

FAT-Schriftenreihe 403

Simulationsbasierte Bewertung von eFuse-Verhalten
im Kontext komplexer Fahrzeugbordnetze



Simulationsbasierte Bewertung von eFuse-Verhalten im Kontext komplexer Fahrzeugbordnetze

Autoren:

Prof. Dr.-Ing. Stephan Frei, TU Dortmund

M. Sc. Michael Gerten, TU Dortmund

M. Sc. Marvin Rübartsch, TU Dortmund

Sven Reitz, Fraunhofer IIS/EAS, Dresden

Jens Warmuth, Fraunhofer IIS/EAS, Dresden

Dr. Roland Jancke, Fraunhofer IIS/EAS, Dresden

**Das Forschungsprojekt wurde mit Mitteln der Forschungsvereinigung
Automobiltechnik e.V. (FAT) gefördert.**

Vorbemerkung

Das Vorhaben wurde begleitet durch den FAT-Arbeitskreis

- VDA/FAT AK 30 "Elektrische Energie" im Cluster „Umwelt, System Straßenverkehr“
Sprecher des Arbeitskreises: Herr Dr. Martin Düsing

Beteiligte Forschungsnehmer waren

- Fraunhofer-Institut Integrierte Schaltungen IIS,
Institutsteil Entwicklung Adaptiver Systeme EAS Dresden
Leiter des Institutsteils: Dr. Peter Schneider
Münchner Str. 16
01187 Dresden
- TU Dortmund
Prof. Dr.-Ing. Stephan Frei
Arbeitsgebiet Bordsysteme
Fakultät für Elektrotechnik und Informationstechnik
Otto-Hahn-Straße 4
44227 Dortmund

Der Bericht gliedert sich nach den einleitenden Bemerkungen entsprechend den Arbeitspaketen des Vorhabens:

- AP1 Nicht-kommandiertes eFuse Verhalten in komplexen Bordnetzen
- AP2 Kommandiertes eFuse Verhalten in komplexen Bordnetzen
- AP3 Simulationstechnik

Die im vorliegenden Bericht beschriebenen Projektergebnisse stammen aus dem Gemeinschaftsprojekt, das von Fraunhofer IIS/EAS und der TU Dortmund, Arbeitsgebiet Bordsysteme, gemeinsam bearbeitet wurde. Von dem Bericht wurden die Abschnitte 2 und 3 von der TU Dortmund verfasst, während die Abschnitte 4 und 5 von Fraunhofer IIS/EAS beschrieben worden sind.

Die an dem Vorhaben Mitwirkenden der beteiligten Forschungsnehmer danken der Forschungsvereinigung Automobiltechnik im Verband der Automobilindustrie e.V. für die finanzielle Unterstützung der Arbeiten und den Mitgliedern des VDA/FAT-AK30 für die inhaltliche Begleitung.

Inhalt

Vorbemerkung.....	- 1 -
1 Überblick.....	- 5 -
2 Nicht-kommandiertes eFuse-Verhalten in komplexen Bordnetzen	- 9 -
2.1 Messtechnische Untersuchungen elektronischer Sicherungen	- 9 -
2.1.1 Untersuchte eFuse (Infineon BTG7003).....	- 9 -
2.1.2 Validierung des Ein- und Ausschaltverhaltens	- 10 -
2.1.3 Validierung des I2t -Verhaltens	- 11 -
2.2 Analyse der Sensoranforderungen für erweiterte Diagnosefunktionen	- 13 -
2.2.1 Sensorkonzept zur Bestimmung des Einschaltwiderstands	- 13 -
2.2.2 Modellierung der Alterungs- und Temperaturabhängigkeit.....	- 14 -
2.2.3 Vorabschätzung mit Worst-Case Annahmen.....	- 16 -
2.2.4 Analyse des Diagnosekonzepts zur Bestimmung des Einschaltwiderstands-	- 17 -
3 Kommandiertes eFuse-Verhalten in komplexen Bordnetzen	- 20 -
3.1 Erweiterung des eFuse-Modells zur externen Ansteuerung	- 21 -
3.2 Entwicklung eines zentralen Lastmanagement-Algorithmus	- 22 -
3.2.1 Prioritätsklassen	- 23 -
3.2.2 Wirksamkeitsbeurteilung.....	- 23 -
3.2.3 Schwellenwerte und Hysterese	- 24 -
3.2.4 Stufenweise Leistungsreduktion	- 25 -
3.2.5 Resultierender Algorithmus	- 25 -
3.3 Demonstration.....	- 26 -
3.3.1 Untersuchtes Beispielbordnetz	- 26 -
3.3.2 Ergebnisse.....	- 28 -
4 Direkter simulativer Vergleich von nicht-kommandiertem und kommandiertem eFuse-	
Verhalten in komplexen Bordnetzen	- 31 -
4.1 Erweiterung der Modelicabibliothek OnboardSystemFaultSimulation	- 32 -
4.1.1 Elektronische Sicherung – eFuse	- 32 -
4.1.2 Triggerlogik für die elektronische Sicherung – eFuse_triggering.....	- 34 -
4.1.3 Power Distribution Unit – PDU.....	- 36 -
4.1.4 Übergeordnete Sicherungsebene – Controller	- 41 -
4.2 Komplexere Bordnetzsimulationen	- 43 -
4.2.1 Szenarium für Simulationstest	- 44 -
4.2.2 Ausfall der elektronischen Lenkung, Variante 1	- 46 -
4.2.3 Ausfall der elektronischen Lenkung, Variante 2	- 48 -
4.2.4 Zuschalten von mehreren nicht-sicherheitsrelevanten Verbrauchern	- 49 -

4.2.5	Serielle Versorgung von PDUs	- 51 -
4.2.6	Fazit	- 53 -
5	Simulationsmethodik	- 55 -
5.1	Herausforderungen der Schutzpfadsimulation.....	- 55 -
5.2	Grundprinzipien effizienter Simulationsmethodik.....	- 56 -
5.2.1	So abstrakt wie möglich, so detailliert wie nötig	- 56 -
5.2.2	Multi-Level- und Multi-Domänen-Simulation.....	- 56 -
5.3	Methoden und Werkzeuge im Detail.....	- 57 -
5.3.1	Dymola / Modelica für Schutzpfad-Komponentenmodelle.....	- 57 -
5.3.2	SystemC-AMS im Detail	- 58 -
5.3.3	FMI und Functional Mock-up Units (FMUs) im Detail	- 59 -
5.3.4	Beschleunigung durch Ausführung von Schutzpfad-Code auf HIL-Hardware via FMU	- 60 -
5.3.5	Zusammenspiel der Methoden und Fazit	- 60 -
6	Zusammenfassung.....	- 61 -
7	Literaturverzeichnis	- 62 -
	Abbildungsverzeichnis.....	- 64 -
	Tabellenverzeichnis.....	- 65 -
	Anhang.....	- 65 -
A.1	Auswertelogik des zentralen Leistungsmanagements.....	- 65 -

1 Überblick

Moderne Fahrzeuge sind mit komplexen Architekturen für das elektrische und elektronische Bordnetz, sogenannten E/E-Architekturen, ausgestattet. Zahlreiche aktuelle Entwicklungen im Bereich der Fahrzeugtechnologien befördern die Tendenz zu immer komplexeren Architekturen. Besonders die vier großen Trends der zukünftigen Mobilität Connectivity, Autonomy, Shared, Electrification (CASE) sorgen für zahlreiche neue und erweiterte Anforderungen an die Fahrzeugbordnetze und befördern die Entwicklung hin zu komplexeren Architekturen.

Die Kommunikation des Fahrzeugs mit der Umgebung, anderen Fahrzeugen, der Infrastruktur und entfernten Servern bringt aus dem Bereich der Konnektivität eine Vielzahl an Anforderungen an die Hard- und Software, die Schnittstellen sowie an Laufzeiten mit sich. Gleichzeitig sind verschiedenste Standards und Nutzungsprofilen zu berücksichtigen. Autonome Fahrfunktionen der unterschiedlichen Level erzeugen deutlich erweiterte Anforderungen an die Absicherung und Verfügbarkeit der Energie- und Informationsflüsse im Fahrzeug. Das Energiebordnetz wird dadurch Teil der sicherheitskritischen Architektur selbst. Neue Geschäftsmodelle im Bereich moderner Mobilitätskonzepte haben ebenfalls Einfluss auf die Vernetzung und die Nutzungsprofile von Fahrzeugen. Schließlich führt auch der Übergang vom Verbrennungsantrieb zu E-Mobilität zu vielfältigen neuen Randbedingungen im Energiebordnetz. Verschiedene Spannungsebenen existieren parallel, versorgen sicherheitskritische wie nichtsicherheitskritische Verbraucher, sichern sich dabei gegenseitig ab und bringen beispielsweise mit On-grid parking und Off-grid parking neue Betriebsmodi und Nutzungsszenarien hervor.

Um diese Herausforderungen zu bewältigen, werden sich zukünftige Energiebordnetze grundlegend von den bekannten konventionellen Architekturen unterscheiden. Neben der Elektrifizierung ist insbesondere die zunehmende Automatisierung von Fahrfunktionen der wichtigste Treiber [1]. Die hierfür notwendigen komplexen Algorithmen zur Verarbeitung der Vielzahl von anfallenden Sensordaten und anschließender Ansteuerung der Aktorik lassen sich nur mit zentralisierten Recheneinheiten bewerkstelligen. Das konventionelle Bordnetz mit zahlreichen, nach Funktionen getrennten Steuergeräten entwickelt sich somit in ein zonales Bordnetz mit einigen wenigen Zonencontrollern. Diese Entwicklungen betreffen sowohl das Informationsbordnetz als auch das Energiebordnetz. In jeder gewählten Topologie müssen alle Komponenten mit elektrischer Energie versorgt werden, je nach Kritikalität auch redundant.

Mit den Entscheidungen zur Architektur geht unmittelbar auch die Frage nach dem Konzept zur Absicherung des Bordnetzes einher. Dabei geht es nicht mehr in erster Linie um die verlässliche Versorgung wichtiger Verbraucher im Netz. Es geht um den Schutz des gesamten Netzes vor Überspannung, Unterspannung, Überstrom und überhöhten Temperaturen, verursacht durch einzelne Verbraucher im Netz. Überlast und Kurzschlüsse sind typische Ursache für derartige Situationen, die erkannt und behandelt werden müssen. Ein geeignetes Absicherungskonzept schließt demnach Verfahren zur Erkennung kritischer Situationen, deren Behandlung beispielsweise durch Abschaltung einzelner Zweige sowie eine zeitliche Abfolge dieser Ereignisse ein.

Moderne Fahrzeug-Bordnetz-Architekturen setzen zunehmend auf elektronische Sicherungen (eFuses) und deren Vorteile [2]. Neben der Möglichkeit, sehr schnell auf Fehler zu reagieren und Verbindungen inaktiv schalten zu können, können sie durch integrierte Strom- und Spannungsmessung wertvolle Diagnoseinformationen liefern [3]. Durch eine optimierte thermische

Überwachung der Leitungen ist außerdem eine Reduktion der notwendigen Kabelquerschnitte und damit eine Kosten- und Gewichtsersparnis denkbar. Gleichzeitig sind aber auch völlig neue Betriebsstrategien und Auslegungsmethoden möglich und nötig.

Im vorliegenden Bericht werden die Potentiale der eFuse in komplexen Bordnetzen untersucht und die Möglichkeiten für deren Nutzung aufgezeigt. Unter komplexen Fahrzeug-Bordnetzen verstehen wir dabei solche mit intelligenten Energieverteilern und mehreren Sicherungsebenen. Das elektrische Bordnetz ist also hierarchisch in verschiedene Ebenen unterteilt, wie in Abbildung 1 dargestellt.

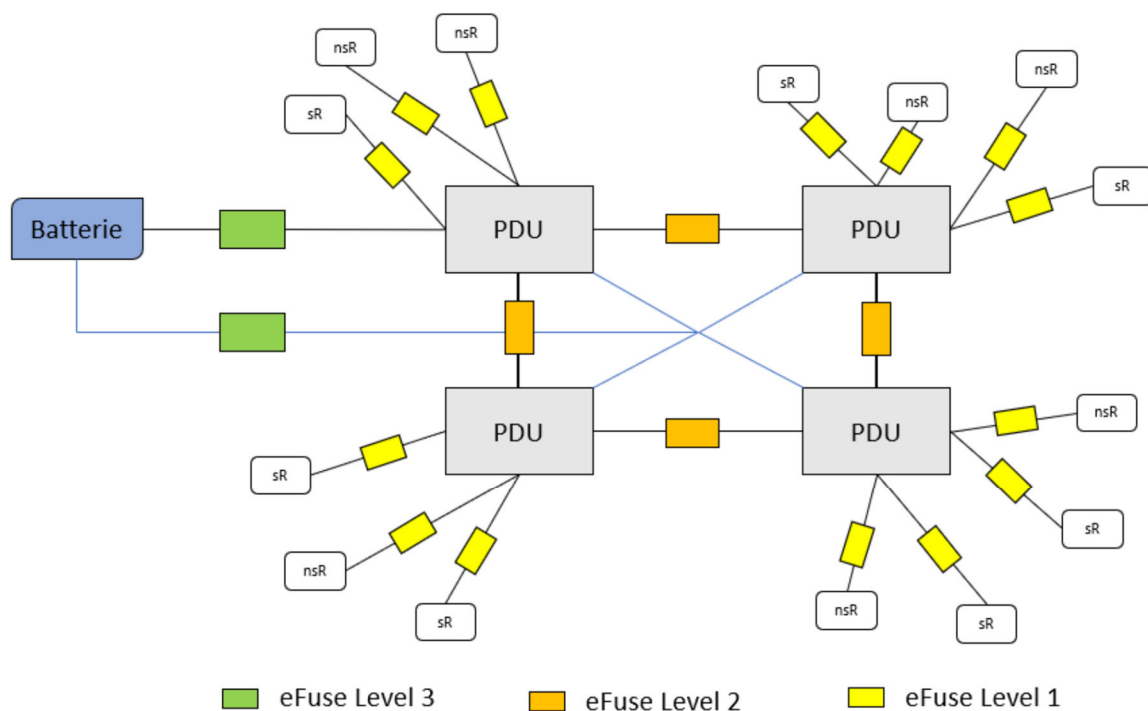


Abbildung 1: Beispielhafte Sicherungsebenen im Bordnetz. eFuses sind der Anschaulichkeit wegen von PDUs getrennt.

Auf den untersten Ebenen sind einzelne Verbraucher mit einer eFuse abgesichert. Die eFuses auf der untersten Ebene können lokal einen Verbraucher im Falle eines Kurzschlusses oder einer Überlast sofort selbstständig abschalten. Dieses Verhalten entspricht im Wesentlichen dem Verhalten der Schmelzsicherung. Hier wird es als „nicht-kommandiertes“ Verhalten bezeichnet, da die Entscheidung zum Abschalten lokal aufgrund der erkannten Überlast am abgesicherten Verbraucher getroffen wird.

Demgegenüber steht das sogenannte „kommandierte“ Verhalten. Hier wird von einer anderen Stelle, beispielsweise einer übergeordneten Sicherungsebene die Abschaltung aktiv ausgelöst. Eine solche Betriebsweise ist mit Schmelzsicherungen nicht möglich, sondern wird erst durch die eFuse umsetzbar. Ein mögliches Szenario für diese Betriebsweise sieht folgendermaßen aus: jeder einzelne Verbraucher arbeitet wie vorgesehen im Nennbereich und hat dabei eine gewisse Leistungsaufnahme. In der Summe überschreiten mehrere Verbraucher aber die Belastungsgrenze der darüberliegenden Sicherungsebene. Diese müsste jetzt abschalten, würde damit aber sämtliche Verbraucher dieses Zweiges im Bordnetz vom Strom trennen.

Eine intelligente Steuerung des Stromverbrauchs könnte demgegenüber auf höherer Ebene eine Überlast erkennen, dann aber entscheiden, welche nicht-sicherheitskritischen Verbraucher mit hoher Last abgeschaltet werden können. Damit lassen sich die sicherheitskritischen Verbraucher am Strom belassen und ein sicherer Betrieb des Fahrzeugs fortsetzen.

Mit ähnlichen Prinzipien und der eFuse als zentralem Element lässt sich auch ein prädiktives Leistungsmanagement im Fahrzeug-Bordnetz umsetzen. Dabei würde nicht erst bei Erreichen einer Überlast auf der höheren Ebene die Abschaltung beginnen. Bereits bei drohender Annäherung an die Lastgrenze, beispielsweise bei 80% der kritischen Last, könnten vorausschauend nicht-sicherheitskritische Verbraucher abgeschaltet werden, um den Fall der Überlast von vornherein zu vermeiden. Auch dazu würden die Strom-Sensoren zur Lasterkennung sowie die Abschaltmöglichkeiten von einzelnen Strompfaden im Bordnetz durch die eFuse genutzt werden. Zusätzlich braucht es Algorithmen mit Heuristiken sowie Informationen zur Kritikalität einzelner Verbraucher um die Entscheidung auf Software-Ebene treffen zu können.

Aufgrund der hohen Systemkomplexität ist die Simulation hierbei ein zentrales Tool, um ein eFuse-basiertes Bordnetz bereits in einer frühen Entwicklungsphase evaluieren und optimieren zu können. Hierfür sind umfassende Modelle notwendig, welche das Verhalten von eFuses und die verschiedenen Wechselwirkungen mit dem Gesamtsystem berücksichtigen [4]. Eine weitere Motivation für die modellbasierte Entwicklung ist hier auch, dass teilweise sehr hohe Leistungen geschaltet werden und damit eine Simulation des Verhaltens ungefährlicher ist als der Test verschiedener Szenarien. In jedem Falle erlaubt eine modellbasierte Entwicklung eine zeit- und kosteneffiziente Untersuchung verschiedener Implementierungsvarianten. Die Optimierung von Betriebsstrategien ist frühzeitig und umfassend möglich. Generell lässt sich die Software einer solchen komplexen Steuerung parallel zur Hardware entwickeln, indem virtuelle Prototypen der Hardware eingesetzt werden. Diese Parallelisierung wird häufig auch als „Shift left“ bezeichnet und führt zu einer deutlichen Zeitersparnis in der Entwicklung und Verifikation. Ein solcher virtueller Prototyp [5] ist auch die Voraussetzung für die Durchführung einer Fehlersimulation. Nur wenn Simulationsmodelle für das nominale Verhalten vorhanden sind, lassen sich dort Fehler einbauen, um simulativ festzustellen, ob die Betriebsstrategien richtig funktionieren und die Sicherheitskonzepte vollständig und rechtzeitig greifen. Ab einem gewissen Sicherheitsniveau sind solche Fehlersimulationen in der Funktionalen Sicherheit (ISO 26262) als ein Mittel zur Absicherung des Entwicklungsprozesses deutlich empfohlen („highly recommended“). Das bedeutet, man kann diese Techniken substituieren, aber es muss („shall“) für einen Ersatz eine schlüssige Begründung gegeben werden.

Mit der eFuse als zentraler Komponente eines künftigen Fahrzeug-Bordnetzes ergeben sich eine Reihe neuer Möglichkeiten für die effiziente Auslegung und den optimalen Betrieb der komplexen Energieversorgung. Dies verlangt aber, die eFuse nicht nur als Ersatz für eine Schmelzsicherung an gleicher Stelle aufzufassen, sondern ihre Potentiale übergreifend über das Bordnetz zu denken und zu implementieren. Dazu bedarf es neuer Entwurfsmethoden, mit denen sehr schnell und effizient eine Vielzahl an Architekturen und Einsatzszenarien untersucht werden können. Diese lassen sich dann hinsichtlich der Erfüllung zentraler Kriterien, beispielsweise aus dem Bereich der funktionalen oder der Sicherheitsanforderungen bewerten und somit optimieren. Die Erforschung solcher Methoden wird im vorliegenden Bericht betrachtet.

Der erste Schwerpunkt befasst sich mit dem nicht-kommandierten Verhalten, also der selbstständigen Reaktion einer eFuse auf Überschreitung von Grenzbedingungen hinsichtlich Strom, Spannung, Temperatur, ... Hier spielen insbesondere die schnelle Erkennung und Reaktion auf solche Ereignisse eine Rolle, während Fehlauslösungen durch Transienten zu vermeiden sind.

Ein zweiter Schwerpunkt sind ganze Einsatzszenarien von Lastsituationen in komplexen Bordnetzen. Dazu werden Konfigurationen mit mehr als 30 eFuses, mehreren PDUs sowie einem zentralen Controller untersucht. Die Signalisierung einer Überlastbedingung an einer eFuse wird an den Controller signalisiert, der gezielt nicht-sicherheitskritische Lasten abschaltet. Damit werden die Grenzbedingungen wieder erfüllt und dabei der Betrieb sicherheitskritischer Lasten erhalten. Zusätzlich wurden Szenarien untersucht, bei denen durch Schalten einer Sicherung kritische Grenzbedingungen an anderer Stelle entstehen können. Mit Hilfe von KI lassen sich kritische von unkritischen Situationen unterscheiden und damit Controller-Entscheidungen unterstützen.

Parallel zu den eigentlichen Anwendungsfällen für den Einsatz von eFuses wurden Methoden für die effiziente Simulation von Fahrzeug-Bordnetzen untersucht. Der dritte Abschnitt des vorliegenden Berichts stellt die Ergebnisse dieser Betrachtungen dar.

2 Nicht-kommandiertes eFuse-Verhalten in komplexen Bordnetzen

Im Rahmen des Arbeitspakets 1 wurde das nicht-kommandierte Verhalten von elektronischen Sicherungen (eFuses) basierend auf Halbleiterschaltern untersucht. Das nicht-kommandierte Verhalten beschreibt hierbei das Eigenverhalten der eFuse, z.B. das Ein- und Ausschaltverhalten, sowie den Auslösealgorithmus bei Überströmen. Hierbei wird die eFuse nicht aktiv angesteuert.

2.1 Messtechnische Untersuchungen elektronischer Sicherungen

Im Folgenden werden die im Projekt durchgeführten messtechnischen Untersuchungen an einer eFuse dargestellt. Die Vorteile des ausgewählten Entwicklungsboards werden herausgestellt und Messungen des nichtkommandierten Verhaltens im Laboraufbau werden analysiert. Der verwendete Algorithmus zum Leitungsschutz wird implementiert und mithilfe der Messungen anhand der genutzten Simulationsmodelle validiert.

2.1.1 Untersuchte eFuse (Infineon BTG7003)

Im Rahmen der Diskussionen im Arbeitskreis wurde eine konkrete elektronische Sicherung ausgewählt. Diese wurde im Labor zur Parametrierung der eFuse-Modelle und zur Validierung anhand von Messergebnissen analysiert. Bei der Sicherung handelt es sich um ein Infineon-IC mit der Bezeichnung PROFET Wire Guard BTG7003 [1] auf einem Entwicklungsboard von Infineon. Abbildung 2 zeigt ein Foto des Entwicklungsboards.

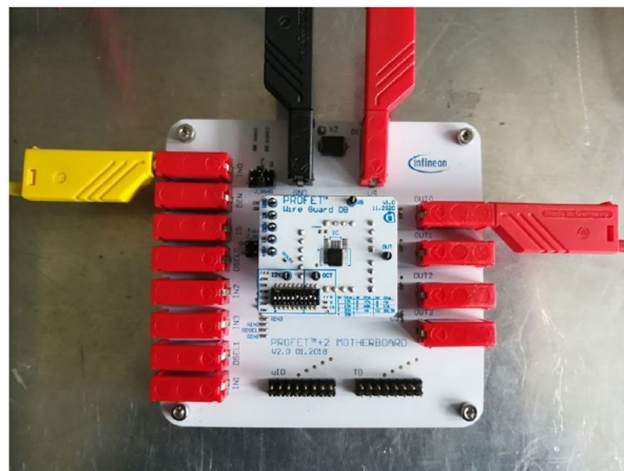


Abbildung 2: Ausgewählte Infineon PROFET Wire Guard BTG7003

Die Sicherung zeichnet sich durch unterschiedliche Diagnosefunktionen, wie eine Strommessung und Berechnung des I^2t -Status zur Bestimmung der thermischen Belastung der Ausgangsleitung aus. Die Implementierung des I^2t -Algorithmus zur Bestimmung der thermischen Leitungsbelastung ist hierbei an die Funktionsweise einer klassischerweise eingesetzten Schmelzsicherung im Fahrzeug angelehnt. Der Algorithmus soll die Leitung zum Steuergerät vor thermischer Überlast absichern und im Falle einer überhöhten Stromaufnahme im Fehlerfall schützen. Außerdem verfügt die Sicherung über grundlegende Schutzfunktionen, wie einer Überstromabschaltung zum Schutz des verwendeten Halbleiterschalters, einer Übertemperaturabschaltung und einem Überspannungsschutz. Der Vorteil des Entwicklungsboards ist hier-

bei, dass die Überstromabschaltung sowie die Grenzwerte zur Berechnung des I^2t -Algorithmus parametrierbar sind. Abbildung 3 zeigt ein Detailfoto, wie die Grenzwerte eingestellt werden können.

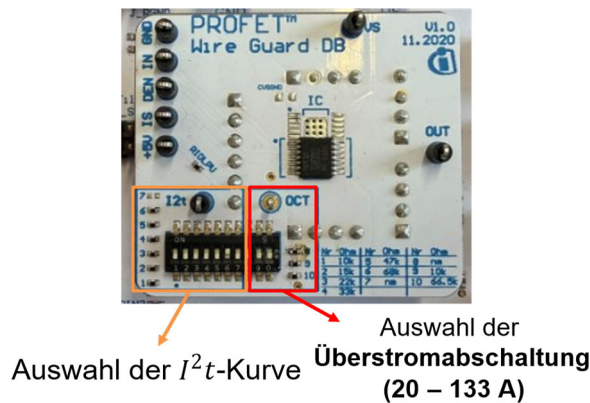


Abbildung 3: Parametrierung der Schutzfunktionen der ausgewählten eFuse

Die Parametrierung erfolgt hier anhand der Auswahl der zugeschalteten Widerstände am Sicherungschip. Für den I^2t -Algorithmus können hierbei mithilfe von 6 hinterlegten Schutzkurven Nominalströme von 9,2 – 19,1 A für verschiedene Steuergeräte ausgewählt werden. Die wesentlich schnellere Überstromabschaltung kann im Bereich von 20 – 133 A realisiert werden. Hierbei schaltet die Sicherung innerhalb weniger μ s direkt ab, um den Halbleiterschalter vor Zerstörung zu schützen.

2.1.2 Validierung des Ein- und Ausschaltverhaltens

Im Vorgängerprojekt „Absicherung moderner Fahrzeug-Bordnetze mittels elektronischer Sicherungen (eFuses)“ wurde ein grundlegendes Simulationsmodell hergeleitet und implementiert. Abbildung 4 zeigt die grundlegenden Funktionsblöcke von eFuses, bzw. dem aufgestellten Simulationsmodell.

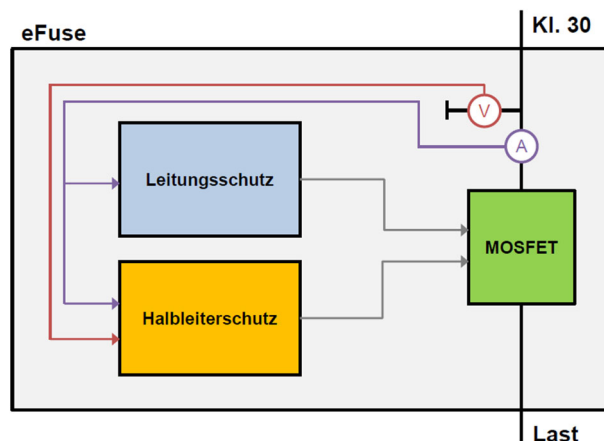


Abbildung 4: Grundlegende Funktionsblöcke elektronischer Sicherungen

Die Hauptfunktionalität der eFuse stellt hierbei der MOSFET dar, der zum Ein- und Ausschalten der Lasten verwendet wird. Eine Strommessfunktionalität wird zur Auswertung der thermischen Last für die Leitungen bereitgestellt. Eine Spannungsmessung der Versorgungsspannung erfolgt in diesem Fall auch. Der Halbleiterschutz stellt hierbei die Überstrom- und Über-temperaturabschaltung dar, die zum Schutz des Halbleiters in der Regel fest implementiert ist.

Der Leitungsschutz kann über verschiedene Algorithmen implementiert sein. Hierfür wird in der Regel der durch die eFuse fließende Strom ausgewertet.

In einem einfachen Messaufbau mit einer rein resistiven Last von $10\ \Omega$ wurden zur Parametrierung des Simulationsmodells zunächst das Ein- und Ausschaltverhalten analysiert. Abbildung 5 zeigt das Ein- und Ausschaltverhalten der eFuse im direkten Vergleich mit den Simulationsergebnissen.

