

Integrierte Modellierung von Zuverlässigkeit und dynamischem Verhalten mechatronischer Systeme

zur Erlangung des akademischen Grades eines
DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)
der Fakultät für Maschinenbau
der Universität Paderborn

genehmigte
DISSERTATION

von
Thorben Henning Kaul, M. Sc.
aus Höxter

Tag des Kolloquiums: 7. Mai 2020
Referent: Prof. Dr.-Ing. habil. Walter Sextro
Korreferent: Prof. Dr.-Ing. Detmar Zimmer

Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Dynamik und Mechatronik der Universität Paderborn.

Mein Dank gilt Herrn Prof. Dr.-Ing. habil. Walter Sextro für das entgegen gebrachte Vertrauen, diese Arbeit am Lehrstuhl zu verfassen. Das Interesse an meiner Arbeit, die fachlichen Diskussionen und die Freiheit zum selbstständigen wissenschaftlichen Arbeiten stellten eine fortwährende Unterstützung während meiner Zeit am Lehrstuhl dar.

Herrn Prof. Dr.-Ing. Detmar Zimmer danke ich für die freundliche Übernahme des Korreferats. Ebenso gilt mein Dank Herrn Prof. Dr.-Ing. Gunter Kullmer und Herrn Dr.-Ing. Tobias Hemsel für die Mitarbeit in der Promotionskommission.

Meinen Kollegen und meiner Kollegin am Lehrstuhl für Dynamik und Mechatronik danke ich für die stets sehr angenehme Arbeitsatmosphäre sowie die zahlreiche Unterstützung während der Erstellung dieser Dissertation. Besonders möchte ich mich bei Amelie Bender, Osarenren Aimiyeqagbon und Dr.-Ing. Tobias Meyer für die hervorragende Zusammenarbeit und die vielen fachlichen Diskussionen bedanken. Ebenso möchte ich mich bei Dr.-Ing. Tobias Hemsel bedanken, auf dessen vielfältige Unterstützung genauso zu zählen war wie auf die der alten und neuen „guten Seele“ Marina Kassühlke und Sandra Siekmann. Mein Dank gilt den Studierenden für das Vertrauen in die Begleitung ihrer Abschlussarbeiten und für die Zusammenarbeit am Lehrstuhl, durch die ich in meiner persönlichen Entwicklung profitiert habe.

Mein ganz besonderer Dank gilt meinen Eltern, Michaela und Werner, sowie meiner Schwester Maike. Ohne ihre Wegweisung und ihren Zuspruch hätte ich diesen Weg nie eingeschlagen und ohne ihre Unterstützung gewiss nicht erfolgreich begangen. Zu guter Letzt gilt mein größter Dank Katharina für ihre Liebe, ihr Verständnis an arbeitsreichen Tagen und ihren Rückhalt in unsicheren Phasen.

Paderborn, im Mai 2020

Thorben Kaul

Kurzfassung

Die digitale Transformation prägt die Entwicklung intelligenter technischer Systeme, welche durch Vernetzung und inhärente Intelligenz einen weiten Funktionsumfang aufweisen. Selbstoptimierende Systeme sind als Stellvertreter der Klasse intelligenter Systeme durch die autonome zielkonforme Adaption des Systemverhaltens charakterisiert. Dabei stellen mechatronische Systeme eine der Grundlagen dieser Systemklasse dar, indem die Umwelt- und Betriebsbedingungen sowie die Systemzustände sensorisch erfasst und das dynamische Systemverhalten zielgerichtet beeinflusst werden. Der große Funktionsumfang intelligenter technischer Systeme geht mit einer Zunahme der Systemkomplexität einher, die eine Herausforderung bei der Absicherung der Verlässlichkeit darstellt. Dem gegenüber bieten insbesondere selbstoptimierende Systeme Potenziale zur Steigerung der Verlässlichkeit. Die Umsetzung entsprechender Maßnahmen erfordert die Unterstützung des Entwicklungsprozesses durch geeignete Methoden. Die Beherrschung der zunehmenden Systemkomplexität ist durch die konsequente Verwendung vorhandener Modelle des Entwicklungsprozesses möglich, wird aber in aktuellen Methoden nicht umfänglich genutzt. Das Ziel ist die Absicherung der Verlässlichkeit bereits in frühen Entwicklungsphasen bis hin in späte Lebenszyklusphasen. Es wird eine Methode für die integrierte Modellierung der Zuverlässigkeit, als zentrale Kenngröße der Verlässlichkeit, und des dynamischen Systemverhaltens entwickelt. Anhand von drei Anwendungsbeispielen wird gezeigt, dass diese Methode die Entwicklung von Maßnahmen zur Steigerung der Verlässlichkeit in selbstoptimierenden Systemen ermöglicht, auf komplexe mechatronische Systeme anwendbar ist und die Absicherung der Verlässlichkeit während des Betriebs durch die Umsetzung eines Digitalen Zwillings unterstützt.

Abstract

The digital transformation establishes the development of intelligent technical systems, for which networks and inherent partial intelligence are mandatory features to enlarge their functionality. Self-optimizing systems as representatives of intelligent systems, are characterized by the autonomous objective-conform adaption of system behavior. Mechatronic systems provide the basic principles of this system class by sensing environmental and operating conditions as well as system states. These capabilities allow for precise manipulation of dynamic system behavior. The wide functionality of intelligent technical systems leads to an increasing system complexity, which is a serious threat regarding their dependability. In contrast to this threat, self-optimizing systems offer means to increase dependability by exploiting potentials of advanced behavior adaption, which are laid in the sophisticated support of the development process. In order to govern the increasing system complexity, the existing models of the development process need to be exploited to analyze dependability from early development stages to subsequent life cycle phases. These existing models are not yet sufficiently used in current approaches. A method for the integrated modeling of reliability, the central attribute of dependability, and dynamic system behavior is developed. Three application examples are investigated to show, that the method is capable of supporting the development of means to increase dependability in self-optimizing systems, the application on complex mechatronic systems and the setup of a digital twin in order to analyze dependability during operation.

Vorveröffentlichungen

Teile der vorliegenden Dissertation oder dazugehörige Vorarbeiten wurden bereits in den nachfolgenden Arbeiten veröffentlicht.

Begutachtete Zeitschriftenartikel

KAUL, T.; MEYER, T.; SEXTRO, W.: Formulation of Reliability-related Objective Functions for Design of Intelligent Mechatronic Systems. In: *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 231 (2017), Nr. 4, S. 390–399.

Begutachtete Beiträge in Tagungsbänden

KAUL, T.; BENDER, A.; SEXTRO, W.: „Digital Twin for Reliability Analysis During Design and Operation of Mechatronic Systems“. In: *Proceedings of the European Safety and Reliability Conference (ESREL2019)*. Hrsg. von Beer und Zio. Singapore: Research Publishing, 2019, S. 2340–2348.

BENDER, A.; KAUL, T.; SEXTRO, W.: „Entwicklung eines Condition Monitoring Systems für Gummi-Metall-Elemente“. In: *Wissenschaftsforum Intelligente Technische Systeme (WInTeSys 2017)*. Paderborn: Verlagsschriftenreihe des Heinz Nixdorf Instituts 2017, S. 347–358.

HENTZE, J.; KAUL, T.; GRÄSSLER, I.; SEXTRO, W.: „Integrated Modeling of Behavior and Reliability in System Development“. In: *DS 87-4 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 4: Design Methods and Tools*. Vancouver 2017, S. 385–394.

SEXTRO, W.; MEYER, T.; KAUL, T.; KIMOTHO, J. K.: „Entwicklung verlässlicher intelligenter Systeme“. In: *VDI-Berichte 2307–28. Tagung Technische Zuverlässigkeit (TTZ 2017) - Entwicklung und Betrieb zuverlässiger Produkte*. Leonberg 2017, S. 17–30.

KAUL, T.; MEYER, T.; SEXTRO, W.: „Modeling of Complex Redundancy in Technical Systems with Bayesian Networks“. In: *Proceedings of the Third European Conference of the Prognostics and Health Management Society 2016*. Bilbao 2017, S. 437–445.

KAUL, T.; MEYER, T.; SEXTRO, W.: „Integrated Model for Dynamics and Reliability of Intelligent Mechatronic Systems“. In: *Proceedings of the European Safety and Reliability Conference (ESREL2015)*. Hrsg. von Podofillini et al. London: Taylor and Francis, 2015.

KAUL, T.; MEYER, T.; SEXTRO, W.: „Integrierte Modellierung der Dynamik und der Verlässlichkeit komplexer mechatronischer Systeme“. In: *10. Paderborner Workshop Entwurf mechatronischer Systeme*. Hrsg. von Gausemeier, J., Dumitrescu, R., Rammig, F., Schäfer, W. und Trächtler, A. Paderborn: Heinz Nixdorf Institut, Universität Paderborn, 2015, S. 101–112.

MEYER, T.; KAUL, T.; SEXTRO, W.: „Advantages of reliability-adaptive system operation for maintenance planning“. In: *Proceedings of the 9th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes*. 2015, S. 940–945.

Nicht begutachtete Beiträge in Tagungsbänden

KAUL, T.; HENTZE, J.; SEXTRO, W.; GRÄSSLER, I.: „Integration von Verlässlichkeitsmodellen der Entwicklung in einen Digitalen Zwilling zur Umsetzung einer vorausschauenden Instandhaltung“. In: *Tagungsband VDI-Mechatronik 2019*. Hrsg. von Bertram, T., Corves, B., Gräßler, I. und Janschek, K. Paderborn 2019, S. 19–24.

Buchbeiträge

MEYER, T.; KAUL, T.; SEXTRO, W.: „Steigerung der Verlässlichkeit mechatronischer Systeme“.
In: *Steigerung der Intelligenz mechatronischer Systeme*. Hrsg. von Trächtler, A. und Gausemeier, J. Berlin, Heidelberg: Springer Vieweg 2018, S. 193–213.

Inhaltsverzeichnis

Nomenklatur	xi
1 Einleitung	1
1.1 Aufbau der Arbeit	4
2 Verlässlichkeit intelligenter technischer Systeme	5
2.1 Grundlagen der Verlässlichkeit	5
2.2 Ausfallverhalten technischer Systeme	10
2.3 Selbstoptimierende technische Systeme	12
2.4 Mehrzieloptimierung	14
2.5 Entwicklung intelligenter, verlässlicher Systeme	15
2.6 Ganzheitliche Modellierung mechatronischer Systeme	18
2.7 Geschlossene Modellierung von Dynamik und Zuverlässigkeit	21
2.8 Fazit	25
3 Integrierte Modellierung von Zuverlässigkeit und dynamischem Verhalten	29
3.1 Integration in Lebenszyklusphasen mechatronischer Systeme	31
3.2 Systemmodell	36
3.3 Lokale Zuverlässigkeitsschätzer	37
3.4 Systemzuverlässigkeitsmodell	39
3.5 Automatisierte Ableitung des Systemzuverlässigkeitsmodells	63
3.6 Zusammenhang zu weiteren Kenngrößen der Verlässlichkeit	84
4 Anwendungsbeispiele	87
4.1 Zuverlässigkeitsadaptive Reibkupplung	87
4.2 Zuverlässigkeitsanalyse einer Windenergieanlage	98
4.3 Umsetzung eines Digitalen Zwillings für die Instandhaltung	105
5 Zusammenfassung und Ausblick	117
Literaturverzeichnis	121
A Anhang	133
A.1 Berechnung der vollständigen gemeinsamen Wahrscheinlichkeitsverteilung und Marginalisierung	133
A.2 Weiche Evidenzen in Fehlerursachen	133
A.3 Algorithmen zur Umsetzung des Zuverlässigkeitsmodells	135

Nomenklatur

Allgemeine Notation

x, X	Skalare Größen
\mathbf{x}, \mathbf{X}	Vektoren / Matrizen
\mathcal{X}	Ereignis
\hat{p}	Optimale Lösung
$(\cdot)^{-1}$	Inverse Funktion

Formelzeichen und Symbole

$\mathbf{A}_{DAG,i}$	Adjazenzmatrix der i -ten Zeitscheibe eines DBNs
A_E	Virtueller Knoten zur Modellierung einer weichen Evidenz über den Knoten A
α	Verhaltensparametrierung der zuverlässigkeitsadaptiven Regelung
α_{ist}	Gemessener Istwert der Verhaltensparametrierung
α_{Nutzer}	Nutzervorgabe der Verhaltensparametrierung
α_{soll}	Sollwert der Verhaltensparametrierung
α_R	Ausgang des Zuverlässigkeitsreglers der Verhaltensparametrierung
C	Dynamische Tragzahl eines Wälzlagers
Δt	Diskretisierungsschrittweite des Betrachtungszeitraums der Zuverlässigkeitsanalyse
η	Kopplungsfaktor zur Abbildung von Redundanzen
\mathbf{E}	Einheitsmatrix
$E(A)$	Weiche Evidenz über den Knoten A
E_f	Reibarbeit
$F(t)$	Verteilung der Ausfallwahrscheinlichkeit
f_l	Mittlere Oszillationsfrequenz innerhalb des l -ten Lastschritts
F_N	Normalkraft
HI	Health Index, abstrahiertes Lebensdauermaß
\widehat{HI}	Gemessener Health Index, abstrahiertes Lebensdauermaß
\widetilde{HI}	Geschätzter Health Index, abstrahiertes Lebensdauermaß
i	Index der Zeitscheiben eines DBN
j	Index der Knoten in i -ter Zeitscheibe eines DBN
k	Ertragbare Ausfälle einer Redundanz Betrachtungszeitraum der Zuverlässigkeitsanalyse in Kupplungsvorgängen
λ	Ausfallrate
$l(k)$	Abnahme des ertragbaren Verschleißes der Reibscheiben im k -ten Kupplungsvorgang
L_{10}	Lebensdauer für 10% ausgefallener Einheiten

l_{max}	Ertragbarer Verschleiß der Reibscheiben
m	Anzahl an Knoten in i -ter Zeitscheibe eines DBN Beteiligte Knoten an einer Resonanz
n	Anzahl an Zeitscheiben eines DBN Anzahl an Experimenten
ϕ_l	Mittlere Winkelamplitude innerhalb des l -ten Lastschritts
\mathbf{p}	freie Systemparameter
p	Berechnungsbeiwert der Wälzlagerauslegung
P	Dynamische äquivalente Belastung eines Wälzlagers
$P(\mathcal{A})$	Wahrscheinlichkeit, dass Ereignis \mathcal{A} eingetreten ist
$P(A, B)$	Gemeinsame Wahrscheinlichkeitsverteilung der Variablen A und B
$P(A B)$	Bedingte Wahrscheinlichkeitsverteilung der Variable A vorausgesetzt Variable B
$P(A = a_1)$	Wahrscheinlichkeit, dass Variable A im Zustand a_1 ist
P_{ea}	Dynamische äquivalente Belastung für Rotorblattlagerungen
$\text{Pa}(X)$	Menge der Elternknoten des Knotens X
p_f	Proportionalitätsfaktor zwischen Reibarbeit und Abnahme des ertragbaren Verschleißes
p_l	Last innerhalb des l -ten Lastschritts
\mathbf{p}_{SVM}	Diagnosewahrscheinlichkeiten einer Zustandsüberwachung auf Basis einer SVM
<i>operational,</i>	Boole'sche Zustände einer Variable
<i>failed</i>	
$R(t)$	Wahrscheinlichkeitsverteilung der Zuverlässigkeit
$R_{A B}(t)$	Lokale Zuverlässigkeit, siehe bedingte Wahrscheinlichkeit $P(A B)$
σ	Standardabweichung
$s(\cdot)$	s -Transformation
$s^{-1}(\cdot)$	Inverse s -Transformation
T	Charakteristische Lebensdauer der Weibull-Verteilung
t	Zeit
t_E	Zeitpunkt einer Beobachtung / Evidenz
t_i	Zeitpunkt im Betrachtungszeitraum der Zuverlässigkeitsanalyse
t_l	Relative zeitliche Dauer des l -ten Lastschritts
t_r	Dauer eines Kupplungsvorgangs
X_i^j	j -ter Knoten in der i -ten Zeitscheibe eines DBN
$\dot{\omega}$	Winkelbeschleunigung

Darstellung von Graphen



Element im Blockschaltbild



Komponente im Topologiegraph



Teil- / System im Topologiegraph



Fehl- / Ausfallzustand im Systemzuverlässigkeitsmodell



Allgemeiner Knoten | Fehlerursache im Systemzuverlässigkeitsmodell



Gerichtete Kante



Gerichtete temporäre Kante

Abkürzungen

AADL	Architecture Analysis and Design Language
BN	Bayes'sches Netz
CAD	Computer Aided Drawing
CAE	Computer Aided Engineering
CONSENS	CONceptual design Specification technique for the ENgineering of mechatronic Systems
CPD	Bedingte Wahrscheinlichkeitsverteilung (Conditional Probability Distribution)
CPT	Bedingte Wahrscheinlichkeitstabelle (Conditional Probability Table)
DAG	Gerichteter azyklischer Graph (Directed Acyclic Graph)
DBN	Dynamisches Bayes'sches Netz
FE	Finite Elemente
FEM	Finite Elemente Methode
FMEA	Fehlermöglichkeits- und -auswirkungsanalyse
FMI	Functional Mock-Up Interface
FSAP	Formal Safety Analysis Platform
HAZOP	Hazard and Operability Study
IWES	Fraunhofer Institute for Wind Energy and Energy System Technology
MBSE	Modelbased Systems Engineering
MOCET	Model based Certification Tool
MTTF	Mean Time To Failure
NREL	National Renewable Energy Laboratory
PFD	Probability of Failure on Demand
PFH	Probability of Failure per Hour
PLM	Produkt-Lebenszyklus-Management
PSO	Particle Swarm Optimization
QFT	Quick Fatigue Tool for Matlab
SE	Systems Engineering
SIL	Sicherheitsintegritätslevel (IEC 62508, EN 62062 und ISO 26262)
SVM	Support Vector Machine
UML	Unified Modeling Language
VHDL-AMS	Very High Speed Integrated Circuit Hardware Description Language – Analog and Mixed-Signals
WEA	Windenergieanlage

1 Einleitung

Die voranschreitende digitale Transformation offenbart vielfältige Herausforderungen und Risiken, die mit einer Teilhabe an einhergehenden Veränderungen aufkommen. Gleichzeitig impliziert sie keimende Chancen, die mit technologieinduzierten Disruptionen gesamtgesellschaftliche Aspekte unserer Ökonomie und Arbeitswelt betreffen. Eine erfolgreiche Partizipation und Gestaltung der vierten industriellen Revolution erfordert darüber hinaus technologische Ansätze wie intelligente technische Systeme [SMK+17, SGE17]. Somit wird eine Teilhabe an den weitreichenden Potenzialen ermöglicht, die vernetzte und autonom agierende moderne technische Systeme bereitstellen und führt zu neuen Geschäftsmodellen in der Wertschöpfungskette technischer Systeme.

In leistungsbezogenen Geschäftsmodellen, wie den Pay-Per-Use-Konzepten, zahlt ein Nutzer nur die vom System abgerufene Leistung ohne das System selbst zu erwerben [GFC13, Cis14, VDI16] – im Gegensatz zu klassischen Geschäftsmodellen, in denen das System selbst Gegenstand ist und typischerweise ein Eigentumswechsel stattfindet. Diesen neuen Geschäftsmodellen folgend wird der Hersteller der Systeme zum Anbieter der Leistung und hat ein hervorgehobenes Interesse an der Steigerung der Verlässlichkeit seiner Systeme, da der unternehmerische Erfolg nicht mehr über den einmaligen Verkauf der Systeme oder der Instandhaltungsverträge abgedeckt wird [VDI16, SMK+17]. In modernen technischen Systemen kommt der Steigerung der Verlässlichkeit somit eine zentrale Bedeutung zu. Mit einer Zunahme der Komplexität, der Interdisziplinarität und dem Grad der Vernetzung von modernen technischen Systemen erwächst gleichermaßen die Herausforderung, diese Systeme in der Entwicklung und im Betrieb zu analysieren und entsprechende Maßnahmen für die Gewährleistung eines zuverlässigen und sicheren Betriebs abzuleiten.

Moderne technische Systeme definieren sich zunehmend über eine Umsetzung ihrer Funktionalität auf Basis ihrer Informationsverarbeitung. In Kombination mit einer Erfassung der Umgebungs- und Betriebsbedingungen sowie einer umfassenden Aktuierung zur zielgerichteten Manipulation des Systemverhaltens ergibt sich ein breiter Funktionsumfang. Neben sensorischen und aktorischen Funktionalitäten integrieren diese Systeme auch kognitive Funktionsansätze. Damit stellen mechatronische Systeme nach wie vor eine der Grundlagen moderner technischer Systeme dar, die darüber hinausgehend eben jene inhärente Intelligenz und Vernetzung aufweisen [TG18]. Intelligente technische Systeme bestehen typischerweise aus Teilsystemen, die Teilfunktionalitäten des Gesamtsystems implementieren. Diese Teilsysteme sind oft geografisch verteilt und kommunizieren beziehungsweise kooperieren unter Ausnutzung ihrer Vernetzung. Eine übergeordnete Funktionalität ergibt sich durch die Interaktion der verteilten Teilsysteme. Dabei ist weder die übergeordnete Funktionalität noch die Rolle und die Anzahl der partizipierenden Teilsysteme statisch, sondern in Abhängigkeit von den Umwelt- und Betriebsbedingungen sowie den Nutzeranforderungen dynamisch. Die Funktionalität des Gesamtsystems wird daher nicht durch eine übergeordnete Steuerung vorgegeben, vielmehr wirken lokale Betriebsstrategien auf das globale Systemverhalten ein. Die dynamische Adaption der

Funktionalität erfordert ein hohes Maß an Autonomie des intelligenten Systems. Selbstoptimierende Systeme sind der Klasse der intelligenten technischen Systeme zugehörig und durch eine Erweiterung der Informationsverarbeitung mechatronischer Systeme charakterisiert [TG18]. Diese Erweiterung der Informationsverarbeitung erfolgt zumeist mithilfe von Ansätzen der adaptiven sowie optimierungsbasierten Regelungstechnik und künstlichen Intelligenz.

Die Selbstoptimierung stellt eine Erweiterung der Funktionalität mechatronischer Systeme dar und geht meist mit einer Zunahme der Systemkomplexität einher [Bir07, SMK+17, TG18]. Eine modellbasierte Absicherung der Anforderungen an das zu entwickelnde System gelingt nur mit einer umfänglichen Abbildung des Systemverhaltens bereits in frühen Phasen des Entwicklungsprozesses. Die Abbildung aller relevanten Aspekte ist durch leistungsfähige Modellierungswerkzeuge umzusetzen und umfasst häufig die Integration von Modellen verschiedener Domänen. Die Handhabung der zahlreichen Modelle erfordert wirksame Methoden, um die Konsistenz der Modelle zueinander während des gesamten Entwicklungsprozesses zu gewährleisten. Besonders in der Entwicklung von Maßnahmen zur Steigerung der Verlässlichkeit kommen zusätzliche Modelle zur Abbildung der unterschiedlichen Aspekte, wie der Verlässlichkeit und des dynamischen Systemverhaltens, zur Anwendung. Diese Modelle begleiten die Entwicklung von frühen Phasen bis in den detaillierten Entwurf eines Systems, seiner Teilsysteme und Systemelemente. Dementsprechend sind aus dem modellbasierten Entwicklungsprozess umfangreiche und zahlreiche Modelle verfügbar [HKG+17]. Der Transfer dieser Modelle in spätere Lebenszyklusphasen ist für eine effiziente Umsetzung verlässlichkeitssteigernder Maßnahmen elementar und durch geeignete Methoden sowie Vorgehensmodelle zu unterstützen. Eine Kombination des bestehenden Vorwissens, in Form von Modellen, mit den durch zunehmende Digitalisierung mechatronischer Systeme zur Verfügung stehenden Daten bietet dabei weitreichende Potenziale.

Ein zentraler Aspekt des Entwicklungsprozesses intelligenter technischer Systeme und insbesondere mechatronischer Systeme ist die Absicherung des dynamischen Systemverhaltens. Mechatronische Systeme, als eine der Grundlagen intelligenter technischer Systeme, sind in der Lage, ihr Systemverhalten zielgerichtet mittels geregelter Aktorik zu beeinflussen. Ein zentrales Hilfsmittel, für die Auslegung und Absicherung der Regelkreise und weiterer Funktionalitäten, sind detaillierte Modelle des dynamischen Systemverhaltens während des Entwicklungsprozesses. Neben der Absicherung von Systemanforderungen an die Dynamik, wird das dynamische Systemverhalten als maßgebliche Eingangsgröße in der Bestimmung von Teilsystem- und Komponentenlebensdauern verwendet. Beispielsweise erfolgt die Bestimmung des Ermüdungsverhaltens von Strukturbauteilen mithilfe von Methoden der Betriebsfestigkeitsanalyse, wobei ausgehend von den auf ein Bauteil wirkenden Betriebslasten, unter Berücksichtigung des Materials, der Geometrie und weiterer Eigenschaften, die auftretenden Bauteilbeanspruchungen bestimmt werden. Diese Bauteilbeanspruchungen sind ursächlich für Risse im Material, die schließlich zu einem Versagen führen. Die auf ein Bauteil wirkenden Betriebslasten sind dabei stark von dem dynamischen Systemverhalten abhängig. Das dynamische Systemverhalten umfasst neben der Eigendynamik auch die gewählte Betriebsstrategie, da diese in intelligenten Systemen einen starken Zusammenhang aufweisen. Insbesondere selbstoptimierende Systeme sind in der Lage, ihr Systemverhalten autonom anzupassen, indem freie Systemparameter, wie Reglerparameter, entsprechend der Betriebsstrategie während des Betriebs neu bestimmt und verwendet werden. Folglich ändert sich das dynamische Systemverhalten in

Abhängigkeit der Betriebsstrategie. Soll beispielsweise die Taktzeit innerhalb eines Produktionssystems verkürzt werden, wird durch die Betriebsstrategie ein verändertes Bewegungsverhalten vorgegeben, sodass die Systemdynamik, im Sinne der Schnelligkeit des Bewegungsablaufs, erhöht wird. Dies führt zu steigenden Aktorkräften oder -momenten, wodurch sich schließlich die wirkenden Betriebslasten auf alle beteiligten Teilsysteme und Komponenten erhöhen. Das Ausfallverhalten technischer Bauteile, Komponenten und Teilsysteme ist typischerweise durch eine Abnahme der Lebensdauer bei steigender Belastung gekennzeichnet [Bir07, Ber08]. In Abbildung 1.1 ist dieser Zusammenhang zwischen Lebensdauer und Belastung mittels der Ausfalldichtefunktion skizziert. Die Ausfalldichtefunktion beschreibt die Wahrscheinlichkeit eines Ausfalls zu einem Zeitpunkt t . Mit erhöhter Belastung treten die Ausfälle früher während der Betriebszeit auf als bei Nennbelastung.

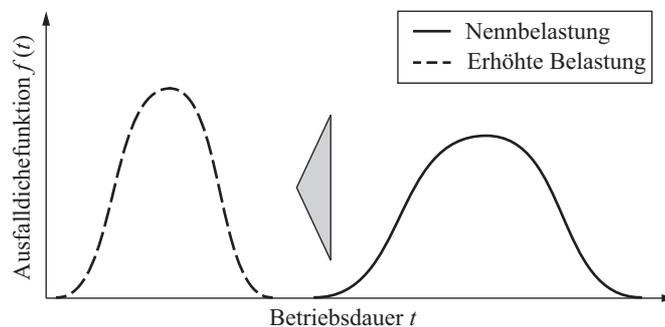


Abbildung 1.1: Typischer Verlauf einer Ausfalldichtefunktion in Abhängigkeit von der Belastung.

Die Auslegung optimaler Betriebsstrategien in intelligenten technischen Systemen erfolgt meist hinsichtlich klassischer Entwurfsziele, wie Effizienz, Komfort und Performanz, in denen die Verlässlichkeit häufig als Entwurfsziel implizit enthalten ist, aber nicht explizit in der Betriebsstrategie berücksichtigt wird. Insbesondere selbstoptimierende Systeme sind durch ihre hohe Flexibilität und Adaptionfähigkeit des Systemverhaltens geeignet, neben den verhaltensorientierten Zielen auch die Verlässlichkeit als zusätzliches Ziel direkt in den Entwurf einer intelligenten Betriebsstrategie zu integrieren. Die, aus dem Entwicklungsprozess vorhandenen, Modelle zur Abbildung des dynamischen Systemverhaltens dienen als Basis für eine Evaluation der Verlässlichkeit. Eine eng verknüpfte Modellierung der Verlässlichkeit des Systems und des dynamischen Systemverhaltens ist für den Entwurf einer umfassenden Betriebsstrategie notwendig, um das Systemverhalten im Betrieb verhaltens- und verlässlichkeitsoptimal zu gestalten. Die geschlossene Modellierung der Verlässlichkeit und des dynamischen Verhaltens setzt die Verknüpfung unterschiedlicher Modellierungsaspekte voraus, die durch erweiterte Modellierungsansätze umzusetzen sind. Eine wirksame Unterstützung des Entwicklungsprozesses erfordert die umfängliche Nutzung bereits vorhandener Modelle des Entwicklungsprozesses. Dabei werden die Modelle als zentrale Informationsquelle genutzt, um detaillierte Verlässlichkeitsanalysen mit möglichst geringen Nutzereingaben und -aufwand umzusetzen. Zudem senkt eine geschlossene Modellierung die Fehleranfälligkeit, die aus einer manuellen Synchronisation verschiedener Modelle entsteht, und trägt somit zu einer Beherrschung der zunehmenden Komplexität intelligenter technischer Systeme im Entwicklungsprozess bei.

1.1 Aufbau der Arbeit

Eine Einordnung aktueller Entwicklungen im Rahmen der digitalen Transformation und den daraus resultierenden Anforderungen an den Entwurf intelligenter technischer Systeme als Weiterentwicklung mechatronischer Systeme ist in **Abschnitt 1** gegeben. Der grundlegende Zusammenhang zwischen Verlässlichkeit und dynamischem Systemverhalten als zentraler Modellierungsaspekt dieser Arbeit wird eingeführt.

Aufbauend auf dieser Einführung wird in **Abschnitt 2** die Verlässlichkeit intelligenter technischer Systeme detailliert betrachtet. Ausgehend von einer Definition grundlegender Begriffe wird auf verschiedene Maßnahmen zur Steigerung der Verlässlichkeit eingegangen. Diese Maßnahmen werden in den Betriebsstrategien mechatronischer Systeme mithilfe von Methoden der Selbstoptimierung umgesetzt. Die Basis dieser Maßnahmen bilden jeweils detaillierte Modelle der Verlässlichkeit und des dynamischen Systemverhaltens. Nach einer Analyse bestehender Modellierungsansätze wird die Zielsetzung dieser Arbeit abgeleitet.

In **Abschnitt 3** erfolgt die Entwicklung und Implementierung einer Methode zur automatisierten Ableitung von Systemzuverlässigkeitsmodellen ausgehend von vorhandenen Modellen des dynamischen Systemverhaltens. Die möglichen Anwendungsfälle einer solchen Methode werden anhand der Lebenszyklusphasen mechatronischer Systeme beschrieben. Es wird die Auswahl eines geeigneten Systemzuverlässigkeitsmodells vorgestellt und davon ausgehend eine Methode zur automatisierten Ableitung entwickelt.

Abschnitt 4 zeigt das Potenzial der entwickelten Methode anhand von drei Anwendungsbeispielen auf:

- Es wird zuverlässigkeitsadaptive Regelung als Beispiel selbstoptimierender Systeme betrachtet. Anhand einer Trockenreibkupplung werden mittels der entwickelten Methode zuverlässigkeitsorientierte Betriebspunkte berechnet. Es wird gezeigt, dass der Entwurf verlässlichkeitssteigernder Maßnahmen in selbstoptimierenden Systemen wirkungsvoll unterstützt werden kann.
- Die Zuverlässigkeitsanalyse einer Windenergieanlage zeigt die Anwendbarkeit der Methode für komplexe Modelle des dynamischen Systemverhaltens.
- Die Umsetzung eines Digitalen Zwillings für einen Wälzlagerprüfstand zeigt das weitreichende Potenzial der Methode auf. Dabei wird das abgeleitete Systemzuverlässigkeitsmodell mit realen Zustandsüberwachungsdaten aktualisiert. Somit wird eine Bestimmung der Systemzuverlässigkeit im Betrieb und weitergehende Analysen, wie Diagnosen im Schadensfall, möglich.

In **Abschnitt 5** folgt eine Zusammenfassung der Arbeit und liefert hinsichtlich der Zielsetzung ein Fazit zu der entwickelten Methode. Die Arbeit schließt mit einem Ausblick auf zukünftige Anwendungen der Methode und möglichen weiterführenden Fragestellungen ab.

2 Verlässlichkeit intelligenter technischer Systeme

Die fortschreitende Digitalisierung ermöglicht die Umsetzung intelligenter technischer Systeme. Deren technische Realisierung stellt die Absicherung der Verlässlichkeit vor Herausforderungen, bietet jedoch auch große Potenziale zur Steigerung der Verlässlichkeit. Mit ihrer inhärenten Teilintelligenz werden selbstoptimierende Systeme betrachtet, die ihr Systemverhalten autonom an sich ändernde Umgebungs- und Betriebsbedingungen, Eigenschaften des Systems selbst sowie Nutzervorgaben anpassen. Ein inhärentes Merkmal dieser Systeme ist die Informationsverarbeitung mit entsprechenden Algorithmen zur Implementierung der Autonomie. Klassische Definition der Verlässlichkeit decken nur unzureichend alle Aspekte ab, sodass eine, aus der Informatik adaptierte, Definition verwendet wird (Abschnitt 2.1).

Selbstoptimierende Systeme bieten die Möglichkeit, durch den Entwurf verlässlichkeitsorientierter Betriebsstrategien, die Verlässlichkeit im Betrieb zu gewährleisten beziehungsweise zu steigern (Abschnitt 2.3). Dazu werden verschiedene Arbeiten vorgestellt und hinsichtlich ihrer Anforderungen aus der Umsetzung bewertet. Der Entwurf verlässlichkeitsorientierter Betriebsstrategien erfordert die Modellierung der Verlässlichkeit und die Abbildung in mathematischen Zielfunktionen für die Berechnung optimaler Betriebspunkte. Diese stellen die Basis für die Implementierung der Betriebsstrategie dar (Abschnitt 2.4 und 2.5). Die Verlässlichkeit eines technischen Systems ist in hohem Maße von deren dynamischen Verhalten geprägt, sodass die Modellierung der Verlässlichkeit auf vorhandenen Modellen der Dynamik aufgesetzt werden kann (Abschnitt 2.6 und 2.7). Eine Analyse des Stands der Forschung zeigt, dass keine geeigneten Verfahren für eine geschlossene Modellierung der Verlässlichkeit komplexer technischer Systeme und des dynamischen Systemverhaltens verfügbar sind, die eine gleichzeitige Analyse der Verlässlichkeit und der Dynamik innerhalb einer Mehrzieloptimierung ermöglichen. Es werden Anforderungen an eine umfassende Methode zur geschlossenen Modellierung der Verlässlichkeit und Dynamik abgeleitet (Abschnitt 2.8).

2.1 Grundlagen der Verlässlichkeit

Die Definition der Verlässlichkeit von Avižienis et. al. [ALR+04] ist der Informatik entliehen und im Rahmen des Sonderforschungsbereichs (SFB) 614 „Selbstoptimierende Systeme des Maschinenbaus“ auf intelligente technische Systeme adaptiert worden. Diese adaptierte Definition eignet sich insbesondere aufgrund der Berücksichtigung des hohen Software-Anteils, wie er in intelligenten technischen Systemen vorhanden ist. Die Verlässlichkeit ist demnach in drei Aspekte – Beeinträchtigungen, Kenngrößen und Mittel – für eine umfängliche Abbildung unterteilt und als „Verlässlichkeitsbaum“ in Abbildung 2.1 dargestellt. Die drei Aspekte werden im Folgenden genauer erläutert und beschrieben.

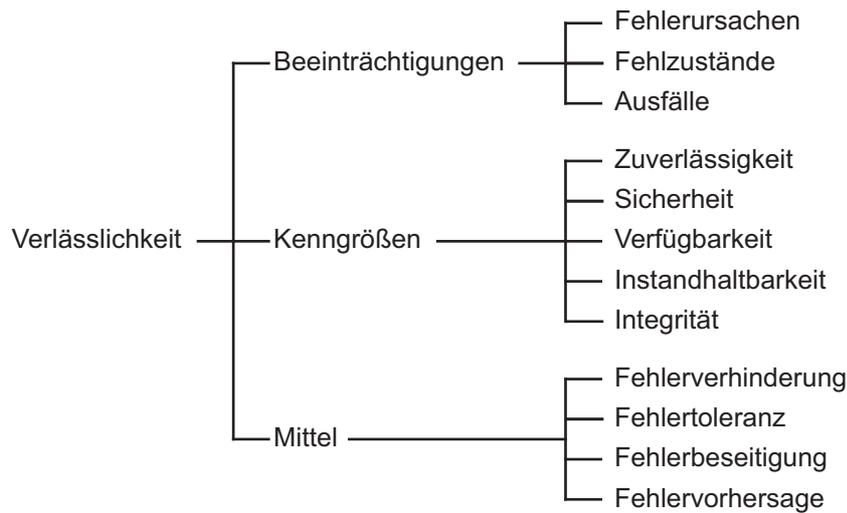


Abbildung 2.1: Verlässlichkeitsbaum nach [ALR+04].

Beeinträchtigungen

Die Beeinträchtigungen der Verlässlichkeit unterteilen sich nach Avizienis et. al. in Fehlerursachen (fault), Fehlzustände (error) und Ausfälle (failure). Diese treten immer in dieser Reihenfolge auf, wobei der Ausfall einer Komponente oder eines Teilsystems sich im System fortpflanzt und somit zu schwerwiegenden Folgen bis zum Gesamtsystemausfall führen kann (vergleiche Abbildung 2.2).



Abbildung 2.2: Beeinträchtigungen der Verlässlichkeit nach [ALR+04].

Es wird zwischen internen und externen Fehlerursachen unterschieden [Jos08]. Dabei sind interne Fehlerursache beispielsweise mechanischer Verschleiß, Degradation, Alterung etc., während bei externen Fehlerursachen ein Ereignis außerhalb der betrachteten Komponente oder des Teilsystems wie der Ausfall einer vorhergehenden Komponente, Umwelteinflüsse etc. zu berücksichtigen sind. Eine Fehlerursache aktiviert einen Fehlzustand der betrachteten Komponente oder des betrachteten Teilsystems. Ein Fehlzustand beschreibt den Degradations- oder Schädigungszustand und kann zu einem Ausfall führen. Ein Ausfall ist ein Ereignis, das beispielsweise bei der Erreichung einer vorher definierten Versagensgrenze eintritt, sodass die Komponente oder das Teilsystem seine Funktion nicht mehr bereitstellen kann. Dabei kann ein Ausfall die externe Fehlerursache für eine weitere Komponente oder ein weiteres Teilsystem darstellen (Abbildung 2.2). Die Beeinträchtigungen der Verlässlichkeit sind zentral für die Modellierung des Ausfallverhaltens, welches über die Kenngrößen hinsichtlich verschiedener Metriken quantifiziert wird.

Kenngrößen

Die Kenngrößen der Verlässlichkeit sind nach Avizienis et. al. als Zuverlässigkeit, Sicherheit, Verfügbarkeit, Instandhaltbarkeit und Integrität beschrieben. Für die Beschreibung der einzelnen Kenngrößen werden bestehende Definitionen und Maße herangezogen.

Das Ausfallverhalten technischer Systeme kann als ein stochastischer Prozess betrachtet werden. Demnach treten die Ausfälle, eine Beeinträchtigung der Verlässlichkeit, zufällig während der Betriebsdauer auf. Ausfälle, die auf einen gemeinsamen Fehlzustand in Folge einer Fehlerursache zurückgehen, werden typischerweise mittels einer Verteilungsfunktion der Wahrscheinlichkeit über die Ausfallzeitpunkte beschrieben. Für eine adäquate Beschreibung der Verteilung der Ausfallzeitpunkte muss eine Dichtefunktion $f(t)$ über die Betrachtungsdauer gefunden werden, welche die Wahrscheinlichkeit eines Ausfalls zu einem beliebigen Zeitpunkt t beschreibt. Das Integral der Dichtefunktion wird als Ausfallwahrscheinlichkeit bezeichnet und gibt an, mit welcher Wahrscheinlichkeit die Komponente, das Teil- oder Gesamtsystem zu einem beliebigen Zeitpunkt t bereits ausgefallen ist:

$$F(t) = \int_0^t f(\tau) d\tau. \quad (2.1)$$

Die Überlebenswahrscheinlichkeit oder **Zuverlässigkeit** $R(t)$ ist als Komplement zur Ausfallwahrscheinlichkeit $F(t)$ definiert:

$$R(t) = 1 - F(t). \quad (2.2)$$

Nach Bertsche ist die Zuverlässigkeit definiert als „die Wahrscheinlichkeit dafür, dass ein Produkt [Komponente, Teil- oder Gesamtsystem] während einer definierten Zeitdauer unter gegebenen Funktions- und Umgebungsbedingungen nicht ausfällt,“ [Ber08] also die ihm zugedachte Funktion erfüllt.

Die statistische Abbildung des Ausfallverhaltens bietet neben der Dichtefunktion, Ausfall- und Überlebenswahrscheinlichkeit weitere Maße, die für eine gewählte Verteilungsfunktion ineinander überführbar sind. So ist die Ausfallrate als Verhältnis der Dichtefunktion zur Zuverlässigkeit definiert:

$$\lambda(t) = \frac{f(t)}{R(t)} \quad (2.3)$$

und beschreibt die Anzahl an Komponenten oder Teilsystemen, die innerhalb eines Zeitintervalls ausfallen, im Verhältnis zu den noch funktionsfähigen Komponenten oder Teilsystemen. Das Ausfallverhalten technischer Systeme ist mithilfe der Ausfallrate in der sogenannten Badewannenkurve in Abbildung 2.3 anschaulich darstellbar. Die Badewannenkurve ist in drei Phasen unterteilt, die jeweils unterschiedliche Verläufe der Ausfallrate aufweisen und auf unterschiedliche Arten von Ausfällen hindeuten. In der ersten Phase treten Frühausfälle der betrachteten Komponente oder des Teilsystems auf, die unter anderem von Produktions-, Herstellungs- oder Montagefehlern hervorgerufen werden. Die zweite Phase ist durch Zufallsausfälle geprägt, die eine nahezu konstante Ausfallrate aufweisen. Die dritte Phase beschreibt Verschleiß- und Ermüdungsausfälle, in der die Ausfallrate stark zunimmt. In der Badewannenkurve ist der Einfluss einer erhöhten Belastung auf die Komponente oder das Teilsystem anschaulich abbildbar. Die Ausfallrate weist nach Birolini [Bir07] einen erhöhten Verlauf auf und steigt in der dritten Phase typischerweise steiler an.

Die Abbildung des Ausfallverhaltens von Zufalls- und Verschleiß- in Phase 2 sowie Ermüdungsausfällen in Phase 3 (vergleiche Abbildung 2.3) ist ein Ziel der Modellbildung dieser Arbeit, da Frühausfälle in Phase 1 durch Maßnahmen der Qualitätssicherung wirksam reduziert werden können [Ber08].

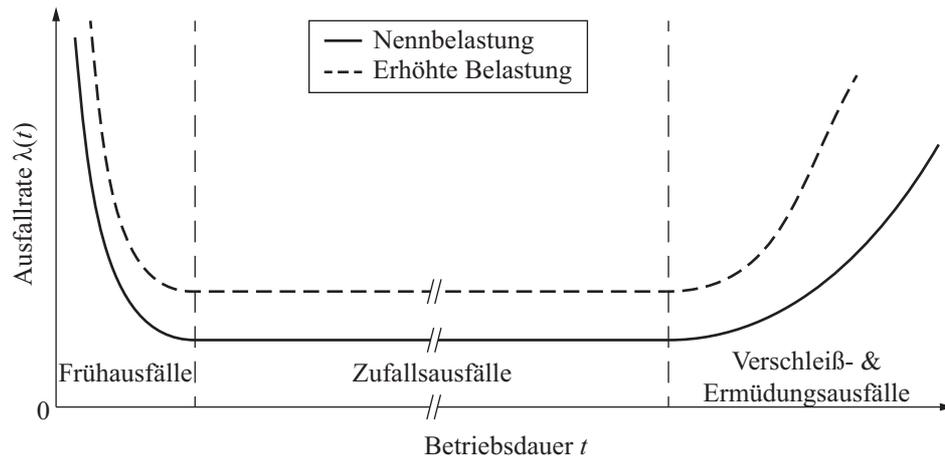


Abbildung 2.3: Schematischer Verlauf der Ausfallrate technischer Systeme als „Badewannenkurve“ nach [Bir07].

Die Überlebenswahrscheinlichkeit $R(t)$ ist in dieser Arbeit das Maß zur Beschreibung der Zuverlässigkeit als Kenngröße der Verlässlichkeit und wird synonym verwendet. Die Beschreibung des Ausfallverhaltens einzelner Komponenten und Teilsysteme erfolgt auf Basis bekannter Funktionen für Wahrscheinlichkeitsverteilungen. Im Rahmen dieser Arbeit werden vorrangig zwei bekannte Verteilungsfunktionen verwendet: die *Exponential*-Verteilung zur Abbildung des Ausfallverhaltens in der zweiten Phase und die *Weibull*-Verteilung zur Modellierung von Verschleiß- und Ermüdungsausfällen in der dritten Phase. In der Exponential-Verteilung ist die konstante Ausfallrate $\lambda = 1/MTTF$ mit $\lambda \in]0, 1[$ als Kehrwert der mittleren Betriebsdauer bis zum Ausfall (Mean Time To Failure, MTTF), unabhängig vom betrachteten Zeitpunkt t , sodass zu jedem Zeitpunkt t die gleiche Anzahl an Komponenten oder Teilsystemen ausfällt. Die Zuverlässigkeit auf Basis der Exponential-Verteilung ist gegeben durch:

$$R(t) = e^{-\lambda t}. \quad (2.4)$$

Die Weibull-Verteilung ist aufgrund der zwei Parameter, Lageparameter T und Formparameter b , im Vergleich zur Exponential-Verteilung für die Abbildung unterschiedlicher Ausfallverhalten, anwendbar. Die Erweiterung der Weibull-Verteilung, um eine ausfallfreie Zeit t_0 als dritten Parameter, wird im Rahmen dieser Arbeit nicht angewandt und daher nicht näher betrachtet. Der Lageparameter T mit $T > 0$ wird als charakteristische Lebensdauer bezeichnet und ist für $R(t=T) = 0,368$ definiert. Die charakteristische Lebensdauer T verschiebt den Verlauf der Zuverlässigkeit über die betrachteten Zeitpunkte t . Der Formparameter b mit $b > 0$ beeinflusst den Gradienten des Verlaufs der Zuverlässigkeit. Für $b=1$ entspricht die Weibull- der Exponential-Verteilung. Die Zuverlässigkeit für die Weibull-Verteilung ist in Gleichung 2.5 dargestellt.

$$R(t) = e^{-\left(\frac{t}{T}\right)^b} \quad (2.5)$$

Die Zuverlässigkeit stellt für die vorliegende Arbeit die zentrale Kenngröße der Verlässlichkeit dar. Weitere Kenngrößen, wie die Verfügbarkeit und die (funktionale) Sicherheit, sind im hohen Maße von ihr abhängig.

Die **Verfügbarkeit** ist nach Bertsche definiert als „die Wahrscheinlichkeit dafür, dass sich ein System zum Zeitpunkt t oder während einer definierten Zeitspanne in einem funktionsfähigen Zustand befindet, wenn es vorschriftsmäßig betrieben und instandgehalten wurde“ [Ber08]. Die Verfügbarkeit kombiniert daher die zwei Kenngrößen Zuverlässigkeit und Instandhaltbarkeit. Bei nicht reparierbaren Systemen findet keine Instandhaltung statt, sodass die Verfügbarkeit der Zuverlässigkeit entspricht [Ber08]. Neben der Zuverlässigkeit nimmt die Auswahl einer geeigneten Instandhaltungsstrategie maßgeblichen Einfluss auf die Verfügbarkeit bei reparierbaren Systemen. Instandhaltungsstrategien unterscheiden sich stark in ihrem methodischen Vorgehen, von rein reaktiven zu präskriptiven Maßnahmen, und den damit verbundenen Aufwand zur Implementierung und der Steigerung der Verfügbarkeit. Die Umsetzung optimaler Instandhaltungsstrategien für komplexe technische Systeme stellt hohe Anforderungen an die Planung und benötigt die Unterstützung geeigneter Methoden und Werkzeuge. In die Betrachtung der Verfügbarkeit gehen, neben der Zuverlässigkeit und der gewählten Instandhaltungsstrategie, weitere Aspekte, wie administrative Stillstandzeiten, logistische Verzögerungen, beispielsweise aufgrund von gebundenen Instandhaltungskapazitäten, und mangelhafte Ersatzteilversorgung, ein.

Die **Sicherheit** technischer Systeme beschreibt die Freiheit von unvertretbarem Risiko in Bezug auf physische Verletzung der Gesundheit von Menschen, Schädigung von Gütern oder Schädigung der Umwelt, wobei eine potentielle Schadensquelle als Gefahr bezeichnet wird [DIN820-120]. Die **funktionale Sicherheit** eines technischen Systems beschreibt die Freiheit von unvertretbarem Risiko in Bezug auf Gefahren, die aus dem funktionalen Versagen, beispielsweise durch Teil-/Ausfälle, resultieren können (nach [ISO26262-1]). Ein Risiko wird meist mittels der Auftretenswahrscheinlichkeit und der Schwere der Auswirkung eines unerwünschten Ereignisses charakterisiert. Die Auftretenswahrscheinlichkeit eines unerwünschten Ereignisses, wie dem Ausfall einer Komponente oder eines Teilsystems, ist von deren Zuverlässigkeit abhängig.

Die **Instandhaltbarkeit** bezieht sich nach VDI 4001-2 auf die Fähigkeit eines technischen Systems „unter gegebenen Anwendungsbedingungen in einem Zustand erhalten beziehungsweise in ihn zurückversetzt werden zu können, indem [es] eine geforderte Funktion erfüllen kann, wobei vorausgesetzt wird, dass die Instandhaltung [...] durchgeführt wird“ [Ver06].

Die **Integrität** beschreibt nach Avizienis et. al. [ALR+04] die Abwesenheit von unzulässigen Veränderungen des Systemzustands. Diese beziehen sich auf die unautorisierte Manipulation oder Löschung von Daten innerhalb des Systems. Dies schließt innerhalb des Systems aufgenommene, gespeicherte und übertragene Daten ein. Die Integrität schließt den Bereich der Sicherheit von Software sowie informationstechnischen Systemen ein.

Mittel

Nach Avizienis et. al. [ALR+04] sind die Mittel oder Maßnahmen zur Steigerung der Verlässlichkeit in Fehlerverhinderung, Fehlertoleranz, Fehlerbeseitigung und Fehler vorhersage unterteilt.

Die **Fehlerverhinderung** ist inhärenter Bestandteil einer jeden Entwicklungsmethodik technischer Systeme. Eines der Hauptziele ist es, mögliche Fehler bereits während des Entwicklungsprozesses zu identifizieren und durch geeignete Maßnahmen zu vermeiden (vergleiche beispielsweise VDI 2206 Entwicklungsmethodik für mechatronische Systeme [VDI04] oder Entwicklungsprozess nach Pahl / Beitz [PBS+13]).

Die **Fehlertoleranz** umfasst nach Avižienis et. al. [ALR+04] Methoden zur Fehlererkennung und -detektion sowie Verfahren für die Wiederherstellung und Aufrechterhaltung zulässiger Systemzustände. Die Verfahren der Wiederherstellung und Aufrechterhaltung beinhalten unter anderem Methoden der Fehlerdiagnose sowie Kompensation von Ausfällen durch Redundanzen im System und Rekonfigurationspotenzialen, um sichere Systemzustände zu erreichen.

Die **Fehlerbeseitigung** im Entwicklungsprozess ist durch die drei Schritte Verifizierung und Validierung, Diagnose sowie Korrektur beschrieben [ALR+04]. Auf Basis einer Verifizierung der Systemspezifikationen und einer Validierung des Systemverhaltens erfolgt die Diagnose der aufgetretenen Fehler, die abschließend korrigiert werden müssen. Die Fehlerbeseitigung während des Betriebs ist Aufgabe der Instandhaltung und wird über die Instandhaltungsstrategie spezifiziert.

Die **Fehlervorhersage** innerhalb der Definition nach Avižienis et. al. beschreibt die Prognose des Systemverhaltens für einen aufgetretenen Fehlerfall. Eine Prädiktion zukünftiger Ausfälle im Rahmen einer zustandsbasierten Instandhaltungsstrategie ist hier zunächst nicht berücksichtigt [ALR+04, SW15]. Im Rahmen der Anwendung der Definition der Verlässlichkeit auf mechatronische Systeme ist eine Einbeziehung von Methoden zur Prognose zukünftiger Ausfälle für die Definition einer optimalen Instandhaltungsstrategie naheliegend.

2.2 Ausfallverhalten technischer Systeme

Die Beeinträchtigungen der Verlässlichkeit, wie sie in Abschnitt 2.1 definiert wurden, liefern die grundlegende Terminologie für die Beschreibung des Ausfallverhaltens technischer Systeme. Das Ausfallverhalten beschreibt die Abhängigkeiten zwischen Fehlerursachen, Fehlzuständen sowie Ausfällen und ist unter anderem von der Architektur des Systems, den individuellen Eigenschaften der im System enthaltenen Komponenten und Teilsystemen sowie von auftretenden externen Ereignissen beeinflusst.

Aus den Mitteln zur Steigerung der Verlässlichkeit sind verschiedene Methoden bekannt, deren Einsatz vornehmlich während des Entwicklungsprozesses durchzuführen ist. Die Einbringung von Redundanz, das mehrfache Vorhandensein von Komponenten oder Teilsystemen als für die reine Funktionserfüllung eines technischen Systems notwendig ist [Bir07], ist die einzige Möglichkeit, die Zuverlässigkeit auf struktureller Seite zu erhöhen. Darüber hinaus ist die Verwendung von Komponenten oder Teilsystemen mit höherer Zuverlässigkeit, beispielsweise aufgrund verbesserter Fertigungstoleranzen oder Materialauswahl, meist nur in engen Grenzen möglich. Redundant ausgelegte Systeme sind jedoch anfällig gegenüber systematischen und abhängigen Ausfällen. Der Grund dafür liegt in der Universalität des Phänomens. Systematische Ausfälle sind hierbei beispielsweise durch Fertigungsfehler hervorgerufen. Diese Ausfälle werden typischerweise durch die Verwendung von Komponenten oder Teilsystemen mit gleicher Funktion von unterschiedlichen Herstellern vermieden. Abhängige Ausfälle treten bei einer Verkettung oder Kaskadierung von mehreren Ausfällen auf [Int11], wie in Abbildung 2.2 gezeigt. Des Weiteren zählen Ausfälle gemeinsamer Ursache dazu. Eine gemeinsame Fehlerursache, wie ein katastrophales externes Ereignis oder der Ausfall eines zentralen Bestandteils des Systems, die zum Ausfall aller Komponenten und Teilsystemen innerhalb einer Redundanz führt, ist eine große Bedrohung der Verlässlichkeit [Sag04, Bir07]. Die hohe Funktions-

integration in mechatronischen Systemen trägt dazu bei, dass aktivierte Fehlerursachen zu Fehlzuständen und schließlich zu Ausfällen von verschiedenen Komponenten und Teilsystemen führen können. Unabhängige Fehlerursachen sind dem gegenüber nicht durch das Auftreten weiterer Fehlerursachen beeinflusst und werden unabhängig aktiviert.

Ausfälle gemeinsamer Fehlerursache

Ausfälle gemeinsamer Fehlerursache sind nach DIN EN ISO 12100 [Deu11] definiert als „Ausfälle verschiedener Einheiten aufgrund eines einzelnen Ereignisses, wobei sich diese Ausfälle nicht gegenseitig beeinflussen“. Nach Jones [Jon12] treten Ausfälle gemeinsamer Ursache nur in redundant ausgelegten Systemen auf, da nur hier mehrere Ausfälle im redundanten System tatsächlich zu einem vollständigen Systemausfall führen. Jones schließt kaskadierte Ausfälle, aufgrund einer gemeinsamen Fehlerursache in redundant ausgelegten Systemen, in die Definition von Ausfällen gemeinsamer Fehlerursache mit ein [Jon12]. Ausfälle gemeinsamer Fehlerursache werden durch interne oder externe Ereignisse ausgelöst. Ein Beispiel für eine externe Fehlerursache ist die Havarie des japanischen Atomkraftwerks Fukushima Daiichi im Jahr 2011. Die gemeinsame Fehlerursache für die Havarie war ein Tsunami, der zu einem kompletten Ausfall des redundant ausgelegten Kühlsystems des Atomkraftwerks führte. Dabei brachen zunächst die externe Stromversorgung und anschließend die Notstromversorgung, bestehend aus Dieselgeneratoren und Batterieeinheiten, zusammen [Jon12].

Redundanz in technischen Systemen

Die Integration von redundanten Komponenten oder Teilsystemen stellt einen wirksamen Ansatz zur Steigerung der Verlässlichkeit dar, wenn die Risiken richtig berücksichtigt werden. Dazu gehören die bereits erwähnten Ausfälle aufgrund gemeinsamer Fehlerursachen, wobei externe Ereignisse mit katastrophalen Auswirkungen nur schwer zu begegnen ist. Die Umsetzung von Redundanzen erhöht die Komplexität des Systems, macht es dadurch anfälliger für Fehler und führt bei nicht sorgfältiger Umsetzung zu einer Reduzierung der Verlässlichkeit [Sag04]. Eine Variante zur Umsetzung von Redundanz ist die Verwendung von gleichartigen Komponenten oder Teilsystemen mit der gleichen Funktionalität. Die Auswahl von baugleichen Einheiten erhöht die Anfälligkeit für Ausfälle gemeinsamer Fehlerursachen, wie Herstellungsfehler oder Unzulänglichkeiten in der Software. In der Literatur sind verschiedene Arten von Redundanzen in technischen Systemen bekannt, so können funktionale Kompensationen durchgeführt werden, indem beispielsweise der Ausfall eines Sensors durch den Einsatz eines regelungstechnischen Beobachters kompensiert werden kann [Ise02]. Diese Redundanzformen verwenden Mehrheitsentscheidungen und Voting-Techniken zur Bestimmung eines Ausfalls und der Auswahl der redundanten Komponenten oder Teilsysteme zur Kompensation [Bir07].

Die im Rahmen dieser Arbeit betrachteten Redundanzkonzepte sind auf Strukturen mit der Redundanzordnung k -aus- m beschränkt. Ein so ausgelegtes System besteht aus m funktionalidentischen Komponenten oder Teilsystemen, von denen mindestens die Anzahl k für die Aufrechterhaltung der Funktionalität notwendig sind. Demnach besteht ein solches redundantes System aus $m - k$ überzähligen Komponenten als für die Erfüllung der Funktionalität erforderlich sind. Diese überzähligen Komponenten stehen für die Kompensation von Ausfällen zur Verfügung.

Neben der Redundanzordnung wird die Redundanzart zur Beschreibung herangezogen. Die Redundanzart charakterisiert das Verhalten des redundanten Systems und wird in *heiß*, *warm* und *kalt*, jeweils *mit* oder *ohne Lastteilung*, unter den beteiligten m Komponenten oder Teilsystemen unterteilt. In den Redundanzarten wird zwischen *aktiven* und *passiven* Komponenten oder Teilsystemen unterschieden. Die aktiven Komponenten und Teilsysteme des redundanten Systems werden entsprechend der Redundanzart belastet und sind einer schnelleren Lebensdauerabnahme unterworfen. Passive Komponenten und Teilsysteme tragen nicht aktiv zur Erfüllung der Funktionalität bei und weisen daher keine oder eine stark verzögerte Lebensdauerabnahme, beispielsweise aufgrund von Alterungsprozessen, auf. Eine **Lastteilung** tritt auf, wenn die wirkende Belastung aufgeteilt und von mehreren Elementen aufgenommen wird. Es folgt ein belastungsabhängiges Ausfallverhalten, das zu einer steigenden Ausfallrate bei zunehmender Anzahl an Ausfällen für die aktiven Komponenten und Teilsysteme führt. In redundanten Systemen mit Lastteilung führt ein Ausfall demnach zu einer Zunahme der individuellen Last, während die individuellen Lasten in einem System ohne Lastteilung invariant gegenüber Ausfällen sind.

In **heiß** ausgelegten redundanten Systemen sind nur aktive Komponenten und Teilsysteme vorhanden, zwischen denen eine Lastteilung möglich ist. Eine Ungleichverteilung der wirkenden Lasten auf die aktiven Komponenten und Teilsysteme ist möglich. Eine heiße Redundanz ohne Lastteilung liegt beispielsweise bei Warnbarken im Straßenverkehr vor, wobei alle Elemente der gleichen Belastung durch Umwelteinflüsse ausgesetzt sind, der Ausfall einer Barke aber keinen Einfluss auf die Lebensdauer der verbleibenden besitzt. Eine Lastteilung liegt beispielsweise bei Bremsfallschirmen vor, die ein Raumschiff nach dem Wiedereintritt bis zum Aufsetzen auf der Erde abbremsen. Typischerweise finden hier drei Fallschirme Anwendung, wobei nur zwei für eine wirksame Verzögerung benötigt werden. Die Lasten durch das Gewicht und die Fallgeschwindigkeit wird unter den drei Fallschirmen aufgeteilt, sodass die individuelle Last auf die verbliebenen Elemente im Falle eines Ausfalls ansteigt.

Die **warme** Redundanz stellt eine Verallgemeinerung der heißen Redundanzart dar, indem sie zusätzlich zu den genannten Aspekten passive Elemente berücksichtigt. Die passiven Elemente weisen ein Ausfallverhalten auf und bilden beispielweise den Ersatzreifen eines Automobils ab. Neben den aktiven Reifen, die Verschleiß unterworfen sind, ist der Ersatzreifen nicht beliebig lange, aufgrund von Alterungsvorgängen, verwendbar. Die Lebensdauer der passiven Elemente liegt dennoch deutlich über der Lebensdauer der aktiven Elemente.

Die **kalte** Redundanz stellt eine idealisierte Abbildung der warmen Redundanzart dar, da die passiven Komponenten und Teilsysteme keiner Lebensdauerabnahme unterliegen. Eine Reduzierung der Lebensdauer dieser Elemente beginnt erst mit der Aktivierung zur Kompensation eines Ausfalls. Redundante Bauteile in elektronischen Schaltungen werden oft mittels kalter Redundanz modelliert, da die Lebensdauerabnahme aufgrund von Alterungsvorgängen meist zu vernachlässigen ist.

2.3 Selbstoptimierende technische Systeme

Selbstoptimierende technische Systeme sind im Rahmen des SFB 614 als intelligente technische Systeme charakterisiert worden, die ihr Verhalten autonom und flexibel auf Nut-

zeranforderungen, Änderungen der Umgebungsbedingungen oder des Systems selbst anpassen können [ADG+09]. Die daraus resultierende zielkonforme Anpassung von Systemparametern oder der Struktur (Rekonfiguration) wird als Selbstoptimierung verstanden [ADG+09]. Das Systemverhalten wird dabei über die gewählten Ziele definiert, wobei die Priorisierung der Ziele die Verhaltensanpassung implementiert. Selbstoptimierung gestaltet sich demnach als kontinuierlicher Prozess des Übergangs von einem Ausgangszustand des Systems hin zu einem zielkonformen Zustand, wie er in Abbildung 2.4 dargestellt ist.

