen fast identisch, die berücksichtigten Fehler sind:

- Stromausfall
- Falsche Daten
- Verlorene Pakete
  - Unterschied zwischen ursprünglichen Netzwerk und Redundanzkonzepten
  - Es wird davon ausgegangen, dass Pakete über die redundanten Pfade seltener verloren gehen.
- Äußere Einflüsse

Fehler durch Redundanz

In diesem Subtree können in beiden Konzepten duplizierte Frames zu Problemen führen. Ansonsten verfügen die Fehlerbäume in diesem Bereich über unterschiedliche Ereignisse. Bei der Ringtopologie wird davon ausgegangen, dass an den beiden Switches ein höheres

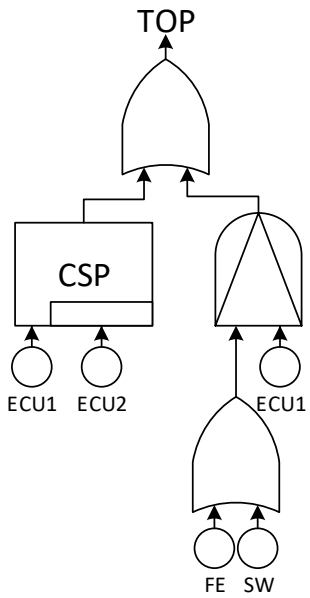


Abbildung 7.9: Dynamische Fehlerbaumelemente vom Knoten *DA_CAM*

Fehlerpotential besteht, da hier die Nachrichten in beide Richtungen geschickt bzw. aus beiden Richtungen empfangen werden. Bei der redundanten Hardware hingegen führt die Redundanz zu einem erhöhten Stromverbrauch, der bei starken Belastungen problematisch sein kann. Zudem besteht die Gefahr, dass der Umschaltvorgang von *ECU1* auf *ECU2* fehlschlägt und somit zum Ausfall führen kann.

Fehler durch Technologie

Da bei dem Einsatz von TTE die synchrone Zeitbasis ein entscheidender Faktor ist, fließt dieser Umstand in alle Subtrees der drei Konzepte ein. Zusätzlich kann es in seltenen Fällen dazu kommen, dass die RC-Nachrichten nicht rechtzeitig beim *Display1* ankommen.

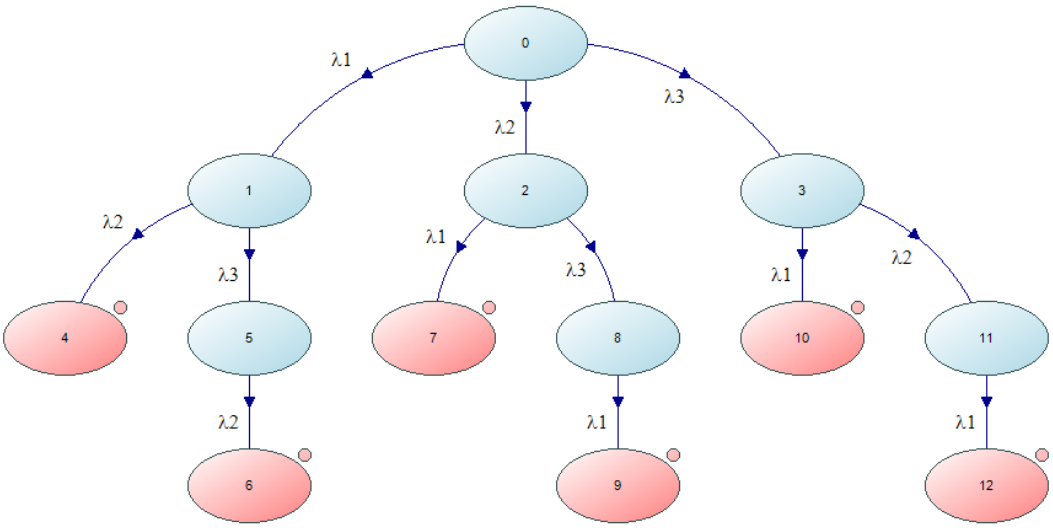


Abbildung 7.10: Markov-Kette für das Standby-Element

8 Analyse der Redundanzkonzepte

Die im vorherigen Kapitel erstellten Fehlerbäume werden in diesem Kapitel analysiert und miteinander verglichen. Außerdem werden die Ausfallwahrscheinlichkeiten bestimmter Komponenten angepasst, um zu untersuchen, wie die verschiedenen Konzepte darauf reagieren. Um die Ergebnisse besser einordnen zu können, wird für die Funktion des kamerabasierten Assistenzsystems ein ASIL B gefordert. Mit den Werten aus Abschnitt 3.3.2 entspäche dies einer maximal erlaubten Ausfallwahrscheinlichkeit von $1 \cdot 10^{-7}$.

8.1 Berechnung und Analyse der Fehlerbäume

Vor der Analyse der Fehlerbäume werden in diesem Abschnitt anfangs die konkreten Ausfallwahrscheinlichkeiten, die für die Redundanzkonzepte angenommen wurden, aufgeführt. Danach erfolgt ein Überblick über die Ergebnisse der einzelnen Bäume bezüglich ihrer Ausfallwahrscheinlichkeiten.