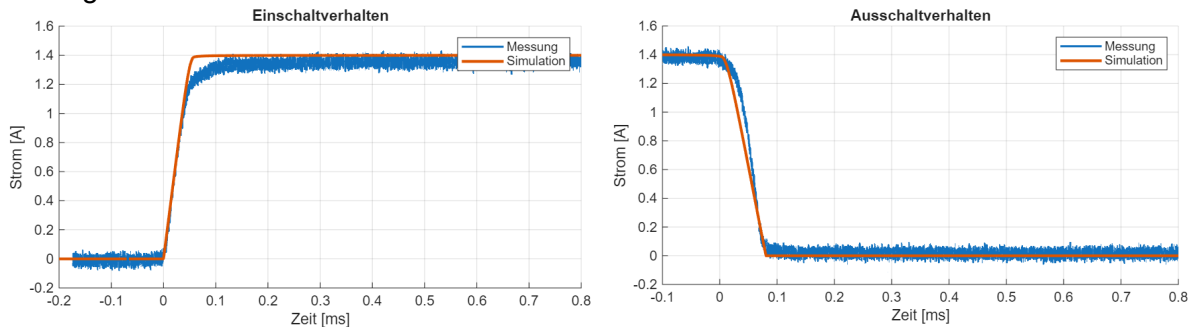


Abbildung 5: Vergleich zwischen Messung und Simulation des parametrierten Simulationsmodells

2.1.3 Validierung des I^2t -Verhaltens

Zum Schutz der an der eFuse angeschlossenen Leitung, verfügt die verwendete eFuse über einen integrierten I^2t -Algorithmus. Abbildung 6 zeigt den Status der I^2t -Berechnung am Diagnoseausgang der Sicherung für unterschiedliche Belastungsströme. Hier wurde der Spannungspegel am Eingang des verwendeten Mikrocontrollers (3,3 V) direkt in den Auslösestatus S_{I^2t} in Prozent umgerechnet. Bei 100 % erfolgt eine Auslösung und mit S_{I^2t} wird ausgedrückt, wie nahe sich die Sicherung an der Auslöseschwelle befindet. Als gewählte Auslösekurve wurde Kurve 6 der Sicherung für einen Nominalstrom eines Steuergerätes von 10,3 A ausgewählt.

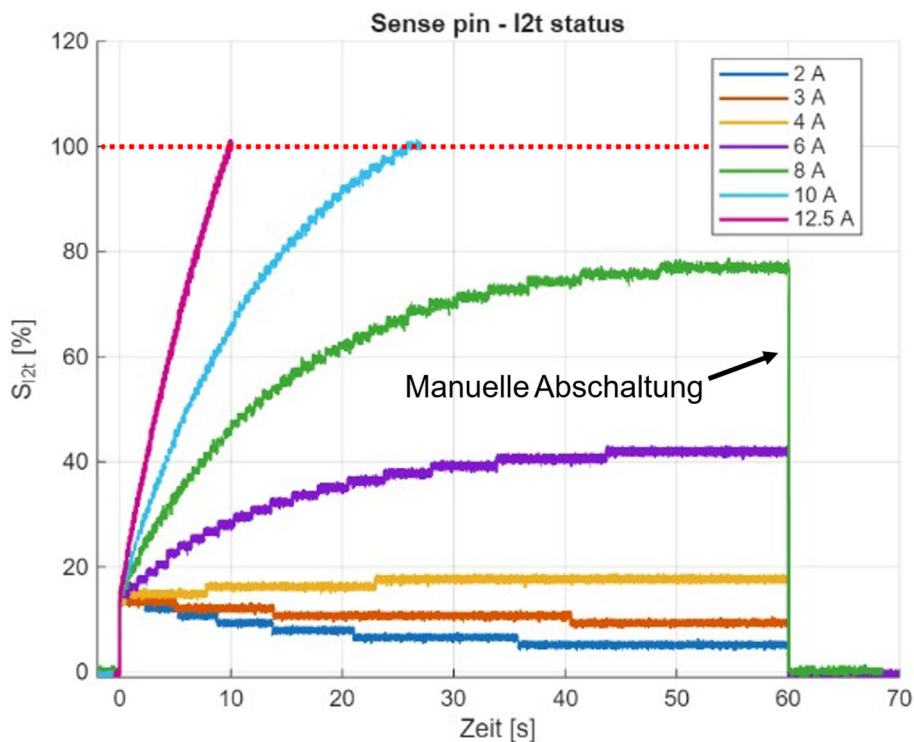


Abbildung 6: Status der I^2t -Berechnung bei unterschiedlichen Belastungsströmen

Es zeigt sich, dass die Berechnung des I^2t -Status, wie im Datenblatt angegeben, erst ab einem Strom von 4 A ausgelöst wird. Bei Erreichen des Status von 100 % löste die Sicherung bei den Belastungsströmen 10 und 12,5 A aus. Die Belastung wurde nach 60 s manuell abgeschaltet.

Die Auslösezeit im Datenblatt ist wie folgt angegeben:

$$t_{\text{auslösen}} = \tau \cdot \ln \left(\frac{I^2 - I_{\text{Offset}}^2}{I^2 - I_{\text{limit}}^2} \right)$$

Hieraus lässt sich nach einer Umformung und Betrachtung der Leistung am MOSFET ein Modellierungsansatz für S_k herleiten:

$$S_k = S_{k-1} \cdot e^{-\frac{\Delta t}{\tau}} + \left(\frac{I_k}{I_{\text{limit}}} \right)^2 \cdot \frac{\Delta t}{\tau}$$

Dieser Ansatz wurde in der Simulation für den Leitungsschutzblock implementiert. Der erste Summand des Modellierungsansatz beschreibt hierbei die Abkühlung der Leitung vom vorherigem Status (S_{k-1}). Der zweite Summand beschreibt die Temperaturerhöhung durch den durch die eFuse fließenden Strom relativ zum eingestellten Limitstrom. Der Limitstrom kann anhand der unterschiedlichen implementierten I^2t -Kurven ausgewählt werden. Eine erste Validierung des Modellierungsansatzes mit den typischen Datenblattangaben für $\tau = 15,1$ s und $I_{\text{limit}} = 10,3$ A zeigte keine gute Übereinstimmung zwischen Messung und Simulation. Allerdings werden auch große Parameterbereiche für die Minimal-, sowie Maximalwerte im Datenblatt gegeben. Die thermische Zeitkonstante ist für ein Intervall $\tau \in [10,6; 19,6]$ s angegeben. Der Strom für die ausgewählte Absicherungskurve im Intervall $I_{\text{limit}} \in [9,2; 11,3]$ A. In diesen angegebenen Intervallen wurde mithilfe der Matlab „Optimization Toolbox“ eine Parameteroptimierung durchgeführt. Abbildung 7 zeigt den Vergleich zwischen den Simulations- und Messergebnissen mit den optimierten Parametern $\tau = 15,35$ s und $I_{\text{limit}} = 9,3$ A.

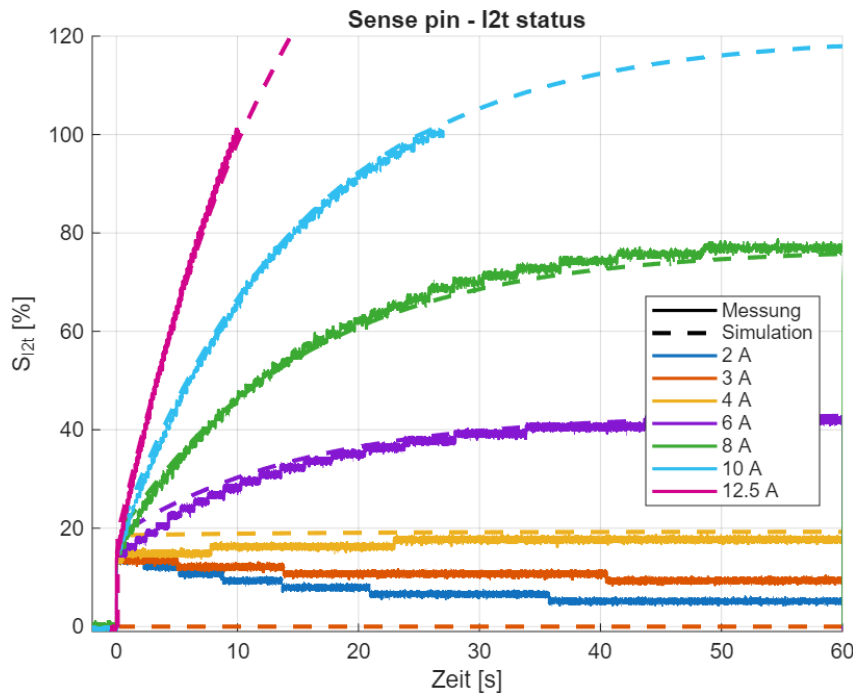


Abbildung 7: Vergleich zwischen Messung und Simulation des parametrierten Simulationsmodells

Die optimierten Parameter liegen im Intervall, das für die eFuse im Datenblatt angegeben ist. Dennoch ist zu beobachten, dass die Werte im Datenblatt stark streuen und somit die Auslösezeiten der eFuses stark variieren können.

2.2 Analyse der Sensoranforderungen für erweiterte Diagnosefunktionen

Um im nicht-kommandierten Verhalten der eFuse aufgrund der internen Temperaturabschaltung durch einen Fehler im Energiebordnetz nicht zu früh abzuschalten, wurde eine Eigendiagnose basierend auf der Sensorik elektronischer Sicherungen analysiert. Der Einschaltwiderstand der eFuse kann hierbei aufgrund von Temperaturunterschieden, sowie Alterungsvorgängen beeinflusst werden. Dies äußert sich in erhöhten Verlusten innerhalb des IC's. Im Vorgängerprojekt „Diagnosekonzepte für zonale und teilredundante Bordnetzarchitekturen“ [2] wurde bereits eine Diagnose der Eingangskapazität von Steuergeräten mit KI-Methoden untersucht. Darauf aufbauend wurde die Eigendiagnose der eFuse analysiert.

2.2.1 Sensorkonzept zur Bestimmung des Einschaltwiderstands

Elektronische Sicherungen besitzen aufgrund ihrer zusätzlichen Sensoren Potential um die Diagnose im Fahrzeug zu erweitern. In der Regel verfügen eFuses über verschiedene Spannungs- und Stromsensoren [3]. Abbildung 8 zeigt eine grundsätzlich vorhandene Sensoranordnung einer einzelnen eFuse innerhalb einer Energieverteilereinheit (Power Distribution Unit) im Fahrzeug.

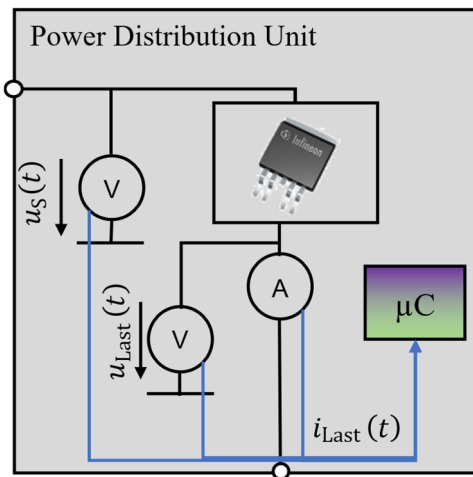


Abbildung 8: Sensorkonzept zur Bestimmung des Einschaltwiderstands

Grundlegend wird bei jeder eFuse der Strom durch die Sicherung $i_{Last}(t)$ gemessen. Auf Basis dieser Messung ist in der Regel der thermische Leitungsschutz implementiert. Zusätzlich dazu verfügen die meisten eFuses über eine Messschaltung zur Messung der Versorgungsspannung $u_S(t)$, bzw. der Ausgangsspannung $u_{Last}(t)$. In dem Sicherungs-IC wird häufig auch die Temperatur überwacht, um aufgrund von thermischen Verlusten rechtzeitig den Halbleiterschalter zu schützen. Ein Problem hierbei kann sein, dass sich dieser Einschaltwiderstand $R_{DS,on}$ aufgrund von Alterung und Temperaturunterschieden verändern kann. Um ein ungewolltes Abschalten im nicht-kommandierten Betrieb zu vermeiden, wird analysiert, welche Sensorqualität nötig ist, um diese Änderungen im Einschaltwiderstand korrekt zu diagnostizieren.

Analytisch lässt sich der Einschaltwiderstand theoretisch direkt aus den gemessenen Signalen bestimmen:

$$R_{DS,on} = \frac{U_{PDU} - U_{Last}}{I_{Last}} = \frac{U_{eFuse}}{I_{Last}} \quad (1)$$

Das Problem hierbei ist, dass die Werte des Einschaltwiderstands im Bereich weniger Milliohm liegen ($\sim 0,5\text{--}5\text{ m}\Omega$). Dementsprechend wird zur korrekten Diagnose eine hohe Sensorauflösung benötigt. Außerdem sind sowohl Temperaturabhängigkeiten als auch Parameterdrifts aufgrund der Alterung zu berücksichtigen. Um den Einschaltwiderstand zu bestimmen, wird

von einem stationären Zustand ausgegangen, in dem der Laststrom des dahinter liegenden Steuergeräts konstant ist.

2.2.2 Modellierung der Alterungs- und Temperaturabhängigkeit

Um das in diesem Projekt erarbeitete Konzept zu erproben, wurden einfache Alterungs- und Temperaturabhängigkeitsmodelle verwendet. Die Temperaturabhängigkeit wurde anhand der ausgewählten Infineon eFuse BTG7003 wie folgt modelliert:

$$R_{DS,on}(T) \approx R_{DS,on,25}(1 + \alpha(T - T_0)) \quad (2)$$

Im Datenblatt sind für die eFuse die Werte bei 25°C als typischer Wert mit 3,6 mΩ und der maximale Einschaltwiderstand ist mit 6,6 mΩ bei 150°C angegeben. Da es sich hier um typische und maximale Werte handelt, wird eine Worst-Case Abschätzung mit:

$$\alpha \approx 6,67 \cdot 10^{-3} \frac{1}{K}$$

durchgeführt. Der Wert wurde aus Gleichung 1 direkt über die Datenblattangaben bestimmt. Um die Temperatur im MOSFET selbst zu bestimmen, wurde hierfür der Laststrom im Bereich von 0 – 19,1 A angenommen. Dies entspricht dem Nennbereich der ausgewählten Sicherung. Aus dem Datenblatt wurde der thermische Widerstand $R_{thJA} = 30,1 \text{ K/W}$ verwendet, um die interne MOSFET Temperatur zu berechnen. Zunächst kann die Verlustleistung im MOSFET bestimmt werden:

$$P = I^2 \cdot R_{DS,on} \quad (3)$$

Aufgrund der Verlustleistung erhöht sich die Junction-Temperatur im Vergleich zur Umgebungstemperatur wie folgt:

$$T_j = T_{amb} + P \cdot R_{thJA} \quad (4)$$

Hierdurch ergibt sich eine neue Junction-Temperatur des MOSFETs. Diese fließt iterativ wieder in Gleichung 3 ein. Sobald ein stationärer Zustand innerhalb eines Toleranzlimits bestimmt wird, wird die Berechnung abgebrochen. Abbildung 9 zeigt sowohl den Einschaltwiderstand in Abhängigkeit des berücksichtigten Laststroms als auch die interne Junction-Temperatur des MOSFETs mit diesem Modellierungsansatz für verschiedene Umgebungstemperaturen.

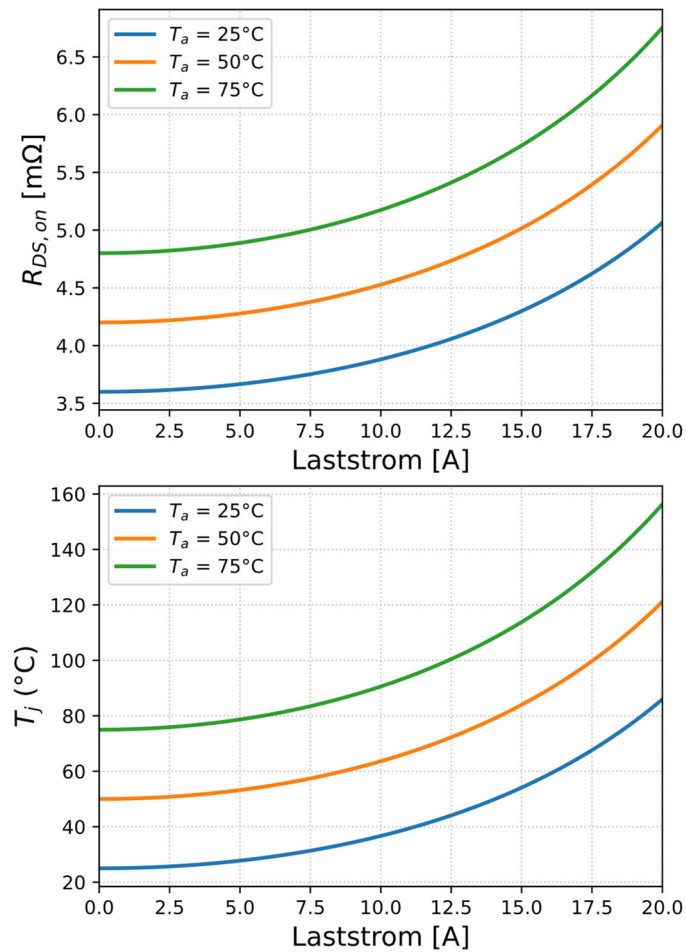


Abbildung 9: Modellierung des Temperaturverhaltens in Abhängigkeit des stationären Dauerstroms

Die Parametrierung von Alterungsmodellen ist häufig mit aufwendigen Messungen verbunden und benötigen in der Regel eine lange Betriebszeit bei erhöhten Temperaturbedingungen [4]. Um lange Alterungsexperimente zu vermeiden und grundlegend das Konzept zur Eigendiagnose in Abhängigkeit dieser Einflüsse zu analysieren, wurde ein einfaches Alterungsmodell verwendet:

$$R_{DS,on,aged} = R_{DS,0} \cdot (1 + \beta \cdot t) \quad (5)$$

Hierbei wurde angenommen, dass das Ende der Lebenszeit bei einer Erhöhung des Einschaltwiderstands von 20 % ($\beta = 0,2$) gegenüber dem initialen Zustand am Anfang seiner Betriebszeit erreicht wird. Abbildung 10 zeigt beispielhaft das modellierte Alterungsverhalten über die Betriebsstunden.

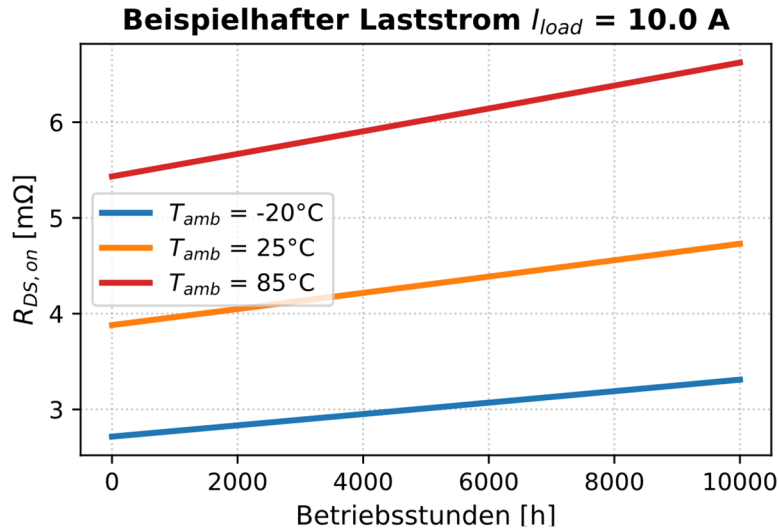


Abbildung 10: Modellierung des Alterungsverhaltens

In der folgenden Untersuchung wird die Zeit nur noch im Intervall $t \in [0 \dots 1]$ betrachtet. Hierbei entspricht 0 einem neuen Bauteil und 1 dem Ende der Lebenszeit.

2.2.3 Vorabschätzung mit Worst-Case Annahmen

Um die notwendige Sensorqualität zu bestimmen, wurde eine Vorabschätzung durchgeführt. Hierfür wurde sich an üblichen Messschaltungen von eFuses orientiert. Es wurde keine weitere Optimierung für den Anwendungsfall vorgenommen um zu analysieren, wie eine möglichst ressourcenarme Integration in einer Energieverteilereinheit realisiert werden kann. Es wurde somit von keinem zusätzlichen Hardwareaufwand zur Optimierung ausgegangen. Hierfür wurde ein Messbereich von 0 – 25 V, sowie ein Strommessbereich von 0 – 130 A angenommen. Wie in Kapitel 2.2.1 dargestellt, sind hierfür drei Sensoren notwendig. Somit lässt sich der relative Gesamtfehler für verschiedene Quantisierungen der Sensoren abschätzen mit:

$$e_{ges} \approx \frac{2 \cdot LSB_U}{\Delta U} + \frac{LSB_I}{I} \quad (6)$$

Hierbei werden im ersten Summanden 2 Spannungsmessungen relativ zur Differenzspannung an der eFuse betrachtet sowie die Strommessung der eFuse relativ zum Strommessbereich. Entscheidend ist hierbei die kleinste Quantisierungsstufe (LSB, Least Significant Byte). Abbildung 11 zeigt verschiedene Quantisierungsaufösungen von 8 – 16 Bit und den relativen Maximalfehler in Abhängigkeit der Lastströme.

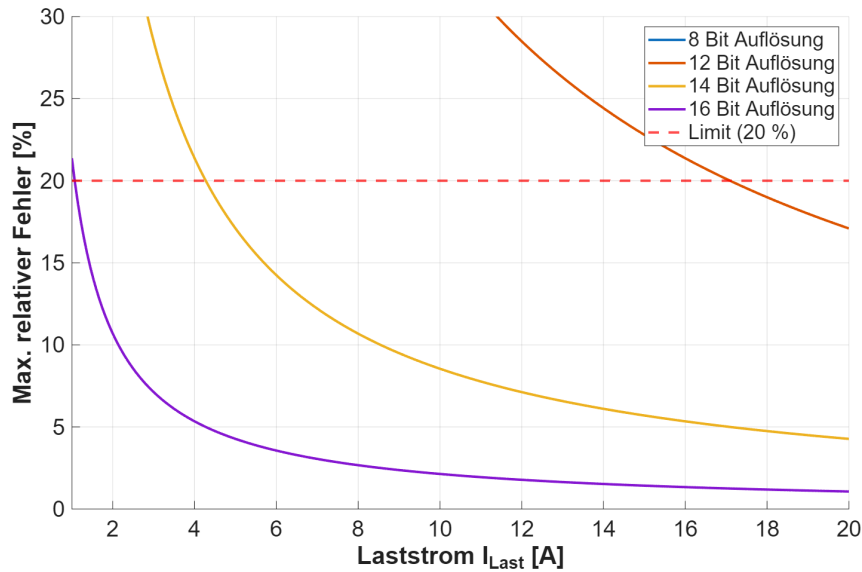


Abbildung 11: Worst-Case Abschätzung der notwendigen Quantisierung der Sensorik

Wenn der Laststrom durch die eFuse größer wird, steigt der Spannungsabfall an der eFuse. Dementsprechend ist es wesentlich einfacher, den Einschaltwiderstand bei möglichst großen Strömen zu bestimmen. Die Worst-Case Abschätzung hat gezeigt, dass eine Sensorauflösung von mindestens 14 bit notwendig ist, um den Einschaltwiderstand mit einem maximalen Fehler unter 20 % zu bestimmen.

2.2.4 Analyse des Diagnosekonzepts zur Bestimmung des Einschaltwiderstands

Um die verschiedenen Einflüsse, wie Temperatur und Alterungsverhalten klar zu trennen, wurden mithilfe der aufgestellten Modelle ein Trainings-, Test und Validierungsdatensatz von 10.000 Daten generiert. Die Diagnose erfolgt mit einem Feed Forward neuronalem Netz. Es wurde eine Netzstruktur mit 3 Layern und 256 Neuronen pro Layer verwendet. Als Aktivierungsfunktion wurde die SiLU (Sigmoid Linear Unit) verwendet. Eine Sigmoid-Funktion am Ausgang begrenzt die Ausgangsgröße, den State of Health (SoH), zwischen 0 und 1.

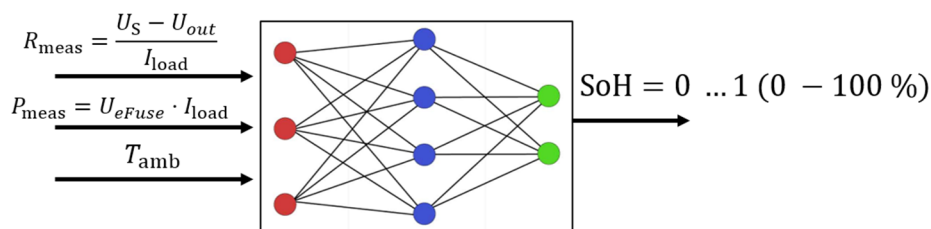


Abbildung 12: Ein- und Ausgangsgrößen zur Diagnose des Alterungszustands

Um den Alterungszustand zu erhalten, wird somit der gemessene Widerstand aus den Messgrößen bestimmt, sowie die Leistung, die sich aus gemessener Spannung und Laststrom ergibt. Die Umgebungstemperatur wird als Referenz genutzt. Als Ausgangsgröße wird der aktuelle Alterungszustand/SoH ausgewertet.

Eine erste Analyse der Genauigkeit des trainierten Modells auf Basis der Daten wurde mit überlagerten Rauscheinflüssen durchgeführt. Hier wurden die Spannungen mit gleichverteiltem Rauschen von 1 mV überlagert und die Stromwerte mit 10 mA, da die Strommessung der eFuses in der Regel wesentlich ungenauer als eine Spannungsmessung ist.

Abbildung 13 zeigt zunächst einen Soll- vs. Ist-Vergleich zwischen den tatsächlichen Alterungszuständen und den vom Modell angenäherten Alterungszuständen. Der untere Teil der Abbildung zeigt den Fehler des Modells in Abhängigkeit des Laststroms.

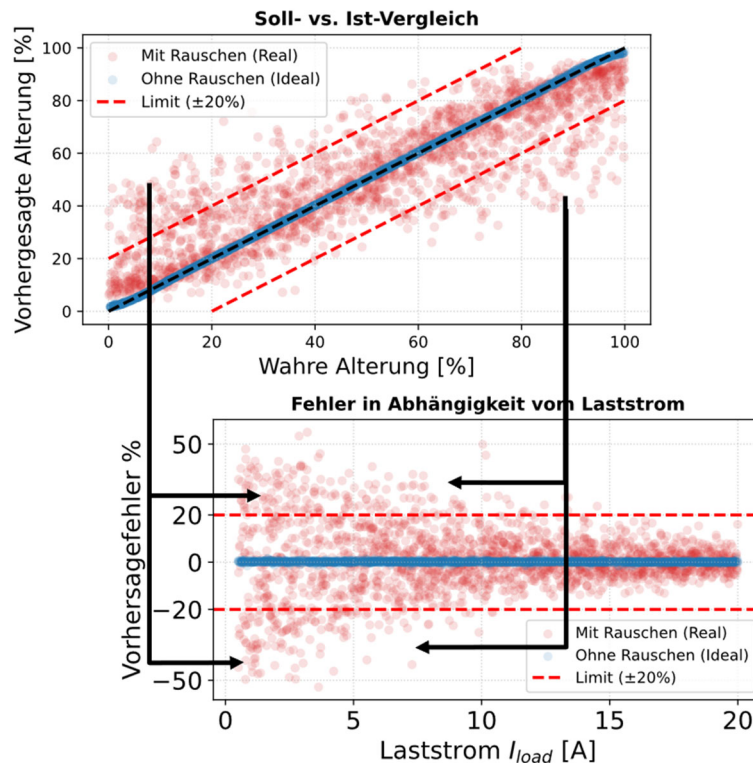


Abbildung 13: Analysen der Alterungsbestimmung

Zunächst kann festgestellt werden, dass das ohne Störungen getestete Modell sehr hohe Genauigkeiten liefert. Die maximale Abweichung beträgt hier 2,24 %. Es kann außerdem gesehen werden, dass bereits bei dieser Überlagerung von Störungen die Parameterregression des Netzes stark beeinflusst wird. Die Abweichungen über dem angenommenen 20 % Genauigkeitslimit sind vor allem in dem unteren Teil der Abbildung zu erkennen. Hier ist wieder ein ähnliches Bild, wie bei der Worst-Case Abschätzung zu sehen. Bei geringen Lastströmen ist die Diagnosegenauigkeit wesentlich schlechter als bei hohen Lastströmen. Um die Worst-Case Abschätzung zu validieren, wurde eine Quantisierung von 16 bit für die Auflösung eines Analog-Digital-Wandlers angenommen. Die Testdaten wurden hierbei mit einem Quantisierungsrauschen für die angenommene Sensordynamik und der Quantisierung von 16 bit mit normalverteiltem Rauschen des LSB überlagert. Abbildung 14 zeigt den Fehler in Abhängigkeit vom Laststrom.

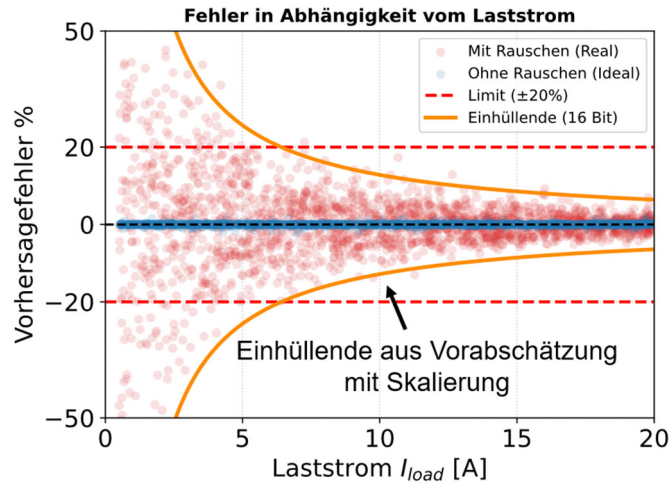


Abbildung 14: Analysen der Alterungsbestimmung

Die Einhüllende wurde aufgrund der vorherigen Worst-Case Abschätzung, die direkt den Fehler der direkten Bestimmung des Widerstandswerts abgeschätzt hat, entsprechend für die relative Auswertung des Alterungszustands mit dem Faktor 6 skaliert:

$$\Delta t = \left(\frac{1}{\beta} \cdot \frac{R_{DS,on,aged}}{R_{DS,0}} \right) \cdot \frac{\Delta R_{DS,on,aged}}{R_{DS,on,aged}} \Leftrightarrow S = \frac{1}{\beta} \cdot \frac{R_{DS,on,aged}}{R_{DS,0}} = \frac{1}{0,2} \cdot 1,2 = 6 \quad (7)$$

Insgesamt deckt sich die Worst-Case Abschätzung sehr gut mit den aufgestellten Analysen, sodass für weitere Untersuchungen hiermit grundlegend untersucht werden kann, wie die Sensorik für diesen Anwendungsfall ausgewählt werden muss. In Anbetracht des zusätzlichen Hardwareaufwands müssen allerdings zunächst Konzepte zur besseren Signalverarbeitung erarbeitet werden, um die Spannungsdifferenz an der Sicherung genauer zu bestimmen. In diesem Projekt wurde zunächst ein Ansatz mit möglichst wenig zusätzlichem Hardwareaufwand verfolgt. Insgesamt eignen sich die aufgestellte datenbasierte Methodik, um Temperatureinflüsse zu berücksichtigen und damit den Einschaltwiderstand der Sicherung zu bestimmen.