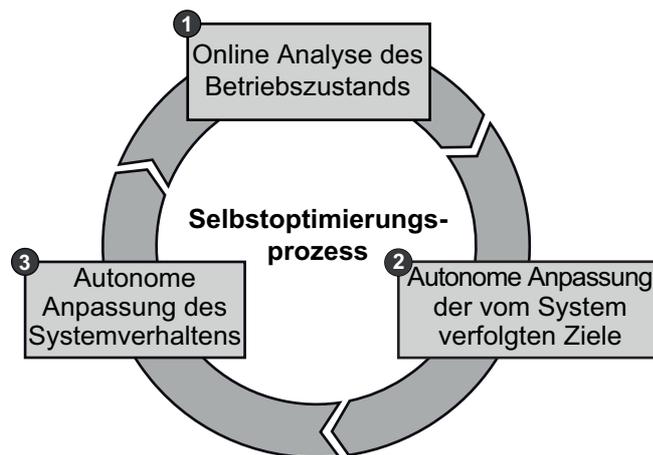


Abbildung 2.4: Ablauf der Verhaltensadaption selbstoptimierender technischer Systeme [GRS14].

Der Selbstoptimierungsprozess wird während des Betriebs kontinuierlich durchlaufen und besteht aus drei Schritten (nach [GRS14]):

Online Analyse des Betriebszustands: In diesem Schritt werden die Nutzeranforderungen, der Zustand des Systems selbst und die Umgebungsbedingungen aufgenommen und analysiert. Diese Überwachung wird durch Kommunikation mit anderen Systemen und erweiternden Methoden wie einer Zustandsüberwachung unterstützt. Ein weiterer Aspekt ist die Evaluierung des Systemverhaltens hinsichtlich der Ziele.

Autonome Anpassung der vom System verfolgten Ziele: Die vom System verfolgten Ziele werden in diesem Schritt gewählt, angepasst oder neu generiert, um auf Änderungen der aktuellen Situation zu reagieren. Eine Auswahl ist für eine endliche Anzahl diskreter Ziele möglich, während eine Anpassung für sich graduell ändernde Ziele durchgeführt werden kann. Die Erzeugung und Formulierung neuer Ziele ist durchführbar, wenn diese unabhängig von bereits existierenden Zielen sind.

Autonome Anpassung des Systemverhaltens: Das Systemverhalten wird auf Basis der Anpassung der Priorisierung der verfolgten Ziele vorgenommen. Dies umfasst die Änderung von Systemparametern wie Reglerparametern und Änderungen an der Struktur des Systems, beispielsweise ermöglicht durch Rekonfigurationspotenziale. Die Anpassung des Systemverhaltens stellt die Rückkopplung des Selbstoptimierungsprozesses auf das eigentliche mechatronische System dar.

Die zyklische Durchführung dieses Prozesses charakterisiert und implementiert die inhärente Teilintelligenz in selbstoptimierenden Systemen. Eine kontinuierliche Anpassung, Auswahl und Einstellung optimaler Betriebspunkte ermöglicht eine zielgerichtete Anpassung des Systemverhaltens auch hinsichtlich konkurrierender Ziele.

2.4 Mehrzieloptimierung

Freie Systemparameter eines mechatronischen Systems werden innerhalb der Selbstoptimierung genutzt, um optimale Betriebspunkte zu finden. Die Ziele eines mechatronischen Systems sind häufig konkurrierend, die Verbesserung eines Ziels führt zu einer Verschlechterung der übrigen Ziele. Für die autonome Anpassung der Ziele in Schritt 2 des Selbstoptimierungsprozesses stellt die Mehrzieloptimierung ein wichtiges Hilfsmittel dar [ADG+09, GRS14]. Ein Mehrzieloptimierungsproblem wird als Minimierungsproblem über Zielfunktionen unter der Berücksichtigung von Nebenbedingungen definiert [GRS14]:

$$\min_{\mathbf{p} \in S} \mathbf{F}(\mathbf{p}), \quad S = \{\mathbf{p} \in \mathbb{R}^n \mid \mathbf{g}(\mathbf{p}) = 0, \mathbf{a}(\mathbf{p}) \leq 0\}. \quad (2.6)$$

Dabei ist $\mathbf{F}(\mathbf{p}) = (f_1(\mathbf{p}), \dots, f_k(\mathbf{p}))$ der Vektor der Zielfunktionen und \mathbf{p} der Vektor der Entwurfsvariablen. Diese freien Variablen werden auch als Optimierungsparameter bezeichnet. Diese Entwurfsvariablen stellen freie Systemparameter dar, mit denen das Systemverhalten hinsichtlich der Ziele zu beeinflussen ist. Der Entwurfsraum S der Optimierungsparameter wird durch die Nebenbedingungen mittels Ungleichungen $\mathbf{a}(\mathbf{p})$ und Gleichung $\mathbf{g}(\mathbf{p})$ begrenzt. Die Zielfunktionen stellen mathematische Beschreibungen der vom System zu verfolgenden Ziele dar und bewerten das Systemverhalten für einen gegebenen Satz von Optimierungsparametern hinsichtlich dieser Ziele. Die Evaluation der Zielfunktionen erfordert typischerweise im Kontext mechatronischer Systeme die Simulation des dynamischen Systemverhaltens für ein *charakteristisches Manöver*. Das charakteristische Manöver repräsentiert dabei typische oder besonders kritische Betriebsmanöver für die das Systemverhalten abgesichert werden muss. In Abhängigkeit der Modellierungstiefe des dynamischen Systemverhaltens können damit bereits einzelne Evaluationen der Zielfunktionen rechenintensiv sein [ADG+09].

In einem Mehrzieloptimierungsproblem existiert typischerweise kein globales Optimum, sondern eine Menge optimaler Kompromisse zwischen den im Konflikt stehenden Zielen, die sogenannte *Pareto-Menge*. Ein Punkt $\hat{\mathbf{p}} \in S$ wird *pareto-optimal* genannt, wenn kein anderer Punkt $\mathbf{p} \in S$ existiert, sodass $\mathbf{F}(\mathbf{p}) \leq \mathbf{F}(\hat{\mathbf{p}})$ [Ehr05]. Die der Pareto-Menge zugehörige Menge an Zielfunktionswerten $\mathbf{F}(\hat{\mathbf{p}})$ wird *Pareto-Front* genannt.

In Abbildung 2.5 ist das Ergebnis eines einfachen Mehrzieloptimierungsproblems mit jeweils zwei Zielfunktionen und Optimierungsparametern dargestellt. Die Pareto-Menge (links) und die Pareto-Front (rechts) sind zusammen mit innerhalb der Optimierungssiterationen verwendeten Optimierungsparametern und den dazugehörigen möglichen Betriebspunkten abgebildet. Die gefundenen pareto-optimalen Betriebspunkte sind jeweils hervorgehoben.

Die Mehrzieloptimierung mechatronischer Systeme stellt aufgrund der hohen Komplexität der Systeme eine Herausforderung für die zur Verfügung stehenden Methoden dar. Diese äußert sich in der Regel darin, dass der Gradient der Zielfunktionen über den gesamten Parameterraum S unbekannt ist und die Pareto-Menge in Folge dessen numerisch angenähert werden muss. Gängige Verfahren wurden in [ADG+09] gesammelt und vorgestellt.

Die Bestimmung der Pareto-Menge eines mechatronischen Systems ermöglicht die Umsetzung einer hohen Flexibilität und Adaptivität durch die Anpassung des Systemverhaltens in Schritt 3 des Selbstoptimierungsprozesses.

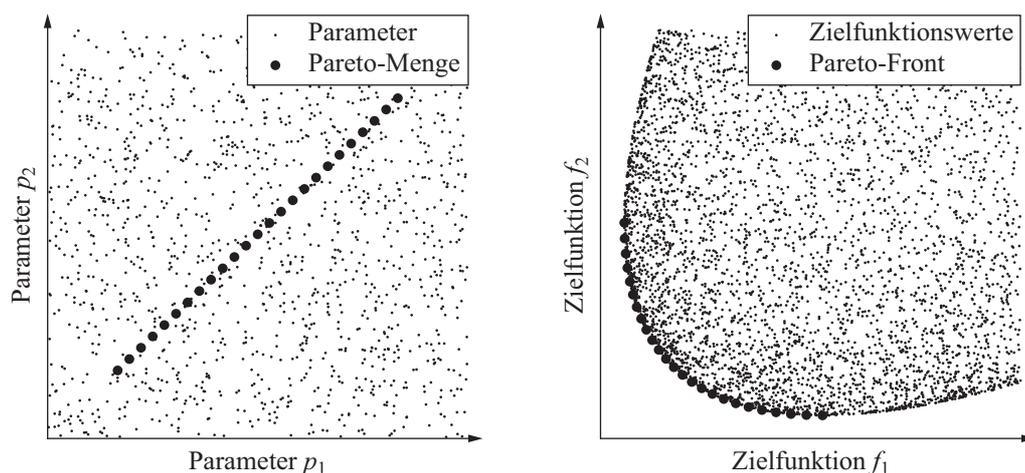


Abbildung 2.5: Beispiel von Optimierungsparametern und Zielfunktionswerten für mögliche und pareto-optimale Betriebspunkte (nach [Mey16]).

2.5 Entwicklung intelligenter, verlässlicher Systeme

Die Absicherung der Verlässlichkeit von intelligenten technischen und auch von selbst-optimierenden Systemen stellt aufgrund der autonomen Verhaltensanpassung eine Herausforderung dar. Diese führt zu einer Vielzahl von teils nicht vorhersehbaren Betriebssituationen, in denen ein zuverlässiger und sicherer Betrieb gewährleistet sein muss. Eine Entwicklungsmethodik für den Entwurf selbstoptimierender Systeme systematisiert den Entwicklungsprozess und stellt verschiedene Hilfsmittel zur Verfügung [GRS+14]. Die steigende Komplexität technischer Systeme bedeutet in der Regel ein erhöhtes Risiko von systematischen Fehlern und damit einhergehend eine allgemein erhöhte Fehleranfälligkeit. Selbstoptimierende Systeme weisen durch ihre erweiterte Funktionalität und erhöhte Variabilität in ihrem Verhalten Potenziale zur Steigerung der Verlässlichkeit durch Robustheit gegenüber beziehungsweise Kompensation von unerwünschten Ereignissen, wie Komponenten- oder Teilsystemausfällen, auf. Im Entwurf dieser Systeme finden demnach verschiedene Methoden Anwendung, um auf Basis einer autonomen Verhaltensadaption eine Steigerung der Verlässlichkeit zu gewährleisten [DHK+09, GRS14, SMK+17].

Das mehrstufige Verlässlichkeitskonzept wurde als Methode zur Absicherung der Sicherheit eines selbstoptimierenden Systems entwickelt [FGM+07]. Das Konzept kombiniert Verfahren zur Erkennung und Klassifizierung möglicher Fehlzustände des Systems und leitet situationsgerechte Gegenmaßnahmen ein. Diese Maßnahmen sind in den Stufen des Konzepts hinterlegt, werden in Abhängigkeit des Fehlzustands gewählt und beinhalten eine Änderung der Priorisierung der Ziele auf Basis der Selbstoptimierung oder leiten direkt in sichere Systemzustände (Safe-States) über. Dieses mehrstufige Verlässlichkeitskonzept wurde auf weitere Kenngrößen der Verlässlichkeit erweitert (vergleiche Abbildung 2.6), sodass neben der Sicherheit auch die Zuverlässigkeit und die Verfügbarkeit gesteigert werden [DHK+09]. Das Konzept ermöglicht schließlich, durch die Integration einer erweiterten zuverlässigkeitsbasierten Überwachung [SW15], die systematische Ausnutzung von Rekonfigurationspotenzialen in mechatronischen Systemen unter Verwendung der Selbstoptimierung. In *Stufe I* befindet sich das System in einem sicheren, regulären Zustand, in dem typischerweise effizienz- und komfortorientierte Ziele gegenüber verlässlichkeitsorientierten Zielen priorisiert sind. In *Stufe II* hat das System vorab definierte verlässlich-

keitsorientierte Schwellwerte (Ausfallrate, Restlebensdauer u. w.) überschritten, sodass eine Priorisierung der verlässlichkeitsorientierten Ziele erfolgt, um ein Erreichen der Ziele sicherzustellen. *Stufe III* stellt einen kritischen Zustand des Systems aufgrund von schwerwiegenden Fehlern dar. Eine exakte Regelung des Systems ist, beispielsweise aufgrund von Sensorausfällen, nicht mehr möglich, sodass erste Notfallroutinen ausgeführt werden. Diese Routinen können vorhandene Rekonfigurationspotenziale des Systems nutzen, um einen sicheren Betrieb bei reduzierter Funktionalität oder Leistungsfähigkeit zu gewährleisten. In *Stufe IV* wird das System umgehend in einen sicheren Zustand versetzt, um unkontrolliertes Verhalten und damit einhergehende schwerwiegende Schäden am System oder Menschen auszuschließen [DHK+09, SW15].

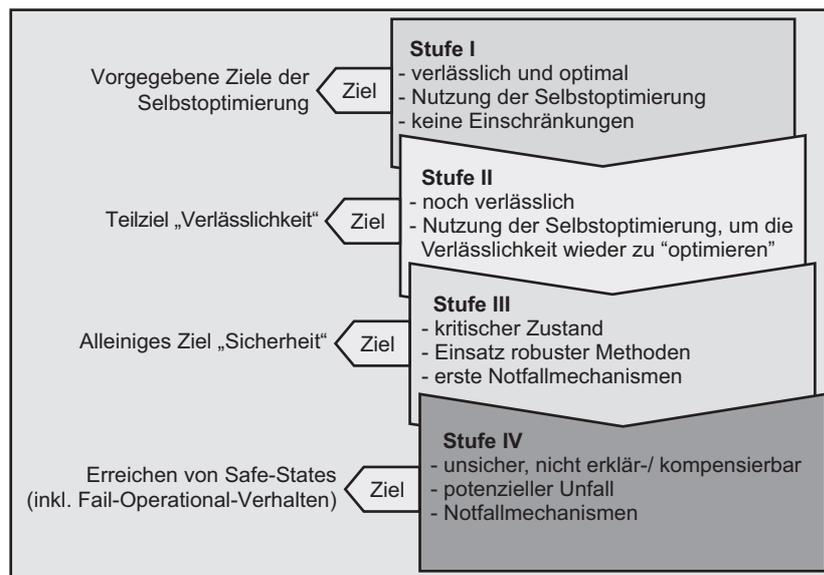


Abbildung 2.6: Mehrstufiges Verlässlichkeitskonzept zur Reaktion auf Komponentenausfälle (nach [DHK+09, SW15]).

Das mehrstufige Verlässlichkeitskonzept stellt eine autonome Umschaltung zwischen diskreten, vorab definierten Systemzuständen, dar. In den Stufen I und II findet eine harte Umschaltung zwischen den Priorisierungen der Ziele statt. Dieses Konzept wurde zu einer kontinuierlichen zuverlässigkeitsadaptiven Regelung [Mey16] weiterentwickelt. Das Ziel dieser Regelung ist eine kontinuierliche Anpassung der Priorisierung pareto-optimaler Betriebspunkte, um verlässlichkeitsorientierte Ziele, wie eine vorgegebene Lebensdauer oder ein Instandhaltungsintervall, auch unter unstetiger Degradation zu erreichen. Die Struktur der Regelung ist in Abbildung 2.7 gezeigt und basiert auf einer Verhaltensanpassung mittels einer Regelung zur Einstellung der pareto-optimalen Betriebspunkte (vergleiche Pareto-Front in Abbildung 2.5) [KRK+13]. Diese Regelung verwendet die α -Parametrisierung, die jedem Betriebspunkt auf der Pareto-Front einen eindeutigen α -Wert zuweist, der als Regelgröße dient.

Innerhalb der Verhaltensregelung ist der Sollwert des Verhaltensparameters α_{soll} definiert über die Summation des Ausgangs des Zuverlässigkeitsreglers α_{R} und der externen Nutzervorgabe α_{Nutzer} , um die berechnete Reglergröße zu überschreiben, sowie über die Rückkoppelung α_{ist} , als Istwert des dynamischen Systems. Die Zuverlässigkeitsregelung bestimmt den Ausgang α_{R} auf Basis einer Sollwertvorgabe des Health Index HI_{soll} und des geschätzten Health Index \widehat{HI} . Dieser wird mittels einer Zustandsüberwachung ausge-

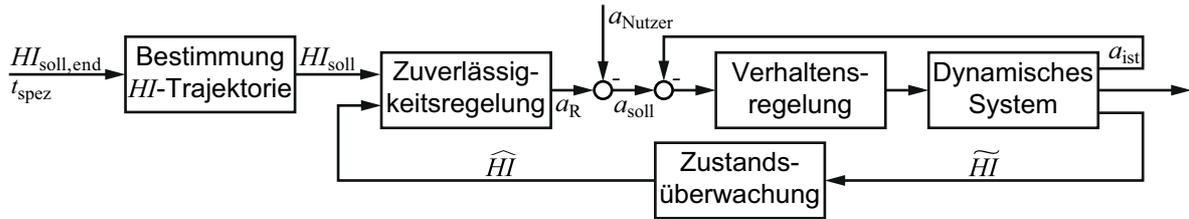


Abbildung 2.7: Struktur der zuverlässigkeitsadaptiven Regelung (nach [Mey16]).

hend von verrauschten Messsignalen des dynamischen Systems \widetilde{HI} bestimmt. Der Sollwert des Health Index für die Zuverlässigkeitsregelung HI_{soll} wird auf Basis des Zielwerts für den Health Index $HI_{soll,end}$ für eine spezifizierte Betriebsdauer t_{spez} berechnet.

Sowohl das mehrstufige Verlässlichkeitskonzept als auch dessen Erweiterung um eine zuverlässigkeitsadaptive Regelung sind durch eine leistungsfähige Zustandsüberwachung als inhärenter Bestandteil charakterisiert. Für eine Zustandsüberwachung technischer Komponenten und Systeme werden drei Ansätze betrachtet [KHV09, Kim16]. Eine einfache Überwachung des Ausfallverhaltens technischer Systeme steht auf Basis der Zuverlässigkeit der betrachteten Komponenten oder Teilsysteme zur Verfügung [KHV09]. Die *zuverlässigkeitsbasierte Überwachung* beschreibt die Degradation einer Komponente oder eines Teilsystems statistisch über eine Gesamtheit gleichartiger Systeme. Der aktuelle Grad der Degradation und die verbleibende Lebensdauer ist somit statistisch über die Ausfallzeiten gleichartiger Systeme beschrieben. Ihre Genauigkeit ist jedoch aufgrund der hohen Streuung der Lebensdauern durch variierende Betriebsbedingungen und Fertigungstoleranzen eingeschränkt. Der aktuelle Schädigungszustand und die verbleibende Lebensdauer individueller Komponenten oder Teilsysteme sind mittels *modell- und datenbasierter Diagnose* und *Prognose* innerhalb einer Zustandsüberwachung detaillierter bestimmbar [KHV09, Kim16]. In modellbasierten Methoden werden physikalische oder empirische Fehler- beziehungsweise Ausfallmodelle verwendet. Diese weisen eine hohe Genauigkeit auf, nehmen jedoch schnell an Komplexität zu und sind somit oft rechenintensiv. Die Fehlermodelle sind typischerweise anwendungsspezifisch und dadurch nur aufwendig an neue Anwendungen anpassbar. Datenbasierte Methoden basieren auf Sensor- oder Betriebsdaten des zu überwachenden Systems und verwenden maschinelle Lernverfahren, um das Degradationsverhalten aus den zugrundeliegenden Daten zu lernen. Diese Methoden sind flexibel auf unterschiedliche Systeme anwendbar und zeigen eine steigende Verbreitung innerhalb industrieller Anwendungen. Eine Übersicht über die verschiedenen Ansätze zur Zustandsüberwachung technischer Systeme ist unter anderem vom Kimotho erarbeitet worden [Kim16].

Das Paradigma der Selbstoptimierung ist nur dann zielführend und umfassend für die Steigerung der Verlässlichkeit mechatronischer Systeme anzuwenden, wenn geeignete mathematische Formulierungen verlässlichkeitsorientierter Ziele verfügbar sind. Diese sind in die Mehrzieloptimierung als Basis für den Entwurf selbstoptimierender Systeme einzubringen, um Betriebspunkte für verlässlichkeitsorientierte Ziele zu bestimmen. Eine wesentliche Grundlage für die Beherrschung der Komplexität dieser Systeme ist die umfassende Nutzung von Modellen im Entwicklungsprozess im Sinne des Modelbased Systems Engineering (MBSE) [GRS+14, WRF15]. Die Entwicklung muss durch leistungsfähige Werkzeuge zur Modellbildung und -analyse unterstützt werden. Einen zentralen Beitrag leisten hier domänenübergreifende Modelle zur ganzheitlichen Abbildung des

dynamischen Systemverhaltens (vergleiche Abschnitt 2.6). Die Abbildung weiterer Modellierungsaspekte wie der Verlässlichkeit ist nur innerhalb einer geschlossenen Modellierung (vergleiche Abschnitt 2.7), aufbauend auf der ganzheitlichen Modellierung des Systemverhaltens, für die Berücksichtigung verlässlichkeitsorientierter Ziele im Entwurf selbstopmierender Systeme zielführend umsetzbar.

2.6 Ganzheitliche Modellierung mechatronischer Systeme

Die ganzheitliche Modellierung beschreibt den domänenübergreifenden Modellierungsansatz für die Abbildung des Verhaltens mechatronischer Systeme. In der Regel liegt der Fokus auf der Abbildung des dynamischen Gesamtsystemverhaltens unter Einbeziehung aller teilhabenden Domänen wie Mechanik, Elektrotechnik und Informatik. Aus der ganzheitlichen Abbildung des dynamischen Verhaltens eines mechatronischen Systems werden mathematische Abbildungen beliebiger Kenngrößen des Systems wie Performanz, Effizienz, Komfort oder Zuverlässigkeit abgeleitet. Die Abbildung dieser Kenngrößen ist die Basis für die Formulierung von Zielfunktionen für eine Optimierung des Systemverhaltens hinsichtlichlicher verschiedener Entwicklungsziele.

Im Folgenden wird eine Auswahl verschiedener Software-Werkzeuge für die ganzheitliche Abbildung mechatronischer Systeme vorgestellt. Zunächst werden Werkzeuge zur Unterstützung der Spezifikation während des Systementwurfs (vergleiche V-Modell in Abbildung 3.3) vorgestellt. Diese bestehen zumeist aus Partialmodellen zur Beschreibung verschiedener Aspekte des Systemverhaltens. Domänenübergreifende Modellierungs- und Analysewerkzeuge bilden das dynamische Systemverhalten detailliert ab und dienen der Eigenschaftsabsicherung, insbesondere während des Entwurfs und der Systemintegration. Abschließend werden Softwarewerkzeuge zur Verknüpfung verschiedener spezialisierter Modelle vorgestellt, die eine Integration verschiedener Modellierungs- und Analysewerkzeuge in eine gemeinsame Entwurfsumgebung darstellen. Einen Überblick über Softwarewerkzeuge zur ganzheitlichen Modellierung mechatronischer Systeme liefert Gausemeier in [GFD+09].

SysML (Systems Modeling Language) ist eine Weiterentwicklung der UML (Unified Modeling Language) hinsichtlich einer standardisierten Programmiersprache des Systems Engineering (SE) zur Modellierung, Analyse und Verifikation von beliebigen, komplexen Systemen [Obj19]. Die UML entstammt der Softwaretechnik und verwendet Zustandsautomaten, um das logische Verhalten eines Systems über Zustände und Transition abzubilden. Die SysML erweitert den Ansatz von UML um zusätzliche Partialmodelle, sogenannte Diagramme, zur vollständigen Modellierung von Anforderungen bis zum konzeptuellen Design. Sie eignet sich daher zur domänenübergreifenden Spezifizierung von komplexen Systemen und deren Komponenten. Die SysML beschreibt die verschiedenen Aspekte eines Systems, wie Struktur, Interaktion der Komponenten, funktions-, signal- und zustandsbasiertes Systemverhalten, Einschränkungen des Systemverhaltens auf Basis physikalischer Gesetze oder Anforderungen an das System mittels Partialmodellen. SysML eignet sich grundsätzlich zur Modellierung mechatronischer Systeme. Thramboulidis [TB10] beschreibt ein erweitertes, sogenanntes 3+1 SysML-Modell. Dieses Modell stützt sich zur domänenübergreifenden Beschreibung des Systems auf SysML und erwei-

tert dies um domänenspezifische Modelle der Mechanik, Elektronik und Informatik. Das 3+1 Modell ist in eine Entwicklungsumgebung für den Entwurf mechatronischer Systeme eingebettet [Thr05] und ermöglicht in diesem Sinne eine ganzheitliche Modellierung.

CONSENS (CONceptual design Specification technique for the ENgineering of mechatronic Systems) ist eine Spezifikationstechnik zur fachdisziplinübergreifenden Beschreibung der Konzeption mechatronischer Systeme. Die Beschreibung der Konzeption umfasst acht Aspekte (Umfeld, Anwendungsszenarien, Anforderungen, Funktionen, Wirkstruktur, Gestalt, Verhalten und Zielsystem) in Partialmodellen [GFD+08]. Diese Partialmodelle stehen zueinander in Beziehung und bilden ein kohärentes Abbild des zu entwickelnden Systems. Die große Ähnlichkeit zwischen den Modellierungsansätzen in SysML und CONSENS ermöglicht eine Erstellung von Systemkonzeptionen für CONSENS innerhalb der SysML [IKD+13].

CAMeL-View (Computer Aided Mechatronic Laboratory – Virtual Engineering Workbench) ist speziell für den modellbasierten Entwurf mechatronischer Systeme entwickelt worden [iXt19]. CAMeL-View bietet einen topologieorientierten Modellierungsansatz, der Standardkomponenten und -teilsysteme aus verschiedenen Domänen wie Mechanik, Hydraulik, Pneumatik und Regelungstechnik in Bibliotheken bereitstellt. Modelle, die in CAMeL-View erstellt wurden, können ohne zusätzliche Softwarewerkzeuge in HiL- (Hardware in the Loop)-Anwendungen unter Echtzeitbedingungen verwendet werden.

MODELICA ist eine quelloffene objektorientierte Programmiersprache zur topologieorientierten, multidomänen Modellierung mehrheitlich physikalischer Systeme [The19c]. Zur Modellierung verschiedener Komponenten aus den Domänen Mechanik, Elektronik, Elektrotechnik, Hydraulik, Pneumatik, Thermodynamik, Regelungstechnik und Prozesstechnik bietet MODELICA umfassende Bibliotheken an Standardkomponenten und -teilsystemen. Diese Bibliotheken ermöglichen eine hohe Wiederverwendbarkeit von Modellen und führen zu einer effizienten Modellbildung. MODELICA ist ein Standardwerkzeug bei der Modellbildung und Analyse mechatronischer Systeme, sodass verschiedene kommerzielle (Dymola [Das19], ANSYS/Simplorer [ANS19] u. w.) und frei verfügbare (OpenModelica [Ope19], Modelicac [The19b] u. w.) Entwicklungsumgebungen existieren.

VHDL-AMS (Very High Speed Integrated Circuit Hardware Description Language - Analog and Mixed-Signals) ist eine vom IEEE-standardisierte Hardware-Beschreibungssprache (IEEE 1076.1-1999). Sie dient der textbasierten Beschreibung der Struktur eines Systems und dessen Komponenten sowie zur allgemeinen Beschreibung des Systemverhaltens. Es können komplexe Systeme mit analogen, digitalen und gemischten Signalen betrachtet werden. VHDL-AMS basiert auf der VHDL (Very High Speed Integrated Circuit Hardware Description Language), die zur Beschreibung digitaler Systeme verwendet wird. Die VHDL-AMS unterstützt die hierarchische Beschreibung und Simulation von zeitkontinuierlichen, -diskreten und gemischten Systemen sowie die Modellierung auf unterschiedlichen Abstraktionsniveaus in elektrischen und nichtelektrischen Domänen. Die zu untersuchenden Systeme werden durch gewöhnliche Differentialgleichungen als Funktionen der Zeit und als algebraische Gleichungen abgebildet. Die VHDL-AMS kann mit Einschränkungen zur ganzheitlichen Modellierung mechatronischer Systeme eingesetzt werden [WRB+04].

MATLAB (MATrix LABoratory) ist eine von Mathworks entwickelte Programmiersprache für numerische Berechnungen auf Basis von Matrizenoperationen und wurde in eine Entwicklungsumgebung mit weitreichendem Funktionsumfang weiterentwickelt [The19a]. MATLAB bietet mittels numerischer und analytischer Berechnungsmethoden eine effi-

ziente Modellierung, Analyse, Aufbereitung und Visualisierung von beliebigen Daten, eine Implementierungs- beziehungsweise Entwicklungsumgebung für Algorithmen und Schnittstellen zu anderen Programmiersprachen wie C/C++, Java, Fortran und VHDL.

Simulink ist eine Toolbox von MATLAB und kann zur domänenübergreifenden signalflussorientierten Modellierung dynamischer Systeme genutzt werden. Die Modellierung erfolgt hierarchisch und kann sowohl zeitkontinuierlich als auch -diskret erfolgen. Die Nutzung von Erweiterungen wie *Simscape* ermöglicht in Simulink die Umsetzung von Mischformen zwischen topologie- und signalflussorientierter Modellierung. Darüber hinaus bietet Simulink umfangreiche Möglichkeiten domänenspezifische Modelle aus anderen Softwarewerkzeugen, wie Mehrkörpermodelle aus MSC ADAMS oder Finite-Elemente-(FE)-Modelle aus ANSYS, im Rahmen einer Co-Simulation einzubinden. Es existieren verschiedene frei verfügbare Alternativen zu MATLAB, die eine hohe Kompatibilität anstreben wie **GNU Octave** oder auf ähnliche Implementierung setzen wie **SciLab**.

Neben den bereits genannten Softwarewerkzeugen und Entwicklungsumgebungen drängen weitere Anbieter auf den Markt der domänenübergreifenden Modellierung, wobei hier die Ursprünge in spezifischen Domänen und Modellierungsansätzen liegen.

ANSYS Workbench / Twin Builder entwickelte sich von einem Softwarewerkzeug für mechanische Strukturanalysen auf Basis der Finite-Elemente-Methode (FEM) zu einer komplexen Entwicklungsumgebung, die verschiedenste Anwendungsbereiche abdeckt [ANS17]. Die Anwendungsbereiche liegen nicht nur im Bereich von FE-Anwendungen, sondern umfassen Methoden zur Erzeugung Digitaler Zwillinge (vergleiche die Definition in Abschnitt 3.1.2) (*ANSYS Twin Builder*) unter Verwendung einer multidomänen Modellierung auf Basis von MODELICA (*ANSYS Simplorer*).

Siemens Mechatronics Concept Designer ist proprietärer Bestandteil der *Siemens Product Lifecycle Management* Software und dient der Entwicklung mechatronischer Systeme mit Fokus auf Automatisierungstechnik [Sie19]. Basierend auf der CAD/CAE-Anwendung *NX* für die Gestaltung des Systems, werden weitere Anwendungen eingebunden, wie die unterstützte Auswahl geeigneter Aktorik und Sensorik. Diese basiert auf einer Simulation des dynamischen Systemverhaltens, der Ableitung des Entwurfs der Elektronik und des Entwurfs der Automation in Form von Ablaufdiagrammen, Regelung und weiteren Analysen.

Eine weitere Möglichkeit, eine domänenübergreifende Modellierung mechatronischer Systeme zu erreichen, ist die Co-Simulation spezialisierter Modelle innerhalb einer gemeinsamen Simulationsumgebung. Hier werden spezialisierte Modelle mittels standardisierter Schnittstellen (FMI, Functional Mock-Up Interface) miteinander verbunden.

ModelCenter Integrate ermöglicht die Automatisierung von Simulationen, indem verschiedenste spezialisierte Modelle über Co-Simulationen innerhalb der Simulationsumgebung verknüpft und beispielsweise für Parameterstudien berechnet werden [Pho19]. Eine Optimierung des Gesamtmodells innerhalb der Simulationsumgebung ist mit vorhandenen Methoden durchführbar.

Datadvance pSeven bietet ähnliche Funktionalitäten wie *ModelCenter Integrate*, weist jedoch einen stärkeren Fokus auf CAD-Modelle und somit auf die Gestalt des Systems auf [Dat19]. Neben automatisierten Parameterstudien und Optimierungen des Systemverhaltens, sind Algorithmen für eine Reduktion des Gesamtmodells auf Basis von Black-Box-Modellen vorhanden, die eine effizientere Analyse ermöglichen.

AVL Model.CONNECT ist auf Entwicklungen im automotive Bereich spezialisiert und ermöglicht eine Verknüpfung verschiedenster spezialisierter Modelle für die multi-domänen Modellierung eines virtuellen Prototypens [AVL19]. Innerhalb des virtuellen Prototypen sind die verschiedenen Modelle über Co-Simulationen eingebunden. Dieser virtuelle Prototyp wird für Optimierungen eingesetzt und besitzt die Möglichkeit, mittels Schnittstellen zu Echtzeit-Systemen zu einem Digitalen Zwilling (vergleiche die Definition in Abschnitt 3.1.2) erweitert zu werden.

2.7 Geschlossene Modellierung von Dynamik und Zuverlässigkeit

Die Entwicklung intelligenter, verlässlischer mechatronischer Systeme (Abschnitt 2.5) auf Basis der Selbstoptimierung benötigt eine geschlossene Modellierung der Zuverlässigkeit und des dynamischen Systemverhaltens für die Formulierung verlässllichkeitsorientierter Ziele [DHK+09]. Die geschlossene Modellierung kann nur mittels geeigneter Modellierungsverfahren umgesetzt werden, eine manuelle Ableitung von Verlässllichkeitskenngrößen ist nicht zielführend für den Entwurf selbstoptimierender Systeme, da diese für jede Zielfunktionsauswertung innerhalb der Mehrzieloptimierung durchgeführt werden müsste.

Die in Abschnitt 2.6 vorgestellten Ansätze für eine ganzheitliche Modellierung mechatronischer Systeme eignen sich direkt oder mittels Erweiterungen für die geschlossene Modellierung der Zuverlässigkeit und des dynamischen Verhaltens. Im Folgenden werden verschiedene Ansätze und Konzepte für eine geschlossene Modellierung vorgestellt, die eine vollständige oder teilweise automatisierte Ableitung von Systemzuverlässigkeitsmodellen beschreiben. Dabei werden häufig *lokale Zuverlässigkeitsschätzer* für die Abbildung der Zuverlässigkeit einzelner Komponenten oder Teilsysteme eingesetzt und innerhalb der geschlossenen Modellierung zu einem Modell der Zuverlässigkeit auf Systemebene zusammengefügt.

Die Spezifikationstechnik CONSENS wurde von Dorociak [Dor15] um Beschreibungsaspekte innerhalb der Partialmodelle hinsichtlich der Zuverlässigkeit und Sicherheit erweitert, um Verlässllichkeitsanalysen zu unterstützen. Es werden die Integration von verschiedenen Methoden, wie Fehlzustandsbäume und Bayes'sche Netze, zur Analyse der Kenngrößen der Verlässllichkeit herangezogen. Eine Abbildung des dynamischen Systemverhaltens ist innerhalb von CONSENS nicht umgesetzt, sodass ein Zusammenhang zwischen Dynamik und Zuverlässigkeit nicht hergestellt wird.

David beschreibt in [DIK10] ein Vorgehen zur Beschreibung der Zuverlässigkeit eines in SysML modellierten technischen Systems mittels einer automatisiert erzeugten FMEA. Es wird für SysML eine Bewertung zur Verwendung in Verlässllichkeitsanalysen aufgeführt und es werden Parallelen zwischen Modellen in SysML und AltaRica aufgezeigt, die für die Erzeugung der FMEA genutzt werden.

Hecht stellt in [HM08] den Algorithmus MOCET (Model based Certification Tool) zur teilweise automatisierten Durchführung einer FMEA auf Basis von Matlab/Simulink-Modellen vor. Aus einem Matlab/Simulink-Modell wird eine Liste der grafischen Elemente extrahiert, die vom Benutzer vollständig manuell hinsichtlich des Ausfallverhaltens der

einzelnen Komponenten und Teilsysteme ausgefüllt wird. Die Methode ist daher nur für wenig komplexe Systeme geeignet.

Fraracci entwirft in [Fra10] einen Algorithmus zur automatisierten Erzeugung einer FMEA aus Ausfallverhaltensmodellen in Matlab/Simulink. Diese Ausfallverhaltensmodelle werden vom Nutzer erstellt und bilden das Ausfall- und Verschleißverhalten der Komponenten und Teilsysteme ab. Die Erzeugung der FMEA erfolgt automatisch, benötigt aber wie der Algorithmus in [HM08] zusätzliche Verlässlichkeitsangaben der Komponenten und Teilsysteme.

Das Software-Werkzeug AutoFMEA wurde von Ricardo plc entwickelt, der Vertrieb aber eingestellt. Die AutoFMEA ist Teil einer Sicherheitsanalyse-Toolbox, die von Ricardo plc für Matlab/Simulink entwickelt wurde. Das Software-Tool ist vollständig in Matlab/Simulink eingebettet und in der Praxis bereits erprobt. Die Generierung und Auswertung der FMEA erfolgt vollständig automatisch. Fehlerarten und -wirkungen müssen vom Benutzer eingegeben werden, da auf keine Bibliothek oder Ähnliches zurückgegriffen werden kann.

Die Ansätze zur automatischen Erzeugung einer FMEA können unterstützend eingesetzt werden, beispielsweise für die Identifikation möglicher Fehlerursachen, liefern aber keine quantitative Analyse der Zuverlässigkeit und sind damit nicht direkt für die Umsetzung verlässlichkeitssteigernder Maßnahmen in selbstoptimierenden Systemen nach Abschnitt 2.3 einsetzbar.

Die geschlossene Modellierung eines elektronischen Schaltkreises mittels VHDL-AMS zur Systembeschreibung und Monte-Carlo-Simulation zur Abbildung des Ausfallverhaltens wird von Bestroy in eine Methode zur Berücksichtigung der Systemzuverlässigkeit im Entwicklungsprozess eingebunden und diskutiert [BML07]. Der Ansatz ist auf elektronische Komponenten und Teilsysteme beschränkt.

Papadopoulos stellt in [PM01] einen Algorithmus zur automatisierten Ableitung von Fehlzustandsbäumen aus Matlab/Simulink-Modellen vor. Das Ausfallverhalten der Komponenten und Teilsysteme wird mittels HAZOP (Hazard and Operability Study) ermittelt. Der Algorithmus analysiert automatisch die Struktur des Matlab/Simulink-Modells. Zusammen mit den Ergebnissen der HAZOP erfolgt die Generierung und Analyse der Fehlzustandsbäume in einem externen Tool, Fault Tree Plus von ISOGRAPH. Der Algorithmus wird unter anderem innerhalb der modelbasierten Sicherheitsanalyse HiP-HOPS (Hierarchically Performed Hazard Origin and Propagation Studies) [PM99] verwendet.

Tafarrod stellt in [TLS08] einen Algorithmus zur teilweise automatisierten Ableitung von Fehlzustandsbäumen aus Matlab/Simulink-Modellen vor. Der entwickelte Algorithmus ist vollständig in Matlab/Simulink implementiert und umfasst die Generierung sowie die Analyse der Fehlzustandsbäume. Die Definition von Primärereignissen hinsichtlich von Ausfällen der beteiligten Komponenten und Teilsysteme sowie deren Zusammenhang im Ausfallverhalten müssen vom Benutzer vorgegeben werden. Die Generierung und Analyse von statischen und dynamischen Fehlzustandsbäumen wird anschließend automatisiert durchgeführt. Die umfangreichen Benutzereingaben und der fehlende Zusammenhang zwischen Zuverlässigkeit und dynamischem Systemverhalten schränken diesen Ansatz ein.

Xiang stellt in [XYM+11] einen Algorithmus zur automatisierten Ableitung von Fehlzustandsbäumen aus verschiedenen Modellen der ganzheitlichen Modellierung vor. Es

wird ein Algorithmus entwickelt, der aus beliebigen Umgebungen zur ganzheitlichen Modellierung ein Modell von verlässlichkeitsrelevanten Informationen, ein Zwischenschritt zu Fehlzustandsbäumen, erstellt. Das Zwischenmodell muss vom Benutzer mit Informationen zu den Ereignissen und dem Ausfallverhalten angereichert werden, wobei ein Zusammenhang zwischen Systemverhalten und Zuverlässigkeit nicht hergestellt wird. Das Vorgehen wird an einem Systemmodell in SysML beispielhaft durchgeführt.

Schallert stellte in [Sch11] einen Ansatz der geschlossenen Modellierung mechatronischer Systeme auf Basis von MODELICA vor. Schallert verwendet ENADOT (Electrical Network Architecture Design Optimization Tool) in MODELICA, das neben einer Bibliothek für Komponenten und Teilsysteme auch die Algorithmen für Verlässlichkeitsanalysen beinhaltet. Das Tool ermöglicht neben der Modellierung des Systemverhaltens auch die Abbildung des Ausfallverhaltens elektrischer Komponenten und Teilsysteme. Als Systemzuverlässigkeitsmodelle stehen Zuverlässigkeitsblockdiagramme und Fehlzustandsbäume zur Verfügung. ENADOT ist für elektrische Systeme in Flugzeugen konzipiert und daher nicht auf beliebige Systeme anwendbar.

Leveson et. al. entwickelten eine teilautomatisierte Ableitung von Fehlzustandsbäumen für die Sicherheitsanalyse von Software auf Basis vordefinierter Vorlagen [LCS91]. Eine Anwendbarkeit auf mechatronische Systeme ist nicht gegeben.

Liggesmeyer et. al. stellten eine automatisierte Ableitung von Fehlzustandsbäumen vor, die auf Basis vorhandener formaler Beschreibungen des Systems wie Zustandsautomaten durchgeführt wird [LR98]. Diese Ableitung liefert zunächst einen Fehlzustandsbaum für eine qualitative Analyse, die Berechnung von Ausfallwahrscheinlichkeiten wird in einem zusätzlichen Tool durchgeführt. Die Methode zur Ableitung von Fehlzustandsbäumen wird unter anderem innerhalb der FSAP (Formal Safety Analysis Platform) [BV03] für modellbasierte Sicherheitsanalysen verwendet.

Bobbio beschreibt in [BPM+01] eine Methode zur Ableitung von Bayes'schen Netzen aus Fehlzustandsbäumen. Die Transformation von beliebigen Fehlzustandsbäumen wird aufgezeigt und an einem Beispiel durchgeführt. Mit einer Ableitung von Fehlzustandsbäumen aus Systemmodellen, wie sie in [PM01, TLS08, XYM+11, Sch11] vorgestellt wurden, ist eine automatisierte Ableitung von Bayes'schen Netzen aus Systemmodellen denkbar.

Die Analyse der Zuverlässigkeit und Sicherheit mechatronischer Systeme und Software mittels Fehlzustandsbäumen stellt einen weitverbreiteten Ansatz dar. Die automatisierte Ableitung der Fehlzustandsbäume aus bestehenden Modellen ist ein wiederkehrender Gegenstand der Forschung. Fehlzustandsbäume bilden das Ausfallverhalten mittels Boole'scher Zusammenhänge ab, was die Modellierungstiefe im Vergleich zu anderen Verfahren einschränkt (siehe die Gegenüberstellung gängiger Verfahren in Abschnitt 3.4.1 und Tabelle 3.1). Darüber hinaus sind die vorgestellten Methoden auf spezifische Modellierungsumgebungen anwendbar, sodass eine allgemeine Verwendung von aus dem Entwicklungsprozess zur Verfügung stehender Modelle nicht gegeben ist.

Pabst stellt ein Modell zur gleichzeitigen Abbildung der physikalischen Systemdynamik und der Dynamik der Änderung der Zuverlässigkeit in einer geschlossenen mathematischen Beschreibung vor [Pab05]. Auf Basis dieser Beschreibung wird eine Regelung der Zuverlässigkeit und des Systemverhaltens entwickelt, wobei die Umsetzung eines solchen Modells aufwendig sowie fehleranfällig sein kann, während der Beschreibungsansatz noch nicht ausreichend verallgemeinert ist.

Einen weiteren Ansatz zur geschlossenen Modellierung von Zuverlässigkeit und dynamischem Verhalten mechatronischer Systeme liefern spezialisierte Modellierungssprachen, die eine Annotation von Ausfallverhalten an Komponenten und Teilsysteme innerhalb eines Systemmodells verwenden.

Joshi entwickelte eine Erweiterung der Systemabbildung in Lustre [HCR+91] um Fehlermodelle einzelner Komponenten und Teilsysteme hin zu LustreFM [Jos08]. Hier kann komplexes Ausfallverhalten abgebildet werden und durch Methoden zur automatisierten Integration der Fehlermodelle in Lustre-Systemmodelle unterstützt werden.

Die AADL (Architecture Analysis and Design Language) nutzt ein Modell des fehlerfreien Systemverhaltens und reichert es mit Fehlermodellen an. Auf Basis dieses geschlossenen Modells können automatisierte Ableitungen von Fehlerbäumen [JVB07] oder von Petri Netzen [RKK07] durchgeführt werden.

AltaRica 3.0 [Pro14] beschreibt das Systemverhalten auf Basis von Zustandsautomaten und ist für die Abbildung von Zuverlässigkeitsaspekten mit verschiedenen Datenbanken verknüpft (unter anderem Mlt-Hdbk 217 [US 95]). Darüber hinaus stellt AltaRica 3.0 die automatisierte Ableitung von Fehlzustandsbäumen zur Verfügung, die in Ansätzen auf den Arbeiten von Liggesmeyer et. al. [LR98] basieren.

Die Verwendung spezialisierter Programmiersprachen schränkt den Einsatz beliebiger, aus dem Entwicklungsprozess zur Verfügung stehender, Systemmodelle deutlich ein.

Ein umfängliches Softwarewerkzeug für die modellbasierte Absicherung von Anforderungen, Sicherheit und Zuverlässigkeit mittels der automatisierten Ableitung einer FMEA und Fehlzustandsbäumen innerhalb einer Modellierungsumgebung (vergleiche ANSYS in Abschnitt 2.6) wird mit ANSYS medini analyze geliefert [ANS19]. Dabei beruht die Systembeschreibung auf der SysML, die mit zusätzlichen Modellen erweitert werden kann, sodass beispielsweise die Zuverlässigkeit einzelner Komponenten über einfache Zuverlässigkeitskataloge abgebildet wird (vergleiche Abschnitt 3.3).

Die geschlossene Modellierung der Systemzuverlässigkeit und des dynamischen Systemverhaltens mechatronischer Systeme wird im Zuge der Umsetzung Digitaler Zwillinge benötigt. Ein Digitaler Zwilling, eine Abbildung des realen Systems unter Zuhilfenahme von realen Systemdaten (eine Definition wird in Abschnitt 3.1.2 vorgestellt), erfordert eine geschlossene Modellierung. Die im Folgenden vorgestellten Arbeiten gehen durch die Integration realer Systemdaten über die ausschließlich geschlossene Modellierung hinaus, liefern aber dennoch Ansätze für eine Umsetzung.

Zhao et. al. [ZTZ13] stellen eine Prognose der verbleibenden Lebensdauer eines Getriebes für Ermüdungsrisse am Zahnfuß der enthaltenen Zahnräder vor. Es wird dafür ein Modell der Dynamik des Getriebes zusammen mit Schadensmodellen und Zustandsüberwachungsdaten des realen Systems kombiniert.

Li et. al. [LHW+17] präzisieren die verbleibende Lebensdauer von Flugzeugstrukturen hinsichtlich des Ermüdungsrisswachstums unter Berücksichtigung der Unsicherheit über verschiedene Parametern, wie auf die Struktur wirkende Lasten, aktuelle Risslänge und Beobachtungen aus Zustandsüberwachungsdaten.

Die hier exemplarisch vorgestellten Arbeiten zur Umsetzung eines Digitalen Zwillings für die Prognose der verbleibenden Lebensdauer sind jeweils sehr anwendungsbezogen und liefern keine allgemeine geschlossene Modellierung der Systemzuverlässigkeit und

des dynamischen Systemverhaltens mechatronischer Systeme. Dies ist typisch für aktuelle Umsetzungen von Digitalen Zwillingen, die zweckgebunden implementiert werden (vergleiche Abschnitt 3.1.2), aber zur Abbildung der Systemzuverlässigkeit auf keine einheitliche Methode zurückgreifen. Die vorgestellten Methoden sind darüber hinaus nicht auf beliebige Modellierungsumgebungen anwendbar, sodass eine Verallgemeinerung auf die zur Verfügung stehenden Modelle des Entwicklungsprozesses nicht stattfindet.

2.8 Fazit

Die Entwicklung intelligenter, verlässlicher Systeme stellt eine große Herausforderung dar, der durch geeignete softwareunterstützte Methoden begegnet werden muss. Selbstoptimierende technische Systeme, in der Klasse intelligenter technischer Systeme, sind für eine Berücksichtigung verlässlichkeitsorientierter Ziele während des Betriebs auf geeignete mathematische Formulierungen dieser Ziele angewiesen. Auf Basis der Definition der Verlässlichkeit ist eine fokussierte Betrachtung der Zuverlässigkeit aufgrund der starken Abhängigkeiten der weiteren Kenngrößen Verfügbarkeit und Sicherheit zielführend. Die Abbildung verlässlichkeitsorientierter Ziele innerhalb der Mehrzieloptimierung erfordert eine geschlossene Modellierung der Abhängigkeit zwischen der Zuverlässigkeit und des dynamischen Systemverhaltens [SW15]. Der schematische Ablauf einer Mehrzieloptimierung für zuverlässigkeits- und verhaltensorientierte Ziele, wie er auf Basis einer manuellen Ableitung eines geeigneten Zuverlässigkeitsmodells gestaltet sein könnte, ist in Abbildung 2.8 dargestellt.

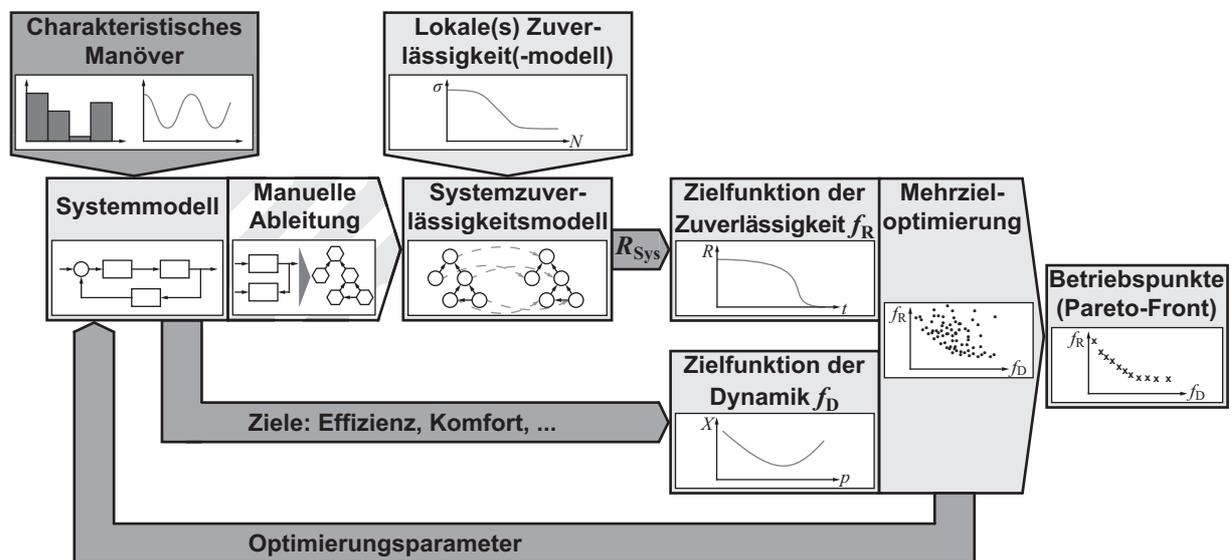


Abbildung 2.8: Klassischer Ansatz zur Abbildung zuverlässigkeits- und verhaltensbasierter Ziele für die Berechnung pareto-optimaler Betriebspunkte mittels Mehrzieloptimierung (vergleiche Abschnitt 2.4).

Die iterativ ablaufende Mehrzieloptimierung verläuft demnach in keinem geschlossenen Zyklus, da eine Auswertung der zuverlässigkeitsorientierten Zielfunktion nicht ohne Nutzereingaben (in Form einer manuellen Ableitung oder Anpassung des Zuverlässigkeitsmodells) in jeder Iteration der Optimierung erfolgen kann. Eine Anwendung von Mehrzieloptimierungsverfahren für Zielformulierungen auf Basis einer allgemeinen Methode zur

Ableitung eines Systemzuverlässigkeitsmodells mit hinreichender Modellierungstiefe ist aktuell nicht gegeben. Folglich ist die Umsetzung verlässlichkeitssteigernder Maßnahmen mithilfe der Selbstoptimierung auf die anwendungsspezifische Umsetzung von verlässlichkeitsorientierten Zielfunktionen beschränkt.

Eine geschlossene Modellierung der Zuverlässigkeit und des dynamischen Systemverhaltens ist in der Lage diese Lücke zu schließen und die verallgemeinerte Formulierung verlässlichkeitsorientierte Zielfunktionen zu ermöglichen. Um eine umfängliche Unterstützung des Entwurfs verlässlichkeitssteigernder Maßnahmen durch Selbstoptimierung zu gewährleisten werden verschiedene Anforderungen an einen Modellierungsansatz zur geschlossenen Abbildung gestellt. Die zentralen Anforderungen sind:

- Modellierung beliebiger mechatronischer Systeme,
- Modellierung der Zuverlässigkeit und des dynamischen Systemverhaltens,
- Anwendung auf beliebige Entwicklungsumgebungen zur ganzheitlichen Modellierung mechatronischer Systeme,
- Geringer Bedarf an Nutzereingaben,
- Hohe Modellierungstiefe der Zuverlässigkeit.

Demnach sollen mechatronische Systeme (nach Abschnitt 2.6) ganzheitlich abgebildet werden und keine Beschränkung hinsichtlich der zu modellierenden Domänen vorhanden sein. Die geschlossene Modellierung erfordert zwingend die Abbildung der Zuverlässigkeit und des dynamischen Systemverhaltens sowie der Interaktion dieser beiden Systemaspekte. Ebenso ist eine effiziente Verwendung vorhandener Informationsquellen gefordert. Daher sollen möglichst aus dem Entwicklungsprozess vorhandene Modelle verwendet werden ohne jedoch Einschränkungen hinsichtlich der verwendeten Entwicklungsumgebung hinzunehmen. Dies hat zum Ziel, dass zusätzliche Nutzereingaben reduziert werden. Die geschlossene Modellierung soll, abgesehen von initialen Eingaben zur Abbildung verschiedener Verlässlichkeitsaspekte, ohne weitere Nutzereingaben während der Optimierung auskommen. Die Abbildung der Zuverlässigkeit soll schließlich in einer hohen Modellierungstiefe erfolgen, um alle relevanten Aspekte des Ausfallverhaltens intelligenter technischer Systeme zu erfassen. Hier soll eine Berücksichtigung von komplexem Systemverhalten zur Steigerung der Robustheit gegenüber Ausfällen von Komponenten oder Teilsystemen, beispielsweise mittels Rekonfiguration als mögliche Umsetzung von Redundanzen, erfolgen.

Die in Abschnitt 2.7 vorgestellten Ansätze zur geschlossenen Modellierung weisen hinsichtlich der Anforderungen verschiedene Einschränkungen für eine allgemeine Formulierung von verlässlichkeitsorientierten Zielen für die Mehrzieloptimierung auf. Die geschlossene Modellierung ist im Stand der Technik zumeist mittels teil- / automatisierter Ableitungen eines Zuverlässigkeitsmodells auf Systemebene umgesetzt. In Abschnitt 2.7 ist gezeigt worden, dass die bestehenden Ansätze verschiedene individuelle Einschränkungen oder eine Kombination von Einschränkungen aufweisen. In Tabelle 2.1 sind diese Methoden zusammenfassend und hinsichtlich der zentralen Anforderungen an eine geschlossene Modellierung bewertet. Hinsichtlich dieses Überblicks ist festzuhalten, dass im Stand der Technik große Einschränkungen der Methoden bezüglich der Anwendbarkeit auf verschiedene Modellierungsumgebungen vorhanden sind. Dies ist damit begründet, dass die Methoden der geschlossenen Modellierung jeweils auf eine spezifische Modellierungsumgebung aufbauen. Die Möglichkeiten zur Einbeziehung weiterer Modelle sind stark eingeschränkt werden nicht in der Bewertung der Zuverlässigkeit berücksichtigt. Als weitere

	Modellierung beliebiger mechatronischer Systeme	Anwendung auf beliebige Modellierungsumgebungen	Modellierung von Dynamik und Zuverlässigkeit	Geringer Bedarf an Nutzereingaben	Hohe Modellierungstiefe der Zuverlässigkeit
[DIK10, HM08, Fra10, PM99]	●	○	○	●	○
[Dor15, PM01, TLS08, XYM+11, LCS91, LR98, BV03]	●	○	○	●	●
[Pab05, HCR+91, Jos08, JVB07, RKK07, Pro14, ANS19]	●	●	○	●	●
[BML07, Sch11, ZTZ13, LHW+17]	○	○	●	●	●

Tabelle 2.1: Zusammenfassende Bewertung der Methoden des Stands der Technik hinsichtlich der Anforderungen an eine geschlossene Modellierung; ●: voll erfüllt, ●: teils erfüllt, ○: nicht erfüllt.

Einschränkung ist zusammenfassend die Modellierung des Einflusses des dynamischen Systemverhaltens auf die Zuverlässigkeit zu nennen. Das zugrundeliegende Systemmodell wird meist ausschließlich dazu verwendet die Topologie eines Zuverlässigkeitsmodells zu identifizieren. Der Einfluss einer Veränderung des dynamischen Systemverhaltens auf die Zuverlässigkeit wird nicht abgebildet. Die Ausnahme stellen hier die Methoden zur Erstellung Digitaler Zwillinge dar, die jedoch hochgradig anwendungsspezifisch umgesetzt worden sind und keine Verallgemeinerung bieten.

Es fehlt eine umfängliche Methode für die geschlossene Modellierung der Zuverlässigkeit und des dynamischen Systemverhaltens für die Entwicklung verlässlichkeitssteigernder Maßnahmen mithilfe der Selbstoptimierung. Die Zielsetzung dieser Arbeit ist daher die Umsetzung einer solchen geschlossenen Modellierung der Zuverlässigkeit und des dynamischen Systemverhaltens, die auf vorhandenen Modellen des Entwurfs mechatronischer Systeme aufsetzt und auf verschiedene Modellierungsumgebungen anwendbar ist. Ausgehend von der Zuverlässigkeit können unter Berücksichtigung zusätzlicher Informationen, wie Instandhaltungsstrategien und Sicherheitsanforderungen, weitere Kenngrößen, wie Sicherheit und Verfügbarkeit, abgebildet werden. Eine geschlossene Modellierung der Zuverlässigkeit und des dynamischen Systemverhaltens mechatronischer Systeme ist daher zielführend für eine Abbildung weiterer Kenngrößen der Verlässlichkeit und erforderlich für den Entwurf von Maßnahmen zur Steigerung der Verlässlichkeit auf Basis der Selbstoptimierung.

3 Integrierte Modellierung von Zuverlässigkeit und dynamischem Verhalten

Die Verfahren zur geschlossenen Modellierung der Zuverlässigkeit und des dynamischen Systemverhaltens im vorigen Abschnitt 2.7 weisen verschiedene Einschränkungen hinsichtlich der Modellierungstiefe, Modellierungsumgebungen, dem Bedarf an zusätzlichen Nutzereingaben und der Umsetzung einer geschlossenen Mehrzieloptimierung mechatronischer Systeme für zuverlässigkeits- und verhaltensbasierte Ziele auf (vergleiche Abbildung 2.8).

Dem gegenüber bietet das in dieser Arbeit entwickelte Verfahren die Möglichkeit einer geschlossenen Berechnung pareto-optimaler Betriebspunkte mittels Mehrzieloptimierungsverfahren. Dies geschieht mithilfe einer Annotation von lokalen Zuverlässigkeitsschätzern an ein bestehendes Systemmodell des dynamischen Verhaltens in Kombination mit einer automatisierten Ableitung eines Systemzuverlässigkeitsmodells. Die *integrierte Modellierung* reduziert die zur Ableitung des Systemzuverlässigkeitsmodells benötigten Nutzereingaben auf Informationen, die nicht aus dem Systemmodell zu extrahieren sind. Das entwickelte Verfahren ist zudem unabhängig von der Modellierungsumgebung des bestehenden Systemmodells des dynamischen Verhaltens und ist damit auf unterschiedliche Umgebungen anwendbar. Die grundlegenden Ansätze des MBSE, bei dem Modelle als zentrale Informationsquelle und dem Informationstransfer dienen, werden durch das entwickelte Verfahren unterstützt. Ein schematisches Ablaufdiagramm der integrierten Modellierung ist in Abbildung 3.1 dargestellt.

Das *integrierte Modell*, bestehend aus dem *Systemmodell* des dynamischen Verhaltens und *lokalen Zuverlässigkeitsschätzern*, wird für ein *charakteristisches Manöver* simuliert. Das charakteristische Manöver stellt dabei eine repräsentative Belastungssituation des zu entwickelnden Systems dar, beispielsweise standardisierte Windlasten und -richtungen für eine Windenergieanlage, und beinhaltet alle relevanten Lastfälle. Daher kann das charakteristische Manöver mehrere Lastfälle enthalten und ermöglicht eine Bewertung des Systems hinsichtlich verschiedener Entwurfsziele, wie Komfort, Effizienz und Zuverlässigkeit. Das Ergebnis der Simulation des charakteristischen Manövers sind unter anderem die Belastungen auf einzelne Komponenten und Teilsysteme. Diese Belastungen dienen als Eingangsgrößen für die lokalen Zuverlässigkeitsschätzer und ermöglichen die individuelle Bewertung der Zuverlässigkeit von Komponenten und Teilsystemen.

Die *automatisierte Ableitung* des Systemzuverlässigkeitsmodells basiert auf den im integrierten Modell enthaltenen Informationen. Zusätzliche Nutzereingaben werden ausschließlich für weitere Modellierungsaspekte, wie Redundanzen oder externe Ereignisse mit Einfluss auf die Systemzuverlässigkeit, benötigt. Die automatisierte Ableitung verwendet die Topologie des Systemmodells, um die Struktur des Systemzuverlässigkeitsmodells zu generieren. Die Parametrierung des Systemzuverlässigkeitsmodells erfolgt über

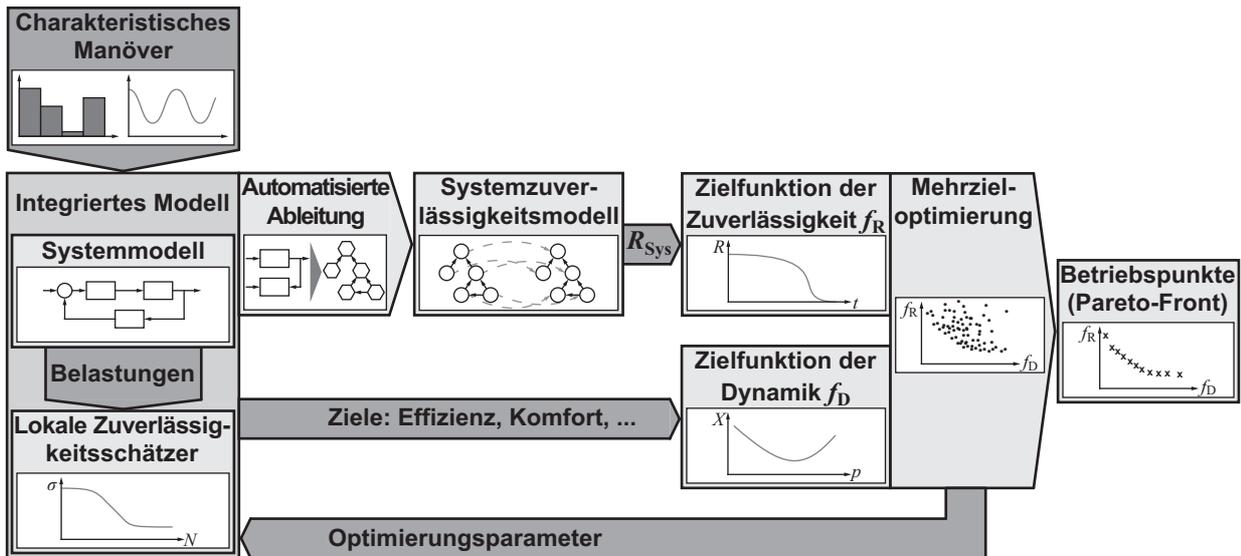


Abbildung 3.1: Ansatz der integrierten Modellierung zur Abbildung von Zuverlässigkeits- und verhaltensbasierten Zielen für die Berechnung von Pareto-optimalen Betriebspunkten mittels Mehrzieloptimierung.

die Ergebnisse der lokalen Zuverlässigkeitsschätzer. Die Analyse des Systemzuverlässigkeitsmodells bietet in Abhängigkeit der Lebenszyklusphase (vergleiche Abbildung 3.2) unterschiedliche Anwendungsmöglichkeiten zur Unterstützung des Entwicklungsprozesses oder zur Verbesserung der Instandhaltung im Betrieb. So ist das Hauptziel der integrierten Modellierung, eine detaillierte Eigenschaftsabsicherung bezüglich der Zuverlässigkeit parallel zum domänenspezifischen Entwurf während des Entwicklungsprozesses. Die Integrität, der aus dem Entwurf vorhandenen Modelle, soll vollständig gewahrt werden: Bestehende Modelle aus dem Entwurf mechatronischer Systeme werden durch die integrierte Modellierung nicht verändert und erfordern keine Anpassungen. Eine weitere Anwendung ist die Umsetzung von Maßnahmen zur Steigerung der Verlässlichkeit in mechatronischen Systemen während des Betriebs auf Basis der Mehrzieloptimierung mithilfe der Selbstoptimierung. Auf Basis einer Mehrzieloptimierung werden während der Entwicklung Pareto-optimale Betriebspunkte bezüglich verschiedener Zielfunktionen der Zuverlässigkeit f_R und der Dynamik f_D des Systems berechnet. Im Betrieb werden diese Betriebspunkte mit den Mitteln der Selbstoptimierung eingestellt. Ein grundlegendes Problem bereits existierender geschlossener Modellierungsverfahren (Abschnitt 2.7) besteht darin, den Einfluss, der durch den Optimierungsalgorithmus variierten Parameter zur Beeinflussung des Systemverhaltens, direkt auf die Systemzuverlässigkeit abzubilden. Die *Optimierungsparameter* stellen freie Parameter des Systemmodells, wie Reglerkonfigurationen und -parameter, Geometrie- und Steifigkeitswerte, dar. Dabei ist durch den Nutzer sicherzustellen, dass diese Parameter einen Einfluss auf die gewählten Ziele ausüben. Aus allen berechneten Zielfunktionswerten wählt der Mehrzieloptimierungsalgorithmus eine Menge an Pareto-optimale Kompromissen zwischen den Zielfunktionen f_R und f_D aus. Für konkurrierende Ziele ergibt sich die *Pareto-Front* als Menge der möglichen Betriebspunkte für ein selbstoptimierendes System.