Die folgenden Wahrscheinlichkeiten, die für die Komponenten und Fehler im Netzwerk angenommen wurden, entsprechen keinen realistischen Daten. Sie dienen dazu anschauliche Beispiele zu ermöglichen, mit denen die Auswirkungen der Redundanzkonzepte dargelegt werden können. Die Ausfallwahrscheinlichkeiten orientieren sich allerdings an den aus den ASIL hervorgehenden Anforderungen (siehe Abschnitt 3.3.2).

Für alle Konzepte werden die gleichen Wahrscheinlichkeiten für den Ausfall der Hardwarekomponenten angenommen:

- *Display1* = $7 \cdot 10^{-8}$
- *DA_CAM* = $1 \cdot 10^{-7}$
- alle Switches = $5 \cdot 10^{-8}$
- alle Kabelverbindungen =$1 \cdot 10^{-8}$

Für das Konzept mit redundanter Hardware gelten folgende Ausfallwahrscheinlichkeiten für die einzelnen Module der dynamischen Redundanz:

- *ECU1* und *ECU2* = $1 \cdot 10^{-7}$
- Switch und Fehlererkennung kombiniert = $6 \cdot 10^{-7}$

Die Fehlerszenarien des ursprünglichen Netzwerks ergeben zusammengefasst:

- Netzwerkfehler = $1,642 \cdot 10^{-8}$
- Technologiefehler = $1,3 \cdot 10^{-10}$

Die Fehlerszenarien der Ringtopologie ergeben zusammengefasst:

- Netzwerkfehler = $1,632 \cdot 10^{-8}$
- Technologiefehler = $1,3 \cdot 10^{-10}$
- Redundanzfehler = $8,05 \cdot 10^{-9}$

Die Fehlerszenarien der redundanten Hardware ergeben zusammengefasst:

- Netzwerkfehler = $1,632 \cdot 10^{-8}$
- Technologiefehler = $1,3 \cdot 10^{-10}$
- Redundanzfehler = $9,5 \cdot 10^{-10}$

Mit den oben angegebenen Eigenschaften ergeben die Berechnungen der Ausfallwahrscheinlichkeiten für das Top-Event:

- ursprüngliches Netzwerk = $3,165 \cdot 10^{-7}$
- Ringtopologie = $3,145 \cdot 10^{-7}$
- redundante Hardware = $8,74 \cdot 10^{-8}$

Werden ausschließlich die Hardwarefehler betrachtet, ohne die zusätzlichen Fehlerszenarien zu berücksichtigen, kommen die folgenden Ergebnisse zustande:

- ursprüngliches Netzwerk = $3 \cdot 10^{-7}$
- Ringtopologie = $2,9 \cdot 10^{-7}$
- redundante Hardware = $7 \cdot 10^{-8}$

8.2 Vergleich der Konzepte

Durch die vergleichsweise starke Redundanz existieren bei der redundanten Hardware hardwaretechnisch fast keine SPOFs mehr. Aus diesem Grund schneidet dieses Konzept am besten bei der Analyse ab. Das ursprüngliche Netzwerk und die Ringtopologie sind hingegen, von den Ausfallwahrscheinlichkeiten her, sehr nah beieinander. Beide Konzepte haben bei der Hardware diverse SPOFs, die zu einem schlechteren Ergebnis beitragen. Betrachtet man lediglich die Hardwareausfälle und ignoriert die anderen Fehlerszenarien, ist die Ringtopologie etwas zuverlässiger. Vor allem aufgrund der Redundanzfehler gleichen sich die Ergebnisse unter Berücksichtigung aller Fehlerszenarien an, da das ursprüngliche Netzwerk keine solcher Fehler hat. Die Gegenüberstellung der Ausfallwahrscheinlichkeiten der beiden Topologien kann dem Diagramm in Abbildung 8.1 entnommen werden. Im Bezug auf das geforderte ASIL erfüllt die redundante Hardware diese Anforderungen. Die anderen beiden Konzepte liegen etwas über dem geforderten Wert und müssten somit überarbeitet werden, um den Anforderungen zu entsprechen.

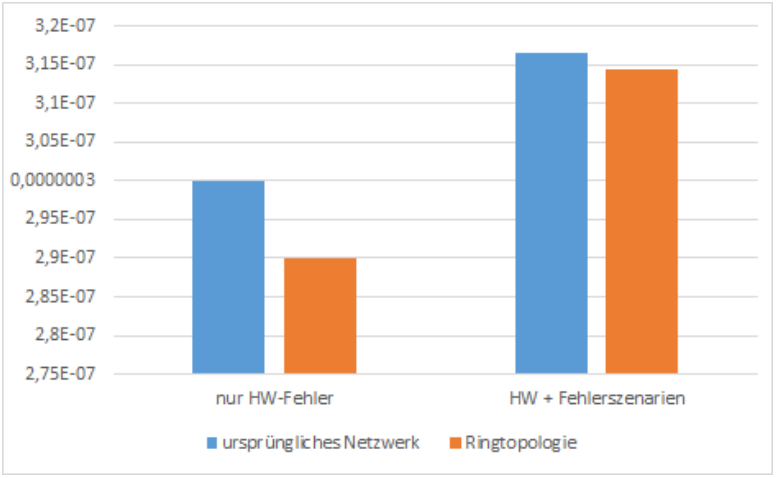


Abbildung 8.1: Gegenüberstellung der Ergebnisse des ursprünglichen Netzwerks und der Ringtopologie