3 Kommandiertes eFuse-Verhalten in komplexen Bordnetzen

In modernen Bordnetzen mit wachsender Anzahl elektrischer Verbraucher gewinnt ein koordiniertes Leistungsmanagement zunehmend an Bedeutung. Insbesondere bei hochautomatisierten Fahrzeugen muss sichergestellt werden, dass sicherheitsrelevante Funktionen auch unter ungünstigen Bedingungen zuverlässig versorgt bleiben. Vor diesem Hintergrund wird im vorliegenden Projekt ein kommandiertes eFuse-Verhalten untersucht, bei dem die Abschaltentscheidung nicht ausschließlich lokal in der Sicherung erfolgt, sondern durch eine übergeordnete Steuerungseinheit beeinflusst wird. Durch eine zentrale Logik kann der Bordnetzzustand beurteilt werden. Interessante Metriken können hier zum Beispiel die Spannungsstabilität oder die thermische Belastung des Systems sein. Durch geeignete Maßnahmen kann anschließend das System stabilisiert werden. Dies ist schematisch in Abbildung 15 dargestellt. Dieses Arbeitspakets fokussiert sich dabei auf das konkrete Ziel der Spannungsstabilität. Durch ein zentrales Lastmanagement soll im Notfall aktiv in die Energieverteilung eingegriffen werden, um drohende Unterspannungen zu verhindern. Ein solcher Eingriff kann vor allem die gezielte Abschaltung von Komfortverbraucher oder deren Leistungsreduktion sein. Auf diese Weise wird die Spannungsstabilität für kritische Lasten gewährleistet. Das Bordnetz muss dazu zentral überwacht werden, was durch die Messfunktionalität der eFuses ermöglicht wird. Als Ergebnis der zentralen Auswertung der vorhandenen Messdaten werden anschließend die Schaltzustände der eFuses basierend auf einer regelbasierten Entscheidungslogik angepasst.

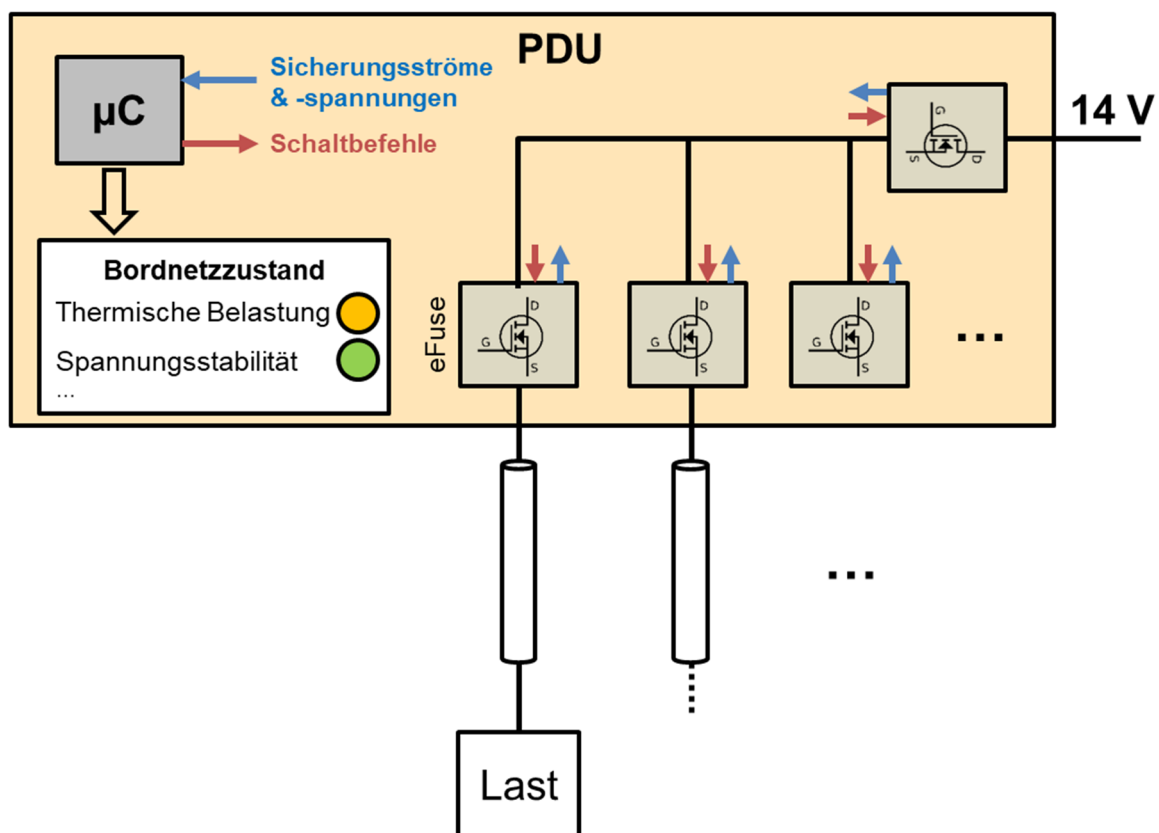


Abbildung 15: Beispielhafte schematische Darstellung eines kommandierten Sicherungsbetriebs im Rahmen einer zentralisierten Bordnetzüberwachung

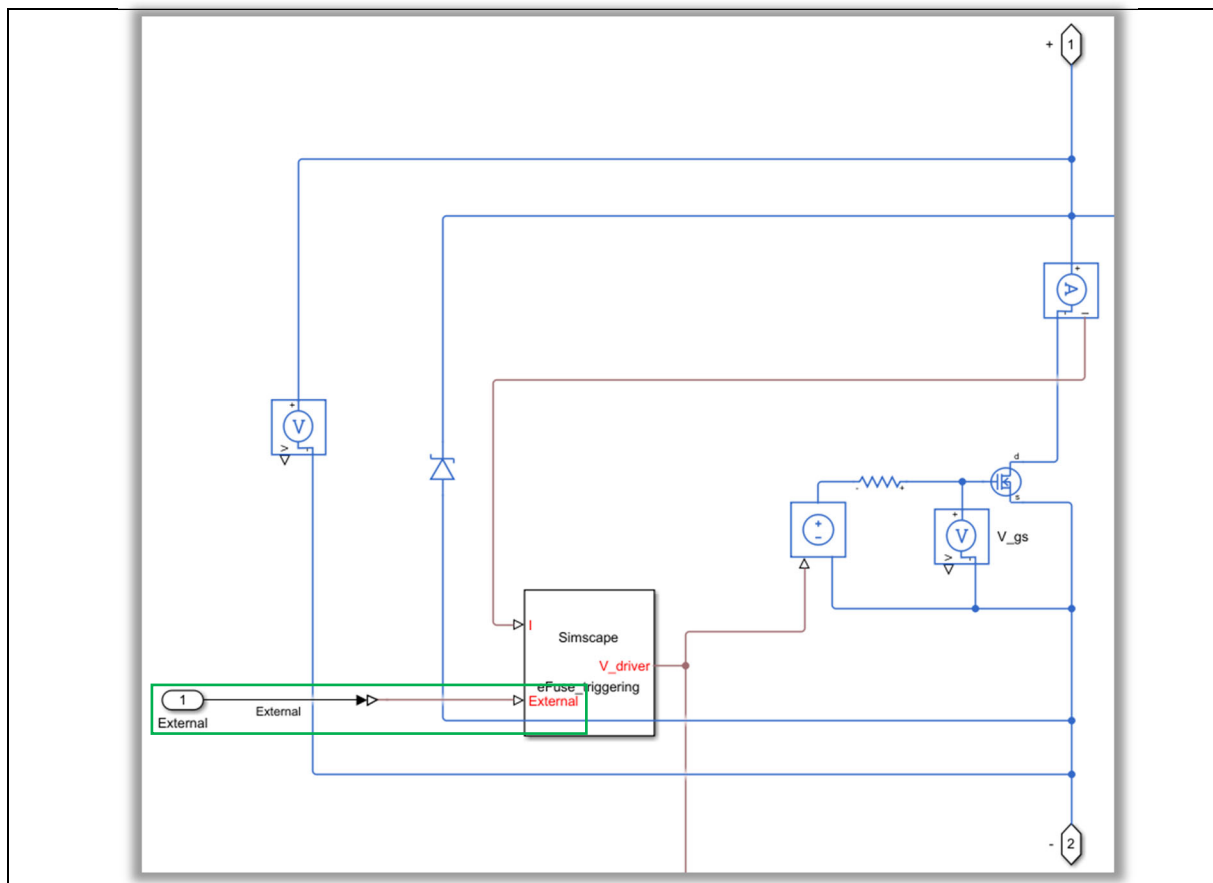
In Kapitel 3.1 wird dafür zunächst das aus dem vorherigen Projekt vorhandene eFuse-Modell erweitert, um die für den kommandierten Betrieb notwendige externe Ansteuerung zu ermöglichen. Anschließend wird in Kapitel 3.2 der entwickelte Leistungsmanagement-Algorithmus im Detail erklärt. Eine beispielhafte Anwendung mit verschiedenen Testszenarien folgt in Kapitel 3.3.

3.1 Erweiterung des eFuse-Modells zur externen Ansteuerung

Um das beschriebene Konzept umzusetzen, wird zunächst das bestehende eFuse-Modell erweitert. In der ursprünglichen Form, hervorgegangen aus dem vorangegangenen Projekt, basiert das Verhalten der eFuse ausschließlich auf lokalen Messgrößen, insbesondere dem Strom durch den jeweiligen Sicherungspfad. Für ein übergeordnetes Lastmanagement ist es jedoch erforderlich, zusätzlich externe Steuersignale berücksichtigen zu können.

Daher wird das Modell um eine externe Ansteuerung ergänzt, die es erlaubt, den Schaltzustand unabhängig von den internen Schutzmechanismen zu beeinflussen. Konkret wird die Gate-Driver-Logik der eFuse so erweitert, dass das Schaltsignal sowohl durch die klassischen Triggerbedingungen, also nicht-kommandiert, aber auch durch ein externes Kommando gesetzt werden kann. Dabei bleibt die interne Schutzfunktion vollständig erhalten; ein Überstrom im Sicherungspfad führt weiterhin autonom zum Abschalten, während die zentrale Steuerung lediglich eine zusätzliche Eingriffsmöglichkeit darstellt.

Ein Screenshot einer Simscape-Implementierung des erweiterten Sicherungsmodells ist in Abbildung 16 dargestellt. Die eigentliche Auslöselogik, die das Gatesignal bestimmt, ist codebasiert implementiert und ebenfalls in Abbildung 16 mit der entsprechenden Erweiterung dargestellt. Auf diese Weise entsteht ein flexibles eFuse-Modell, das sowohl Schutz- als auch Steuerungsfunktionen erfüllt.



```

events
    when edge(I >= I_OCP)
        violation_time_OCP = time;
        state_OCP = 0;
    elseif edge(I < I_OCP)
        state_OCP = 1;
    elseif edge(I >= I_trigger)
        violation_time_normal = time;
        state_normal = 0;
    elseif edge(I < I_trigger)
        state_normal = 1;
    end
    when edge(external <= 0.5)
        V_driver_event = {0, 'V'};
    elseif edge(external > 0.5) && state_OCP==1)
        V_driver_event = {1, 'V'};
    elseif edge(state_OCP==0 && ((time-violation_time_OCP) >= t_OCP)))
        V_driver_event = {0, 'V'};
    elseif edge(state_normal==0 && ((time-violation_time_normal) >= trigger_delay)))
        V_driver_event = {0, 'V'};
    end
end

equations
    V_driver == V_driver_event;
end
    
```

Abbildung 16: Erweiterung des bestehenden eFuse-Modells. Simscape-Blockschaltbild mit externem Schaltsignal (oben) und erweiterte Auslöselogik (unten)

3.2 Entwicklung eines zentralen Lastmanagement-Algorithmus

Aufbauend auf dem erweiterten eFuse-Modell wird nun ein zentraler Lastmanagement-Algorithmus entwickelt. Dieser verarbeitet die im Bordnetz verfügbaren Messgrößen und leitet daraus geeignete Maßnahmen zur Stabilisierung der Versorgung ab. Eingangsdaten sind insbesondere die von den eFuses gemessenen Ströme. Auch die Spannungen der einzelnen Lasten werden als bekannt angenommen. Damit ergibt sich die in Abbildung 17 abgebildete allgemeine Struktur des Lastmanagements.

Das grundlegende Ziel besteht darin, bei drohender Unterspannung mit minimalem Eingriff in den Komfort eine maximale Stabilisierung der sicherheitsrelevanten Verbraucher zu erreichen. Hierzu wird ein regelbasierter Ansatz gewählt, der sowohl die Wichtigkeit einzelner Verbraucher als auch die Wirksamkeit möglicher Maßnahmen berücksichtigt und anschließend die Entscheidung mit optimalem Kosten-/Nutzen-Verhältnis trifft. Wie genau Kosten und Nutzen durch den Algorithmus abgeschätzt werden und welche weiteren Aspekte beachtet werden müssen, wird im Folgenden erklärt.

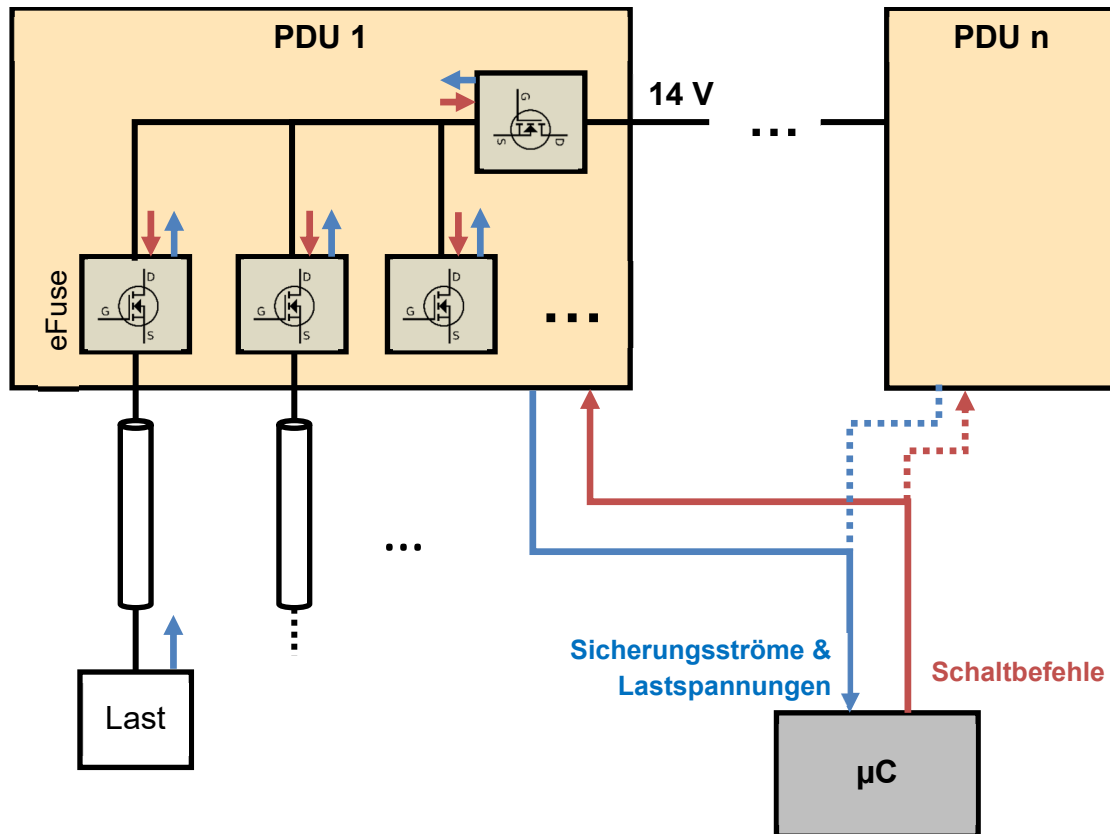


Abbildung 17: Schematische Darstellung des zentralen Lastmanagements in elektronisch abgesicherten Bordnetzen

3.2.1 Prioritätsklassen

Ein erster wesentlicher Bestandteil des Algorithmus ist die Einteilung der Verbraucher in Prioritätsklassen. Diese spiegeln wider, wie kritisch eine Abschaltung oder Leistungsreduktion für den Fahrbetrieb ist. Komfortverbraucher erhalten dabei eine niedrigere Priorität als sicherheitsrelevante Lasten. Wenn nicht anders definiert, wird für eine beliebige Komfortlast i eine Default-Priorität von $P_i = 1$ verwendet.

Das Konzept kann darüber hinaus auch erweitert werden, um eine weitere Differenzierung der Kostengewichtung zu ermöglichen. Denkbar ist hier beispielsweise die Berücksichtigung von Kriterien wie der maximal zulässigen Abschaltdauer, das Wiederanlaufverhalten oder die Wirkzeit, die vergeht, bis der Nutzer eine Abschaltung des Verbrauchers bemerkt.

3.2.2 Wirksamkeitsbeurteilung

Neben der Priorität spielt die Wirksamkeit einer Maßnahme eine zentrale Rolle. Es soll sichergestellt werden, dass eine Maßnahme auch wirklich zu einer signifikanten Verbesserung der Versorgungsstabilität der kritischen Verbraucher führt. Auf Basis der bekannten Sicherungsströme kann die Wirkung eines Schaltvorgangs auf das System abgeschätzt werden. Mit dem Sicherungsstrom eines abzuschaltenden Verbrauchers ist auch dessen zu erwartende Stromänderung ΔI bekannt. Über die ebenfalls bekannten Leitungswiderstände kann somit die Spannungsänderung ΔU im System bestimmt werden.

Abbildung 18 zeigt dies an einem einfachen Beispiel. Soll Last 2 stabilisiert werden, indem Last 1 abgeschaltet wird, beträgt $\Delta I = I_{\text{load},1}$. Der Betrag des Spannungsanstiegs der Last 2 ergibt sich aufgrund des Widerstands $R_{\text{wire},1}$ der gemeinsamen Versorgungsleitung damit in erster Näherung zu $\Delta U = \Delta I \cdot R_{\text{wire},1}$. Soll dagegen Last 1 stabilisiert werden, indem Last 2

abgeschaltet wird, beträgt $\Delta I = I_{\text{load},2}$. Der Betrag des Spannungsanstiegs der Last 1 ergibt sich dann ebenfalls zu $\Delta U = \Delta I \cdot R_{\text{wire},1}$.

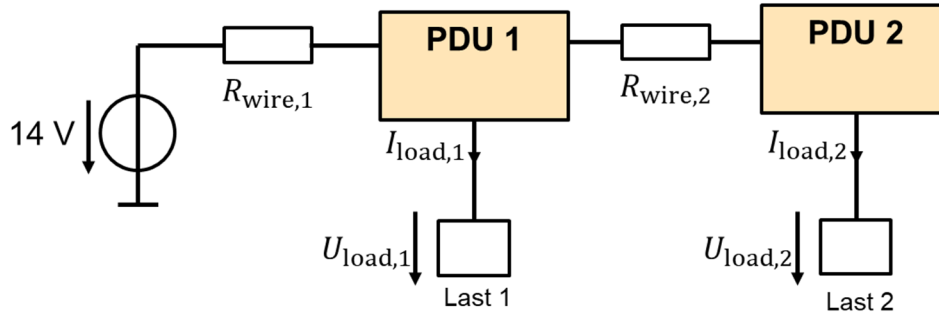


Abbildung 18: Minimalbeispiel zur Verdeutlichung der Wirksamkeitsbeurteilung

Verallgemeinert ist also der Widerstand der Versorgungsleitungen relevant, über die die verschiedenen PDUs miteinander gekoppelt sind. Diese müssen bestimmt werden und ergeben sich in Form einer Matrix zu:

$$\mathbf{R} = (R_{\text{PDU},i \rightarrow \text{PDU},j})_{i,j}, \quad i, j \in \{1, \dots, N\} \quad (8)$$

N bezeichnet hierbei die Anzahl der PDUs. Damit ist $R_{\text{PDU}1 \rightarrow \text{PDU}2}$ beispielsweise proportional zu der Spannungsänderung, die eine von PDU 2 versorgte Last erfährt, wenn eine Last von PDU 1 ihren Strom ändert:

$$\Delta V_{\text{load},2} = I_{\text{load},1} \cdot R_{\text{PDU}1 \rightarrow \text{PDU}2} \quad (9)$$

Die Widerstandsmatrix bildet somit die gegenseitige Beeinflussung der einzelnen Teilbordnetze ab und erlaubt eine systemweite Bewertung einzelner Maßnahmen. Somit wird nicht nur der lokale Effekt einer Abschaltung berücksichtigt, sondern auch deren Auswirkung auf andere Netzbereiche. In Kombination mit den Lastprioritäten, d. h. den Kosten eines solchen Eingriffs, kann im Falle einer drohenden Unterspannung die effektivste Stabilisierungsmaßnahme ausgewählt werden.

3.2.3 Schwellenwerte und Hysterese

Um den Eingriffszeitpunkt festzulegen, müssen Spannungsgrenzwerte definiert werden. Das Lastmanagement wird aktiv, sobald die Spannung einer sicherheitsrelevanten Last unter einen Warnschwellwert U_{lim} fällt. Dieser liegt bewusst geringfügig oberhalb der tatsächlichen kritischen Unterspannungsgrenze U_{crit} , um die elektrischen Zeitkonstanten des Systems zu kompensieren. Würde eine Lastmanagement-Maßnahme erst bei Erreichen von U_{crit} eingeleitet, würde die verzögerte Wirkung erst eintreten, wenn die Spannung sich bereits im kritischen Bereich befindet. Konkret wird im Folgenden von einer beispielhaften Unterspannungsgrenze $U_{\text{crit}} = 9 \text{ V}$ ausgegangen. Das Lastmanagement wird dann bereits bei Unterschreitung einer Spannung von $U_{\text{lim}} = 9,3 \text{ V}$ aktiv.

Ergänzend ist eine Hysterese sinnvoll, wenn zuvor abgeschaltete Verbraucher wieder eingeschaltet werden sollen. Dies verhindert, dass das System bei kleinen Spannungsschwankungen wiederholt zwischen verschiedenen Schaltzuständen wechselt. Ein stabiles Verhalten ohne unnötige Schaltvorgänge wird so erzielt.

3.2.4 Stufenweise Leistungsreduktion

Neben dem vollständigen Abschalten von Verbrauchern ist auch die Möglichkeit einer stufenweisen Leistungsreduktion interessant. Diese kann insbesondere bei regelbaren Lasten sinnvoll sein, wie z.B. Sitzheizung oder Lüfter. Führt hier bereits eine teilweise Reduktion der Leistungsaufnahme, zu einer ausreichenden Stabilisierung der Spannung, ergibt sich ein signifikanter Komfortgewinn im Vergleich zur kompletten Abschaltung.

In das vorhandene Konzept kann ein solches Vorgehen durch zusätzliche Prioritätsstufen realisiert werden. Da eine Leistungsreduktion weniger Kosten in Form eines Komfortverlusts verursacht, kann die Priorität entsprechend geringer gewählt werden als für eine Totalabschaltung. Auch mehrere Drosselstufen mit entsprechender Prioritätsabstufung sind denkbar.

3.2.5 Resultierender Algorithmus

Die zuvor beschriebenen Aspekte werden zu einem iterativen Entscheidungsalgorithmus zusammengeführt. Der aktuelle Zustand des Bordnetzes wird kontinuierlich bewertet, indem die minimalen Spannungen der sicherheitsrelevanten Verbraucher mit den festgelegten Spannungsgrenzen verglichen werden. Liegt keine kritische Situation vor, verbleibt das System im Normalbetrieb.

Wird U_{lim} unterschritten, droht also eine kritische Unterspannung, werden alle abschaltbaren Komfortverbraucher als mögliche Kandidaten betrachtet. Für jeden Kandidaten wird ein Bewertungsmaß berechnet, das die erwartete Spannungsverbesserung mit den Kosten der Maßnahme ins Verhältnis setzt. Anschließend wird diejenige Maßnahme ausgewählt, die das beste Verhältnis aus Kosten und Nutzen aufweist.

Nach jeder Maßnahme wird der Systemzustand erneut bewertet, sodass mehrere Eingriffe hintereinander möglich sind, bis alle sicherheitsrelevanten Lastspannungen größer U_{lim} sind. Die Implementierung erfolgt als iterative Schleife mit begrenzter Anzahl von Durchläufen und einer Mindestanforderung an die erzielte Verbesserung, um unnötige Eingriffe zu vermeiden. Abbildung 19 zeigt ein Zustandsdiagramm des implementierten Algorithmus. Der gesamte MATLAB-Code der Auswertelogik befindet sich im Anhang A.1.

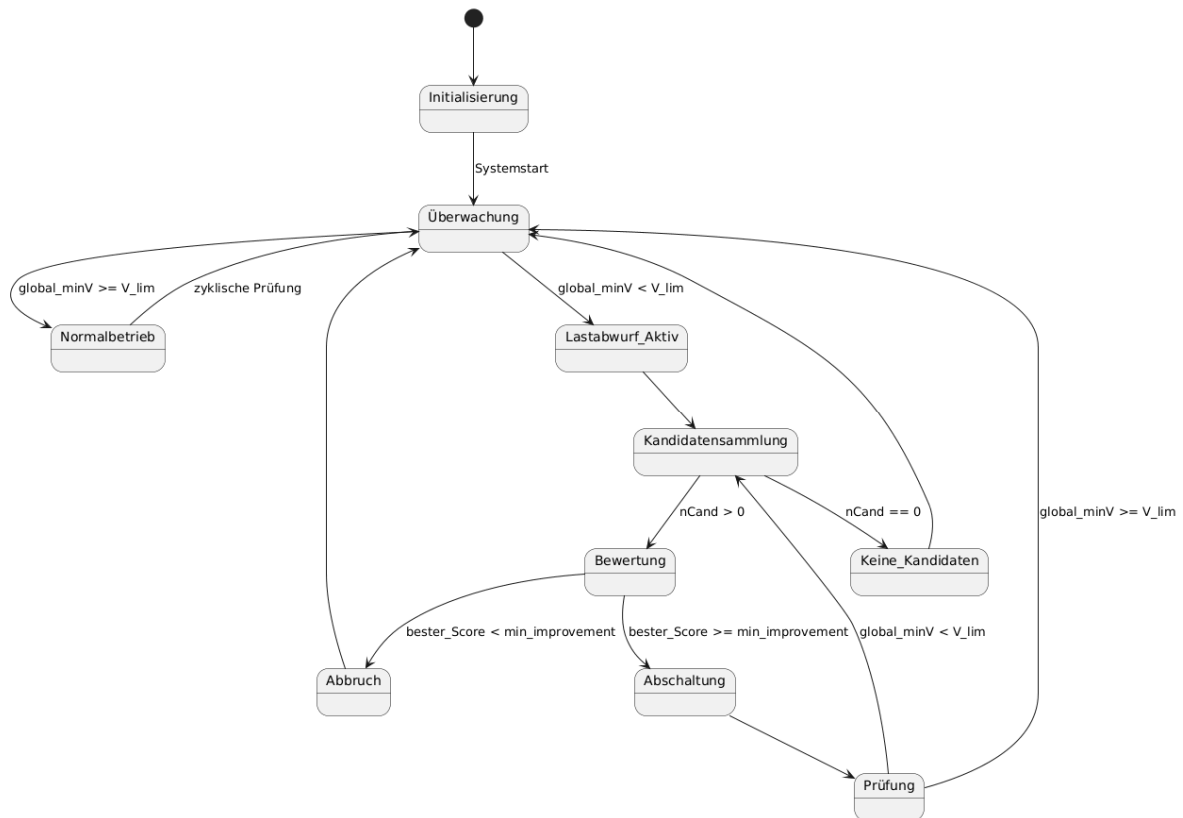


Abbildung 19: Zustandsdiagramm des resultierenden Lastmanagement-Algorithmus

3.3 Demonstration

Der entwickelte Algorithmus wird nun in einem realistischen, großen System angewendet. Das Bordnetz und der Lastmanagement-Algorithmus werden hierzu in MATLAB/Simscape implementiert und verschiedene kritische Lastszenarien untersucht.

3.3.1 Untersuchtes Beispielbordnetz

Das untersuchte Energiebordnetz besitzt einen zonalen Aufbau mit insgesamt drei PDUs, sowie eine Gesamtzahl von 200 unterschiedlichen Verbrauchern mit unterschiedlichem Strombedarf [5]. Eine schematische Darstellung des Bordnetzes ist in Abbildung 20 abgebildet. Die Lasten mit geraden Nummern werden als sicherheitsrelevant angenommen. Der Einfachheit halber wird für alle Lasten die Default-Priorität von 1 verwendet. Tabelle 1 zeigt die nominalen Stromaufnahmen der Verbraucher. Während diese Nominalströme für die Auslegung der Lastleitungen und eFuses zugrunde werden, können die Lasten dynamisches Verhalten mit variablen Lastströmen aufweisen.