Im weiteren Verlauf dieses Abschnitts wird die integrierte Modellierung von Zuverlässigkeit und dynamischem Verhalten vorgestellt. Zunächst wird die Umsetzung in die verschiedenen Lebenszyklusphasen anhand der Integration in den Entwicklungsprozess

mechatronischer Systeme, basierend auf VDI 2206 und anhand der Implementierung eines Digitalen Zwillings, erläutert. Anschließend wird eine geeignete Modellierungsmethode zur Abbildung der Systemzuverlässigkeit ausgewählt und die in der integrierten Modellierung verwendeten Modelle charakterisiert sowie Anforderungen benannt. Das abzuleitende Systemzuverlässigkeitsmodell und verschiedene Modellierungsaspekte werden detailliert beschrieben, um anschließend die automatisierte Ableitung vorzustellen. Ausgehend von dem Systemzuverlässigkeitsmodell werden unterschiedliche Aspekte einer weiteren Verwendung diskutiert, wie die Formulierung von zuverlässigkeitsorientierten Zielfunktionen für Systemoptimierungen sowie Schnittstellen und Abhängigkeiten zu weiteren Kenngrößen der Verlässlichkeit.

3.1 Integration in Lebenszyklusphasen mechatronischer Systeme

Jedes technische System durchläuft während seines Lebenszykluses verschiedene Phasen, in denen jeweils verschiedene Anforderungen an das System gestellt werden. Die formale Definition dieser Phasen ist branchen- beziehungsweise anwendungsspezifisch ausgeprägt (eine Übersicht ist in [WRF15] zu finden) und ermöglicht die strukturierte Betrachtung des Systems über den gesamten Lebenszyklus. Der Lebenszyklus kann mit einer ersten Idee oder einem Konzept beginnen, nach Entwicklung und Produktion die Nutzung sowie den Service umschließen und schließlich mit der Dekommissionierung und Entsorgung enden. Eine generalisierte Darstellung über die Lebenszyklusphasen technischer Systeme ist mit ISO/IEC/IEEE 15288:2015 gegeben und in Abbildung 3.2 gezeigt.

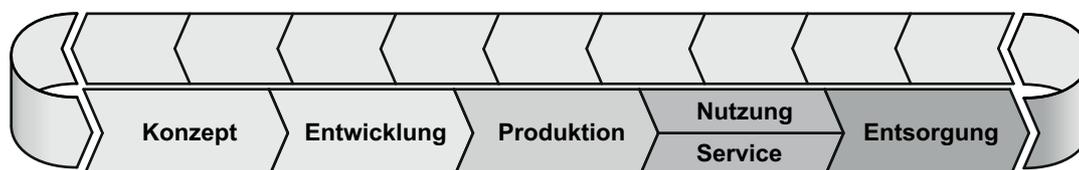


Abbildung 3.2: Generalisierte Lebenszyklusphasen technischer Systeme nach ISO/IEC/IEEE 15288:2015

Die Beschreibung der generalisierten Lebenszyklusphasen basiert auf Walden [WRF15]. In der *Konzeptphase* wird zunächst ein abstraktes vorläufiges Konzept, aufbauend auf einer Analyse der Notwendigkeit einer Neu- oder Weiterentwicklung, der Potenziale und Möglichkeiten, welche durch organisatorische Gegebenheiten oder technologische Entwicklungen ermöglicht werden, erstellt und Anforderungen definiert. Auf Basis des vorläufigen Konzepts werden verschiedene weitere entwickelt, aus denen eines für die weitere Entwicklung ausgewählt wird. Die *Entwicklung* technischer Systeme greift das ausgewählte Konzept auf und beginnt mit der Ausdifferenzierung der Anforderungen, dem frühzeitigen Entwurf des Systems, der Ausarbeitung von domänenspezifischen Teillösungen und der Integration dieser in das Gesamtsystem. Während der Entwicklung wird der Entwurf kontinuierlich gegenüber den Anforderungen abgeglichen. Dies geschieht zumeist gegen Ende der Entwicklung mittels prototypischen Aufbauten zur Verifizierung und Validierung des entwickelten Systems. Innerhalb dieser Arbeit wird für die Entwicklungsphase der modellbasierte Entwurf mechatronischer Systeme nach VDI 2206 zugrunde gelegt.

In der Phase der *Produktion* des Systems können Modifikationen des Entwurfs erforderlich werden, um die Herstellbarkeit zu gewährleisten, die Produktionskosten zu senken oder um nachträglich Systemfunktionalitäten zu integrieren. Änderungen des Systems, als auch dessen Produktion, erfordern eine Verifizierung anhand der aus der Konzept- und Entwicklungsphase bekannten Anforderungen. In der *Nutzungsphase* wird das technische System unter nominalen Betriebs- und Umweltbedingungen verwendet, um die geforderte Funktionalität bereitzustellen. Der Umfang an Funktionalität kann über der Betriebsdauer durch Modifikationen und Erweiterungen angepasst werden. Die *Servicephase* des technischen Systems findet parallel zu dessen Nutzung statt und dient der Bereitstellung, Aufrechterhaltung und Anpassung der Systemfunktionalität. Damit verbundene Aktivitäten sind die Instandhaltung, die Modifizierung, (Funktions-)Erweiterung und Optimierung zur Absicherung eines kontinuierlichen Betriebs, der Reduzierung von Betriebskosten oder Maßnahmen zur Verlängerung der Betriebsdauer. In der letzten Phase des Lebenszyklus wird das technische System und seine betreffenden Services außer Betrieb genommen. Das System wird stillgelegt beziehungsweise dekommissioniert und der *Entsorgung* zugeführt.

Die Integration der integrierten Modellierung von Zuverlässigkeit und dynamischem Systemverhalten in die Lebenszyklusphasen mechatronischer Systeme wird im Folgenden anhand der Phasen Entwicklung (Abschnitt 3.1.1), Nutzung und Service (Abschnitt 3.1.2) betrachtet.

3.1.1 Unterstützung des Entwicklungsprozesses

Der Entwicklungsprozess mechatronischer Systeme wird in der Richtlinie VDI 2206 beschrieben. Diese Richtlinie systematisiert mittels des sogenannten V-Modells (vergleiche Abbildung 3.3 innerer Teil) den Makrozyklus des modellbasierten Entwurfs mechatronischer Systeme in drei Phasen: *Systementwurf*, *domänenspezifischer Entwurf* und *Systemintegration*. Diese drei Phasen werden parallel durch die *Modellbildung und -analyse* unterstützt und abgesichert. Zu Beginn des Entwicklungsprozesses stehen die Anforderungen an das System. Diese werden während des Entwurfsfortschritts mit dem Entwicklungsstand abgeglichen, sodass über die *Eigenschaftsabsicherung* sichergestellt ist, dass das System die Anforderungen erfüllt. Das Ergebnis des Entwicklungsprozesses im V-Modell ist das Produkt, wobei es sich hier um eine, dem zunehmenden Reifegrad des Systems entsprechende, Instanz, wie Labormuster, Prototyp oder Seriensystem handeln kann. Das zu entwickelnde System wird mittels weiterer Durchläufe des V-Modells weiter bis zur Serienreife konkretisiert. Die integrierte Modellierung von Zuverlässigkeit und dynamischem Verhalten ist parallel zum V-Modell in allen drei Entwurfsphasen und Produktreifegraden für die Absicherung der Anforderungen an die Verlässlichkeit einsetzbar, sodass Schnittstellen zu weiteren Kenngrößen für die Absicherung verwendet werden können (vergleiche Abschnitte 2.1 und 3.6). Die Integration in den Entwicklungsprozess nach VDI 2206 ist in Abbildung 3.3 gezeigt.

In der Phase des Systementwurfs beginnt die abstrakte Beschreibung des Systems anhand einer Funktionsstruktur mit der Hierarchisierung der Hauptfunktion in Teil- und Unterfunktionen auf Basis der Anforderungen. Anschließend werden Wirkprinzipien und entsprechende Lösungselemente zur Erfüllung von Teilfunktionen gesucht. Diese Lösungselemente werden zur Erfüllung der Hauptfunktion zu einer Wirkstruktur verknüpft. Zumeist stehen verschiedene Wirkprinzipien und Lösungselemente zur Erfüllung einer Teil-

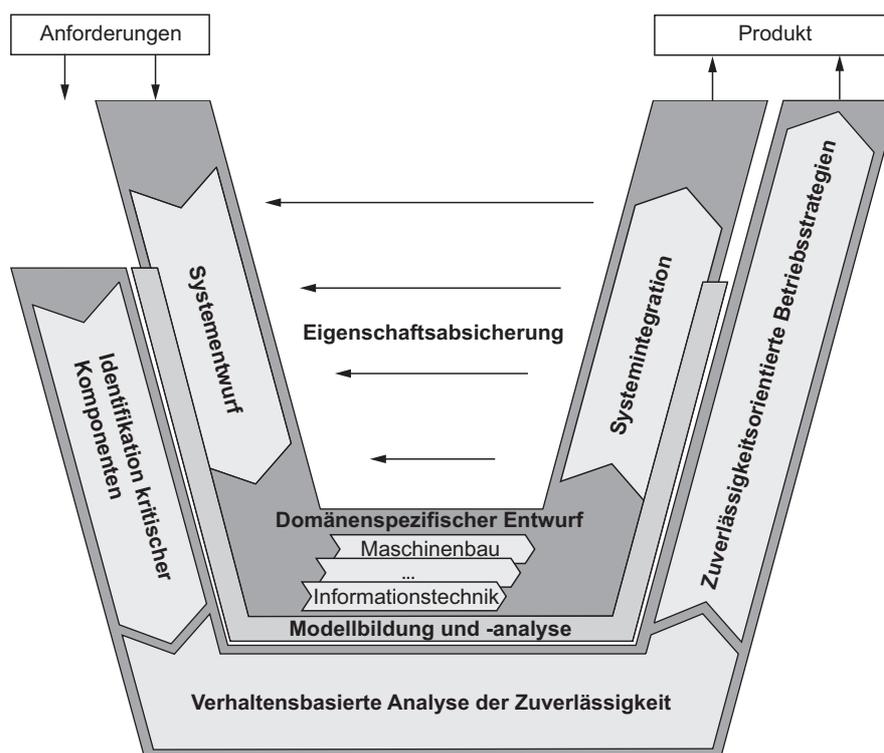


Abbildung 3.3: Integrierte Modellierung zur Unterstützung des Entwicklungsprozesses mechatronischer Systeme nach VDI 2206 [HKG+17].

funktion zur Verfügung, sodass verschiedene Lösungsvarianten gebildet werden können. Diese prinzipiellen Lösungsvarianten werden anhand technischer und wirtschaftlicher Kriterien einer Bewertung unterzogen und eine Variante ausgewählt. Dieses Lösungskonzept enthält am Ende der Phase des Systementwurfs alle wesentlichen Wirkungsweisen, funktionalen Zusammenhänge sowie die Architektur des zu entwickelnden Systems. Die integrierte Modellierung ist in dieser Phase für die *Identifikation kritischer Komponenten* und Teilsysteme einsetzbar, sobald die Wirkstruktur mit allen wesentlichen funktionalen Zusammenhängen verfügbar ist. Dies ist gleichbedeutend mit dem Einsetzen der Modellbildung und -analyse im V-Modell. Der Ansatz kann auch zur Bewertung der prinzipiellen Lösungsvarianten herangezogen werden. Eine Aussage über die Zuverlässigkeit des zukünftigen Systems ist nur qualitativ über kritische Komponenten, Teilsysteme oder deren Kombinationen möglich. Als Grundlage einer ersten Analyse dient die Fehlermöglichkeits- und -auswirkungsanalyse (FMEA) für die gewählten Lösungselemente und -varianten. Die FMEA dient hier hauptsächlich der Erfassung möglicher Fehlerursachen und resultierender Fehlzustände sowie Ausfällen. Auf dieser Basis können abstrakte Zuverlässigkeitsschätzer ausgewählt werden, die eine erste grobe Näherung an die resultierende Zuverlässigkeit des Gesamtsystems und somit eine quantitative Bewertung liefern [Dor15]. So ist es möglich, bereits in frühen Entwicklungsphasen Schwachstellen in der Architektur zu erkennen und zu beheben.

Im domänenspezifischen Entwurf wird das gewählte Lösungskonzept in den beteiligten Domänen weiter konkretisiert und die Entwicklung durch geeignete Softwarewerkzeuge unterstützt. Innerhalb dieser Entwurfsphase werden detaillierte Modelle des zu entwickelnden Systems erstellt. Es werden meist domänenübergreifende Simulationswerkzeuge für die Modellierung des dynamischen Systemverhaltens verwendet, um eine ganzheitli-

che Modellierung innerhalb einer gemeinsamen Umgebung zu realisieren (vergleiche Abschnitt 2.6). Auf Grundlage dieser Modelle ermöglicht die integrierte Modellierung eine *verhaltensbasierte Analyse der Verlässlichkeit* des aktuellen Entwicklungsstands des Systems. Die fortschreitende Konkretisierung des Systems macht eine detaillierte Beschreibung des dynamischen Verhaltens des Systems, seiner Architektur und der funktionalen Zusammenhänge verfügbar, die eine genauere Abbildung der Zuverlässigkeit ermöglichen. Im Sinne des MBSE stellen die vorhandenen Entwicklungsmodelle die zentrale Informationsquelle für die Absicherung der Zuverlässigkeit auf Systemebene dar, die nicht erst mit der *Systemintegration* beginnt [WRF15, HKG+17].

Innerhalb der Entwurfsphase *Systemintegration* werden die im domänenspezifischen Entwurf erarbeiteten Teillösungen zu einem Gesamtsystem verknüpft. Der Fokus liegt hierbei auf der Analyse und Absicherung der Schnittstellen und der Kompatibilität der Teillösungen, wobei wieder auf domänenübergreifende Simulationswerkzeuge und Modelle aus der vorigen Entwurfsphase zurückgegriffen wird. Die integrierte Modellierung dient auch hier primär der Absicherung der Zuverlässigkeit des integrierten Systems. Darüber hinaus können *verlässlichkeitsorientierte Betriebsstrategien*, wie Maßnahmen zur Steigerung der Verlässlichkeit entwickelt werden. Unter Berücksichtigung der domänenübergreifenden Modelle werden mittels Mehrzieloptimierung pareto-optimale Betriebspunkte berechnet und der Entwicklung einer Betriebsstrategie des mechatronischen Systems zugeführt. Mithilfe der integrierten Modellierung ist es direkt möglich zuverlässigkeitsorientierte Entwurfsziele innerhalb der Mehrzieloptimierung zu berücksichtigen.

Die integrierte Modellierung von Zuverlässigkeit und dynamischem Verhalten baut systematisch auf vorhandenen Entwicklungsmodellen des Entwurfs mechatronischer Systeme auf. Somit ist eine wirkungsvolle Unterstützung des Entwicklungsprozesses umgesetzt, mit der parallel zum Entwurfsfortschritt die Zuverlässigkeit des zu entwickelnden Systems bewertet werden kann.

3.1.2 Digitaler Zwilling für die Instandhaltung

In den parallelen Lebenszyklusphasen (Nutzung und Service) ist die Instandhaltung ein wichtiger Bestandteil zur Gewährleistung einer hohen Verfügbarkeit des Systems. Mit der steigenden Komplexität intelligenter technischer Systeme nehmen auch die Anforderungen an die Instandhaltung dieser Systeme zu. Im Rahmen der Digitalisierung stehen vermehrt Betriebs- und zusätzlich Messdaten der Systeme zur Verfügung, die in Kombination mit bestehenden Modellen des Entwicklungsprozesses zu einem *Digitalen Zwilling* integriert werden. Ein Digitaler Zwilling ist nach [Kuh17, Stu17, BR16] definiert als eine vollständige, virtuelle Repräsentation eines realen Objekts (Systems, Produkts oder Prozesses) und beinhaltet alle für den gewählten Anwendungsfall relevanten Informationen. Der Digitale Zwilling wird demnach für einen definierten Zweck erstellt, sodass eine vollumfängliche Abbildung aller denkbaren Aspekte des realen Objekts nicht zielführend ist. Ein solcher Zweck kann beispielsweise die Umsetzung einer vorausschauenden Instandhaltung, Zustandsüberwachung, Integration von Systemerweiterungen oder -modifikationen im After-Sales-Management oder zum Transfer von Daten aus der Anwendung in den Entwicklungsprozess zukünftiger Produktgenerationen sein [Stu17, GV17]. Über definierte Schnittstellen ist der Digitale Zwilling mit dem realen Objekt verbunden, sodass dieser das reale Objekt stets hinreichend genau abbildet. Dies können in frühen Lebenszyklusphasen (Konzept und Entwicklung) Schnittstellen zu vorhandenen Datenbanken,

Produkt-Lebenszyklus-Management-(PLM)-Systeme oder ähnlichen Informationsquellen sein. Mit fortschreitendem Lebenszyklus werden die Schnittstellen zu diesen Informationsquellen um Schnittstellen zu lokalen Mess- und Betriebsdatenerfassungssystemen bis hin zu dezentralen, cloud-basierten Diensten ergänzt.

Der für diese Arbeit relevante Digitale Zwilling dient der Umsetzung und Unterstützung der Instandhaltung mechatronischer Systeme und wird auf Basis der integrierten Modellierung von Zuverlässigkeit und dynamischem Verhalten umgesetzt. Abbildung 3.4 gibt einen Überblick über die Verknüpfung der Lebenszyklusphasen mit dem Digitalen Zwilling, den maßgeblichen Schnittstellen und den relevanten Modellen.

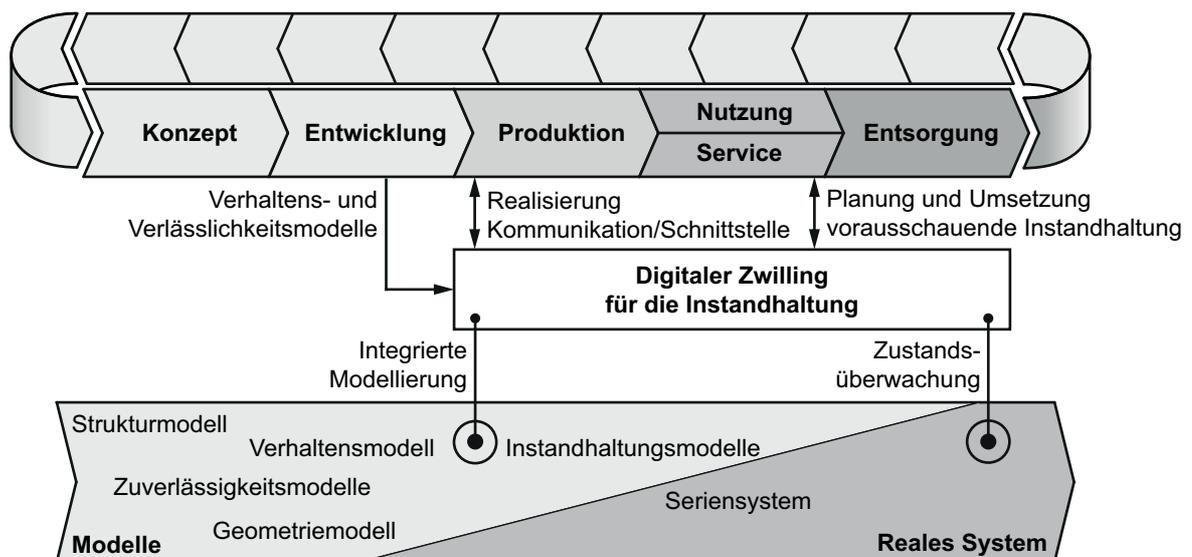


Abbildung 3.4: Digitaler Zwilling für die Unterstützung der Instandhaltung mit den adressierten Lebenszyklusphasen Produktion, Nutzung und Service (nach [BR16, WRF15])

Die Existenz des Digitalen Zwillings beginnt nach der *Produktion* des Seriensystems und wird über dessen *Nutzung* und *Service* aufrechterhalten. Der gewählten Definition eines Digitalen Zwillings folgend, ist eine Beschreibung für reale Systeme, wie Muster oder Prototypen, in der *Konzept-* und *Entwicklungsphase* ausgeschlossen. Es können jedoch Erkenntnisse und Erfahrungen aus der Umsetzung dieser als Informationsgrundlage für die Realisierung des Digitalen Zwillings im Seriensystem eingesetzt werden. Zusammen mit dem Fokus auf die Instandhaltung ist das Vorhandensein einer Schnittstelle zu realen Schädigungszuständen des Systems durch Methoden der Zustandsüberwachung notwendig und in frühen Phasen des Lebenszyklus nicht möglich. Eine Berücksichtigung des Digitalen Zwillings während der Produktion zur Abbildung individueller Ausprägungen, beispielsweise aufgrund von Fertigungstoleranzen, ist für den gewählten Zweck nicht zielführend. Individuelle Ausprägungen des Systems, insbesondere der überwachten Komponenten oder Teilsysteme, werden mittels der Zustandsüberwachung berücksichtigt. Bei der Umsetzung der Zustandsüberwachung muss sichergestellt sein, dass die Datenbasis zur Erstellung des Modells alle wesentlichen Eigenschaften und Einflüsse auf die Zuverlässigkeit des Systems während des Betriebs beinhaltet. Eine Verwendung in der *Entsorgungsphase* ist für einen digitalen Zwilling mit Fokus auf die Instandhaltung nicht vorgesehen.

Die Schnittstellen des Digitalen Zwillings zum realen mechatronischem System sind für deren Umsetzung von großer Bedeutung und werden durch die integrierte Modellierung von Zuverlässigkeit und dynamischem Verhalten sowie durch Methoden der Zustandsüberwachung realisiert. Die integrierte Modellierung liefert das grundlegende Modell des Digitalen Zwillings durch Transformation von aus der Entwicklung bekannten Systemmodellen. In dieses Systemmodell sind verschiedene Teilmodelle, wie Struktur-, Verhaltens-, Geometrie- und Zuverlässigkeitsmodelle, integriert. Das aus dieser Transformation resultierende Systemzuverlässigkeitsmodell wird mittels der Zustandsüberwachung einzelner Komponenten oder Teilsysteme des realen Systems verbunden. Die Zustandsüberwachung am realen System dient der Bestimmung aktueller Schädigungszustände von zuvor als kritisch eingestufte Komponenten oder Teilsysteme. Diese bilden die vom realen System bereitgestellten Daten auf für den Digitalen Zwilling relevante Größen bezüglich der aktuellen Schädigung ab. Die Bereitstellung dieser Informationen im Digitalen Zwilling führt zu einer über der Laufzeit aktuellen Repräsentation der Systemzuverlässigkeit.

Die Realisierung eines Digitalen Zwillings für die Instandhaltung mechatronischer Systeme bietet weitreichende Potenziale für die Anwendung. Die integrierte Modellierung ermöglicht die Weiterverwendung von Modellen der Entwicklung als Basis des Digitalen Zwillings, reduziert den Aufwand bei der Umsetzung und minimiert das Risiko von Modellierungsfehlern bei denen Wirkzusammenhänge nicht oder inkorrekt abgebildet werden. In der Nutzungsdauer eines mechatronischen Systems können Erweiterungen oder Änderungen durchgeführt werden, um beispielsweise die Funktionalität abzusichern oder zu erweitern. Dies erfordert meist eine Anpassung des Systems selbst, sodass auch der Digitale Zwilling angepasst werden muss. Werden diese Änderungen modellbasiert abgesichert, können diese Modelle über die integrierte Modellierung in den Digitalen Zwilling überführt werden. Die Überprüfung der Annahmen aus der Entwicklung zur Absicherung der Zuverlässigkeit durch die Anreicherung von realen Daten im Digitalen Zwilling, kann die integrierte Modellierung verbessern und deren Aussagen mit zunehmender Erfahrung und Datenmenge präzisieren und validieren. Der Digitale Zwilling erlaubt Rückschlüsse auf Komponenten und Teilsysteme für die keine Zustandsüberwachung vorhanden ist und ermöglicht darüber hinaus eine Diagnosefähigkeit im Falle eines Systemausfalls bei unbekannter Ursache.

3.2 Systemmodell

Das Systemmodell bildet das dynamische Verhalten des betrachteten Systems in der geforderten Modellierungstiefe ab und wird während des Entwicklungsprozesses mechatronischer Systeme, beispielsweise nach VDI 2206, aufgebaut und ausdetailliert. Es stellt die zentrale Informationsquelle für die automatisierte Ableitung des Systemzuverlässigkeitsmodells (vergleiche Abschnitt 3.5) dar. Die Architektur des betrachteten Systems wird über die Topologie des Systemmodells abgebildet und enthält alle der Modellierungstiefe entsprechenden wesentlichen funktionalen Zusammenhänge der beteiligten Komponenten und Teilsysteme. Ein domänenübergreifender Modellierungsansatz ist daher obligatorisch zur ganzheitlichen Abbildung des mechatronischen Systems (vergleiche Abschnitt 2.6). Die integrierte Modellierung von Zuverlässigkeit und dynamischem Verhalten beinhaltet eine in Abschnitt 3.5 vorgestellte automatisierte Ableitung des Systemmodells in ein Modell der Systemzuverlässigkeit und stellt daher an die Implementierung des Systemmodells verschiedene Anforderungen:

- Das Systemmodell muss alle Signale und Größen für die Bestimmung der wirkenden Belastungen auf die in der Zuverlässigkeitsanalyse zu betrachtenden Komponenten und Teilsysteme liefern.
- Die Modelldateien des Systemmodells müssen vollständig zugänglich, frei verfügbar und alle notwendigen Informationen über die Systemstruktur und -hierarchie sowie Simulationsergebnisse für die Belastungen auf einzelne Komponenten und Teilsysteme bereitstellen.
- Das Systemmodell sollte einem topologieorientierten Modellierungsansatz¹, aus welchem die Komponenten- und Teilsystemstruktur sowie die Hierarchie des Systems hervorgeht, folgen.

Ein signalflussorientierter Modellierungsansatz² ist grundsätzlich ebenfalls für die automatisierte Ableitung des Systemzuverlässigkeitsmodells geeignet. Die Zuordnung von Signalen zu Belastungen einzelner Komponenten und Teilsysteme ist jedoch deutlich erschwert und benötigt mehr Nutzereingaben.

3.3 Lokale Zuverlässigkeitsschätzer

Lokale Zuverlässigkeitsschätzer dienen dazu, die auf einzelne Komponenten oder Teilsysteme wirkende Belastung auf deren Zuverlässigkeit abzubilden. Auf Basis der Interaktion und Kombination der Zuverlässigkeitsschätzer auf lokaler Ebene wird auf Systemebene die Zuverlässigkeit des Gesamtsystems ermittelt.

Der Zusammenhang zwischen Belastung, Beanspruchung und Schädigung wird basierend auf [Wol08, KP14, San08] skizziert. Die auf Komponenten oder Teilsystemen wirkenden Belastungen sind verschiedenen Ursprungs, wie mechanische (Schwingungen, Stöße u. w.), elektrische (Stromdichte, elektrische Felder u. w.) Belastungen oder Belastungen aufgrund von Umgebungsbedingungen (Temperatur, Feuchtigkeit, UV-Strahlung u. w.). Diese Belastungen können simultan auftreten und resultieren in unterschiedlichen Beanspruchungen (mechanische Spannung und Dehnung, Variation von Materialeigenschaften und -struktur u. w.) der betrachteten Komponente oder des betrachteten Teilsystems. Beanspruchungen führen zu unterschiedlichen Schadensformen (Materialermüdung, mechanischer Verschleiß, Korrosion u. w.), die gegenüber einer Versagensgrenze oder eines Ausfallkriteriums verglichen werden, um die Lebensdauer der betrachteten Komponente oder des betrachteten Teilsystems zu bestimmen. Erreicht die Schädigung die Versagensgrenze oder ist das Versagenskriterium erfüllt, tritt das Lebensdauerende ein. Dieses Ereignis wird als Fehlerursache bezeichnet (vergleiche Abschnitt 2.1). Eine Übersicht über die in dieser Arbeit verwendeten lokalen Zuverlässigkeitsschätzer ist in Abbildung 3.5 dargestellt.

Die lokalen Zuverlässigkeitsschätzer in dieser Arbeit sind in zuverlässigkeits- sowie zustandsbasierte Verfahren unterteilt und weisen eine Steigerung der Modellierungstiefe be-

¹Topologie- beziehungsweise objektorientierte Modellierungsansätze bilden den physischen Systemaufbau ab. Dabei enthalten die Objekte die mathematische Beschreibung des Verhaltens.

²Signalflussorientierte Modellierungsansätze bilden das Systemverhalten auf Basis eines physikalischen Ersatzschaltbildes ab. Die zugrundeliegende mathematische Beschreibung wird als Blockschaltbild dargestellt. Dabei wird typischerweise die Ausgangsgröße eines Elements auf Basis gegebener Eingangsgrößen berechnet.

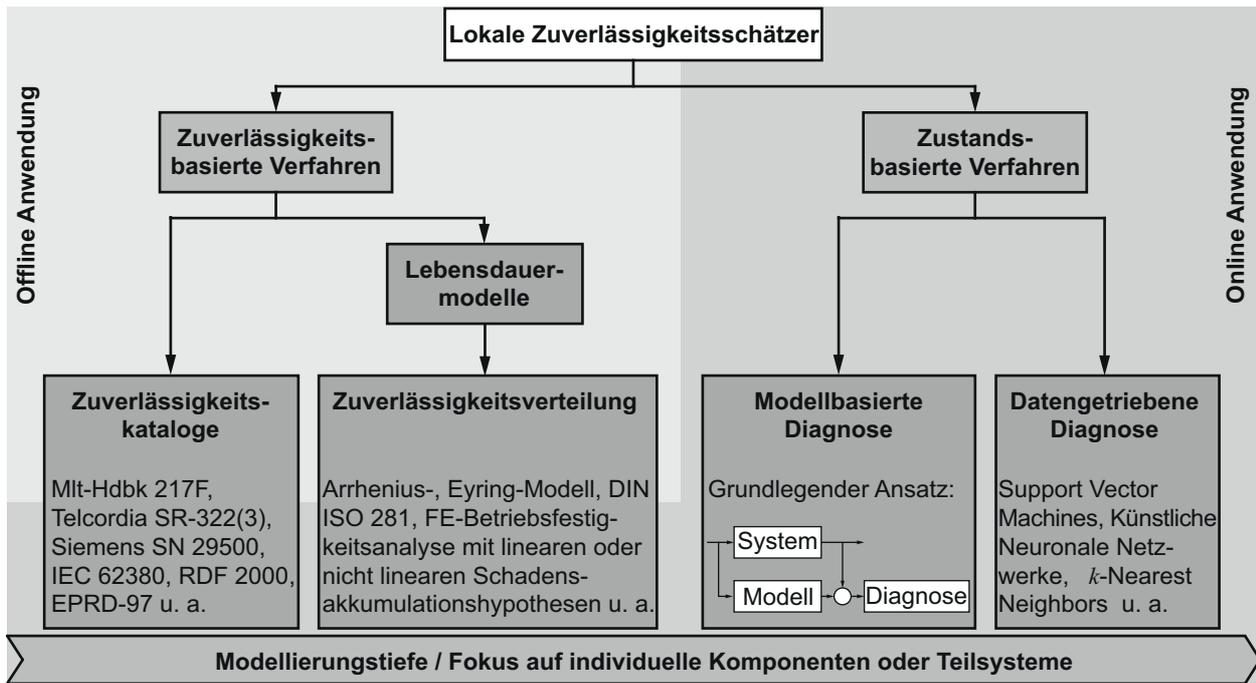


Abbildung 3.5: Übersicht über lokale Lebensdauerschätzer mit Beispielmustern und -verfahren.

ziehungsweise einen wachsenden Fokus auf individuelle Komponenten oder Teilsysteme auf. Die zuverlässigkeitsbasierten Verfahren dienen primär der Prognose der Zuverlässigkeit im Entwurf mechatronischer Systeme, während die zustandsbasierten Verfahren erst im Betrieb Anwendung finden.

Zu Beginn des Entwurfsprozesses sind noch keine konkreten Ausprägungen der einzusetzenden Komponenten oder Teilsysteme innerhalb des Systems bekannt, sodass bei einer frühzeitigen Bewertung der Systemzuverlässigkeit auf verschiedene Standards, *katalogbasierte Zuverlässigkeitsmodelle*, mit hohem Abstraktionsgrad zurückgegriffen werden muss. Es ist anzumerken, dass die Angaben zu Zuverlässigkeiten in Abhängigkeit des gewählten Standards sehr stark streuen, sodass auf möglichst eine Datenbasis zurückgegriffen werden sollte [US 95]. Darüber hinaus sind die angenommenen Ausfallraten meist deutlich höher als in der Realität [MP10]. Umfangreiche Datengrundlagen für komponenten- und teilsystembasierte Zuverlässigkeiten bieten unter anderem die folgenden Standards: Mlt-Hdbk 217F, Telcordia SR-322(3), Siemens SN-29500, IEC 62380, RDF 2000, EPRD-97 und NPRD-95. Die genannten Standards verwenden in der Regel konstante Ausfallraten (Exponential-Verteilung für Zufallsausfälle, vergleiche Abschnitt 2.1) die durch Faktoren an Produktqualität, Einsatzbedingungen u. w. angepasst werden [MP10]. Das dynamische Verhalten des zugrundeliegenden Systems geht nur kumuliert mittels der faktoriellen Berücksichtigung in die Zuverlässigkeit ein. Mit Fortschritt des Entwurfsprozesses entstehen detaillierte Modelle des dynamischen Verhaltens der Systeme und erlauben eine verhaltensbasierte Analyse der Zuverlässigkeit. Dies erfordert lokale Zuverlässigkeitsschätzer mit erhöhter Modellierungstiefe, welche die Zuverlässigkeit der betrachteten Komponente oder des Teilsystems in Abhängigkeit der wirkenden Belastung abbilden. Die Zuverlässigkeitsschätzer sind typischerweise durch zwei Modellierungsaspekte gekennzeichnet. Zunächst wird die Lebensdauer bezüglich einer Überlebens- oder Ausfallwahrscheinlichkeit

mittels *Lebensdauermodellen* berechnet³. Hier werden neben genormten Berechnungsvorschriften (DIN ISO 281 u. w.) verschiedene Lebensdauermodelle (Arrhenius, Eyring u. w.) angewandt. Eine größere Modellierungstiefe wird für Betriebsfestigkeitsanalysen auf Basis der FEM unter Verwendung verschiedener Schadensakkumulationshypothesen erreicht [San08]. Dabei kann auch auf effiziente bruchmechanische Berechnungsverfahren [Grü18] zurückgegriffen werden. Weiterhin wird die Lebensdauer von Komponenten oder Teilsystemen unter mechanischem Verschleiß auf Basis von Volumenabtragsmodellen berechnet. Dazu wird das Verschleißvolumen bis zu einer Verschleißgrenze in Abhängigkeit des verbleibenden Verschleißvorrats berechnet [KMS15a, KMS15b]. Aufbauend auf den Lebensdauermodellen wird im folgenden Schritt eine Verteilungsfunktion der Zuverlässigkeit angenommen, in die unter anderem die zuvor bestimmte Lebensdauer als Parameter eingeht. Es ist zu beachten, dass mögliche weitere Parameter der Verteilungsfunktion auch von der wirkenden Belastung abhängen (z. B. Formparameter b der Weibull-Verteilung), aber nur selten durch geeignete Modelle berücksichtigt werden.

Die zustandsbasierten Verfahren dienen der Diagnose des aktuellen Schädigungszustands der betrachteten Komponenten oder des Teilsystems und sind den Methoden der Zustandsüberwachung technischer Systeme zuzuordnen [Kim16]. Diese Methoden erfordern ein reales System zur Erzeugung der Diagnosemodelle. Daher sind die zustandsbasierten Methoden im Entwurf mechatronischer Systeme nicht anwendbar. Sowohl den modellbasierten als auch den datengetriebenen Verfahren geht typischerweise eine FMEA zur Identifizierung aller möglichen Schädigungsarten der zu überwachenden Komponente oder des zu überwachenden Teilsystems voraus. In den modellbasierten Ansätzen wird ein physikalisches oder empirisches Verhaltensmodell erstellt und mit realen Mess- oder Betriebsdaten verglichen. Das Residuum dient schließlich als Eingang in das Diagnosesystem, welches die Klassifikation des Schädigungszustands vornimmt [Kim16]. Die datengetriebenen Ansätze benötigen eine ausreichende Datenbasis zur Erstellung eines Modells zur Diagnose. Diese Datenbasis sollte alle zu klassierenden Schädigungszustände enthalten, um diese in der Anwendung identifizieren zu können [Kim16]. Die Ergebnisse der zustandsbasierten Verfahren werden als zusätzliche Systembeobachtung während des Betriebs des überwachten Systems in einen Digitalen Zwilling integriert.

Eine umfassende Übersicht über zuverlässigkeitsbasierte Verfahren zur Schätzung der Zuverlässigkeit technischer Komponenten oder Teilsysteme ist von Wolters [Wol08] und Hasbal [HB16b] erarbeitet worden. Kommentare zum Einsatzbereich und zur Anwendung katalogbasierter Zuverlässigkeitsschätzer sind in [MP10] zu finden. Eine Übersicht über zustandsbasierte Verfahren ist unter anderem von Kimotho [Kim16] verfasst worden.

3.4 Systemzuverlässigkeitsmodell

Die Bewertung der Systemzuverlässigkeit soll während des Entwurfs und des Betriebs mittels einer durchgängigen Methode unterstützt werden. Diese Methode erfordert ein Modell zur Abbildung der Abhängigkeiten zwischen Komponenten und Teilsystemen sowie ihrer lokalen Zuverlässigkeit auf die Gesamtsystemzuverlässigkeit. Es werden ausschließlich Systeme bis zu ihrem ersten Ausfall betrachtet. Hierzu wird ein Vergleich

³Exemplarisch sei die DIN ISO 281 zur Berechnung der Lebensdauer von Wälzlagern genannt. Das Ergebnis dieser Berechnung ist die Betriebsdauer nach der 10% der Gesamtheit aller betrachteten Wälzlager unter den angenommenen Bedingungen ausgefallen sind. Dies entspricht einer Zuverlässigkeit von 90%.

unterschiedlicher Methoden hinsichtlich verschiedener Bewertungskriterien aus Modellierung und Analyse der Systemzuverlässigkeit herangezogen. Im Anschluss an die Auswahl wird die ausgewählte Methode zunächst allgemein vorgestellt. Anschließend erfolgt die Übertragung auf den Modellierungsansatz zur Abbildung der Systemzuverlässigkeit. Verschiedene Ansätze zur Modellierung von Zuverlässigkeitsaspekten, wie sie in Abschnitt 2.2 vorgestellt wurden, werden aufgegriffen und ihre Umsetzung diskutiert.

3.4.1 Auswahl

Die Auswahl eines geeigneten Systemzuverlässigkeitsmodells erfolgt auf Basis der Ergebnisse von [SW15] zur Bewertung verschiedener Methoden der Analyse der Systemzuverlässigkeit. Es wurden Systemzuverlässigkeitsmodelle anhand verschiedener Bewertungskriterien aus Modellierung und Analyse, die auch in dieser Arbeit hohe Relevanz aufweisen, miteinander verglichen und bewertet. Die Bewertungskriterien aus [SW15] werden ergänzt und im Folgenden näher erläutert:

Komponenten und Teilsysteme mit mehreren Fehlzuständen: Komponenten und Teilsysteme technischer Systeme weisen in der Regel kein rein Boole'sches Ausfallverhalten (mögliche Zustände: funktionsfähig und defekt) auf, sondern sind durch mehrere Fehlzustände charakterisiert. Die Darstellung mehrerer Fehlzustände einer Komponente oder eines Teilsystems erhöht neben der Modellierungstiefe auch die Kompaktheit der Darstellung und ist somit einer zielgerichteten Modellierung dienlich.

Verschleiß- und Ermüdungsausfälle: Technische Systeme sind typischerweise Degradation unterworfen, sodass diese statistisch mittels geeigneter Verteilungsfunktionen modellierbar sind. Mechanischer Verschleiß ist mittels der Weibull-Verteilung hinreichend genau abbildbar, während bei elektronischen Bauteilen zumeist eine Exponential-Verteilung zur Modellierung von Zufallsausfällen angesetzt wird. Das Systemzuverlässigkeitsmodell soll möglichst viele unterschiedliche Verteilungsfunktionen zur Abbildung des Ausfallverhaltens einzelner Komponenten oder Teilsysteme unterstützen.

Komplexe Abhängigkeiten: Systemzuverlässigkeitsmodelle, die auf Boole'scher Algebra basieren, setzen die Unabhängigkeit von Fehlzuständen und Ausfällen zwischen Komponenten und Teilsystemen voraus. Funktionale und temporale Abhängigkeiten zwischen Komponenten und Teilsystemen treten häufig in technischen Systemen in Form von Redundanzen, Fehlerpropagierung, Fehler gemeinsamer Ursache und Lastaufteilung zwischen beteiligten Elementen auf [TTS98]. Die Abbildung dieser Abhängigkeiten ist für die Modellierungsgenauigkeit von hoher Bedeutung.

Berechnung von Minimalschnitten: Minimalschnitte oder auch kritische Pfade zeigen die Kombination gemeinsamer Fehlerursachen auf, die zu einem Ausfall des Gesamtsystems führen. Somit geben die Minimalschnitte einen Hinweis auf mögliche Schwachstellen im System, die einer genaueren Untersuchung zugeführt werden sollten.

Berechnung von Importanzkenngrößen: Die Berechnung von Importanzkenngrößen liefern einen Hinweis auf die Kritikalität von Komponenten oder Teilsystemen und ermöglicht die Identifikation von Schwachstellen im System. So können bereits im Entwurf kritische Komponenten oder Teilsysteme identifiziert und Maßnahmen zur Steigerung der Zuverlässigkeit ergriffen werden.

Berechnung der Ausfallwahrscheinlichkeiten: Das Systemzuverlässigkeitsmodell soll eine quantitative Analyse der Systemzuverlässigkeit ermöglichen, dessen Ergebnis die Systemzuverlässigkeit über der Betriebsdauer ist. Die Systemzuverlässigkeit kann für weiterführende Arbeiten verwendet werden.

Diagnose des Systemzustands: Intelligente technische Systeme sind durch eine steigende Systemkomplexität charakterisiert. Die Ursache für einen Fehlzustand oder Ausfall eines Systems zu finden ist somit schwierig und benötigt Möglichkeiten zur Diagnose der Fehlerursache.

Prognose des Systemzustands: Die Prognose zukünftiger Systemzustände erlaubt proaktives Handeln und somit die Vermeidung von Fehlzuständen oder Ausfällen des Systems durch gezielte Instandhaltungsmaßnahmen.

Berücksichtigung zusätzlicher Beobachtungen: Das Systemzuverlässigkeitsmodell ist durch Informationen aus Beobachtungen oder Messungen des Systems erweiterbar, sodass neue Systemkenntnisse in die Analyse einbezogen werden. Diese Beobachtungen sind deterministisch oder Unsicherheiten unterworfen. Deterministische Beobachtungen beschreiben den Zustand einer Komponente vollständig, während Unsicherheiten bezüglich des Zustands einer Komponente Messunsicherheiten oder die Vertraulichkeit der Messung repräsentieren.

Berechnungsaufwand: Die Abbildung der Zuverlässigkeit intelligenter technischer Systeme führt zu komplexen Modellen. Es ist daher von hoher Relevanz, effiziente Analysemethoden verfügbar zu machen, um den Berechnungsaufwand zu reduzieren.

	Fehlerbäume	Dynamische Fehlerbäume	Zuverlässigkeitsblockdiagramme	Dynamische Zuverlässigkeitsblockdiagramme	Markov-Modelle	Petri-Netze	Bayes'sche Netze	Dynamische Bayes'sche Netze
Modellierung								
Mehrere Zustände	–	–	–	–	+	+	+	+
Verschleißausfälle	+	+	+	+	–	+	+	+
Komplexe Abhängigkeit	–	+	–	+	+	+	o	+
Analyse								
Minimalschnitte	+	+	+	+	–	–	o	o
Importanzkenngrößen	+	+	+	+	–	–	o	o
Ausfallwahrscheinlichkeit	+	+	+	+	+	+	+	+
Diagnose	–	–	–	–	o	o	+	+
Prognose	o	o	o	o	–	o	+	+
Zusätzliche Beobachtungen	–	–	–	–	–	–	+	+
Berechnungsaufwand	o	–	o	–	–	–	o	o

Tabelle 3.1: Bewertung der Methoden; +: geeignet, o: bedingt geeignet, –: nicht geeignet.
Nach [SW15]

Die in Tabelle 3.1 dargestellte Bewertung der Methoden zur Modellierung und Analyse der Systemzuverlässigkeit stellt jeweils unterschiedliche Stärken und Schwächen der einzelnen Methoden heraus. Fehlerbäume und Zuverlässigkeitsblockdiagramme zählen zu den klassischen Ansätzen und haben Stärken in der Modellierung und Analyse einfacher Systeme, weisen jedoch Schwächen in diesen Aspekten bei komplexen Systemen auf. Die dynamischen Fehlerbäume und Zuverlässigkeitsblockdiagramme sind in der Modellierungstiefe sowie in der kompakten Darstellung komplexer System beschränkt und müssen zur Analyse aufwendig simuliert werden. Die Nachteile der fehlenden Modellierungstiefe und des großen Berechnungsaufwands teilen die Markov-Modelle. Diese sind auf exponentialverteilte Übergangswahrscheinlichkeiten beschränkt und eine kompakte Darstellung komplexer Systeme ist aufgrund des starken Anstiegs an abzubildenden Systemzuständen nicht gegeben. Petri-Netze haben in der kompakten Modellierung komplexer Systeme ihre Stärken, wobei diese mit hohem Rechenaufwand numerisch analysiert werden müssen. Dynamische Bayes'sche Netze erfüllen die aufgestellten Anforderungen am besten und werden für die Modellierung und Analyse der Systemzuverlässigkeit herangezogen. Diese haben gegenüber statischen Bayes'schen Netzen den Vorteil, dass sie temporale Abhängigkeiten abbilden und damit auf eine größere Modellierungstiefe zurückgegriffen wird. Darüber hinaus sind viele klassische Methoden in dynamische Bayes'sche Netze transformierbar: Fehlerbäume [BPM+01], dynamische Fehlerbäume [BD05], Zuverlässigkeitsblockdiagramme [TTS98], dynamische Zuverlässigkeitsblockdiagramme [LYM16] und unter Einschränkungen Markov-Modelle [Mur02] sowie Petri Netze [WMOS+12].

3.4.2 Bayes'sche Netze und Modellierungsaspekte

Die Annahme unabhängiger Ereignisse, wie Komponenten- oder Teilsystemausfälle, führt oft zu einer stark vereinfachten Abbildung des realen Ausfallverhaltens technischer Systeme. Dies wird deutlich, wenn die Fehlerfortpflanzung, wie in Abbildung 2.2 dargestellt, betrachtet wird. Ein Ausfall wird im Rahmen der Fortpflanzung zu einer externen Fehlerursache einer weiteren Komponente oder Teilsystems. Somit ist ein möglicher Fehlzustand oder Ausfall von einem vorgehenden Ereignis, einem Ausfall, abhängig. Dieser Zusammenhang wird über bedingte Wahrscheinlichkeiten abgebildet.

Ausgehend von dem Bayes Theorem über bedingte Wahrscheinlichkeiten, werden die Grundlagen des verwendeten Systemzuverlässigkeitsmodells erläutert. Es werden zunächst statische Bayes'sche Netze vorgestellt und verschiedene Modellierungs- und Analyseaspekte diskutiert, die für eine Verwendung als Zuverlässigkeitsmodell notwendig sind. Dynamische Bayes'sche Netze stellen eine Erweiterung zu statischen Bayes'schen Netzen dar. Die betrachtete Zeit wird diskretisiert und für jeden Zeitschritt ein Bayes'sches Netz, sogenannte Zeitscheiben, aufgebaut, wobei die unterschiedlichen Zeitscheiben über Abhängigkeiten verbunden sind. Dadurch sind die grundlegenden Modellierungs- und Analysemethoden für beide Arten Bayes'scher Netze anwendbar und werden aus Gründen der Lesbarkeit zunächst für statische Bayes'sche Netze vorgestellt.

3.4.2.1 Bayes Theorem

Sei \mathcal{B} ein von \mathcal{A} abhängiges Ereignis, sodass \mathcal{B} unter der Bedingung eintritt, wenn zuvor \mathcal{A} eingetreten ist. Dann beschreibt $P(\mathcal{B}|\mathcal{A})$ die bedingte Wahrscheinlichkeit für den Eintritt von \mathcal{B} . Mit dieser Definition ergibt sich das Gesetz der totalen Wahrscheinlichkeiten zur

Berechnung der gemeinsamen Wahrscheinlichkeit für den Eintritt der Ereignisse \mathcal{A} und \mathcal{B} zu:

$$P(\mathcal{A} \cap \mathcal{B}) = P(\mathcal{A}|\mathcal{B})P(\mathcal{B}) = P(\mathcal{B}|\mathcal{A})P(\mathcal{A}). \quad (3.1)$$

Daraus folgt direkt das Bayes Theorem für die Ereignisse \mathcal{A} und \mathcal{B} zur Berechnung bedingter Wahrscheinlichkeiten als Grundlage für Bayes'sche Netze:

$$P(\mathcal{B}|\mathcal{A}) = \frac{P(\mathcal{A}|\mathcal{B})P(\mathcal{B})}{P(\mathcal{A})}. \quad (3.2)$$

Hier werden $P(\mathcal{B}|\mathcal{A})$ und $P(\mathcal{A}|\mathcal{B})$ jeweils als a posteriori Wahrscheinlichkeit, also nach Beobachtung oder Berechnung erhaltenes Wissen, bezeichnet. Das spätere Analyseziel, die zu berechnende Systemzuverlässigkeit, stellt demnach auch eine a posteriori Wahrscheinlichkeit dar. Daneben gelten $P(\mathcal{A})$ und $P(\mathcal{B})$ als a priori Wahrscheinlichkeiten, die durch Vorwissen bekannt sind, wie Komponentenzuverlässigkeiten aus Lebensdauerversuchen oder Simulation.

Die bisher betrachteten Ereignisse \mathcal{A} und \mathcal{B} können als Elementarereignisse eines Zufallsexperiments A betrachtet werden, wenn diese nur ein Ergebnis des Zufallsexperiments beinhalten. Das Zufallsexperiment A selbst wird durch eine Funktion, auch Zufallsvariable, beschrieben, wobei zu jedem der m Ergebnisse des Experiments ein entsprechender Wert der Zufallsvariable $\{a_1, \dots, a_m\}$, im Folgenden als Zustand bezeichnet, definiert wird. Weist eine Zufallsvariable zwei Zustände auf, wird diese als binär bezeichnet. Zufallsvariablen mit mehr als zwei Zuständen werden als graduiert bezeichnet. Für die Zufallsvariablen wird im Folgenden angenommen, dass diese diskrete und sich gegenseitig ausschließende Zustände aufweisen. Somit beschreibt $A = a_j$, dass die Zufallsvariable A sich im Zustand a_j befindet. Daraus ergibt sich nach Gleichung 3.1 in Analogie zu Gleichung 3.2 für zwei Zufallsvariablen A und B das Bayes Theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \quad (3.3)$$

Die Funktion der Verteilung $P(A = a_j)$ bildet die Wahrscheinlichkeit ab, dass die Zufallsvariable A genau den Zustand a_j annimmt und es gilt:

$$P(A) = (P(A = a_1), \dots, P(A = a_m)), \quad P(A = a_j) \geq 0, \quad \sum_{j=1}^m P(A = a_j) = 1. \quad (3.4)$$

3.4.2.2 Marginalisierung

Aus dem Gesetz für die totale Wahrscheinlichkeit für Zufallsvariablen ergibt sich analog zu Gleichung 3.1

$$P(A, B) = P(A|B)P(B) = P(B|A)P(A) \quad (3.5)$$

für die gemeinsame Verteilungsfunktion $P(A, B)$. Die Berechnung der Wahrscheinlichkeitsverteilung $P(A)$ aus der gemeinsamen Verteilungsfunktion $P(A, B)$ wird als Marginalisierung bezeichnet. Hierzu werden alle Wahrscheinlichkeiten der gemeinsamen Ereignisse, in denen A zusammen mit B eintritt, mit folgender Notation aufsummiert:

$$P(A) = \sum_B P(A, B) = P(A, B = b_1) + \dots + P(A, B = b_i), \quad (3.6)$$

wobei b_i alle infrage kommenden Zustände von B beschreibt. So ist es möglich, Zufallsvariablen aus einer gemeinsamen Verteilungsfunktion durch Marginalisierung zu entfernen.

3.4.2.3 Statische Bayes'sche Netze

Ein Bayes'sches Netz bildet auf kompakte Weise die gemeinsame Wahrscheinlichkeitsverteilung über eine endliche Menge von Zufallsvariablen ab. Diese Modelle basieren auf einem gerichteten azyklischen Graphen (directed acyclic graph, DAG) bestehend aus Knoten und Kanten. Die Knoten repräsentieren eine Menge von k Zufallsvariablen $\nu = \{X_1, X_2, \dots, X_k\}$, deren bedingte Abhängigkeiten untereinander von den Kanten abgebildet werden. Jede Zufallsvariable in ν besitzt eine endliche Menge an Zuständen für welche die in Gleichung 3.4 genannten Bedingungen gelten und ist jeweils mit einer bedingten Wahrscheinlichkeitsverteilung (conditional probability distribution, CPD) $P(X_i|\text{Pa}(X_i))$ beschrieben, wobei $\text{Pa}(X_i)$ alle Elternknoten⁴ von X_i darstellt. Weist ein Knoten X_i keine Elternknoten auf, reduziert sich die CPD zu einer nicht bedingten Wahrscheinlichkeitsverteilung $P(X_i)$ und stellt somit a priori Wissen über das System ν dar. Ein Bayes'sches Netz, das für ν aufgebaut wird, repräsentiert eine eindeutige multivariate Verteilungsfunktion $P(\nu)$, gegeben durch das Produkt aller im Modell enthaltenen CPDs:

$$P(\nu) = \prod_{i=1}^k P(X_i|\text{Pa}(X_i)), \quad (3.7)$$

mit X_i für den i -ten Knoten und $\text{Pa}(X_i)$ als die Gesamtheit aller Elternknoten von X_i . Folglich beschreibt $P(X_i|\text{Pa}(X_i))$ die Wahrscheinlichkeit für das Eintreten von X_i für vorhandenes Wissen über die Elternknoten $\text{Pa}(X_i)$.

Sei die Menge von diskreten, binären Zufallsvariablen $\gamma = \{A, B, C\}$ gegeben und in dem Bayes'schen Netz in Abbildung 3.6 abgebildet. Die in γ enthaltenen Zufallsvariablen weisen aus Gründen der übersichtlichen Darstellung jeweils Boole'sche Zustände auf.

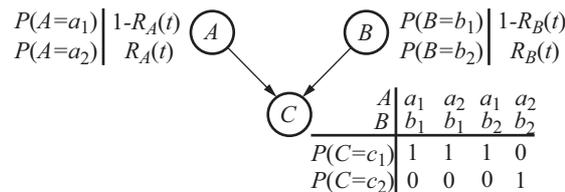


Abbildung 3.6: Einfaches Bayes'sches Netz für eine OR-Verknüpfung in Variable C von zwei Variablen A und B .

Für diskrete Zufallsvariablen in γ wird die CPD mittels bedingter Wahrscheinlichkeitstabellen (conditional probability tables, CPTs) dargestellt. In der CPT $P(C|A, B)$ für Zufallsvariable C werden die Wahrscheinlichkeiten für den Eintritt der Zustände von $C = (c_1, c_2)$ für jede mögliche Kombination von Zuständen der Elternknoten A und B zusammengefasst. Die CPTs $P(A)$ und $P(B)$ enthalten das a priori Wissen in Form der Eintrittswahrscheinlichkeiten der Verteilungsfunktionen $R_A(t)$ und $R_B(t)$. Die a posteriori Wahrscheinlichkeitsverteilung $P(C)$ wird nach Gleichung 3.6 aus der gemeinsamen Verteilungsfunktion $P(\gamma)$ marginalisiert und stellt in dieser Arbeit ein typisches Analyseziel

⁴Als Elternknoten $\text{Pa}(X_i)$ werden alle Knoten bezeichnet, die eine Kante in Richtung eines betrachteten Knotens X_i aufweisen.

eines Bayes'schen Netzes dar. Eine solche Berechnung ist exemplarisch in Abschnitt A.1 durchgeführt.

3.4.2.4 Dynamische Bayes'sche Netze

Dynamische Bayes'sche Netze stellen eine Erweiterung von statischen Bayes'schen Netzen um temporale Abhängigkeiten dar. Diese Erweiterung wird zur Modellierung zeitdiskreter stochastischer Prozesse verwendet. Die Diskretisierung der Betrachtungszeit erfolgt mittels Zeitscheiben. Zu jedem Betrachtungszeitpunkt wird ein Bayes'sches Netz des Systems aufgebaut, das über temporale Kanten (gestrichelt) mit der nächsten Zeitscheibe verbunden ist. Eine schematische Darstellung des in Abbildung 3.6 vorgestellten Systems ist in Abbildung 3.7 gezeigt, wobei das System um $n+1$ Zeitscheiben zur Abbildung der diskretisierten Betrachtungszeit t_0, \dots, t_n ergänzt wird.

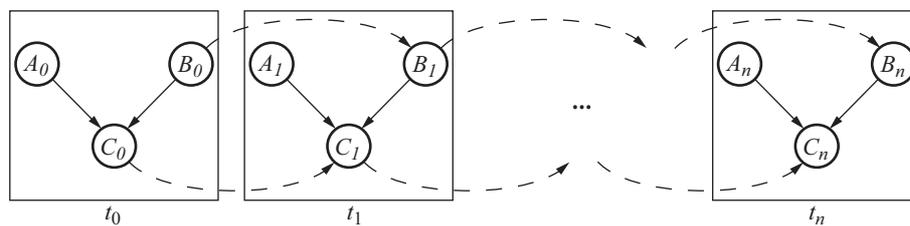


Abbildung 3.7: Schematische Darstellung eines dynamisches Bayes'sches Netz, in dem ein zeitabhängiges System in Zeitscheiben zu unterschiedlichen Zeitpunkten t_0, \dots, t_n abgebildet wird.

Unter der Annahme, dass die Parameter der CPDs der betrachteten Zufallsvariablen zeitinvariant (Markov-Eigenschaft) und der DAG des Systems ebenfalls zeitinvariant ist, wird durch „Aufrollen“ der Zeitscheiben eine kompaktere Darstellung als in Abbildung 3.7 gewählt [Mur02]. Demnach ist durch die Darstellung von zwei Zeitscheiben t_i und t_{i+1} eine vollständige Abbildung dynamischer Bayes'scher Netze, wie in Abbildung 3.8, gegeben.

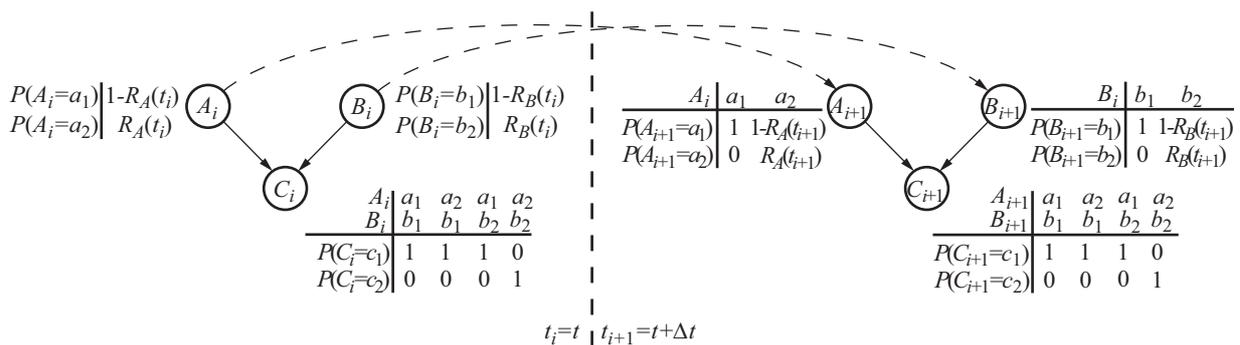


Abbildung 3.8: Einfaches dynamisches Bayes'sches Netz für das System aus Abbildung 3.6 in der Darstellung von zwei Zeitschritten t_i und t_{i+1} .

Aus der Markov-Eigenschaft folgt, dass die CPDs nur Exponentialverteilungsfunktionen (vergleiche Gleichung 2.4 und Abschnitt 2.1) aufweisen dürfen, da diese nur von der Diskretisierungsschrittweite Δt und nicht von der absoluten Betrachtungszeit t_i abhängen:

$$P(A_{t+\Delta t}|A_t) = \frac{P(A_{t+\Delta t} \cap A_t)}{P(A_t)} = \frac{P(A_{t+\Delta t})}{P(A_t)} \Rightarrow \frac{e^{-\lambda(t+\Delta t)}}{e^{-\lambda t}} = e^{-\lambda \Delta t}. \quad (3.8)$$

Dies stellt jedoch eine starke Einschränkung der Modellierungstiefe hinsichtlich der Abbildung des Ausfallverhaltens technischer Systeme dar. So können Verschleiß- und Ermüdungsausfälle nicht ausreichend detailliert abgebildet werden, da diese typischerweise über die Weibull-Verteilung darzustellen sind.