8.3 Veränderte Eigenschaften

Um die Reaktionen der Konzepte zu vergleichen, wenn bestimmte Situationen eintreten, wurden Änderungen an den Ausfallwahrscheinlichkeiten einiger Elemente vorgenommen. Zu Beginn

dieses Abschnitts werden die einzelnen Szenarios beschrieben und die Ergebnisse aufgelistet. Am Ende erfolgt eine Analyse, wie sich die unterschiedlichen Konzepte verhalten.

8.3.1 Kabelverbindungen

In diesem Szenario wird die Ausfallwahrscheinlichkeit der Kabel des gesamten Netzwerks von $1 \cdot 10^{-8}$ auf $1 \cdot 10^{-7}$ gesetzt. Ein solcher Fall könnte eintreten, wenn ein System mit günstigerer Hardware getestet werden soll, welche eine höhere Ausfallwahrscheinlichkeit hat.

Die Analysen der Bäume ergibt die folgenden Werte für die Top-Events:

- ursprüngliches Netzwerk = $5,865 \cdot 10^{-7}$
- Ringtopologie = $4,945 \cdot 10^{-7}$
- redundante Hardware = $8,74 \cdot 10^{-8}$

8.3.2 Switches

Die Ausfallwahrscheinlichkeit der Switches wird in diesem Szenario von $5 \cdot 10^{-8}$ auf $5 \cdot 10^{-7}$ erhöht. Dieses Szenario könnte ebenfalls eintreten, wenn ein Netzwerk mit anderer Hardware getestet werden soll.

Die Analysen der Bäume ergibt die folgenden Werte für die Top-Events:

- ursprüngliches Netzwerk = $1,216 \cdot 10^{-6}$
- Ringtopologie = $1,214 \cdot 10^{-6}$
- redundante Hardware = $8,74 \cdot 10^{-8}$

8.3.3 Äußere Einwirkung

Das Szenario fügt eine zusätzliche äußere Einwirkung hinzu, welche sich auf das gesamte System auswirkt und direkt zum Eintreten des Top-Events führt. Hierbei könnte es sich beispielsweise um eine neue Funktion im Auto handeln, die zusätzliche elektrische Interferenzen verursacht. Die neue Störung hat eine Eintrittswahrscheinlichkeit von $1 \cdot 10^{-7}$.

Die Analysen der Bäume ergibt die folgenden Werte für die Top-Events:

- ursprüngliches Netzwerk = $4,165 \cdot 10^{-7}$
- Ringtopologie = $4,145 \cdot 10^{-7}$
- redundante Hardware = $1,874 \cdot 10^{-7}$

8.3.4 Zusammenfassung der Ergebnisse

Alle Ergebnisse der Szenarien werden in dem Diagramm in Abbildung 8.2 gezeigt. Der erste Teil des Diagramms zeigt zum Vergleich die Ergebnisse der unveränderten Redundanzkonzepte.

Bei den Ergebnissen der fehlerhaften Kabelverbindungen fällt auf, dass sie anscheinend keine Auswirkungen auf das Top-Event des Fehlerbaums der redundante Hardware haben. Das liegt daran, dass die zwei Pfade komplett unabhängig voneinander sind. Die Ausfallwahrscheinlichkeit des AND-Gates unter dem sich die Pfade mit den fehlerhaften Verbindungen befinden ändert sich von $9,1 \cdot 10^{-15}$ auf $1 \cdot 10^{-13}$. Da dies viel niedriger ist als das Ergebnis des Top-Events, wirkt sich dieser Wert nicht spürbar darauf aus.

Bei dem Vergleich der Werte des ursprünglichen Netzwerk mit denen der Ringtopologie ist zu erkennen, dass sich die fehlerhaften Verbindungen stärker auf die ursprüngliche Topologie auswirken. Bei dieser verschlechtert sich der Wert um $2,865 \cdot 10^{-7}$ bei der Ringtopologie hingegen nur um $2,045 \cdot 10^{-7}$. Hier macht sich die zusätzliche redundante Verbindung über *Switch3* positiv bemerkbar.

Auch bei den fehlerhaften Switches kann, wie schon bei dem vorherigen Szenario, keine Veränderung bei der redundanten Hardware festgestellt werden. Wie zuvor liegt auch das an den komplett redundanten Pfaden, sodass sich der Wert des AND-Gates wieder nur minimal ändert: von $9,1 \cdot 10^{-15}$ auf $5,355 \cdot 10^{-13}$.

Der Vergleich der beiden andern Netzwerke zeigt, dass diese nahezu identische Ergebnisse besitzen. Vergleicht man die genauen Änderungen zu den vorherigen Wahrscheinlichkeiten, legt die ursprüngliche Topologie um $9,16 \cdot 10^{-7}$ zu und die Ringtopologie um $9,24 \cdot 10^{-7}$. In diesem Fall ist die Veränderung bei dem Ring sogar etwas schlechter, da hier auch auf dem redundanten Pfad noch ein zusätzlicher Switch vorhanden ist und die höhere Ausfallwahrscheinlichkeit sich stärker bemerkbar macht.