Um die Zeitkonstanten und dynamischen Effekte des Bordnetzes adäquat abzubilden, werden außerdem die Eingangskondensatoren betrachtet, welche zur Spannungsstabilisierung der einzelnen Lasten üblich sind. Diese müssen kurzzeitige Spannungseinbrüche überbrücken können. Basierend auf Anforderungen aus dem LV-124-Prüfpuls E-10 wird sichergestellt, dass kurzzeitige Unterbrechungen von bis zu 100 μ s keine unzulässigen Auswirkungen auf die Verbraucher haben dürfen. Somit ergeben sich die in Tabelle 2 gezeigten Kapazitätswerte für die jeweiligen Verbraucher. Die Werte des seriellen Ersatzwiderstands (ESR) sind anhand üblicher Datenblattangaben gefittet.

Die folgenden Simulationen umfasst nun mehrere Szenarien, die jeweils unterschiedliche Lastprofile abbilden.

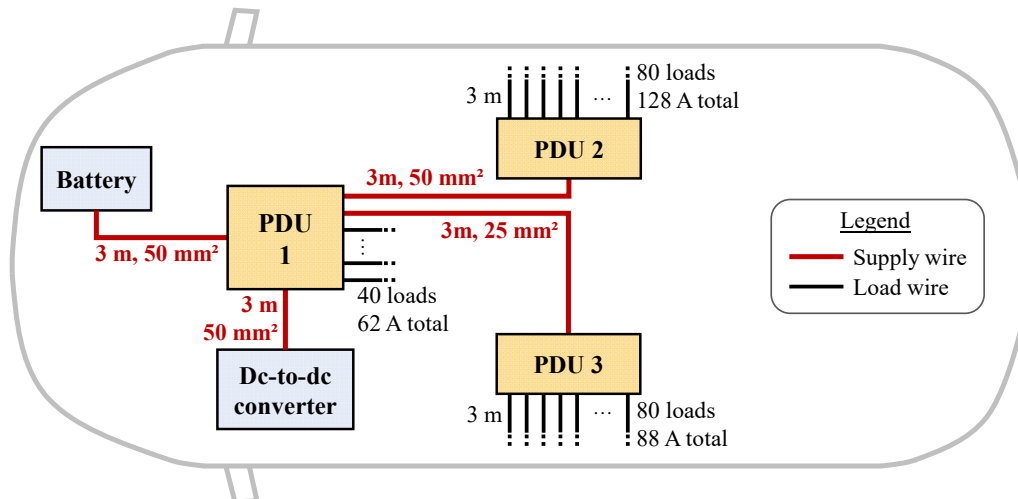


Abbildung 20: Topologie des untersuchten Energiebordnetzes

Tabelle 1: Nominalströme der Lasten des untersuchten Bordnetzes

	PDU 1	PDU 2	PDU 3
Verbraucherstrom	Anzahl	Anzahl	Anzahl
50 A		1	
40 A			1
20 A	1	1	1
10 A	2	2	1
5 A	3	4	1
1 A	4	12	6
0,1 A	30	60	70
Stromsumme in A	62	128	88

Tabelle 2: Kapazität und ESR der Stabilisierungskondensatoren der jeweiligen Lasten

Strom	Kapazität	
	C_{load}	R_{ESR}
50 A	1000 μ F	52 m Ω
40 A	680 μ F	59 m Ω
20 A	330 μ F	84 m Ω
10 A	150 μ F	141 m Ω
5 A	68 μ F	269 m Ω
1 A	47 μ F	269 m Ω
0.1 A	47 μ F	269 m Ω

Das beschriebene System wird wie erwähnt in MATLAB/Simscap implementiert. Ein Screenshot ist in Abbildung 21 gezeigt. Die Auswertung erfolgt codebasiert mit dem in Anhang A.1 dargestellten MATLAB-Code. Dieser wertet die aktuellen Strom- und Spannungswerte des Systems wie beschrieben aus und steuert anschließend die eFuses an.

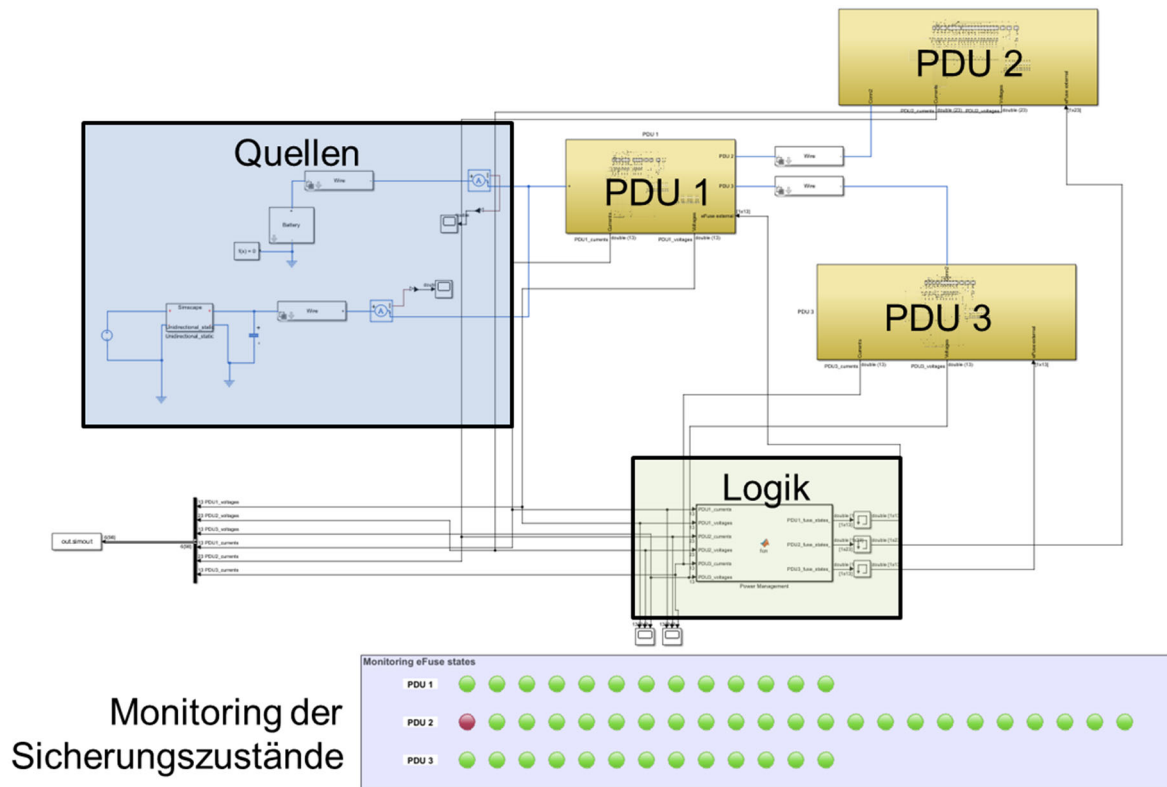


Abbildung 21: Screenshot der Implementierung des Beispielsystems inklusive Lastmanagement in MATLAB/Simulink

3.3.2 Ergebnisse

Das wie beschrieben implementierte Beispielsystem wird nun simuliert. Hierbei werden mehrere Szenarien mit unterschiedlicher Lastdynamik untersucht, in denen sich ohne Leistungsmanagement zu kritischen Versorgungssituationen kommt.

Szenario 1: Einzelner Eingriff zur Spannungsstabilisierung

Für das erste Lastszenario sind die eFuse-Ströme aller PDUs in Abbildung 22 dargestellt, aufgeteilt nach Komfortverbrauchern und sicherheitsrelevanten Verbrauchern. Abbildung 23 zeigt die dazugehörigen Lastspannungen. Mehrere Lasten erhöhen im betrachteten Szenario ihre Stromaufnahme, bspw. die Lasten 1, 41 und 42. Diese erhöhte Stromaufnahme im System führt zu Spannungseinbrüchen. Der sicherheitsrelevanten Last 42 droht daher unter die kritische Versorgungsspannungsgrenze von 9 V zu fallen. Das Lastmanagement detektiert dies jedoch und greift entsprechend ein. Durch eine gezielte Abschaltung des Komfortverbrauchers Last 41 wird die Unterspannung verhindert und das System so erfolgreich stabilisiert.

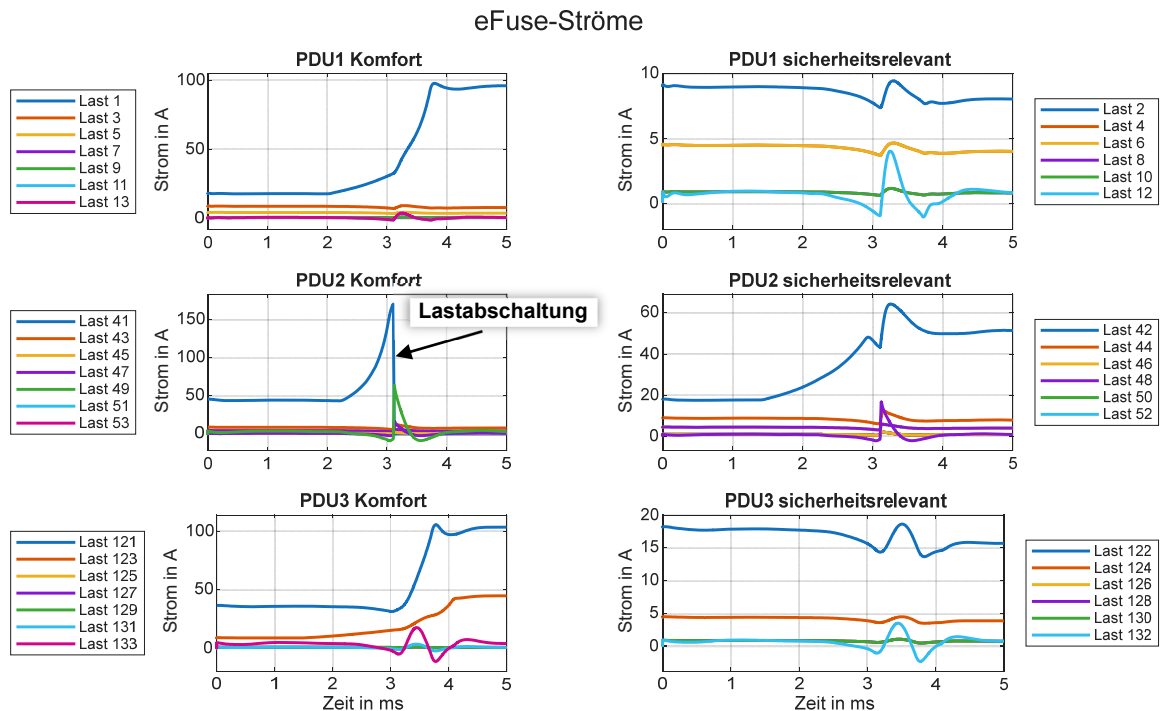


Abbildung 22: eFuse-Ströme der Komfortlasten (links) und der sicherheitskritischen Lasten (rechts) während Szenario 1

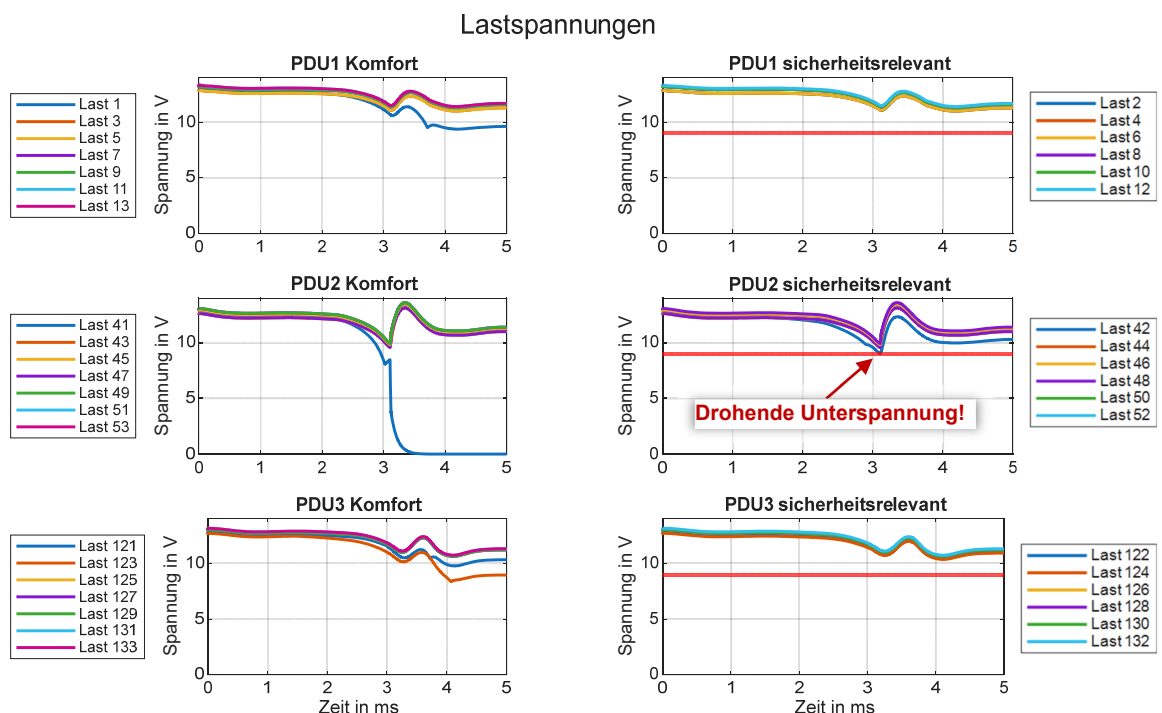


Abbildung 23: Lastspannungen der Komfortlasten (links) und der sicherheitskritischen Lasten (rechts) während Szenario 1

Szenario 2: Stufenweise Leistungsreduktion

Das gezeigte Szenario 1 konnte die Energieversorgung durch hartes Abschalten eines Komfortverbrauchers stabilisieren. Wie in Abschnitt 3.2.4 erläutert, kann je nach Verbraucher auch eine gezielte Leistungsreduktion eine alternative Maßnahme darstellen.

Abbildung 24 zeigt daher das gleiche Lastszenario, nun aber mit der Option, die Stromaufnahme der Last 41 auf 50 A zu begrenzen. Wie zu erkennen ist, verhindert auch diese Maßnahme die Unterspannung effektiv und ermöglicht dabei außerdem eine zumindest teilweise Verfügbarkeit der zuvor abgeschalteten Last.

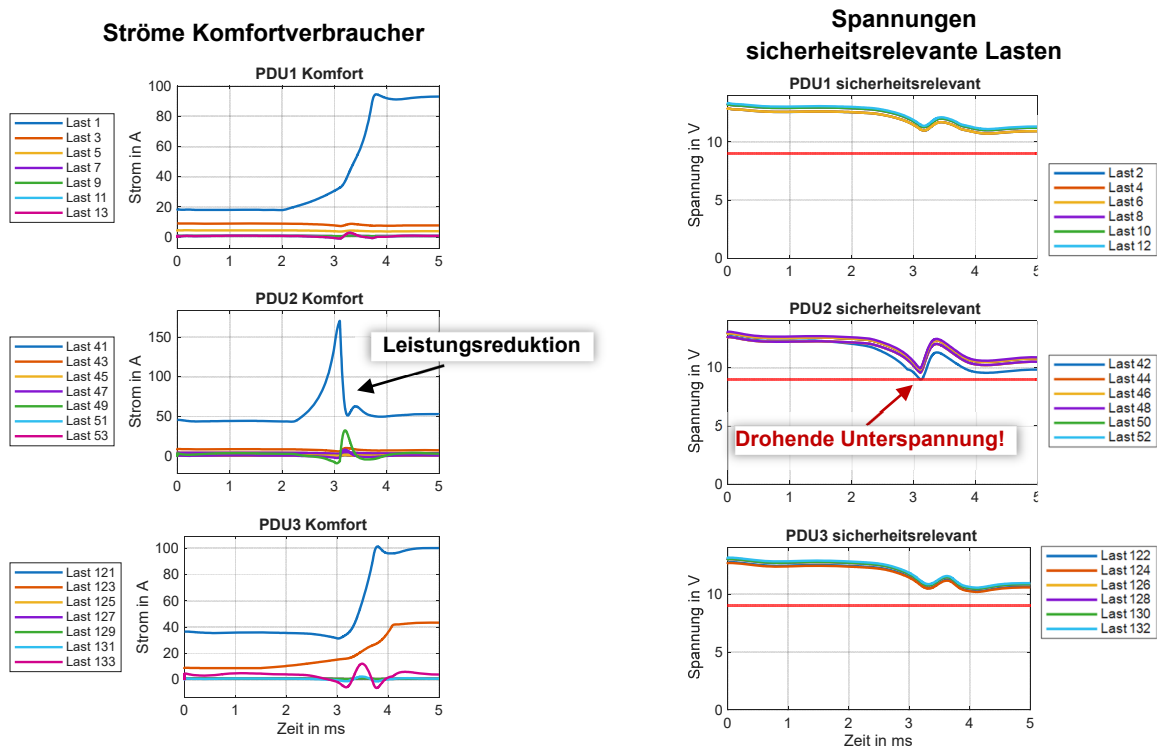


Abbildung 24: eFuse-Ströme der Komfortlasten (links) und der Lastspannungen der sicherheitskritischen Lasten (rechts) während Szenario 2

Szenario 3: Mehrfache Eingriffe zur Spannungsstabilisierung

Ausgehend von den vorherigen Szenarien soll nun ein Szenario untersucht werden, in dem mehrere Aktionen notwendig sind, um alle sicherheitskritischen Verbraucher zu stabilisieren. Es wird nun daher eine dementsprechend veränderte Lastdynamik angenommen. Wie die Ergebnisse in Abbildung 25 zeigen, führt stabilisiert die Abschaltung von Last 41 bei etwa 3,1 ms zunächst weiterhin die sicherheitskritische Last 42. Bei etwa 4,3 ms droht hier nun jedoch erneut eine kritische Unterspannung, weshalb ein weiterer Eingriff notwendig ist. In diesem Fall identifiziert der Lastmanagement-Algorithmus hier die Last 121 aus Kosten-Nutzen-Perspektive als optimale Maßnahme. Diese Komfortlast wird somit auch abgeschaltet; dies führt wie gewünscht zu einer Stabilisierung des sicherheitskritischen Verbrauchers.

Zusammengefasst zeigen die simulativen Untersuchungen damit die Eignung des entwickelten Leistungsmanagements zur effektiven und selektiven Stabilisierung der Energieversorgung in eFuse-basierten Bordnetzen.

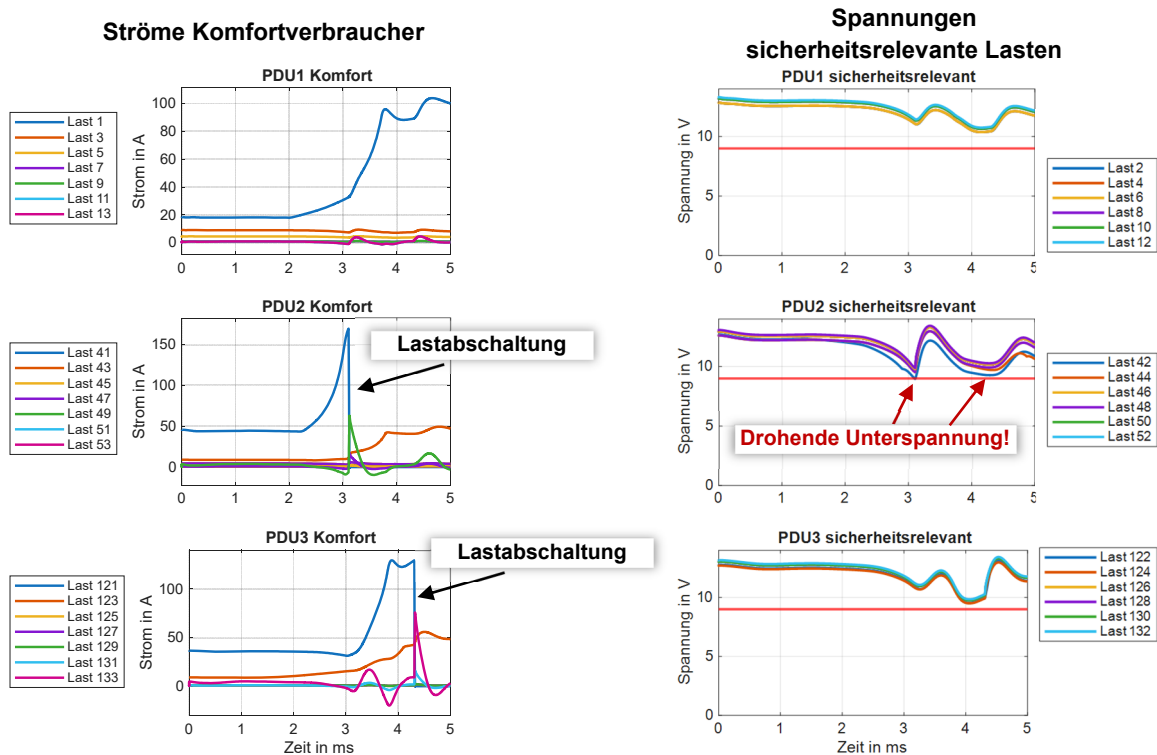


Abbildung 25: eFuse-Ströme der Komfortlasten (links) und der Lastspannungen der sicherheitskritischen Lasten (rechts) während Szenario 3

4 Direkter simulativer Vergleich von nicht-kommandiertem und kommandiertem eFuse-Verhalten in komplexen Bordnetzen

Wie in der Einleitung beschrieben, enthalten komplexe Fahrzeug-Bordnetze u.a. intelligente Energieverteiler und mehrere Sicherungsebenen. Das elektrische Bordnetz ist hierarchisch in verschiedene Ebenen unterteilt, wie in Abbildung 1 dargestellt.

Wir betrachten und vergleichen hier das „nicht-kommandierte“ Verhalten mit dem „kommandierten“ Verhalten einer eFuse. Beim „nicht-kommandierten“ Verhalten können die eFuses auf der untersten Ebene lokal einen Verbraucher im Falle eines Kurzschlusses oder einer Überlast sofort selbstständig abschalten. Dieses Verhalten entspricht im Wesentlichen dem Verhalten einer Schmelzsicherung. Die Entscheidung zum Abschalten erfolgt lokal aufgrund der erkannten Überlast am abgesicherten Verbraucher. Beim „kommandierten“ Verhalten hingegen wird von einer übergeordneten Sicherungsebene oder einem zentralen Controller die Abschaltung aktiv ausgelöst. Eine solche Betriebsweise ist mit Schmelzsicherungen nicht möglich, sondern wird erst durch die eFuse umsetzbar. So können z.B. sicherheitsrelevante Lasten bei einer Überlast weiter versorgt werden, indem nicht sicherheitsrelevante Lasten intelligent und kommandiert abgeschaltet werden.

Hierfür ist die Simulation des Verhaltens und der Steuerung des gesamten Bordnetzes essentiell. Auf die in den vorhergehenden Projekten der Arbeitsgruppe im AK30 entwickelte und ständig erweiterte Modellbibliothek in der Beschreibungssprache Modelica konnte hierbei sehr

gut aufgebaut werden. Als komplexere Bordnetzvariante sollte das Bordnetz aus Abbildung 1 in ein Gesamtmodell umgesetzt werden. Hierzu musste einige Modelle der Bibliothek erweitert und ein neues Controllermodell erstellt werden.

4.1 Erweiterung der Modelicabibliothek OnboardSystemFaultSimulation

4.1.1 Elektronische Sicherung – eFuse

Das Modell **eFuse** stellt eine vereinfachte elektronische Sicherung dar, die verschiedene Schutzmechanismen implementiert. Die Sicherung überwacht den Stromfluss und reagiert auf Überstrombedingungen. Sie verfügt über Funktionen wie Triggerung bei Überstrom, Verzögerung vor dem Auslösen und eine Rücksetzlogik. Die Sicherung enthält elektrische und logische Komponenten, um das Verhalten der Schutzschaltung zu simulieren. Als schaltende Komponente dient ein MOSFET.

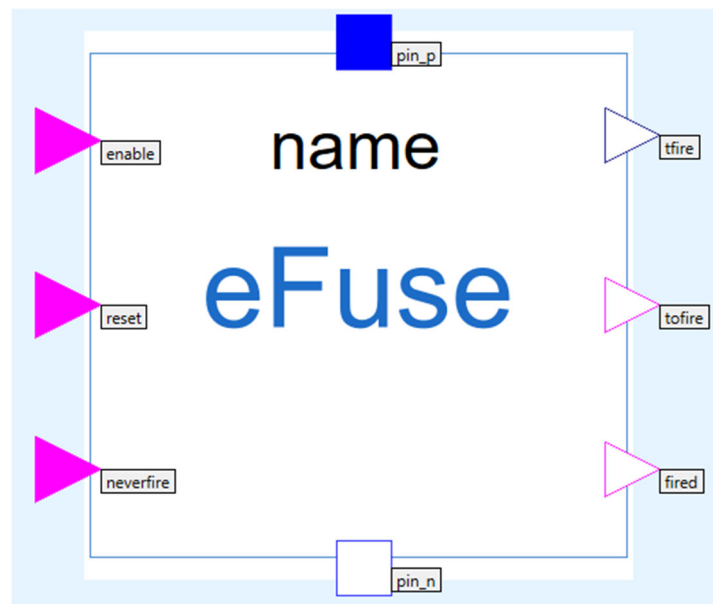


Abbildung 26: Symbol der erweiterten eFuse

Das Modell besitzt sowohl elektrische Anschlüsse (Leistungspfad) als auch boolesche und reellwertige Signal-Schnittstellen zur Ansteuerung und Statusabfrage. Erweitert wurde das Modell im aktuellen Projekt zum einen um das Signal „neverfire“ – die eFuse misst den Strom und meldet ihren Status an den übergeordneten Controller, löst aber selbst nie aus. Zum anderen gibt es nun den reellwertigen Ausgang „tfire“, welcher an den Controller den Auflösezeitpunkt (aktueller Zeitpunkt + Auslöseverzögerung) meldet. So kann der Controller innerhalb dieser Zeit mögliche Aktionen zur Auslöseunterbindung, wie z.B. das Abschalten anderer unkritischer Verbraucher, initiieren.

1. Elektrische Anschlüsse (Leistungspfad)

Name	Typ	Richtung	Beschreibung
pin_p	PositivePin	Elektrisch (bidirektional)	Positiver Leistungsanschluss. Eingang des zu schützenden Strompfads; hier liegt auch die Versorgung für Spannungssensor, Snubber-Netzwerk (R/C) und die Zener-Clamps an.
pin_n	NegativePin	Elektrisch (bidirektional)	Negativer Leistungsanschluss / Bezugspotential. Rückleiter des Strompfads, an den u. a. Source des MOSFETs, Spannungssensoren und die Zener-Clamps angeschlossen sind.

2. Eingänge

Name	Typ	Beschreibung
Enable	BooleanInput	Freigabesignal. Aktiviert die Auslöselogik (eFuse_triggering); ohne Freigabe erfolgt keine Stromfluss.
Reset	BooleanInput	Rücksetzsignal. Setzt einen ausgelösten Zustand der internen Triggerlogik zurück (z. B. nach einem Fehlerfall).
Neverfire	BooleanInput	Sperrsignal. Unterdrückt dauerhaft ein Auslösen der Sicherung, unabhängig vom erkannten Überstrom (z. B. für Testzwecke oder definierte Ausnahmefälle).

3. Ausgänge

Name	Typ	Beschreibung
Tofire	BooleanOutput	Zeigt an, dass die Auslösebedingung erkannt wurde und der Auslösevorgang unmittelbar bevorsteht bzw. eingeleitet wird.
Fired	BooleanOutput	Zeigt an, dass die Sicherung tatsächlich ausgelöst hat (Endzustand nach erfolgter Abschaltung).
Tfire	RealOutput	Ausgelöste bzw. berechnete Auslösezeit (Zeitpunkt/Dauer bis zum Fire-Ereignis), geliefert von der internen Triggerlogik.

4. Relevante Parameter der eFuse

Parameter	Wert	Bedeutung
t_OCP	16 μ s	Verzögerungszeit für den schnellen Überstromschutz (Overcurrent Protection).
I_OCP	10 A	Auslöseschwelle für den schnellen Überstromschutz.
I_trigger	1 A	Auslöseschwelle für die „normale“ Überlasterkennung.
trigger_delay	10 ms	Verzögerungszeit für die normale Überlasterkennung.

Parameter	Wert	Bedeutung
trigger_delay_tofire	1 ms	Verzögerungszeit der Meldung an den Controller für die normale Überlasterkennung.