Für die Modellierung von Zufallsvariablen mit zeitvarianten CPDs, wie der Weibull-Verteilung, werden dem Modell in jeder Zeitscheibe eine entsprechende Zufallsvariable hinzugefügt [Mur02], wobei dies wieder der Darstellung in Abbildung 3.7 entspräche. Um dennoch die kompakte grafische Darstellung in zwei Zeitscheiben zu verwenden, wird folgende Notation verwendet: Eine Zufallsvariable Z , die aufgrund zeitvarianter CPDs in jeder i -ten Zeitscheibe auftaucht, wird als Z_i dargestellt. Basierend auf [Mur02], bildet die Zeitscheibe t_i das System zu Beginn des Betrachtungszeitraums ab und die Zeitscheibe t_{i+1} das System zu einem beliebigen Betrachtungszeitpunkt. Temporale Abhängigkeiten, also Kanten, die beide Zeitscheiben miteinander verbinden, werden gestrichelt dargestellt. Die Einführung der Notation der bedingten Wahrscheinlichkeiten in den Zeitscheiben erfolgt in den Abschnitten 3.4.3.1, 3.4.3.3 und 3.5.2.5.

Sei die bekannte Menge von m diskreten Zufallsvariablen $\nu = \{X_1, X_2, \dots, X_m\}$ auf eine zeitdiskrete Darstellung über $i \in \{0, \dots, n\}$ Zeitscheiben erweitert, sodass sich zu jedem Zeitschritt i die Menge an Zufallsvariablen $\nu_i = \{X_i^1, \dots, X_i^m\}$ ergibt. Daraus folgt nach Gleichung 3.7 die gemeinsame Verteilungsfunktion $P(\nu_i)$ über die Zeitscheiben i mit:

$$P(\nu_{1:n}) = \prod_{i=0}^n \prod_{j=1}^m P(X_i^j \mid \text{Pa}(X_i^j)). \quad (3.9)$$

Hier beschreibt X_i^j die j -te Zufallsvariable in der i -ten Zeitscheibe, wobei $\text{Pa}(X_i^j)$ wieder die Elternknoten beschreibt. Dabei sind die Elternknoten der Zufallsvariable X_i^j sowohl in derselben Zeitscheibe i als auch in beliebigen vorangegangenen Zeitscheiben zu finden. In der vorliegenden Arbeit werden jedoch nur temporale Kanten zwischen zwei benachbarten Zeitscheiben t_i und t_{i+1} betrachtet.

Ähnlich wie in den statischen Bayes'schen Netzen am Beispiel des Systems γ , stellt die über $i \in \{0, \dots, n\}$ Zeitscheiben abgebildete a posteriori Wahrscheinlichkeitsverteilung $P(C_i)$, aus der gemeinsamen Verteilungsfunktion $P(A_i, B_i, C_i)$ marginalisiert, in dieser Arbeit ein typisches Analyseziel eines dynamischen Bayes'schen Netzes dar. Das Ergebnis ist folglich ein zeitdiskreter Verlauf der marginalisierten Wahrscheinlichkeitsverteilung $P(C_i)$.

3.4.2.5 Bedingte Unabhängigkeit und d -Separation

Die Prüfung der bedingten Unabhängigkeit von Zufallsvariablen beziehungsweise Variablenmengen stellt für Bayes'sche Netze eine zentrale Aufgabe dar und bildet die Basis für verschiedene Anwendungen, wie der Inferenz. Der DAG eines Bayes'schen Netzes bildet kausale Abhängigkeiten zwischen den im Netz abgebildeten Zufallsvariablen ab. Daher genügt es zur Ermittlung der bedingten Unabhängigkeit, den DAG des Netzes zu betrachten. Seien zwei Zufallsvariablen U und V in einem Bayes'schen Netz bedingt unabhängig im Bezug auf eine weitere Zufallsvariable X , es wird also keine Information von U zu V unter Beteiligung von X und anders herum übertragen, sind diese d -separiert. Eingängige Definitionen für die d -Separation werden von Pearl [Pea87] und Jensen [JN09] geliefert und sollen anhand der in Abbildung 3.9 dargestellten DAGs erläutert werden.

Auf Basis der in Abbildung 3.9 gezeigten Verknüpfungen müssen nach der Definition von Jensen einer der zwei folgenden Aspekte für die d -Separation gelten:

- die Verknüpfung ist seriell oder divergierend und es liegt Evidenz⁵ für X vor,
- die Verknüpfung ist konvergierend und weder für X noch für eine nachfolgende Variable liegt Evidenz vor.

Der erste Aspekt der Definition setzt voraus, dass eine Evidenz für die Zufallsvariable X vorliegt. Für diesen Fall und serielle oder divergierende Verknüpfungen (vergleiche Abbildung 3.9 b) und c)) ist V d -separiert von U , da der Informationsfluss durch den bekannten Wert der Variable X blockiert wird. Bezüglich des zweiten Aspektes der Definition sind die Zufallsvariablen U und V d -separiert, da der Pfad von U zu V an X blockiert ist. Ist ein DAG konvergierend (vergleiche Abbildung 3.9 a)), fließt keine Information von U zu V durch diesen Pfad. Daher sind U und V d -separiert.

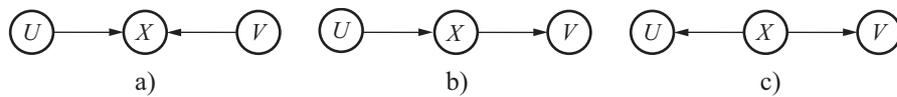


Abbildung 3.9: Grundlegende Verknüpfungen innerhalb Bayes'scher Netze: konvergierend a); seriell b); divergierend c).

3.4.2.6 Disjunkte Verknüpfungen

Die Abhängigkeiten zwischen Zufallsvariablen in Bayes'schen Netzen werden über die CPTs quantifiziert. Die Anzahl an Einträgen der CPTs, somit auch die Anzahl an zu bestimmenden Parametern, steigt exponentiell mit der Anzahl an Elternknoten und Zuständen, der zu berücksichtigenden Zufallsvariablen. Ein verbreiteter Ansatz zur Reduzierung der zu bestimmenden Parameter, ist die Verwendung disjunkter Verknüpfungen mit Unsicherheiten über die Wechselwirkung der Zufallsvariablen. Aus diesen Verknüpfungen ergibt sich ein linearer Zusammenhang zwischen zu bestimmenden Einträgen der CPT und der Anzahl der Elternknoten. Die im Folgenden vorgestellten und im Systemzuverlässigkeitsmodell verwendeten disjunkten Verknüpfungen mit Unsicherheiten (noisy-OR) und deren Verallgemeinerung (noisy-MAX) werden diskutiert.

In Abbildung 3.6 ist eine deterministische OR-Verknüpfung der Variablen A und B über die Variable C modelliert. Die Modellierung der Wechselwirkung zwischen den Zufallsvariablen A und B auf die Zufallsvariable C mittels disjunkter Verknüpfung setzt voraus, dass jede der betrachteten Ursachen, A und B , einen Effekt auf C besitzen. Es dürfen darüber hinaus keine signifikanten Wechselwirkung zwischen den Ursachen, A und B , auftreten, sodass diese unabhängig Einfluss auf C ausüben [Pea87].

Noisy-OR-Verknüpfungen stellen probabilistische Erweiterungen zu deterministischen OR-Verknüpfungen von binären Zufallsvariablen dar, indem Unsicherheiten berücksichtigt werden. Darüber hinaus legen diese die Annahme zugrunde, dass es in jeder Ursache einen Zustand gibt, der eine kausale Auswirkung auf den Effekt bewirkt (sogenannter distinguished state). Ein übersichtlicher Ansatz zur Implementierung ist nach [Zag10] in Abbildung 3.10 als Erweiterung für das bekannte System aus Abbildung 3.6 dargestellt. Basierend auf der deterministischen OR-Verknüpfung in der CPT der Zufallsvariable

⁵Der Zustand der Zufallsvariable X ist aufgrund von Beobachtung oder Messung mit Bestimmtheit bekannt. Vergleiche auch Abschnitt 3.4.2.8.

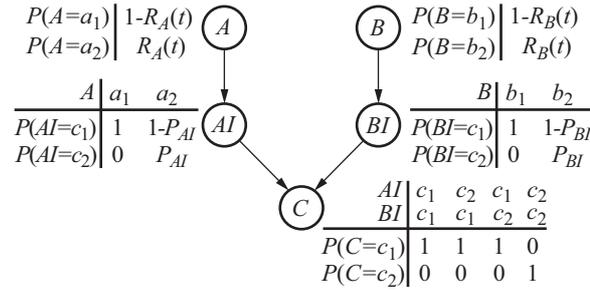


Abbildung 3.10: Erweiterung des Bayes'sches Netzes in Abbildung 3.6 zu einer Noisy-OR-Verknüpfung durch Hinzufügen von Inhibitorzufallsvariablen AI und BI .

C und den bekannten Ursachen A und B , werden sogenannte Inhibitorenzufallsvariablen AI und BI hinzugefügt. Die Inhibitoren weisen die gleichen Zustände c_1 und c_2 auf wie die Zufallsvariable C und bilden jeweils die Unsicherheit der Wechselwirkungen zwischen A und B auf C mittels der Wahrscheinlichkeiten P_{AI} und P_{BI} ab. Dabei gilt $P(AI = c_1 | A = a_1) = 1$ und $P(BI = c_1 | B = b_1) = 1$, sodass jeweils nur für ein Ereignis $A = a_1$ und $B = b_1$ die Unsicherheit für den Eintritt von $C = c_2$ maßgeblich ist [Zag10]. In diesem Beispiel sind $A = a_1$ und $B = b_1$ die Zustände der jeweiligen Ursache mit deterministischem Effekt (distinguished state). Allgemein gilt, dass wenn die Ursache sich in diesem frei wählbaren Zustand befindet, die hervorgerufene Auswirkung auf einen Effekt immer deterministisch ist. Die Modellierung von Noisy-OR-Verknüpfungen ist auf beliebig, aber endlich viele Ursachen erweiterbar. Die Anzahl der zu bestimmenden Parameter steigt in gleichem Maße mit der Anzahl an Ursachen.

Eine Erweiterung von Noisy-OR-Verknüpfungen auf gradierte Zufallsvariablen stellen sogenannte Noisy-MAX-Verknüpfungen dar [Hen89, Die93]. Am Beispiel aus Abbildung 3.10 bedeutet dies, dass die Zufallsvariablen A , B und C jeweils $n_{a/b/c} \geq 2$ unterschiedliche Zustände besitzen, wobei den Zuständen des Effekts C jeweils eine sogenannte Intensität zugeordnet ist. Aus der Intensität ergibt sich eine Reihenfolge der Zustände, die beispielsweise als zunehmende Degradation bis zum Ausfall der Komponente oder des Teilsystems C interpretiert wird. Zur Beschreibung der Wahrscheinlichkeiten in den CPTs der Inhibitorzufallsvariablen AI und BI wird die (uvw) -Notation eingeführt [ZD06]. Hierbei beschreibt u einen Index der Ursachen A und B , v nimmt jeweils auf den v -ten Zustand der Ursachen A und B Bezug und w weist auf den w -ten Zustand des Effekts C hin. Für die CPTs der Inhibitoren AI und BI gelten nach [ZD06] folgende Voraussetzung: Die Zustände der Inhibitoren AI und BI sind gleich der Zustände des Effekts C – analog zu den bereits diskutierten Noisy-OR-Verknüpfungen. Zusätzlich gelten folgende Bedingungen für die Einträge P_{uvw} der jeweiligen CPT:

$$P_{uvw} = \begin{cases} 1, & \text{für } v = 1; \quad w = 1, \\ 0, & \text{für } v = 1; \quad w \neq 1, \\ p \in [0, 1], & \text{für } v \neq 1. \end{cases} \quad (3.10)$$

Hierbei werden die Einträge der CPTs p_{uvw} auch als Noisy-MAX-Parameter bezeichnet. Die CPT der Ursache C stellt dabei nach wie vor eine deterministische OR-Verknüpfung dar.

Jedes Modell eines technischen Systems bildet die zu modellierenden Aspekte bezüglich

der vorgegebenen Modellierungstiefe genau ab, sodass meist nur das dominierende Verhalten betrachtet wird. Demnach gibt es immer Ursachen eines Effekts, die nicht explizit im Bayes'schen Netz abgebildet sind, der sogenannte *leak* [Hen89, Die93]. Der *leak* wird als zusätzliche Ursache Z_{leak} der Noisy-MAX-Verknüpfung hinzugefügt und fasst alle nicht explizit modellierten Ursachen zusammen. Dem bisherigen Modellierungsansatz folgend wird der Ursache Z_{leak} ein Inhibitor ZI_{leak} zugeordnet [ZD06]. Die CPT des Inhibitors ZI_{leak} beinhaltet somit Wahrscheinlichkeiten zur Abbildung des möglichen Einflusses nicht betrachteter Ursachen auf den Effekt C .

Viele der verfügbaren Modellierungs- und Analyseprogramme bilden disjunkte Verknüpfungen als ein Merkmal eines Knotens ab, sodass eine Abbildung der in Abbildung 3.10 gezeigten Struktur entfällt und lediglich die Noisy-OR- beziehungsweise Noisy-MAX-Parameter (vergleiche P_{AI} und P_{BI} in Abbildung 3.11) eingegeben werden müssen. Diese Reduzierung beruht auf der Vernachlässigung der Darstellung der Zustände mit deterministischem Effekt (hier a_1 und b_1). Die kompakte Darstellung in Abbildung 3.11 ist äquivalent zu Abbildung 3.10. Nach Henrion [Hen89] werden in Noisy-OR- beziehungsweise Noisy-MAX-Verknüpfungen nur die Ursachen A und B betrachtet, die für sich allein einen Einfluss auf den Effekt C aufweisen. Alle weiteren Kombinationen werden in der Darstellung der CPT vernachlässigt. Des Weiteren ist die CPT um einen *leak*-Anteil erweitert. Die Darstellung von CPTs einer Noisy-OR- beziehungsweise Noisy-MAX-Verknüpfung erfolgt im weiteren Verlauf der Arbeit, wie in Abbildung 3.11 gezeigt, wobei die Verwendung von *leak* als Repräsentation von Unsicherheit optional ist.

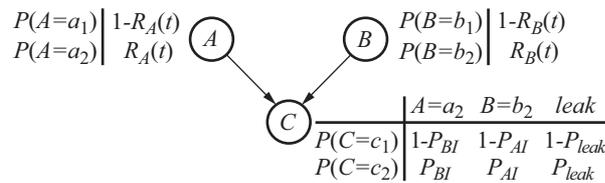


Abbildung 3.11: Kompakte Darstellung des Bayes'schen Netzes aus Abbildung 3.10 zu einer Noisy-OR-Verknüpfung in Zufallsvariable C mit *leak*.

3.4.2.7 Konjunkte Verknüpfungen

Die Modellierung konjunkter Verknüpfungen in Bayes'schen Netzen stellt die gleichen Anforderungen an die beteiligten Zufallsvariablen wie es bei disjunkten Verknüpfungen der Fall ist (vergleiche Abschnitt 3.4.2.6). In Abbildung 3.12 ist eine deterministische AND-Verknüpfung der Variablen A und B (Ursachen) in der Variablen C (Effekt) modelliert. Diese Verknüpfung zeigt sich über die binären Einträge in der CPT der Variable C .

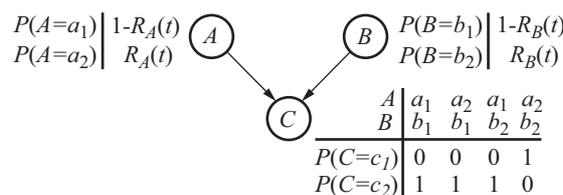


Abbildung 3.12: Einfaches Bayes'sches Netz für eine AND-Verknüpfung in Variable C von zwei Variablen A und B .

Ähnlich wie bei disjunkten Verknüpfungen existiert auch für konjunkte Verknüpfungen eine probabilistische Erweiterung, die Noisy-AND-Verknüpfung. Die Noisy-AND-Verknüpfungen und deren zusätzliche Erweiterung auf gradierte Zufallsvariablen, die Noisy-MIN-Verknüpfungen, stellen komplementäre Modelle zu den bereits diskutierten Noisy-OR- und Noisy-MAX-Verknüpfungen dar [Zag10]. Die Modellierung von Noisy-AND-Verknüpfung ist sehr ähnlich zu der von Noisy-OR-Verknüpfungen. In dem in Abbildung 3.10 gezeigten Ansatz muss zur Abbildung einer Noisy-AND-Verknüpfung die deterministische OR-Verknüpfung in Variable C durch eine deterministische AND-Verknüpfung ersetzt werden (vergleiche Variable C in Abbildung 3.12). Dies ist gleichbedeutend mit einem Tausch des Zustands mit deterministischem Effekt (distinguished state, vergleiche CPTs in Abbildung 3.13 für $P(C|A=a_1)$ und $P(C|B=b_1)$). Folglich ist eine Noisy-AND-Verknüpfung analog zu einer Noisy-OR-Verknüpfung abbildbar. Dieser Zusammenhang ist darüber hinaus auch auf die Erweiterungen Noisy-MIN- und Noisy-MAX-Verknüpfungen übertragbar [Zag10].

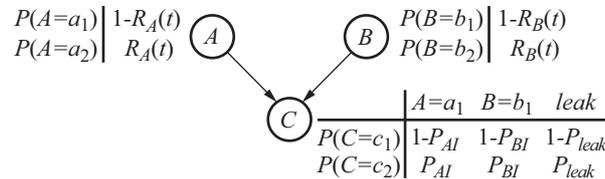


Abbildung 3.13: Kompakte Darstellung des Bayes'schen Netzes aus Abbildung 3.12 zu einer Noisy-AND-Verknüpfung in Zufallsvariable C mit *leak*.

3.4.2.8 Evidenzen

Das Systemwissen, welches in einem Bayes'schen Netz abgebildet wird, kann durch Beobachtungen oder Messungen des Systems zur Laufzeit aktualisiert werden. Die Beobachtungen eines Systems werden dabei als *Evidenzen* bezeichnet und drücken aus, dass der Zustand einer oder mehrerer Zufallsvariablen vollständig oder teilweise bekannt ist. Diese Beobachtungen sind in deterministische (*hard*) und unsichere Evidenzen eingeteilt [Pea87, VKV02]. Dabei entspricht eine deterministische Evidenz einer vollständigen Beobachtung einer Zufallsvariable. Deterministische Evidenzen weisen Zufallsvariablen auf Basis von Beobachtungen mit Bestimmtheit einen Zustand zu. Unsichere Evidenzen werden in virtuelle (*virtual*) und weiche (*soft*) Beobachtungen unterteilt. Die virtuelle Evidenz bezieht sich auf die Unsicherheit bezüglich der Konfidenz über eine Beobachtung („Wie sehr vertraue ich der Messung?“) und wird über ein Verhältnis von Wahrscheinlichkeiten (*likelihood ratio*) angegeben. Die weiche Evidenz bezieht sich auf quantifizierbare Größen, wie die Messunsicherheit, und wird mittels einer Wahrscheinlichkeitsverteilung über die Zustände der betrachteten Zufallsvariable abgebildet.

Die harte Evidenz wird als Sonderfall der weichen Evidenz betrachtet. Unter der Voraussetzung einer binären Zufallsvariable $X = (x_1, x_2)$ wird aus einer weichen Evidenz mit der Wahrscheinlichkeitsverteilung $E(X=x_1) = p_E$ für $p_E = 1 \vee p_E = 0$ eine harte Evidenz. In einer weichen Evidenz gilt $p_E \in]0, 1[$.

Im Rahmen dieser Arbeit werden weiche Evidenzen in Abschnitt 4.3 für die Modellierung von Messunsicherheiten für Zufallsvariablen verwendet und daher im Folgenden detaillierter vorgestellt. Aus Gründen der übersichtlichen Darstellung werden statische

Bayes'sche Netze betrachtet, das Vorgehen ist jedoch auf dynamische Bayes'sche Netze übertragbar.

Betrachtet wird das Bayes'sche Netz γ aus Abbildung 3.6, das eine konvergierende Verknüpfung darstellt. Daraus folgt nach der Definition von Jensen (vergleiche Abschnitt 3.4.2.5), dass die Zufallsvariablen A und B d -separiert sind. Daher soll zur Darstellung des Vorgehens zur Berücksichtigung von weichen Evidenzen zunächst nur die Untermenge von Zufallsvariablen $\{A, C\}$ betrachtet werden. Mittels Gleichung 3.7 wird anschließend die gemeinsame Verteilung für das gesamte Bayes'sche Netz $\gamma = \{A, B, C\}$ berechnet.

Aus Messungen am realen System ist die Wahrscheinlichkeitsverteilung $E(A)$ über die Zustände (a_1, a_2) der Zufallsvariable A bekannt. Die Wahrscheinlichkeitsverteilung $E(A)$ resultiere aus einer Messunsicherheit und soll als weiche Evidenz für die weitere Analyse des Bayes'schen Netzes berücksichtigt werden. Die Analyse des Netzes über Inferenz auf Basis des Bayes Theorems 3.3 ist hier nicht direkt anwendbar. Das Bayes Theorem setzt voraus, dass der Zustand einer Zufallsvariable exakt bekannt ist, also ein Ereignis eingetreten ist. Die weiche Evidenz beschreibt mittels der Wahrscheinlichkeitsverteilung $E(A)$ die Unsicherheit über den exakten Zustand von A . Für die Analyse des Bayes'schen Netzes $\{A, C\}$ mit der weichen Evidenz $E(A)$ wird daher der Satz von Jeffrey (*Jeffrey's Rule*) [Jef83] angewandt:

$$P(C) = \sum_A P(C|A)E(A). \quad (3.11)$$

Die Anwendbarkeit des Satzes von Jeffrey muss für das vorliegende Problem geprüft werden, um sicherzustellen, dass die Beobachtung einer Zufallsvariable unabhängig vom Inferenzziel ist. Als Voraussetzung für die Anwendbarkeit wird nach Valtorta [VKV02] eine virtuelle Zufallsvariable A_E mit A als einzigem Elternknoten dem Netz hinzugefügt (vergleiche Abbildung 3.14). Diese Zufallsvariable wird als virtuell bezeichnet, da sie der Implementierung von Evidenzen dient und keine Repräsentation im realen System besitzt. Anschließend muss geprüft werden, ob die virtuelle Zufallsvariable A_E d -separiert von der Zielvariable C unter Berücksichtigung einer harten Evidenz von A ist. Nach der Definition von Jensen stellt das vorliegende Bayes'sche Netz aus $\{A_E, A, C\}$ eine divergierende Verknüpfung dar, die durch die Evidenz in A d -separiert ist (vergleiche Abschnitt 3.4.2.5). Der Satz von Jeffrey 3.11 ist demnach anwendbar und die weiche Evidenz im Bayes'schen Netz zu berücksichtigen.

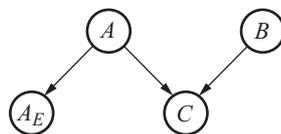


Abbildung 3.14: Erweiterung des Bayes'sches Netzes in Abbildung 3.6 um die virtuelle Variable A_E .

Viele der verfügbaren Modellierungs- und Analyseprogramme bilden virtuelle und insbesondere weiche Evidenzen als ein Merkmal einer Zufallsvariablen ab, sodass nur die Parameter der Evidenz eingegeben werden müssen, nicht jedoch das Bayes'sche Netz händisch um ein virtuelle Variable erweitert und die Unabhängigkeit der Variablen geprüft werden muss.

3.4.2.9 Inferenz

Das Ziel der Analyse von Bayes'schen Netzen liegt in der Bestimmung der aus der gemeinsamen Wahrscheinlichkeitsverteilung marginalisierten Wahrscheinlichkeiten, d. h. die Bestimmung der Wahrscheinlichkeit mit der sich eine Zufallsvariable in einem Zustand befindet (vergleiche Abschnitt A.1). Für die Diagnose ist der Rückschluss auf eine mögliche Ursache bei einer eingetretenen oder beobachteten Folge $P(\text{Ursache}|\text{Folge})$ das Ziel, während die Prognose für die Berechnung der Wahrscheinlichkeit einer möglichen Folge für eine gegebene Ursache $P(\text{Folge}|\text{Ursache})$ eingesetzt wird.

Zur Analyse Bayes'scher Netze werden Inferenz-Algorithmen eingesetzt, die in exakte Verfahren sowie in deterministische und stochastische Näherungsverfahren unterteilt werden. Verschiedene Inferenz-Algorithmen und detaillierte Beschreibungen sind in [Pea87, JN09, Mur02] zu finden. Die Anwendbarkeit der verschiedenen Verfahren ist von der Komplexität des zu analysierenden Bayes'schen Netzes abhängig. So sind Bayes'sche Netze, die als Polytrees entworfen sind, d. h. es darf höchstens einen Weg zwischen zwei Knoten geben, recheneffizient mit exakten Verfahren lösbar. Exakte Verfahren liefern genaue Lösungen für die Inferenz, sind allerdings für steigende Systemkomplexität rechenintensiv oder nicht mehr anwendbar. Der Vorteil von Näherungsverfahren liegt in der Anwendbarkeit auf komplexe Systeme, sodass insbesondere stochastische Verfahren eine Lösung liefern. Dies geht jedoch zu Lasten einer im Vergleich typischerweise höheren Berechnungsdauer.

Die im Rahmen dieser Arbeit erzeugten dynamischen Bayes'schen Netze bilden das Ausfallverhalten komplexer technischer Systeme ab. Diese technischen Systeme sind durch unterschiedlichste Wahrscheinlichkeitsverteilungen zur Abbildung des Ausfallverhaltens, einer hohen Anzahl an Komponenten sowie Teilsystemen und Abhängigkeiten zwischen diesen gekennzeichnet, sodass die resultierenden dynamischen Bayes'schen Netze eine hohe Komplexität bezüglich der Anzahl an Zufallsvariablen als auch der Anzahl ihrer Abhängigkeiten aufweisen. Daher werden Näherungsverfahren problemspezifisch ausgewählt und angewandt.

3.4.3 Modellierung der Systemzuverlässigkeit

Die Beeinträchtigungen der Verlässlichkeit und insbesondere die Zuverlässigkeit technischer Systeme, ihrer Teilsysteme und Komponenten (vergleiche Abbildung 2.2), stellen die zentralen Aspekte des Modellierungsansatzes dieser Arbeit dar. In der Literatur sind verschiedene Modellierungsansätze für Bayes'sche Netze zur Abbildung der Zuverlässigkeit technischer Systeme zu finden [Hen89, TTS98, BPM+01, SMW06, LP07, GB16]. Verschiedene Aspekte dieser Ansätze werden für die Entwicklung des Modellierungsansatzes aufgegriffen und erweitert.

Der hier vorgestellte Modellierungsansatz stellt ein, auf der Definition der Verlässlichkeit technischer Systeme systematisch aufbauendes, Vorgehen dar. Dieses beinhaltet die Entwicklung eines verallgemeinerten Modellierungsansatzes, die systematische und strukturelle Reduzierung der Modellkomplexität sowie die Erweiterung des Modells um zusätzliche Modellierungsaspekte. Dabei findet die strukturelle Reduzierung des Netzes auf Graphenebene statt und nimmt keinen Einfluss auf die Modellierungstiefe und -umfang. Der Entwurf des Modellierungsansatzes ist in Abbildung 3.15 schematisch dargestellt.

Ausgehend von den Beeinträchtigungen der Verlässlichkeit werden diese für jede zu berücksichtigende Komponente oder jedes zu berücksichtigende Teilsystem des zu model-

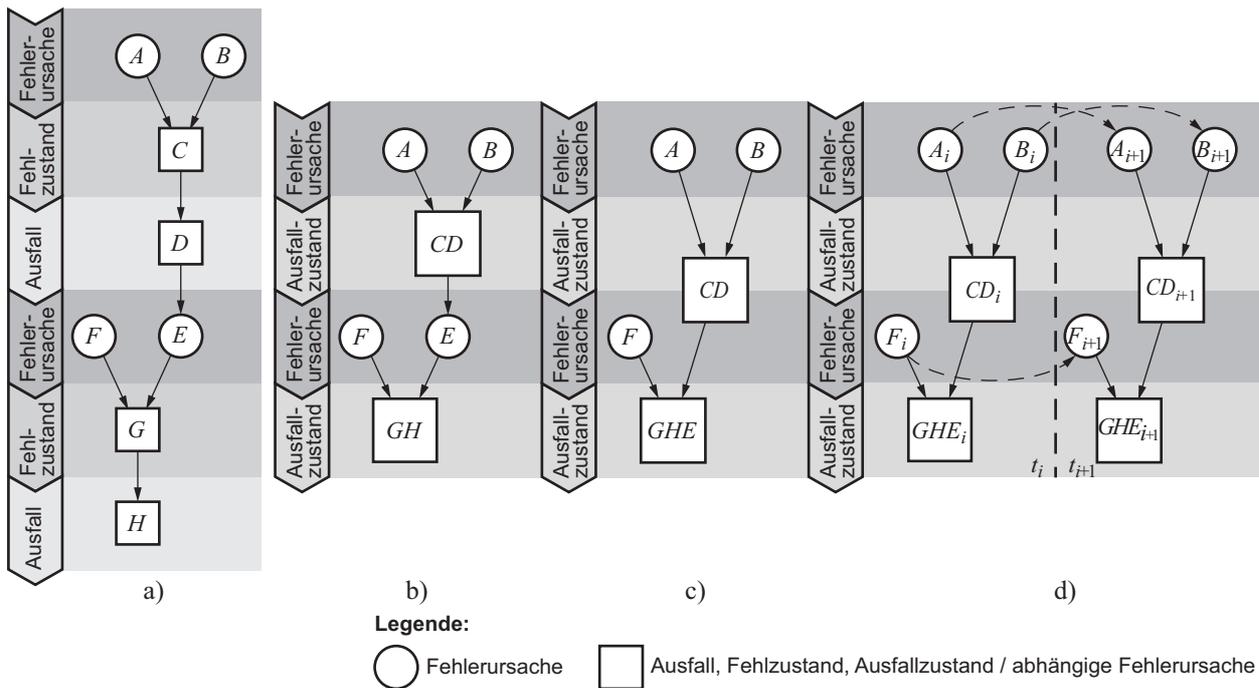


Abbildung 3.15: Entwurf eines Modellierungsansatzes zur Abbildung der Beeinträchtigungen der Verlässlichkeit nach Abbildung 2.2 mittels dynamischer Bayes'scher Netze: a) *Modellierungsansatz* mit statischen Bayes'schen Netzen; b) *Zusammenfassung* von Fehlzuständen C, G und Ausfällen D, H zu Ausfallzuständen CD und GH ; c) *Zusammenfassung* von externen Fehlerursachen durch Fortpflanzung von Ausfallzuständen CD in CDE ; d) *Erweiterung* des reduzierten Modells um eine Diskretisierung der Betriebsdauer zu dynamischen Bayes'schen Netzen.

lierenden technischen Systems aufgebaut. Modellierungsziel ist es, die Auswirkungen einzelner Fehlerursachen über die Abhängigkeiten zwischen den Komponenten und Teilsystemen bis auf Systemebene zu propagieren, um die Systemzuverlässigkeit abzubilden.

Jede Komponente beziehungsweise jedes Teilsystem wird zunächst durch Zufallsvariablen entsprechend ihrer jeweiligen Fehlerursache, ihres jeweiligen Fehlzustands und Ausfalls repräsentiert (vergleiche Abbildung 3.15 a)). Eine Übersicht über grundlegende Modellierungsaspekte für die Beeinträchtigungen ist in Tabelle 3.2 dargestellt. Demnach werden Fehlerursachen jeweils mittels einer Zufallsvariablen abgebildet, was die Unabhängigkeit der Fehlerursachen voneinander unterstreicht. Eine Fehlerursache wird jeweils als binäre Zufallsvariable modelliert. Eine Ursache bildet ein Ereignis ab, das ausschließlich bereits (*active*) oder noch nicht (*dormant*) eingetreten ist. Es wird zwischen internen und externen Fehlerursachen unterschieden (vergleiche Abschnitt 2.1). Die Eintrittswahrscheinlichkeiten werden über Komponenten- beziehungsweise Teilsystemzuverlässigkeiten als a priori Wissen abgebildet und im Rahmen der integrierten Modellierung als lokale Zuverlässigkeiten bezeichnet. Ein oder mehrere Fehlerursachen führen zu verschiedenen Fehlzuständen der Komponente oder des Teilsystems, die in einer Zufallsvariablen zusammengefasst sind. Der Fehlzustand einer Komponente oder eines Teilsystems wird mittels einer Zufallsvariable mit beliebig, aber endlich vielen Zuständen, modelliert. Hierzu werden Wahrscheinlichkeiten angegeben, um die probabilistischen oder möglicherweise kausalen Auswirkung einer Fehlerursache auf einen Fehlzustand zu beschreiben. Diese basieren

Beeinträchtigung	Zustände	CPT-Einträge
Fehler- ursache	Binäre Zustände (<i>active, dormant</i>)	Eintrittswahrscheinlichkeit (Komponentenzuverlässigkeit etc.), Fortpflanzungswahrscheinlichkeit
Fehl- zustand	Beliebig, endlich viele Zustände (Schädigungs- bzw. Gesundheitszustände)	Auswirkungswahrscheinlichkeit
Ausfall	Binäre Zustände (<i>operational, failed</i>)	Auswirkungswahrscheinlichkeit

Tabelle 3.2: Modellierung der Beeinträchtigungen der Verlässlichkeit auf Basis von Abbildung 3.15 a)

typischerweise auf Expertenwissen. Die Fehlzustände bedingen einen Ausfall der Komponente oder des Teilsystems. Ausfälle werden wieder mittels binärer Zufallsvariablen abgebildet, um einen Ausfall (*failed*) oder störungsfreien Betrieb (*operational*) abzubilden. Die Wahrscheinlichkeit, mit der ein Fehlzustand zu einem Ausfall führt, wird auf Basis von Expertenwissen angegeben und quantifiziert damit den Zusammenhang zwischen Fehlzustand und Ausfall. Ausfälle treten dabei als externe Fehlerursache für nachfolgende Komponenten oder Teilsysteme auf (vergleiche Abschnitt 3.4.3.2) und aktivieren Fehlerursachen über Fortpflanzungswahrscheinlichkeiten.

Ein so aufgebautes Bayes'sches Netz ist bereits als Systemzuverlässigkeitsmodell zu verwenden, weist aber Einschränkungen hinsichtlich der Übersichtlichkeit der Darstellung, der Anzahl an benötigten Nutzereingaben sowie der Modellierungstiefe auf. Daher wird das Bayes'sche Netz zunächst strukturell vereinfacht, indem Fehlzustand- und Ausfallzufallsvariablen einer Komponente oder eines Teilsystems zu *Ausfallzuständen* zusammengefasst werden (vergleiche Abbildung 3.15 b)). Dies hat zur Folge, dass der Ausfall einer Komponente oder eines Teilsystems kein binäres Ereignis mehr darstellt, sondern differenzierter hinsichtlich des Zustands mittels gradierter Zufallsvariablen beschrieben wird. Die Anzahl an für den Nutzer zu schätzenden Parametern wird somit auf ein beherrschbares Maß reduziert. In Analogie zu Tabelle 3.2 sind die Modellierungsaspekte für Zufallsvariablen der Ausfallzustände in Tabelle 3.3 abgebildet.

Beeinträchtigung	Zustände	CPT-Einträge
Fehler- ursache	Binäre Zustände (<i>active, dormant</i>)	Eintrittswahrscheinlichkeit (Komponentenzuverlässigkeit etc.)
Ausfall- zustand	Beliebig, endlich viele Zustände (<i>operational, ..., failed</i>)	Auswirkungswahrscheinlichkeit, Fortpflanzungswahrscheinlichkeit

Tabelle 3.3: Modellierungsaspekte der Beeinträchtigungen der Verlässlichkeit in Abbildung 3.15 b, c, d)

Die Zufallsvariable des Ausfallzustands weist dieselben Zustände (*operational*, *failed*) der vorherigen Zufallsvariable des Ausfalls auf, um das Ausfallverhalten abzubilden. Als Inkremente zwischen diesem Boole'schen Verhalten dienen die ehemaligen Fehlzustände, sodass das Ausfallverhalten strukturell gleich abgebildet wird. Dies ermöglicht für die Eingabe der Auswirkungswahrscheinlichkeiten in die CPTs der Zufallsvariablen eine Betrachtung des ungünstigsten Ausfallverhaltens, indem jeder Eintritt einer Fehlerursache direkt zu einem Ausfallzustand *failed* führt (vergleiche Abschnitt 3.4.2.6).

Der nächste Schritt in der Reduzierung des Bayes'schen Netzes betrachtet die Fortpflanzung von Ausfällen einer Komponente oder eines Teilsystems auf weitere Teilsysteme, sogenannte kaskadierte Ausfälle. Hier treten Ausfälle als externe Fehlerursache einer nachfolgenden Komponente oder Teilsystems auf. Die externe Fehlerursache wird nach der vorangegangenen Einführung der Ausfallzustände direkt durch diese kausal aktiviert. Die Ausfallzustände werden über Fortpflanzungswahrscheinlichkeiten in den externen Fehlerursachen berücksichtigt. Für die weitere strukturelle Reduzierung des Bayes'schen Netzes werden die Fortpflanzungswahrscheinlichkeiten in die Ausfallzustände der nachfolgenden Komponente oder des Teilsystems verschoben (vergleiche Abbildung 3.15 c)). Die Zufallsvariablen für die entsprechenden externen Fehlerursachen werden vernachlässigt. Damit ist die strukturelle Reduzierung des Bayes'schen Netzes abgeschlossen. Mit dem Ziel der Steigerung der Modellierungstiefe wird die Betrachtungszeit diskretisiert und das Bayes'sche Netz zu jedem diskreten Zeitpunkt betrachtet, wobei temporale Abhängigkeiten zwischen den Betrachtungszeitpunkten hinzugefügt werden. Aus dieser Erweiterung folgt ein dynamisches Bayes'sches Netz (vergleiche Abbildung 3.15 d)). In den folgenden Kapiteln werden der allgemeine Modellierungsansatz sowie zwei Ansätze zur Abbildung spezieller Zuverlässigkeitsaspekte (Ausfälle gemeinsamer Ursache und Redundanz) mittels dynamischer Bayes'scher Netze detaillierter vorgestellt.

3.4.3.1 Allgemeiner Modellierungsansatz

Aufbauend auf dem Entwurf des Modellierungsansatzes zur Abbildung der Systemzuverlässigkeit wird im Folgenden die konkrete Implementierung mittels dynamischer Bayes'scher Netze anhand eines Beispielsystems dargestellt. Betrachtet wird das technische System C mit dem Blockschaltbild in Abbildung 3.16, bestehend aus zwei Komponenten A und B .

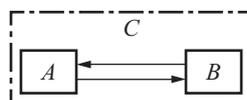


Abbildung 3.16: Blockschaltbild eines Systems C mit Komponenten A und B .

Die Topologie des zu untersuchenden Systems wird zunächst abstrahiert und mittels eines Topologiegraphen dargestellt. Dieser gerichtete Graph dient der Abbildung aller Wirkzusammenhänge, die innerhalb des Systemmodells modelliert sind, und berücksichtigt die Hierarchie des Systems. Mehrfache Abhängigkeiten zwischen unterschiedlichen Komponenten oder Teilsystemen werden jeweils als eine gerichtete Kante zusammengefasst. Komponenten in einem Teilsystem weisen immer Kanten zum Teilsystemknoten auf, während Abhängigkeiten zwischen Komponenten in unterschiedlichen Teilsystemen immer über die höchste involvierte Hierarchiestufe propagiert werden. Der Topologiegraph

des technischen Systems C ist in Abbildung 3.17 dargestellt. Auf Basis des Topologiegraphen erfolgt die automatisierte Ableitung des Systemzuverlässigkeitsmodells (vergleiche Abschnitt 3.5).



Abbildung 3.17: Topologiegraph des technischen Systems C mit Komponenten A und B .

Für die Komponenten A und B seien jeweils zwei Fehlerursachen V, U und G, H identifiziert worden. Darüber hinaus seien beide Komponenten in ihrem Ausfallverhalten aufgrund der Rückkopplung im Blockschaltbild voneinander abhängig. Das System C weise selbst keine internen Fehlerursachen auf. Das beschriebene technische System ist in Abbildung 3.18 als dynamisches Bayes'sches Netz als Modell der Systemzuverlässigkeit dargestellt.

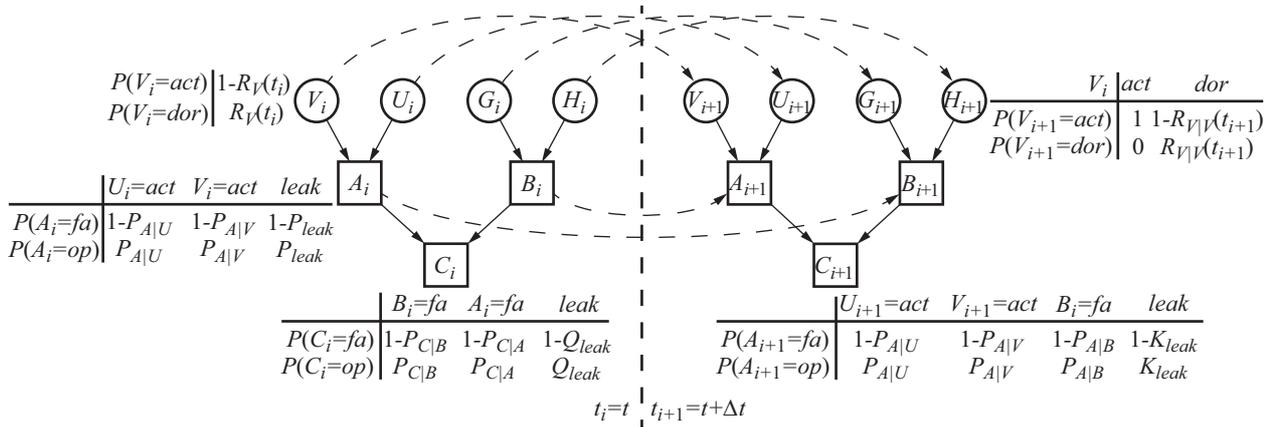


Abbildung 3.18: Modellierungsansatz mittels dynamischer Bayes'scher Netze mit Unsicherheit.⁶

In den Zeitscheiben t_i und t_{i+1} sind die Beeinträchtigungen, Fehlerursachen und Ausfallzustände sowie deren grundlegenden Abhängigkeiten abgebildet. Der DAG ist zeitinvariant und damit topologisch äquivalent innerhalb der Zeitscheiben. Die Komponenten A und B weisen jeweils zwei Fehlerursachen V und U beziehungsweise G und H auf. Die Ausfallzustände der Komponenten A und B sind als externe Fehlerursachen der Ausfallzustände des Teilsystems C modelliert und bilden damit die Hierarchie innerhalb des zu modellierenden technischen Systems ab. Ein Ausfall des Teilsystems C ist damit von den Ausfällen seiner enthaltenen Komponenten A und B abhängig. Aus dem Blockschaltbild des technischen Systems geht hervor, dass die Komponenten A und B jeweils voneinander abhängig sind, da das Blockschaltbild eine zyklische Topologie aufweist. Daraus soll hier gefolgert werden, dass das Ausfallverhalten beider Komponenten durch jeweils voneinander abhängigen Ausfällen charakterisiert ist. Abhängige Ausfälle

⁶Alle nicht dargestellten CPTs von Ausfallzustandsvariablen sind zeitinvariant. Alle nicht abgebildeten CPTs von Fehlerursachenvariablen sind analog zu den dargestellten CPTs umzusetzen. Die Notation $R_{V|V}$ beschreibt die bedingte lokale Zuverlässigkeiten nach Gleichungen 2.4 einer Komponente V zum Zeitpunkt t_{i+1} .

zwischen den Komponenten A und B beziehungsweise zwischen dem Teilsystem C und den Komponenten A und B werden auch als kaskadierte Ausfälle bezeichnet [Int11]. Diese sind nicht in einer Zeitscheibe des dynamischen Bayes'schen Netzes abbildbar, da diese Modelle nur für azyklische Graphen definiert sind (vergleiche Abschnitt 3.4.2.3). Um Zyklen im Blockschaltbild technischer Systeme und damit auch in ihrem Ausfallverhalten abzubilden, erfolgt die Erweiterung des statischen Netzes auf ein dynamisches Netz mittels Duplikation der Zeitscheibe t_i in weitere n Zeitscheiben (vergleiche Darstellung in zwei Zeitscheiben t_i und t_{i+1} in Abbildung 3.18). Die Abhängigkeit der Komponenten A und B wird als temporale Abhängigkeit, also über einen Zeitschritt Δt wirkend, betrachtet (vergleiche Kanten⁷ (A_i, B_{i+1}) und (B_i, A_{i+1}) in Abbildung 3.18). Für temporale Abhängigkeiten gilt im Systemzuverlässigkeitsmodell, dass ausschließlich Kanten aus Zeitscheibe t_i zur Zeitscheibe t_{i+1} zulässig sind. Dies hat zur Folge, dass Fehlerursachen über mehrere Zeitscheiben hinweg propagiert werden können, bis sie einen Einfluss auf den Zielknoten ausüben. Eine Erläuterung zu diesem Phänomen ist im Anhang A.2 gegeben. In einem hierarchisch aufgebauten System tritt dieses Phänomen jedoch nicht auf, da alle Einflüsse innerhalb der Hierarchie an den Systemknoten propagiert werden. Die Fehlerursachen in Zeitscheibe t_i werden über temporale Kanten (V_i, V_{i+1}) , (U_i, U_{i+1}) , (G_i, G_{i+1}) , (H_i, H_{i+1}) , jeweils mit der entsprechenden Instanz in Zeitscheibe t_{i+1} , verbunden. Somit wird der Zeitverlauf des Ausfallverhaltens des betrachteten Systems über eine Diskretisierung der Betrachtungszeitpunkte der Wahrscheinlichkeitsverteilungsfunktion innerhalb der Fehlerursachen darstellbar. Aus Abschnitt 3.4.2.4 ist bekannt, dass Verteilungsfunktionen, welche die Markov-Eigenschaft erfüllen, abbildbar sind. Für Verteilungsfunktionen, die nicht die Markov-Eigenschaft erfüllen, beispielsweise die Weibull-Verteilung (vergleiche Gleichung 2.5 und Abschnitt 2.1), ist die bedingte Wahrscheinlichkeit $P(A_{t+\Delta t}|A_t)$ von der aktuellen Betrachtungszeit t abhängig. In Gleichung 3.12 ist beispielhaft die bedingte Wahrscheinlichkeit $P(A_{t+\Delta t}|A_t)$ einer Weibull-Verteilung für eine Zufallsvariable A gezeigt:

$$P(A_{t+\Delta t}|A_t) = \frac{P(A_{t+\Delta t} \cap A_t)}{P(A_t)} = \frac{P(A_{t+\Delta t})}{P(A_t)} \Rightarrow \frac{e^{(-\lambda(t+\Delta t))^b}}{e^{(-\lambda t)^b}}. \quad (3.12)$$

Daraus folgt, dass für die Diskretisierung der Betrachtungsdauer die absoluten Betrachtungszeitpunkte berücksichtigt werden müssen. Verteilungsfunktionen werden im vorgestellten Modellierungsansatz nur zur Abbildung von Fehlerursachen verwendet. Für Auswirkungswahrscheinlichkeiten in den Ausfallzuständen werden zeitinvariante Wahrscheinlichkeiten angenommen. Es können jedoch auch Verteilungsfunktionen implementiert werden.

Die CPTs der Fehlerursachen weisen stets die gleiche Struktur auf. Bereits in Zeitscheibe t_i eingetretene Fehlerursachen werden in folgenden Zeitscheiben t_{i+1} unverändert mittels binärer CPT-Einträge über die jeweiligen temporalen Kanten propagiert (vergleiche dazu $P(V_{i+1}|V_i = act) = \{1, 0\}$ in CPT $P(V_{i+1}|V_i)$ in Abbildung 3.18). So lange die Fehlerursache in der Zeitscheibe t_i nicht eingetreten ist, erfolgt die weitere Modellierung der Verteilungsfunktion mit $P(V_{i+1}|V_i = dor) = \{1 - R_V(t_{i+1}), R_V(t_{i+1})\}$ in CPT $P(V_{i+1}|V_i)$ wie in Abbildung 3.18. Aus dieser Modellierung resultiert, dass keine reparierbaren Systeme darstellbar sind.

Die zusätzliche Berücksichtigung von externen Ereignissen mit Einfluss auf das Ausfall-

⁷Darstellung von gerichteten Kanten in Graphen (*Startknoten*, *Endknoten*) nach [Die17].

verhalten des betrachteten Systems ist mittels der Integration weiterer Variablen jeweils als Fehlerursache in die Zeitscheiben des dynamischen Bayes'schen Netzes möglich. Die Integration weiterer Modellierungsaspekte, wie zusätzliche Abhängigkeiten im Ausfallverhalten der Komponenten und Teilsysteme, werden mittels Hinzufügen von Kanten zwischen den betroffenen Elementen realisiert und in Abschnitt 3.5 diskutiert.

Im gewählten Modellierungsansatz, basierend auf dynamischen Bayes'schen Netzen als Modell der Systemzuverlässigkeit, werden zwei unterschiedliche Arten von Unsicherheiten berücksichtigt. Diese Unsicherheiten werden in den Ausfallzuständen abgebildet und optional berücksichtigt. Das folgenschwerste Ausfallverhalten einer Komponente oder eines Teilsystems stellt eine OR-Verknüpfung in den Ausfallzuständen dar, d. h. der Eintritt einer einzelnen Fehlerursache führt direkt und kausal zu einem Ausfall. Um einen konservativen Modellierungsansatz zu wählen, werden CPTs der Ausfallzustände A , B und C mittels noisy-MAX-Verknüpfung als Verallgemeinerung der OR-Verknüpfung modelliert, in denen Unsicherheiten berücksichtigt werden können. Die Ausfallzustände erfordern mindestens die Zustände *failed* und *operational*. Der Zusammenhang zwischen Fehlerursachen und Ausfallzuständen wird über Einträge in den CPTs der Ausfallzustände abgebildet. Ist dieser Zusammenhang kausal, gilt beispielsweise in der CPT der Zufallsvariable A_i für die Einträge $P_{A|U}, P_{A|V} \in \{0, 1\}$. Wird dieser Zusammenhang mit Unsicherheit hinsichtlich der Auswirkung von Fehlerursachen auf die Ausfallzustände einer Komponente oder eines Teilsystems abgebildet, gilt beispielsweise in der CPT der Zufallsvariable A_i für die Einträge $P_{A|U}, P_{A|V} \in]0, 1[$. Unsicherheiten bei der Auswirkung von abhängigen Ausfällen folgen diesem Modellierungsansatz. Exemplarisch wird für die Abhängigkeit zwischen den Ausfallzuständen B_i und A_{i+1} die CPT von A_{i+1} betrachtet. Auch hier gilt für die Einträge der CPT $P_{A|U}, P_{A|V}, P(A|B) \in]0, 1[$. Die zweite Art von Unsicherheiten bezieht sich auf die gewählte Modellierungstiefe. Nicht explizit modellierte Fehlerursachen werden im *leak*-Anteil der CPTs der Ausfallzustände (vergleiche $P_{leak}, Q_{leak}, K_{leak}$ in Abbildung 3.18) berücksichtigt. Der *leak*-Anteil wird als Unsicherheit hinsichtlich der vollständigen Berücksichtigung der dominierenden Fehlerursachen betrachtet.

Das primäre Analyseziel des Modells ist die Abbildung der Systemzuverlässigkeit über der Betrachtungsdauer $R_{Sys}(t_i)$. Dazu werden Inferenzalgorithmen verwendet, die das Modell jeweils für bestimmte Knoten auswerten. Für diese Zielknoten werden alle Systemknoten C_i gewählt, da diese das System in jeder Zeitscheibe repräsentieren. Daraus folgt für die Systemzuverlässigkeit $R_{Sys}(t_i) = P(C_i)$. Für $t_i \in \{t_0, \dots, t_n\}$ mit $i \in \{0, \dots, n\}$ wird der Verlauf der Systemzuverlässigkeit über der diskretisierten Betrachtungsdauer ermittelt.

3.4.3.2 Ausfälle gemeinsamer Fehlerursache

Zu den abhängigen Ausfällen zählen, neben den bereits diskutierten kaskadierten Ausfällen, auch die Ausfälle gemeinsamer Fehlerursache (vergleiche Abschnitt 2.2). Der Modellierungsansatz folgt der Abbildung von Ausfällen gemeinsamer Fehlerursache nach O'Connor [O'C13] und wird anhand einer auf ein statisches Bayes'sches Netz reduzierten und angepassten Darstellung des aus Abbildung 3.18 bekannten Systems erläutert. In Abbildung 3.19 ist eine Zeitscheibe des bekannten Systems gezeigt. Das Teilsystem C wird hier nicht berücksichtigt. Die Fehlerursache U habe nun Einfluss auf die Ausfallzustände A und B und stelle die gemeinsame Fehlerursache für Ausfälle von A und B dar. Auf die Darstellung mittels dynamischer Bayes'scher Netze wird verzichtet, da der zu modellierende Zuverlässigkeitsaspekt keine temporalen Abhängigkeiten bedingt.

Ebenso wird auf die Abbildung von Unsicherheiten mittels *leak* aus Gründen der Übersichtlichkeit verzichtet. Die Modellierung ist in den allgemeinen Modellierungsansatz, wie in Abschnitt 3.4.3.1 beschrieben, vollständig integrierbar.

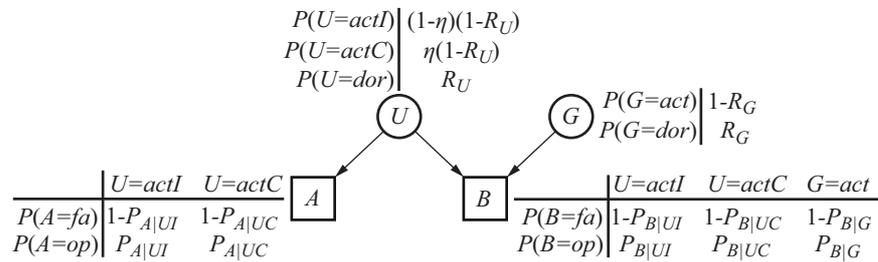


Abbildung 3.19: Bayes'sche Netze zur Modellierung von Ausfällen gemeinsamer Fehlerursache.

Der gewählte Modellierungsansatz zur Abbildung von Ausfällen gemeinsamer Fehlerursache legt die Annahme zugrunde, dass eine Fehlerursache individuellen und gemeinsamen Einfluss auf ihre nachfolgenden Ausfallzustände aufweist. Dieser Einfluss wird über den Parameter η („Kopplungsfaktor“) als Wahrscheinlichkeit, dass eine gemeinsame Fehlerursache sich auf weitere Komponenten oder Teilsysteme ausbreitet, abgebildet. Dabei gilt für den Kopplungsfaktor $\eta \in]0, 1[$, wobei $\eta = 1$ einer sich ausbreitenden gemeinsamen Fehlerursache entspricht. Dem gegenüber bildet ein Kopplungsfaktor $\eta = 0$ nur individuelle Auswirkungen auf einzelne Komponenten oder Teilsysteme ab.

In den betroffenen Ausfallzuständen der nachfolgenden Komponenten A und B quantifizieren die Wahrscheinlichkeiten $P(A|UI)$ und $P(B|UI)$ den Einfluss der gemeinsamen Fehlerursache U als individuelle Auswirkung. Die Wahrscheinlichkeiten $P(A|UC)$ und $P(B|UC)$ beschreiben die Auswirkung der gemeinsamen Fehlerursache U auf die Ausfallzustände A und B .

3.4.3.3 Redundante Teilsysteme und Komponenten

Redundante technische Systeme weisen nach Abschnitt 2.2 abhängige Ausfälle der beteiligten Komponenten und Teilsysteme auf und wurden in *heiße*, *warme* und *kalte* Redundanzarten unterteilt. Die Integration von redundanten Komponenten und Teilsystemen in technische Systeme ist der Fehlertoleranz aus den Mitteln der Verlässlichkeit (vergleiche Abbildung 2.1) zuzuordnen.

Es sind verschiedene analytische Ansätze zur Modellierung von Redundanzen bekannt [Bea78, KZ03, HX10, BB15], die in Bayes'schen Netzen anwendbar sind. Diese sind einfach in den allgemeinen Modellierungsansatz zu integrieren und recheneffizient. Allerdings ist die Abbildung der strukturellen Zusammenhänge von Fehlerursachen und Ausfallzuständen innerhalb des Systemzuverlässigkeitsmodells stark verwischt. Stattdessen wird eine Modellierung nach [KMS16] vorgestellt, die diese Zusammenhänge in den allgemeinen Modellierungsansatz integriert. Der Modellierungsansatz ermöglicht sowohl die Abbildung homogener gleichartiger redundanter Komponenten und Teilsysteme sowie die Abbildung nicht gleichartiger (diversitärer) redundanter Komponenten und Teilsysteme [MP10] jeweils für die drei Redundanzarten heiß, warm und kalt. Bei homogener Redundanz wird angenommen, dass alle Fehlerursachen der redundanten Elemente die gleiche Eintrittswahrscheinlichkeit aufweisen. Bei diversitärer Redundanz gilt demnach, dass al-

le Fehlerursachen der redundanten Elemente ungleiche Eintrittswahrscheinlichkeiten aufweisen. Im Folgenden wird ein solcher Modellierungsansatz auf Basis der Definition in Abschnitt 2.2 für k -aus- m Strukturen anhand der drei Arten von Redundanz für Systeme mit und ohne Lastteilung vorgestellt.

An einem 2-aus-3 redundanten System D mit den Komponenten A , B und C wird im Folgenden der Modellierungsansatz für die verschiedenen Redundanzarten diskutiert. Zur Erhaltung einer kompakten Darstellung werden mit Ausnahme heißer Redundanzen ohne Lastteilung, nur jeweils eine belastungsabhängige Fehlerursache ($V_{i/i+1}$, $G_{i/i+1}$, $K_{i/i+1}$ in Abbildung 3.21) pro Komponente betrachtet. Diese Fehlerursachen sind direkt durch Ausfälle der redundanten Komponenten ($A_{i/i+1}$, $B_{i/i+1}$, $C_{i/i+1}$) aufgrund ihrer Aktivierung zur Kompensation der Ausfälle und Lastteilung betroffen, sodass ein beschleunigter Eintritt wahrscheinlicher ist. Weitere Fehlerursachen, die nicht von der Lastteilung berührt sind, werden integriert (vergleiche Abschnitt 3.4.3.1). Die Modellierung redundanter Systeme berücksichtigt die Modellierung von Unsicherheiten wie in Abschnitt 3.4.3.1 vorgestellt. Ausnahmen stellen die Knoten der redundanten Teilsysteme (vergleiche $D_{i/i+1}$ in Abbildungen 3.20 und 3.21) dar. Diese Knoten sind maßgeblich für die Abbildung von Redundanzen, da hier eine konjunkte Verknüpfung zur maßgeblichen Beschreibung der Redundanzart implementiert ist. Eine Berücksichtigung von Unsicherheiten ist hier nicht vorgesehen, da dies als eine Aufweichung der Funktionalität der Redundanz zu interpretieren und damit nicht zielführend ist.

Technische Systeme mit *heißer Redundanz ohne Lastteilung* werden über statische Bayes'sche Netz mittels einer AND-Verknüpfung abgebildet (vergleiche Abbildung 3.20). Im redundanten System D_i sind alle Komponenten A_i , B_i und C_i und ihre Fehlerursachen V_i , G_i und K_i voneinander unabhängig, sodass keine temporalen Abhängigkeiten zur Abbildung des Ausfallverhaltens modelliert werden müssen. Die Fehlerursachen sind als unabhängig zu betrachten, da keine passiven Elemente vorhanden sind und es zwischen den ausschließlich aktiven Elementen zu keiner Lastteilung kommt. Die konjunkte Verknüpfung der Ausfallzustände der Komponenten A_i , B_i und C_i im redundanten Teilsystem D_i wird mittels einer CPT mit binären Einträgen modelliert, um das kausale Ausfallverhalten einer k -aus- m Redundanz abzubilden. Eine noisy-MIN-Verknüpfung ist in diesem Modellierungsansatz für Redundanzarten für $k > 1$ nicht anwendbar, da diese Zusammenhänge nicht abbildbar sind.

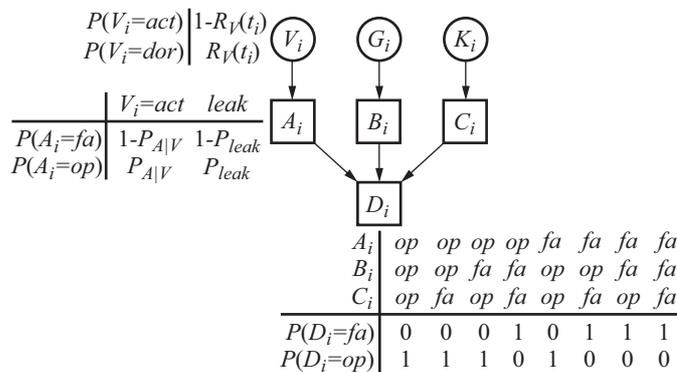


Abbildung 3.20: Statisches Bayes'sches Netz zur Abbildung einer heißen 2-aus-3-Redundanz ohne Lastteilung.⁸

Die Modellierung von Redundanzen mit temporalen Abhängigkeiten, Ausfälle von Komponenten oder Teilsystemen nehmen Einfluss auf die Zuverlässigkeit der verbleibenden Elemente. Diese Modellierung erfolgt mittels dynamischer Bayes'scher Netze, wie in Abbildung 3.21 dargestellt. Neben den in Abschnitt 3.4.3.1 eingeführten temporalen Kanten zwischen Fehlerursachen (V_i, V_{i+1}) , (G_i, G_{i+1}) und (K_i, K_{i+1}) werden Abhängigkeiten zwischen Ausfallzuständen und Fehlerursachen eingeführt. Diese temporalen Kanten (A_i, G_{i+1}) und (A_i, K_{i+1}) bilden den Einfluss eines Ausfalls von Komponente A auf das Ausfallverhalten der Komponenten B und C über ihre Fehlerursachen G_{i+1} und K_{i+1} ab. Der gleiche Zusammenhang gilt für Ausfälle der Komponenten B und C mit den jeweiligen Einflüssen auf die Fehlerursachen V_{i+1} und K_{i+1} beziehungsweise V_{i+1} und G_{i+1} .

In der Abbildung der *warmen Redundanz ohne Lastteilung* und der folgenden Redundanzarten sei Komponente C passiv und werde erst nach Ausfall einer Komponente A oder B aktiviert. Für die passive Komponente gelte $R_K(t_i) \gg R_{V/G}(t_i)$ und $R_K(t_i) \neq 1$ für $t_i > t_0$, sodass eine Degradation der Komponente durch Alterung modelliert und eine idealisierte Betrachtung der passiven Elemente, wie in kalter Redundanz, ausgeschlossen wird. Allgemein gilt für die Modellierung von passiven gegenüber aktiven Elementen $R_{Pass}(t_i) \gg R_{Akt}(t_i)$ und $R_{Pass}(t_i) \neq 1$ für $t_i > t_0$. Aktive Elemente sind aufgrund fehlender Lastteilung invariant gegenüber Ausfällen der beteiligten Komponenten. Demnach gelte nach Aktivierung der Komponente jeweils für die Fehlerursachen $V_i, V_{i+1}, G_i, G_{i+1}$ und K_i, K_{i+1} , dass $R_V = R_{V|B} = R_{V|C} = R_{V|BC}$, $R_G = R_{G|A} = R_{G|C} = R_{G|AC}$ beziehungsweise $R_K = R_{K|A} = R_{K|B} = R_{K|AB}$. Dieser Zusammenhang der aktiven Elemente gilt auch für *kalte Redundanz ohne Lastteilung*. Einen signifikanten Unterschied zur warmen Redundanz stellt die idealisierte Betrachtung der passiven Elemente dar, die keiner Degradation unterliegen. Es gelte daher für alle $m - k$ passiven Elemente in kalter Redundanz $R_{Pass,1\dots(m-k)}(t_i) = 1$ für $t_i = t_0 \dots t_n$. Im vorliegenden System entspricht damit $R_K(t_i) = 1$ für die Fehlerursache K_i, K_{i+1} der Komponente C .