Das Hinzufügen der zusätzlichen Störung hat Auswirkungen auf alle Topologien. Da es sich, im Vergleich zu allen anderen Events, um eine verhältnismäßig hohe Wahrscheinlichkeit handelt und zudem noch als SPOF direkt unter dem Top-Event angesiedelt ist, wirkt sie sich auch stark auf die Ergebnisse aus. Die Ausfallwahrscheinlichkeiten aller Top-Events verschlechtert sich genau um den Wert der Störung: $1 \cdot 10^{-7}$. Im Hinblick auf das geforderte ASIL B, würde das Konzept der redundanten Hardware nun ebenfalls nicht mehr die nötige Ausfallwahrscheinlichkeit erreichen.

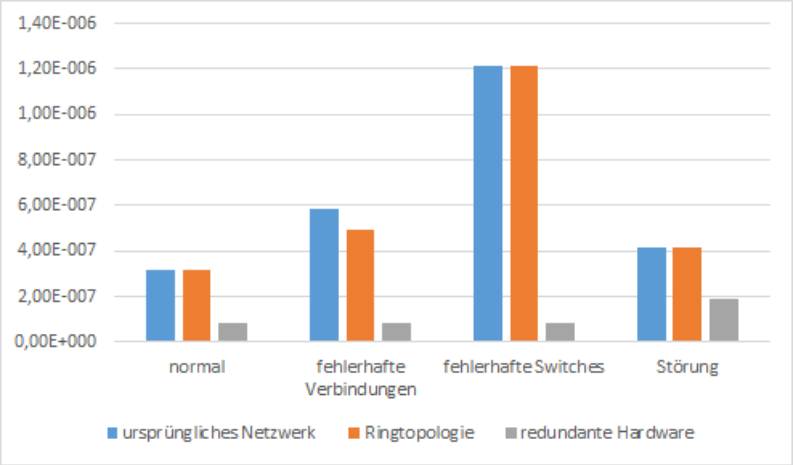


Abbildung 8.2: Gegenüberstellung der Ausfallwahrscheinlichkeiten mit veränderten Eigenschaften

8.4 Fazit

Bezüglich der Redundanzkonzepte kann gesagt werden, dass die Topologie mit der redundanten Hardware eindeutig die besten Ergebnisse erzielt hat. Bei diesem Konzept wurde aber auch am meisten Aufwand betrieben diese Redundanz zu ermöglichen. Es wurde über den zusätzlichen Switch ein komplett neuer Pfad zwischen der Kamera und dem Display etabliert. Des Weiteren wurde der Kameraknoten komplett neu als dynamische Redundanz entwickelt und so ein weiterer SPOF entfernt. Im Hinblick auf die wirtschaftlichen Anforderungen, die an das System gestellt werden, ist dieses Konzept auch das problematischste, da vor allem für die Entwicklung, aber auch von den Stückkosten her, deutlich höhere Kosten entstehen.

Die Ringtopologie konnte in dem hier untersuchten Szenario keine deutlichen Verbesserungen der Ausfallwahrscheinlichkeiten gegenüber der ursprünglichen Topologie erzielen. Es konnte aber bei den veränderten Ausfallwahrscheinlichkeiten gezeigt werden, dass sie das Potential hat das System positiv zu beeinflussen.

Es ist im Hinblick auf beide Konzepte aber zu beachten, dass hier lediglich die Kommunikation zwischen zwei Endpunkten untersucht wurde. Die redundanten Elemente, welche zusätzliche Pfade zur Verfügung stellen, bieten aber anderen sicherheitskritischen Kommunikationen ebenfalls neue Wege ihre Daten sicher zum Ziel zu übertragen. Es können also alle Netzwerkteilnehmer davon profitieren, was die Kosten relativiert.

Für das in dieser Arbeit untersuchte Szenario der Kommunikation zwischen Kamera und Display verbessert die Ringtopologie die Sicherheit des Netzwerks nicht genügend. Die redundante

Hardware hingegen ist mit den zusätzlichen redundanten Komponenten etwas überdimensioniert. Eine an dieses Konzept angelehnte reduzierte Lösung würde wahrscheinlich immer noch eine gute Verbesserung der Ausfallwahrscheinlichkeiten bringen aber gleichzeitig auch weniger Kosten verursachen.

Im Bezug auf die Fehlerbäume konnte gezeigt werden, dass sie sich eigenen Netzwerktopologien abzubilden und unterschiedliche Konzepte miteinander zu vergleichen. Mit einer geeigneten Software ist es möglich schnell Änderungen an vielen Ereignissen gleichzeitig vorzunehmen. Dies macht die Untersuchung des selben Systems in unterschiedlichen Situationen möglich.

Wenn die Fehlerszenarien aller Komponenten und die Ereignisse, die sonst noch in irgendeiner Weise auf das System einwirken, erfasst wurden, steht ein bekannte und weit verbreitete Analysemethode zur Verfügung. Es lohnt sich also diesen Ansatz für die Entwicklung neuer Kommunikationsarchitekturen für Fahrzeugnetzwerke einzusetzen.