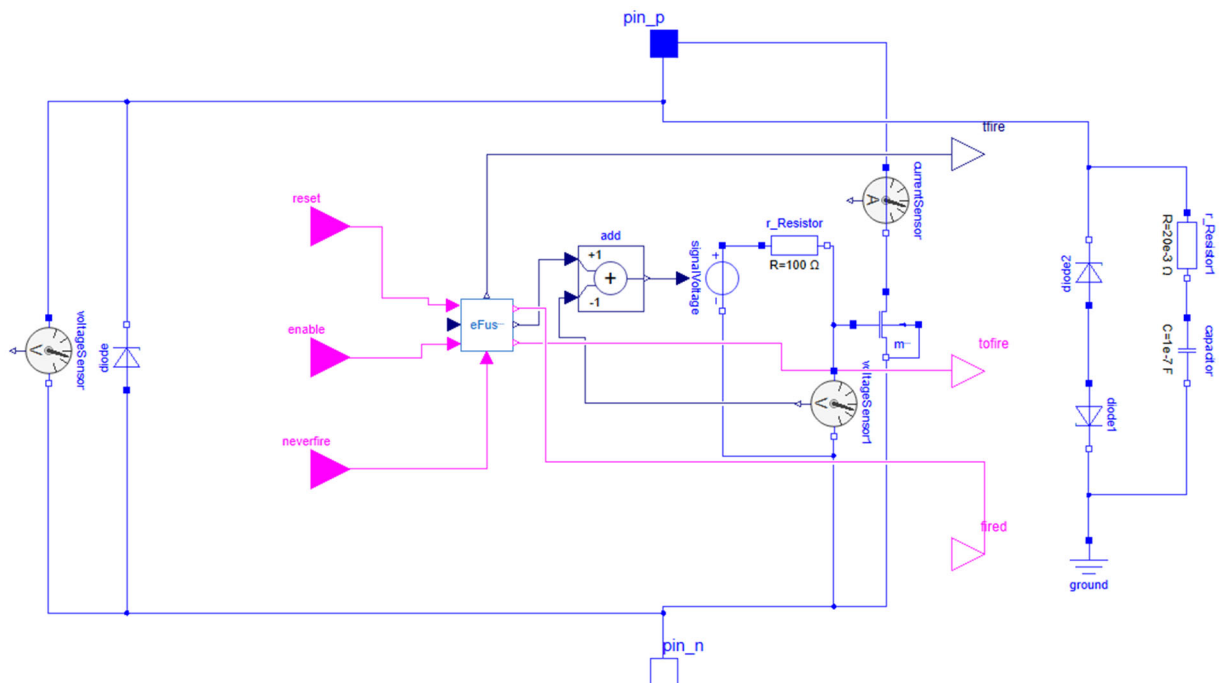


Abbildung 27: Komponenten des eFuse-Modells

4.1.2 Triggerlogik für die elektronische Sicherung – eFuse_triggering

Das Modell **eFuse_triggering** implementiert die Auslöselogik einer eFuse (angelehnt an den BTS50010). Es misst den aktuellen Sicherungsstrom und löst verschiedene Schutzmechanismen aus, basierend auf Überstrombedingungen (I_{OCP}) oder normalen Überlastbedingungen ($I_{trigger}$). Entsprechend der notwendigen Erweiterung am eFuse-Modell wurden auch hier der Eingang „neverfire“ und der Ausgang „tfire“ sowie die dafür notwendige Logik implementiert.

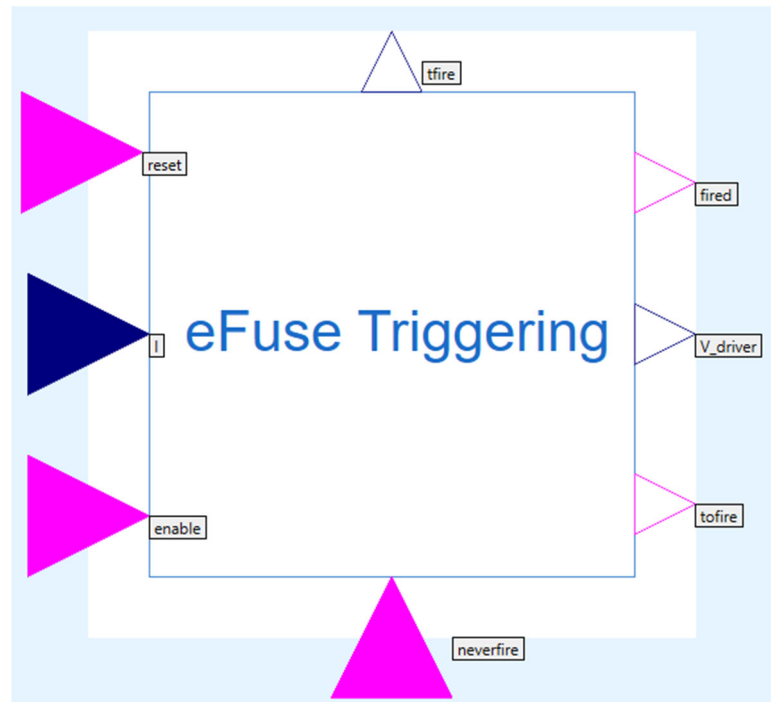


Abbildung 28: Symbol von eFuse_triggering

1. Eingänge

Signal	Typ	Einheit	Beschreibung
I	RealInput	A	Gemessener Strom durch die Sicherung. Grundlage für die gesamte Trigger-Logik (OCP- und Normal-Überlast-Erkennung).
enable	BooleanInput	–	Freigabesignal. Ist enable = false, werden alle Ausgänge zwangsweise deaktiviert ($V_{driver} = 0$, fired = false, tofire = false, tfire = -1), unabhängig vom Stromverlauf.
reset	BooleanInput	–	Setzt bei steigender Flanke (edge(reset)) alle internen Zustände (state_OCP, state_normal, state_tofire, triggered, triggered_tofire) auf den Ausgangszustand zurück.
neverfire	BooleanInput	–	Blockiersignal. Ist neverfire = true, bleibt die Sicherung dauerhaft "nicht ausgelöst": V_{driver} bleibt auf 10 V, fired/tofire bleiben false, unabhängig vom Strom.

2. Ausgänge

Signal	Typ	Einheit	Beschreibung
V_driver	RealOutput	V	Gate-Treiberspannung. 10 V im Normalbetrieb, 0 V sobald triggered = true (Sicherung ausgelöst) oder enable = false.
fired	BooleanOutput	–	Latched-Status: true, sobald die Sicherung endgültig ausgelöst hat (über OCP- oder Normal-Überlastpfad). Bleibt gesetzt bis zum nächsten reset.

Signal	Typ	Einheit	Beschreibung
tofire	BooleanOutput	–	Vorwarnsignal: true, sobald der Normal-Überlastpfad die kurze Vorlaufzeit (trigger_delay_tofire) überschritten hat – zeigt an, dass die Sicherung "gleich" auslösen wird. Hinweis: Wird derzeit nur vom Normal-Überlastpfad gesetzt, nicht vom OCP-Pfad.
tfire	RealOutput	s	Zeitstempel des voraussichtlichen/tatsächlichen Auslösezeitpunkts (dtfire), sobald tofire = true. Andernfalls -1.

3. Parameter (direkt von eFuse durchgereicht)

Parameter	Wert	Beschreibung
I_OCP	10 A	Verzögerungszeit für den schnellen Überstromschutz (Overcurrent Protection).
t_OCP	16 μ s	Auslöseschwelle für den schnellen Überstromschutz.
I_trigger	1 A	Auslöseschwelle für die „normale“ Überlasterkennung.
trigger_delay	10 ms	Verzögerungszeit für die normale Überlasterkennung.
trigger_delay_tofire	1 ms	Verzögerungszeit der Meldung an den Controller für die normale Überlasterkennung.

Interne Auslöselogik (zum Verständnis der Ausgänge)

1. Overcurrent Protection (OCP) – Kurzschlusschutz
 - Schwelle: I_OCP = 10 A
 - Verzögerung: t_OCP = 16 μ s
 - Bei Überschreitung länger als t_OCP $\rightarrow$ triggered = true $\rightarrow$ fired = true, V_driver = 0
2. Normal-Überlastpfad
 - Schwelle: I_trigger = 1 A
 - Vorwarnverzögerung: trigger_delay_tofire = 1 ms $\rightarrow$ setzt tofire = true, tfire = dtfire
 - Auslöseverzögerung: trigger_delay = 10 ms $\rightarrow$ setzt triggered = true $\rightarrow$ fired = true, V_driver = 0

Beide Pfade werden durch reset zurückgesetzt und durch enable = false bzw. neverfire = true überstimmt (siehe Eingangstabelle).

4.1.3 Power Distribution Unit – PDU

Das Modell **PDU** simuliert eine elektrische Verteilungseinheit mit elektronischen Sicherungen (eFuses) und Unterstützung für vier gesteuerte Lasten. Die Sicherungen überwachen den Stromfluss, können durch Boolesche Signale aktiviert/deaktiviert werden und bieten Schutz vor Überstrom. Das Modell unterstützt Reset- und Fired-Signale, um den Status der Sicherungen in Echtzeit zu kontrollieren.

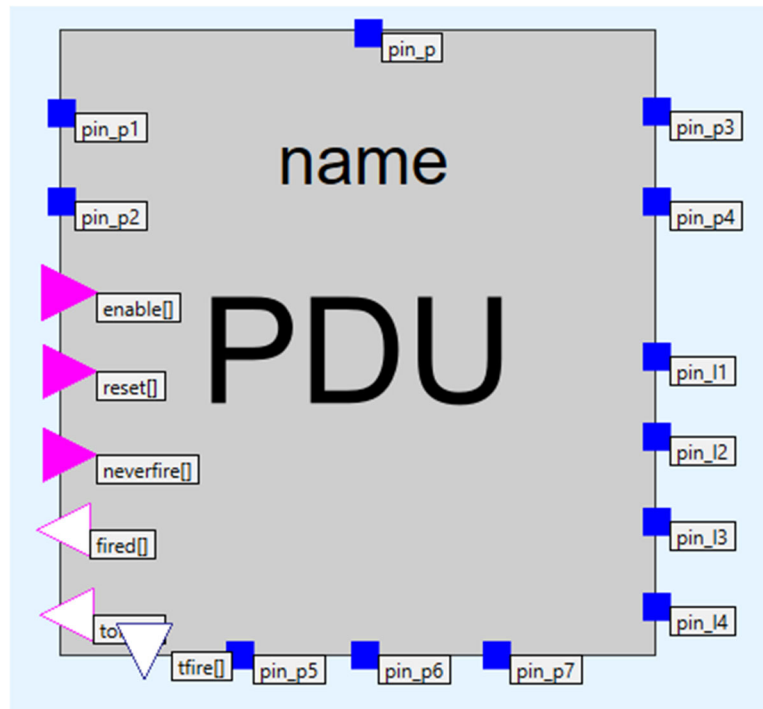


Abbildung 29: Symbol einer PDU

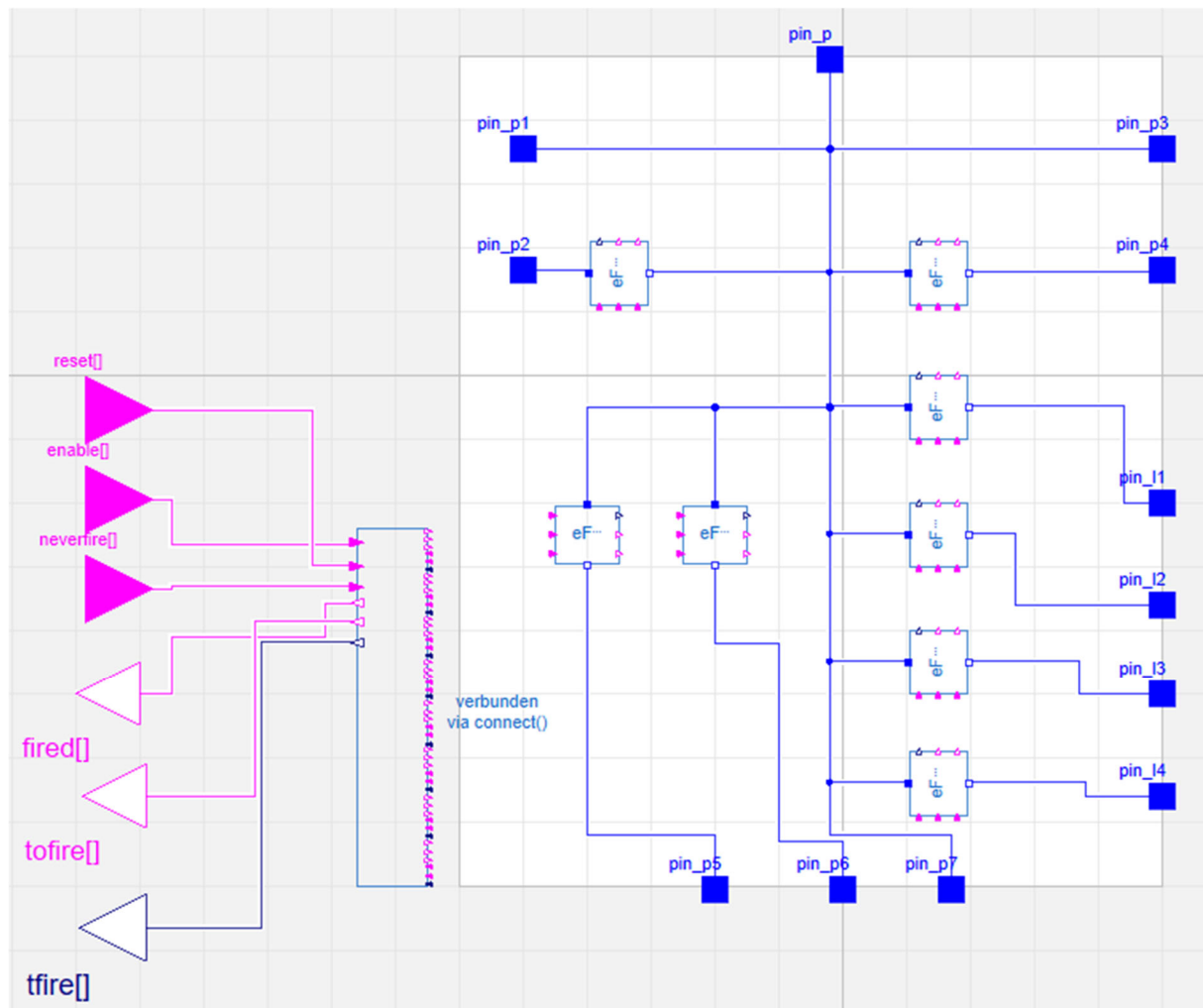


Abbildung 30: Komponenten der PDU

Die PDU verteilt eine gemeinsame Eingangsspannung (Sammelschiene) auf acht unabhängig überwachte und abgesicherte Anschlüsse: vier Versorgungsein- und -gänge (p2, p4, p5, p6) und vier Lastausgänge (I1 bis I4). Jeder Anschluss besitzt eine eigene elektronische Sicherung (eFuse), die den fließenden Strom permanent überwacht und den Anschluss bei Überlast oder Kurzschluss selbstständig abschaltet. Die Pins pin_p1, pin_p3, pin_p und pin_p7 sind intern direkt miteinander verbunden und bilden die Sammelschiene der PDU. Die PDU fasst die Ansteuer- und Statussignale aller acht eFuses zu gemeinsamen Vektor-Schnittstellen zusammen, sodass eine übergeordnete Steuerung alle Ausgänge einheitlich überwachen und bedienen kann.

Erweitert wurde das Modell der PDU von zwei auf vier Lastausgänge sowie um die Verbindung sämtlicher eFuse-Steuer- und Signalleitungen über die Vektorschnittstellen nach außen für den Anschluss an eine zentrale Steuereinheit.

1. Elektrischer Aufbau

Alle acht eFuses hängen elektrisch an derselben Sammelschiene: Die Pins pin_p1, pin_p3, pin_p und pin_p7 sind intern direkt miteinander verbunden und bilden den gemeinsamen, un-abgesicherten Versorgungseingang der PDU. Von dieser Sammelschiene zweigen acht parallele eFuse-Zweige ab, jeder mit eigenem Ausgangspin:

- Versorgungsausgänge: pin_p2, pin_p4, pin_p5, pin_p6 – jeweils über eFuse_p2, eFuse_p4, eFuse_p5, eFuse_p6 abgesichert.

- Lastausgänge: pin_l1, pin_l2, pin_l3, pin_l4 – jeweils über eFuse_l1, eFuse_l2, eFuse_l3, eFuse_l4 abgesichert.

Ein Fehler oder eine Überlast an einem Ausgang wirkt sich also ausschließlich auf dessen eigene eFuse aus – die übrigen sieben Ausgänge bleiben unbeeinflusst in Betrieb. Die Sammelschiene selbst besitzt keine eigene Absicherung, da sie nur Verteilerfunktion hat.

2. Index-Zuordnung der Vektorsignale

Alle Vektor-Ein-/Ausgänge (enable, reset, neverfire, fired, tofire, tfire) besitzen die Länge $n = 8$. Der Index jedes Elements ist über folgende Konstanten fest zugeordnet:

Index	Konstante	zugehörige eFuse	Ausgang
1	p2	eFuse_p2	pin_p2
2	p4	eFuse_p4	pin_p4
3	p5	eFuse_p5	pin_p5
4	p6	eFuse_p6	pin_p6
5	l1	eFuse_l1	pin_l1
6	l2	eFuse_l2	pin_l2
7	l3	eFuse_l3	pin_l3
8	l4	eFuse_l4	pin_l4

3. Elektrische Anschlüsse (Pins)

Pin	Typ	Beschreibung
pin_p1, pin_p3, pin_p, pin_p7	PositivePin	Gemeinsame Versorgungssammelschiene. Alle vier Pins sind intern direkt miteinander verbunden (unabgesichert) und bilden den Eingang für alle 8 eFuses.
pin_p2	PositivePin	Abgesicherter Versorgungsausgang, geschaltet über eFuse_p2.
pin_p4	PositivePin	Abgesicherter Versorgungsausgang, geschaltet über eFuse_p4.
pin_p5	PositivePin	Abgesicherter Versorgungsausgang, geschaltet über eFuse_p5.
pin_p6	PositivePin	Abgesicherter Versorgungsausgang, geschaltet über eFuse_p6.
pin_l1	PositivePin	Abgesicherter Lastausgang, geschaltet über eFuse_l1.
pin_l2	PositivePin	Abgesicherter Lastausgang, geschaltet über eFuse_l2.

Pin	Typ	Beschreibung
pin_l3	PositivePin	Abgesicherter Lastausgang, geschaltet über eFuse_l3.
pin_l4	PositivePin	Abgesicherter Lastausgang, geschaltet über eFuse_l4.

4. Steuer- und Statussignale (Vektoren, n = 8)

Signal	Typ	Beschreibung
enable[n]	BooleanInput	Freigabesignal je eFuse. enable[i] = false deaktiviert den zugehörigen Ausgang unabhängig vom Stromverlauf.
reset[n]	BooleanInput	Rücksetzsignal je eFuse. Setzt bei steigender Flanke die internen Trigger-Zustände der zugehörigen eFuse zurück.
neverfire[n]	BooleanInput	Blockiersignal je eFuse. neverfire[i] = true verhindert das Auslösen der zugehörigen eFuse dauerhaft.
fired[n]	BooleanOutput	Latched-Auslösestatus je eFuse. true, sobald die zugehörige eFuse endgültig ausgelöst hat.
tofire[n]	BooleanOutput	Vorwarnsignal je eFuse. true, sobald die zugehörige eFuse kurz vor dem Auslösen steht.
tfire[n]	RealOutput	Zeitstempel (s) des voraussichtlichen/tatsächlichen Auslösezeitpunkts je eFuse, sonst -1.

Die Verschaltung der Vektoren auf die einzelnen eFuse-Instanzen erfolgt intern über die Hilfskomponente eFuseSignals (Examples.eFuseSignals_reset_neverfire_time), die je eFuse ein skalares Signalpaar (z. B. ep2/fp2/tofp2/nfp2/tfp2/rp2 für eFuse_p2) auf den entsprechenden Vektorindex abbildet.

5. Parametrierung der einzelnen eFuses

Alle 8 eFuse-Instanzen verwenden trigger_delay = 10 ms und trigger_delay_tofire = 1 ms. Die Strom-Schwellen unterscheiden sich nach Ausgangstyp:

eFuse	I_OCP	I_trigger	trigger_delay	trigger_delay_tofire
eFuse_p2	200 A	100 A	10 ms	1 ms
eFuse_p4	200 A	100 A	10 ms	1 ms
eFuse_p5	200 A	100 A	10 ms	1 ms
eFuse_p6	200 A	100 A	10 ms	1 ms
eFuse_l1	200 A	10 A	10 ms	1 ms
eFuse_l2	200 A	10 A	10 ms	1 ms

eFuse	I_OCP	I_trigger	trigger_delay	trigger_delay_tofire
eFuse_I3	200 A	10 A	10 ms	1 ms
eFuse_I4	200 A	10 A	10 ms	1 ms

4.1.4 Übergeordnete Sicherungsebene – Controller

Das Modelica-Modell **Controller** bildet die Steuerungslogik eines Stromverteilungssystems mit einem Hauptpfad ("main"), einem Backbone-Pfad ("backbone") und vier nachgeschalteten Power Distribution Units (PDU1–PDU4) ab. Jede PDU verfügt über acht schaltbare Ausgänge: vier Ports (P2, P4, P5, P6) und vier Lasten (Load1–Load4).

Der Controller setzt für jeden Ausgang Enable- (e), Reset- (r) und Neverfire-Signale (nf) und wertet im Gegenzug Rückmeldungen der eFuses aus: "tofire"-Signale (bevorstehendes Auslösen), "fired"-Signale (tatsächliches Auslösen sowie tfire-Signale (Auslösezeitpunkt). Alle Ereignisse werden über Konsolenausgaben protokolliert.

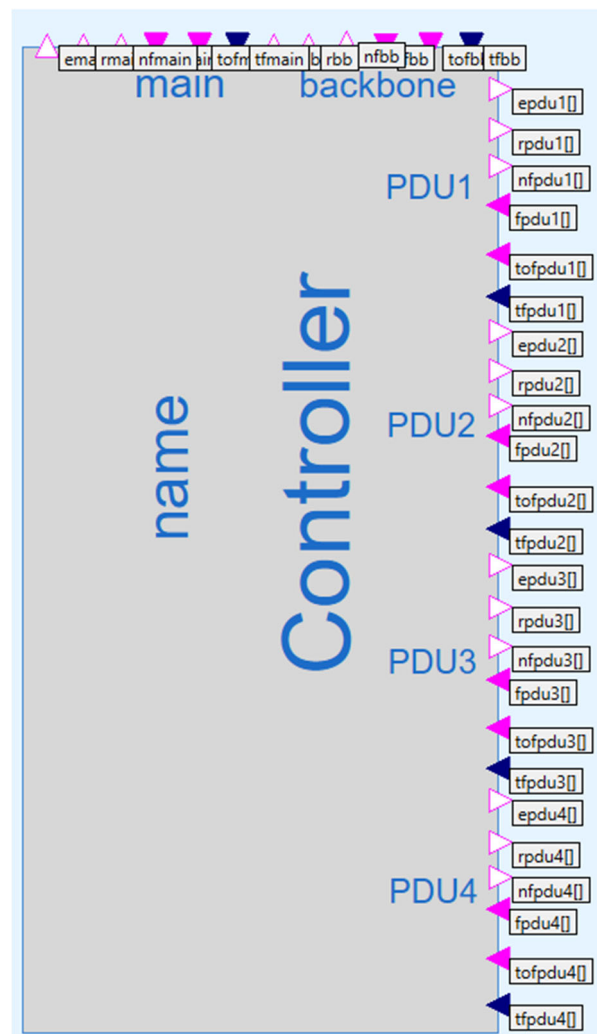


Abbildung 31: Symbol des Controllers - übergeordnete Steuereinheit

1. Konstanten und Indizierung

Die acht Ausgänge jeder PDU (die acht internen eFuses der PDU) werden über benannte Konstanten indiziert (n = 8 Ausgänge je PDU):

Konstante	Index	Bedeutung
p2	1	Port 2
p4	2	Port 4
p5	3	Port 5
p6	4	Port 6
l1	5	Load 1
l2	6	Load 2
l3	7	Load 3
l4	8	Load 4

2. Ausgänge des Controllers (an die eFuses)

Signal	Typ	Bedeutung
emain, ebb	BooleanOutput	Freigabe (Enable) Main / Backbone
rmain, rbb	BooleanOutput	Reset Main / Backbone
nfmain, nfbb	BooleanOutput	Neverfire -Signal Main / Backbone
epdu1...epdu4[n]	BooleanOutput[8]	Freigabe (Enable) je Ausgang der jeweiligen PDU
rpdu1...rpdu4[n]	BooleanOutput[8]	Reset je Ausgang der jeweiligen PDU
nfpdu1...nfpdu4[n]	BooleanOutput[8]	Neverfire -Signal je Ausgang der jeweiligen PDU

3. Eingänge des Controllers (von den eFuses)

Signal	Typ	Bedeutung
fmain, fbb	BooleanInput	"Fired" – eFuse hat ausgelöst (Main / Backbone)
tofmain, tofbb	BooleanInput	"Tofire" – eFuse steht kurz vor Auslösung (Main / Backbone)
tfmain, tfbb	RealInput	Zeitpunkt des (bevorstehenden) Auslösens (Main / Backbone)
fpdu1...fpdu4[n]	BooleanInput[8]	"Fired" je Ausgang der jeweiligen PDU
tofpdu1...tofpdu4[n]	BooleanInput[8]	"Tofire" je Ausgang der jeweiligen PDU
tfpdu1...tfpdu4[n]	RealInput[8]	Auslösezeitpunkt je Ausgang der jeweiligen PDU

4. Ereignis- und Ablauflogik (algorithm-Abschnitt)

Im algorithm-Abschnitt werden Zustandsänderungen der Signale erkannt (when-Klauseln) und auf der Konsole ausgegeben. Dies sind u.a. Meldungen zu bevorstehenden (tofire) eFuse-Auslösungen mit dem entsprechenden Zeitpunkt (tfire), Meldungen zu tatsächlichen Auslösungen sowie zu kommandierten Zu- oder Abschaltungen von eFuses.

Ein kleines Beispiel:

```

when tofpdu2[p4] then
  myPrint(time,"eFuse [PDU2::P4] tofire",tfpdu2[p4]);
  epdu2_l2:=false;
  epdu2_l3:=false;
end when;

```

Dieses Codeschnipsel demonstriert, dass bei einem bevorstehenden Auslösen der eFuse am Anschluss P4 der PDU2 eine Meldung ausgegeben wird inklusive aktueller Zeitstempel sowie des geplanten Auslösezeitpunkts und im gleichen Zuge die nicht sicherheitsrelevanten Lasten L2 und L3 abgeschaltet werden (enable-Signale der entsprechenden eFuses auf false gesetzt). So wird versucht, den Versorgungsstrom am Anschluss P2 wieder genügend zu reduzieren, um mögliche sicherheitsrelevante Lasten an den Anschlüssen L1 und L4 weiter versorgen zu können.

4.2 Komplexere Bordnetzsimulationen

Als eine Variante eines komplexeren Bordnetzes wurde die Variante des Bordnetzes mit den verschiedenen Sicherungsebenen aus Abbildung 1 in Modelica umgesetzt. In Abbildung 32 sind die Hauptversorgungs-eFuse links oben, die eFuse für die Backbone links in der Mitte und der zentrale Controller links unten ersichtlich. Jeder der vier PDUs enthält acht eFuses für die jeweils 4 Lastausgänge sowie die weiteren schaltbaren Anschlüsse. Entsprechend Abbildung 1 versorgt jede PDU drei bis 4 Lasten, teilweise sicherheitsrelevant, teilweise nicht. Die dickeren schwarzen Linien sind die schaltbaren Versorgungsleitungen zu und zwischen den PDUs, die dickeren roten Linien stellen die sternförmig verschaltete Backbone dar. Die Elemente in den Leitungen sind entsprechende Leitungsmodelle. Zwischen Controller und den beiden externen eFuses sowie zu sämtlichen PDUs und deren jeweils acht eFuses verlaufen sehr viele Steuerleitungen, die hier der Übersichtlichkeit wegen ausgeblendet sind. Somit werden in Summe 32 eFuses in dieser Variante verwendet.

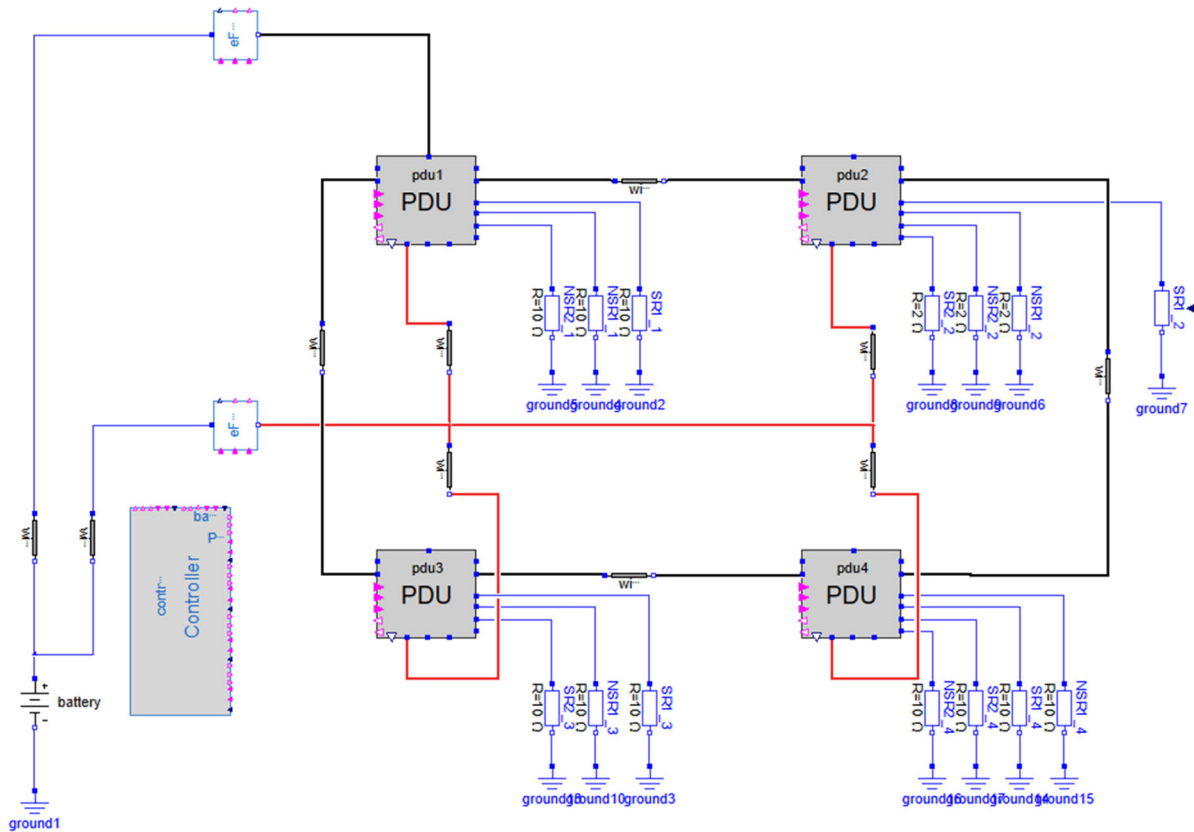


Abbildung 32: komplexeres Bordnetzmodell in Modelica

4.2.1 Szenarium für Simulationstest

Um das Zusammenspiel zwischen Controller und den eFuses sowie die Steuerbarkeit und Rückmeldungen der eFuses zu verifizieren, wurde folgendes Szenarium genutzt:

- die sicherheitsrelevante Last SR1 an PDU2 hat einen Fehler bei $t = 0,2 \text{ s}$
- die sicherheitsrelevante Last SR1 an PDU3 hat einen Fehler bei $t = 0,4 \text{ s}$
- die sicherheitsrelevante Last SR1 an PDU4 wird bei $t = 0,4 \text{ s}$ vom Controller deaktiviert

Bei sämtlichen nachfolgenden Untersuchungen sind die Zeit zwischen Fehler und Meldung an den Controller auf 1 ms eingestellt sowie die Zeit zwischen Fehler und Auslösen der eFuse auf 10 ms.