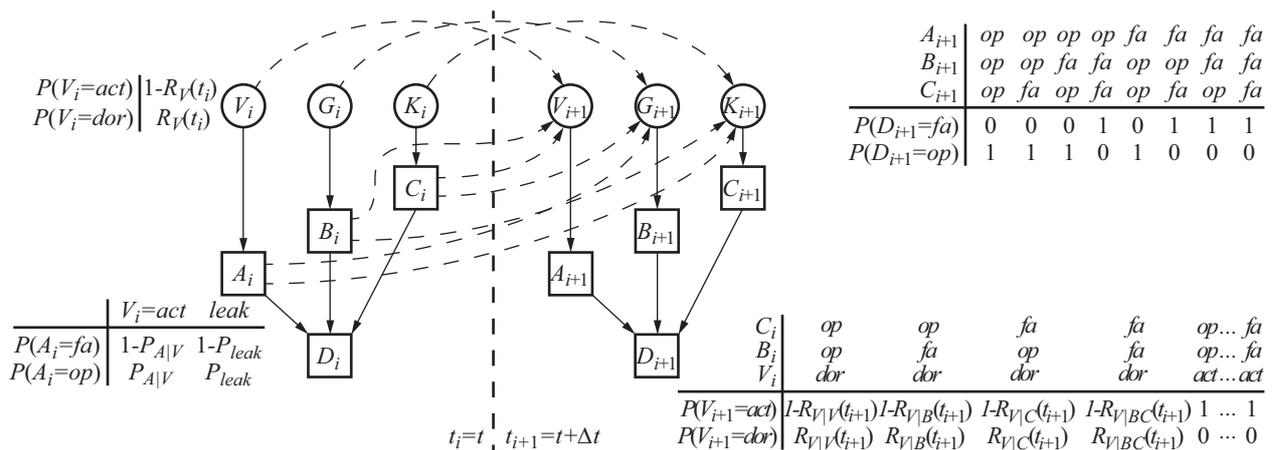


Abbildung 3.21: Dynamisches Bayes'sches Netz nach allgemeinen Modellierungsansatz um zusätzliche temporale Abhängigkeiten zur Abbildung komplexer 2-aus-3-Redundanzen erweitert.⁹

⁸Die Notation $R_{X|Y}$ beschreibt die bedingte lokale Zuverlässigkeiten nach Gleichungen 3.12 und 2.4 einer Komponente X unter der Bedingung, dass eine Komponente Y ausgefallen ist.

⁹Die Notation $R_{X|YZ}$ beschreibt die bedingte lokale Zuverlässigkeiten nach Gleichungen 3.12 und 2.4 einer Komponente X unter der Bedingung, dass zwei Komponenten Y und Z ausgefallen sind.

In *heißer Redundanz mit Lastteilung* treten nur aktive Elemente auf. Die Lastteilung zwischen den aktiven Elementen erfordert die Modellierung von Zuverlässigkeiten der aktiven Elemente in Abhängigkeit von Ausfällen, sodass die Zuverlässigkeit mit steigender Anzahl von Ausfällen sinkt. Für das betrachtete Beispiel gelte für die Zuverlässigkeiten innerhalb der Fehlerursache der Komponenten $R_{V|V} > R_{V|B} \vee R_{V|C} > R_{V|BC}$, $R_{G|G} > R_{G|A} \vee R_{G|C} > R_{G|AC}$ und $R_{K|K} > R_{K|A} \vee R_{K|B} > R_{K|AB}$. Die genaue Ausprägung der Lastteilung ist innerhalb der lokalen Zuverlässigkeitsschätzer implementiert. In *warmer Redundanz mit Lastteilung* treten passive Elemente auf, die Degradation unterliegen, wie es bereits für warme Redundanz ohne Lastteilung erläutert wurde. Die aktiven Elemente unterliegen einer Lastteilung, sodass dieselben Bedingungen gelten wie für die heiße Redundanz mit Lastteilung. Die *kalte Redundanz mit Lastteilung* teilt die bereits beschriebenen Aspekte für passive Elemente mit der kalten Redundanz ohne Lastteilung, während die aktiven Elemente mit Lastteilung den Bedingungen für die heiße Redundanz mit Lastteilung entsprechen. Eine Übersicht über verschiedene Aspekte des gewählten Modellierungsansatzes zur Abbildung allgemeiner k -aus- m redundanter Systeme ist in Tabelle 3.4 gezeigt.

Art	Temporale Abhängigkeiten	Fehlerursache passiver Elemente	Fehlerursache aktive Elemente
Ohne Lastteilung			
<i>Heiß</i>	Nein	Entfällt	Unabhängig von Ausfällen:
<i>Warm</i>	Ausfallzustände zu Fehlerursachen	$R_{Pass}(t_i) \gg R_{Akt}(t_i)$; hier: $R_K(t_i) \gg R_{V/G}(t_i)$	$R_{Akt,1\dots k}(t_i)$; hier: $R_{V V} = R_{V B} = R_{V C} = R_{V BC}$,
<i>Kalt</i>	Ausfallzustände zu Fehlerursachen	$R_{Pass,1\dots(m-k)}(t_i)=1$; hier: $R_K(t_i) = 1$	$R_{G G} = R_{C A} = R_{G C} = R_{G AC}$, $R_{K K} = R_{K A} = R_{K B} = R_{K AB}$.
Mit Lastteilung			
<i>Heiß</i>	Ausfallzustände zu Fehlerursachen	Entfällt	Abhängig von Ausfällen:
<i>Warm</i>	Ausfallzustände zu Fehlerursachen	$R_{Pass}(t_i) \gg R_{Akt}(t_i)$; hier: $R_K(t_i) \gg R_{V/G}(t_i)$	$R_{Akt,1\dots k}(t_i)$; hier: $R_V > R_{V B} \vee R_{V C} > R_{V BC}$,
<i>Kalt</i>	Ausfallzustände zu Fehlerursachen	$R_{Pass,1\dots(m-k)}(t_i)=1$; hier: $R_K(t_i) = 1$	$R_G > R_{G A} \vee R_{G C} > R_{G AC}$, $R_K > R_{K A} \vee R_{K B} > R_{K AB}$.

Tabelle 3.4: Übersicht der Modellierungsansätze zu allgemeinen k -aus- m redundanten Systemen und den in Abbildungen 3.20 und 3.21 dargestellten Systemen.

Der gewählte Modellierungsansatz zur Abbildung redundanter Systeme bildet mit Ausnahme der heißen Redundanz ohne Lastteilung nur exponentialverteilte Eintrittswahrscheinlichkeiten der Fehlerursachen ab. Dies gilt auch für die genannten analytischen Ansätze [Bea78, KZ03, HX10]. Verteilungsfunktionen, die nicht die Markov-Eigenschaft erfüllen, sind in der Berechnung der bedingten Wahrscheinlichkeit im Rahmen des vorgestellten Modellierungsansatzes von der absoluten Betrachtungszeit abhängig (vergleiche die Weibull-Verteilung 3.12). Eine Berücksichtigung der absoluten Ausfallzeiten einzelner Komponenten ist innerhalb der Zustände der Fehlerursachen aufbauend auf Boudali

[BD05] möglich. Die Anzahl der Zustände der betroffenen Fehlerursachen wäre dann abhängig von den $i - 1$ vorangegangenen Zeitschritten sowie der Anzahl $m - k$ an benötigten aktiven Elementen und würde damit stark ansteigen. Ein solches Modell wäre zu unübersichtlich, um es sinnvoll in den bestehenden Modellierungsansatz zu integrieren.

3.5 Automatisierte Ableitung des Systemzuverlässigkeitsmodells

Der zentrale Bestandteil der integrierten Modellierung von Zuverlässigkeit und dynamischem Verhalten wird durch die *automatisierte Ableitung* des Systemzuverlässigkeitsmodells aus dem integrierten Modell beschrieben. Das Ablaufdiagramm der automatisierten Ableitung ist in Abbildung 3.22 dargestellt und zeigt vier übergeordnete Bestandteile: *Schnittstelle zum Systemmodell*, *Erzeugung des DAG* und *Erzeugung der Wahrscheinlichkeiten* des Systemzuverlässigkeitsmodells sowie die *Schnittstelle zur Modellierungsumgebung der Systemzuverlässigkeit*. Die Schnittstellen zum Systemmodell und zum konkreten Systemzuverlässigkeitsmodell sind modellierungsumgebungspezifische Bestandteile, es gibt für jede Modellierungsumgebung ein Modul zur Implementierung der Schnittstelle. Die Schnittstelle zum Systemmodell dient der Identifikation der Topologie des Systemmodells und der Bereitstellung der Simulationsergebnisse. Es folgt die Erzeugung des DAG und der im Systemzuverlässigkeitsmodell enthaltenen Wahrscheinlichkeiten mit den untergliederten Bestandteilen *Erzeugung eines generalisierten DAGs*, der *Reduzierung des generalisierten DAGs* und dem *Hinzufügen weiterer Modellierungsaspekte* in den DAG sowie das *Hinzufügen lokaler Zuverlässigkeitsschätzer* und die abschließende *Erzeugung der CPTs* des Systemzuverlässigkeitsmodells. Das Ergebnis der automatisierten Ableitung ist eine von der Modellierungsumgebung des System- und Systemzuverlässigkeitsmodells unabhängige vollständige Beschreibung eines dynamischen Bayes'schen Netzes zur Abbildung der Zuverlässigkeit des betrachteten mechatronischen Systems. Die Berechnung und Analyse des dynamischen Bayes'schen Netzes (vergleiche Abschnitte 3.4.2.9 und 3.5.3) erfordert geeignete Software-Anwendungen, die über eine Schnittstelle eingebunden wird. Die Implementierung der automatisierten Ableitung orientiert sich an Entwurfsmustern der Software-Entwicklung [GHJ+05]. Diese Entwurfsmuster beschreiben abstrakte Lösungsansätze für die Konzeptbildung und Entwicklung von Software. Die Schnittstellen zum Systemmodell und -zuverlässigkeitsmodell sowie zu den lokalen Zuverlässigkeitsschätzern sind dem Entwurfsmuster des Beobachters angelehnt. Die konkrete Implementierung spezifischer Schnittstellen zum Systemmodell und -zuverlässigkeitsmodell sowie der Zuverlässigkeitsschätzer ist entkoppelt von der Erzeugung des Systemzuverlässigkeitsmodells und nur über dedizierte Funktionen anzusprechen. Die Erzeugung des Systemzuverlässigkeitsmodells erhält jeweils nur eine Liste mit den zur Verfügung stehenden Schnittstellen zum Systemmodell und -zuverlässigkeitsmodell sowie zu den Zuverlässigkeitsschätzern und kann jeweils über eine Schnittstelle Informationen austauschen. Dies erhöht die Flexibilität und Erweiterbarkeit der automatisierten Ableitung durch das Hinzufügen weiterer Schnittstellen und Zuverlässigkeitsschätzer.

Die automatisierte Ableitung des Systemzuverlässigkeitsmodells wird anhand eines einfachen illustrierenden Beispiels erläutert. Hierzu wird der idealisierte Regelkreis eines beliebigen mechatronischen Systems (vergleiche Blockschaltbild in Abbildung 3.23) betrachtet. Die in einem mechatronischen System enthaltene Sensorik (vergleiche Abbildung 3.24)

wird in dieser Darstellung vernachlässigt, da für das Messglied für einfache Betrachtungen eines Regelkreises meist ideales Übertragungsverhalten angenommen wird. Die Summationsstelle wird für die folgende Beschreibung der Ableitung vernachlässigt, da sie in einem realen System typischerweise innerhalb der Informationsverarbeitung umgesetzt ist.

3.5.1 Schnittstelle zu Systemmodellen

Die Schnittstelle der automatisierten Ableitung zum Systemmodell wird über zwei unterschiedliche Parser¹⁰ zur Identifikation der Topologie und der Verwaltung der Simulation implementiert. Aus Abschnitt 2.6 sind verschiedene Modellierungsansätze und -umgebungen für die ganzheitliche Abbildung und Analyse mechatronischer Systeme bekannt. Es ist ersichtlich, dass für jede Modellierungsumgebung, sofern diese die in Abschnitt 3.2 aufgestellten Anforderungen an Systemmodelle für die integrierte Modellierung erfüllen, eine dedizierte Schnittstelle notwendig ist. Diese ist an die individuellen Modell- und Ergebnisdatensyntax sowie -struktur angepasst, ermöglicht die Datenextraktion und die abschließende Speicherung in einem von der Modellierungsumgebung unabhängigen Datenobjekt. Die zwei Parser innerhalb der Schnittstelle haben unterschiedliche Aufgaben: der Topologie-Parser analysiert die Topologie des Systemmodells und überführt diese in eine verallgemeinerte graphenbasierte Darstellung, während der Simulations-Parser die Simulation des Systemmodells verwaltet, diese parametrisiert, startet und die Ergebnisse an das Datenobjekt übergibt. Die Schnittstelle zu Systemmodellen entkoppelt die eigentliche Erzeugung des Systemzuverlässigkeitsmodells von der gewählten Modellierungsumgebung und ermöglicht eine einfache Anpassung der gesamten Methode an unterschiedliche Systemmodelle.

3.5.1.1 Datenobjekt

In Tabelle 3.5 sind die wichtigsten Attribute des Datenobjekts *IntModData* zur plattformunabhängigen Beschreibung des Systemzuverlässigkeitsmodells erläutert. Dieses Datenobjekt fasst alle Informationen, die für die Modellierung und Analyse des Systemzuverlässigkeitsmodells in einer beliebigen Modellierungsumgebung notwendig sind, zusammen und wird während des Ablaufs in Abbildung 3.22 erstellt und erweitert. Es stellt die zentrale Informationsdatenbank der integrierten Modellierung dar und sichert mittels interner Methoden die Integrität der nach jedem durchgeführten Arbeitsschritt hinzugefügten Datensätze innerhalb der Attribute. Die Objektattribute sind in drei Kategorien einteilbar: initiale Nutzereingaben (*directory*, *simulation* und *optParameters*), extrahierte Daten des Systemmodells (*model*, *name*, *dynamicLoads*) und abgeleitete Attribute (*reliability* und *bayesNet*). Die aufgeführten Objektattribute sind in weiteren Unterattributen ausdifferenziert, auf deren Darstellung aus Gründen der Lesbarkeit verzichtet wird. Die initialen Nutzereingaben, als notwendige Bedingung zum Start der integrierten Modellierung, sind auf die Auswahl eines geeigneten Systemmodells, mittels Angabe des Speicherorts *directory* und passender Simulationsparameter *simulation*, beschränkt. Die Simulationsparameter umfassen dabei Eingaben, wie Start- und Endzeit der Simulation, die optionale Auswahl eines Integrationsverfahrens und dessen spezifischer Parameter.

¹⁰Parser sind Softwareprogramme für die Syntaxanalyse einer Eingabe. Hierbei handelt es sich um die Extraktion, Identifikation und Zerlegung, sowie Umwandlung der in der Eingabe enthaltenen Informationen in ein für die Weiterverarbeitung geeignetes Format [GJ08].

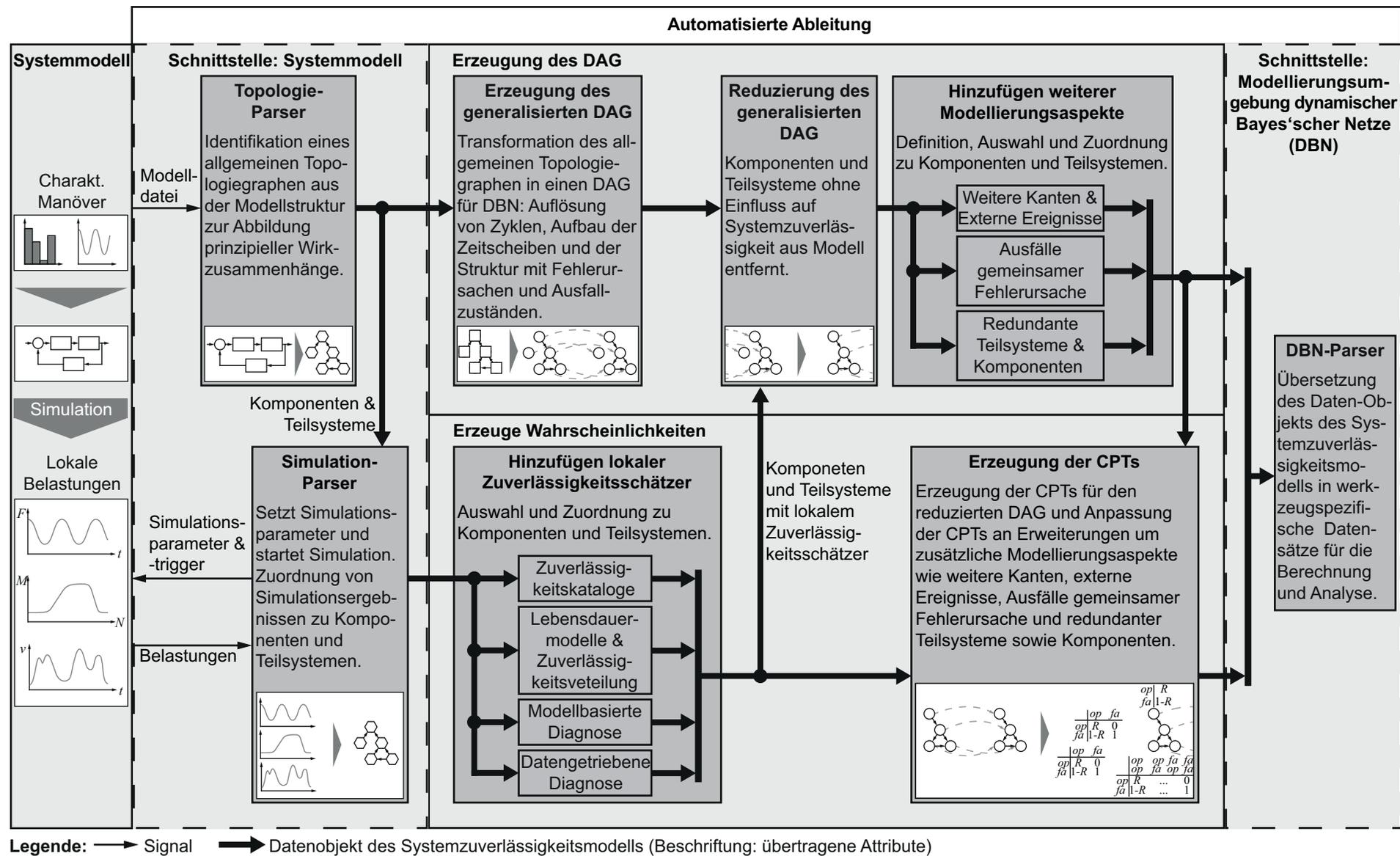


Abbildung 3.22: Ablaufdiagramm der automatisierten Ableitung eines Systemzuverlässigkeitsmodells aus dem Systemmodell.

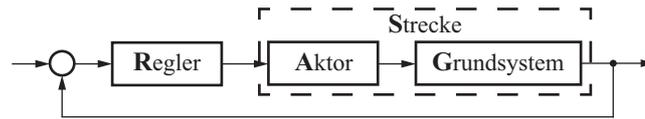


Abbildung 3.23: Idealisierter Regelkreis eines beliebigen mechatronischen Systems ohne Messglied.

Die Schnittstelle zu dem Systemmodell extrahiert verschiedene Informationen aus dem Modell. Zunächst wird die verwendete Modellierungsumgebung identifiziert und gespeichert (*model*), um geeignete Topologie- und Simulations-Parser auszuwählen. Diese analysieren die Modelldatei des Systemmodells und liefern die Bezeichnungen der enthaltenen Komponenten und Teilsysteme für das Attribut *name*. Die Simulationsergebnisse werden erfasst und nach der Ordnung der Komponenten und Teilsysteme im Attribut *name* im Attribut *dynamicLoads* abgelegt. Die Topologie des Systemmodells wird zunächst temporär gespeichert und nach verschiedenen Bearbeitungsschritten innerhalb der im Attribut *bayesNet* abgeleiteten weiteren Attribute direkt abgelegt. Auf Basis der Simulationsergebnisse werden ausgewählten Komponenten und Teilsystemen lokale Zuverlässigkeitsschätzer zugeordnet und berechnet. Die ermittelten Zuverlässigkeiten werden zusammen mit den Zuverlässigkeitsschätzern und möglichen Modellierungsaspekten, wie Redundanzen und Fehlern gemeinsamer Ursache, im Attribut *reliability* zusammengefasst.

Objekt.Attribut	Beschreibung
<i>IntModData.directory</i>	Ordnerpfad zum Speicherort des Systemmodells
<i>IntModData.model</i>	Modellierungsumgebung des Systemmodells
<i>IntModData.name</i>	Bezeichnung der Komponenten und Teilsysteme im Systemmodell
<i>IntModData.simulation</i>	Simulationsparameter des Systemmodells
<i>IntModData.optParameters</i>	Optimierungsparameter mit Startwerten
<i>IntModData.dynamicLoads</i>	Simulationsergebnisse des Systemmodells mit Zuordnung zu Komponenten und Teilsystemen
<i>IntModData.reliability</i>	Lokale Zuverlässigkeitsschätzer (Instanzen der Klasse <i>Estimator</i> , Tabelle 3.6), berechnete lokale Zuverlässigkeiten (Instanzen der Klasse <i>Distribution</i> , Tabelle 3.7) und Modellierungsaspekte der Zuverlässigkeit einzelner Komponenten und Teilsysteme
<i>IntModData.bayesNet</i>	Plattformunabhängige Beschreibung des dynamischen Bayes'schen Netzes: DAG, CPTs, Variablennamen, Analysezielknoten und Modellierungsumgebung

Tabelle 3.5: Übersicht über die maßgeblichen Attribute des Datenobjekts *IntModData* zur plattformunabhängigen Beschreibung des Systemzuverlässigkeitsmodells.

3.5.1.2 Topologie-Parser

Der Entwicklungsprozess mechatronischer Systeme beruht auf der stetigen Konkretisierung eines zunächst groben Entwurfs bis zu einer detaillierten Ausarbeitung des serienreifen Systems. Im Rahmen dieses Prozesses werden Teilfunktionen durch dedizierte Lösungselemente oder Kombination aus diesen in Teilsystemen bereitgestellt. Ein Teilsystem zur prinzipiellen Abbildung vernetzter mechatronischer Systeme ist in Abbildung 3.24 dargestellt. Hier sind die grundlegenden Elemente (Informationsverarbeitung, Sensoren, Aktoren und Grundsystem) durch Energie-, Material- und Informationsflüsse miteinander verknüpft. Eine Unterbrechung dieser Flüsse kann zu einem Fehlzustand und möglicherweise zu einem Ausfall des mechatronischen Teilsystems führen. Diese Flüsse sind die maßgeblichen Indikatoren für die Analyse der Wirkzusammenhänge und des Ausfallverhaltens der im System enthaltenen Elemente. Demnach werden durch die automatisierte Ableitung neben internen nur externe Fehlerursachen betrachtet, die sich über die Flüsse in Flussrichtung im System fortpflanzen. Abhängigkeiten im Ausfallverhalten, die gegenläufig zur Flussrichtung auftreten oder nicht durch die Flüsse abgebildet werden [Jos08], werden durch Nutzereingaben innerhalb des Moduls *Hinzufügen weiterer Modellierungsaspekte* dem Systemzuverlässigkeitsmodell hinzugefügt.

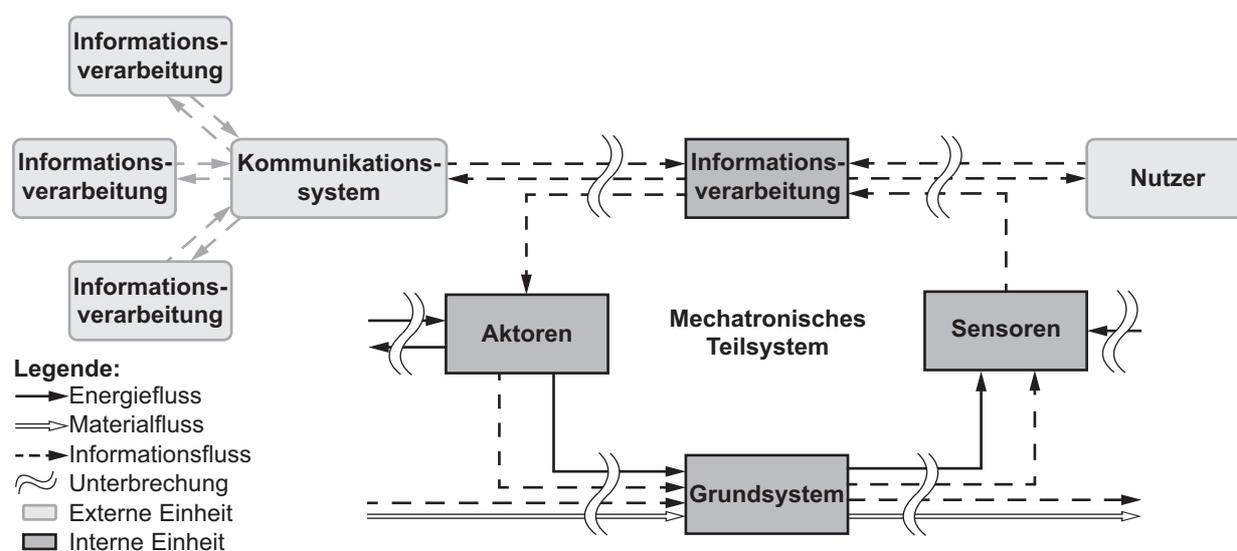


Abbildung 3.24: Vernetztes mechatronisches Teilsystem mit möglichen Flussunterbrechungen (nach [TG18]).

Die automatisierte Ableitung ist in MATLAB¹¹ implementiert und verfügt über Schnittstellen zu Modellierungsumgebungen des Systemmodells, wie MSC ADAMS¹², Simulink/SimScape und DYMOLA¹³/Modelica. Diese teils proprietären Schnittstellen ermöglichen die Verwaltung von Simulationen, sind aber für die Analyse der Topologie nur eingeschränkt anwendbar. Die offenen textbasierten Modelldateien des Systemmodells erlauben jedoch eine Analyse der Topologie, sodass auch ohne direkte Unterstützung durch eine vorhandene Schnittstelle die Topologie des Systemmodells identifiziert werden kann. Der erste Schritt des Topologie-Parsers, nach der Auswahl eines Systemmodells, ist die Identifikation der im System enthaltenen Komponenten und Teilsysteme sowie

¹¹Aktuelle Implementierung in MathWorks MATLAB, Version 2017b.

¹²Aktuelle Implementierung für MSC ADAMS, Version 2015.

¹³Aktuelle Implementierung für DYMOLA, Version 2018 FD01.

die anschließende Speicherung im Attribut *name* des Datenobjekts als Eingangsgröße des Simulations-Parsers. In einem zweiten Schritt werden die Wirkzusammenhänge analysiert und im Topologiegraph zusammengefasst.

Der Topologiegraph wird, wie auch der DAG des dynamischen Bayes'schen Netzes, als Adjazenzmatrix abgebildet. Die $m \times m$ Adjazenzmatrix des Topologiegraphen $\mathbf{S} = (s_{j,l})$ für die Anzahl an Knoten m mit j als Index des Startknotens und l als Index des Endknotens ist nach [Die17] definiert mit:

$$s_{j,l} = \begin{cases} 1, & \text{für Kanten,} \\ 0. & \end{cases} \quad (3.13)$$

Im Topologiegraph sind selbstreferenzierende Kanten, Start- und Endknoten identisch, sodass $s_{j,l} = 1$ mit $j = l$ gilt, zulässig. Allgemeiner ist der Topologiegraph ein zyklischer Graph, sodass gilt $\forall(j,l), s_{j,l} \in \{0,1\}$. Zyklen im Topologiegraphen haben zwei Ursachen: zum einen werden Rückkopplungen im Systemmodell, beispielsweise aufgrund eines Regelkreises, als Zyklen abgebildet, auf der anderen Seite werden mittels des Topologieparsers auch kinematische Kopplungen zwischen Körpern berücksichtigt. Diese sind als bidirektional im Sinne der Wirkzusammenhänge (*Actio* und *Reactio*) zu betrachten und abzubilden. Die Richtung der Flüsse des Systemmodells wird in den Topologiegraphen übertragen und übernommen. Der Topologiegraph des Beispielsystems ist in Abbildung 3.25 und die Adjazenzmatrix \mathbf{S}_{Topo} in Gleichung 3.14 dargestellt. Die Darstellungen eines Graphen sind äquivalent. In dieser Arbeit werden Graphen vornehmlich als grafische Abbildung dargestellt, die Adjazenzmatrix wird insbesondere zur Erläuterung der Erzeugung des DAG herangezogen.

$$\mathbf{S}_{Topo} = \begin{array}{c|ccccc} & Sys & R & S & G & A \\ \hline Sys & 0 & 0 & 0 & 0 & 0 \\ R & 1 & 0 & 1 & 0 & 0 \\ S & 1 & 1 & 0 & 0 & 0 \\ G & 0 & 0 & 1 & 0 & 0 \\ A & 0 & 0 & 1 & 1 & 0 \end{array} \quad (3.14)$$

Ein Mittel zur Beherrschung der Komplexität mechatronischer Systeme ist die Hierarchisierung zur Strukturierung und gegebenenfalls Modularisierung des Systems und ist damit auch im Systemmodell enthalten. Die im Systemmodell abgebildete Hierarchie wird im Systemzuverlässigkeitsmodell ebenfalls modelliert. Dazu ist es notwendig, dass bereits die Topologie-Parser diese Hierarchie berücksichtigen und aufrechterhalten. Die Umsetzung der Hierarchie im Topologiegraphen folgt der Forderung, dass alle Knoten der Komponenten und Teilsysteme innerhalb einer Hierarchieebene eine Kante in Richtung des Knotens auf nächsthöherer Hierarchieebene aufweisen (vergleiche Kanten (A, S) , (G, S) , (R, Sys) und (S, Sys) in Abbildung 3.25). Die Kanten in der Hierarchie des Topologiegraphen verlaufen daher immer *bottom-up*. Kanten, welche die Hierarchieebenen top-down durchlaufen, sind nicht zulässig. Dem Topologiegraphen wird immer ein Knoten auf oberster Hierarchieebene hinzugefügt, der das zu analysierende System abbildet (vergleiche Knoten *Sys* in Abbildung 3.25). Aus dieser Abbildung der Hierarchie des Systemmodells ergibt sich eine baumartige Struktur des Topologiegraphen, die eine Transformation in einen DAG unterstützt.

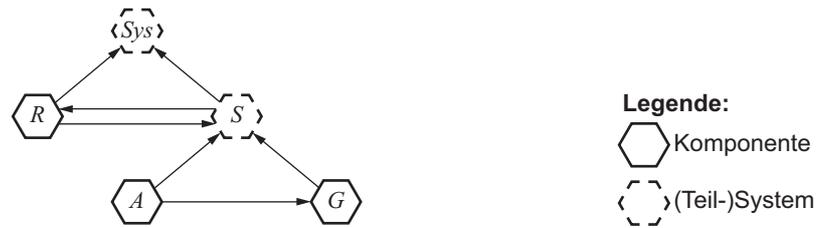


Abbildung 3.25: Abgeleiteter Topologiegraph basierend auf dem System in Abbildung 3.23 und Adjazenzmatrix S_{Topo} in Gleichung 3.14.

3.5.1.3 Simulation-Parser

Eine Simulation des Systemmodells über die gesamte Betriebsdauer ist aufgrund der verfügbaren Ressourcen nicht zielführend. Die Zeitkonstanten vieler mechatronischer Systeme liegen, insbesondere aufgrund elektrischer Signale, im Bereich von ms oder weniger. Daher sind entsprechend kleine Schrittweiten oder adaptive Schrittweiten erforderlich und folglich eine vollständige Simulation aufwendig. Dies führt je nach Komplexität des Systemmodells zu einem großen Berechnungsaufwand, einer hohen Simulationsdauer sowie Speicherbedarfen. Es ist daher nur zielführend das Systemmodell für ein repräsentatives Manöver zu berechnen, die resultierenden Belastungen auf die Komponenten und Teilsysteme aufzunehmen und einer Zuverlässigkeitsberechnung zuzuführen (vergleiche Abschnitt 3.3). Um diese Belastungen der automatisierten Ableitung zugänglich zu machen, wird eine Schnittstelle zur Verwaltung der Simulation des Systemmodells benötigt – der Simulations-Parser. Dieser wird eng mit dem Topologie-Parser für jede Modellierungsumgebung des Systemmodells zusammen entwickelt. Für den Simulations-Parser sind, im Gegensatz zum Topologie-Parser, oft von der Modellierungsumgebung selbst bereitgestellte Schnittstellen zur Verwaltung der Simulation verfügbar. Dies erlaubt eine effiziente Implementierung und Einsatz des Simulations-Parsers. Grundlegende Parameter der Simulation werden über diese Schnittstelle verwaltet, wobei diese aus vorherigen Analysen des Systemmodells bekannt sind und verwendet werden können. Die Simulation wird über diese Schnittstelle gestartet. Es ist darauf zu achten, dass die Simulationsergebnisse in einer externen Datei gespeichert werden, um diese der automatisierten Ableitung zuzuführen. Diese Simulationsergebnisse werden typischerweise in einer dem XML-Standard¹⁴ ähnlichen Datei abgelegt. Dies ermöglicht die Verwendung verfügbarer XML-Parser, um die in der Datei enthaltenen Informationen effizient in das Attribut des Datenobjekts *intModData.dynamicLoads*, entsprechend der identifizierten Komponenten und Teilsysteme im Attribut *name*, abzulegen. Die Verfüg- und Anwendbarkeit bekannter XML-Parser ist, in Anbetracht zahlreicher Signale pro Komponente oder Teilsystem, einer effizienten Umsetzung dienlich.

3.5.2 Erzeugung des Systemzuverlässigkeitsmodells

Die Erzeugung des Systemzuverlässigkeitsmodells ist in zwei gekoppelte Programmstrukturen zur Erzeugung des DAG und der CPTs unterteilt. Der Ablauf und die Koppelung der Bestandteile ist Abbildung 3.22 zu entnehmen. Die Beschreibung der Programmteile

¹⁴Extensible Markup Language (XML) ist eine Meta-Sprache zur Abbildung hierarchisch strukturierter Daten in einer Textdatei. XML-Derivate übernehmen die grundlegenden Regeln der XML-Metasprache zur Definition eines neuen Dokumententyps einer spezifischen Anwendung [BPSM+97]

folgt diesem Ablauf und endet mit der Bereitstellung des vollständigen Datenobjekts für die anschließende Analyse des Systemzuverlässigkeitsmodells.

3.5.2.1 Hinzufügen lokaler Zuverlässigkeitsschätzer

Die Erzeugung des Systemzuverlässigkeitsmodells beginnt mit der Auswahl geeigneter lokaler Zuverlässigkeitsschätzer auf Basis der Ergebnisse einer FMEA des zu untersuchenden Systems. Die FMEA liefert mögliche Fehlerursachen des Systems einhergehend mit einer Bewertung der Kritikalität und damit, neben dem Systemmodell, die zusätzlichen Informationen, die für eine Zuverlässigkeitsanalyse notwendig sind. Aufbauend auf den Ergebnissen der FMEA werden Optimierungsmaßnahmen definiert, um Fehlerursachen mit einer hohen Kritikalität, als kumulatives Maß der Schwere und Bedeutung, der Entdeckungs- und Auftretenswahrscheinlichkeit, zu beseitigen oder zu entschärfen [DIN06]. Die übrigen Fehlerursachen werden direkt mittels Wahrscheinlichkeitsverteilungen für externe Einflüsse beziehungsweise Fehlerursachen oder mittels lokaler Zuverlässigkeitsschätzer für interne Fehlerursachen berücksichtigt. Externe Einflüsse und Fehlerursachen, wie katastrophale Ereignisse, ausbleibende oder fehlerhafte Instandhaltung, werden über externe Ereignisse dem Systemzuverlässigkeitsmodell hinzugefügt (vergleiche Abschnitt 3.5.2.4). Interne Fehlerursachen, die durch Zufalls-, Verschleiß- und Ermüdungsausfälle (vergleiche Abbildung 2.3) auftreten, werden in der integrierten Modellierung durch lokale Zuverlässigkeitsschätzer abgebildet. Die internen Fehlerursachen stehen im Fokus der lokalen Zuverlässigkeitsschätzer, da hier die Art der Nutzung und damit schließlich das dynamische Verhalten des Systems maßgeblich ist. Erreicht eine Schädigung durch Ermüdung, Verschleiß, Korrosion oder ähnliche Schadensformen eine vorab definierte Versagensgrenze oder ist ein Ausfallkriterium erfüllt, wird die Fehlerursache als eingetreten betrachtet [Ber08] und ihre Auswirkung auf Fehlzustände des Systems innerhalb des Systemzuverlässigkeitsmodells abgebildet.

Die in Abschnitt 3.3 vorgestellten lokalen Zuverlässigkeitsschätzer sind in nahezu beliebigen anwendungsspezifischen oder verallgemeinerten Ausprägungen umsetzbar, sodass eine Implementierungsstrategie für eine flexible Integration notwendig ist. Die Darstellung der Implementierung der lokalen Zuverlässigkeitsschätzer beschränkt sich hier auf zuverlässigkeitsbasierte Verfahren. Die vollständige Integration zustandsbasierter Zuverlässigkeitsschätzer in die Implementierung der integrierten Modellierung setzt definierte Schnittstellen zu dem zu untersuchenden realen System und zu bestehenden Strukturen des Produktlebenszyklusmanagements voraus. Eine solche Integration führt schließlich zu der Umsetzung eines Digitalen Zwillings. Eine vollständige Integration wird im Rahmen dieser Arbeit nicht durchgeführt, das Potenzial zur Umsetzung eines Digitalen Zwillings jedoch in Abschnitt 4.3 aufgezeigt.

Der Anforderung einer flexiblen Integration verschiedenster lokaler Zuverlässigkeitsschätzer folgend, werden die Methoden in Parameterschätzer der Verteilungsfunktion und entsprechende Verteilungsfunktionen der Zuverlässigkeit für die Implementierung modularisiert. Die Parameterschätzer stellen den Zusammenhang zwischen der lokal wirkenden Belastung, Beanspruchung und Schädigung auf Parameter der Verteilungsfunktion der Zuverlässigkeit, wie die charakteristische Lebensdauer der Weibullverteilung, her. Die Verteilungsfunktionen bilden das Ausfallverhalten ab und werden durch die Parameterschätzer parametrisiert.

Die Implementierung der Parameterschätzer orientiert sich wieder am Entwurfsmuster des Beobachters aus der Softwareentwicklung. Eine übergeordnete Klasse *LifetimeEstimators* identifiziert und lokalisiert alle für die integrierte Modellierung zur Verfügung stehenden Parameterschätzer und verwaltet den Datenaustausch mit den weiteren Bestandteilen der automatisierten Ableitung sowie dem Datenobjekt *IntModData*. Die Parameterschätzer sind jeweils in eigenen Klassen implementiert, die sich in ihrer Struktur, Methoden und Attributen gleichen. Daher werden die Parameterschätzer exemplarisch an einer Klasse *Estimator* vorgestellt. In Tabelle 3.6 sind alle gemeinsamen Attribute der Parameterschätzer gezeigt und beschrieben. Anhand der Bezeichnung der Attribute wird die Klasse *Estimator* als Parameterschätzer durch die übergeordnete Klasse *LifetimeEstimators* identifiziert und dem Nutzer in Form einer Liste für die Auswahl eines lokalen Zuverlässigkeitsschätzers bereitgestellt. Die Attribute der Klasse *Estimator* sind anhand ihrer Erzeugung in zwei Kategorien einzuteilen: die durch Nutzereingaben bereitgestellten und die automatisch erzeugten. Neben der Auswahl eines geeigneten lokalen Zuverlässigkeitsschätzers für eine Fehlerursache, ist der Nutzer aufgefordert, diesen mittels des Attributs *EstimatorParameters* zu parametrieren. Der Aufwand der Parametrierung ist stark divergent und abhängig von dem gewählten lokalen Zuverlässigkeitsschätzer. Die Parameter der katalogbasierten Zuverlässigkeitsschätzer sind an Produktqualität, Einsatzbedingungen und weitere Aspekte leicht anzupassende Faktoren. Die Umsetzung einer Betriebsfestigkeitsanalyse erfordert hingegen Erfahrung und Systemkenntnis bei der Auswahl von Methoden und Parametern. Die weiteren Attribute des Parameterschätzers

Klasse.Attribut	Beschreibung
Nutzereingabe	
<i>Estimator.EstimatorParameters</i>	Auflistung der zu setzenden Parameter
Automatisiert erzeugt	
<i>Estimator.DefaultDistribution</i>	Editierbare Standardvorgabe einer Zuverlässigkeitsverteilungsfunktion
<i>Estimator.DefaultDistribution-ParameterValues</i>	Editierbare Standardvorgabe der Parameter der Zuverlässigkeitsverteilungsfunktion in <i>DefaultDistribution</i>
<i>Estimator.EstimatorName</i>	Bezeichnung des lokalen Zuverlässigkeitsschätzers
<i>Estimator.Description</i>	Allgemeine Beschreibung des lokalen Zuverlässigkeitsschätzers
<i>Estimator.T_x</i>	Nominelles Ergebnis eines Lebensdauerschätzers
<i>Estimator.T_{char}</i>	Charakteristische Lebensdauer auf Basis des nominalen Ergebnisses T_x mit Gleichung 3.15
<i>Estimator.MTTF</i>	Mittlere Ausfalldauer auf Basis der gewählten Verteilungsfunktion <i>DefaultDistribution</i>

Tabelle 3.6: Übersicht über die notwendigen Attribute der Klasse *Estimator* eines Parameterschätzers eines beliebigen lokalen Zuverlässigkeitsschätzers.

werden automatisch erzeugt (*EstimatorName*, *Description*, T_x , T_{char} und *MTTF*) oder sind Standardvorgaben (*DefaultDistribution*), deren Änderung durch den Nutzer optional ist. Das Attribut T_x , als Lebensdauer zu einem Zuverlässigkeitswert $x = 1 - R(T_x)$, stellt das Ergebnis des Parameterschätzers dar und wird durch Aufruf der klasseninternen Methoden, die den eigentlichen Berechnungsformalismus des Parameterschätzers bereitstellen, berechnet. Um das Ergebnis des Parameterschätzers für verschiedene Verteilungsfunktionen der Zuverlässigkeit nutzbar zu machen, wird T_x in die charakteristische Lebensdauer T_{char} der Weibull-Verteilung mit $1 - R(T_{char}) = 0,632$ und die mittlere Ausfalldauer *MTTF* für die Exponential-Verteilung mit $1 - R(MTTF) = 0,5$ umgerechnet. Unter der Annahme einer Weibull-Verteilung wird die Lebensdauer T_x in die charakteristische Lebensdauer nach [Ber08] mit

$$T_{char} = \frac{T_x - f_{t_0} T_x}{\sqrt{-\ln(1-x)}} + f_{t_0} T_x \quad (3.15)$$

für $f_{t_0} = t_0/B_{10}$ als faktorielle Darstellung der ausfallfreien Zeit t_0 , dem Formparameter b der Weibull-Verteilung und x als Wert der Ausfallwahrscheinlichkeit, umgerechnet. Die Vernachlässigung des Faktors f_{t_0} und die Annahme $b = 1$ ermöglichen eine Umrechnung der Lebensdauer T_x in die *MTTF* für eine Exponential-Verteilung. Alle Parameterschätzer müssen die Anforderung bezüglich einer einheitlichen Schnittstelle zu den Verteilungsfunktionen erfüllen, sodass die Lebensdauerwerte in Stunden anzugeben sind. Eine Umrechnung der Lebensdauer in andere Bezugsgrößen, wie Anzahl Zyklen, Schwingspiele und andere, erfolgt erst anhand der Systemzuverlässigkeit und des gewählten charakteristischen Manövers. Zusammen mit den Werten im Attribut *DefaultDistributionParameterValues* ist die Verteilungsfunktion der Zuverlässigkeit vollständig definiert und wird an die Klasse *Distribution* übergeben.

Die Verteilungsfunktionen folgen derselben Implementierungsstrategie wie die Parameterschätzer. Eine übergeordnete Klasse *Distributions* identifiziert und verwaltet alle zur Verfügung stehenden Verteilungsfunktionen, die exemplarisch mittels der Klasse *Distribution* abgebildet werden. Die Klasse *Distribution* repräsentiert wieder die Gleichanteile innerhalb der Struktur und den Attributen. In Tabelle 3.7 sind die gemeinsamen Attribute der Verteilungsfunktionen dargestellt. Im Unterschied zu dem Parameterschätzer sind keine direkten Nutzereingaben bei Verwendung der Standardvorgabe zu einer Verteilungsfunktion innerhalb eines Parameterschätzers vorgesehen, da alle notwendigen Parameter bereits in *Estimator.DefaultDistributionParameterValues* / T_x / T_{char} / *MTTF* zur Verfügung stehen. Dem gegenüber erfordert die Modellierung einer externen Fehlerursache, wie eines zusätzlichen Ereignisses, Nutzereingaben für die Parameter der Verteilungsfunktion, da diese nicht über einen Parameterschätzer vorgegeben werden. Die maßgeblichen Ergebnisse der Verteilungsfunktion *Distribution* sind die lokalen Zuverlässigkeiten R beziehungsweise deren Komplement, die Ausfallwahrscheinlichkeit F , als Eintrittswahrscheinlichkeit einer Fehlerursache. Die bedingten Zuverlässigkeiten *condR* und die bedingten Ausfallwahrscheinlichkeiten *condF* sind jeweils über dem Vektor der Betrachtungszeit tR gegeben. Die bedingten Zuverlässigkeiten beschreiben dabei die Abhängigkeit der Zuverlässigkeit über einen Zeitschritt Δt für eine Variable A als $P(A_{t+\Delta t}|A_t)$ für das Systemzuverlässigkeitsmodell. Die weiteren Ergebnisse werden in den Attributen f für die Zuverlässigkeitsdichtefunktion, E für den Mittelwert, Var für die Varianz und $lambda$ für die Ausfallrate in Abhängigkeit der Betrachtungsdauer bereitgestellt.

Klasse.Attribut	Beschreibung
<i>Distribution.DisParameters</i>	Auflistung der zu setzenden Parameter
<i>Distribution.DisName</i>	Bezeichnung des lokalen Zuverlässigkeitsschätzers
<i>Distribution.Description</i>	Allgemeine Beschreibung der Zuverlässigkeitsverteilungsfunktion
<i>Distribution.tR</i>	Zeitvektor der Betrachtungszeit
<i>Distribution.tR_unit</i>	Einheit der Betrachtungszeit
<i>Distribution.R / .F</i>	Zuverlässigkeit/Ausfallwahrscheinlichkeit in Abhängigkeit der Betrachtungszeit <i>Distribution.tR</i>
<i>Distribution.condR / .condF</i>	Bedingte Zuverlässigkeit/Ausfallwahrscheinlichkeit in Abhängigkeit der Betrachtungszeit <i>Distribution.tR</i> $P(A_{t+\Delta t} A_t)$
<i>Distribution.f</i>	Zuverlässigkeitsdichtefunktion in Abhängigkeit der Betrachtungszeit <i>Distribution.tR</i>
<i>Distribution.E</i>	Mittelwert
<i>Distribution.Var</i>	Varianz
<i>Distribution.lambda</i>	Ausfallrate in Abhängigkeit der Betrachtungszeit <i>Distribution.tR</i>

Tabelle 3.7: Übersicht über die notwendigen Attribute der Klasse *Distribution* einer beliebigen Zuverlässigkeitsverteilungsfunktion zur funktionsunabhängigen Beschreibung.

Die vorgestellten Klassen der Parameterschätzer *Estimator* und der Verteilungsfunktionen der Zuverlässigkeit *Distribution* werden für konkrete Zuverlässigkeitsschätzer instanziiert. Eine Instanz der Klassen beschreibt also mittels der zugewiesenen Parameter eine spezifische Fehlerursache des Systems (vergleiche [GHJ+05] für Instanziierung von Klassen). Die Instanzen eines Parameterschätzers und einer Verteilungsfunktion bilden somit die Instanz eines lokalen Zuverlässigkeitsschätzers, die vollständig oder in Teilen für weitere Fehlerursachen verwendet werden kann.

3.5.2.2 Erzeugung des generalisierten DAG

Die Erzeugung der Struktur des Systemzuverlässigkeitsmodells beginnt mit der Erzeugung eines generalisierten DAG und beschreibt die Transformation des Topologiegraphen in einen vorläufigen DAG des Systemzuverlässigkeitsmodells. Dieser weist jedoch noch überzählige Knoten und nicht die gewünschte Modellierungstiefe auf. Der Aufbau des generalisierten DAG aus dem Topologiegraphen gliedert sich in drei Teilbereiche:

- Auflösung von Zyklen,
- Erweiterung um Zeitscheiben,
- Hinzufügen von Fehlerursachen und Ausfallzuständen.

Der Topologiegraph des zu untersuchenden Systems wird zunächst auf Zyklen untersucht. Die Detektion von Zyklen erfolgt mittels der Zentralität des Graphen. Die Zentralität bewertet den Abstand aller Knoten zueinander. Demnach gilt ein Knoten als zentral, wenn der größte Abstand zu weiteren Knoten möglichst klein ist. Der Abstand zwischen Knoten ist für Graphen ohne Kantengewichte, wie der Topologiegraph \mathbf{S}_{Topo} in Abbildung 3.25 für den $s_{j,l} = 1 \forall j, l$ gilt, gleich der Anzahl an Kanten zwischen zwei beliebigen Knoten [Die17]. Die Zentralität eines Graphen kann mittels des Exponentials seiner Adjazenzmatrix \mathbf{A} berechnet werden und ist durch die folgende Potenzreihe definiert [NFB18]:

$$e^{\mathbf{A}} = \mathbf{E} + \mathbf{A} + \frac{\mathbf{A}^2}{2!} + \cdots + \frac{\mathbf{A}^k}{k!} + \cdots = \sum_{k=0}^{\infty} \frac{\mathbf{A}^k}{k!} \quad (3.16)$$

mit der Einheitsmatrix \mathbf{E} . Algorithmen zur effizienten Berechnung der Potenzreihe 3.16 sind in [Hig05] auf Basis der Padé Approximation beschrieben. Die Einträge des Matrixexponentials $e^{\mathbf{A}}(j, l) > 1$ entsprechen einer Kante (j, l) in der Adjazenzmatrix \mathbf{A} . Gilt für die Diagonalelemente des Matrixexponentials

$$e^{\mathbf{A}}(j, j) > 1 \quad (3.17)$$

so existiert ein Pfad, über möglicherweise mehrere Kanten und Knoten, für den Anfangs- und Endknoten identisch sind und dem j -ten Knoten des Graphen entspricht [NFB18]. Die Anwendung dieses Zusammenhangs in Gleichung 3.17 auf die Adjazenzmatrix des Topologiegraphen \mathbf{S}_{Topo} in Gleichung 3.14 liefert $j=2 \wedge 3$, demnach sind die Knoten R und S Teil eines Zyklus. Anschaulich ist der Topologiegraph \mathbf{S}_{Topo} azyklisch, wenn eine beliebige Reihenfolge der Knoten existiert, sodass für die obere Dreiecksmatrix $j < l \Rightarrow s_{j,l} = 0$ gilt. Für den Topologiegraphen \mathbf{S}_{Topo} existiert eine solche Reihenfolge nicht, da über die Knoten R und S ein Zyklus enthalten ist. Dieser Zyklus bildet einen Teil der funktionalen Zusammenhänge zwischen den Komponenten und Teilsystemen ab und kann daher nicht vernachlässigt werden. Der gewählte Ansatz zum Auflösen zyklischer Strukturen im Topologiegraphen sieht allgemein die Transformation in einen DAG vor. Der DAG beinhaltet die Ausfallzustände des Systems sowie die an einem Zyklus beteiligten Kanten als temporale Abhängigkeiten innerhalb des DAG. Dazu sind nach der Transformation drei Vorgehen für unterschiedliche Ausprägungen von Zyklen vorgesehen.

Das Systemzuverlässigkeitsmodell bildet das zu untersuchende System in Zeitscheiben, Instanzen des Systems zu diskreten Zeitpunkten der Betriebsdauer (siehe Abschnitt 3.4.3.1), ab. Der dazugehörige DAG mit einer $m \times 2m$ Adjazenzmatrix $\mathbf{A}_{DAG,i} = (a_{j,l})$ ist für jede Zeitscheibe i nach folgender Struktur aufgebaut:

$$\mathbf{A}_{DAG,i} = \left[\mathbf{A}_{intra,t_i} \mid \mathbf{A}_{inter,t_i,t_{i+1}} \right] \quad (3.18)$$

mit der $m \times m$ Adjazenzmatrix des Intra-DAG \mathbf{A}_{intra,t_i} , die den Graphen innerhalb einer Zeitscheibe zum Zeitpunkt t_i abbildet und der $m \times m$ Adjazenzmatrix des Inter-DAG $\mathbf{A}_{inter,t_i,t_{i+1}}$, die den Graphen zwischen zwei Zeitscheiben t_i und t_{i+1} darstellt. Für die letzte Zeitscheibe $i = n$ wird der Inter-DAG $\mathbf{A}_{inter,t_i,t_{i+1}}$ nicht benötigt, demnach ist die Adjazenzmatrix $\mathbf{A}_{DAG,n} = \mathbf{A}_{intra,t_n}$ der Größe $m \times m$. Die Annahme strukturell zeitinvarianter Zeitscheiben ergibt $\mathbf{A}_{DAG,i} = \mathbf{A}_{DAG,i+1}$ mit einer Ausnahme für $i = n$. Der Topologiegraph wird als Intra-DAG $\mathbf{A}_{intra,t_i} = \mathbf{S}_{Topo}$ verwendet, die Adjazenzmatrix des Inter-DAG ist zunächst eine Null-Matrix $\mathbf{A}_{inter,t_i,t_{i+1}} = \mathbf{0}$.

Die drei Ausprägungen von Zyklen innerhalb des Intra-DAG \mathbf{A}_{intra,t_i} werden nacheinander aufgelöst. *Selbstreferenzierende Kanten*, Start- und Endknoten ist identisch, beinhalten keine für das Systemzuverlässigkeitsmodell relevanten Informationen und werden aus dem Topologiegraphen gelöscht. Diese Kanten treten als Einträge der Hauptdiagonalen in der Adjazenzmatrix im Intra-DAG \mathbf{A}_{intra,t_i} mit $a_{j,j} = 1$ auf und werden zu $a_{j,j} = 0 \forall j$ gesetzt. *Bidirektionale Kanten* treten bei Zyklen auf, bei denen zwischen zwei direkt benachbarten Knoten Kanten existieren (vergleiche Knoten R und S im \mathbf{S}_{Topo} in Abbildung 3.25). In der Adjazenzmatrix des Intra-DAG \mathbf{A}_{intra,t_i} treten bidirektionale Kanten als symmetrische Einträge $a_{j,l} = a_{l,j}$ auf. Diese symmetrischen Einträge des Intra-DAG werden gelöscht mit $a_{j,l} = a_{l,j} = 0$ und in den Inter-DAG $\mathbf{A}_{inter,t_{i+1}}$ innerhalb der Adjazenzmatrix $A_{DAG,i}$ verschoben $a_{j,l+m} = a_{l,j+m} = 1$. Daraus ergibt sich nach Gleichung 3.18 die folgende Adjazenzmatrix für den Zeitschritt t_i :

$$\mathbf{A}_{DAG,i} = \begin{array}{c|cccccc|cccc} & Sys_i & R_i & S_i & G_i & A_i & Sys_{i+1} & R_{i+1} & S_{i+1} & G_{i+1} & A_{i+1} \\ \hline Sys_i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ R_i & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ S_i & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ G_i & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A_i & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \quad (3.19)$$

mit der äquivalenten Darstellung in Abbildung 3.26. Aus Gründen einer konsistenten Darstellung über zwei Zeitscheiben sind die Kanten des Intra-Grapens $\mathbf{A}_{intra,t_{i+1}}$ ausgegraut abgebildet. Aufgrund von Rückkopplungen innerhalb des zu untersuchenden Systems können *komplexe Zyklen* mit verschachtelten Zyklen über mehrere Knoten auftreten. Die Identifikation und Auflösung erfolgt iterativ über die enthaltenen Zyklen. Zyklen im Inter-DAG $\mathbf{A}_{inter,t_{i,i+1}}$ sind ausgeschlossen, da dieser gezielt algorithmisch aufgebaut wird. Mittels des Matrixexponentials wird detektiert, ob nach den zwei Maßnahmen zur Auflösung von Zyklen noch weitere im Intra-DAG \mathbf{A}_{intra,t_i} verblieben sind. Mittels Gleichung 3.17 wird der erste Knoten j , der Teil eines Zyklus ist, identifiziert und alle Elternknoten betrachtet. Die Elternknoten des j -ten Knotens werden nacheinander geprüft, ob diese Teil eines Zyklus sind (Gleichung 3.17). Ist der aktuell geprüfte Elternknoten Teil eines Zyklus, wird die Kante zum j -ten Knoten im Intra-DAG \mathbf{A}_{intra,t_i} gelöscht und in den Inter-DAG $\mathbf{A}_{inter,t_{i,i+1}}$ als temporale Abhängigkeit hinzugefügt. Der Ansatz ist im Anhang A.3 als Pseudocode in Algorithmus 1.1 dargestellt.

Im Intra-DAG \mathbf{A}_{intra,t_i} existieren keine selbstreferenzierenden oder komplexen Zyklen, sodass sich nach Auflösen der bidirektionalen Abhängigkeit zwischen den Knoten R und S der Graph $\mathbf{A}_{DAG,i}$ in Abbildung 3.26 analog zur Adjazenzmatrix 3.19 ergibt.

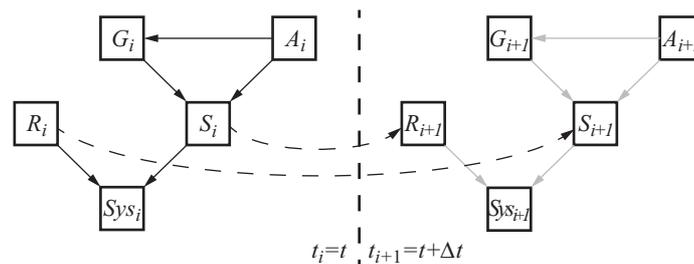


Abbildung 3.26: DAG der Ausfallzustände mit Adjazenzmatrix $\mathbf{A}_{DAG,i}$ in Gleichung 3.19 aus Topologiegraph in Abbildung 3.25.

Der DAG in Abbildung 3.26 ist ein Graph der Ausfallzustände des zu untersuchenden Systems. Zur Abbildung der Systemzuverlässigkeit fehlen noch n_U Fehlerursachen, die in Form lokaler Zuverlässigkeitsschätzer den Komponenten und Teilsystemen zugeordnet sind. Für das Beispielsystem seien $n_U = 3$ lokale Zuverlässigkeitsschätzer ausgewählt, parametrisiert und berechnet worden, sodass n_U Datensätze, wie in Tabellen 3.6 und 3.7 dargestellt, verfügbar sind. So seien für Knoten R ein und für Knoten G zwei Zuverlässigkeitsschätzer definiert und als Fehlerursachen W für Ausfallzustand R und Fehlerursachen U und V für Ausfallzustand G dem Graphen $\mathbf{A}_{DAG,i}$ hinzugefügt worden. Die Fehlerursachen werden dem Modellierungsansatz folgend mit einer Kante (*Fehlerursache, Ausfallzustand*) zu ihrem Ausfallzustand im Intra-DAG \mathbf{A}_{intra,t_i} verbunden. Darüber hinaus werden temporale Abhängigkeiten als Kanten (*Fehlerursache in t_i , Fehlerursache in t_{i+1}*) im Inter-DAG $\mathbf{A}_{inter,t_i,t_{i+1}}$ hinzugefügt. Damit ändert sich auch die Größe der Adjazenzmatrix $\mathbf{A}_{DAG,i}$ zu $m_v \times 2m_v$ mit $m_v = m + n_U$. Der resultierende verallgemeinerte DAG des Systemzuverlässigkeitsmodells ist in Abbildung 3.27 gezeigt.

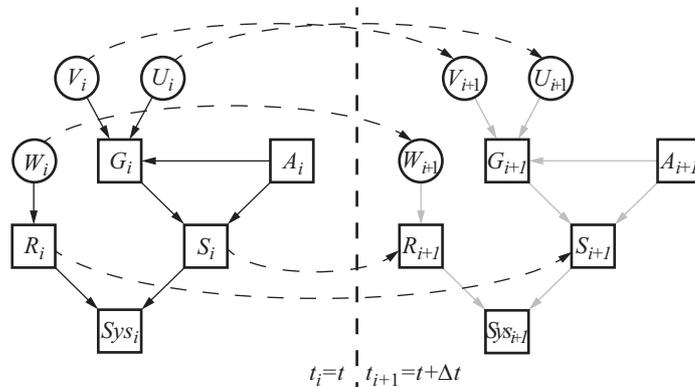


Abbildung 3.27: DAG der Ausfallzustände aus Abbildung 3.26 um Fehlerursachen erweitert.

3.5.2.3 Reduzierung des generalisierten DAG

Der verallgemeinerte DAG $\mathbf{A}_{DAG,i}$ ist als Systemzuverlässigkeitsmodell verwendbar, beinhaltet jedoch noch überzählige Knoten und Kanten, um die der Graph reduziert werden kann. Das Vorgehen ist, basierend auf den zu reduzierenden Elementen, in zwei Kategorien einteilbar:

- Ausfallzustände ohne Elternknoten,
- Knoten ohne Einfluss auf den Zielknoten.

Ausfallzustände ohne Elternknoten besitzen keine Fehlerursache und sind von keinem Ausfallzustand abhängig, sodass sie auf unterster Ebene in die Hierarchie (vergleiche Abschnitt 3.5.1.2) eingebunden sind. Ein Ausfallzustand ohne Fehlerursache oder Beitrag zu hierarchischen Abhängigkeiten beinhaltet keine für die Systemzuverlässigkeit relevanten Informationen und wird vernachlässigt. Der Ausfallzustand $A_{i/i+1}$ erfüllt diese Bedingungen und wird zusammen mit allen von ihm ausgehenden Kanten (A_i, G_i) , (A_i, S_i) , (A_{i+1}, G_{i+1}) und (A_{i+1}, S_{i+1}) aus dem Graphen gelöscht. Weitere Informationen innerhalb des Datenobjekts *IntModData* werden ebenfalls gelöscht. Die Adjazenzmatrix $\mathbf{A}_{DAG,i}$ wird um die Anzahl an Ausfallzuständen ohne Elternknoten n_A reduziert und entspricht demnach einer $m_{r1} \times 2m_{r1}$ -Matrix mit $m_{r1} = m_v - n_A$. Um eine effiziente

Berechnung des späteren Systemzuverlässigkeitsmodells zu ermöglichen, wird ein Index des Zielknotens j_{target} definiert, der das Analyseziel darstellt. Dieser Knoten tritt in jeder Zeitscheibe auf. *Knoten ohne Einfluss auf den Zielknoten* haben dementsprechend keinen Einfluss auf das Analyseziel und werden vernachlässigt. Unter dieser Annahme werden zudem Teilgraphen ohne Einfluss auf das Analyseziel detektiert und gelöscht. Dazu werden die Teilgraphen Intra-DAG \mathbf{A}_{intra,t_i} und Inter-DAG $\mathbf{A}_{inter,t_i,t_{i+1}}$ der Adjazenzmatrix $\mathbf{A}_{DAG,i}$ addiert und die Elemente der Hauptdiagonalen in der resultierenden Adjazenzmatrix \mathbf{A}_{ges} zu Null gesetzt. Die Addition ist notwendig, um temporale Abhängigkeiten, die nur im Inter-DAG enthalten sind, zu berücksichtigen. Mittels des Matrixexponentials $e^{\mathbf{A}_{ges}}$ wird die Erreichbarkeit aller Knoten zum Zielknoten j_{target} geprüft. Allen n_e Knoten, denen kein Einfluss nachgewiesen wird, werden aus dem Datenobjekt *IntModData* und der Adjazenzmatrix $\mathbf{A}_{DAG,i}$ gelöscht, sodass diese zu einer $m_{r2} \times 2m_{r2}$ -Matrix mit $m_{r2} = m_{r1} - n_e$ reduziert wird. Das Vorgehen ist im Anhang A.3 als Pseudocode in Algorithmus 1.2 dargestellt.