9 Zusammenfassung & Ausblick

Ziel dieser Arbeit war es ein Analyseverfahren zu finden, mit dem redundante Kommunikationsarchitekturen im Automobil analysiert und bewertet werden können. Für den Entwicklungsprozess einer neuen Technologie ist das ein besonders wichtiger Schritt. Die Fehlerbaumanalyse, welche zu diesem Zweck eingesetzt werden soll, erfasst die Gefahren und Bedrohungen eines Systems und liefert letztendlich eine Wahrscheinlichkeit, mit der das System zum benötigten Zeitpunkt nicht zur Verfügung steht. Die Anwendung dieser Analysemethode wurde im Hinblick auf verschiedene redundante Kommunikationsarchitekturen untersucht.

In der Einleitung wurde die Notwendigkeit neuer leistungsfähigerer Kommunikationstechnologien im Automobil gezeigt. Sollen Daten sicherheitskritischer Systeme über das Netzwerk versendet werden, besteht der Bedarf an echtzeitfähigen und robusten Lösungen. Ein Ansatz ist die Zuverlässigkeit durch den Einsatz von Redundanz zu erhöhen. Um solche Konzepte zu analysieren bietet sich die Fehlerbaumanalyse an.

Mit CAN, LIN, MOST und FlexRay wurden aktuell, in Fahrzeugen verbauten, Bussysteme vorgestellt, deren Eigenschaften aber nicht den kommenden Anforderungen an Bandbreite und Echtzeitfähigkeit entsprechen. Deswegen spielt das Thema Ethernet im Auto eine wichtige Rolle in der Forschung. Es gibt einige Ansätze, die das Standard-Ethernet um Konzepte erweitern, welche es ermöglichen genaue Vorhersagen zum Ankunftszeitpunkt der Nachrichten zu treffen. Dies ist für die sicherheitskritischen Anwendungen essentiell. Solche Anforderungen gehen unter anderem aus der ISO 26262 hervor, die als Grundlage zur Entwicklung sicherheitskritischer elektronischer Systeme im Automobil dient.

Bei der Redundanz stehen einer Einheit mehr Mittel zur Verfügung als eigentlich notwendig. Beim Ausfall einer Komponente kann immer noch der redundante Teil die Aufgaben übernehmen. Ein System kann auf unterschiedliche Art und Weise mit Redundanz abgesichert werden. Es wurden Konzepte vorgestellt, wie mit Hilfe redundanter Topologiekonzepte ein Kommunikationsnetzwerk erweitert werden kann. Es besteht aber auch die Möglichkeit Hardware mehrfach zu verbauen, um so Defekte der Einheiten kompensieren zu können. Diese Hardwareredundanz kann wiederum unterschiedlich stark ausgeprägt sein, um beispielsweise auch dem Ausfall mehrerer Elemente widerstehen zu können. Als dritte Möglichkeit wird

die Softwareredundanz erläutert. Hierbei wird die gleiche Software von unterschiedlichen Entwicklerteams programmiert. So besteht die Chance die Gefahr von Programmierfehlern zu reduzieren.

Das nächste Kapitel befasst sich mit der statischen und der dynamischen Fehlerbaumanalyse. Die Statische bildet die Grundlage und ermöglicht den Aufbau der Bäume hauptsächlich mit AND- und OR-Gates. Die Blätter des Baums sind Ereignissen, denen eine Ausfallwahrscheinlichkeit zugeordnet wird. Aus dem Aufbau der Gates und Ereignisse, ergibt sich letztendlich die Wahrscheinlichkeit der Nichtverfügbarkeit des Systems zu einem bestimmten Zeitpunkt. Durch die dynamische Fehlerbaumanalyse können weitere Gates in die Bäume integriert werden um Abhängigkeiten und Redundanz darstellen zu können. Am Ende des Kapitels wird noch ein Tool vorgestellt, mit dem die Fehlerbäume in einer Software erstellt und analysiert werden können.

Darauf folgt das Thema, welche Fehlerszenarien in einem Kommunikationsnetzwerk auftreten können und wie diese in die Fehlerbäumen einfließen. Diese Szenarien wurden im nächsten Kapitel genutzt, um ein, auf einem Serienfahrzeug basierenden, TTE-Netzwerk zu untersuchen. Zu der Topologie wurden zusätzlich zwei Redundanzkonzepte entwickelt, die die Zuverlässigkeit des Netzwerks verbessern sollen. Das erste Redundanzkonzept erweitert das Netz um einen zusätzlichen Pfad, sodass eine Ringtopologie entsteht, über die in beide Richtungen kommuniziert werden kann. Das zweite Konzept sieht vor mit Hilfe mehrerer redundanter Hardwareelemente die Ausfallwahrscheinlichkeit des Systems zu senken. Für alle drei Netzwerke sind die dazugehörigen Fehlerbäume entwickelt worden.