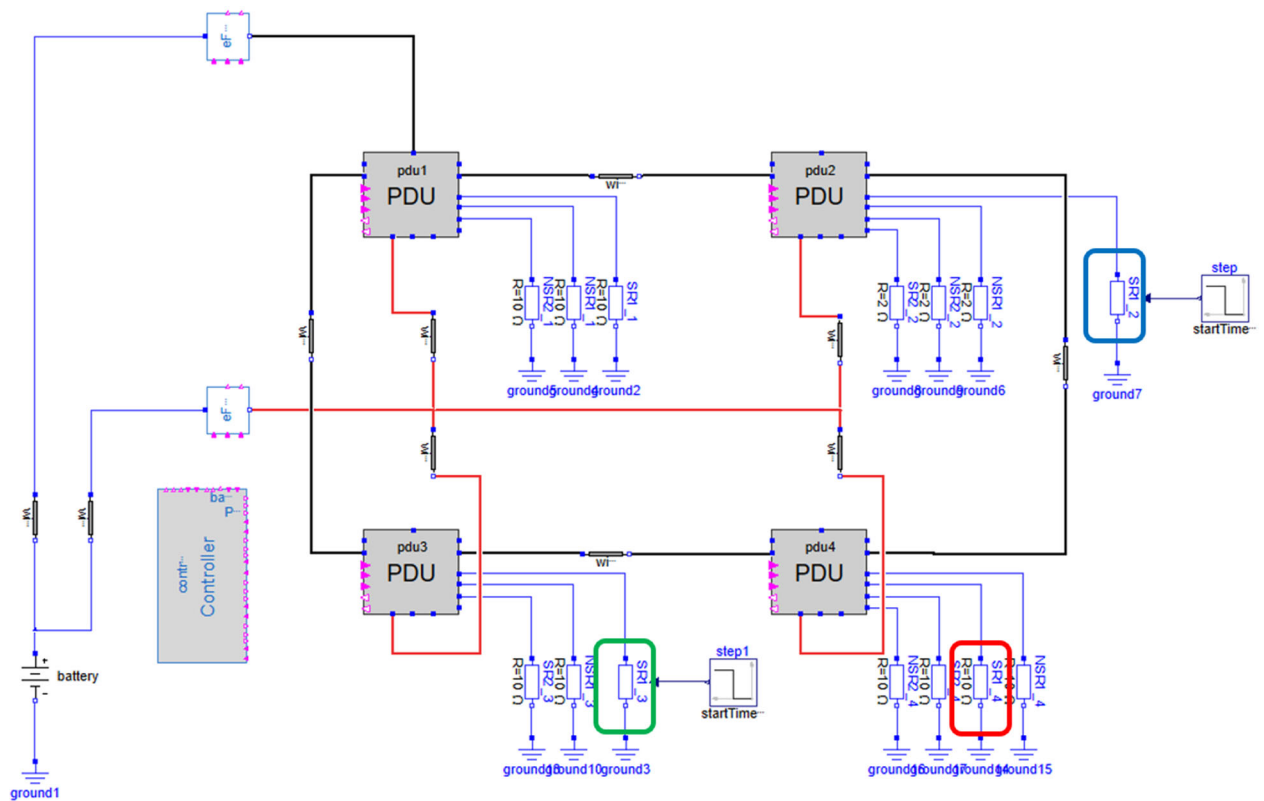


Abbildung 33: Testszenarium

Anbei die Ausgaben des Controllers auf der Konsole. 1 ms nach Fehler meldet die eFuse am Lastausgang 1 der PDU2, dass sie bei 0,21 s auslösen würde. Gleiches gilt für die Haupt-eFuse, da ihr Maximalstrom ebenso überschritten wird. Bei 0,21 s hat die eFuse in der PDU2 ausgelöst. Die Haupt-eFuse arbeitet weiter, ihr Maximalstrom ist wieder unterschritten. Die eFuse für die Last 1 an PDU3 meldet bei 0,301 s, dass sie bei 0,31 s auslösen würde. Zu diesem Zeitpunkt löst sie dann auch aus. Schließlich wird bei 0,5 s das enable-Signal für die eFuse der Last 2 an PDU4 auf false gesetzt und die Last damit abgeschaltet.

Controller-Ausgaben:

```

-----
====> 0.201 s: eFuse [PDU2::Load1] tofire at 0.21 s.
-----
====> 0.201 s: eFuse [Main] tofire at 0.21 s.
-----
====> 0.21 s: eFuse [PDU2::Load1] fired.
-----
====> 0.301 s: eFuse [PDU3::Load1] tofire at 0.31 s.
-----
====> 0.31 s: eFuse [PDU3::Load1] fired.
-----
====> 0.5 s: eFuse [PDU4::Load2] disabled.
    
```

In folgender Abbildung sind die Stromverläufe an den drei untersuchten Lasten dargestellt. Die Signalisierungen an den Controller und die Steuerung der eFuse vom Controller wurden zumindest ansatzweise nachgewiesen.

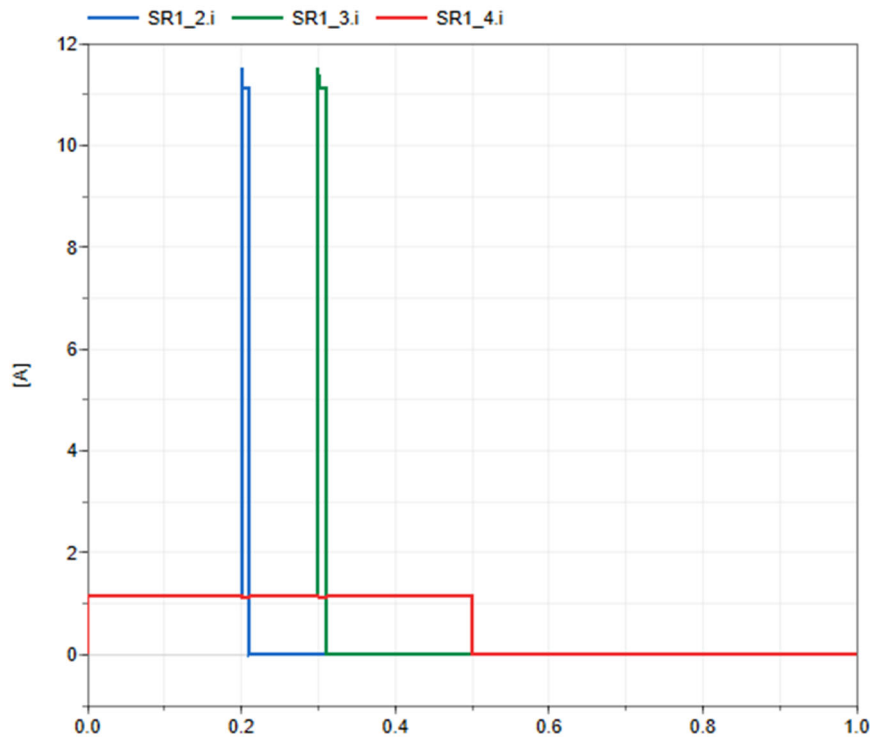


Abbildung 34: Stromverläufe beim Testszenarium (x-Achse: Zeit [s])

4.2.2 Ausfall der elektronischen Lenkung, Variante 1

Bei diesem Szenarium wird angenommen, dass die elektronische Lenkung (sicherheitsrelevante Last 1 an PDU2) zum Zeitpunkt 0,2 s einen stark erhöhten Strom benötigt, aber ihre eigene eFuse noch nicht überlastet ist. Dies könnte z.B. durch eine mechanische Blockade, Leckstrom o.a. bedingt sein.

Angenommen wird außerdem, dass in diese Fall keine Backbone verbaut ist. Aufgrund der weiteren drei Lasten an PDU2 würden allerdings die eFuses an den Versorgungsanschlüssen P2 bzw., P4 der PDU2 (siehe Abbildung) überlastet sein.

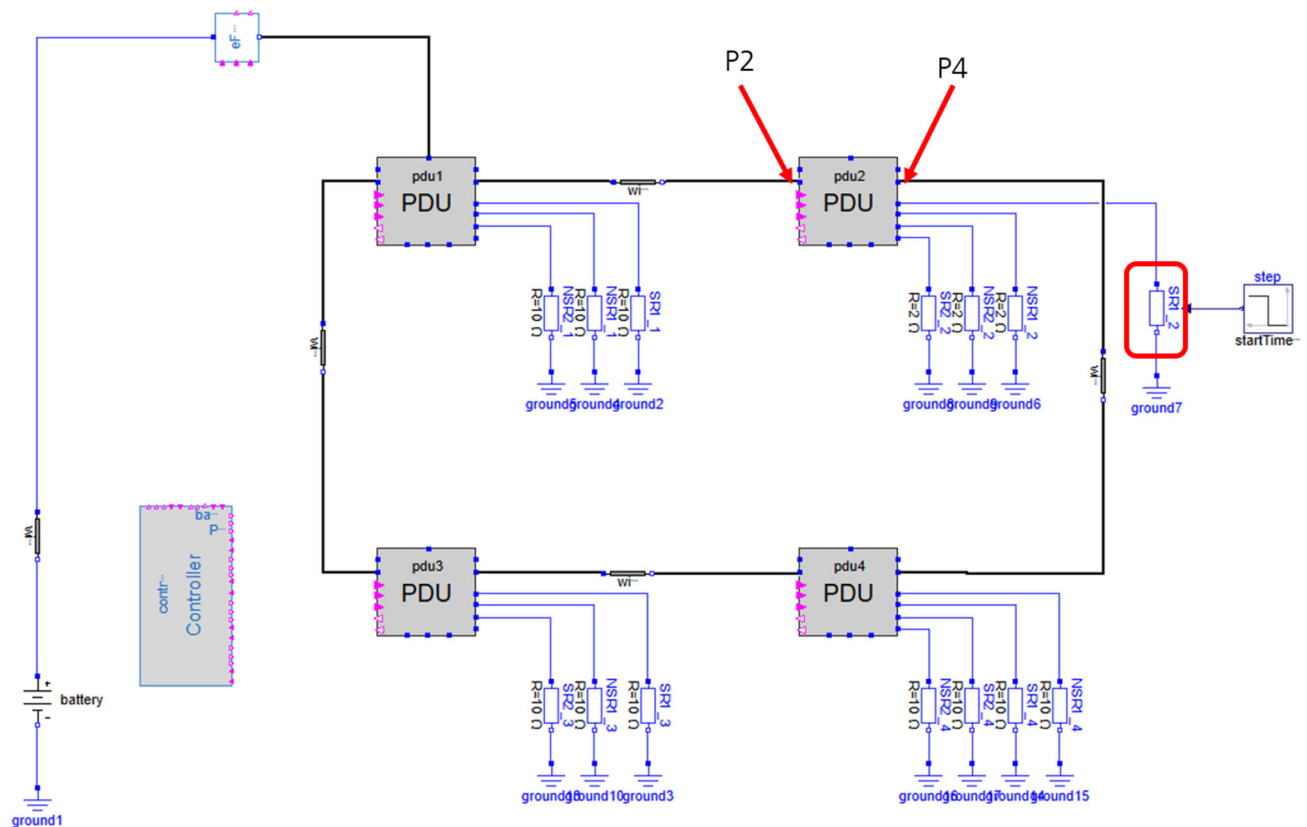


Abbildung 35: Ausfall der elektronischen Lenkung, Variante 1

1. Nicht-kommandiertes Verhalten

Der Controller ist auch bei den Untersuchungen zum nicht-kommandierten Verhalten vorhanden, um die Konsolenausgaben zu erhalten, er steuert allerdings die eFuses nicht selbst an.

Controller-Ausgaben:

====> 0.201 s: eFuse [PDU2::P2] tofire at 0.21 s.

====> 0.21 s: eFuse [PDU2::P2] fired.

====> 0.211 s: eFuse [PDU2::P4] tofire at 0.22 s.

====> 0.22 s: eFuse [PDU2::P4] fired.

Wie erwartet, meldet die eFuse am Anschluss P2 der PDU2 zum Zeitpunkt 0,201 s ihr bevorstehendes Auslösen und löst bei 0,21 s auch aus. Genau zu diesem Zeitpunkt trägt dann die eFuse am Anschluss P4 der PDU2 den Gesamtstrom, meldet ihr Auslösen dem Controller und löst dann folglich 10 ms später auch aus. **Die sicherheitsrelevante Last Lenkung ist stromlos.**

Die Stromkurven für sämtlich weiteren Fälle wurden in den Treffen zum Projekt ausführlich dargestellt und diskutiert, würden hier aber die Übersichtlichkeit stark reduzieren.

2. Kommandiertes Verhalten

Der Controller als zentrale Steuereinheit erkennt hierbei, dass der Strom an Versorgungsanschluss P2 zu groß ist und kann darauf geeignet reagieren, indem er die beiden nicht-sicherheitsrelevanten Lasten NRS1 und NRS2 an der PDU2 abschaltet, um möglichst den Gesamtstrom der PDU genügend zu reduzieren.

Controller-Ausgaben:

```
====> 0.201 s: eFuse [PDU2::P2] tofire at 0.21 s.
```

```
====> 0.201 s: eFuse [PDU2::Load2] disabled.
```

```
====> 0.201 s: eFuse [PDU2::Load3] disabled.
```

Wie die Konsolenausgaben zeigen, reicht das kommandierte Abschalten der beiden Lasten NRS1 und NRS2 aus, um **die Lenkung weiter mit Strom zu versorgen und keinen kritischen Zustand zu erreichen**.

4.2.3 Ausfall der elektronischen Lenkung, Variante 2

Bei dieser 2. Variante ist eine zentrale sternförmige Backbone vorhanden (grün markiert), aber vorerst nicht zugeschaltet. Analog zu Variante 1 benötigt die Lenkung an PDU2 bei 0,2 s einen erhöhten Strom, welcher in Summe mit dem andere die Lasten an dieser PDU nicht durch die Versorgungsanschlüsse PS2 bzw. P4 bereitgestellt werden können.

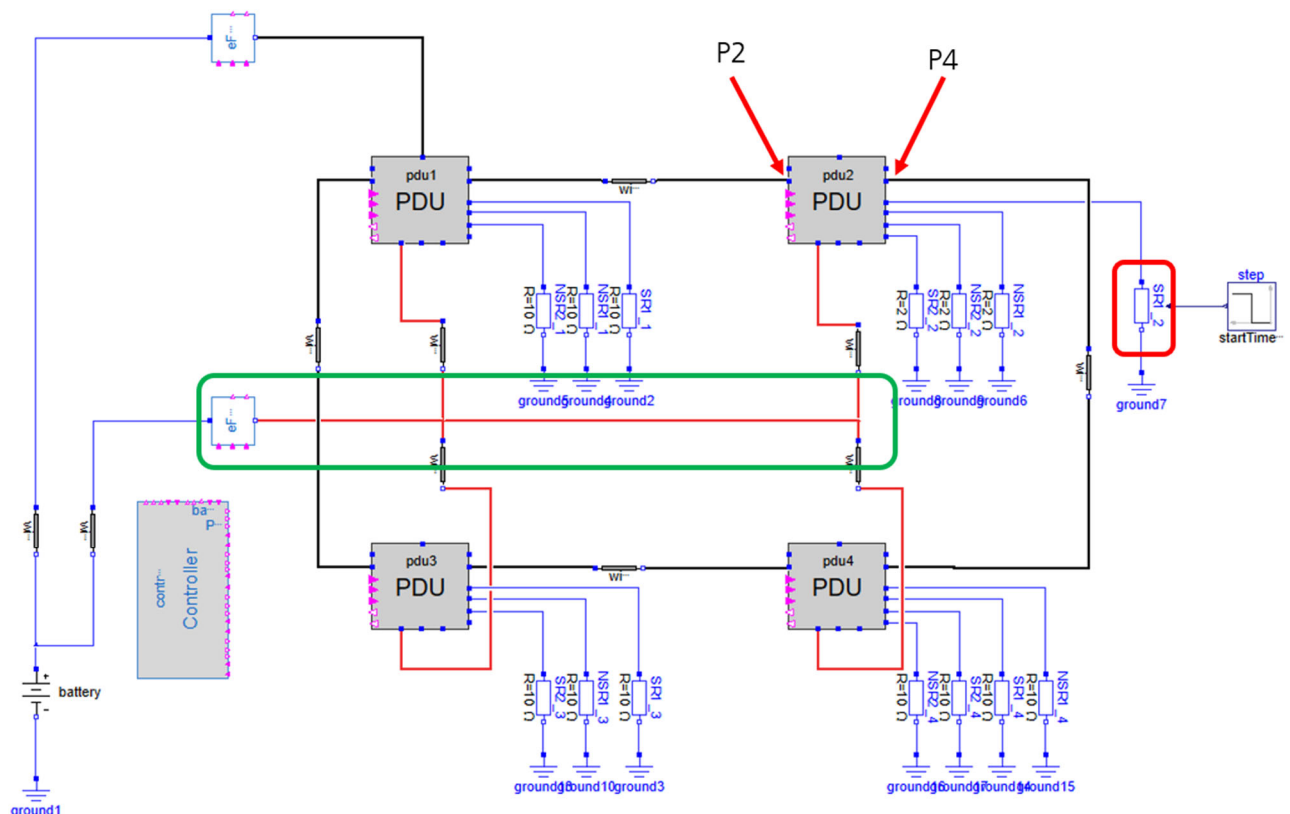


Abbildung 36: Ausfall der elektronischen Lenkung, Variante 2

1. Nicht-kommandiertes Verhalten

Controller-Ausgaben:

```
-----  
====> 0.201 s: eFuse [PDU2::P2] tofire at 0.21 s.  
-----  
====> 0.21 s: eFuse [PDU2::P2] fired.  
-----  
====> 0.211 s: eFuse [PDU2::P4] tofire at 0.22 s.  
-----  
====> 0.22 s: eFuse [PDU2::P4] fired.
```

Identisch zu Variante 1 lösen beim nicht-kommandierten Verhalten zuerst die eFuse am Anschluss P2 der PDU2 und danach wegen des hohen Gesamtstroms die eFuse am Anschluss P4 aus. **Die sicherheitsrelevante Last Lenkung ist stromlos.**

2. Kommandiertes Verhalten

Der Controller reagiert hierbei auf die Meldung der eFuse am Versorgungsanschluss P2, dass sie demnächst auslösen wird. Bei dieser Variante hat er die Möglichkeit, die Backbone zuzuschalten. Es bleiben sogar die beiden nicht-sicherheitsrelevanten Lasten NRS1 und NRS2 an der PDU2 in Betrieb, der Gesamtstrom durch die versorgenden eFuses reicht dafür aus.

Controller-Ausgaben:

```
-----  
====> 0.201 s: eFuse [PDU2::P2] tofire at 0.21 s.  
-----  
====> 0.201 s: eFuse [PDU2::BackBone] enabled.
```

Die Konsolenausgaben zeigen, dass durch das kommandierte Verhalten **die Lenkung weiter mit Strom wird und kein kritischer Zustand erreicht wird.**

4.2.4 Zuschalten von mehreren nicht-sicherheitsrelevanten Verbrauchern

Bei diesem Szenarium werden an der PDU2 die nichtsicherheitsrelevanten Lasten 1 und 2 bei 0,2 s bzw. bei 0,4 s zugeschaltet (türkis). Eine Backbone ist nicht vorhanden. Die Summe der Ströme durch die zwei sicherheitsrelevanten (rot) und die zwei zugeschalteten Lasten ist für die eFuses an den Versorgungsanschlüssen P2 bzw. P4 zu hoch und diese würden auslösen. Die Einzelströme der Lasten lösen deren eigene eFuses noch nicht aus.

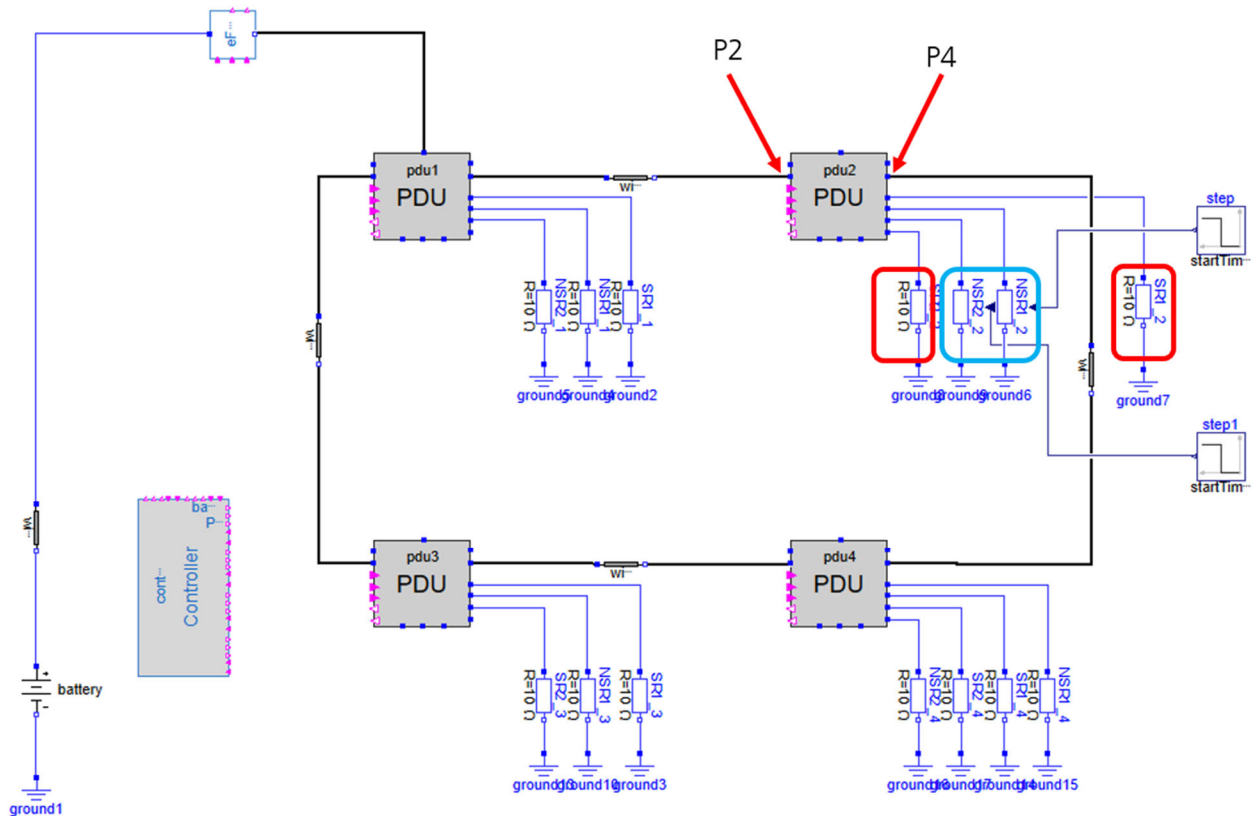


Abbildung 37: Zuschalten von mehreren nicht-sicherheitsrelevanten Verbrauchern

1. Nicht-kommandiertes Verhalten

Controller-Ausgaben:

```

=====
====> 0.2 s: eFuse [PDU2::L2] enabled.
=====
====> 0.4 s: eFuse [PDU2::L3] enabled.
=====
====> 0.401 s: eFuse [PDU2::P2] tofire at 0.41 s.
=====
====> 0.41 s: eFuse [PDU2::P2] fired.
=====
====> 0.411 s: eFuse [PDU2::P4] tofire at 0.42 s.
=====
====> 0.42 s: eFuse [PDU2::P4] fired.
=====
    
```

Die Konsolenausgabe bestätigt, dass nach dem Zuschalten der zweiten nichtsicherheitsrelevanten Last (Last 3 an PDU2) bei 0,4 s auch bei diesem Szenario zuerst die eFuse am Anschluss P2 der PDU2 und danach wegen des hohen Gesamtstroms die eFuse am Anschluss P4 auslösen. **Die sicherheitsrelevante Last Lenkung ist stromlos.**

2. Kommandiertes Verhalten

Nach dem Zuschalten der zweiten nichtsicherheitsrelevanten Last erhält der Controller die Meldung des bevorstehenden Auslösens der eFuse am Versorgungsanschluss P2 bei 0,41 s zum Zeitpunkt 0,401 s. Er reagiert, indem er die beiden nicht-sicherheitsrelevanten Lasten NRS1 und NRS2 an der PDU2 abschaltet, um möglichst den Gesamtstrom der PDU genügend zu reduzieren.

Controller-Ausgaben:

```
-----  
====> 0.2 s: eFuse [PDU2::L2] enabled.  
-----  
====> 0.4 s: eFuse [PDU2::L3] enabled.  
-----  
====> 0.401 s: eFuse [PDU2::P2] tofire at 0.41 s.  
-----  
====> 0.401 s: eFuse [PDU2::L2] disabled.  
-----  
====> 0.401 s: eFuse [PDU2::L3] disabled.
```

Die Konsolenausgaben zeigen, dass durch das kommandierte Abschalten der beiden nichtsicherheitsrelevanten Lasten **die Lenkung weiter mit Strom wird und kein kritischer Zustand erreicht wird.**

4.2.5 Serielle Versorgung von PDUs

Bei diesem Szenarium wird die PDU2 nur von PDU1 schaltbar versorgt und die PDU4 wiederum schaltbar von PDU2. Sämtliche sicherheitsrelevante Lasten sind bereits in Betrieb. Die nichtsicherheitsrelevanten Lasten 1 und 2 der PDU2 werden bei 0,1 s bzw. 0,2 s zugeschaltet. Dies führt noch zu keiner Überlast an der schaltbaren Verbindung P2 zur versorgenden PDU1. Schließlich werden noch die beiden nichtsicherheitsrelevanten Lasten 1 und 2 der PDU4 bei 0,3 s bzw. 0,4 s zugeschaltet. Die einzelnen eFuses der Lasten sowie die eFuse der schaltbaren Verbindung P4 von PDU2 zu PDU4 sind hierbei noch nicht überlastet, jedoch die eFuse bei P2 durch den Gesamtstrom für die beiden seriell angebundenen PDUs. Diese würde nun auslösen.

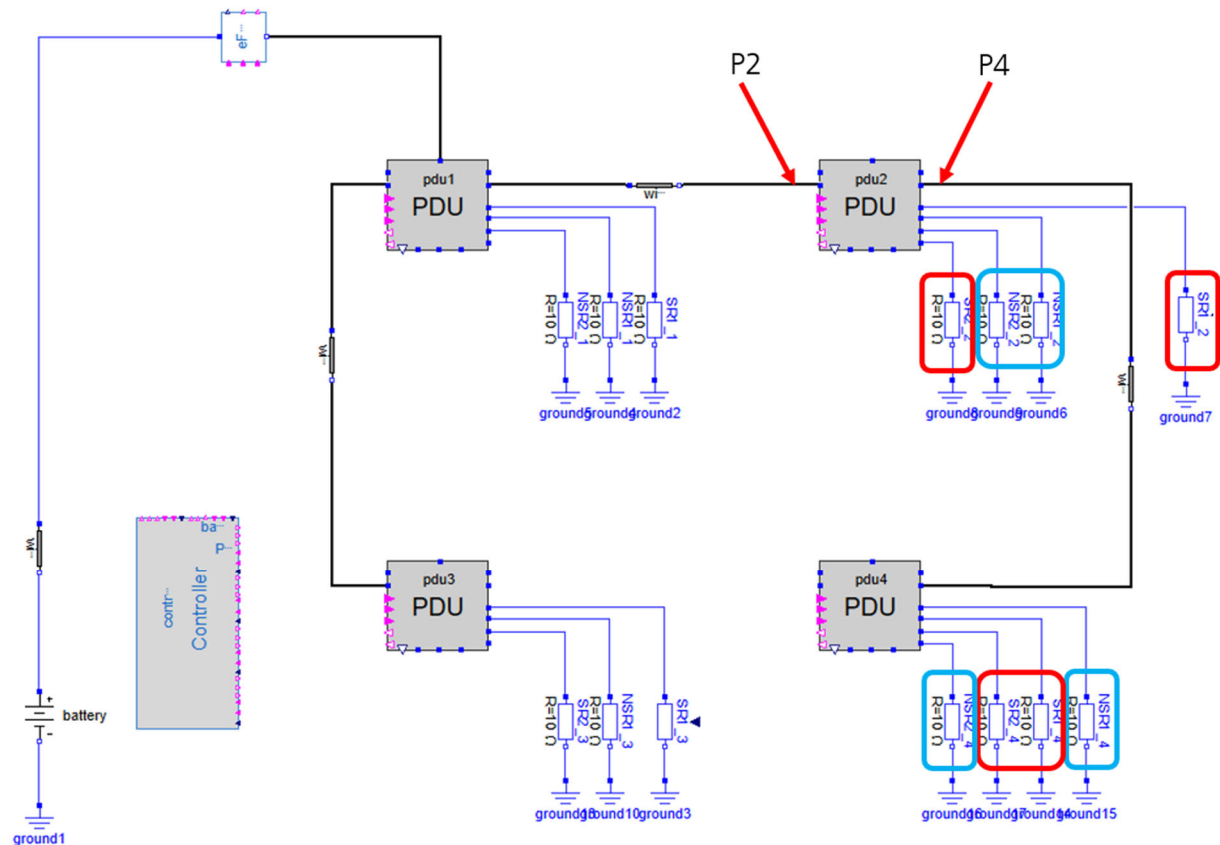


Abbildung 38: Serielle Versorgung zweier PDUs

1. Nicht-kommandiertes Verhalten

Controller-Ausgaben:

```

=====
====> 0.1 s: eFuse [PDU2::L2] enabled.
=====
====> 0.2 s: eFuse [PDU2::L3] enabled.
=====
====> 0.3 s: eFuse [PDU4::L1] enabled.
=====
====> 0.4 s: eFuse [PDU4::L4] enabled.
=====
====> 0.401 s: eFuse [PDU2::P2] tofire at 0.41 s.
=====
====> 0.41 s: eFuse [PDU2::P2] fired.
=====
    
```

Die Konsolenausgabe zeigt das Zuschalten der vier nichtsicherheitsrelevanten Lasten und dass die eFuse am Anschluss P2 der PDU2 mit Zuschalten der vierten Last wegen des hohen Gesamtstroms nach 1 ms ihr Auslösen ankündigt und bei 0,41 s auch auslöst. **Die sicherheitsrelevante Last Lenkung ist stromlos.**

2. Kommandiertes Verhalten

Analog zum nicht-kommandierten Verhalten erhält der Controller die Meldung des bevorstehenden Auslösens der eFuse am Versorgungsanschluss P2 bei 0,41 s zum Zeitpunkt 0,401 s. Er reagiert, indem er sämtliche nicht-sicherheitsrelevanten Lasten der beiden PDUs PDU2 und PDU4 abschaltet, um möglichst den Gesamtstrom von PDU1 zu PDU2 genügend zu reduzieren.