Nach der Durchführung der zwei Vorgehen zur Reduzierung des verallgemeinerten DAG $\mathbf{A}_{DAG,i}$, ergibt sich der Graph eines Systemzuverlässigkeitsmodells, wie in Abbildung 3.28 für das Beispielsystem in Abbildung 3.25 dargestellt.

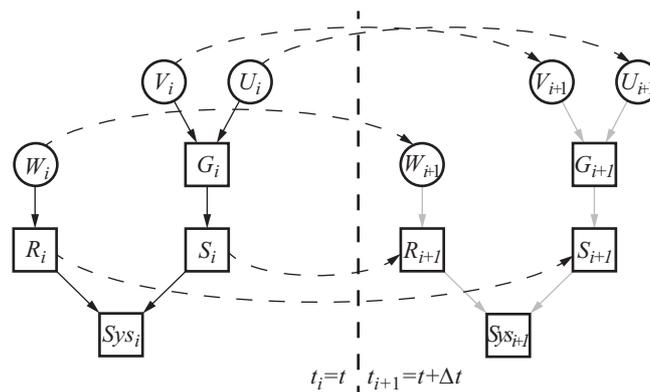


Abbildung 3.28: DAG aus Abbildung 3.27 um Ausfallzustand $A_{i/i+1}$ reduziert.

3.5.2.4 Hinzufügen weiterer Modellierungsaspekte

Der reduzierte Graph des Systemzuverlässigkeitsmodells in Abbildung 3.28 kann zur detaillierten Abbildung der Systemzuverlässigkeit um weitere Modellierungsaspekte erweitert werden:

- zusätzliche Abhängigkeiten zwischen bestehenden Ausfallzuständen,
- externe Ereignisse,
- Ausfälle gemeinsamer Ursache,
- redundante Teilsysteme und Komponenten.

Die beschriebenen Modellierungsaspekte weisen Schnittmengen auf, so sind zusätzliche Abhängigkeiten und externe Ereignisse teilweise notwendige Bestandteile der Modellierung von Ausfällen gemeinsamer Ursache sowie redundanter Teilsysteme und Komponenten. Die Erweiterung um Modellierungsaspekte erfolgt auf Basis von Nutzereingaben, da das Systemmodell nur Indizien für ihre Existenz liefert.

Die Ergänzung *zusätzlicher Abhängigkeiten* ist auf bestehende Ausfallzustände beschränkt, da Fehlerursachen nur auf den dazugehörigen Ausfallzustand verweisen dürfen. Zusätzliche Abhängigkeiten werden als Kanten dem Intra-DAG \mathbf{A}_{intra,t_i} oder dem Inter-DAG $\mathbf{A}_{inter,t_i,t_{i+1}}$ der Adjazenzmatrix $\mathbf{A}_{DAG,i}$ in allen Zeitscheiben $i \in \{0, \dots, n\}$ hinzugefügt. Jede Erweiterung um eine Kante wird nach Gleichungen 3.16 und 3.17 geprüft, ob durch diese ein Zyklus entsteht, gegebenenfalls wird die Erweiterung verweigert.

Die Erweiterung des Systemzuverlässigkeitsmodells um *externe Ereignisse* stellt einen weiteren Modellierungsaspekt dar. Externe Ereignisse treten als externe Fehlerursache auf und sind demnach mit der Definition eines lokalen Zuverlässigkeitsschätzers verbunden. Externe Ereignisse weisen in der Regel keinen direkten Einfluss des dynamischen Verhaltens des zu untersuchenden Systems auf, sodass der Zuverlässigkeitsschätzer meist über eine Wahrscheinlichkeitsverteilung ohne Parameterschätzer modelliert wird. Die Forderung, dass externe Ereignisse als Fehlerursachen modelliert werden, führt dazu, dass diese dem Modellierungsansatz folgend über temporale Abhängigkeiten verbunden werden. Neben den temporalen Abhängigkeiten muss eine Kante (*Externes Ereignis, Ausfallzustand*) zu einem Ausfallzustand definiert werden, da sonst kein Einfluss auf den Zielknoten j_{target} möglich wäre. Die Adjazenzmatrix $\mathbf{A}_{DAG,i}$ wird demnach von $m_{r2} \times 2m_{r2}$ auf $m_z \times 2m_z$ mit $m_z = m_{r2} + z$ für z zusätzliche Ereignisse erweitert. Das Beispielsystem mit dem Graphen $\mathbf{A}_{DAG,i}$ in Abbildung 3.28 wird um das externe Ereignis mit den Knoten E_i und E_{i+1} , mit den Kanten (E_i, G_i) und (E_{i+1}, G_{i+1}) im Intra-DAG sowie der Kante (E_i, E_{i+1}) im Inter-DAG erweitert. Daraus ergibt sich der Graph in Abbildung 3.29.

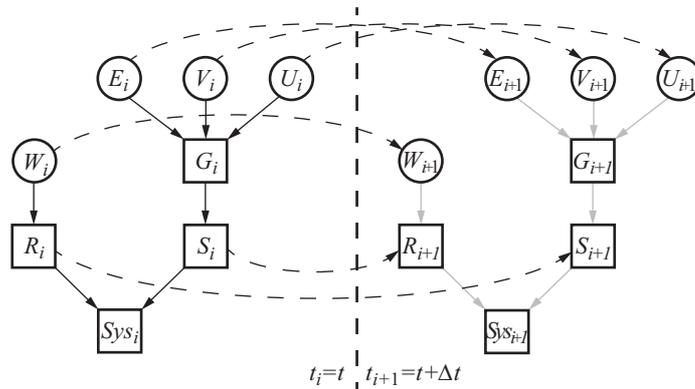


Abbildung 3.29: Erweiterung des reduzierten DAG aus Abbildung 3.28 um externes Ereignis $E_{i/i+1}$ als Fehlerursache des Ausfallzustands $R_{i/i+1}$.

Ausfälle gemeinsamer Fehlerursache werden anhand einer bestehenden internen oder externen Fehlerursache modelliert. Neben der Auswahl einer Fehlerursache muss durch den Nutzer mindestens eine Kante zu einem weiteren Ausfallzustand (*gemeinsame Ursache, weiterer Ausfallzustand*) definiert werden, um die Auswirkung einer Fehlerursache auf mindestens zwei Komponenten oder Teilsysteme abzubilden. Die zusätzliche Kante wird dem Intra-DAG \mathbf{A}_{intra,t_i} hinzugefügt. Im Beispielsystem $\mathbf{A}_{DAG,i}$ soll das hinzugefügte externe Ereignis E die gemeinsame Fehlerursache der Ausfallzustände G und R sein. Demnach ergibt sich der in Abbildung 3.30 gezeigte Graph.

Redundante Teilsysteme und Komponenten werden anhand bestehender Ausfallzustände und Fehlerursachen modelliert. Eine Modellierung anhand hinzugefügter Ereignisse ist unzulässig, da für die Modellierung der Redundanz das dynamische Systemverhalten maßgeblich ist und zudem das Hinzufügen von Ausfallzuständen nicht vorgesehen

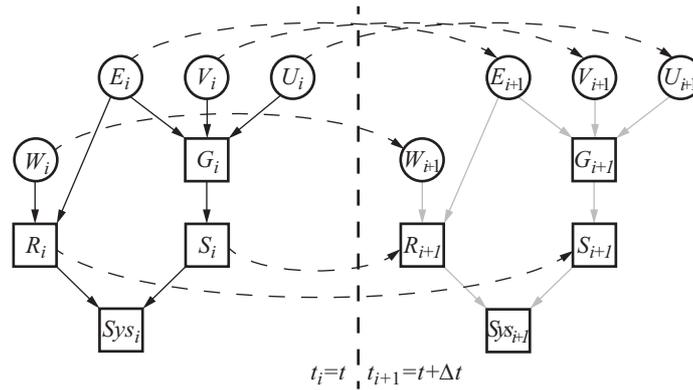


Abbildung 3.30: Erweiterung des reduzierten DAG aus Abbildung 3.28 um externes Ereignis $E_{i/i+1}$ als gemeinsame Fehlerursache der Ausfallzustände $R_{i/i+1}$ und $G_{i/i+1}$.

ist. Die Modellierung von Redundanzen basiert auf Nutzereingaben zur Auswahl der beteiligten Komponenten und Teilsysteme, deren beeinflusste Fehlerursachen sowie der Redundanzart und -ordnung, wobei diversitäre Redundanzen akzeptiert werden. Interne Fehlerursachen sind von der redundanten Auslegung betroffen, jedoch ist dieser Einfluss auf Ursachen mit einer Abhängigkeit zum dynamischem Systemverhalten verbunden. Jede Komponente und jedes Teilsystem der Redundanz ist mit dem Ausfallzustand und mindestens einer Fehlerursache beteiligt. Redundanzarten mit Lastteilung sind aktuell nicht umgesetzt, da hier die Zuverlässigkeit und das dynamische Verhalten des zu untersuchenden Systems in gegenseitiger Abhängigkeit stehen. Sinkt die Anzahl aktiver Komponenten oder Teilsysteme durch Ausfälle, steigt im Gegenzug die auf die verbliebenen aktiven Elemente wirkende Last mit Auswirkungen auf die lokale Zuverlässigkeit. Der Ausfall einer Komponente oder eines Teilsystems wird erst über das Systemzuverlässigkeitsmodells bestimmt, sodass entgegen des aktuellen Informationsflusses, vom Systemmodell über die automatisierte Ableitung zum Systemzuverlässigkeitsmodell, eine Rückführung vom Systemzuverlässigkeitsmodell zum Systemmodell erfolgen müsste, um eine Änderung der Konfiguration der Redundanz zu initiieren. Eine solche Rückführung widerspricht dem für diese Arbeit aufgestellten Modellierungsansatz, dass bestehende Modelle des Entwurfs mechatronischer Systeme durch die integrierte Modellierung nicht verändert werden, um eine möglichst große Akzeptanz zu erreichen (vergleiche Einführung Abschnitt 3). Diese Forderung schränkt die Modellierungstiefe ein, setzt aber eine Schnittstelle zwischen Systemzuverlässigkeits- und Systemmodell voraus, die ohne Änderungen am Systemmodell nicht robust umsetzbar erscheint. Ein Vorschlag für die Umsetzung der Modellierung von Redundanzen mit Lastteilung in dynamischen Bayes'schen Netzen ist in [KMS16] gezeigt, wobei die Ausfallraten der Fehlerursachen faktoriell angepasst werden, um Lastteilung abzubilden. Die Abbildung redundanter (Teil-)Systeme erfordert neben der Auswahl der redundant ausgelegten Komponenten und Teilsystemen auch die Definition eines übergeordneten Teilsystems, dessen Ausfallzustand die konjunkte Verknüpfung der redundanten Elemente enthält (vergleiche Knoten D_i in Abbildungen 3.20 und 3.21). Ein solcher Ausfallzustand kann, aus dem Modellierungsansatz heraus, nicht dem System hinzugefügt werden. Ein entsprechendes Teilsystem sollte im Systemmodell jedoch stets abgebildet sein, da alle redundanten Elemente dieselbe Funktion bereitstellen und diese erst über das Teilsystem im Gesamtsystem integriert ist.

Die Integration *heißer Redundanz ohne Lastteilung* erfordert keine Änderungen am DAG des Systemmodells $\mathbf{A}_{DAG,i}$, da keine Abhängigkeiten zwischen den beteiligten Fehlerursachen und Ausfallzuständen bestehen. Es werden zunächst nur die beteiligten Komponenten sowie Teilsysteme ausgewählt und im Datenobjekt *IntModData* abgelegt. Die Erweiterung des DAG $\mathbf{A}_{DAG,i}$ um eine *warme* oder *kalte k-aus-m* Redundanz erfordert $m(m-1)$ zusätzliche Kanten. Für alle $m(m-1)$ Kanten (*Ausfallzustand* $_{u,i}$, *Fehlerursache* $_{i+1,v}$) gelte $u \neq v$ mit $\forall(u, v) \in \{1, \dots, m\}$ für zeitinvariante DAG (vergleiche Abbildung 3.21). Dem folgend verweisen alle Kanten von den beteiligten Ausfallzuständen in Zeitscheibe i zu den entsprechenden Fehlerursachen in Zeitscheibe $i+1$, jedoch nicht zu der Fehlerursache oder den Fehlerursachen, die dem Ausfallzustand als Ursprung der Kante zugeordnet sind. Die Änderung des DAG für redundante Komponenten und Teilsysteme ist damit abgeschlossen. Eine illustrierende Integration in das Beispielsystem eines Regelkreises ist in der gewählten Darstellungstiefe nicht sinnvoll umsetzbar.

3.5.2.5 Erzeugung der CPTs

Die Erzeugung der CPTs des Systemzuverlässigkeitsmodells erfolgt nach Abschluss der Erzeugung des DAG, da die Anzahl an Elternknoten (definiert im DAG) neben den Zuständen jedes Knotens maßgeblich für die Größe der CPTs ist. Die CPTs werden im Datenobjekt *IntModData* für jede Zeitscheibe i gespeichert. Im Gegensatz zum DAG $\mathbf{A}_{DAG,i}$ sind die CPTs zeitvariant, um die Verteilungsfunktionen der Eintrittswahrscheinlichkeit der Fehlerursachen (vergleiche bedingte Wahrscheinlichkeit der Weibullverteilung in Abhängigkeit der absoluten Betrachtungszeit in Gleichung 3.12) abzubilden. Demnach müssen für die Anzahl an Knoten m_z in einer Zeitscheibe i und der Anzahl an Zeitscheiben $n+1$ insgesamt $m_z(n+1)$ CPTs im Systemzuverlässigkeitsmodell erzeugt werden.

Die Grundlage der Erzeugung von CPTs der Fehlerursachen bilden die Annahmen des allgemeinen Modellierungsansatzes. Demnach sind die Fehlerursachen ausschließlich über die Zustände $\{active, dormant\}$ definiert und werden auf Basis der Wahrscheinlichkeiten der lokalen Zuverlässigkeitsschätzer parametrisiert. Direkte Änderungen an den Zuständen und den Einträgen der CPTs der Fehlerursachen, wie die Berücksichtigung von Unsicherheiten, sind nicht zulässig, sodass Änderungen nur implizit über das Hinzufügen weiterer Modellierungsaspekte, wie Ausfälle gemeinsamer Ursache und redundante Komponenten sowie Teilsysteme, umsetzbar sind. Eine Übersicht über die CPTs der Fehlerursachen ist in Tabelle 3.8 exemplarisch an einer Fehlerursache U gegeben.

Die Definitionen der CPTs der Fehlerursachen sind in die Zeitscheiben t_0 und t_i untergliedert: In Zeitscheibe t_0 weist die Fehlerursache U_0 keine Elternknoten auf, in Zeitscheibe t_i sind temporale Abhängigkeiten (U_0, U_i) vorhanden, wobei für Redundanzen komplexere temporale Abhängigkeiten modelliert werden. Die CPTs der Fehlerursachen ohne weitere Modellierungsaspekte in Zeitscheiben t_0 und t_i sind im Rahmen des allgemeinen Modellierungsansatzes bereits beschrieben worden (Abschnitt 3.4.3.1). Dazu werden die Ergebnisse der lokalen Zuverlässigkeitsschätzer herangezogen, wobei $R_U(t_0)$ die lokale Zuverlässigkeit in Zeitscheibe t_0 beschreibt und $R_U(t_i|t_{i-1})$ die bedingte Zuverlässigkeit in Zeitscheibe t_i in Abhängigkeit einer vorhergegangenen Zeitscheibe t_{i-1} (vergleiche Gleichungen 2.4 und 3.12).

Die Definition der CPT einer gemeinsamen Fehlerursache in Zeitscheibe t_0 ist für ein statisches Bayes'sches Netz bereits in Abschnitt 3.4.3.2 diskutiert worden. Diese Definition wird für die CPT in Zeitscheibe t_i um die Annahme erweitert, dass eine Aktivie-

Zeit- scheibe	Modellie- rungsaspekt	CPT der Fehlerursachen		
	Standard	$\frac{P(U_0 = active)}{1 - R_U(t_0)}$	$\frac{P(U_0 = dormant)}{R_U(t_0)}$	
t_0	Ausfälle gemeinsamer Ursache	$\frac{P(U_0 = activeI)}{(1-\eta)(1-R_U(t_0))}$	$\frac{P(U_0 = activeC)}{\eta(1-R_U(t_0))}$	$\frac{P(U_0 = dormant)}{R_U(t_0)}$
	Redundanz	$\frac{P(U_0 = active)}{1 - R_U(t_0)}$	$\frac{P(U_0 = dormant)}{R_U(t_0)}$	
	Standard	$\frac{P(U_i = active)}{P(U_i = dormant)}$	$\begin{array}{ c} active \\ dormant \end{array}$	$\begin{array}{ c} 1 \\ 0 \end{array}$
				$\begin{array}{ c} 1 - R_U(t_i t_{i-1}) \\ R_U(t_i t_{i-1}) \end{array}$
t_i	Ausfälle gemeinsamer Ursache	$\frac{P(U_i = activeI)}{P(U_i = activeC)}$	$\begin{array}{ c} activeI \\ activeC \end{array}$	$\begin{array}{ c} dormant \\ \end{array}$
				$\begin{array}{ c} (1-\eta)(1-R_U(t_i t_{i-1})) \\ \eta(1-R_U(t_i t_{i-1})) \\ R_U(t_i t_{i-1}) \end{array}$
	Redundanz	Heiß, ohne Lastteilung: siehe Standard in t_i . Alle weiteren Varianten, siehe Tabelle 3.9.		

Tabelle 3.8: Übersicht über CPTs der Fehlerursachen an einem Beispielknoten U für verschiedene Modellierungsaspekte.

rung der individuellen oder gemeinsamen Fehlerursache, entspricht den Zuständen *activeI* und *activeC*, sich über die Zeitscheiben hinweg mit $P(U_i = activeI|activeI) = 1$ und $P(U_i = activeC|activeC) = 1$ fortpflanzt. Für die nicht aktivierte Fehlerursache $U_0 = dormant$ werden die lokalen Zuverlässigkeiten mit den Zusammenhängen wie in Zeitscheibe t_0 für Ausfälle gemeinsamer Ursache angewandt.

Die Fehlerursachen k -aus- m redundanter Komponenten oder Teilsysteme in Zeitscheibe t_0 sind als normale CPTs der Fehlerursachen modelliert, da keine Abhängigkeiten zu Elternknoten existieren. In Zeitscheibe t_i wird zwischen verschiedenen Modellierungsansätzen zur Abbildung verschiedener Redundanzarten (vergleiche Abbildung 3.4.3.3) unterschieden. Fehlerursachen von Komponenten oder Teilsystemen in heißen redundanten Systemen ohne Lastteilung weisen keine Abhängigkeiten untereinander auf und werden daher als normale Fehlerursache modelliert. Alle weiteren komplexen Redundanzarten weisen Abhängigkeiten der Fehlerursachen untereinander auf und werden nach dem Modellierungsansatz in Abbildung 3.21 abgebildet. Dies erfordert eine CPT der Fehlerursache U_i in Abhängigkeit der beteiligten Ausfallzustände (Elternknoten $1, \dots, m$) und der Fehlerursache U_0 aus Zeitscheibe t_0 , wie in folgender Tabelle 3.9 gezeigt.

Die Definition der CPT erfordert die Unterscheidung verschiedener Ausfallszenarien der k -aus- m Redundanz, wobei n_{Aus} die Anzahl an ausgefallenen Komponenten oder Teilsystemen innerhalb der Redundanz beschreibt. Für $n_{Aus} = 0$ liegen keine Ausfälle vor und die Fehlerursache U ist in einer vorherigen Zeitscheibe t_{i-1} nicht aktiviert worden,

	$n_{Aus}=0$	$n_{Aus} \leq (m-k)$	$n_{Aus} > (m-k)$	U_{i-1}
Elternknoten 1	<i>op</i>	<i>op</i> ... <i>op</i>	<i>op</i> ... <i>fa</i>	<i>op</i> ... <i>fa</i>
⋮	⋮	⋮	⋮	⋮
Elternknoten m	<i>op</i>	<i>fa</i> ... <i>fa</i>	<i>fa</i> ... <i>fa</i>	<i>op</i> ... <i>fa</i>
U_{i-1}	<i>dor</i>	<i>dor</i> ... <i>dor</i>	<i>dor</i> ... <i>dor</i>	<i>act</i> ... <i>act</i>
$P(U_i = act)$	$1 - R_U(t_i t_{i-1})$	$1 - R_{U\{m\}}(t_i t_{i-1})$	1 ... 1	1 ... 1
$P(U_i = dor)$	$R_U(t_i t_{i-1})$	$R_{U\{m\}}(t_i t_{i-1})$	0 ... 0	0 ... 0

Tabelle 3.9: Allgemeine Struktur der CPT der Fehlerursache einer Komponente oder eines Teilsystems innerhalb einer komplexen k -aus- m Redundanz (vergleiche Abbildung 3.21).

sodass die Fehlerursache U_i durch die bedingte Zuverlässigkeit $R_U(t_i|t_{i-1})$ beschrieben wird. Treten Ausfälle im redundanten System mit $n_{Aus} \leq (m-k)$ auf, sodass das System funktionsfähig bleibt, aber lokale Zuverlässigkeiten an die veränderliche Lastsituation angepasst werden müssen, werden die bedingten Zuverlässigkeiten in Abhängigkeit der Ausfälle $R_{U\{m\}}(t_i|t_{i-1})$ verwendet. Für den Fall $n_{Aus} > (m-k)$, dass mehr Ausfälle auftreten als das System kompensieren kann, wird das redundante System als ausgefallen betrachtet und die Fehlerursache U_i unabhängig von U_{i-1} aktiviert. Die Fehlerursache U_i wird dem allgemeinen Modellierungsansatz folgend aktiviert, wenn die vorherige Fehlerursache U_{i-1} bereits aktiviert wurde.

Die CPTs der Ausfallzustände werden als noisy-MAX-Verknüpfungen implementiert und werden unter Ausnutzung der Annahme, dass alle Elternknoten unabhängig Einfluss auf den Ausfallzustand ausüben, separat für jeden Elternknoten beschrieben. Diese von einander unabhängigen Beschreibungen werden für jeden Elternknoten zusammengesetzt, sodass sich eine vollständige CPT als noisy-MAX-Verknüpfung ergibt. Diese separaten Einträge in die CPTs der Ausfallzustände werden ausgehend von Standardvorgaben erzeugt. Diese Standardvorgaben sind für alle Ausfallzustände implementiert und bilden das Ausfallverhalten des zu untersuchenden Systems konservativ ab, ohne Unsicherheiten zu berücksichtigen. Demnach werden alle Ausfallzustände als konjunkte Verknüpfung der Elternknoten, interne und externe Fehlerursachen, modelliert, sodass jeder Eintritt einer Fehlerursache zu einem Ausfall des aktuellen Ausfallzustands führt. Ausfallzustände sind mit der Standardvorgabe für Zustände $\{failed, operational\}$ modelliert. Zusätzliche Zustände werden als graduelle Abstufungen zwischen den Standardvorgaben betrachtet, sodass sich für eine Erweiterung der Zustände das folgende Schema ergibt $\{failed, \dots, AdditionalState, \dots, operational\}$. Alle Standardvorgaben bezüglich der Zustände sowie der Einträge der CPTs der Ausfallzustände sind unter Berücksichtigung der Anforderungen des Modellierungsansatzes editierbar (vergleiche Abschnitt 3.4.3.1). Eine Überschreibung der Standardvorgabe der Zustände der Ausfallzustände erfordert immer die Definition einer entsprechenden CPT, um dem Einfluss des zusätzlichen Zustands innerhalb des Ausfallzustands abzubilden. In Tabelle 3.10 sind die Standardvorgaben für die CPTs eines Ausfallzustands A anhand der möglichen Elternknoten kategorisiert.

Die CPT-Einträge im Ausfallzustand A_i für verschiedene Elternknoten als noisy-MAX-Verknüpfung beinhalten (in grau hinterlegt) den Zustand des Elternknotens mit kausaler Auswirkung auf den Ausfallzustand (distinguished state). Die Standardvorgabe für CPT-

CPT Elternknoten		CPT-Einträge der Ausfallzustände				
	Interne Fehlerursache		<i>active</i>	<i>dormant</i>		
		$P(A_i = failed)$	1	0		
		$P(A_i = operational)$	0	1		
Noisy-MAX	Ausfallzustand (externe Fehlerursache)		<i>failed</i>	<i>operational</i>		
		$P(A_i = failed)$	1	0		
		$P(A_i = operational)$	0	1		
	Leak		<i>Leak</i>			
		$P(A_i = failed)$	0			
		$P(A_i = operational)$	1			
Standard	Ausfallzustand (Redundanz)	Elternknoten 1	$n_{Aus} \leq (m-k)$		$n_{Aus} > (m-k)$	
		\vdots	<i>op</i>	<i>op</i>	<i>op</i>	<i>fa</i>
		Elternknoten <i>m</i>	<i>op</i>	<i>op</i>	<i>op</i>	<i>fa</i>
		$P(A_i = failed)$	0	0	1	1
		$P(A_i = operational)$	1	1	0	0

Tabelle 3.10: Übersicht über CPT-Einträge der Ausfallzustände an einem Beispielknoten A für verschiedene Modellierungsaspekte. Grau unterlegte Spalten zeigen distinguished state in noisy-MAX-Verknüpfungen (vergleiche Abschnitt 3.4.2.6).

Einträge für eine interne Fehlerursache sieht vor, dass eine Berücksichtigung gradueller Zustände nicht stattfindet, sodass eine Aktivierung der Fehlerursache immer zu einem Ausfallzustand *failed* führt. Ein äquivalentes Vorgehen ist für die Standardvorgabe der CPT-Einträge bei einer externen Fehlerursache vorgesehen, wobei diese graduelle Zustände aufweisen kann. Diese graduellen Zustände beschreiben typischerweise verschiedene Degradationsstufen und werden einer konservativen Abschätzung des Ausfallverhaltens nach dem Ausfallzustand *failed* zugeschrieben. Eine Erweiterung der CPT-Einträge, um Unsicherheiten für graduelle Zustände sowohl des betrachteten Ausfallzustands als auch externer Fehlerursachen, ist für eine zielführende und effiziente Modellierung des Ausfallverhaltens zu verwenden, da die Standardvorgaben nur eine vereinfachte Abschätzung liefern. Die Standardvorgabe für den *leak*-Anteil der Fehlerursachen wird über alle Ausfallzustände von A_i getroffen. Für den CPT-Eintrag $P(A_i = operational | leak)$ haben aktivierte, aber nicht explizit modellierte, Fehlerursachen keinen Einfluss auf den Ausfallzustand A_i .

Die CPT für den Ausfallzustand eines redundanten Teilsystems wird als normale CPT

modelliert, um das Ausfallverhalten des k -aus- m redundanten Systems abzubilden. Demnach werden zwei Fälle unterschieden, in denen das redundante System trotz $n_{Aus} \leq (m-k)$ Ausfällen funktionsfähig ist oder das System mit $n_{Aus} > (m-k)$ Ausfällen komplett ausfällt. Die Berücksichtigung gradueller Ausfallzustände der Elternknoten erfolgt analog zu den externen Fehlerursachen.

Die Reglementierung des Aufbaus der CPTs ermöglicht eine Vorgabe von Standardeinträgen, die eine Simulationsfähigkeit des Systemzuverlässigkeitsmodells, auf Basis der Definition von lokalen Zuverlässigkeitsschätzern, gewährleisten. Der Nutzer erhält die Möglichkeit, die Modellierung des Ausfallverhaltens durch die Einbringung weiterer Modellierungsaspekte, wie zusätzliche graduelle Zustände in den Ausfallzuständen und Unsicherheiten in den CPT-Einträgen, zu detaillieren.

3.5.3 Schnittstelle zu Modellierungs- und Analysewerkzeugen dynamischer Bayes'scher Netze

Die Komplexität der abgeleiteten dynamischen Bayes'schen Netze zur Abbildung der Systemzuverlässigkeit erfordert leistungsfähige Modellierungs-, Berechnungs- und Analysewerkzeuge. Die kommerziell oder quelloffen zur Verfügung stehenden Werkzeuge weisen jeweils spezifische Vor- und Nachteile auf: Die Abbildbarkeit von noisy-MAX-Verknüpfungen, die Verfügbarkeit analytischer und numerischer Analysealgorithmen oder die Verfügbarkeit weitergehender Analysemethoden, wie die Diagnose der wahrscheinlichsten Fehlerursache für einen gegebenen Ausfall (*most probable explanation*). Zu Modellierungswerkzeugen dynamischer Bayes'scher Netze berücksichtigt die Schnittstelle diese Aspekte und ermöglicht die Integration verschiedener Werkzeuge^{15,16} über spezialisierte Parser. Die Schnittstelle ist aus Sicht der Implementierung strukturell gleich aufgebaut wie die Topologie- und Simulation-Parser, sodass eine übergeordnete Klasse alle zur Verfügung stehenden Parser ermittelt und dem Nutzer zur Auswahl übergibt. In jedem Parser sind Methoden hinterlegt, um den im Datenobjekt *IntModData* abgelegten DAG und die CPTs in ein für das gewählte Modellierungswerkzeug geeignetes Format zu übertragen. Die Parser ermöglichen neben der Modellierung des dynamischen Bayes'schen Netzes deren Berechnung und Entgegennahme der Ergebnisse von einem vom Nutzer zu wählenden Knoten als Analyseziel in Form der Systemzuverlässigkeit. Die Systemzuverlässigkeit wird abschließend im Datenobjekt *IntModData* abgelegt.

3.6 Zusammenhang zu weiteren Kenngrößen der Verlässlichkeit

Die Zuverlässigkeit eines technischen Systems besitzt als zentrale Kenngröße großen Einfluss auf deren Verfügbarkeit und Sicherheit als weitere Kenngrößen der Verlässlichkeit (vergleiche Abschnitt 2.1). In Abschnitt 2.8 ist als Zielsetzung dieser Arbeit die Abbildung der Zuverlässigkeit formuliert worden. Im Folgenden werden Schnittstellen dieser Abbildung zu weiteren Kenngrößen aufgezeigt. Eine Integration der Systemzuverlässigkeit, aufbauend auf der integrierten Modellierung, in weiterführende Analysen der Ver-

¹⁵Genie & Smile von BayesFusion, LLC (www.bayesfusion.com) (besucht am 12.10.2019)

¹⁶Bayes Net Toolbox for Matlab (BNT) [Mur+01]

fügarkeit und der Sicherheit ist demnach für eine effiziente Abbildung der Kenngrößen der Verlässlichkeit technischer Systeme zielführend.

Verfügbarkeit

Die Modellierung der Verfügbarkeit technischer Systeme setzt die Berücksichtigung von reparierbaren Systemen mit zusätzlichen Modellierungsaspekten, wie Reparaturdauer, Ersatzteillagerung, Verfügbarkeit von Instandhaltungsressourcen und Weiteren, voraus [Bir07, Ber08]. Diese Modellierungsaspekte können mit dynamischen Bayes'schen Netzen nicht umfänglich abgebildet werden, sodass spezialisiertere Modellierungsansätze notwendig sind. In der Literatur werden Petri Netze zur Abbildung der Verfügbarkeit technischer Systeme mit hoher Modellierungstiefe eingesetzt [Sch92, Poz06, Rei10, LZB15] und finden insbesondere Anwendung bei der Modellierung und Analyse flexibler Produktionssysteme im Kontext Industrie 4.0 [LZB17]. In Abbildung 3.31 ist ein mögliches Petri Netz zur Modellierung der Verfügbarkeit einer Flotte von n Systemen für eine Instandhaltungsmannschaft gezeigt. Die Instandhaltung der Systeme ist priorisiert, basierend auf der Struktur der Sub-Graphen Sys_i mit $i \in (1, n)$ und einer beliebig kleinen Verzögerung in der Transition T_{Prio} [Sch92]. Bei gleichzeitigem Ausfall der Systeme Sys_i und Sys_{i+1} wird nach einer Reihenfolge für $i < i+1$ instandgehalten. Die Transition $T_{Reparatur}$ beschreibt die mittlere Reparaturdauer, während die Transition $T_{R_{Sys}(t)}$ Ausfallzeiten des betrachteten Systems enthält, die mittels Monte-Carlo-Simulation des Petri-Netzes [Poz06] entsprechend der zugrundeliegenden Verteilungsfunktion $R_{Sys}(t)$ gezogen werden. Die Verteilungsfunktion der Systemzuverlässigkeit $R_{Sys}(t)$ wird durch die integrierte Modellierung bereitgestellt und ermöglicht eine Analyse der Verfügbarkeit einer Flotte von Systemen unter Berücksichtigung des detaillierten Ausfallverhaltens der enthaltenen Systeme [HB16a].

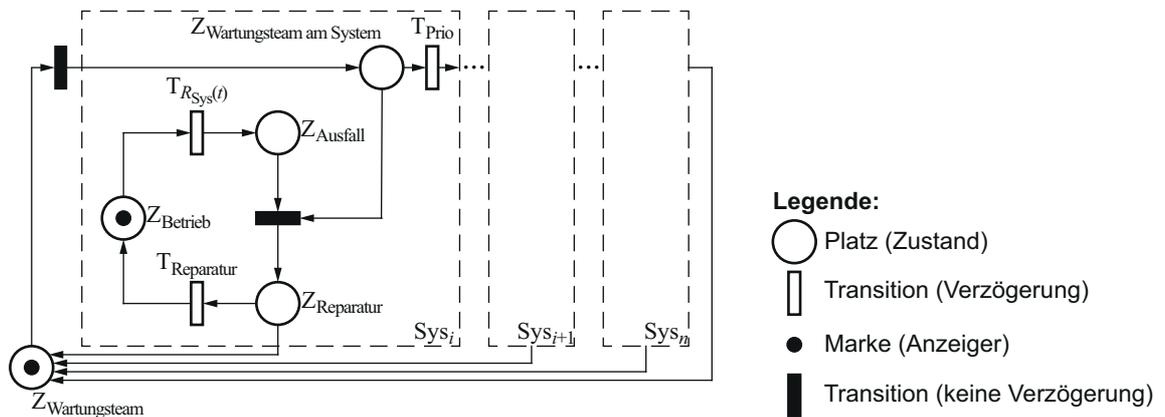


Abbildung 3.31: Petri Netz einer Flotte von reparierbaren Systemen auf Basis der Systemzuverlässigkeit $R_{Sys}(t)$ (nach [MKS15])

Funktionale Sicherheit

Auf Basis der in Abschnitt 2.1 gezeigten grundlegenden Zusammenhänge zwischen den Kenngrößen der Verlässlichkeit, erarbeitete Dorociak [Dor15] einer Systematik zur Absicherung der Zuverlässigkeit und Sicherheit mechatronischer Systeme in frühen Entwicklungsphasen. Es werden Fehlerbäume aus Spezifikationsmodellen abgeleitet und für eine

Analyse der Zuverlässigkeit und funktionalen Sicherheit verwendet. Tabako erweiterte diesen Ansatz zur Abschätzung der funktionalen Sicherheit auf Basis statischer Bayes'scher Netze [TB16]. Die Norm IEC 62508 gilt als allgemeine Basis weiterer Sicherheitsnormen, wie EN 62062 oder ISO 26262, und verwendet sogenannte Sicherheitsintegritätslevel (SIL) zur Bewertung der Sicherheit eines technischen Systems. Diese Bewertung kann quantitativ erfolgen, sodass den SILs PFD- (Probability of Failure on Demand) und PFH-Werte (Probability of Failure per Hour) zugeordnet sind. Die PFD- und PFH-Werte können mithilfe der integrierten Modellierung detailliert abgebildet werden. Damit wird eine präzise Absicherung der funktionalen Sicherheit auf Systemebene ermöglicht. Eine Einführung in die Sicherheitsanalyse auf Basis verschiedener Normen ist in [WK07] gegeben.

4 Anwendungsbeispiele

In diesem Kapitel werden drei Anwendungsbeispiele der integrierten Modellierung gezeigt, die der Validierung der entwickelten Methode und der Darstellung der Leistungsfähigkeit sowie des Potenzials in verschiedenen Lebenszyklusphasen dienen. Den drei Anwendungen liegen Systemmodelle zugrunde, die in unterschiedlichen Modellierungsumgebungen umgesetzt sind, um ein breites Spektrum unterstützter Modellierungswerkzeuge aufzuzeigen. Eine grundlegende Motivation der integrierten Modellierung, ist die Unterstützung einer Optimierung des dynamischen Systemverhaltens hinsichtlich zuverlässigkeitsorientierter Ziele. Die Unterstützung der Umsetzung selbstoptimierender technischer Systeme wird am Beispiel der optimierungsbasierten Regelung der Zuverlässigkeit einer Reibkupplung mittels der Integration zuverlässigkeitsorientierter Ziele gezeigt. Die Analyse der Systemzuverlässigkeit einer Windenergieanlage, als Beispiel eines komplexen mechatronischen Systems, wird durchgeführt. Der Fokus liegt dabei auf der Anwendung der Methode auf komplexe Systeme und die Integration von spezialisierten lokalen Zuverlässigkeitsschätzern. Mithilfe der integrierten Modellierung ist eine durchgängige Nutzung von Modellen des Entwurfs mechatronischer Systeme über verschiedene Lebenszyklusphasen gegeben. Die effiziente Umsetzung eines Digitalen Zwillings für die Instandhaltung wird auf Basis von Systemzuverlässigkeitsmodellen, ausgehend von der integrierten Modellierung in Kombination mit datenbasierten Verfahren der Zustandsüberwachung, dargestellt.

4.1 Zuverlässigkeitsadaptive Reibkupplung

In Abschnitt 2.5 werden die Potenziale selbstoptimierender Systeme zur Steigerung der Verlässlichkeit anhand verschiedener Methoden aufgezeigt. Die zuverlässigkeitsadaptive Regelung nach Meyer [Mey16] stellt hier einen leistungsfähigen Ansatz dar, der anhand des experimentellen Aufbaus einer Trockenkupplung validiert wurde (vergleiche Abbildung 4.1). Die Regelung basiert auf der anwendungsspezifischen Berechnung eines komfort- und eines zuverlässigkeitsorientierten Ziels zur Bestimmung von geeigneten Betriebspunkten.

Das Anwendungsbeispiel soll zeigen, dass mittels der integrierten Modellierung formulierte zuverlässigkeitsorientierte Ziele für die Bestimmung von Betriebspunkten erfolgreich innerhalb der zuverlässigkeitsadaptiven Regelung nach Meyer [Mey16] einsetzbar sind. Eine Validierung am realen Prüfstand wird nicht durchgeführt, da bereits Meyer zeigte, dass die Regelungsstrategie ausreichend robust ist, um Abweichungen zwischen modellbasiert und experimentell bestimmten Betriebspunkten kompensieren zu können (vergleiche Abbildung 4.12 in [Mey16]). Die Validierung der mittels der integrierten Modellierung bestimmten Betriebspunkte für die zuverlässigkeitsadaptive Regelung erfolgt daher modellbasiert.

4.1.1 Prüfstand und Regelungsstrategie

Die Validierung der zuverlässigkeitsadaptiven Regelung erfolgte in [Mey16] anhand des experimentellen Aufbaus der Reibkupplung in Abbildung 4.1. Der Prüfstand bildet Kupplungsvorgänge ab, indem der Antriebsmotor mit einer konstanten Drehzahl rotiert und die Reibscheiben während des Kupplungsvorgangs mit einer variablen Trajektorie der Normalkraft zwischen beiden Reibscheiben aufeinander gepresst werden. Der Kupplungsvorgang ist abgeschlossen, wenn die Reibscheibe der Last aus dem Stillstand auf die Antriebsdrehzahl beschleunigt wurde. Die Last wird virtuell durch ein Fahrzeugmodell abgebildet und das berechnete Lastmoment über den Lastmotor aufgeprägt. Die Reibscheiben sind aus Filz gefertigt und weisen einen beabsichtigten hohen mechanischen Verschleiß auf, sodass Lebensdauerexperimente über eine geringe Anzahl an Kupplungsvorgängen durchgeführt werden können. Die Lebensdauer der Reibscheiben ist deutlich geringer als die der anderen Komponenten – die integrierte Modellierung bestätigt dies (vergleiche Abschnitt 4.1.2).

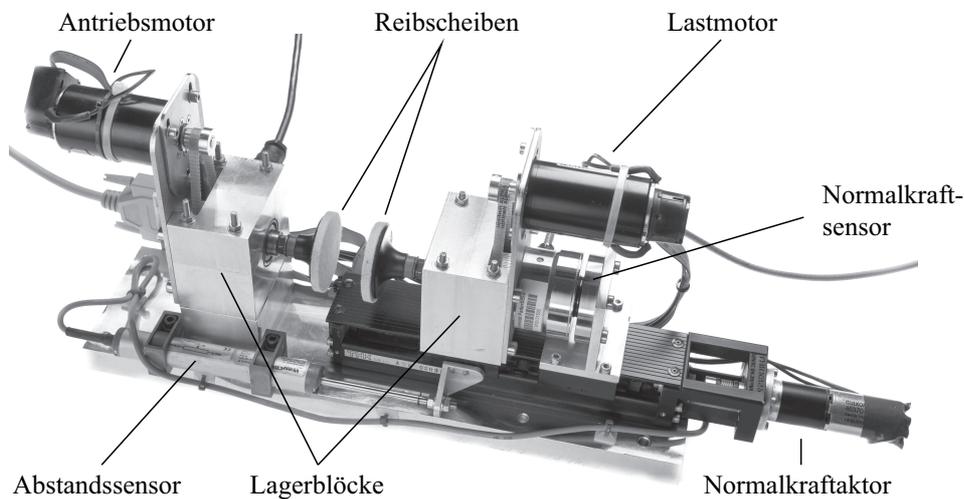


Abbildung 4.1: Reibkupplung nach [Mey16].

Die Reibkupplung wurde modellbasiert hinsichtlich zwei antagonistischer Entwurfsziele optimiert und die Pareto-Menge bestimmt. Dabei wurde durch Meyer [Mey16] die Reibarbeit E_f als äquivalent einer Verlässlichkeitskenngröße, basierend auf der Annahme, dass die umgesetzte Reibarbeit pro Kupplungsvorgang proportional zur Lebensdauerabnahme sei, gewählt. Daneben wurde die maximale rotatorische Beschleunigung $\max(\dot{\omega}(t))$ des Lastmotors während eines Kupplungsvorgangs als Äquivalent eines Komfortmaßes für ein virtuelles Fahrzeug gewählt. Als freie Systemparameter \mathbf{p} zur Beeinflussung beider Ziele wurde die Trajektorie der Normalkraft F_N zwischen den Reibscheiben über den Kupplungsvorgang sowie die Dauer des Kupplungsvorgangs t_r gewählt. Die Zielfunktionen ergeben sich nach [Mey16] demnach für die in einem Kupplungsvorgang umgesetzte Reibarbeit mit:

$$f_1 = E_f(t_r) \quad (4.1)$$

und für den Komfort mit Beginn des Kupplungsvorgangs zum Zeitpunkt t_0 mit:

$$f_2 = \max(\dot{\omega}(t)), \quad t \in [t_0, t_0 + t_r]. \quad (4.2)$$

Das Mehrzieloptimierungsproblem ist durch Meyer unter anderem für die Randbedingungen $t_r \in [5, 30]$ sec, $F_N(t = 0) = 0$ N sowie $F_N(t_r) = 100$ N mit $k = 3$ Stützstellen der Normalkrafttrajektorie F_N definiert worden:

$$\mathbf{p} = \begin{bmatrix} F_N\left(\frac{1}{k+1} \cdot 1 \cdot t_r\right) \\ F_N\left(\frac{1}{k+1} \cdot 2 \cdot t_r\right) \\ \vdots \\ F_N\left(\frac{1}{k+1} \cdot k \cdot t_r\right) \\ t_r \end{bmatrix}, \quad (4.3)$$

$$\hat{\mathbf{p}} = \min_{\mathbf{p}}(f_1(\mathbf{p}), f_2(\mathbf{p})). \quad (4.4)$$

Als Ergebnis erhält Meyer eine Pareto-Menge von Normalkrafttrajektorien für unterschiedliche Gewichtungen der Ziele. Diese Pareto-Menge dient als Basis für die zuverlässigkeitsadaptive Regelung und ist gleichzeitig, die in diesem Anwendungsbeispiel maßgebliche, Eingangsgröße.

Die Struktur der zuverlässigkeitsadaptiven Regelung ist in Abbildung 2.7 gezeigt und in Abschnitt 2.5 erläutert worden. Die Regelung basiert auf der Bestimmung eines Health Index \widehat{HI} , der den relativen verbleibenden ertragbaren Verschleiß der Reibscheiben als Lebensdauermaß beschreibt. Die Bestimmung des Health Index \widehat{HI} basiert im experimentellen Aufbau auf der Messung der Dicke der Reibscheiben \widehat{HI} , die Messrauschen unterworfen ist und einem generischen Degradationsmodell, deren Parameter über einen Kalmanfilter bestimmt werden. In der Mehrzieloptimierung erfolgt die Bestimmung des Health Index \widehat{HI} modellbasiert über die umgesetzte Reibarbeit $\Delta E_f = E_f(k)$ pro Kupplungsvorgang (k), wobei die gesamte umgesetzte Reibarbeit als $E_f = \sum \Delta E_f$ definiert ist. Die Messungenauigkeit wird über die Addition von Rauschen auf den modellbasiert bestimmten Health Index berücksichtigt (vergleiche [Mey16]).

Die Zuverlässigkeitsregelung wählt geeignete α_{soll} -Werte, basierend auf dem Health Index \widehat{HI} , für die Einstellung der Betriebspunkte auf der Pareto-Front mit dem Ziel, die gewählte Lebensdauer zu erreichen. Dabei bildet die s -Transformation die Regelgröße α auf die Pareto-Menge der optimalen Systemparameter über die Pareto-Front ab, während die inverse s^{-1} -Transformation Systemparameter der Pareto-Menge auf α -Werte abbildet. Eine Evaluation der Zielfunktionen für jeden Kupplungsvorgang liefert α_{ist} als Systemausgang mithilfe der inversen s^{-1} -Transformation. Die Differenz $\alpha_{soll} - \alpha_{ist}$ dient als Eingang für den Verhaltensregler, der als modellprädiktive Regelung ausgelegt ist und die Regelgröße für das dynamische System liefert. Der Verhaltensparameter α ist anhand der zugrundeliegenden Pareto-Front definiert. In Abbildung 4.2 ist die von Meyer für das Optimierungsproblem 4.4 gefundene Pareto-Front dargestellt. Die Bestimmung des Verhaltensparameters α ist nach [Mey16] als Verhältnis des Abstands zwischen den Extrema der Paretofront D_p und des Abstands des Schnittpunkts der orthogonalen Projektion auf eine Gerade zwischen den Extrema zum Minimum der Pareto-Front D_a definiert:

$$\alpha = 1 - \frac{2D_a}{D_p}. \quad (4.5)$$

Eine detaillierte Beschreibung des Prüfstands, der Entwicklung der Regelung sowie der Formulierung des Optimierungsproblems ist [Mey16] zu entnehmen.

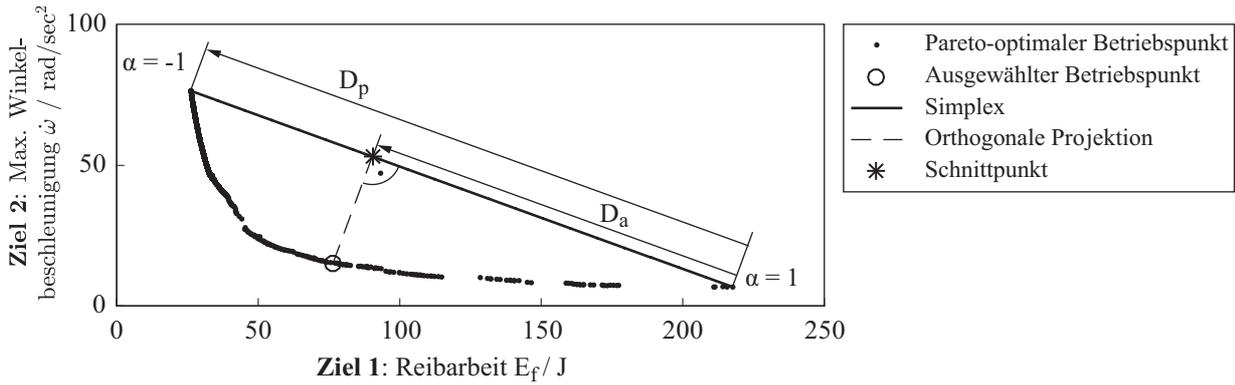


Abbildung 4.2: Pareto-Front der Reibkupplung für das Optimierungsproblem 4.4 und Definition des Verhaltensparameter α nach [Mey16].

4.1.2 Integrierte Modellierung

Die Entwicklung der Reibkupplung ist modellbasiert erfolgt, wobei u. a. ein Modell des dynamischen Systemverhaltens in Matlab / Simulink zum Einsatz gekommen ist. Dieses Modell wird für die integrierte Modellierung der Dynamik und der Zuverlässigkeit der Reibkupplung genutzt. Das Matlab / Simulink-Modell ist für den Einsatz in der integrierten Modellierung im Rahmen einer besseren Verwendbarkeit angepasst worden, da Matlab / Simulink zunächst einen signalflussorientierten Modellierungsansatz verfolgt (vergleiche Abschnitt 2.6). Daher sind, mithilfe von Hierarchisierungen, einzelne Modellbestandteile entsprechend ihrer Zugehörigkeit zu physischen Komponenten innerhalb des Modells zusammengefasst. Dieser Vorgang hat keinen Einfluss auf das dynamische Systemverhalten, sondern ermöglicht lediglich eine verbesserte Zuordnung der lokalen Zuverlässigkeitsschätzer. Darüber hinaus wird die Variable „Kupplungsvorgänge“ anstatt der Betriebsdauer verwendet, um die Zuverlässigkeit darzustellen.

Der Optimierung der Reibkupplung im folgenden Abschnitt 4.1.3 geht eine Analyse der Systemzuverlässigkeit voraus. Zunächst werden für das Anwendungsbeispiel lokale Zuverlässigkeitsschätzer ausgewählt, um die Zuverlässigkeit verschiedener Komponenten abzubilden und kritische Komponenten zu identifizieren [QB18]. Nach Meyer [Mey16] ist bekannt, dass die Reibscheiben innerhalb des Systems die dominierenden Ausfälle aufweisen. Ein lokaler Zuverlässigkeitsschätzer wird nach dem linearen Degradationsmodell für mechanischen Verschleiß von Meyer umgesetzt [Mey16]. Die Abnahme des verbleibenden ertragbaren Verschleißes der Reibscheiben $l(k)$ nach jedem Kupplungsvorgang k sei demnach mit dem Faktor p_f proportional zur umgesetzten Reibarbeit E_f :

$$l(k) = p_f \cdot E_f(k). \quad (4.6)$$

Der ertragbare Verschleiß der Reibscheiben $l_{max} = 5$ mm ist bekannt, sodass die Lebensdauer der Reibscheiben unter der Annahme einer über die gesamte Lebensdauer konstanten umgesetzten Reibarbeit E_f als Anzahl an ertragbaren Kupplungsvorgängen berechnet wird. Die lokale Zuverlässigkeit wird über die Simulation einer Stichprobe von $n = 20$ Reibscheiben durchgeführt, wobei der ertragbare Verschleiß l_{max} als normalverteilt mit einer Standardabweichung von $\sigma = 0,004$ mm zur Abbildung von Toleranzen eingeht. Die Parameter einer Weibull-Verteilung werden anhand der Ausfallzeiten bestimmt.

Lokale Lebensdauerschätzer auf Basis der DIN ISO 281 werden für die Wälzlager in den Lagerblöcken der antriebs- und lastseitigen Welle der Reibscheiben definiert. In jedem Lagerblock sind zwei baugleiche Wälzlager verbaut, die unterschiedliche Lasten aufnehmen: eine Axialkraft (Normalkraft zwischen den Reibscheiben) und eine Radialkraft (Zugspannung aufgrund des übertragenen Drehmoments am Zahnriemen).

Der Antriebs- und Lastmotor sind ebenfalls baugleich und weisen jeweils mehrere Fehlerursachen auf: ein Ausfall eines der beiden internen Wälzlager sowie ein Ausfall der Wicklungsisolation (für die Abbildung mehrerer Fehlerursachen einer Komponente oder eines Teilsystems, vergleiche Abschnitt 3.4.3.1). Der lokale Zuverlässigkeitsschätzer zur Abbildung der Wicklungsisolation basiert auf einem empirischen Modell nach Brancato [Bra92], in dem die Nennbelastung sowie -betriebsbedingungen mit den tatsächlich Auftretenden verglichen werden und eine Verlängerung oder Reduzierung einer nominellen Lebensdauer bestimmt wird. Diesen Ausfällen wird eine Exponential-Verteilung zugrundegelegt.

Der Abstandssensor zur Bestimmung des verbleibenden ertragbaren Verschleißes ist ein variabler elektrischer Widerstand, dessen Zuverlässigkeit auf Basis einer Abschätzung des Mlt-Hdbk 217F [US 95] für eine Exponential-Verteilung bestimmt wird.

Die lokalen Zuverlässigkeitsschätzer der antriebs- und lastseitigen Lagerblöcke (vergleiche Abbildung 4.3 a) und b)) verwenden eine Betriebsfestigkeitsanalyse auf Basis der FEM für die Abbildung von Materialermüdung als Ausfallursache [KMS17]. Die Integra-

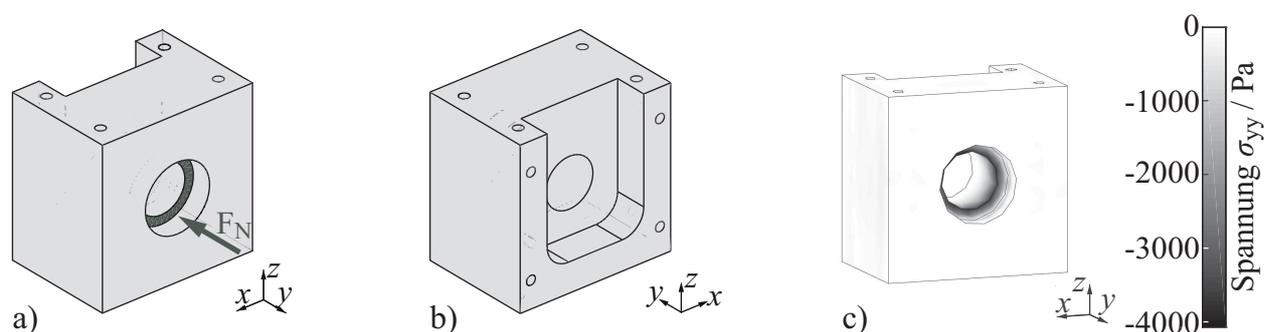


Abbildung 4.3: Forderansicht einer Lageraufnahme mit Krafteinleitung F_N durch Kuppelungsaktuierung (a), Rückansicht der Lageraufnahme (b) und Spannungsverteilung auf Basis einer FE-Analyse bei Belastung durch F_N (c). Nach [KMS17].

tion komplexer Zuverlässigkeitsschätzer wird anhand von Betriebsfestigkeitsanalysen in Kombination mit der Bestimmung der Bauteilbeanspruchung für dynamische Lasten aus dem Systemmodell gezeigt – eine detaillierte Abbildung der Bauteilbeanspruchung wird zugunsten einer vereinfachten Modellierung vernachlässigt. Die Bestimmung der Bauteilbeanspruchung mithilfe der FEM erfolgt daher unter vereinfachenden Annahmen, wie isotropes Material (Aluminium), linearem Materialverhalten ohne Plastifizierung und linearen Randbedingungen. So wird angenommen, dass die unteren Kanten der vier Durchgangsbohrungen zur Fixierung der Lagerblöcke auf der Grundplatte keine Verschiebung erfahren. Die resultierende Normalkraft F_N zwischen den Reibscheiben wird als Bauteilbelastung auf die Lagerblöcke aufgeprägt (vergleiche Abbildung 4.3 a)). Die Normalkraft F_N wird als axial wirkende Last über das Wälzlager auf den Lagerblock aufgebracht,

wobei angenommen wird, dass die Last gleichmäßig über den kompletten Bohrungsabsatz verteilt ist. Die Radiallast am zweiten Wälzlager, induziert über das übertragene Drehmoment am Zahnriemen, ist im Vergleich zur Normalkraft F_N deutlich geringer und wird zugunsten einer effizienteren Modellierung vernachlässigt. Die Berechnung der Bauteilbeanspruchung auf Basis einer Simulation des dynamischen Systemverhaltens zur Bestimmung der resultierenden Normalkraft F_N , wird in Matlab mit der Partial Differential Equation Toolbox durchgeführt. Der Funktionsumfang der Toolbox ist im Vergleich zu vollwertigen FEM-Werkzeugen, wie ANSYS oder Abaqus, stark eingeschränkt, aber für das Ziel einer Validierung für die Integration komplexer lokaler Zuverlässigkeitsschätzer ausreichend. So bietet die Toolbox beispielsweise nur Tetraeder als Elemente für die FEM. Das Ergebnis der Bestimmung der Bauteilbeanspruchung ist exemplarisch anhand der Normalspannung σ_{yy} zum Zeitpunkt t_r in Abbildung 4.3 c) dargestellt, wobei die Normalspannung σ_{yy} der höchsten auftretenden Bauteilbeanspruchung entspricht. Die Berücksichtigung einer Festigkeitshypothese erfolgt in einem nachgelagerten Schritt innerhalb der Betriebsfestigkeitsanalyse. Die Bauteilbeanspruchung bildet die Basis für die Betriebsfestigkeitsanalyse mithilfe des Quick Fatigue Tool for Matlab (QFT) [Val16]. Das QFT beinhaltet eine Materialdatenbank für die Charakterisierung der Materialeigenschaften. Der Materialdatenbank folgend, wird die Festigkeitshypothese nach Findley [Fin59] ausgewählt, die eine Bestimmung der Zeitfestigkeit auf Basis von elastischen Dehnungsanteilen durchführt. Die Bestimmung der Zeitfestigkeit wird anhand der maximalen Schubspannung und der wirkenden Normalspannung in einer kritischen Ebene innerhalb des Bauteils bestimmt, in der eine Rissinitiierung wahrscheinlich stattfinden wird.

Die Abbildung des Ausfallverhaltens der Reibkupplung berücksichtigt keine weiteren Modellierungsaspekte, wie Redundanz oder Ausfälle gemeinsamer Fehlerursache. Es wurden keine Redundanzen bei dem Entwurf der Reibkupplung vorgesehen. Auf die Abbildung von Ausfällen gemeinsamer Fehlerursache wird verzichtet, da durch das Fehlen von Redundanz jeder Ausfall einer Komponente zu einem Ausfall des Gesamtsystems propagiert wird.

Die integrierte Modellierung der Reibkupplung liefert die Zuverlässigkeit der Komponenten und des Gesamtsystems wie in Abbildung 4.4 dargestellt. Das Ergebnis bestätigt eines der Entwurfsziele der Reibkupplung, dass die Reibscheiben die dominierenden Ausfälle im System darstellen. Da der Prüfstand keine Redundanzen enthält und die lokalen Zuverlässigkeiten der übrigen Komponenten und Teilsysteme deutlich später abnimmt, liegen die lokale Zuverlässigkeit der Reibscheiben und die Systemzuverlässigkeit aufeinander. Über den gewählten Betrachtungszeitraum ist keine Reduzierung der Zuverlässigkeit des Abstandssensors und der Wälzlager in den antriebs- und lastseitigen Lagerblöcken am Zahnriemen oder an der Reibscheibe (Wälzlager_AZ, Wälzlager_AR, Wälzlager_LZ und Wälzlager_LR) feststellbar. Die Antriebsmotoren zeigen aufgrund der Kombination von drei Fehlerursachen eine leichte Reduktion der Zuverlässigkeit, weisen aber nach $6 \cdot 10^7$ Kupplungsvorgängen eine Zuverlässigkeit von 96% auf. Die Zuverlässigkeiten der zwei Lagerblöcke wurde nicht berücksichtigt, da diese für die gegebene dynamische Belastung nicht ausfallen. Dies ist aufgrund der sehr geringen maximalen Belastung von 100 N zu erwarten. Für die Gewährleistung eines recheneffizienten Modells, werden die lokalen Zuverlässigkeitsschätzer der Lagerblöcke für die nachfolgende Optimierung nicht berücksichtigt.

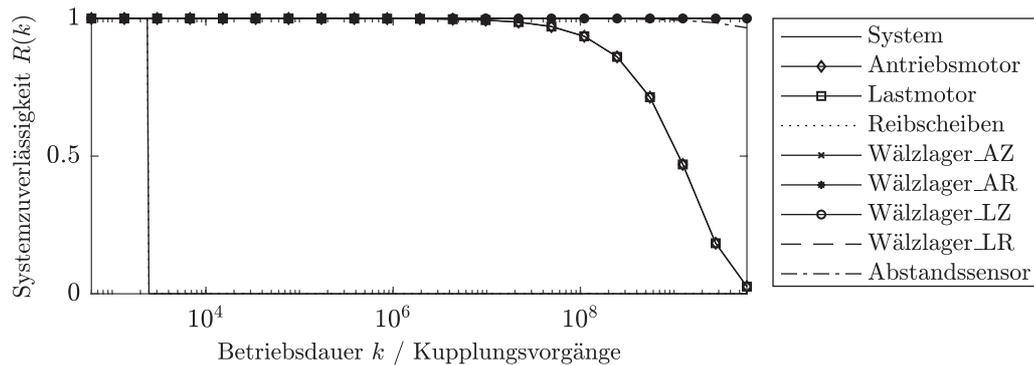


Abbildung 4.4: System- und Komponentenzuverlässigkeit.

4.1.3 Mehrzieloptimierung

Das Mehrzieloptimierungsproblem aus Gleichung 4.4 wird zusammen mit den Optimierungsparametern aus Gleichung 4.3 und der Zielfunktion des Komforts in Gleichung 4.2 übernommen. Eines der Ziele der integrierten Modellierung ist die allgemeine Formulierung von zuverlässigkeitsorientierten Zielfunktionen, sodass die von Meyer gewählte anwendungsspezifische Zielfunktion in Gleichung 4.1 ersetzt werden kann. Eine Berücksichtigung der Zuverlässigkeit auf Systemebene innerhalb der zuverlässigkeitsadaptiven Regelung erfordert die Formulierung einer geeigneten Zielfunktion. Das Systemzuverlässigkeitsmodell der integrierten Modellierung ermöglicht, neben der Betrachtung von Komponenten oder Teilsystemen, grundsätzlich auch die Berücksichtigung der Systemzuverlässigkeit als Zielgröße. Eine Zielfunktion auf Basis der Systemzuverlässigkeit erweitert den Funktionsumfang der Regelung maßgeblich. Die Regelung stellt nicht mehr nur die Lebensdauer einer zuvor ausgewählten Komponente sicher, sondern hebt die Absicherung einer gewünschten Lebensdauer auf Systemebene. In einem selbstoptimierenden System, in dem die Zuverlässigkeiten einzelner Komponenten oder Teilsysteme für verschiedene Betriebspunkte gegenläufig sein können (die Steigerung der Zuverlässigkeit einer Komponente oder eines Teilsystems führt zu einer Reduktion der Zuverlässigkeit weiterer Komponenten oder Teilsysteme), ist die integrierte Modellierung für die Formulierung zuverlässigkeitsorientierter Zielfunktionen auf Systemebene zielführend. Das bisherige Vorgehen auf Basis anwendungsspezifischer Zielfunktionen erfordert immer die Formulierung individueller Zielfunktionen. Die integrierte Modellierung ermöglicht hingegen die Nutzung einer gemeinsamen Methode zur Formulierung von Zielfunktionen. Das Ausfallverhalten der Reibkupplung ist stark von den Ausfällen der Reibscheiben dominiert, sodass die Verwendung einer Zielfunktion auf Basis der Systemzuverlässigkeit ähnliche Ergebnisse liefern wird, wie die anwendungsspezifische Zielfunktion in Gleichung 4.1.

Als Vorbereitung zur Formulierung einer Zielfunktion hinsichtlich der Systemzuverlässigkeit wird zunächst der Betrachtungszeitraum über die Kupplungsvorgänge k definiert, da eine große Anzahl an Zeitscheiben im Systemzuverlässigkeitsmodell die Berechnungsdauer verlängern. Von Interesse ist hier der Bereich der Kupplungsvorgänge k , in dem die Systemzuverlässigkeit stark absinkt, sodass für die Analyse der Systemzuverlässigkeit innerhalb der Mehrzieloptimierung für $k \in [0, 10^5]$ gewählt wird.