Die Analyse und Bewertung dieser Bäume wurde im darauffolgenden Kapitel gezeigt. Die Ergebnisse der Ausfallwahrscheinlichkeiten konnten so miteinander verglichen werden. Im nächsten Schritt sind diverse Ausfallwahrscheinlichkeiten verändert worden um die Reaktion der unterschiedlichen Konzepte zu erfassen. Das Ergebnis dieses Kapitels ist, dass die Lösung mit redundanter Hardware die besten Werte erreicht. Allerdings zu dem Preis, dass die Kosten und der Aufwand für die Implementierung am höchsten sind. Die Ringtopologie konnte sich in diesem Szenario nicht deutlich von der ursprünglichen Konfiguration absetzen, hatte dafür aber auch nur ein sehr geringes Maß an zusätzlichen Kosten.

Abschließend ist noch zu sagen, dass sich der Einsatz der Fehlerbaumanalyse während der Entwicklung einer Kommunikationsarchitektur bewährt hat und es sich lohnt diesen Ansatz weiter zu verfolgen.

Zukünftige Arbeiten zum Thema der Fehlerbaumanalyse für Kommunikationsarchitekturen im Automobil könnten sein:

Das Thema der Fehlerszenarien ist noch nicht ausgereizt. Einige der in dieser Arbeit verwendeten Beispiele könnten noch detaillierter dargestellt werden.

Taucht man noch weiter in die Materie der Fehlerbäume ein, ist es möglich auch zeitliche Abläufe in die Fehlerbäume zu integrieren. Das könnten beispielsweise aufeinanderfolgende Fehler innerhalb eines bestimmten Zeitraums sein oder nur vorübergehende Fehler, die sich nach einer gewissen Zeit selbst reparieren. Dadurch würde ein sehr dynamisches System entstehen, was zu einer höheren Anzahl dynamischer Gates führt. Dies wiederum könnte die Voraussetzung schaffen, dass alternative Berechnungsmethoden für die dynamischen Gates verwendet werden müssen.

Das Design einer kompletten redundanten Kommunikationsarchitektur, das möglichst alle kritischen Bereiche des Netzwerks umfasst. Hierzu müssen aber erst Vorarbeiten, wie die detaillierte Erfassung der Fehlerszenarien.

Literaturverzeichnis

[ARINC 664 2009] Aeronautical Radio Incorporated: Avionics Full-Duplex Switched Ethernet. 2009. – Forschungsbericht

[AS 6802 2016] AS 6802: Time-Triggered Ethernet / SAE. SAE International, 2016. – 09-11-2016

[Brajou und Ricco 2004] Brajou, F. ; Ricco, P.: The Airbus A380 - an AFDX-based flight test computer concept. In: *Proceedings AUTOTESTCON 2004.*, Sept 2004, S. 460–463. – ISSN 1088-7725

[DIN 25424-1 1981] Deutsches Institut für Normung: Fehlerbaumanalyse; Methode und Bildzeichen / DIN. Berlin : Beuth, 1981 (25424-1). – 00-09-1981

[DIN 25424-2 1990] Deutsches Institut für Normung: Fehlerbaumanalyse; Handrechenverfahren zur Auswertung eines Fehlerbaumes / DIN. Berlin : Beuth, 1990 (25424-2). – 00-04-1990

[DIN 40041 1990] Deutsches Institut für Normung: Zuverlässigkeit; Begriffe / DIN. Berlin : Beuth, 1990 (40041). – 00-12-1990

[DIN EN 61025 2007] Deutsches Institut für Normung: Fehlzustandsbaumanalyse / DIN. Berlin : Beuth, 2007 (61025). – 00-08-2007

[DIN EN 62439-2 2010] Deutsches Institut für Normung: Industrielle Kommunikationsnetze - Hochverfügbare Automatisierungsnetze - Teil 2: Medienredundanz-Protokoll (MRP) / DIN. Berlin : Beuth, 2010 (62439-2). – 00-09-2010

[DIN EN 62439-3 2013] Deutsches Institut für Normung: Industrielle Kommunikationsnetze - Hochverfügbare Automatisierungsnetze - Teil 3: Parallelredundanz-Protokoll (PRP) und nahtloser Hochverfügbarkeits-Ring (HSR) / DIN. Berlin : Beuth, 2013 (62439-3). – 00-06-2013

[Dugan u. a. 1992] Dugan, J. B. ; Bavuso, S. J. ; Boyd, M. A.: Dynamic fault-tree models for fault-tolerant computer systems. In: *IEEE Transactions on Reliability* 41 (1992), Sep, Nr. 3, S. 363–377. – ISSN 0018-9529

[Esteve u. a. 2012] Esteve, M. A. ; Katoen, J. P. ; Nguyen, V. Y. ; Postma, B. ; Yushtein, Y.: Formal correctness, safety, dependability, and performance analysis of a satellite. In: *2012 34th International Conference on Software Engineering (ICSE)*, June 2012, S. 1022–1031. – ISSN 0270-5257

[Firoozshahi 2010] Firoozshahi, A.: High speed redundant Multi-Network DCS-based in innovative tank gauging control system. In: *Advanced Computer Control (ICACC), 2010 2nd International Conference on* Bd. 1, March 2010, S. 333–337

[Gohil u. a. 2011] Gohil, S. ; Basavalingarajaiah, A. ; Ramachandran, V.: Redundancy management and synchronization in avionics communication products. In: *Integrated Communications, Navigation and Surveilance Conference (ICNS), 2011*, May 2011, S. C3–1–C3–8. – ISSN 2155-4943