Controller-Ausgaben:

```
-----
====> 0.1 s: eFuse [PDU2::L2] enabled.
-----
====> 0.2 s: eFuse [PDU2::L3] enabled.
-----
====> 0.3 s: eFuse [PDU4::L1] enabled.
-----
====> 0.4 s: eFuse [PDU4::L4] enabled.
-----
====> 0.401 s: eFuse [PDU2::P2] tofire at 0.41 s.
-----
====> 0.401 s: eFuse [PDU2::L2] disabled.
-----
====> 0.401 s: eFuse [PDU2::L3] disabled.
-----
====> 0.401 s: eFuse [PDU4::L1] disabled.
-----
====> 0.401 s: eFuse [PDU4::L4] disabled.
```

Die Konsolenausgaben zeigen, dass durch das kommandierte Abschalten der vier nicht-sicherheitsrelevanten Lasten **die Lenkung weiter mit Strom wird und kein kritischer Zustand erreicht wird.**

4.2.6 Fazit

Die durchgeführten Simulationen des komplexeren Bordnetzmodells mit 32 eFuses zeigen, dass das Zusammenspiel zwischen zentralem Controller und den einzelnen eFuses zuverlässig funktioniert. In allen untersuchten Szenarien – Fehlerfälle an sicherheitsrelevanten Lasten, Überlast der elektronischen Lenkung, Zuschalten mehrerer nicht-sicherheitsrelevanter Verbraucher sowie serielle Versorgung von PDUs – konnten die eFuses drohende Überlastungen rechtzeitig an den Controller melden und bei Bedarf selbstständig auslösen.

Der entscheidende Mehrwert liegt jedoch im kommandierten Verhalten des Controllers: Durch gezieltes, vorausschauendes Abschalten nicht-sicherheitsrelevanter Lasten – bzw. bei vorhandener Backbone durch deren Zuschaltung – gelingt es in allen Testfällen, die Stromversorgung sicherheitsrelevanter Verbraucher wie der elektronischen Lenkung aufrechtzuerhalten und ein Auslösen der übergeordneten eFuses zu verhindern. Damit wird deutlich, dass ein intelligent gesteuertes eFuse-Management gegenüber einer rein passiven, nicht-kommandierten Absicherung einen erheblichen Sicherheitsgewinn bietet, da kritische Systeme auch unter Lastspitzen oder Fehlerbedingungen funktionsfähig bleiben. Die Ergebnisse bestätigen somit

die grundsätzliche Eignung des simulierten Steuerungskonzepts für komplexere, mehrstufig abgesicherte Bordnetzarchitekturen.

5 Simulationsmethodik

Für die Auslegung von eFuses und des durch sie abgesicherten Bordnetzes sind Systemsimulationen unverzichtbar: Nur so lässt sich prüfen, wie unterschiedliche eFuse-Typen auf realistische, meist aus Messfahrten abgeleitete Lastszenarien reagieren, welcher Typ für eine gegebene Bordnetzarchitektur geeignet ist und wie sich der Einsatz von eFuses auf Leitungsquerschnitte, Redundanzen und die Gesamtarchitektur auswirkt. Der Schutzpfad – bestehend aus Leitung, eFuse-Schaltelement, Treiber-/Steuerlogik und Diagnosefunktion – muss dabei sowohl in seinem schnellen elektrischen Schaltverhalten als auch in seinem langsameren thermischen Verhalten korrekt abgebildet werden.

Der Anspruch ist, verbesserte Simulationsverfahren für genau diese Aufgabenstellung bereitzustellen. Mit Dymola/Modelica, SystemC-AMS und dem FMI/FMU-Standard werden drei Bausteine vorgestellt, die – gezielt auf Schutzpfad- und eFuse-Anwendungen bezogen – gemeinsam eine effiziente, aber dennoch aussagekräftige Simulation ermöglichen. Es wird gezeigt, wie sich mittels FMU realer Mikrocontroller-/Schutzpfadcode auf HIL-Hardware beschleunigt ausführen lässt.

5.1 Herausforderungen der Schutzpfadsimulation

Die Simulation von eFuse-Schutzpfaden vereint mehrere Anforderungen, die einzeln schon anspruchsvoll sind und in Kombination die eigentliche Herausforderung ausmachen:

- Strom-Zeit-Auslösekennlinie (Itrip): Die eFuse muss innerhalb eines definierten Strom-Zeit-Fensters auslösen – weder zu früh (Fehlauslösung bei zulässigen Einschaltströmen) noch zu spät (unzureichender Leitungsschutz). Diese Kennlinie muss im Modell parametrierbar hinterlegt und über den gesamten Lastbereich korrekt nachgebildet werden.
- Safe Operating Area (SOA): Das Schaltelement darf nur innerhalb eines sicheren Betriebsbereichs aus Strom, Spannung und Temperatur betrieben werden. Die Simulation muss erkennen, wenn eine Lastkombination die SOA verlässt, auch wenn die Itrip-Zeit allein dies nicht anzeigt.
- Thermisch-elektrische Kopplung: Die Verlustleistung im Schaltelement und in der Leitung erwärmt das Bauteil, die Temperatur beeinflusst wiederum den Widerstand und damit die Auslösekennlinie selbst. Elektrisches und thermisches Verhalten müssen daher gekoppelt, nicht getrennt betrachtet werden.
- Stark unterschiedliche Zeitkonstanten („steife“ Systeme“): Kurzschlussabschaltungen laufen im μ s-Bereich ab, PWM-Ansteuerung typischerweise im kHz-Bereich, während thermische Ausgleichsvorgänge Sekunden bis Minuten dauern und Batteriezustände sich über eine ganze Fahrt entwickeln. Eine effiziente Simulation muss all diese Zeitskalen gleichzeitig handhaben, ohne die schnellste Dynamik für den gesamten Lauf als Schrittweite erzwingen zu müssen.
- Energieverbrauch/Verlustleistung als Systemgröße: Wie im Arbeitspaket beschrieben, betrachten Systemsimulationen häufig nur die Funktion, nicht aber den Energieverbrauch der Blöcke. Für den Schutzpfad ist gerade die Verlustleistung im Schaltelement die entscheidende Größe für Auslöseverhalten und thermische Belastung und muss daher explizit annotiert werden.
- Fehlerinjektion für die Absicherung: Für den Nachweis der Schutzfunktion müssen Kurzschlüsse, Überlasten und Kommunikationsfehler gezielt injiziert und deren Auswirkung auf Auslösezeit, SOA und Diagnosemeldung untersucht werden können – auch

das erfordert Multi-Level-fähige Modelle, die sowohl Fehlerursache (elektrisch) als auch Fehlerreaktion (Steuerlogik) abbilden.

5.2 Grundprinzipien effizienter Simulationsmethodik

5.2.1 So abstrakt wie möglich, so detailliert wie nötig

Der zentrale Gedanke effizienter Schutzpfadsimulation besteht darin, jede Modellkomponente nur so detailliert zu modellieren, wie es für die jeweilige Fragestellung notwendig ist. Für die Untersuchung eines gesamten Bordnetzes über einen kompletten Fahrzyklus genügt für die meisten eFuses ein abstrahiertes Schalter-Modell mit hinterlegter Itrip-Kennlinie und SOA-Grenze. Für die Bauteilauslegung, die thermische Grenzwertermittlung oder EMV-Untersuchungen ist dagegen ein elektrisch und thermisch fein aufgelöstes Modell des Schaltelements erforderlich. Effiziente Methodik bedeutet, beide Anforderungen im selben Simulationslauf zu vereinen, statt sich für das gesamte Bordnetz auf eine einzige Abstraktionsebene festzulegen.

5.2.2 Multi-Level- und Multi-Domänen-Simulation

Ein anschauliches, werkzeugunabhängiges Beispiel für den Nutzen von Multi-Level-Simulation liefert der Vergleich einer exakten Schaltungssimulation mit einer abstrahierten Verhaltenssimulation am Beispiel eines Phase-Locked Loop (PLL).

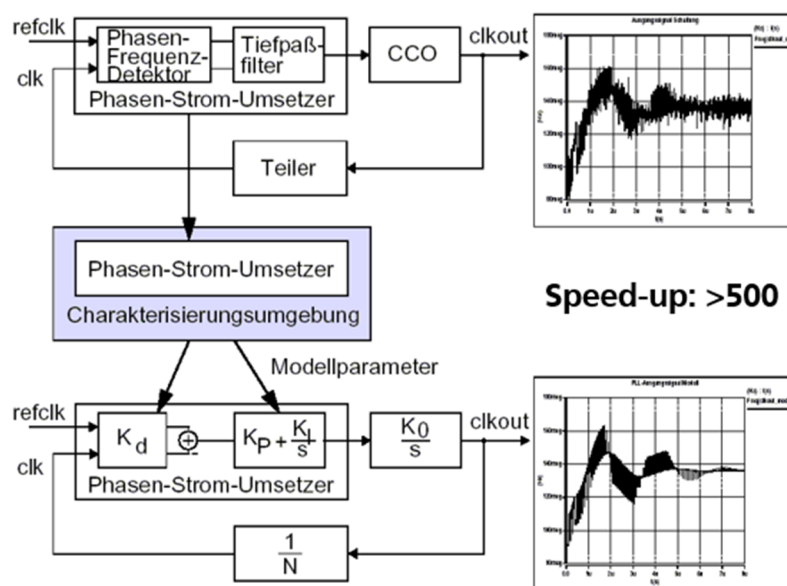


Abbildung 39: Phase-Locked-Loop (PLL)

Tabelle 3: Vergleich von exakter und abstrahierter Modellierung am Beispiel eines PLL

Kenngröße	Exakte Modellierung (Schaltungsebene)	Abstrakte Modellierung (Verhaltensebene)
Modellumfang	928 nichtlineare Elemente (Transistormodelle)	Kompaktes Verhaltensmodell
Simulationszeit für 10 μ s	$\approx$ 13 Stunden	$\approx$ 85 Sekunden

Kenngröße	Exakte Modellierung (Schaltungsebene)	Abstrakte Modellierung (Verhaltensebene)
Speedup	Referenz	> 500×

Der Vergleich zeigt, welches Effizienzpotenzial in der gezielten Abstraktion steckt: Ein auf Transistorebene modelliertes PLL mit 928 nichtlinearen Elementen benötigt für 10 μ s Simulationszeit rund 13 Stunden Rechenzeit, während ein äquivalentes Verhaltensmodell dieselbe Zeitspanne in etwa 85 Sekunden simuliert – ein Speedup von über 500. Dasselbe Prinzip lässt sich unmittelbar auf den eFuse-Schutzpfad übertragen.

Tabelle 4: Übertragung des Abstraktionsprinzips auf die eFuse-Schutzpfadmodellierung.

Kenngröße	Exaktes eFuse-Modell (Schaltungs-/Bauteilebene)	Abstrahiertes eFuse-Modell (Schutzfunktionsebene)
Modellumfang	MOSFET-Kennlinienfeld, parasitäre Elemente, verteiltes thermisches Netzwerk von Chip, Gehäuse und Leiterbahn	Ein-/Aus-Schalter mit variablem R_{on} , hinterlegte Itrip-Kennlinie und SOA-Grenze
Abgebildete Effekte	Nichtlineares Schaltverhalten, lokale Erwärmung, Alterung, EMV-relevantes Schaltflankenverhalten	Auslösezeitpunkt, mittlerer Verlustleistungsverlauf, Zustand (leitend/gesperrt/Fehler)
Typischer Einsatzzweck	Bauteilauslegung, thermische Grenzwertermittlung, EMV-Analyse	Bordnetz- und Architektur-Systemsimulation, Lastszenarien über gesamte Fahrzyklen
Rechenaufwand	Hoch – kleine Zeitschritte wegen steiler Schaltflanken erforderlich	Gering – Zustandswechsel ereignisbasiert, dazwischen große Schrittweiten möglich

Damit Multi-Level-Simulation praktisch nutzbar wird, sind zwei Voraussetzungen zentral: erstens Werkzeuge, die selbst mehrere Abstraktionsebenen und Domänen (elektrisch, thermisch, steuerungstechnisch) in einem Modell zulassen (Abschnitt 4.1 und 4.2), und zweitens simulatorübergreifende, offene Schnittstellen zur Modellintegration, wenn Teilmodelle – etwa exaktes eFuse-Bauteilmodell und Bordnetz-Systemmodell – aus unterschiedlichen Werkzeugen stammen (Abschnitt 4.3).

5.3 Methoden und Werkzeuge im Detail

5.3.1 Dymola / Modelica für Schutzpfad-Komponentenmodelle

Dymola (Dynamic Modeling Laboratory) ist eine von Dassault Systèmes entwickelte Modellierungs- und Simulationsumgebung, die auf der offenen, objektorientierten Sprache Modelica basiert. Modelica beschreibt physikalische Systeme akausal über Differential-algebraische Gleichungen (DAE) statt über signalflossorientierte Blockdiagramme. Dadurch lassen sich für den Schutzpfad elektrische Komponenten (Leitung, Schaltelement, Kontaktwiderstände) unmittelbar mit thermischen Komponenten (Wärmekapazität und -widerstand von Chip, Gehäuse, Leiterbahn) in einem gemeinsamen, wiederverwendbaren Modell koppeln, ohne dass

die Kopplung von Hand programmiert werden muss – die physikalische Erhaltungsgleichung (Wärmestrom = elektrische Verlustleistung) ergibt sich direkt aus der akausalen Verschaltung von elektrischem und thermischem Anschluss.

Für die eFuse-Modellierung ist das insbesondere für die Mixed-Signal-Makromodelle relevant: Ein- und Ausgangsstufe (Leistungsanschlüsse) werden auf elektrischer Ebene modelliert, um Pinompatibilität mit der umgebenden Bordnetzschaltung zu wahren, während das Schaltverhalten im Kern – abhängig vom Anwendungsfall – entweder als detailliertes MOSFET-Modell mit Kennlinienfeld oder als abstrahierter, zustandsbehafteter Schalter mit hinterlegter Itrip-/SOA-Kennlinie ausgeführt wird. Dymola bietet dafür umfangreiche, domänenübergreifende Modellbibliotheken (elektrisch, thermisch, regelungstechnisch) sowie die Möglichkeit, eigene Komponenten zu erstellen oder bestehende zu parametrieren. Es unterstützt zudem die Nachbildung von Echtzeit-Verhalten für Hardware-in-the-Loop-Simulationen und bietet vollständige FMI-Unterstützung (Export und Import).

Dymola damit primär als Werkzeug zur physikalisch fundierten, thermisch-elektrisch gekoppelten Modellierung der eFuse- und Leitungs-Komponenten sowie zur Herleitung und Validierung der abstrahierten Itrip-/SOA-Kennlinien relevant, die anschließend in schnelleren Systemmodellen verwendet werden.

5.3.2 SystemC-AMS im Detail

SystemC-AMS (Analog/Mixed-Signal Extensions, standardisiert als IEEE Std 1666.1-2016) erweitert die aus dem Chip- und Systementwurf stammende Bibliothek SystemC um Sprachmittel zur Modellierung analoger und gemischt analog-digitaler Systeme. Ziel ist die Erstellung ausführbarer, C++-basierter virtueller Prototypen, in denen digitale Hard-/Software-Architekturen (SystemC, TLM), eingebettete Treiber- und Schutzsoftware und kontinuierliche bzw. analoge Strompfad-Funktionalität nahtlos zusammenspielen – für den eFuse-Schutzpfad also genau die Kombination aus Leistungsstrompfad, Ansteuerlogik und Diagnosesoftware.

Die drei Modellierungsstile (Models of Computation)

- Timed Data Flow (TDF): zeitdiskrete, dataflow-basierte Modellierung mit statischem Ausführungsplan (TDF-Cluster). Für den Schutzpfad geeignet für die PWM-Ansteuerlogik, die zyklische Abtastung des Stroms zur Überstromerkennung sowie für Filter- und Glättungsfunktionen im Diagnosepfad. Da der Ausführungsplan zur Übersetzungszeit feststeht, entfällt der Overhead der dynamischen Ereignissteuerung des SystemC-Kerns, was die Simulation erheblich beschleunigt.
- Linear Signal Flow (LSF): kontinuierliche, signalflussorientierte Modellierung mit vordefinierten Blöcken (Übertragungsfunktionen, Integratoren, Regler). Geeignet für Regelkreise wie den Stromregler oder die Glättung des Verlustleistungssignals vor der Auswertung gegen die Itrip-Kennlinie.
- Electrical Linear Networks (ELN): konservative, netzwerkbasierende Modellierung mit klassischen elektrischen Komponenten wie Widerständen, Kondensatoren und (gesteuerten) Strom-/Spannungsquellen. Der eigentliche Leistungsstrompfad der eFuse – Leitungswiderstand, Schaltwiderstand, Lastkapazität – lässt sich hiermit knotenbasiert und damit physikalisch korrekt (Kirchhoffsche Gesetze) abbilden, ohne den Aufwand einer vollständigen Schaltungssimulation.

Zwischen den drei Bereichen vermitteln eingebaute Konverterports (z. B. sca_tdf2v/sca_v2tdf zwischen TDF und ELN), sodass digitale Schaltzustände aus der Ansteuerlogik unmittelbar in

den elektrischen Strompfad eingespeist und umgekehrt Ströme/Spannungen aus dem Strompfad wieder für die digitale Auswertung (Überstromerkennung, Diagnose) abgetastet werden können, ohne dass der Modellierer eigenen Konvertierungscode schreiben muss.

Da SystemC-AMS auf SystemC und damit auch auf Transaction-Level-Modeling (TLM) aufbaut, lässt sich der oben skizzierte Strompfad zusätzlich mit einem virtuellen Prozessormodell koppeln, auf dem die reale eFuse-Treiber- und Diagnosesoftware ausgeführt wird – etwa über die Anbindung eines Instruction-Set-Simulators (ISS) an das SystemC-AMS-Modell. Dadurch lässt sich bereits vor dem Aufspielen auf reale Hardware prüfen, ob die Steuerungssoftware korrekt auf simulierte Überstrom- und Kurzschlussereignisse reagiert, während der elektrische Strompfad weiterhin mit der für die Überstromerkennung notwendigen Zeitauflösung simuliert wird. Für Feedback-lastige Abschnitte des Schutzpfads (z. B. Stromregelung) und für ereignisgetriebene Abschnitte (z. B. Kurzschlussabschaltung) lassen sich unterschiedliche numerische Integrationsstrategien innerhalb von SystemC-AMS kombinieren, um die Simulation weiter zu beschleunigen.

5.3.3 FMI und Functional Mock-up Units (FMUs) im Detail

Das Functional Mock-up Interface (FMI) ist ein herstellerunabhängiger, offener Standard, der ursprünglich im Rahmen des Projekts MODELISAR spezifiziert wurde und heute unter dem Dach der Modelica Association weiterentwickelt wird. FMI definiert ein einheitliches Austauschformat für Simulationsmodelle unterschiedlicher Werkzeuge in Form von Functional Mock-up Units (FMUs) – zip-basierten Paketen, die eine XML-Schnittstellenbeschreibung (modelDescription.xml), die eigentliche Modellfunktionalität (als Binärcode oder Quellcode) sowie optionale Ressourcen und Dokumentation enthalten.

Für die eFuse-Modellierung ist das in FMI 3.0 neu eingeführte Clock-Element `pwm_clock` interessant. Es erlaubt, die Auswertung der Schutzlogik exakt auf die PWM-Taktung der eFuse-Ansteuerung zu synchronisieren.

FMI 3.0 definiert drei Interaktionsarten zwischen FMUs, die für unterschiedliche Abschnitte des Schutzpfads unterschiedlich geeignet sind:

- **Model Exchange (ME):** Die FMU liefert lediglich Zustandsableitungen; die numerische Integration erfolgt zentral durch das importierende Werkzeug (den „Master“). Für den elektrischen Strompfad vorteilhaft, wenn er gemeinsam mit weiteren Bordnetzkomponenten von einem einheitlichen, steifheitsangepassten Solver integriert werden soll.
- **Co-Simulation (CS):** Die FMU bringt einen eigenen Solver mit und führt die Zeitintegration selbstständig aus; der Master ruft dazu lediglich getter-/setter-Funktionen (`fmi3GetFloat64/fmi3SetFloat64` u. a.) an den vereinbarten Kommunikationspunkten auf. Für ein bereits fein austariertes eFuse-Modell mit eigenem thermisch-elektrischem Solver aus Dymola ist dies häufig die praktikablere Variante, da die Modellinterna nicht offengelegt werden müssen.
- **Scheduled Execution (SE):** ein externer, häufig echtzeitfähiger Scheduler löst über `fmi3ActivateModelPartition` gezielt einzelne Modellpartitionen aus. Dieses Ausführungsmodell entspricht unmittelbar der Funktionsweise realer Mikrocontroller-Firmware, die durch Timer- oder Interrupt-Ereignisse (z. B. den PWM-Takt der eFuse-Ansteuerung) getriggert wird, und ist damit die naheliegende Wahl, wenn realer Schutzpfad-Code auf eine HIL-Plattform gebracht werden soll.

5.3.4 Beschleunigung durch Ausführung von Schutzpfad-Code auf HIL-Hardware via FMU

Über die reine Modellintegration hinaus eröffnet der FMI-Standard eine weitere, für die eFuse-Absicherung besonders wirkungsvolle Möglichkeit: den nativen Mikrocontrollercode der eFuse-Ansteuer- und Schutzlogik – etwa die Überstromerkennung, die PWM-Ansteuerung oder die Itrip-Auswertung – nicht als abstrahiertes PC-Simulationsmodell nachzubilden, sondern als Source-Code- oder Objekt-FMU zu verpacken und auf eine echtzeitfähige HIL-Plattform zu laden. Die FMU enthält dabei denselben C-Code, der später auch auf dem realen eFuse-Steuergerät läuft; sie wird lediglich für die Zielhardware des HIL-Simulators übersetzt und über deren I/O-Kanäle mit dem übrigen, elektrisch nachgebildeten Bordnetzmodell verbunden.

Der Effizienzgewinn dieses Vorgehens ist demjenigen der bereits diskutierten Modell-Abstraktion ähnlich, entsteht aber auf andere Weise: Statt die Schutzlogik in einem allgemeinen, interpretierenden PC-Simulator nachzubilden, wird identischer, bereits für die Zielplattform kompilierter Code direkt auf spezialisierter Echtzeit- bzw. FPGA-Hardware ausgeführt. Dadurch entfällt der Interpretations- und Kommunikations-Overhead klassischer Simulationskerne, und die Schutzfunktion läuft in Echtzeit oder deutlich schneller als in Echtzeit. Es ergibt sich daraus die Möglichkeit, die eFuse-Ansteuer- und Schutzlogik unverändert vom Steuergerät auf einen HIL-fähigen Prüfstand zu übertragen und dort mit dem in Dymola/SystemC-AMS entwickelten, elektrisch-thermisch gekoppelten Bordnetzmodell inklusive gezielter Fehlerinjektion (Kurzschluss, Überlast) zu untersuchen.

5.3.5 Zusammenspiel der Methoden und Fazit

Effiziente Simulation von eFuse-Schutzpfaden in Fahrzeug-Bordnetzen erfordert mehr als ein einzelnes Werkzeug: Sie basiert auf dem Zusammenspiel geeigneter Modellierungsparadigmen, gezielter Abstraktion nach dem Prinzip „so detailliert wie nötig, so abstrakt wie möglich“ und offener Integrationsstandards.

Die vorgestellten Bausteine ergänzen sich: Dymola/Modelica liefert das physikalisch genaue, thermisch-elektrisch gekoppelte eFuse- und Leitungsmodell inklusive der für das Energiemanagement und die SOA-Prüfung benötigten Verlustleistungsannotation. SystemC-AMS ermöglicht die effiziente Kopplung dieses Strompfad-Modells (ELN) mit der Ansteuer- und Diagnosenlogik (TDF/LSF) sowie mit realer eingebetteter Software in einem gemeinsamen virtuellen Prototyp. FMI/FMU stellt sicher, dass Modelle aus diesen und weiteren Werkzeugen werkzeugunabhängig ausgetauscht und in einer gemeinsamen Co-Simulation zusammengeführt werden können und erlaubt darüber hinaus, denselben Mechanismus zu nutzen, um realen Schutzpfad-Code unverändert als FMU auf echtzeitfähige HIL-Hardware zu bringen und damit von einer reinen Softwaresimulation zu einer beschleunigten, hardwarenahen Absicherung überzugehen.

6 Zusammenfassung

Der Abschlussbericht „Simulationsbasierte Bewertung von eFuse-Verhalten im Kontext komplexer Fahrzeugbordnetze“ untersucht das Potenzial elektronischer Sicherungen (eFuses) für moderne, zunehmend zonal aufgebaute Fahrzeugbordnetze. Hintergrund ist, dass die CASE-Trends (Connectivity, Autonomy, Shared, Electrification) komplexere E/E-Architekturen mit mehreren Sicherungsebenen erfordern, in denen eFuses gegenüber klassischen Schmelzsicherungen zusätzliche Diagnose- und Steuerungsmöglichkeiten bieten.

Mittels einer Infineon BTG7003 wurden Ein-/Ausschaltverhalten sowie der I²t-Auslösealgorithmus messtechnisch validiert; ein angepasstes Simulationsmodell zeigt gute Übereinstimmung mit den Messungen. Zusätzlich wurde untersucht, wie präzise der Einschaltwiderstand zur Alterungs- und Temperaturdiagnose bestimmbar ist. Eine Worst-Case-Abschätzung ergab eine notwendige Sensorauflösung von mindestens 14 Bit; ein darauf trainiertes neuronales Netz konnte den Alterungszustand mit für die Praxis ausreichender Genauigkeit bestimmen, wobei die Diagnosegüte bei niedrigen Lastströmen deutlich abnimmt.

Ein zentraler, regelbasierter Lastmanagement-Algorithmus wurde entwickelt, der Verbraucher nach Prioritätsklassen bewertet, die Wirksamkeit von Abschaltmaßnahmen über eine Leitungswiderstandsmatrix zwischen PDUs abschätzt und bei drohender Unterspannung (Schwellwert mit Hysterese) gezielt nicht-sicherheitsrelevante Lasten abschaltet oder reduziert. In einer Simulation mit drei PDUs und 200 Verbrauchern konnte der Algorithmus in mehreren Szenarien (Einzeleingriff, stufenweise Leistungsreduktion, mehrfache Eingriffe) drohende Unterspannungen sicherheitsrelevanter Lasten erfolgreich verhindern.

In Modelica wurde die eFuse-Bibliothek (eFuse, Triggerlogik, PDU, Controller) erweitert und ein komplexes Bordnetz mit 32 eFuses aufgebaut. In allen getesteten Fehlerszenarien (Lastfehler, Überlast der elektronischen Lenkung, Zuschalten mehrerer Komfortlasten, serielle PDU-Versorgung) führte das nicht-kommandierte Verhalten zum Abschalten der sicherheitsrelevanten Last, während der zentrale Controller durch gezieltes Abschalten von Komfortlasten bzw. Zuschalten einer Backbone-Verbindung die Versorgung sicherheitsrelevanter Verbraucher aufrechterhalten konnte – ein klarer Sicherheitsgewinn intelligenten eFuse-Managements gegenüber rein passiver Absicherung.