Die in Abschnitt 2.5 vorgestellten Methoden dienen der Steigerung der Verlässlichkeit selbstoptimierender Systeme. Die Zuverlässigkeit stellt eine zentrale Kenngröße der Verlässlichkeit dar, sodass als Optimierungsziel die „Steigerung der Zuverlässigkeit“ verfolgt wird. Eine Verwendung der Systemzuverlässigkeit zu einem festen Zeitpunkt ist als Zielfunktion nicht sinnvoll. Weiteres Optimierungspotenzial des Systems könnte ungenutzt bleiben, wenn noch zum gewählten Zeitpunkt eine Zuverlässigkeit an der oberen Grenze des Definitionsbereichs erreicht werden kann. Viel mehr ist es zielführend einen Abfall der Zuverlässigkeit möglichst spät während der Betriebsdauer herbeizuführen. Die Reibkupplung soll demnach eine Systemzuverlässigkeit von 95% für möglichst viele Kupplungsvorgänge nicht unterschreiten. Optimierungsprobleme werden typischerweise als Minimierungsproblem definiert, sodass folgender Zusammenhang zwischen Minimierung und Maximierung einer beliebigen Größe a genutzt wird: $\max(a) = \min(-a)$. Damit wird die Zielfunktion formuliert:

$$f_1 = -k \text{ für } R_{Sys}(k) = 0,95, k \in [0, 10^5]. \quad (4.7)$$

Das Systemzuverlässigkeitsmodell wird für diskrete Zeitpunkte über den Betrachtungszeitraum berechnet, wobei die genaue Anzahl an Kupplungsvorgängen k für $R_{Sys}(k) = 0,95$ nicht bekannt ist. Es ist daher notwendig, zwischen den diskreten Zeitpunkten des Systemzuverlässigkeitsmodells zu interpolieren, um die Anzahl an Kupplungsvorgängen für $R_{Sys}(k) = 0,95$ zu bestimmen.

Die Reibkupplung weist keine Rekonfigurationspotenziale auf, sodass eine zeitinvariante Topologie des Systems vorliegt. Daher wird vor Beginn der Optimierung die Topologieanalyse des Systemmodells (vergleiche Ablauf der integrierten Modellierung in Abbildung 3.22) einmalig durchgeführt, um die Berechnungsdauer jedes Iterationsschritts der Optimierung zu reduzieren. Die gefundenen pareto-optimalen Betriebspunkte, hinsichtlich der gewählten Zielfunktionen, sind als Ergebnis der Optimierung als Pareto-Front in Abbildung 4.5 dargestellt und ersetzen die von Meyer gefundene Pareto-Front aus Abbildung 4.2.

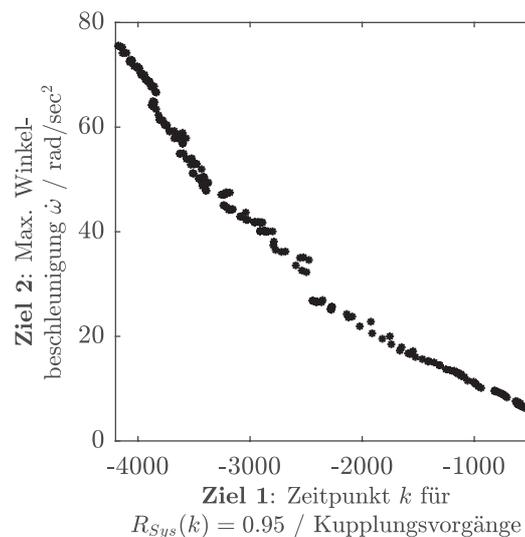


Abbildung 4.5: Pareto-Front der modellbasierten Mehrzieloptimierung der Reibkupplung.

Die Betriebspunkte sind über die Funktionswerte der Zielfunktion 4.7 und 4.2 abgebildet. Das Ziel 1 beschreibt die negativen Werte der Kupplungsvorgänge k , für die eine Systemzuverlässigkeit $R_{Sys}(k) = 0,95$ erreicht werden konnte. Ziel 2 beschreibt die jeweils dazu berechneten Maximalwerte der Winkelbeschleunigung der Lastseite der Kupplung während eines Kupplungsvorgangs. Die Formulierung der ersten Zielfunktion 4.7 wurde durch Negierung des Zielfunktionswerts von einem Maximierungs- in ein Minimierungsproblem überführt. Daher weisen Betriebspunkte oben links eine hohe Winkelbeschleunigung (geringer Komfort) und eine hohe Anzahl an Kupplungsvorgängen (hohe Zuverlässigkeit) auf. Dem gegenüber entsprechen Betriebspunkte unten rechts einer geringen Winkelbeschleunigung (hoher Komfort) und einer niedrigen Anzahl an ertragbaren Kupplungsvorgängen (verminderte Zuverlässigkeit).

4.1.4 Ergebnisse

Die in Abschnitt 4.1.3 gefundene Pareto-Menge und -front dient als Eingangsgröße für die zuverlässigkeitsadaptive Regelung. Für die Validierung der Pareto-Menge zur Verwendung in der Regelung wird dasselbe Simulink-Modell der Systemdynamik verwendet wie in der Optimierung. Die Bestimmung des Health Indexes \widetilde{HI} erfolgt am realen Prüfstand über die Messung der verbleibenden Dicke der Reibscheiben über den Abstandssensor. Die Messunsicherheiten dieses Sensors werden modellbasiert über die Addition von Rauschen auf die nach Gleichung 4.6 berechnete Abnahme des verbleibenden ertragbaren Verschleißes der Reibscheiben berücksichtigt.

Für die Validierung der Pareto-Menge auf Basis der integrierten Modellierung werden zwei Anwendungsfälle der Reibkupplung betrachtet:

- Konstante gewünschte Lebensdauer $k_{soll} = 4000$ Kupplungsvorgänge,
- Änderung der gewünschten Lebensdauer nach $k = 2000$ Kupplungsvorgänge von $k_{soll} = 4000$ Kupplungsvorgänge auf $k_{soll} = 5000$ Kupplungsvorgänge.

Die Validierung erfolgt anhand einer Stichprobe von $n = 30$ Reibscheiben in *simulierten Lebensdauertests* unter Berücksichtigung einer normalverteilten Herstellungstoleranz der Reibscheiben und der Messunsicherheit bei der Bestimmung des verbleibenden ertragbaren Verschleißes. Die Herstellungstoleranz wurde bereits im lokalen Zuverlässigkeitsschätzer der Reibscheiben in Abschnitt 4.1.2 betrachtet. Die Ergebnisse der Anwendungsfälle sind jeweils anhand der Darstellung der Verhaltensparametrierung α_{soll} , Variable für die Auswahl eines Betriebspunkts auf der Pareto-Front durch die Regelung und α_{ist} , Auswahlvariable des tatsächlich am System eingestellten Verhaltens, und dementsprechenden Betriebspunkt auf der Pareto-Front, gezeigt. Darüber hinaus wird eine Weibull-Verteilung der Zuverlässigkeit der Reibscheiben anhand der Ausfallzeiten angenähert und die Kupplungsvorgänge entsprechend den Zuverlässigkeiten von 50% und 95% gekennzeichnet. Die Darstellung der Ergebnisse ist in Analogie zu der Darstellung von Meyer [Mey16] gewählt worden, um die Vergleichbarkeit zu erhöhen.

In Abbildung 4.6 ist die Reibkupplung ohne Zuverlässigkeitsregelung dargestellt, um eine Referenz zu den Anwendungsfällen der Validierung bereitzustellen. Der Verhaltensparameter wird konstant mit $\alpha = -0,1374$ für eine leichte Priorisierung des Zuverlässigkeitsziels gewählt (vergleiche [Mey16]). Die Ausfallzeiten der Reibkupplung streuen innerhalb der Stichprobe über einen Bereich von $R(k = 4553) = 0,95$ und $R(k = 4614) = 0,50$.

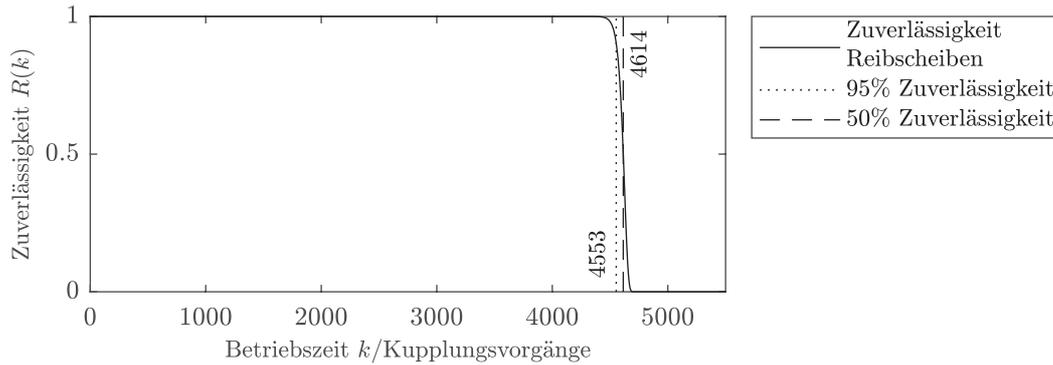


Abbildung 4.6: Zuverlässigkeit der Reibscheiben für eine konstante Verhaltensparametrierung mit $\alpha = -0,1374$ in einer Stichprobe von $n = 30$ Reibscheiben.

Die zuverlässigkeitsadaptive Regelung ist im ersten Anwendungsfall für eine Lebensdauer von $k_{soll} = 4000$ Kupplungsvorgänge in der Lage, die Streuung der Ausfallzeiten zu reduzieren und die gewünschte Lebensdauer einzustellen. In Abbildung 4.7 ist der Verlauf der Ist- und Sollwerte der Verhaltensparametrierung α dargestellt, wobei der Mittelwert und die Standardabweichung $\sigma(\alpha)$ jeweils abgebildet sind.

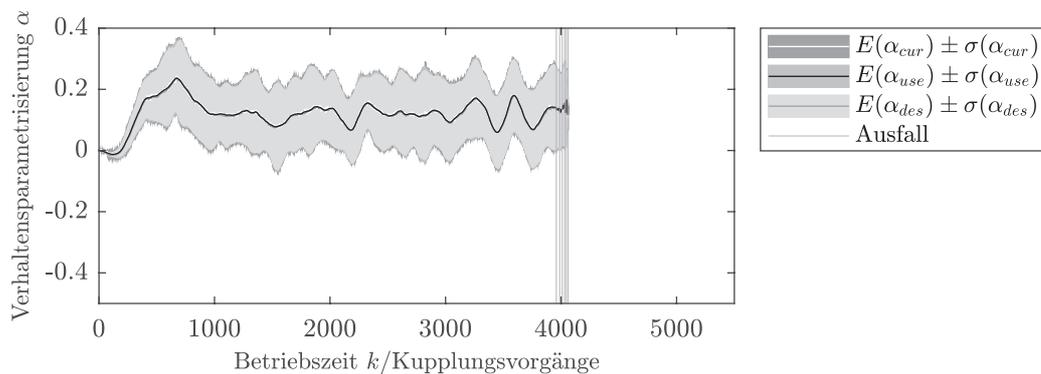


Abbildung 4.7: Verhaltensparametrierung α für eine gewünschte Lebensdauer von 4000 Kupplungsvorgänge.

Der Verhaltensparameter steigt zunächst von einem Startwert von $\alpha_{soll} = 0$ an, was als Priorisierung des Komforts zu interpretieren ist. Die darauf folgenden Veränderungen des Mittelwerts sind auf den begrenzten Prädiktionshorizont der modellprädiktiven Regelung innerhalb des Zuverlässigkeitsreglers, der Messunsicherheit bei der Bestimmung des Health Index \widehat{HI} sowie der geringen Anzahl an gefundenen Betriebspunkten auf der Pareto-Front zurückzuführen. In Abbildung 4.5 sind 204 Betriebspunkte bestimmt worden. Im Gegensatz dazu nutzte Meyer in seiner Arbeit 1951 Betriebspunkte (vergleiche [Mey16]). Meyer merkte bereits an, dass die Anzahl an pareto-optimalen Betriebspunkten Auswirkungen auf die Regelgüte der zuverlässigkeitsadaptiven Regelung aufweist. Aufgrund der geringen Anzahl an Betriebspunkten kann es zu größeren Sprüngen im Verlauf der Verhaltensparametrierung α kommen. Die Ergebnisse in Abbildung 4.7 zeigen jedoch, dass auch mit einer deutlich verminderten Anzahl an Betriebspunkten ein erfolgreicher Betrieb der zuverlässigkeitsadaptiven Regelung möglich ist.

Der Verlauf der Zuverlässigkeit der Reibscheiben in Abbildung 4.8 verdeutlicht die erfolgreiche Anwendbarkeit der integrierten Modellierung für die zuverlässigkeitsadaptive Regelung, da die Streuung der Ausfallzeiten im Vergleich zur Verhaltensregelung mit konstanter Verhaltensparametrierung in Abbildung 4.6 reduziert worden ist. Im Hinblick auf die Arbeiten von Meyer ist anzumerken, dass ein Unterschreiten der gewünschten Lebensdauer möglichst vermieden werden soll, was in der experimentellen Validierung in [Mey16] nachgewiesen werden konnte. Hinsichtlich dieser Anforderungen, ist die Differenz von 3979 Kupplungsvorgänge für eine Zuverlässigkeit von 95% zu einer gewünschten Lebensdauer von 4000 Kupplungsvorgänge zu groß. Es ist zu vermuten, dass der Grund hierzu in der geringen Anzahl an Betriebspunkten zu suchen ist, da der experimentelle Funktionsnachweis von Meyer dieses Verhalten nicht aufweist.

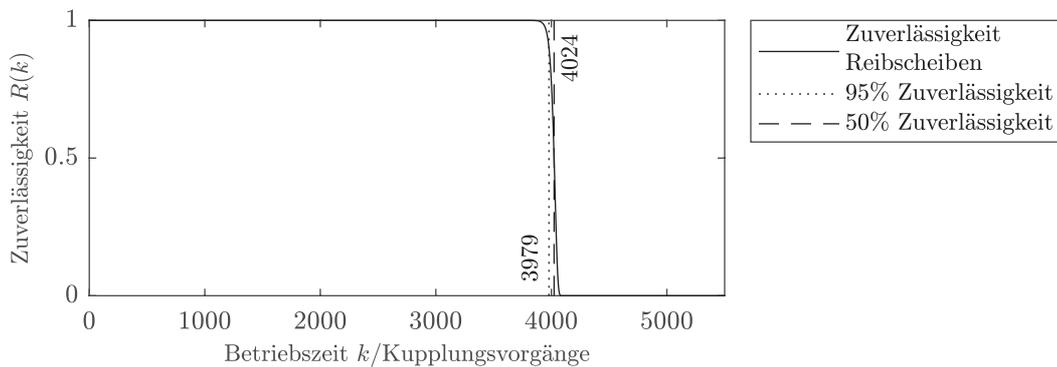


Abbildung 4.8: Zuverlässigkeit der Reibscheiben für eine gewünschte Lebensdauer von 4000 Kupplungsvorgängen in einer Stichprobe von $n=30$ Reibscheiben.

Der zweite Anwendungsfall der Validierung, eine Änderung der gewünschten Lebensdauer nach $k=2000$ Kupplungsvorgängen von $k_{soll}=4000$ Kupplungsvorgängen auf $k_{soll}=5000$ Kupplungsvorgängen, zeigt auch eine erfolgreiche Anwendbarkeit der integrierten Modellierung. In Abbildung 4.9 ist der Verlauf der Verhaltensparametrierung zunächst ähnlich wie in Abbildung 4.7, fällt aber mit der Änderung der gewünschten Lebensdauer stark ab. Der Abfall der Verhaltensparametrierung ist als eine Änderung der Priorisierung der Ziele von Komfort zu Zuverlässigkeit zu interpretieren. Mit einer deutlichen Verlängerung der gewünschten Lebensdauer wählt die Regelung einen Betriebspunkt, der zu einer deutlichen Reduktion des auftretenden Verschleißes der Reibscheiben führt. Die Zuverlässigkeit der Reibscheiben für eine Änderung der gewünschten Lebensdauer ist in Abbildung 4.10 dargestellt. Die gewünschte Lebensdauer wird erreicht, jedoch zu Lasten einer stärkeren Streuung der Ausfallzeiten im Vergleich zu der konstanten Verhaltensparametrierung wie in Abbildung 4.6. Es tritt dazu das gleiche Problem der Unterschreitung der gewünschten Lebensdauer auf wie es bereits in Abbildung 4.8 beobachtet wurde. Die zugenommene Streuung und die Unterschreitung der gewünschten Lebensdauer sind wieder auf die geringe Anzahl an Betriebspunkten zurückzuführen.

Das Anwendungsbeispiel der zuverlässigkeitsadaptiven Regelung einer Reibkupplung zeigt, dass die integrierte Modellierung für die Formulierung von zuverlässigkeitsorientierten Zielfunktionen auf Systemebene für selbstoptimierende Systeme einsetzbar ist. Die zuverlässigkeitsadaptive Regelung konnte auch für eine deutlich reduzierte Anzahl an Betriebspunkten erfolgreich modellbasiert in Betrieb genommen werden.

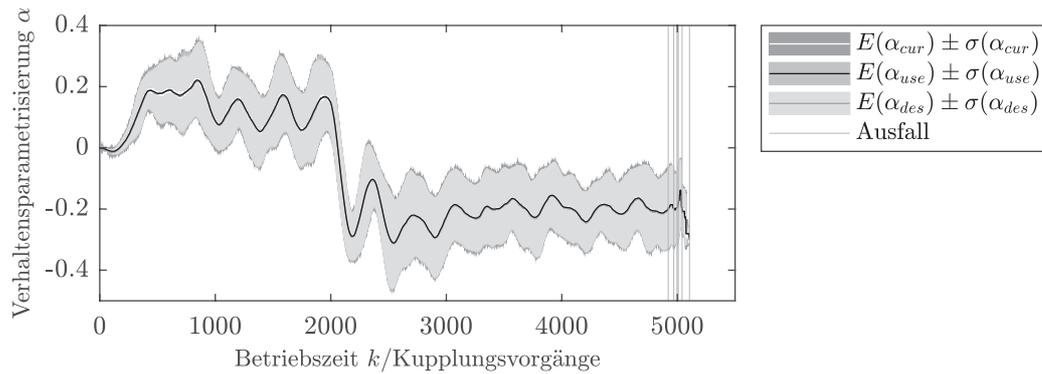


Abbildung 4.9: Verhaltensparametrisierung α für eine Änderung der gewünschten Lebensdauer von 4000 auf 5000 Kupplungsvorgänge nach durchgeführten 2000 Kupplungsvorgängen.

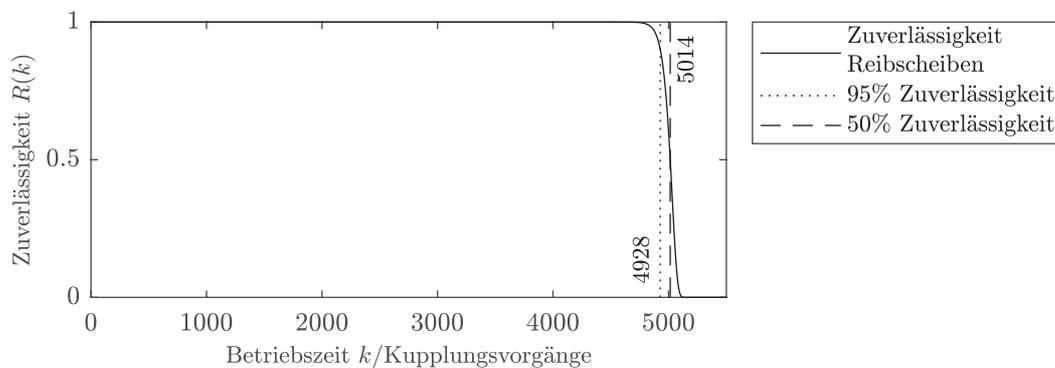


Abbildung 4.10: Zuverlässigkeit der Reibscheiben für eine Änderung der gewünschten Lebensdauer von 4000 auf 5000 Kupplungsvorgängen nach durchgeführten 2000 Kupplungsvorgängen in einer Stichprobe von $n=30$ Reibscheiben.

4.2 Zuverlässigkeitsanalyse einer Windenergieanlage

Windenergieanlagen (WEAs) dienen der Energieversorgung auf Basis unbegrenzt zur Verfügung stehender Quellen. Diese Anlagen werden zu Lande (onshore) und zu Wasser (offshore), entweder schwimmend oder mit Gründung, aufgestellt. Aktuelle WEAs erreichen eine Nennleistung von 12 MW bei einer Gesamthöhe von 260 m und einem Rotordurchmesser von 220 m.¹⁷ WEAs sind aufgrund von rauen Umweltbedingungen hohen Lasten ausgesetzt und weisen darüber hinaus eine große Komplexität auf, unter anderem durch individuelle Regelung der Rotorblätter zur Reduzierung wirkender Lasten und zur Steigerung der Effizienz. Demnach stellen WEAs komplexe mechatronische Systeme dar, deren Zuverlässigkeit aufgrund hoher Instandhaltungskosten ein zentrales Entwicklungsziel darstellt.

Die Entwicklung von WEAs setzt detaillierte Modelle voraus, um das dynamische Verhalten abzubilden und entsprechende Anforderungen zu validieren und zu verifizieren. Um Forschungseinrichtungen unabhängige Tätigkeiten im Bereich der Gestaltung der

¹⁷Entspricht Haliade-X 12 MW Offshore-WEA von General Electric Company.

Anlagen, der Regelungsstrategien und Entwicklungsmethoden zu ermöglichen, sind verschiedene Referenzmodelle für Lastsimulationen entwickelt worden, unter anderem NREL 5-MW Offshore Baseline Turbine [JBM+09], DTU 10MW Turbine [BZB+13] oder Fraunhofer IWES Wind Turbine IWT-7.5-164 [SRB+17]. Das Modell der Referenzanlage des Fraunhofer IWES¹⁸ dient als Anwendungsbeispiel für die integrierte Modellierung. Diese weist eine Nennleistung von 7,5 MW bei einer Rotordurchmesser von 164 m auf.

4.2.1 Referenzmodell – *OneWind*[®] Modelica Bibliothek

Das Referenzmodell IWT-7.5-164 ist in der Fraunhofer *OneWind*[®] Modelica Bibliothek implementiert und hinsichtlich der Modellierung von Rotorblättern und Turmstrukturen für gekoppelte Lastsimulationen entwickelt worden. Das gesamte Modell ist bezüglich seiner Modellierungstiefe und Topologie anpassbar. Unter anderem sind verschiedene Konfigurationen des Triebstrangs inklusive Generator verfügbar sind. Onshore sowie offshore WEAs mit unterschiedlichen Gründen sind vorhanden und es ist zwischen starren und flexiblen Körpern für Rotorblätter und Turmstruktur mit verschiedenen aerodynamischen Lastmodellen auszuwählen. Eine Übersicht über grundlegende Komponenten und Teilsysteme des Referenzmodells ist in Abbildung 4.11 (links) gezeigt. Die Modellierung der gesamten Struktur, bestehend aus flexiblen Rotorblättern und einer flexiblen Turmstruktur, ist als flexibles Mehrkörpersystem implementiert, um die Anzahl an Freiheitsgraden im Vergleich zu einem Ansatz basierend auf der FEM zu reduzieren. Die Freiheitsgrade der flexiblen Körper werden mittels modaler Reduktion reduziert, um eine effiziente Berechnung zu gewährleisten [TGS+14, SRB+17, LT17].

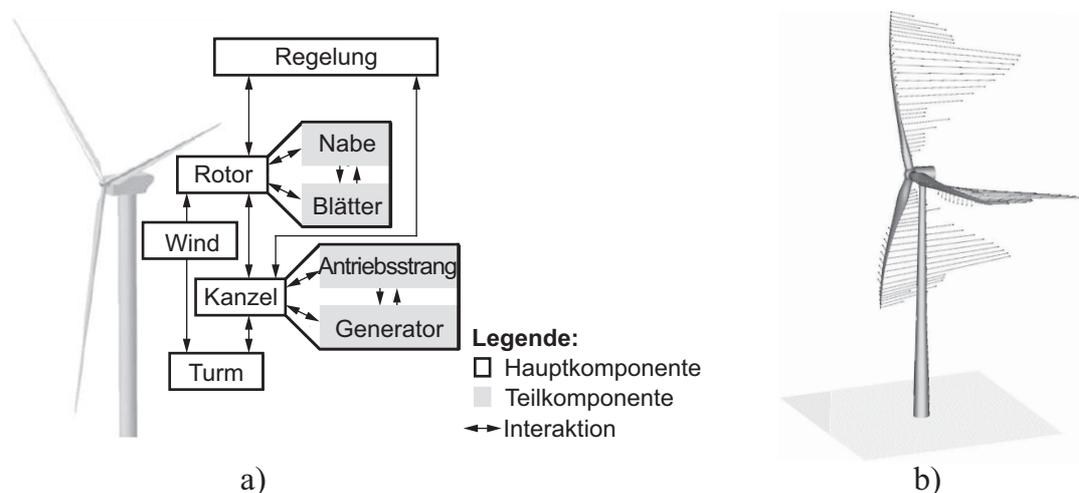


Abbildung 4.11: Struktur der Implementierung einer onshore WEA in der *OneWind*[®] Modelica Bibliothek [TGS+14] (a) und Animation auf Basis einer Simulation mit Vektoren der Windgeschwindigkeiten an den Rotorblättern (b).

Die Anregung dieser gekoppelten flexiblen Komponenten, Rotorblätter und Turmstruktur, erfolgt über aerodynamische Lasten. Aerodynamische Kräfte und Drehmomente

¹⁸Der Autor dankt dem Fraunhofer Institute for Wind Energy and Energy System Technology IWES Northwest für die Bereitstellung des Referenzmodells IWT-7.5-164 Rev. 2.5.

wirken durch Windfelder auf die Rotorblätter oder die Turmstruktur, die mittels segmentierter aerodynamischer Lastmodelle entlang der jeweiligen Komponenten berechnet werden. Die aerodynamischen Lasten wirken als äußere Kräfte und Drehmomente auf die flexiblen Strukturen. Die jeweilige flexible Struktur wird an diskreten Punkten an ein entsprechendes aerodynamisches Lastmodell gekoppelt und ermöglicht die Simulation der Auslenkung der Struktur für gegebene aerodynamische Lasten [TGS+14].

Die *OneWind*[®] Modelica Bibliothek stellt unterschiedliche Lastmodelle und Schnittstellen zu externen Lastmodellen für eine Koppelung zur Verfügung. Die maßgeblichen Lastfälle basieren auf einer aerodynamischen Anregung durch verschiedene Windfelder. Für diese Windfelder stehen deterministische und stochastische Modelle zur Verfügung. Während deterministische Modelle Kombinationen verschiedener Windböen verwenden, wird für die Erzeugung turbulenter Windfelder das stochastische Simulationswerkzeug *TurbSim* [JK12] über Schnittstellen integriert. *TurbSim* stellt dabei mittels eines stochastischen Modells Zeitreihen von Windfeldern auf Basis numerischer Simulation bereit, die als Lastfälle verwendet werden [SRB+17]. Darüber hinaus sind für offshore WEA fluiddynamischen Lastmodelle zur Abbildung von Strömung und Wellengang sowie zur zusätzlichen Abbildung von Auftrieb bei schwimmenden Anlagen [LT17] vorhanden. Die auf die WEA wirkenden Lasten können eine Kombination der Lastfälle der verschiedenen aero- und fluiddynamischen Anregungen sowie der elektromechanischen Last des Generators darstellen [LT17, TGS+14].

Der Generator sowie weitere Komponenten und Teilsysteme des Triebstrangs sind im Hinblick auf ihre mechanischen Eigenschaften implementiert und durch detailliertere Modelle austauschbar. Ein Bus-System stellt die Verbindung zwischen allen Komponenten und Teilsystemen her, die Informationen für die Regelung der WEA bereitstellen.

Innerhalb der *OneWind*[®] Modelica Bibliothek sind verschiedene Regelungsstrategien für einen drehzahlvariablen Teil- und Vollastbetrieb der IWT-7.5-164 hinterlegt. Diese umfassen, neben klassischen Ansätzen der Regelungstechnik für die individuelle und gemeinsame Regelung der Rotorblattneigungswinkel und des Generator Drehmoments, fortgeschrittene Verfahren der Regelungstechnik unter anderem für die optimale Einstellung des Generators hinsichtlich der Effizienz [SRB+17]. Eine detaillierte Beschreibung der implementierten Regelungsstrategien ist in [SRB+17] gegeben.

Im Rahmen der Anwendung der integrierten Modellierung wird eine onshore installierte IWT-7.5-164 mit jeweils modalreduzierten flexiblen Rotorblatt- und Turmstrukturen betrachtet. Es wird die Fraunhofer IWES Referenzregelung für eine Regelung aller Rotorblattneigungswinkel für drehzahlvariablen Teil- und Vollastbetrieb verwendet [SRB+17]. Das Referenzmodell IWT-7.5-164 weist eine hohe Komplexität auf, da es verschiedene Funktionen und Modellierungsansätze integriert, die Konfigurierbarkeit des gesamten Systemmodells ermöglicht, die Verwendung starrer oder flexibler Mehrkörpersysteme enthält und verschiedene Regelungsstrategien mit individueller Einstellung der Rotorblattneigungswinkel bereitstellt. Die Anwendung der integrierten Modellierung fokussiert die Verwendung lokaler Zuverlässigkeitsschätzer für die Betriebsfestigkeitsanalyse der Rotorblätter aus Faserverbundmaterial und der Turmstruktur auf Basis der Simulation eines Lastfalls.

Das Referenzmodell IWT-7.5-164 wird für eine Analyse der Systemzuverlässigkeit mithilfe der integrierten Modellierung für konstante Windverhältnisse mit einer mittleren Windgeschwindigkeit auf Nabenhöhe $v_{Nabe} = 11,4 \text{ m/s}$ aus dem Stillstand angeregt und bis auf Nennleistung beschleunigt. Die Simulation mit einer Dauer von 120 s liefert alle

für die Parametrierung der lokalen Zuverlässigkeitsschätzer innerhalb der integrierten Modellierung notwendigen dynamischen Lasten auf die Komponenten und Teilsysteme. Für die spätere Abbildung zusätzlicher Umweltbedingungen wird angenommen, dass die IWT-7.5-164 onshore im Landkreis Paderborn installiert ist.

4.2.2 Modellierungsaspekte der Zuverlässigkeit

Das Referenzmodell wurde insbesondere für die Auslegung neuartiger Rotorblattdesigns entwickelt. Die Modellierungstiefe ist hinsichtlich des dynamischen Verhaltens der Rotorblätter und der Turmstruktur im Vergleich zu weiteren Komponenten und Teilsystemen ausgeprägter gewählt worden. Beispielsweise sind Komponenten und Teilsysteme zu nennen, die eine Verstellung der Rotorblattneigungswinkel gewährleisten, dem Schwenken der Kanzel dienen oder dem Triebstrang zugehörig sind. Für die Komponenten und Teilsysteme, die in reduzierter Modellierungstiefe abgebildet sind, fehlen Informationen für die Parametrierung geeigneter lokaler Zuverlässigkeitsschätzer, sodass diese vernachlässigt werden. Die Analyse der Systemzuverlässigkeit greift auf anwendungsspezifische Methoden der Windenergiebranche zurück. Die Methoden werden in den lokalen Zuverlässigkeitsschätzern für die Betriebsfestigkeitsanalyse der flexiblen Strukturen eingesetzt oder erweitern genormte Methoden der Wälzlagerberechnung (DIN ISO 281 [DI10]) um Berechnungsvorschriften zur Berücksichtigung der Dimensionen der Komponenten. Darüber hinaus werden weitere Modellierungsaspekte, wie ein externes Ereignis als gemeinsame Fehlerursache, hinzugefügt.

4.2.2.1 Spezialisierte lokale Zuverlässigkeitsschätzer

WEAs sind strukturell stark beanspruchte Systeme, sodass eine Analyse der Systemzuverlässigkeit die Betriebsfestigkeit der einzelnen Komponenten, insbesondere der Rotorblatt- und Turmstrukturen, berücksichtigen muss. Dazu wurde vom U.S. National Renewable Energy Laboratory (NREL) ein Softwarewerkzeug (Mlife [Hay12, HB12]) entwickelt, um Betriebsfestigkeitsanalysen auf Basis gekoppelter Lastsimulationen von gesamten WEAs durchzuführen. Mlife nutzt die lineare Schadensakkumulation nach Palmgren und Miner [San08] unter Berücksichtigung verschiedener Aspekte, wie der Mittelspannungskorrektur nach Goodman [HB12], wobei das Spannungsverhältnis zwischen dem simulierten Lastkollektiv und den der Wöhlerlinie zugrundeliegenden Daten angeglichen wird. Die Lebensdauer einer Komponente wird über eine Extrapolation des akkumulierten Schadens auf Basis des Lastkollektivs bestimmt. Werden unterschiedliche Lastfälle, verschiedene Windanregungen, simuliert, werden diese auf Basis der Windgeschwindigkeit über die Häufigkeit der dadurch hervorgerufenen Lasten gewichtet und in die Lebensdauerberechnung mit einbezogen. Die Schadensakkumulation in Mlife verwendet zur Bildung des Lastkollektivs das Rainflow-Verfahren [San08]. Die Belastungen auf die Strukturbauteile der WEA müssen für eine Betriebsfestigkeitsanalyse in Beanspruchungen, Spannungen und Dehnungen übertragen werden. Anstatt eines FE-Modells der jeweiligen betrachteten Struktur, nutzt Mlife die Berechnung einer schadensäquivalenten Belastung. Diese äquivalente Belastung ist durch eine konstante Amplitude und Frequenz gekennzeichnet, welche dieselbe Schädigung wie das zugrundeliegende Lastspektrum hervorruft [HB12]. Die Lebensdauer der Komponente wird schließlich über die Schwingspiele des Lastkollektivs bis zum Ausfall bestimmt. Für die Verwendung in einem lokalen Zuverlässigkeits-

schätzer wird die Lebensdauer der betrachteten Komponente für verschiedene normalverteilte Wöhlerexponenten [San08] berechnet und eine Weibull-Verteilung über diese Daten angenähert.

Die Topologie des Referenzmodells enthält keine Abbildungen einer Rotorblattwälzlagerung, welche die Verstellung des Neigungswinkels ermöglicht. Daher wird der lokale Zuverlässigkeitsschätzer für ein Rotorblattwälzlager als eine Fehlerursache des jeweiligen Rotorblatts modelliert. Die jeweiligen Rotorblätter stellen die benötigten dynamischen Lasten auf eine solche Lagerung als Schnittgrößen mit der Nabe zur Verfügung. Schwack et. al. [SSP+16] entwickelten ein zweireihiges Vierpunktkugellager, das zur Lagerung eines Rotorblatts in der betrachteten WEA IWT-7.5-164 eingesetzt werden könnte. Die konstruktiven Parameter sind neben den dynamischen Lasten bekannt, sodass ein lokaler Zuverlässigkeitsschätzer vollständig parametrierbar ist. Dieser lokale Zuverlässigkeitsschätzer legt die DIN ISO 281 [DI10] zugrunde, die auf der Annahme rotierender Wälzlager beruht. Wälzlagerungen in Rotorblätter von WEAs erfahren hingegen oszillierende Bewegungen ohne vollständige Rotationen auszuführen. Dies führt dazu, dass für oszillierende Bewegungen bei auftretenden radialen Lasten nicht alle Wälzkörper und nur kleine Bereiche der Laufflächen beansprucht werden, während die Verteilung des Schmiermittels innerhalb des Wälzlagers unzureichend sein kann. Es werden daher weiterführende Ansätze auf Basis der DIN ISO 281 betrachtet, die diese Phänomene berücksichtigen [HRB09, SSP+16].

Nach DIN ISO 281 wird die nominelle Lebensdauer L_{10} in 10^6 Umdrehungen für statistisch 10 % ausgefallene Wälzlager nach Gleichung 4.8 berechnet.

$$L_{10} = \left(\frac{C}{P} \right)^p \quad (4.8)$$

C ist die dynamische Tragzahl des Wälzlager, P die dynamische äquivalente Belastung und p ein Berechnungsbeiwert des Wälzlagers. Die, der Bestimmung der dynamischen Tragzahl C , zugrundeliegenden Gleichungen sind abhängig von der Gestalt, Material, Hauptlastrichtung und Herstellungsgenauigkeit des Wälzlagers. Die Abhängigkeiten von Geometrie, Werkstoff und Herstellungsgenauigkeit der einzelnen Bauteile des Wälzlagers sind durch den Berechnungsbeiwert f_c beschrieben, der in die Berechnung der dynamischen Tragzahl C eingeht. Der Berechnungsbeiwert f_c wird in [HRB09] auf Basis der DIN ISO 281 für Wälzlager mit großen Durchmesser extrapoliert und stellt eine der Erweiterungen dar. Die Gestaltungsrichtlinie zur Berechnung von Rotorblattlagerungen des NREL [HRB09] empfiehlt die Bestimmung der dynamischen äquivalenten Belastung P_{ea} (vergleiche P in Gleichung 4.8) mittels:

$$P_{ea} = \left(\frac{\sum_{l=1}^n p_l^p f_l t_l \phi_l}{\sum_{l=1}^n f_l t_l \phi_l} \right)^{1/p} \quad (4.9)$$

mit n Anzahl an Klassen oder Lastschritten, f_l mittlere Frequenz der Oszillation in l -ter Klasse oder Lastschritt, t_l relative zeitliche Dauer der l -ten Klasse oder des l -ten Lastschritts bezogen auf die Simulationsdauer und ϕ_l mittlere Winkelamplitude. Dabei entspricht p_l der Last innerhalb der l -ten Klasse oder des l -ten Lastschritts:

$$p_l = 0,75 F_{r,l} + F_{a,l} + \frac{\sqrt{M_{x,l}^2 + M_{y,l}^2}}{0,5 D_{pw}} \quad (4.10)$$

mit $F_{r,l}$ und $F_{a,l}$ als dynamische radial und axial Kräfte, $M_{x,l}$ und $M_{y,l}$ als radiale Biegemomente und D_{pw} als Teilkreisdurchmesser des Wälzlagers [SSP+16, HRB09]. Damit ist die Lebensdauer des Wälzlagers eines Rotorblatts in Stunden nach [SSP+16] als Erweiterung zu Gleichung 4.8 zu berechnen nach:

$$L_{10,h,WEA} = \left(\frac{C}{P_{ea}} \right)^p \cdot 10^6 \cdot \frac{1}{3600 \sum_{l=1}^n f_l t_l} \cdot \frac{1}{0,3617}. \quad (4.11)$$

Die Summe $\sum_{l=1}^n f_l t_l$ beschreibt hier die mittlere Oszillationsfrequenz in Hertz, während der zweite Bruch auf die mittlere Betätigungsdauer der Regelung der Rotorblattneigungswinkel bezugnimmt. Diese sei nach Schwack et. al. nur zu 36,17% der Betriebszeit einer WEA aktiv und daher in der Berechnung der Wälzlagerlebensdauer zu berücksichtigen [SSP+16].

4.2.2.2 Externes Ereignis als gemeinsame Fehlerursache

WEAs sind aufgrund ihrer exponierten Installationsstandorte und Höhe einem erhöhten Risiko von Blitzeinschlägen ausgesetzt. Obwohl aktuelle WEAs mit Blitzschutzmaßnahmen ausgerüstet sind, sind Blitzeinschläge dennoch ursächlich für 3,66% der Störfälle [DHR07]. Versagt das Blitzschutzsystem, treten teils katastrophale Schäden an Rotorblättern, Haupt- und Rotorblattlagerung, Generator, Elektronik, falls vorhanden Getriebe und weiteren Komponenten und Teilsystemen auf, die zum Totalverlust des Gesamtsystems, beispielsweise durch Brand, führen können [DHR07, Cai19]. Blitzeinschläge werden daher als externe Ereignisse in der integrierten Modellierung berücksichtigt, die als gemeinsame Fehlerursachen für die drei Rotorblätter und das Hauptlager des Triebstrangs abgebildet werden. Die Integration erfolgt analog zu dem in Abschnitt 3.5.2.4 beschriebenen Vorgehen für das Hinzufügen weiterer Modellierungsaspekte.

Die Abbildung von Blitzeinschlägen setzt die Bestimmung der Wahrscheinlichkeit eines solchen Ereignisses voraus, die hier anhand des folgenden Vorgehens in grober Näherung bestimmt wird. In Anlehnung an IEC 62305 [Int13] wird das Blitzkugelverfahren angewandt, um einen Gefährdungsbereich für Blitzeinschläge für die WEA zu bestimmen. Dabei wird eine Kugel um das zu schützende Objekt abgerollt, um damit mögliche Eintrittsstellen eines Blitzes zu ermitteln [KSM12]. Das Verfahren ist in Abbildung 4.12 gezeigt, wobei der Gefährdungsbereich als Rechteck angenähert wird. Die Wahrscheinlich-

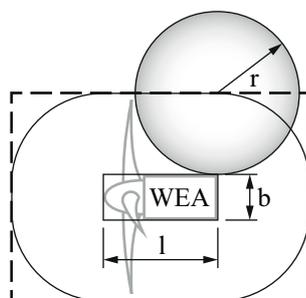


Abbildung 4.12: Abschätzung des Gefährdungsbereichs für Blitzeinschläge der WEA (nach IEC 62305).

keit eines Blitzeinschlags in die WEA wird über folgenden Zusammenhang abgeschätzt:

$$P_{Bl} = \frac{n_{Bl,Pb} \cdot a_{Bl,WEA}}{a_{Pb}} \quad (4.12)$$

wobei $n_{Bl,Pb} = 0,208$ 1/h die Anzahl an Blitzeinschlägen im Landkreis Paderborn im Jahr 2017 pro Stunde [Bli18], $a_{Pb} = 1245,56$ km² die Fläche des Landkreises Paderborn [Bli18] und $a_{Bl,WEA} = (2r + b)(2r + l)$ die Fläche des Gefährdungsbereichs für Blitzeinschläge nach Abbildung 4.12 mit $r = 60$ m, $b = 5$ m und $l = 15$ m beschreiben. Aufgrund nicht verfügbarer Informationen bezüglich der Verteilung über den Betrachtungszeitraum, wird eine stetige Gleichverteilung angenommen mit der Verteilungsfunktion:

$$F(t) = \begin{cases} 0, & \text{für } t \leq a, \\ \frac{t-a}{b-a}, & \text{für } a < t < b, \\ 1, & \text{für } t \geq b. \end{cases} \quad (4.13)$$

Die bedingten Wahrscheinlichkeiten für das Systemzuverlässigkeitsmodell werden analog zu Gleichung 3.12 für die Weibull-Verteilung erzeugt. Das externe Ereignis ist damit vollständig definiert, sodass die Abbildung als gemeinsame Fehlerursache der drei Rotorblätter und des Hauptlagers umgesetzt wird. Zunächst werden zusätzliche Kanten vom externen Ereignis zu den Ausfallzuständen hinzugefügt (vergleiche Abschnitt 3.5.2.4), um den gemeinsamen Einfluss des externen Ereignisses abzubilden. Der Einfluss des externen Ereignisses als individuelle oder gemeinsame Fehlerursache wird über den Kopplungsfaktor $\eta = 0,5$ beschrieben (vergleiche Abschnitt 3.4.3.2). Die Auswirkungen eines Blitzschlags auf die Rotorblätter und das Hauptlager sind für die individuelle und die gemeinsame Fehlerursache jeweils mit einer Unsicherheit $P(\{\text{Rotorblatt}, \text{Hauptlager}\} | \text{Blitz}) = 0,4$ modelliert (vergleiche Abschnitt 3.4.3.1).

4.2.3 Ergebnisse

Die Systemzuverlässigkeitsanalyse der IWT-7.5-164 mittels der integrierten Modellierung liefert die in Abbildung 4.13 dargestellten Ergebnisse der Gesamtanlage und Eintrittswahrscheinlichkeiten ausgewählter Fehlerursachen. Daher werden nur die Fehlerursachen eines Rotorblatts dargestellt, wobei F6_Rotorblatt dem Wälzlager der Neigungsverstellung entspricht. Die weiteren Fehlerursachen (F2–F5) entsprechen Fehlerursachen der Struktur des Rotorblatts aufgrund unterschiedlicher Belastungen. Der Fehlerursache F2_Rotorblatt liegt als Belastung das Biegemoment in Schlagrichtung des Rotorblatts zugrunde. Das wirkende Biegemoment führt auf Basis des lokalen Zuverlässigkeitsschätzers (mlife) zu keiner signifikanten Schädigung, sodass die Zuverlässigkeit nicht abnimmt. Die Fehlerursache F5_Rotorblatt wird auf Basis der dynamischen Trägheitskräfte entlang der Längsachse berechnet und zeigt hinsichtlich der Zuverlässigkeit ein gleiches Verhalten wie die Fehlerursache F2_Rotorblatt. Die Fehlerursachen F3_Rotorblatt und F4_Rotorblatt entsprechen dem Biegemoment über die Längsachse des Rotorblatts sowie der resultierenden Kraft an der Nabe und zeigen jeweils sehr hohe Zuverlässigkeiten. Exemplarisch für die Turmstruktur ist die Fehlerursache F1_Turm berücksichtigt, die das Biegemoment am Übergang zwischen Turm und Gründung verwendet. Im Vergleich zu den Fehlerursachen der Rotorblätter ist die Zuverlässigkeit der Turmstruktur deutlich reduziert. Dies erscheint hinsichtlich des Aufbaus der WEA, große Masse des Rotors und

Kanzel auf schlanker Turmstruktur, plausibel, da über die Turmstruktur sämtliche Lasten aufgenommen werden müssen. Der Fehlerursache des Hauptlagers F1_Hauptlager liegt ein lokaler Zuverlässigkeitsschätzer nach DIN ISO 281 (vergleiche WEA-spezifische Änderungen in Abschnitt 4.2.2.1) zugrunde. Die Abschätzung der Wahrscheinlichkeit eines Blitzeinschlags (Abschnitt 4.2.2.2) liegt in einem ähnlichen Bereich wie die Zuverlässigkeit der verbauten Wälzlager.

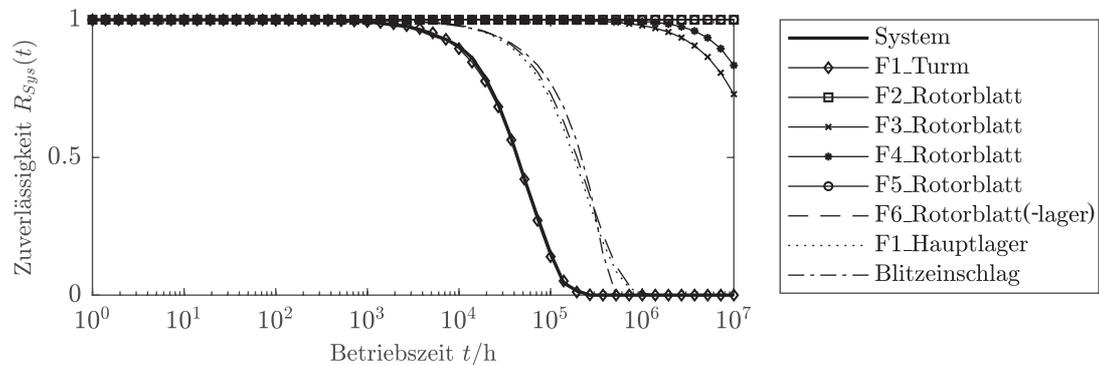


Abbildung 4.13: Systemzuverlässigkeit der WEA und Eintrittswahrscheinlichkeiten ausgewählter Fehlerursachen.

Die Ergebnisse der Systemzuverlässigkeitsanalyse zeigen, dass die integrierte Modellierung für komplexe Systemmodelle, wie das Referenzmodell IWT-7.7-164, erfolgreich anwendbar ist und branchenübliche Entwicklungswerkzeuge (Mlife) in das Vorgehen implementierbar sind. Die Integrität vorhandener Entwicklungsmodelle, wie das Referenzmodell, wird gewahrt und um lokale Zuverlässigkeitsschätzer erweitert. Zudem sind zusätzliche Modellierungsaspekte zur Abbildung der Zuverlässigkeit, wie Blitzeinschläge als gemeinsame Fehlerursache, integrierbar. Somit ermöglicht die integrierte Modellierung eine verallgemeinerte Methode für die geschlossene Modellierung der Zuverlässigkeit und des dynamischen Systemverhaltens ohne auf eine ausschließliche anwendungsspezifische Implementierung der Zuverlässigkeitsanalyse angewiesen zu sein. Die Durchführung von Zuverlässigkeitsanalysen während des Entwicklungsprozesses kann dadurch effizienter umgesetzt werden.

4.3 Umsetzung eines Digitalen Zwillings für die Instandhaltung

Die integrierte Modellierung bietet weitreichende Potenziale für die Umsetzung eines Digitalen Zwillings aus bestehenden Entwurfsmodellen mechatronischer Systeme [KHS+19, KBS19]. In diesem Anwendungsbeispiel soll eine Umsetzung an einem Wälzlagerprüfstand durchgeführt werden. Betrachtet wird der modulare Wälzlagerprüfstand in Abbildung 4.14, bestehend aus einem Antriebsmotor (1), einer Drehzahl- und -momentenmesswelle (2), einem Prüfmodul (3) zur gezielten Belastung eines Wälzlagers, einer Trägheit (4) und eines Abtriebsmotors (5), um den Antriebsstrang zu verspannen. Darüber hinaus ist der Prüfstand mit verschiedenen Sensoren und einer Datenaufnahme ausgestattet, um die Gehäuse- und Schmiermitteltemperatur im Prüfmodul, die Motorströme an An- und

Abtriebsseite, Beschleunigung am Gehäuse des Prüfmoduls und Radialkraft auf das Testlager zu erfassen. In das Prüfmodul (3) werden gezielt vorgeschädigte Wälzlager eingebaut und zielgerichtet mittels einer einstellbaren Radiallast oder Verkipfung des Wälzlagers belastet und Zustandsüberwachungsdaten aufgenommen. Die Testlager, deren Schädigungszustand bekannt ist, werden nur zur Datenaufnahme im Prüfstand verwendet, eine weitere Schädigung ist nicht vorgesehen. Die resultierenden Datensätze beinhalten Schadenssignaturen in den Messgrößen, die dem bekannten Schädigungszustand des geprüften Wälzlagers entsprechen. Auf Basis dieser Datensätze sollen Methoden der Zustandsüberwachung entwickelt und validiert werden. Eine detaillierte Beschreibung des Prüfstands, des Prüfablaufs und der aufgenommenen Datensätze ist in [Les17] zu finden.¹⁹

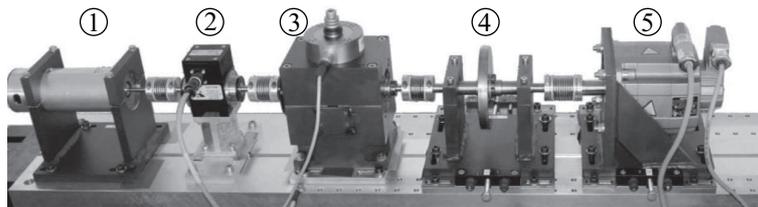


Abbildung 4.14: Wälzlagerprüfstand mit fünf Modulen zur Erzeugung von Zustandsüberwachungsdaten (nach [LKZ+16]).

Der zu entwickelnde Digitale Zwilling soll die Instandhaltung in verschiedenen Aspekten unterstützen. Der Digitale Zwilling liefert eine ganzheitliche Abbildung der Systemzuverlässigkeit zu jedem beliebigen Betrachtungszeitpunkt, die für kritische Komponenten und Teilsysteme durch eine Zustandsüberwachung aktualisiert wird. Damit ist eine Diagnose im Falle eines Systemausfalls möglich, die modellbasiert auf die wahrscheinlichste Fehlerursache schließen lässt. Für Komponenten oder Teilsysteme, die parallel arbeiten ohne redundant ausgelegt zu sein, ist die Zustandsüberwachung nur einer Einheit denkbar, da auf den Zustand der übrigen Einheiten geschlossen wird. Durch die Weiterverwendung von Modellen aus dem Entwurf des zu überwachenden Systems ist mittels der integrierten Modellierung eine hohe Durchgängigkeit der Modelle über die Lebenszyklusphasen gewährleistet.

4.3.1 Datenbasis und zustandsbasierte Diagnose

In den Prüfstand werden für die Datenaufnahme bereits vorgeschädigte Wälzlager eingesetzt, daher liefert der Prüfstand Datensätze für zahlreiche Ausprägungen verschiedener Schädigungszustände. Diese liegen jedoch nur diskret für jedes geprüfte Wälzlager vor. Eine kontinuierliche Schätzung des aktuellen Schädigungszustands eines Wälzlagers über die Betriebsdauer liegt nicht vor.

Für die Umsetzung eines Digitalen Zwillings soll auf die bestehenden Datensätze des 6203 Rillenkugellagers zurückgegriffen werden. Für dieses Wälzlager wurden durch Lessmeier [LKZ+16] verschiedene Schädigungszustände durch beschleunigte Lebensdauererprobung erzeugt, klassifiziert und die Fehlerursachen (Oberflächenrisse verschiedener

¹⁹Der Autor dankt Prof. Dr.-Ing. D. Zimmer, Dr.-Ing. C. Lessmeier und dem Lehrstuhl für Konstruktions- und Antriebstechnik (KAT) der Universität Paderborn für die Bereitstellung der Zustandsüberwachungsdaten unter: mb.uni-paderborn.de/kat/datacenter.

Ausprägung in den Laufflächen des Innen- und Außenrings, sog. Pitting) analysiert. In Abbildung 4.15 sind die identifizierten Klassen beispielhaft über zwei Merkmale (Wavelet-Energie und Peak-to-Peak) aus Beschleunigungssignalen dargestellt. Die Beschleunigungen wurden am Gehäuse des Prüfmoduls aufgenommen. Auf Basis einer Analyse der Schädigungen wird den identifizierten Klassen direkt eine Fehlerursache zugeordnet. Die unterschiedlichen Ausprägungen des Pittings bezeichnen hier die Größe einzelner Ausbrüche (vergleiche Pitting 1-3). Daneben wird noch ein Schädigungszustand unterschieden für den mehrere Ausbrüche geringer Größe auftreten (vergleiche multiples Pitting 1).

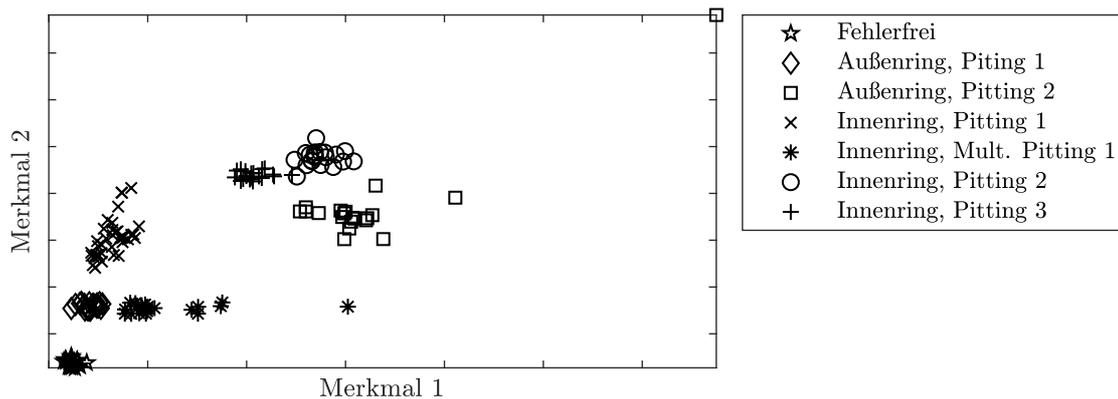


Abbildung 4.15: Identifizierte Klassen von Schädigungszuständen der untersuchten Wälzlager (nach [KHS+19]).

Auf Basis der Datensätze wird eine datenbasierte Diagnose der Schädigungszustände mittels einer Support Vector Machine (SVM) umgesetzt. Die zu entwickelnde Diagnose stellt ein Mehrklassen-Klassifikationsproblem dar, wobei SVM zunächst für binäre Klassifikationsaufgaben anwendbar sind. Für die Mehrklassen-Klassifikation werden für n_c Klassen $n_c(n_c - 1)/2$ binäre Klassifikatoren aufgebaut, die alle Klassen gegeneinander bewerten und für die jeweils entsprechenden Datensätze trainiert werden. Auf Basis der paarweisen Kopplung werden die Ergebnisse der binären Klassifikatoren zu einem Wahrscheinlichkeitsmaß zur Beschreibung der Zugehörigkeit zu den Klassen zusammengefügt [WLW04]. Das Ergebnis der SVM für eine Messung ist eine Zeilenmatrix der Zugehörigkeitswahrscheinlichkeiten:

$$\mathbf{p}_{SVM} = [p_{Ff}, p_{A1}, p_{A2}, p_{I1}, p_{IM1}, p_{I2}, p_{I3}] \quad (4.14)$$

zu den einzelnen Klassen, wobei $\sum \mathbf{p}_{SVM} = 1$ gilt. Die Reihenfolge der Werte in der Zeilenmatrix der Zugehörigkeitswahrscheinlichkeiten \mathbf{p}_{SVM} entspricht den Klassen in Abbildung 4.15. Die Parameter der verwendeten SVM werden mittels Particle Swarm Optimization (PSO) auf das Klassifikationsproblem angepasst [Kim16]. Die SVM wird auf verschiedene Merkmale der Beschleunigungssignale trainiert. Hierzu werden 23 Merkmale aus dem Zeit-, Frequenz- und Zeitfrequenzbereich berechnet und neun Merkmale ausgewählt. Diese Merkmalsauswahl stellt die Grundlage für eine hohe Klassifikationsgüte dar. Das trainierte SVM-Modell wird anhand von Testdatensätzen validiert und erzielt sehr gute Klassifikationsgüten von 100% im Vergleich zu einer SVM mit händisch angepassten Parametern mit einer Klassifikationsgüte von 92,5% [KHS+19, KBS19].

4.3.2 Ableitung des Systemzuverlässigkeitsmodells

Die Ableitung des Systemzuverlässigkeitsmodells erfolgt mittels der integrierten Modellierung auf Basis des Mehrkörpermodells des modularen Wälzlagerprüfstands in Abbildung 4.16. Das Mehrkörpermodell ist in MSC Adams / View implementiert. Das Mehrkörpersystem bildet reduziert das dynamische Verhalten des Prüfstands ab und beinhaltet alle wesentlichen Komponenten. Diese sind als Starrkörper mit jeweils einem rotatorischen Freiheitsgrad modelliert. Die einzelnen Module sind über Torsionssteifigkeiten zur Abbildung der Kupplungen miteinander gekoppelt. Das Testlager wird mit einer konstanten Radialkraft $F_{\text{Prüf}}$ beaufschlagt, um die im Prüfmodul aufgebrachte Last zu simulieren. Der modulare Wälzlagerprüfstand wird stets im stationären Betrieb verwendet, die Antriebsdrehzahl $n_{\text{Antrieb}} = 1500 \text{ min}^{-1}$ und die aufgeprägte Radiallast $F_{\text{Prüf}} = 400 \text{ N}$ sind konstant über die Betriebsdauer, sodass auf eine detaillierte Modellierung des dynamischen Systemverhaltens mit Drehzahlregelung verzichtet wurde.

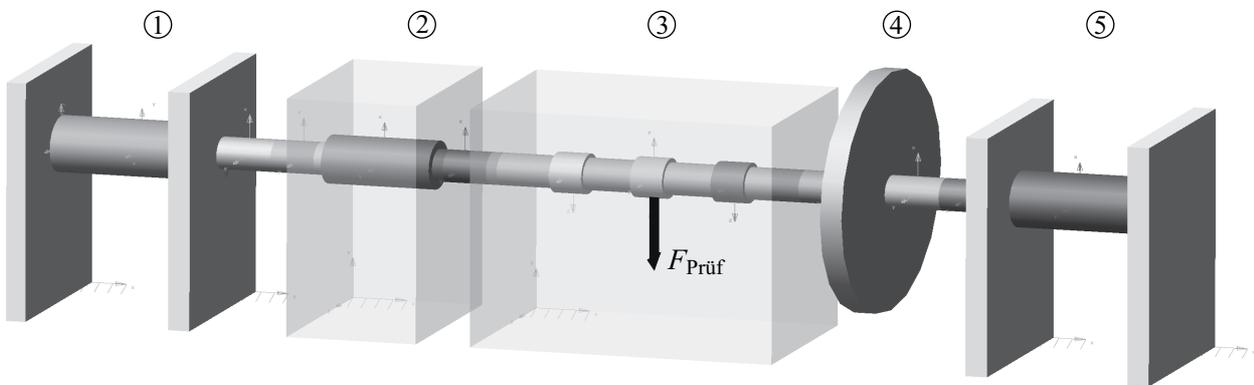


Abbildung 4.16: Mehrkörpermodell des modularen Wälzlagerprüfstands aus dem Entwurf. Die Positionsnummern entsprechen Abbildung 4.14.

Die Abbildung der lokalen Zuverlässigkeiten basiert im Wesentlichen auf einer Betrachtung der Wälzlager (je zwei Wälzlager pro Modul und das Testlager). Zusätzlich werden die Isolation der Wicklungen im An- und Abtriebsmotor (lokaler Zuverlässigkeitsschätzer nach [Bra92]), der Spulen in der Messwelle (nach [US 95]) sowie die Thermoelemente zur Überwachung der Gehäuse- und Schmiermitteltemperatur im Prüfmodul (nach [US 95]) betrachtet.

Die Abbildung der Systemzuverlässigkeit erfolgt ohne Berücksichtigung weiterer Modellierungsaspekte, wie Redundanzen, Ausfälle gemeinsamer Ursache oder Unsicherheiten, sodass sich das DBN in Abbildung 4.17 in der Darstellung in zwei Zeitscheiben ergibt. Die Knoten der Fehlerursache für das Testlager $F1_Testlager_i/i+1$ und die Knoten des Ausfallzustands $Testlager_i/i+1$ sind jeweils hervorgehoben und für die Umsetzung des Digitalen Zwillings von besonderer Bedeutung.