[Goraj und Harada 2012] Goraj, M. ; Harada, R.: Migration paths for IEC 61850 substation communication networks towards superb redundancy based on hybrid PRP and HSR topologies. In: *Developments in Power Systems Protection, 2012. DPSP 2012. 11th International Conference on*, April 2012, S. 1–6

[IEEE 802.1 TSN Task Group] IEEE 802.1 TSN Task Group: *IEEE 802.1Q - Virtual LANs*. – URL `http://www.ieee802.org/1/pages/802.1Q.html`

[IEEE 802.1BA 2011] Institute of Electrical and Electronics Engineers: IEEE 802.1BA - IEEE Standard for Local and Metropolitan Area Networks—Audio Video Bridging (AVB) Systems. URL `http://www.ieee802.org/1/pages/802.1ba.html`. – Zugriffsdatum: 19.03.2017, September 2011. – Forschungsbericht

[IEEE 802.1CB 2013] Institute of Electrical and Electronics Engineers: IEEE 802.1CB - Frame Replication and Elimination for Reliability. URL `http://www.ieee802.org/1/pages/802.1cb.html`. – Zugriffsdatum: 19.03.2017, Juni 2013. – Forschungsbericht

[IEEE 802.1Qat 2006] Institute of Electrical and Electronics Engineers: IEEE 802.1Qat - Stream Reservation Protocol. URL `http://www.ieee802.org/1/pages/802.1at.html`. – Zugriffsdatum: 19.03.2017, September 2006. – Forschungsbericht

[IEEE 802.1Qav 2009] Institute of Electrical and Electronics Engineers: IEEE 802.1Qav - Forwarding and Queuing Enhancements for Time-Sensitive Streams. URL `http://www.ieee802.org/1/pages/802.1av.html`. – Zugriffsdatum: 19.03.2017, Dezember 2009. – Forschungsbericht

[IEEE 802.1Qbu 2012] Institute of Electrical and Electronics Engineers: IEEE 802.1Qbu - Frame Preemption. URL `http://www.ieee802.org/1/pages/802.1bu.html`. – Zugriffsdatum: 19.03.2017, Mai 2012. – Forschungsbericht

[IEEE 802.1Qbv 2012] Institute of Electrical and Electronics Engineers: IEEE 802.1Qbv - Enhancements for Scheduled Traffic. URL `http://www.ieee802.org/1/pages/802.1bv.html`. – Zugriffsdatum: 19.03.2017, Mai 2012. – Forschungsbericht

[IEEE 802.1Qcc 2013] Institute of Electrical and Electronics Engineers: IEEE 802.1Qcc - Stream Reservation Protocol (SRP) Enhancements and Performance Improvements. URL `http://www.ieee802.org/1/pages/802.1cc.html`. – Zugriffsdatum: 19.03.2017, Oktober 2013. – Forschungsbericht

[IEEE 802.3 2015] IEEE 802.3 Ethernet Working Group: 802.3 - IEEE Standard for Ethernet. URL `http://www.ieee802.org/3/`. – Zugriffsdatum: 19-03-2017, September 2015. – Forschungsbericht

[Karimi u. a. 2010] Karimi, A. ; Zarafshan, F. ; Jantan, A. b. ; Ramli, A. R. b. ; Saripan, M. I. b.: An optimal parallel average voting for fault-tolerant control systems. In: *2010 International Conference on Networking and Information Technology*, June 2010, S. 360–363. – ISSN 2324-819X

[Kirrmann und Dzung 2006] Kirrmann, H. ; Dzung, D.: Selecting a Standard Redundancy Method for Highly Available Industrial Networks. In: *Factory Communication Systems, 2006 IEEE International Workshop on*, 2006, S. 386–390

[Leu u. a. 2015] Leu, K. L. ; Huang, H. ; Chen, Y. Y. ; Huang, L. R. ; Ji, K. M.: An intelligent brake-by-wire system design and analysis in accordance with ISO-26262 functional safety standard. In: *2015 International Conference on Connected Vehicles and Expo (ICCVE)*, Oct 2015, S. 150–156

[Merle u. a. 2009] Merle, Guillaume ; Roussel, Jean-Marc ; Lesage, Jean-Jacques ; Bobbio, Andrea: Algebraic Expression of the Structure Function of a subclass of Dynamic Fault Trees. In: *2nd IFAC Workshop on Dependable Control of Discrete Systems (DCDS'09)*.