Es wurden Methoden für eine effiziente Schutzpfadsimulation vorgestellt, die den Herausforderungen unterschiedlicher Zeitkonstanten, thermisch-elektrischer Kopplung und Fehlerinjektion begegnen. Leitprinzip ist „so abstrakt wie möglich, so detailliert wie nötig“ Vorgestellt werden Dymola/Modelica für physikalisch gekoppelte Komponentenmodelle, SystemC-AMS (TDF/LSF/ELN) zur Kopplung von Steuerungssoftware mit dem elektrischen Strompfad sowie der FMI/FMU-Standard, der auch die Ausführung realen Schutzpfad-Codes auf HIL-Echtzeithardware ermöglicht.

Es zeigt sich, dass eFuses weit über den Ersatz von Schmelzsicherungen hinaus als aktives, diagnosefähiges Element eines intelligenten, mehrstufigen Bordnetz-Energiemanagements genutzt werden können.

7 Literaturverzeichnis

- [1] Forschungsvereinigung Automobiltechnik (FAT) im Verband der Automobilindustrie (VDA), Roadmap für die Automobilität der Zukunft, Grundstein einer Innovationspartnerschaft, 2020.
- [2] Infineon Technologies AG, Automotive high-voltage electronic fuse demonstrator - User guide, [Online] https://www.infineon.com/dgdl/Infineon-UM_2211_PL52_2212_165057_Automotive_high_voltage.pdf-UserManual-v01_00-EN.pdf, 2022.
- [3] M. Rübartsch, M. Gerten und S. Frei, „On-Line Diagnosis of Supply Voltage Stabilizing Capacitors in Automotive Electronic Systems,“ in *VPPC*, 2023.
- [4] M. Gerten, S. Frei, M. Kiffmeier und O. Bettgens, „Influence of Electronic and Melting Fuses on the Transient Behavior of Automotive Power Supply Systems,“ *IEEE Transactions on Transportation Electrification*, Bd. 10, Nr. 2, 2024.
- [5] T. Markwirth, G. Pachiana, C. Sohrmann, M. Meddeb, G. Bublitz und H. Wagenonner, „Virtual testing of overtemperature protection algorithms in automotive smart fuses,“ *Design and Verification Conference and Exhibition Europe*, 2023.
- [6] FAT 305: Simulationsgestützte Methodik zum Entwurf intelligenter Energiesteuerung in zukünftigen Kfz-Bordnetzen, Berlin: Forschungsvereinigung Automobiltechnik E.V., 2018.
- [7] „The Modelica Association,“ [Online]. Available: <https://www.modelica.org/>. [Zugriff am 18 06 2020].
- [8] „OpenModelica,“ [Online]. Available: <https://www.openmodelica.org/>. [Zugriff am 18 06 2020].
- [9] M. Baumann, A. S. Abouzari, C. Mayer, S. S. Shekhawat, L. T. Peters und H.-G. Herzog, „Resource-Saving Modeling of an Electronic Fuse in Vehicular Power Systems,“ *IEEE Vehicle Power and Propulsion Conference*, 2023.
- [10] S. Schumi und F. Schipperges, „Short Circuit and Avalanche Effects in 12V Power Distribution for Automated Driving,“ *IEEE International Conference on Mechatronics*, 2019.
- [11] C. Mayer, M. Baumann, B. Eisenmann und H.-G. Herzog, „A Review of Electronic Fuses: Challenges and Opportunities for Future Vehicular Power Systems,“ *IEEE Transactions Transportation Electrification*, 2025.

- [12] Infineon Technologies AG, „Datenblatt BTS7008-1EPA Smart High-Side Power Switch,“ 2024. [Online]. Available: https://www.infineon.com/dgdl/Infineon-BTS7008-1EPA-DataSheet-v01_11-EN.pdf?fileId=5546d462636cc8fb0163fe7f1b8308ad.
- [13] STMicroelectronics, „Datenblatt VNF9D5F 2-channel high-side driver,“ 2025. [Online]. Available: <https://www.st.com/resource/en/datasheet/vnf9d5f.pdf>.
- [14] STMicroelectronics, „Datenblatt VNF1048F High-side switch Controller,“ 2023. [Online]. Available: <https://www.st.com/resource/en/datasheet/vnf1048f.pdf>.
- [15] M. Gerten, S. Frei, M. Kiffmeier und O. Bettgens, „Voltage Stability of Automotive Power Supplies During Tripping Events of Melting and Electronic Fuses,“ *VTC2022-Spring*, 2022.
- [16] L. Gysen, M. Ayeb und L. Brabetz, „Cable Bundle Protection and Cross-Section Reduction by using a centralized Smart Fusing Strategy,“ *IEEE ESARS-ITEC*, 2018.
- [17] S. Önal, A. Henke und S. Frei, „Switching Strategies for Smart Fuses Based on Thermal Models of Different Complexity,“ *International Conference on Ecological Vehicles and Renewable Energies (EVER)*, 2020.
- [18] A. Henke und S. Frei, „Fast Analytical Approaches for the Transient Axial Temperature Distribution in Single Wire Cables,“ *IEEE Transactions on Industrial Electronics*, Bd. 69, Nr. 4, 2022.
- [19] N. Zaman, *Automotive Electronics Design Fundamentals*, Springer International Publishing Switzerland, 2015.
- [20] A. E. Mejdoubi, A. Oukaour, H. Chaoui, H. Gualous, J. Sabor und Y. Slamani, „Prediction Aging Model for Supercapacitor's Calendar Life in Vehicular Applications,“ *IEEE Transactions on Vehicular Technology*, Bd. 65, Nr. 6, pp. 4253-4263, 2016.
- [21] M. Rübartsch und S. Frei, „A Novel Approach for Testing the Robustness of Networked Components in Vehicles Against Dynamic Power Supply Faults,“ *IEEE Transactions on Industry Applications*, Bd. early access, pp. 1-9, 2025.
- [22] Forschungsvereinigung Automobiltechnik e.V., *FAT-Schriftenreihe 378: Diagnosekonzepte für zonale und teilredundante Bordnetze*, Forschungsvereinigung Automobiltechnik e.V., 2024.

Abbildungsverzeichnis

Abbildung 1: Beispielhafte Sicherungsebenen im Bordnetz. eFuses sind der Anschaulichkeit wegen von PDUs getrennt.	- 6 -
Abbildung 2: Ausgewählte Infineon PROFET Wire Guard BTG7003.....	- 9 -
Abbildung 3: Parametrierung der Schutzfunktionen der ausgewählten eFuse	- 10 -
Abbildung 4: Grundlegende Funktionsblöcke elektronischer Sicherungen	- 10 -
Abbildung 5: Vergleich zwischen Messung und Simulation des parametrierten Simulationsmodells	- 11 -
Abbildung 6: Status der $I2t$ -Berechnung bei unterschiedlichen Belastungsströmen	- 11 -
Abbildung 7: Vergleich zwischen Messung und Simulation des parametrierten Simulationsmodells	- 12 -
Abbildung 8: Sensorkonzept zur Bestimmung des Einschaltwiderstands	- 13 -
Abbildung 9: Modellierung des Temperaturverhaltens in Abhängigkeit des stationären Dauerstroms.....	- 15 -
Abbildung 10: Modellierung des Alterungsverhaltens	- 16 -
Abbildung 11: Worst-Case Abschätzung der notwendigen Quantisierung der Sensorik ..	- 17 -
Abbildung 12: Ein- und Ausgangsgrößen zur Diagnose des Alterungszustands	- 17 -
Abbildung 13: Analysen der Alterungsbestimmung.....	- 18 -
Abbildung 14: Analysen der Alterungsbestimmung.....	- 19 -
Abbildung 15: Beispielhafte schematische Darstellung eines kommandierten Sicherungsbetriebs im Rahmen einer zentralisierten Bordnetzüberwachung.....	- 20 -
Abbildung 16: Erweiterung des bestehenden eFuse-Modells. Simscape-Blockschaltbild mit externem Schaltsignal (oben) und erweiterte Auslöselogik (unten)	- 22 -
Abbildung 17: Schematische Darstellung des zentralen Lastmanagements in elektronisch abgesicherten Bordnetzen	- 23 -
Abbildung 18: Minimalbeispiel zur Verdeutlichung der Wirksamkeitsbeurteilung	- 24 -
Abbildung 19: Zustandsdiagramm des resultierenden Lastmanagement-Algorithmus	- 26 -
Abbildung 20: Topologie des untersuchten Energiebordnetzes.....	- 27 -
Abbildung 21: Screenshot der Implementierung des Beispielsystems inklusive Lastmanagement in MATLAB/Simscape.....	- 28 -
Abbildung 22: eFuse-Ströme der Komfortlasten (links) und der sicherheitskritischen Lasten (rechts) während Szenario 1	- 29 -
Abbildung 23: Lastspannungen der Komfortlasten (links) und der sicherheitskritischen Lasten (rechts) während Szenario 1	- 29 -
Abbildung 24: eFuse-Ströme der Komfortlasten (links) und der Lastspannungen der sicherheitskritischen Lasten (rechts) während Szenario 2	- 30 -
Abbildung 25: eFuse-Ströme der Komfortlasten (links) und der Lastspannungen der sicherheitskritischen Lasten (rechts) während Szenario 3	- 31 -
Abbildung 26: Symbol der erweiterten eFuse	- 32 -
Abbildung 27: Komponenten des eFuse-Modells.....	- 34 -
Abbildung 28: Symbol von eFuse_triggering.....	- 35 -
Abbildung 29: Symbol einer PDU	- 37 -
Abbildung 30: Komponenten der PDU	- 38 -
Abbildung 31: Symbol des Controllers - übergeordnete Steuereinheit	- 41 -
Abbildung 32: komplexeres Bordnetzmodell in Modelica	- 44 -
Abbildung 33: Testszenarium	- 45 -

Abbildung 34: Stromverläufe beim Testszenarium (x-Achse: Zeit [s])	- 46 -
Abbildung 35: Ausfall der elektronischen Lenkung, Variante 1	- 47 -
Abbildung 36: Ausfall der elektronischen Lenkung, Variante 2	- 48 -
Abbildung 37: Zuschalten von mehreren nicht-sicherheitsrelevanten Verbrauchern	- 50 -
Abbildung 38: Serielle Versorgung zweier PDUs	- 52 -
Abbildung 39: Phase-Locked-Loop (PLL)	- 56 -

Tabellenverzeichnis

Tabelle 1: Nominalströme der Lasten des untersuchten Bordnetzes.....	- 27 -
Tabelle 2: Kapazität und ESR der Stabilisierungskondensatoren der jeweiligen Laten....	- 27 -
Tabelle 3: Vergleich von exakter und abstrahierter Modellierung am Beispiel eines PLL. -	56 -
Tabelle 4: Übertragung des Abstraktionsprinzips auf die eFuse-Schutzpfadmodellierung.-	57 -

Anhang

A.1 Auswertelogik des zentralen Leistungsmanagements

```
function [PDU1_fuse_states_,PDU2_fuse_states_, PDU3_fuse_states_] = power_man-
agement(PDU1_currents, PDU1_voltages, PDU2_currents, PDU2_voltages,
PDU3_currents, PDU3_voltages)

    % define comfort and safety-relevant loads
    PDU1_comfort_idx = 1:2:13;
    PDU1_safety_relevant_idx = 2:2:13;

    PDU2_comfort_idx = 1:2:23;
    PDU2_safety_relevant_idx = 2:2:23;

    PDU3_comfort_idx = 1:2:13;
    PDU3_safety_relevant_idx = 2:2:13;

    % eFuse switching states
    persistent PDU1_fuse_states;
    persistent PDU2_fuse_states;
    persistent PDU3_fuse_states;

    % Effective supply resistance
    % E.g. R_PDU1_PDU2: Current change delta_I at PDU 1, approximate voltage
    change at PDU 2 delta_V = delta_I*R_PDU1_PDU2
    % E.g. R_PDU3_PDU3: Current change delta_I at PDU 3, approximate voltage
    change at PDU 3 delta_V = delta_I*R_PDU3_PDU3
    R_PDU1_PDU1 = 5e-3;
    R_PDU1_PDU2 = 5e-3;
    R_PDU1_PDU3 = 5e-3;
    R_PDU2_PDU1 = 5e-3;
    R_PDU2_PDU2 = 6e-3;
    R_PDU2_PDU3 = 5e-3;
    R_PDU3_PDU1 = 5e-3;
    R_PDU3_PDU2 = 5e-3;
    R_PDU3_PDU3 = 7e-3;
```

```

% Thresholds, hysteresis
V_warn = 9.3;
V_crit = 9.2;
% minimal necessary improvement
min_improvement = 1e-3;

% Comfort load priorities (1 = niedrigster "Kosten"-Faktor).
P1_priorities = ones(size(PDU1_comfort_idx));
P2_priorities = ones(size(PDU2_comfort_idx));
P3_priorities = ones(size(PDU3_comfort_idx));

% default: all loads/eFuses activated
if isempty(PDU1_fuse_states)
    PDU1_fuse_states = ones(1, 13);
end
if isempty(PDU2_fuse_states)
    PDU2_fuse_states = ones(1, 23);
end
if isempty(PDU3_fuse_states)
    PDU3_fuse_states = ones(1, 13);
end

% normalize
PDU1_currents = PDU1_currents(:)';
PDU2_currents = PDU2_currents(:)';
PDU3_currents = PDU3_currents(:)';
PDU1_voltages = PDU1_voltages(:)';
PDU2_voltages = PDU2_voltages(:)';
PDU3_voltages = PDU3_voltages(:)';

% temporary copies
currents1 = PDU1_currents;
currents2 = PDU2_currents;
currents3 = PDU3_currents;
volts1 = PDU1_voltages;
volts2 = PDU2_voltages;
volts3 = PDU3_voltages;

% current minimum voltage of sr-loads
minV1 = min(volts1(PDU1_safety_relevant_idx));
minV2 = min(volts2(PDU2_safety_relevant_idx));
minV3 = min(volts3(PDU3_safety_relevant_idx));
global_minV = min([minV1, minV2, minV3]);

% If no voltage below warning level, do nothing
if global_minV >= V_warn
    PDU1_fuse_states_ = PDU1_fuse_states;
    PDU2_fuse_states_ = PDU2_fuse_states;
    PDU3_fuse_states_ = PDU3_fuse_states;
    return;
end

% Iteratives Greedy-Shedding
max_iterations = 20; % safety
iter = 0;

% Max possible candidates = total comfort outputs
maxCand = length(PDU1_comfort_idx) + length(PDU2_comfort_idx) +
length(PDU3_comfort_idx);

```

```

pdu_list = zeros(1, maxCand);      % which PDU (1/2/3)
idx_list = zeros(1, maxCand);      % index within PDU
deltaI_list = zeros(1, maxCand);   % current contribution
priority_list = zeros(1, maxCand); % priority
score_list = -inf(1, maxCand);     % score
improvement_list = zeros(1, maxCand);
dV_mat = zeros(maxCand, 3);        % dV for PDU1,PDU2,PDU3
nCand = 0;

while (global_minV < V_warn) && (iter < max_iterations)
    iter = iter + 1;

    % Reset candidate counter and arrays
    nCand = 0;
    score_list(:) = -inf;
    improvement_list(:) = 0;
    dV_mat(:, :) = 0;
    pdu_list(:) = 0;
    idx_list(:) = 0;
    deltaI_list(:) = 0;
    priority_list(:) = 0;

    % Gather candidates (only currently ON comfort loads with I>0)
    % PDU1
    for k = 1:length(PDU1_comfort_idx)
        idx = PDU1_comfort_idx(k);
        if (PDU1_fuse_states(idx) == 1) && (currents1(idx) > 0)
            nCand = nCand + 1;
            pdu_list(nCand) = 1;
            idx_list(nCand) = idx;
            deltaI_list(nCand) = currents1(idx);
            priority_list(nCand) = P1_priorities(k);
        end
    end
    % PDU2
    for k = 1:length(PDU2_comfort_idx)
        idx = PDU2_comfort_idx(k);
        if (PDU2_fuse_states(idx) == 1) && (currents2(idx) > 0)
            nCand = nCand + 1;
            pdu_list(nCand) = 2;
            idx_list(nCand) = idx;
            deltaI_list(nCand) = currents2(idx);
            priority_list(nCand) = P2_priorities(k);
        end
    end
    % PDU3
    for k = 1:length(PDU3_comfort_idx)
        idx = PDU3_comfort_idx(k);
        if (PDU3_fuse_states(idx) == 1) && (currents3(idx) > 0)
            nCand = nCand + 1;
            pdu_list(nCand) = 3;
            idx_list(nCand) = idx;
            deltaI_list(nCand) = currents3(idx);
            priority_list(nCand) = P3_priorities(k);
        end
    end

    % If no candidates -> break

```

```

if nCand == 0
    break;
end

% Current minima
cur_minV1 = min(volts1(PDU1_safety_relevant_idx));
cur_minV2 = min(volts2(PDU2_safety_relevant_idx));
cur_minV3 = min(volts3(PDU3_safety_relevant_idx));
cur_global_minV = min([cur_minV1, cur_minV2, cur_minV3]);

% Evaluate candidates
for iCand = 1:nCand
    deltaI = deltaI_list(iCand);
    pdu_src = pdu_list(iCand);

    % compute dV on each PDU
    if pdu_src == 1
        dV1 = deltaI * R_PDU1_PDU1;
        dV2 = deltaI * R_PDU1_PDU2;
        dV3 = deltaI * R_PDU1_PDU3;
    elseif pdu_src == 2
        dV1 = deltaI * R_PDU2_PDU1;
        dV2 = deltaI * R_PDU2_PDU2;
        dV3 = deltaI * R_PDU2_PDU3;
    else % pdu_src == 3
        dV1 = deltaI * R_PDU3_PDU1;
        dV2 = deltaI * R_PDU3_PDU2;
        dV3 = deltaI * R_PDU3_PDU3;
    end

    % predicted new voltages (simple additive model)
    new_volts1 = volts1 + dV1;
    new_volts2 = volts2 + dV2;
    new_volts3 = volts3 + dV3;

    % new minima for safety-relevant
    new_minV1 = min(new_volts1(PDU1_safety_relevant_idx));
    new_minV2 = min(new_volts2(PDU2_safety_relevant_idx));
    new_minV3 = min(new_volts3(PDU3_safety_relevant_idx));
    new_global_minV = min([new_minV1, new_minV2, new_minV3]);

    % improvement
    improvement = new_global_minV - cur_global_minV;

    % priority penalty (aggressive if below critical)
    if cur_global_minV < V_crit
        priority_penalty = priority_list(iCand) * 0.5;
    else
        priority_penalty = priority_list(iCand);
    end

    % compute score (guard division)
    if priority_penalty <= 0
        score = improvement;
    else
        score = improvement / priority_penalty;
    end

    % store results

```

```
        score_list(iCand) = score;
        improvement_list(iCand) = improvement;
        dV_mat(iCand,1) = dV1;
        dV_mat(iCand,2) = dV2;
        dV_mat(iCand,3) = dV3;
    end

    % choose best candidate
    [best_score, best_idx_rel] = max(score_list(1:nCand));

    % if best improvement too small -> stop
    if best_score < min_improvement
        break;
    end

    % apply shutdown for chosen candidate
    chosen_pdu = pdu_list(best_idx_rel);
    chosen_idx = idx_list(best_idx_rel);
    chosen_dV = dV_mat(best_idx_rel, :);

    if chosen_pdu == 1
        PDU1_fuse_states(chosen_idx) = 0;
        currents1(chosen_idx) = 0;
    elseif chosen_pdu == 2
        PDU2_fuse_states(chosen_idx) = 0;
        currents2(chosen_idx) = 0;
    else % chosen_pdu == 3
        PDU3_fuse_states(chosen_idx) = 0;
        currents3(chosen_idx) = 0;
    end

    % update voltages with estimated effect
    volts1 = volts1 + chosen_dV(1);
    volts2 = volts2 + chosen_dV(2);
    volts3 = volts3 + chosen_dV(3);

    % update global minima
    minV1 = min(volts1(PDU1_safety_relevant_idx));
    minV2 = min(volts2(PDU2_safety_relevant_idx));
    minV3 = min(volts3(PDU3_safety_relevant_idx));
    global_minV = min([minV1, minV2, minV3]);

end % while

% return persistent states
PDU1_fuse_states_ = PDU1_fuse_states;
PDU2_fuse_states_ = PDU2_fuse_states;
PDU3_fuse_states_ = PDU3_fuse_states;
end
```

Bisher in der FAT-Schriftenreihe erschienen (ab 2021)

Nr.	Titel
340	Schnelle, breitbandige Datenübertragung zwischen Truck und Trailer als Voraussetzung für das hochautomatisierte Fahren von Lastzügen, 2021
341	Wasserstoffkompatibilität von Aluminium-Legierungen für Brennstoffzellenfahrzeuge, 2021
342	Anforderungen an eine elektrische Lade- und Wasserstoffinfrastruktur für gewerbliche Nutzfahrzeuge mit dem Zeithorizont 2030, 2021
343	Objective assessment of database quality for use in the automotive research and development process, 2021
344	Review of non-exhaust particle emissions from road vehicles, 2021
345	Ganzheitliche Betrachtung von Rollwiderstandsverlusten an einem schweren Sattelzug unter realen Umgebungsbedingungen, 2021
346	Studie zur Abschätzung der Anwendungspotentiale, Risiken und notwendigen Forschungsbedarfe bei der Verwendung von Glashohlkugeln in Kombination mit thermoplastischem Schaumspritzguss, 2021
347	Typgenehmigungsanforderungen an Level-3-Autobahnssysteme - Hintergrundbetrachtungen zu technischen Anforderungen für eine automatisierte Fahrfunktion, 2021
348	Einfluss der Kantenbearbeitung von Aluminiumblechen auf das Restumformvermögen sowie die Festigkeitseigenschaften unter quasistatischer und schwingender Beanspruchung, 2021
349	Verstärkung dünner formgehärteter Bauteile mittels FVK-Verrippungen, 2021
350	HMI Anforderungen für den automatisierten Individualverkehr unter Berücksichtigung von Leistungsmöglichkeiten und -grenzen älterer Nutzer, 2021
351	Compatibility of polymers for fuel cell automobiles, 2021
352	Entwicklung einer gewichtsoptimierten Batteriegehäusestruktur für Volumenfahrzeuge, 2021
353	Charakterisierung und Modellierung des Deformations- und Versagensverhaltens von nicht-faserverstärkten Thermoplasten unter mehrachsiger Crashbelastung, 2021
354	Untersuchung zum thermischen Komfort im Pkw für den Grenzbereich des Luftzugempfindens, 2021
355	Anforderungen an die Güte, Verfügbarkeit und Vorausschau einer Reibwertschätzung aus Funktionssicht, 2021
356	Entwicklung einer standardisierten Prüfanordnung zur Bewertung der Übernahmeleistung beim automatisierten Fahren, 2022
357	Vorstudie zu Verkehrsemissionen - Räumlich und zeitlich aufgelöste Daten durch Schwarmmessungen, 2022
358	Produktivitätssteigerung und Kostensenkung der laser-additiven Fertigung für den Automobilbau, 2022
359	Analyse der Einflussfaktoren auf die Abweichung zwischen CFD und Fahrversuch bei der Bestimmung des Luftwiderstands von Nutzfahrzeugen mit Fokus auf den Ventilationswiderstand von Nfz-Rädern, 2022
360	Werkstoffmodelle und Kennwertermittlung für die industrielle Anwendung der Umform- und Crash-Simulation unter Berücksichtigung der thermischen Behandlungen beim Lackieren im Prozess bei hochfesten Werkstoffen, 2022

- 361 Compatibility of polymers for fuel cell automobiles, 2022
- 362 Ermüdung kurzfaserverstärkter thermoplastischer Polymerwerkstoffe, 2022
- 363 Market research and definition of procedure to comparison of comfort measuring systems for a vehicle cabin, 2022
- 364 Methodische Ansätze zur Auswahl von Bordnetzstrukturen mit erhöhten Zuverlässigkeitsanforderungen, 2022
- 365 Fahrwiderstand von Lenk- und Liftachsen in Kurven und auf gerader Strecke unter realen Umgebungsbedingungen, 2022
- 366 Klimadaten und Nutzungsverhalten zu Auslegung, Versuch und Simulation an Kraftfahrzeug-Kälte-/Heizanlagen, 2022
- 367 Experimentelle und numerische Untersuchung des selbsttätigen Losdrehens von Schraubenverbindungen mit konstanten und variablen Amplituden und Entwicklung einer Bewertungsmethode, 2022
- 368 Objective assessment of database quality for use in the automotive research and development process – Part 2, 2023
- 369 Level 2 hands-off – Recommendations and guidance, 2023
- 370 Funktionale Sicherheitsbewertung und Cybersecurity Analysen relevanter Use Cases für die Datenübertragung zwischen Truck und Trailer als Voraussetzung für das hochautomatisierte Fahren von Lastzügen, 2023
- 371 Study on the technical evaluation of decentralization based de-identification procedures for personal data in the automotive sector, 2023
- 372 Legal evaluation of decentralization based de-identification procedures for personal and non-personal data in the automotive sector, 2023
- 373 Quantifizierung der mechanischen Belastbarkeit von Infrarot-Schweißverbindungen in zyklisch belasteten Thermoplast-Bauteilen, 2023
- 374 Lebensdauerbewertung von geschweißten Verbindungselementen unter Montagevorspannung, 2023
- 375 Einfluss verschiedener Scherschneidparameter auf die elektro-magnetischen Eigenschaften von NO-Elektroblech automobiler Traktionsantriebe, 2023
- 376 Automatisierte Demontage von Traktionsmotoren der E-Mobilität - Eine Studie zur Optimierung der Demontage, 2023
- 377 Untersuchungen zum Einfluss von feuchtem Wasserstoff auf die Spannungsrisskorrosionsempfindlichkeit von Aluminium-Legierungen für den Einsatz in Brennstoffzellenfahrzeugen, 2024
- 378 Diagnosekonzepte für zonale und teilredundante Bordnetzarchitekturen, 2024
- 379 Dynamische Erfassung und Beurteilung von Situationsbewusstsein im Kontext des automatisierten Fahrens, 2024
- 380 Charakterisierung zukunftssträchtiger Zellmaterialien im Hinblick auf deren Anforderungen an das Batteriepack, 2024
- 381 Fahrdynamik des Automatisierten Fahrens, 2024
- 382 Forschungsperspektiven für Mobilität in klimaneutralen Städten 2045 - Explorative Szenarioanalyse und innovationspolitische Handlungsempfehlungen, 2024
- 383 Codierung und Analyse der AO-Klassifikation für Fuß- und Sprunggelenksverletzungen zur Evaluation potentieller Langzeitfolgen, 2024
- 384 EMV-Nachweis der Störfestigkeit auf Komponenten- und Systemebene für FailOp ab Level 3 im Hinblick auf die Funktionssicherheit - Erster Projektteil, 2024

- 385 Retrospektive Berechnung des Crashpulses aus Fahrzeugdeformationen basierend auf EES-Berechnung von Fahrzeug-Voxelmodellen unter Berücksichtigung des zeitlichen Verlaufes, 2024
- 386 Energieverluste infolge von Rad-/Achsfeststellungen am schweren Sattelzug bei realen Umgebungsbedingungen, 2025
- 387 Recycling von Permanentmagneten und Bewertung der Rezyklierbarkeit von Permanentmagnet-Synchronmotoren, 2025
- 388 Graphen- und heuristikbasierte Topologieoptimierung von 3D-Crash-Strukturen von Personenkraftwagen, 2025
- 389 Bewertung und Erhöhung des Potentials von Binder Jetting durch Nutzung kosteneffizienter Stahlpulver für den Einsatz in der Automobilindustrie, 2025
- 390 Driver performance models as reference for the quality of automated driving functions, 2025
- 391 Experimentelle und numerische Analyse des Fließverhaltens von hochviskosen Wärmeleitstoffen im Fertigungsprozess, 2025
- 392 Maßnahmen zum Aufbau und Erhalt von Situationsbewusstsein im Kontext des automatisierten Fahrens, 2025
- 393 Reduktion von Kupfer und Eisen in geschmolzenem Aluminium – modernes Aluminium aus alten Fahrzeugen, 2025
- 394 PCM-F³: PCM Fit For Future, 2025
- 395 Blech-Umformeinfluss mittels Machine Learning für Konzept-Crashsimulationen (UmMatCra-ML), 2025
- 396 Absicherung moderner Fahrzeug-Bordnetze mittels elektronischer Sicherungen (eFuses), 2025
- 397 Fahrdynamik des automatisierten Fahrens 2.0, 2025
- 398 Machbarkeitsstudie zur Koppelung von Methoden des maschinellen Lernens und CFD zur Vorhersage von Be- und Enttaugung an Fahrzeugscheiben, 2025
- 399 Charakterisierung der Radumströmung bei realer Reifendeformation, 2026
- 400 Reduzierung von Kaltaufschweißungen durch gezielte Anpassung des Seebeck-Koeffizienten von Werkzeugwerkstoffen mittels Laserauftragschweißen, 2026
- 401 Variable Digitalisierung von Materialkarten für FVK und ihre Rezyklate, 2026
- 402 EMV-Nachweis der Störfestigkeit auf Komponenten- und Systemebene für FailOp ab Level 3 im Hinblick auf die Funktionssicherheit – Erster und zweiter Projektteil, 2026
- 403 Simulationsbasierte Bewertung von eFuse-Verhalten im Kontext komplexer Fahrzeugbordnetze, 2026

Impressum

Herausgeber	FAT Forschungsvereinigung Automobiltechnik e.V. Behrenstraße 35 10117 Berlin Telefon +49 30 897842-0 Fax +49 30 897842-600 www.fat-ev.de
ISSN	2192-7863
Copyright	Forschungsvereinigung Automobiltechnik e.V. (FAT) 2026

Verband der Automobilindustrie e.V. (VDA)
Behrenstraße 35, 10117 Berlin
www.vda.de
Twitter @VDA_online

VDA | Verband der
Automobilindustrie

Forschungsvereinigung Automobiltechnik e.V. (FAT)
Behrenstraße 35, 10117 Berlin
www.vda.de/fat

FAT | Forschungsvereinigung
Automobiltechnik