Bari, Italy, Juni 2009, S. 129–134. – URL https://hal.archives-ouvertes.fr/hal-00394459. – Zugriffsdatum: 2017-03-09

[Merle u. a. 2010] Merle, Guillaume ; Roussel, Jean-Marc ; Lesage, Jean-Jacques ; Vayatis, Nicolas: Analytical Calculation of Failure Probabilities in Dynamic Fault Trees including Spare Gates. In: Ben J.M. Ale, Enrico Z. (Hrsg.): *European Safety and Reliability Conference (ESREL 2010)*. Rhodes, Greece : Taylor & Francis, September 2010, S. pp. 794–801. – URL https://hal.archives-ouvertes.fr/hal-00516893. – Zugriffsdatum: 2017-03-09

[Nalos und Schulz 1965] Nalos, E. J. ; Schulz, R. B.: Reliability and Cost of Avionics. In: *IEEE Transactions on Reliability* R-14 (1965), Oct, Nr. 2, S. 120–130. – ISSN 0018-9529

[Nassaj und Barabady 2016] Nassaj, A. ; Barabady, J.: Fault tree analysis of oil and gas distillation tower and application of Bayesian Networks. In: *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Dec 2016, S. 1669–1673

[Pullum 2001] Pullum, Laura L.: *Software fault tolerance techniques and implementation*. Boston : Artech House, 2001. – ISBN 978-1580531375

[Rao u. a. 2009] Rao, K. D. ; Gopika, V. ; Rao, V.V.S. S. ; Kushwaha, H.S. ; Verma, A.K. ; Srividya, A.: Dynamic fault tree analysis using Monte Carlo simulation in probabilistic safety assessment. In: *Reliability Engineering & System Safety* 94 (2009), Nr. 4, S. 872 – 883. – URL http://www.sciencedirect.com/science/article/pii/S0951832008002354. – Zugriffsdatum: 2017-01-09. – ISSN 0951-8320

[Schilling 2009] Schilling, S. J.: *Beitrag zur dynamischen Fehlerbaumanalyse ohne Modulbildung und zustandsbasierte Erweiterungen*, Bergische Universität Wuppertal, Dissertation, 2009. – URL http://elpub.bib.uni-wuppertal.de/servlets/DerivateServlet/Derivate-1483/dd0904.pdf. – Zugriffsdatum: 2016-06-16

[Schneele und Geyer 2012] Schneele, S. ; Geyer, F.: Comparison of IEEE AVB and AFDX. In: *2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC)*, Oct 2012, S. 7A1–1–7A1–9. – ISSN 2155-7195

[Sghairi u. a. 2008] Sghairi, M. ; Bonneval, A. d. ; Crouzet, Y. ; Aubert, J. J. ; Brot, P.: Architecture Optimization Based on Incremental Approach for Airplane Digital Distributed

Flight Control System. In: *World Congress on Engineering and Computer Science 2008, WCECS '08. Advances in Electrical and Electronics Engineering - IAENG Special Edition of the*, Oct 2008, S. 13–20

[Spitzer 2006] Spitzer, Cary R.: *Avionics: development and implementation.* Boca Raton : CRC Press, 2006. – ISBN 0849384419

[Steinbach u. a. 2012] Steinbach, Till ; Lim, Hyung-Taek ; Korf, Franz ; Schmidt, Thomas C. ; Herrscher, Daniel ; Wolisz, Adam: Tomorrow's In-Car Interconnect? A Competitive Evaluation of IEEE 802.1 AVB and Time-Triggered Ethernet (AS6802). In: *2012 IEEE Vehicular Technology Conference (VTC Fall).* Piscataway, New Jersey : IEEE Press, September 2012. – ISSN 1090-3038

[Steinbach u. a. 2015] Steinbach, Till ; Lim, Hyung-Taek ; Korf, Franz ; Schmidt, Thomas C. ; Herrscher, Daniel ; Wolisz, Adam: Beware of the Hidden! How Cross-traffic Affects Quality Assurances of Competing Real-time Ethernet Standards for In-Car Communication. In: *2015 IEEE Conference on Local Computer Networks (LCN)*, Oktober 2015. – Accepted for publication

[Storey 1996] Storey, Neil R.: *Safety Critical Computer Systems.* Boston : Addison-Wesley Longman Publishing Co., Inc., 1996. – ISBN 0201427877

[Teener 2015] Teener, Michael J.: *A Time-Sensitive Networking Primer: Putting It All Together.* 2015. – URL `https://drive.google.com/file/d/0B6Xurc4m_PVsZ1lzWWoxS0pTNVE/view`. – Zugriffsdatum: 19-03-2017. – ISPCS 2015 in Beijing

[Thums 2004] Thums, Andreas: *Formale Fehlerbaumanalyse*, Universität Augsburg, Dissertation, 2004

[TSN TG] Institute of Electrical and Electronics Engineers: *Time-Sensitive Networking Task Group.* – URL `http://www.ieee802.org/1/pages/tsn.html`. – Zugriffsdatum: 19-03-2017

[Vesely u. a. 2002] Vesely, W. ; Dugan, J. ; Fragola, J. ; Minarick ; Railsback, J.: Fault Tree Handbook with Aerospace Applications / National Aeronautics and Space Administration. Washington, DC, 2002. – Handbook

[Yuyan u. a. 2015] Yuyan, C. ; Jian, W. ; Rong, X. ; Xinmin, W.: Fault tree analysis of electro-mechanical actuators. In: *2015 34th Chinese Control Conference (CCC)*, July 2015, S. 6392–6396

[Zimmermann und Schidgall 2014] Zimmermann, Werner ; Schidgall, Ralf: *Bussysteme in der Fahrzeugtechnik: Protokolle, Standards und Softwarearchitektur.* 5. Auflage. Wiesbaden : Springer Vieweg, 2014

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 21. März 2017 Stefan Buschmann