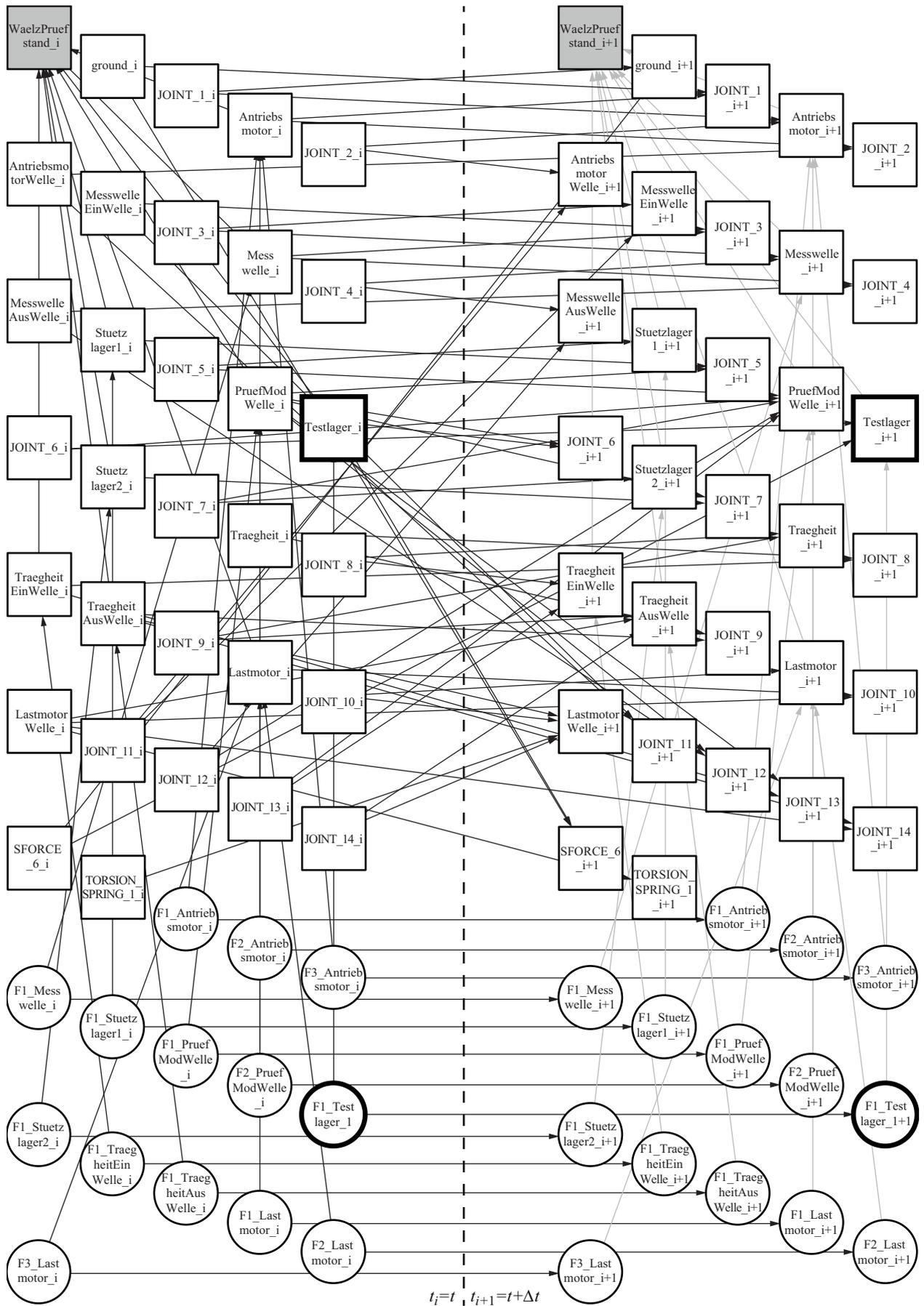


Abbildung 4.17: Systemzuverlässigkeitsmodell des Wälzlagerprüfstands im Entwurf.

4.3.3 Erstellung des Digitalen Zwillings

Der Digitale Zwilling wird auf Basis des in Abbildung 4.17 dargestellten Systemzuverlässigkeitsmodells und der Zustandsüberwachung aus Abschnitt 4.3.1 umgesetzt. Aus der vorhandenen Datenbasis der Zustandsüberwachung [LKZ+16] werden die in Tabelle 4.1 gezeigten Datensätze der Zugehörigkeitswahrscheinlichkeiten \mathbf{p}_{SVM} für die Integration in den Digitalen Zwilling verwendet. Die Datensätze sind aufgrund einer hohen Zugehörigkeitswahrscheinlichkeit zu jeweils einer Klasse ausgewählt worden, sodass eine dominierende Wahrscheinlichkeit für eine Klasse enthalten ist. Die diskreten Zustandsbeobachtungen sind aufgrund der Art der Durchführung und der Vorbereitung der Experimente keinem bestimmten Zeitpunkt der Betriebsdauer t_i zuzuordnen und in keine kausale Reihenfolge ihres Auftretens zu bringen [LKZ+16]. Daher wird angenommen, dass die verschiedenen Zustandsüberwachungsdaten (Tabelle 4.1) jeweils zu demselben Betriebszeitpunkt $t_E = 1000$ h zur Verfügung stehen, sodass die Auswirkungen verschiedener Schädigungszustände des überwachten Wälzlagers (Testlager) auf die Systemzuverlässigkeit gegenübergestellt werden können.

Diagnostizierte Klasse	Wahrscheinlichkeit der Klassenzugehörigkeit \mathbf{p}_{SVM}							
	Fehlerfrei	Außenring, Pitting 1	Außenring, Pitting 2	Innenring, Pitting 1	Innenring, Mult. Pitting 1	Innenring, Pitting 2	Innenring, Pitting 3	
Fehlerfrei	0,774	0,094	0,016	0,047	0,028	0,020	0,022	
Außenring, Pitting 1	0,079	0,765	0,016	0,065	0,034	0,019	0,022	
Außenring, Pitting 2	0,014	0,018	0,784	0,026	0,036	0,042	0,08	
Innenring, Pitting 1	0,013	0,04	0,01	0,855	0,049	0,01	0,023	
Innenring, M. Pitting 1	0,026	0,038	0,02	0,07	0,786	0,022	0,039	
Innenring, Pitting 2	0,013	0,017	0,034	0,047	0,02	0,808	0,063	
Innenring, Pitting 3	0,018	0,024	0,045	0,075	0,071	0,046	0,72	

Tabelle 4.1: Diagnosewahrscheinlichkeiten \mathbf{p}_{SVM} für gewählte Evidenzen zu jeder möglichen Klasse.

Ein Datensatz der Zustandsüberwachung \mathbf{p}_{SVM} stellt für das Systemzuverlässigkeitsmodell eine Beobachtung des Systems mit Messunsicherheiten (entspricht hier der Klassifikationsunsicherheit) dar, die als weiche Evidenz (vergleiche Abschnitt 3.4.2.8) in den entsprechenden Zeitscheiben berücksichtigt wird. Die Beobachtung wird in Zeitscheibe t_E in das Systemzuverlässigkeitsmodell eingebracht und die temporalen Abhängigkeiten zwischen den Zeitscheiben t_{E-1} und t_E werden gelöscht (eine Erläuterung zu diesem Vorgehen ist im Anhang A.2 gegeben). Neben diesen strukturellen Änderungen, werden die CPTs der Fehlerursache $F1_Testlager_i/i+1$ und des Ausfallzustands $Testlager_i/i+1$ für alle Zeitscheiben mit $t_i > t_E$ (vergleiche Abbildung 4.17) angepasst.

Der für Wälzlager verwendete lokale Zuverlässigkeitsschätzer nimmt keine detaillierte Unterscheidung der Ausfallarten vor. So ist es für die Bestimmung der lokalen Zuverlässigkeit unerheblich, ob in einem Wälzlager Pitting in den Laufflächen oder der Bruch eines Wälzkörpers auftritt. Der lokale Zuverlässigkeitsschätzer bildet alle identifizierten Klassen der Zustandsüberwachung als mögliche Fehlerursache aufgrund seiner geringen Modellierungstiefe bereits ab. Daher werden die Zustandsüberwachungsdaten als weiche Evidenz in die Fehlerursache des Testlagers in Zeitscheibe $t_i = t_E = 1000$ h in den Knoten $F1_Testlager_i$ (vergleiche Abbildung 4.17) integriert. Der allgemeine Modellierungsansatz besagt, dass eine Fehlerursache als binäre Zufallsvariable mit den Zuständen *active* und *dormant* modelliert wird (Abschnitt 3.4.3.1). Für die Integration einer weichen Evidenz wird diese Annahme erweitert, sodass eine Evidenz eine beliebige Granularität der Zustände (*active*, ..., *dormant*) aufweisen kann. Daraus ergibt sich die Anforderung, dass die Klassen der Zustandsüberwachung für eine erleichterte Integration in das Systemzuverlässigkeitsmodell angepasst werden müssen. Für die Fehlerursache wird folgender Zusammenhang zwischen Zuständen im Systemzuverlässigkeitsmodell und Klassen der Zustandsüberwachung gewählt: $active \cup \{Außenring, Pitting2; Innenring, Pitting3\}$ und $dormant \cup \{Fehlerfrei\}$. Die übrigen Zustände entsprechen den Klassen, sodass sich für den Knoten der Fehlerursache $F1_Testlager_i$ mit weicher Evidenz \mathbf{p}_{SVM} (Gleichung 4.14) die CPT in Tabelle 4.2 ergibt.

<i>active</i>	<i>Außenring, Pitting 1</i>	<i>Innenring, Pitting 1</i>	<i>Innenring, Mult. Pitting 1</i>	<i>Innenring, Pitting 2</i>	<i>dormant</i>
$p_{A2} + p_{I3}$	p_{A1}	p_{I1}	p_{IM1}	p_{I2}	p_{Ff}

Tabelle 4.2: CPT des Knotens $F1_Testlager_i$ zum Zeitpunkt der Evidenz $t_i = t_E = 1000$ h mit den Wahrscheinlichkeiten der Klassenzugehörigkeit \mathbf{p}_{SVM} .

Der allgemeine Modellierungsansatz gibt vor, dass Fehlerursachen nur auf einen Ausfallzustand wirken, einzige Ausnahme sind Ausfälle gemeinsamer Fehlerursache (vergleiche Abschnitt 3.4.3.2). Die CPT des Ausfallzustands $Testlager_i/i+1$ muss an die neue Anzahl an Zuständen des Elternknotens $F1_Testlager_i/i+1$ angepasst werden, um die Auswirkungen verschiedener Fehlerursachen auf die Ausfallzustände abzubilden. Die Definition von bedingten Auswirkungswahrscheinlichkeiten der Fehlerursachen auf die Ausfallzustände in der CPT des Ausfallzustands $Testlager_i/i+1$ erfolgt auf Basis von Expertenwissen. Die bedingten Auswirkungswahrscheinlichkeiten bezüglich der Fehlerursache $F1_Testlager_i$ auf den Ausfallzustand $Testlager_i$ sind in der CPT in Tabelle 4.3 dargestellt und weisen steigende Wahrscheinlichkeiten für zunehmende Schwere der Schädigung des Testlagers auf. Diese Auswirkungswahrscheinlichkeiten entsprechen Unsicherheiten bezüglich der Auswirkung von Fehlerursachen (vergleiche Abschnitt 3.4.3.1). Der in Tabelle 4.3 dargestellte Einfluss der Fehlerursache $F1_Testlager_i$ auf den Ausfallzustand $Testlager_i$ über die bedingten Wahrscheinlichkeiten gilt zudem für alle weiteren Ausfallzustände in Zeitscheiben $t_i > t_E$ unabhängig weiterer Elternknoten aufgrund temporärer Abhängigkeiten.

Neben den Änderungen des Systemzuverlässigkeitsmodells in Zeitscheibe $t_i = t_E$, wird die CPT der Fehlerursache $F1_Testlager_i+1$ in Zeitscheibe t_{i+1} an die Zustände der Fehlerursache $F1_Testlager_i$ in Zeitscheibe t_i angepasst (vergleiche Tabelle 4.4). Die Feh-

$F1_Testlager_i$	<i>active</i>	<i>Außenring, Pitting 1</i>	<i>Innenring, Pitting 1</i>	<i>Innenring, Multi. Pitting 1</i>	<i>Innenring, Pitting 2</i>	<i>dormant</i>
$P(\text{Testlager}_i = op)$	0	0,4	0,4	0,5	0,7	1
$P(\text{Testlager}_i = fa)$	1	0,6	0,6	0,5	0,3	0

Tabelle 4.3: CPT der Komponente Testlager_i für $t_i = t_E = 1000$ h.

Ursache $F1_Testlager_{i+1}$ besitzt ausschließlich einen Elternknoten $F1_Testlager_i$. Da im Anschluss an die Evidenz in Zeitscheibe t_E keine weiteren Informationen über die weitere Entwicklung der Verteilung der Zustände vorliegen, wird angenommen, dass diese konstant ist. Diese Annahme wird mittels der Einträge auf der Hauptdiagonalen der CPT berücksichtigt. Um eine Prognose der Systemzuverlässigkeit zu ermöglichen, wird die bedingte Zuverlässigkeit $R(t_{i_v+1}|t_{i_v})$ nach Gleichung 3.12 in die CPT für $P(F1_Testlager_{i+1} = active|F1_Testlager_i = active)$ integriert. Die Prognose basiert auf der Verwendung des lokalen Zuverlässigkeitsschätzers, sodass eine virtuelle Zuverlässigkeit $R(t_{i_v})$ entsprechend der aktuellen Wahrscheinlichkeit der Zugehörigkeit zur Klasse *Fehlerfrei* der Zustandsüberwachung p_{Ff} als Referenz gewählt wird. Die virtuelle Zuverlässigkeit $R(t_{i_v})$ entspricht dabei der Zugehörigkeitswahrscheinlichkeit p_{Ff} zu einem Zeitpunkt $t_{i_v} \in \{t_0, \dots, t_n\}$. Aufbauend auf der virtuellen Zuverlässigkeit $R(t_{i_v})$ wird die bedingte Wahrscheinlichkeit $R(t_{i_v+1}|t_{i_v})$ für die Integration in die CPT (Tabelle 4.4) nach Gleichung 3.12 berechnet. Das Vorgehen zur Bestimmung der virtuellen Zuverlässigkeit der überwachten Komponente $R(t_{i_v})$ ist im Anhang A.3 als Pseudocode in Algorithmus 1.3 dargestellt.

Nachdem alle Änderungen am Systemzuverlässigkeitsmodell durchgeführt worden sind, wird dieses für Evidenzen aus der Zustandsüberwachung zu einem Zeitpunkt $t_E = 1000$ h analysiert. Die Systemzuverlässigkeit aus dem Entwurf und basierend auf Evidenzen (vergleiche Tabelle 4.1) werden in Abbildung 4.18 gegenübergestellt.

Die Systemzuverlässigkeit weist einen signifikanten Einfluss durch die Zustandsüberwachungsdaten auf. Dies ist durch mehrere Aspekte begründet. Im Vergleich aller Kompo-

$F1_Testlager_i$	<i>active</i>	<i>Außenring, Pitting 1</i>	<i>Innenring, Pitting 1</i>	<i>Innenring, Multi. Pit. 1</i>	<i>Innenring, Pitting 2</i>	<i>dormant</i>
$P(F1_Testlager_{i+1} = active)$	$1 - R(t_{i_v+1} t_{i_v})$	0	0	0	0	0
$P(F1_Testlager_{i+1} = Außenr., Pit.1)$	0	1	0	0	0	0
$P(F1_Testlager_{i+1} = Innenr., Pit.1)$	0	0	1	0	0	0
$P(F1_Testlager_{i+1} = In., Mult.Pit.1)$	0	0	0	1	0	0
$P(F1_Testlager_{i+1} = Innenr., Pit.2)$	0	0	0	0	1	0
$P(F1_Testlager_{i+1} = dormant)$	$R(t_{i_v+1} t_{i_v})$	0	0	0	0	1

Tabelle 4.4: CPT der Komponente $F1_Testlager_{i+1}$ für alle Zeitscheiben $t_i > t_E = 1000$ h.

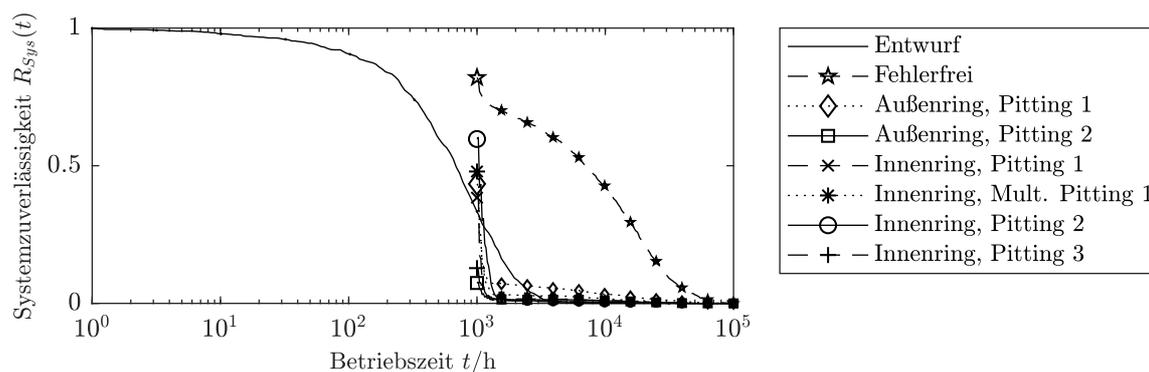


Abbildung 4.18: Systemzuverlässigkeit auf Basis des Entwurfsmodells und des Digitalen Zwillings für unterschiedliche Schädigungszustände des Testlagers (vergleiche Tabelle 4.1).

nenen und Teilsysteme weist das Testlager die geringste Zuverlässigkeit auf. Die CPT des Ausfallzustands $Testlager_i/i+1$ in Tabelle 4.3 zeigt eine starke Kopplung zwischen den Schädigungszuständen innerhalb der Fehlerursache $F1_Testlager_i/i+1$ und dem Ausfall der Komponente. Zudem weist der Wälzlagerprüfstand keine Redundanz auf, sodass der Ausfall einer Komponente oder eines Teilsystems direkt zu einem Systemausfall führt. Ein als fehlerfrei diagnostiziertes Testlager führt zu einem starken Anstieg der Systemzuverlässigkeit, die mittels der virtuellen lokalen Zuverlässigkeit bis zum Systemausfall prognostiziert wird. Die diagnostizierten Schädigungszustände *Außenring, Pitting 2* und *Innenring, Pitting 3* führen nahezu direkt zu einem Systemausfall, da diese zwei Zustände direkt zu einer Aktivierung der Fehlerursache führen und damit einen Ausfall der Komponente bewirken. Die entsprechenden Zugehörigkeitswahrscheinlichkeiten (Tabelle 4.1) weisen keine deterministische Zuordnung auf, sodass in der Folge der Evidenz die Systemzuverlässigkeit nicht direkt auf null sinkt. Im Vergleich dazu führen die Schädigungszustände *Außenring, Pitting 1*, *Innenring, Pitting 1*, *Innenring, Mult. Pitting 1* und *Innenring, Pitting 2* zunächst zu einer Steigerung der Systemzuverlässigkeit. Diese sinkt aufgrund der virtuellen lokalen Zuverlässigkeit und der starken Kopplung eines Ausfalls an die Schädigungszustände rasch ab.

Die gezeigte Integration von Zustandsüberwachungsdaten in ein Systemzuverlässigkeitsmodell ist für beliebige Zeitpunkte t_E innerhalb der Betriebsdauer des Systems anwendbar und ermöglicht eine Prädiktion der Systemzuverlässigkeit auf Basis aktueller Zustandsüberwachungsdaten.

4.3.4 Erweiterung des modularen Wälzlagerprüfstands

Der Wälzlagerprüfstand wird um ein Verzahnungsmodul erweitert, um das Schwingungsverhalten einer Verzahnung als Störungseinfluss auf die Zustandsüberwachung abzubilden. Es wird angenommen, dass der Prüfstand nach 1000 h Betriebsdauer erweitert wird. Die Erweiterung des Prüfstands wird modellbasiert abgesichert, um sicherzustellen, dass der Prüfstand auch nach der Umrüstung die geforderten Anforderungen erfüllt. Die Betriebsbedingungen werden wieder über den Betriebszeitraum konstant gehalten, zusätzlich wird ein Lastmoment $M_{Last} = 0,7 \text{ Nm}$ aufgeprägt, um die Verzahnung zu verspannen. Das erstellte Mehrkörpermodell ist in Abbildung 4.19 dargestellt.

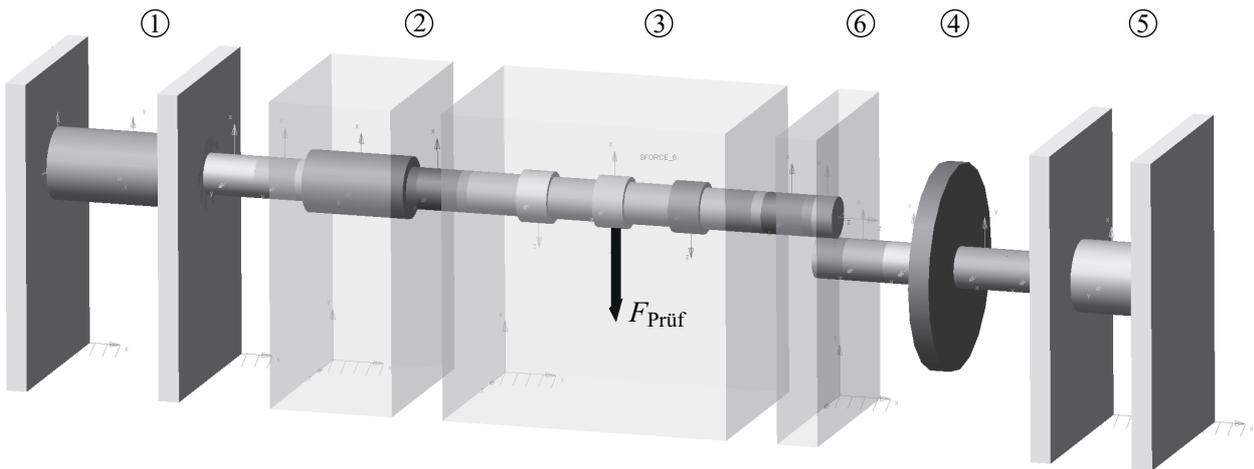


Abbildung 4.19: Mehrkörpermodell des um ein Verzahnungsmodul erweiterten Wälzlagerprüfstands. Die Positionennummern entsprechen Abbildung 4.20.

Nach der modellbasierten Absicherung der dynamischen Eigenschaften wird der reale Prüfstand umgebaut. Der resultierende Aufbau ist in Abbildung 4.20 dargestellt.

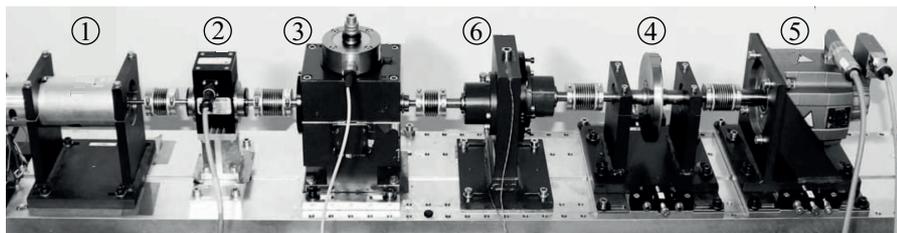


Abbildung 4.20: Wälzlagerprüfstand erweitert um ein Verzahnungsmodul Modul (6) (nach [LERB+14]).

An das Zahnradmodul werden für die vier enthaltenen Wälzlager jeweils lokale Zuverlässigkeitsschätzer annotiert. Die eingebauten Wälzlager werden als neu betrachtet, sodass sie jeweils initial eine Zuverlässigkeit von eins aufweisen. Das Testlager wird im Zuge des Umbaus gegen ein fehlerfreies getauscht, um zunächst den Einfluss der Verzahnung auf den Prüfstand abzubilden und zu erfassen. Die übrigen lokalen Zuverlässigkeitsschätzer aus Abschnitt 4.3.2 werden für $t = 1000$ h weiter berechnet. Der Ersatz des Testlagers kann als harte Evidenz interpretiert werden, da der Zustand des Wälzlagers mit Bestimmtheit bekannt ist.

In Abbildung 4.21 ist die Systemzuverlässigkeit auf Basis des aus dem Entwurf bekannten Verlaufs und einer Erweiterung des Prüfstands um ein Verzahnungsmodul und eines Austauschs des Testlagers bei $t = 1000$ h dargestellt. Die resultierende Systemzuverlässigkeit ist im Vergleich zu einer weichen Evidenz auf Basis einer Zustandsüberwachung (vergleiche Abbildung 4.18) für ein Wälzlager im Zustand *Fehlerfrei* höher, da keine Unsicherheiten, wie bei der Zustandsüberwachung, berücksichtigt werden müssen.

Auf Basis der integrierten Modellierung von Zuverlässigkeit und dynamischem Systemverhalten ist eine effiziente Anpassung des Digitalen Zwillings an Änderungen am realen System umsetzbar, sofern Änderungsmaßnahmen eine modellbasierte Absicherung

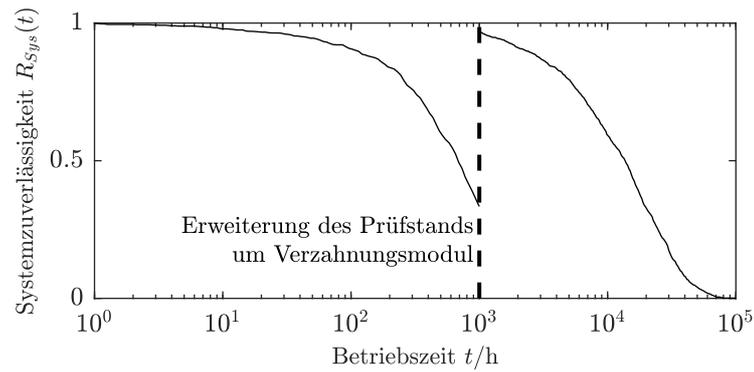


Abbildung 4.21: Resultierende Systemzuverlässigkeit nach Erweiterung des Wälzlagerprüfstands um ein Verzahnungsmodul.

zugrundeliegt. In Kombination mit der Integration der Methode in bestehende PLM-Systeme und über standardisierte Schnittstellen zu Zustandsüberwachungsdaten, ist die Umsetzung und Wartung eines Digitalen Zwillings für die Instandhaltung zielgerichtet zu realisieren.

5 Zusammenfassung und Ausblick

Die digitale Transformation führt zu der Verbreitung intelligenter technischer Systeme, die, neben einer durch inhärente Intelligenz einhergehende Ausweitung der Funktionalität, durch eine starke Steigerung der Systemkomplexität charakterisiert sind. Die zunehmende Komplexität stellt eine große Herausforderung für die Absicherung der Verlässlichkeit dieser Systeme dar. Selbstoptimierende technische Systeme wurden als Stellvertreter der Klasse der intelligenten technischen Systeme vorgestellt und Maßnahmen für die Steigerung der Verlässlichkeit anhand dieser Systeme dargelegt. Dabei wurde die Verlässlichkeit hinsichtlich ihrer Beeinträchtigungen, Kenngrößen und Mittel definiert, um den zunehmenden Anteil der Informationsverarbeitung an der Systemfunktionalität in intelligenten Systemen zu berücksichtigen. Die Zuverlässigkeit wurde als zentrale Kenngröße der Verlässlichkeit identifiziert, da weitere Kenngrößen, wie die Sicherheit und Verfügbarkeit, eine hohe Abhängigkeit aufweisen. Daher wurden verstärkt Maßnahmen für die Steigerung der Verlässlichkeit hinsichtlich der Zuverlässigkeit betrachtet. Die Umsetzung wirksamer Maßnahmen zur Steigerung der Verlässlichkeit in intelligenten technischen Systemen basiert auf einem modellbasierten Entwicklungsprozess. Dieser wurde anhand des Entwurfs mechatronischer Systeme nachvollzogen, da diese nach wie vor die Basis intelligenter technischer Systeme bilden. Es wurde gezeigt, dass für die Umsetzung von Maßnahmen zur Steigerung der Verlässlichkeit eine geschlossene Modellierung der Zuverlässigkeit auf Systemebene und des dynamischen Systemverhaltens erforderlich ist. Diese Maßnahmen sind inhärenter Bestandteil der Betriebsstrategie mechatronischer Systeme mit dem expliziten Teilziel der Steigerung der Zuverlässigkeit und lassen sich mithilfe des Paradigmas der Selbstoptimierung umsetzen.

Eine Analyse und Bewertung des Stands der Technik zu geschlossenen Modellierungsansätzen für die Abbildung von Zuverlässigkeit und dynamischem Systemverhalten zeigte eine substantielle Lücke auf, die eine Anwendung im Entwicklungsprozess selbstoptimierender technischer Systeme nicht zulässt. Der Entwurf von selbstoptimierenden technischen Systemen nutzt typischerweise Methoden der Mehrzieloptimierung, um optimale Kompromisse zwischen konkurrierenden Entwurfszielen zu finden. Die gefundenen Betriebspunkte werden vom System autonom auf Basis von Beobachtungen der Umwelt- und Betriebsbedingungen, Nutzeranforderungen sowie Änderungen des Systems selbst ausgewählt, um das Systemverhalten zielgerichtet zu beeinflussen. Aus der Bewertung des Stands der Technik und den Anforderungen aus der Entwicklung selbstoptimierender Systeme wurde herausgestellt, dass eine allgemeingültige Methode zur Formulierung zuverlässigkeitsorientierter Ziele, welche die Anforderungen erfüllt, nicht vorhanden ist. Auf Basis der Anforderungen an eine solche Methode wurde die Zielsetzung dieser Arbeit konkretisiert. Demnach soll ein Modellierungsansatz für die geschlossene Abbildung der Zuverlässigkeit auf Systemebene und des dynamischen Systemverhaltens entwickelt werden, der den allgemeinen Anforderungen entspricht: Abbildung des prinzipiellen Zusammenhangs zwischen Zuverlässigkeit und Dynamik sowie Verwendung von bestehenden Modellen des Entwicklungsprozesses auf Basis unterschiedlicher Modellierungsumge-

bungen. Hinsichtlich der Abbildung des Ausfallverhaltens innerhalb des Systemzuverlässigkeitsmodells ist eine hohe Modellierungstiefe gefordert. Die Ableitung des Systemzuverlässigkeitsmodells soll ohne umfangreiche Nutzereingaben auskommen und innerhalb einer Optimierung eingesetzt werden können. Die integrierte Modellierung von Zuverlässigkeit und dynamischem Verhalten wurde im Hinblick auf diese Zielsetzung im Rahmen dieser Arbeit entwickelt und anhand von drei Anwendungsbeispielen validiert. Die Unterstützung des Entwurfs von Maßnahmen zur Steigerung der Verlässlichkeit selbstoptimierender Systeme wurde durch die Berechnung von Betriebspunkten auf Basis der integrierten Modellierung für eine zuverlässigkeitsadaptive Regelung gezeigt. Nach einer initialen Auswahl und Teilparametrierung der lokalen Zuverlässigkeitsschätzer wurde ein Systemzuverlässigkeitsmodell innerhalb einer Mehrzieloptimierung verwendet. Die gefundenen Betriebspunkte wurden erfolgreich in der zuverlässigkeitsadaptiven Regelung eingesetzt. Die Komplexität des Systemmodells wurde im zweiten Anwendungsbeispiel deutlich erhöht. Es wurde eine Systemzuverlässigkeitsanalyse für eine Windenergieanlage durchgeführt, wobei das zugrundeliegende Systemmodell und die verwendeten lokalen Zuverlässigkeitsschätzer eine höhere Modellierungstiefe aufweisen. Darüber hinaus wurde gezeigt, dass die Systemzuverlässigkeitsanalyse durch die Integration weiterer Modellierungsaspekte, wie externer Ereignisse und Ausfälle gemeinsamer Ursache, umfassend zu erweitern ist. Das dritte Anwendungsbeispiel zeigt die weiterführenden Potenziale der integrierten Modellierung anhand der Umsetzung eines Digitalen Zwillings für die Instandhaltung auf. Die integrierte Modellierung ermöglicht die durchgängige Verwendung von Modellen der Entwicklung in späten Lebenszyklusphasen und die Kombination mit datenbasierten Verfahren, wie der Zustandsüberwachung, für eine wirkungsvolle Unterstützung der Instandhaltung.

Im Rahmen einer kritischen Diskussion der entwickelten Methode hinsichtlich der Anforderungen an eine integrierte Modellierung der Zuverlässigkeit und des dynamischen Systemverhaltens sind, obgleich der erfolgreichen Anwendungen, Einschränkungen bei der Abbildung der Zuverlässigkeit gegeben. Die Abbildung von Redundanzen mit Lastteilung ist aktuell noch nicht umgesetzt. Zu diesen redundanten Systemen können auch selbstoptimierende Systeme mit Rekonfigurationspotenzial gezählt werden. Die durch Ausnutzung der Rekonfiguration hervorgerufene Änderung der Topologie des betrachteten Systems ist als Redundanz mit Lastteilung abbildbar. Die Abbildung des Ausfallverhaltens solcher redundanter Systeme erfordert, neben der Modellierung des Einflusses des dynamischen Systemverhaltens auf die Zuverlässigkeit, zusätzlich die Modellierung einer Abhängigkeit des dynamischen Systemverhaltens vom Ausfallverhalten. Die Modellierung dieser Rückführung aus dem Zuverlässigkeitsmodell auf das Modell des dynamischen Systemverhaltens macht eine Anpassung des zugrundeliegenden Systemmodells notwendig. Eine solche Anpassung widerspricht dem für diese Arbeit aufgestellten Modellierungsansatz, dass bestehende Modelle des Entwurfs mechatronischer Systeme durch die integrierte Modellierung nicht verändert werden. Eine Möglichkeit zur Umsetzung der Rückführung des Ausfallverhaltens auf das dynamische Systemverhalten könnten zukünftig einheitliche Modellstandards und -schnittstellen darstellen.

In der automatisierten Transformation des Zuverlässigkeitsmodells können nur Abhängigkeiten zwischen Komponenten und Teilsystemen berücksichtigt werden, die in den Modellen des dynamischen Systemverhaltens, beispielsweise in Form von Signalpfaden, bereits enthalten sind. Dies stellt eine Einschränkung in der Modellierungstiefe dar, die

aktuell nur durch zusätzliche Nutzereingaben kompensiert werden kann. Die Einbeziehung von Nutzereingaben für die Abbildung zusätzlicher Modellierungsaspekte im Ausfallverhalten des zu analysierenden Systems können durch die Nutzung weiterer, aus dem Entwicklungsprozess zur Verfügung stehender, Modelle des Umfelds und der Anwendungsszenarien weiter reduziert werden. Hier kann die Verknüpfung mit Methoden zur automatisierten Erzeugung von FMEAs, wie sie in Abschnitt 2.6 vorgestellt wurden, zielgerichtet eingesetzt werden, um beispielweise mögliche Fehlerursachen wiederkehrender Komponenten und Teilsysteme in Datenbanken verfügbar zu machen und Nutzereingaben weiter zu reduzieren.

Im Rahmen des Entwicklungsprozesses sind die Korrelation und Synchronisation unterschiedlichster Modelle für die gesamtheitliche Abbildung eines Systems notwendig. Die automatisierte Synthese von zusätzlichen Modellen bietet eine Möglichkeit zur Beherrschung der in Zukunft weiter steigenden Systemkomplexität. Die integrierte Modellierung von Zuverlässigkeit und dynamischem Systemverhalten stellt hier einen Beitrag zur Unterstützung des Entwicklungsprozesses intelligenter technischer Systeme dar.

Es wurde gezeigt, dass sich die integrierte Modellierung für die Umsetzung eines Digitalen Zwillinges eignet. Ein solcher Digitaler Zwilling kann für eine Anpassung der Betriebspunkte parallel zum Betrieb genutzt werden. Die Annahmen, die im Entwurf der Betriebsstrategie für die Bestimmung der Betriebspunkte getroffen wurden, gelten möglicherweise nicht über die gesamte Betriebsdauer des Systems. Veränderliche Betriebs- und Umweltbedingungen sowie Degradation des Systems selbst werden im Digitalen Zwilling berücksichtigt und bilden das System zum aktuellen Zeitpunkt ab. Auf Basis des Digitalen Zwillinges ist eine Berechnung der Betriebspunkte mithilfe von Mehrzieloptimierungsverfahren möglich, sodass das System stets optimal eingestellt wird.

Literaturverzeichnis

- [ANS19] ANSYS medini Technologies AG. *ANSYS medini analyze: Quality, Safety and Reliability Engineering*. 2019. URL: <https://www.ansys.com/products/systems/ansys-medini-analyze> (besucht am 18.06.2019).
- [ANS17] ANSYS, I. „Spotlight on the Digital Twin“. In: *ANSYS Advantage* 11.1 (2017).
- [ANS19] ANSYS, I. *Simplorer*. 2019. URL: <https://www.ansys.com/-/media/ansys/corporate/resourcelibrary/brochure/ansys-simplorer-brochure-140.pdf> (besucht am 12.10.2019).
- [AVL19] AVL. *AVL Model.CONNECT*. 2019. URL: <https://www.avl.com/-/model-connect-> (besucht am 12.10.2019).
- [ADG+09] Adelt, P., Donoth, J., Gausemeier, J., Geisler, J., Henkler, S., Kahl, S., Klöpper, B., Krupp, A., Münch, E., Oberthür, S., Paiz, C., Porrmann, M., Radkowski, R., Romaus, C., Schmidt, A., Schulz, B., Voeking, H., Witkowski, U., Witting, K. und Znamenshchikov, A. *Selbstoptimierende Systeme des Maschinenbaus: Definitionen, Anwendungen, Konzepte*. Bd. Band 234. HNI-Verlagsschriftenreihe. 2009.
- [ALR+04] Avižienis, A., Laprie, J.-C., Randell, B. und Landwehr, C. „Basic Concepts and Taxonomy of Dependable and Secure Computing“. In: *IEEE transactions on dependable and secure computing* 1.1 (2004), S. 11–33.
- [BZB+13] Bak, C., Zahle, F., Bitsche, R., Kim, T., Yde, A., Henriksen, L. C., Hansen, M. H., Blasques, Jose Pedro Albergaria Amaral, Gaunaa, M. und Natarajan, A. „The DTU 10-MW reference wind turbine“. In: *Danish Wind Power Research 2013*. 2013.
- [BB15] Bathelt, L. und (Betreuer) Kaul, T. *Modellierung der Verlässlichkeit komplexer redundanter Systeme: unveröffentlichte Bachelorarbeit*. Universität Paderborn, 2015.
- [Bea78] Beaudry, M. D. „Performance-related reliability measures for computing systems“. In: *IEEE Transactions on Computers* 6 (1978), S. 540–547.
- [Ber08] Bertsche, B. *Reliability in automotive and mechanical engineering: determination of component and system reliability*. Springer Science & Business Media, 2008.
- [BML07] Bestory, C., Marc, F. und Levi, H. „Statistical analysis during the reliability simulation“. In: *Microelectronics Reliability* 47.9 (2007), S. 1353–1357.
- [Bir07] Birolini, A. *Reliability Engineering: Theory and Practice*. 5. Aufl. Springer, 2007.

- [Bli18] Blitz-Informationdienst von Siemens. *Siemens Blitzatlas 2017*. Hrsg. von Siemens AG. Nürnberg, 2018. URL: <https://www.siemens.com/press/de/feature/2015/corporate/2015-08-blids.php#event-toc-1> (besucht am 22. 08. 2019).
- [BPM+01] Bobbio, A., Portinale, L., Minichino, M. und Ciancamerla, E. „Improving the analysis of dependable systems by mapping fault trees into Bayesian networks“. In: *Reliability Engineering & System Safety* 71.3 (2001), S. 249–260.
- [BR16] Boschert, S. und Rosen, R. „Digital Twin - The Simulation Aspect“. In: *Mechatronic Futures*. Hrsg. von Hehenberger, P. und Bradly, D. Springer, Cham, 2016, S. 59–74.
- [BD05] Boudali, H. und Dugan, J. B. „A discrete-time Bayesian network reliability modeling and analysis framework“. In: *Reliability Engineering & System Safety* 87.3 (2005), S. 337–349.
- [BV03] Bozzano, M. und Villafiorita, A. „Improving System Reliability via Model Checking: The FSAP/NuSMV-SA Safety Analysis Platform“. In: *International Conference on Computer Safety, Reliability, and Security*. 2003, S. 49–62.
- [Bra92] Brancato, E. L. „Estimation of Lifetime Expectancies of Motors“. In: *IEEE Electrical Insulation Magazine* 8.3 (1992), S. 5–13.
- [BPSM+97] Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E. und Yergeau, F. „Extensible Markup Language (XML)“. In: *World Wide Web Journal* 2.4 (1997), S. 27–66.
- [Cai19] Caithness Windfarm Information Forum. *Wind Turbine Accident and Incident Compilation*. Hrsg. von Caithness Windfarm Information Forum. Caithness, 2019.
- [Cis14] Cisk, M. „Muster und das Generieren von Ideen für Geschäftsmodellinnovationen“. Dissertation. Universität St. Gallen, 2014. (Besucht am 20. 04. 2017).
- [DIN06] DIN, E. N. „60812: Analysetechniken für die Funktionsfähigkeit von Systemen–Verfahren für die Fehlzustandsart- und -auswirkungsanalyse (FMEA)“. In: *DIN, Germany* (2006).
- [Das19] Dassault Systems. *DYMOLA Systems Engineering*. 2019. URL: <https://www.3ds.com/products-services/catia/products/dymola/> (besucht am 12. 10. 2019).
- [Dat19] Datadvance. *Datadvance pSeven*. 2019. URL: <https://www.datadvance.net/product/pseven/> (besucht am 12. 10. 2019).
- [DIK10] David, P., Idasiak, V. und Kratz, F. „Reliability study of complex physical systems using SysML“. In: *Reliability Engineering & System Safety* 95.4 (2010), S. 431–450.

-
- [DHK+09] Dell'Aere, A., Hirsch, M., Klöpper, B., Koester, M., Krupp, A., Krüger, M., Müller, T., Oberthür, S., Pook, S., Priesterjahn, C., Romaus, C., Schmidt, A., Sondermann-Wölke, C., Tichy, M., Voeking, H. und Zimmer, D. *Verlässlichkeit selbstoptimierender Systeme: Potenziale nutzen und Risiken vermeiden*. Bd. 235. HNI-Verlagsschriftenreihe. Heinz-Nixdorf-Institut, 2009.
- [Deu11] Deutsches Institut für Normung. *Sicherheit von Maschinen – Allgemeine Gestaltungsleitsätze – Risikobeurteilung und Risikominderung*. Berlin, 2011.
- [DI10] Deutsches Institut für Normung und International Organization for Standardization. *Wälzlager – Dynamische Tragzahlen und nominelle Lebensdauer*. Berlin, 2010-10-01.
- [Die17] Diestel, R. *Graphentheorie*. 5. Aufl. Springer Spektrum, 2017.
- [Die93] Diez, F. J. „Parameter adjustment in Bayes networks. The generalized noisy OR-gate“. In: *Uncertainty in Artificial Intelligence*. Elsevier, 1993, S. 99–105.
- [Dor15] Dorociak, R. K. *Systematik zur frühzeitigen Absicherung der Sicherheit und Zuverlässigkeit fortschrittlicher mechatronischer Systeme: Dissertation*. Bd. 340. Verlagsschriftenreihe des Heinz Nixdorf Instituts. Paderborn, 2015.
- [DHR07] Durstewitz, M., Hahn, B. und Rohrig, K. *Wissenschaftliches Mess- und Evaluierungsprogramm zur Fördermaßnahme "250 MW-Wind" (WEMP): Abschlussbericht zum Forschungsvorhaben 03W0001J*. Hrsg. von ISET e.V. Kassel, 2007.
- [Ehr05] Ehrgott, M. *Multicriteria Optimization*. Bd. 491. Springer Science & Business Media, 2005.
- [Fin59] Findley, W. N. „A Theory for the Effect of Mean Stress on Fatigue of Metals Under Combined Torsion and Axial Load or Bending“. In: *Journal of Engineering for Industry* 81.4 (1959), S. 301–305.
- [FGM+07] Frank, U., Giese, H., Müller, T., Oberthür, S., Romaus, C., Tichy, M. und Voeking, H. „Potenziale und Risiken der Selbstoptimierung für die Verlässlichkeit mechatronischer Systeme“. In: *5. Paderborner Workshop Entwurf mechatronischer Systeme*. Bd. 210. Verlagsschriftenreihe des Heinz Nixdorf Instituts, Paderborn, 2007.
- [Fra10] Fraracci, A. *Model-based Failure-modes-and-effects analysis and its application to Aircraft Subsystems*. 2010.
- [GHJ+05] Gamma, E., Helm, R., Johnson, R. und Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Boston [u.a.], 2005.
- [GFC13] Gassmann, O., Frankenberger, K. und Csik, M. *Geschäftsmodelle entwickeln: 55 innovative Konzepte mit dem St. Galler Business Model Navigator*. München : Hanser, 2013.
- [GFD+08] Gausemeier, J., Frank, U., Donoth, J. und Kahl, S. „Spezifikationstechnik zur Beschreibung der Prinziplösung selbstoptimierender Systeme des Maschinenbaus“. In: *Konstruktion, Ausgabe 7/8-2008* (2008).

- [GFD+09] Gausemeier, J., Frank, U., Donoth, J. und Kahl, S. „Specification Technique for the Description of Self-Optimizing Mechatronic Systems“. In: *Research in Engineering Design* 20.4 (2009), S. 201–223.
- [GRS14] Gausemeier, J., Rammig, F.-J. und Schäfer, W., Hrsg. *Design Methodology for Intelligent Technical Systems: Develop Intelligent Technical Systems of the Future*. Springer-Verlag, Heidelberg, Germany, 2014.
- [GRS+14] Gausemeier, J., Rammig, F. J., Schäfer, W., Sextro, W. und et.al. *Dependability of Self-Optimizing Mechatronic Systems*. Springer, 2014.
- [GB16] Göllner, D. und (Betreuer) Kaul, T. *Gemeinsame Abbildung verschiedener Fehlerarten in Bayes’schen Netzen: unveröffentlichte Studienarbeit*. Universität Paderborn, 2016.
- [GV17] Grieves, M. und Vickers, J. „Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems“. In: *Transdisciplinary perspectives on complex systems*. Springer, 2017, S. 85–113.
- [Grü18] Grübel, A. „Effiziente bruchmechanische Herangehensweisen für eine wirtschaftliche Produktentstehung und einen sicheren Bauteilbetrieb“. Dissertation. Paderborn: Universität Paderborn, 2018.
- [GJ08] Grune, D. und Jacobs, C. J. H. *Parsing Techniques: A Practical Guide*. 2. Aufl. New York: Springer-Verlag, 2008.
- [HB16a] Hagedorn, O. E. C. und (Betreuer) Kaul, T. *Analyse der Verfügbarkeit mechatronischer Systeme auf Basis von Bayes’schen Netzen als Modelle der Systemzuverlässigkeit: unveröffentlichte Bachelorarbeit*. Universität Paderborn, 2016.
- [HCR+91] Halbwachs, N., Caspi, P., Raymond, P. und Pilaud, D. „The Synchronous Data Flow Programming Language LUSTRE“. In: *Proceedings of the IEEE* 79.9 (1991), S. 1305–1320.
- [HRB09] Harris, T., Rumbarger, J. H. und Butterfield, C. P. *Wind Turbine Design Guideline DG03: Yaw and Pitch Rolling Bearing Life*. 2009.
- [HB16b] Hasbal, G. und (Betreuer) Kaul, T. *Belastungsabhängige Lebensdauerprognose und Zustandsüberwachung von Maschinen, -komponenten und Anlagen: unveröffentlichte Bachelorarbeit*. Universität Paderborn, 2016.
- [Hay12] Hayman, G. J. *MLife Theory Manual for Version 1.00*. Hrsg. von National Renewable Energy Laboratory (NREL), Golden, CO. 2012.
- [HB12] Hayman, G. J. und Buhl JR., M. *MLife User’s Guide for Version 1.00*. Hrsg. von National Renewable Energy Laboratory (NREL), Golden, CO. 2012.
- [HM08] Hecht, H. und Menes, R. „Software FMEA Automated and as a Design Tool“. In: *Proceedings of the Society of Automotive Engineers* (2008).
- [Hen89] Henrion, M. „Some Practical Issues in Constructing Belief Networks“. In: *Uncertainty in Artificial Intelligence*. 1989, S. 161–173.
- [HKG+17] Hentze, J., Kaul, T., Gräßler, I. und Sextro, W. „Integrated Modeling of Behavior and Reliability in System Development“. In: *DS 87-4 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 4: Design Methods and Tools*. 2017.

-
- [Hig05] Higham, N. J. „The Scaling and Squaring Method for the Matrix Exponential Revisited“. In: *SIAM Journal on Matrix Analysis and Applications* 26.4 (2005), S. 1179–1193.
- [HX10] Huang, L. und Xu, Q. „Lifetime reliability for load-sharing redundant systems with arbitrary failure distributions“. In: *IEEE Transactions on Reliability* 59.2 (2010), S. 319–330.
- [Int13] International Electrotechnical Commission. *Protection Against Lightning*. 2013.
- [Int11] International Organization for Standardization. *Road Vehicles - Functional Safety*. Berlin, 2011.
- [Ise02] Isermann, R. „Fehlertolerante Komponenten für Drive-by-Wire-Systeme“. In: *ATZ - Automobiltechnische Zeitschrift* 104.4 (2002), S. 382–391. URL: <http://dx.doi.org/10.1007/BF03224405>.
- [IKD+13] Iwanek, P., Kaiser, L., Dumitrescu, R. und Nyßen, A. „Fachdisziplinübergreifende Systemmodellierung mechatronischer Systeme mit SysML und CONSENS: Modellbasierte Systementwicklung 2“. In: *Tag des Systems Engineerings*. Hrsg. von Maurer, M. und Schulze, S.-O. Tag des Systems Engineering. Carl Hanser Verlag, 2013, S. 337–346.
- [Jef83] Jeffrey, R. C. *The Logic of Decision*. 2. Chicago: University of Chicago Press, 1983.
- [JN09] Jensen, F. V. und Nielsen, T. D. *Bayesian Networks and Decision Graphs*. Springer, 2009.
- [Jon12] Jones, H. „Common cause failures and ultra reliability“. In: *42nd International Conference on Environmental Systems*. 2012.
- [JK12] Jonkman, B. J. und Kilcher, L. *TurbSim User’s Guide*. 2012. (Besucht am 23. 11. 2017).
- [JBM+09] Jonkman, J., Butterfield, S., Musial, W. und Scott, G. *Definition of a 5-MW reference wind turbine for offshore system development*. 2009.
- [Jos08] Joshi, A. „Behavioral fault modeling and model composition for model-based safety analysis“. Dissertation. University of Minnesota, 2008.
- [JVB07] Joshi, A., Vestal, S. und Binns, P. „Automatic Generation of Static Fault Trees from AADL Models“. In: *DSN Workshop on Architecting Dependable Systems*. 2007.
- [KP14] Kapur, K. C. und Pecht, M. *Reliability Engineering*. New Jersey: John Wiley & Sons, 2014.
- [KHS+19] Kaul, T., Hentze, J., Sextro, W. und Gräßler, I. „Integration von Verlässlichkeitsmodellen aus der Entwicklung in einen Digitalen Zwilling zur Umsetzung einer vorausschauenden Instandhaltung“. In: *Tagungsband VDI-Mechatronik*. Hrsg. von Bertram, T., Corvers, B., Gräßler, I. und Janschek, K. 2019, S. 19–24.
- [KMS15a] Kaul, T., Meyer, T. und Sextro, W. „Integrated Model for Dynamics and Reliability of Intelligent Mechatronic Systems“. In: *European Safety and Reliability Conference (ESREL2015)*. Hrsg. von Podofillini et al. London: Taylor and Francis, 2015.

- [KMS15b] Kaul, T., Meyer, T. und Sextro, W. „Integrierte Modellierung der Dynamik und der Verlässlichkeit komplexer mechatronischer Systeme“. In: *10. Paderborner Workshop Entwurf mechatronischer*. Hrsg. von Gausemeier, J., Dumitrescu, R., Rammig, F. J., Schäfer, W. und Trächtler, A. HNI-Verlagsschriftenreihe. Paderborn: Heinz Nixdorf Institut, Universität Paderborn, 2015.
- [KMS16] Kaul, T., Meyer, T. und Sextro, W. „Modeling of Complex Redundancy in Technical Systems with Bayesian Networks“. In: *Proceedings of the Third European Conference of the Prognostics and Health Management Society 2016*. 2016.
- [KMS17] Kaul, T., Meyer, T. und Sextro, W. „Formulation of reliability-related objective functions for design of intelligent mechatronic systems“. In: *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 231.4 (2017), S. 390–399.
- [KBS19] Kaul, T., Bender, A. und Sextro, W. „Digital Twin for Reliability Analysis During Design and Operation of Mechatronic Systems“. In: *Proceedings of the 29th European Safety and Reliability Conference*. Hrsg. von Beer, M. und Zio, E. Singapore: Research Publishing, 2019.
- [KSM12] Kern, A., Schelthoff, C. und Mathieu, M. „Probability of Lightning Strikes to Air-Terminations of Structures Using the Electro-Geometrical Model theory and the statistics of lightning current parameters“. In: *Atmospheric research* 117 (2012), S. 2–11.
- [Kim16] Kimotho, J. K. *Development and Performance Evaluation of Prognostic Approaches for Technical Systems: Dissertation*. Bd. 4. Schriften des Lehrstuhls für Dynamik und Mechatronik, Hrsg. vom Prof. Dr.-Ing. habil. W. Sextro. Aachen: Shaker, 2016.
- [KHV09] Kothamasu, R., Huang, S. H. und VerDuin, W. H. „System health monitoring and prognostics—a review of current paradigms and practices“. In: *Handbook of Maintenance Management and Engineering*. Springer, 2009, S. 337–362.
- [KRK+13] Krüger, M., Ramirez, A., Keßler, J. H. und Trächtler, A. „Discrete objective-based control for self-optimizing systems“. In: *2013 American Control Conference*. 2013, S. 3403–3408.
- [Kuh17] Kuhn, T. „Digitaler Zwilling“. In: *Informatik-Spektrum* 40.5 (2017), S. 440–444.
- [KZ03] Kuo, W. und Zuo, M. J. *Optimal Reliability Modeling: Principles and Applications*. Hoboken, New Jersey: John Wiley & Sons, 2003.
- [LP07] Langseth, H. und Portinale, L. „Bayesian networks in reliability“. In: *Reliability Engineering & System Safety* 92.1 (2007), S. 92–108.
- [LT17] Leimeister, M. und Thomas, P. „The OneWind Modelica Library for Floating Offshore Wind Turbine Simulations with Flexible Structures“. In: *Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, May 15-17, 2017*. 2017, S. 633–642.

-
- [Les17] Lessmeier, C. „Datenbasierte Zustandsüberwachung von Wälzlagerschäden in elektromechanischen Antriebssystemen“. Dissertation. Paderborn: Universität Paderborn, 2017.
- [LERB+14] Lessmeier, C., Enge-Rosenblatt, O., Bayer, C. und Zimmer, D. „Data Acquisition and Signal Analysis from Measured Motor Currents for Defect Detection in Electromechanical Drive Systems“. In: *European Conference of the Prognostics and Health Management Society 2014*. 2014.
- [LKZ+16] Lessmeier, C., Kimotho, J. K., Zimmer, D. und Sextro, W. „Condition Monitoring of Bearing Damage in Electromechanical Drive Systems by Using Motor Current Signals of Electric Motors: A Benchmark Data Set for Data-Driven Classification“. In: *Proceedings of the European Conference of the Prognostics and Health Management Society*. 2016, S. 05–08.
- [LCS91] Leveson, N. G., Cha, S. S. und Shimeall, T. J. „Safety Verification of Ada Programs Using Software Fault Trees“. In: *IEEE software* 8.4 (1991), S. 48–59.
- [LHW+17] Li, H., Hu, Z., Wang, J. und Meng, X. „Short-term fatigue analysis for tower base of a spar-type wind turbine under stochastic wind-wave loads“. In: *International Journal of Naval Architecture and Ocean Engineering* (2017).
- [LYM16] Li, K., Yi, R. und Ma, Z. „Reliability analysis of dynamic reliability blocks through conversion into dynamic bayesian networks“. In: *Industrial Engineering and Engineering Management (IEEM), 2016 IEEE International Conference on*. 2016, S. 1330–1334.
- [LR98] Liggesmeyer, P. und Rothfelder, M. „Improving System Reliability with Automatic Fault Tree Generation“. In: *Proceedings of the 28th Annual International Symposium on Fault-Tolerant Computing*. 1998, S. 90–99.
- [LZB15] Long, F., Zeiler, P. und Bertsche, B. „Potentials of coloured petri nets for realistic availability modelling of production systems in Industry 4.0“. In: *European Safety and Reliability Conference (ESREL2015)*. Hrsg. von Podofillini et al. Bd. 7. London: Taylor and Francis, 2015.
- [LZB17] Long, F., Zeiler, P. und Bertsche, B. „Modelling the flexibility of production systems in Industry 4.0 for analysing their productivity and availability with high-level Petri nets“. In: *IFAC-PapersOnLine* 50.1 (2017), S. 5680–5687.
- [Mey16] Meyer, T. „Optimization-Based Reliability Control of Mechatronic Systems“. Dissertation. Paderborn: Universität Paderborn, 2016.
- [MKS15] Meyer, T., Kaul, T. und Sextro, W. „Advantages of reliability-adaptive system operation for maintenance planning“. In: *Proceedings of the 9th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes*. 2015, S. 940–945.
- [MP10] Meyna, A. und Pauli, B. *Taschenbuch der Zuverlässigkeitstechnik: Quantitative Bewertungsverfahren*. München [u.a.], 2010.
- [Mur02] Murphy, K. P. „Dynamic Bayesian Networks: Representation, Inference and Learning“. Dissertation. University of California, Berkeley, 2002.

- [Mur+01] Murphy, K. P. u. a. „The Bayes Net Toolbox for Matlab“. In: *Computing science and statistics* 33.2 (2001), S. 1024–1034.
- [NFB18] Nathan, E., Fairbanks, J. und Bader, D. „Ranking in Dynamic Graphs Using Exponential Centrality“. In: *Complex Networks & Their Applications VI*. Hrsg. von Cherifi, C., Cherifi, H., Karsai, M. und Musolesi, M. Cham: Springer International Publishing, 2018, S. 378–389.
- [O’C13] O’Connor, A. „A general cause based methodology for analysis of common cause and dependent failures in system risk and reliability assessments“. Dissertation. University of Maryland, 2013.
- [Obj19] Object Management Group. *OMG System Modeling Language v1.6*. 2019. URL: <https://www.omg.org/spec/SysML/> (besucht am 28. 08. 2019).
- [Ope19] OpenModelica. *OpenModelica*. 2019. URL: <https://openmodelica.org/> (besucht am 12. 10. 2019).
- [Pab05] Pabst, I. „On Modeling, Analysis, and Synthesis of Generalized Reliability Control Systems“. Dissertation. Universität Duisburg-Essen, 2005.
- [PBS+13] Pahl, G., Beitz, W., Schulz, H.-J. und Jarecki, U. *Konstruktionslehre: Grundlagen erfolgreicher Produktentwicklung. Methoden und Anwendung*. Springer-Verlag, 2013.
- [PM01] Papadopoulos, Y. und Maruhn, M. „Model-Based Synthesis of Fault Trees From Matlab-Simulink Models“. In: *Dependable Systems and Networks, 2001. DSN 2001. International Conference on*. 2001, S. 77–82.
- [PM99] Papadopoulos, Y. und McDerimid, J. A. „Hierarchically Performed Hazard Origin and Propagation Studies“. In: *International Conference on Computer Safety, Reliability, and Security*. 1999, S. 139–152.
- [Pea87] Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Elsevier, 1987.
- [Pho19] Phoenix Integration. *ModelCenter Integrate*. 2019. URL: <https://www.phoenix-int.com/product/modelcenter-integrate/> (besucht am 12. 10. 2019).
- [Poz06] Pozsgai, P. „Realitätsnahe Modellierung und Analyse der operativen Zuverlässigkeitskennwerte technischer Systeme“. Dissertation. Stuttgart: Universität Stuttgart, 2006.
- [Pro14] Prosvirnova, T. „AltaRica 3.0: A Model-Based Approach for Safety Analyses“. Diss. Ecole Polytechnique, 2014.
- [QB18] Qi, Y. und (Betreuer) Kaul, T. *Fehlzustandsart-, -auswirkungs- und -kritizitätsanalyse (FMECA) einer selbstoptimierenden Reibkupplung: unveröffentlichte Studienarbeit*. Universität Paderborn, 2018.
- [Rei10] Reisig, W. *Petrinetze: Modellierungstechnik, Analysemethoden, Fallstudien*. Wiesbaden, 2010.
- [RKK07] Rugina, A.-E., Kanoun, K. und Kaâniche, M. „A System Dependability Modeling Framework Using AADL and GSPNs“. In: *Architecting Dependable Systems IV*. Springer, 2007, S. 14–38.

-
- [Sag04] Sagan, S. D. „The Problem of Redundancy Problem: Why more Nuclear Security Forces May Produce less Nuclear Security“. In: *Risk Analysis* 24.4 (2004), S. 935–946.
- [SMW06] Salem, A. B., Muller, A. und Weber, P. „Dynamic Bayesian Networks in System Reliability Analysis“. In: *IFAC Proceedings Volumes* 39.13 (2006), S. 444–449. (Besucht am 20.04.2017).
- [SGE17] Sallaba, M., Gentner, A. und Esser, R. *Grenzenlos vernetzt: Smarte Digitalisierung durch IoT, Digital Twins und die Supra-Plattform*. 2017. URL: https://www2.deloitte.com/content/dam/Deloitte/de/Documents/technology-media-telecommunications/TMT_Digital_Twins_Studie_Deloitte.pdf (besucht am 22.08.2019).
- [San08] Sander, M. *Sicherheit und Betriebsfestigkeit von Maschinen und Anlagen*. Springer, 2008.
- [Sch11] Schallert, C. „Inclusion of Reliability and Safety Analysis Methods in Modelica“. In: *8th International Modelica Conference*. 2011.
- [Sch92] Schneeweiss, W. G. *Zuverlässigkeitstechnik: von den Komponenten zum System*. Datakontext-Verlag, 1992.
- [SSP+16] Schwack, F., Stammer, M., Poll, G. und Reuter, A. „Comparison of life calculations for oscillating bearings considering individual pitch control in wind turbines“. In: *Journal of Physics: Conference Series*. Bd. 753. 2016, S. 112013.
- [SRB+17] Sevin, A., Rosemeier, M., Bätge, M., Braun, R., Meng, F., Shan, M., Horte, D., Balzani, C., Reuter, A., Daniele, E., Thomas, P. und Popko, W. *IWES Wind Turbine IWT-7.5-164*. Hrsg. von Fraunhofer Institute for Wind Energy and Energy System Technology (IWES) Northwest. Bremerhaven, 2017.
- [SMK+17] Sextro, W., Meyer, T., Kaul, T. und Kimotho, J. K. „Entwicklung verlässlicher, intelligenter Systeme: Herausforderungen und Chancen“. In: *VDI-Berichte 2307–28. Tagung Technische Zuverlässigkeit (TTZ 2017) - Entwicklung und Betrieb zuverlässiger Produkte*. Leonberg 2017, 2017, S. 17–30.
- [Sie19] Siemens AG. *Siemens Mechatronics Concept Designer*. 2019. URL: <https://www.plm.automation.siemens.com/global/en/products/mechanical-design/mechatronic-concept-design.html> (besucht am 12.10.2019).
- [SW15] Sondermann-Wölke, C. *Entwurf und Anwendung einer erweiterten Zustandsüberwachung zur Verlässlichkeitssteigerung selbstoptimierender Systeme: Dissertation*. Bd. 3. Schriften des Lehrstuhls für Dynamik und Mechatronik, Hrsg. vom Prof. Dr.-Ing. habil. W. Sextro. Aachen: Shaker, 2015.
- [Stu17] Stuecka, R. *Digitaler Zwilling in der Produktion: Von der digitalen Planung bis zum physischen Produkt und zurück: Zukunft Industrien*. Online, 2017.

- [TB16] Tabako Noutang, C. D. und (Betreuer) Kaul, T. *Modellbasierte Analyse der Zuverlässigkeit und der funktionalen Systemsicherheit komplexer mechatronischer Systeme: unveröffentlichte Masterarbeit*. Universität Paderborn, 2016.
- [TLS08] Tajarrood, F. und Latif-Shabgahi, G. „A novel methodology for synthesis of fault trees from MATLAB-Simulink model“. In: *World Academy of Science, Engineering and Technology* 41 (2008), S. 630–636.
- [The19a] The MathWorks, I. *Matlab*. 2019. URL: <https://www.mathworks.com/products/matlab.html> (besucht am 12. 10. 2019).
- [The19b] The Modelica Association. *Modelica Tools: Modelicac*. 2019. URL: <https://www.modelica.org/tools> (besucht am 12. 10. 2019).
- [The19c] The Modelica Association. *Modelica*. 2019. URL: <https://www.modelica.org/> (besucht am 12. 10. 2019).
- [TGS+14] Thomas, P., Gu, X., Samlaus, R., Hillmann, C. und Wihlfahrt, U. „The OneWind™ Modelica Library for Wind Turbine Simulation with Flexible Structure-Modal Reduction Method in Modelica“. In: *Proceedings of the 10 th International Modelica Conference; March 10-12; 2014; Lund; Sweden*. 2014, S. 939–948.
- [Thr05] Thramboulidis, K. „Model-Integrated Mechatronics-Toward a New Paradigm in the Development of Manufacturing Systems“. In: *IEEE Transactions on Industrial Informatics* 1.1 (2005), S. 54–61.
- [TB10] Thramboulidis, K. und Buda, A. „3+ 1 SysML view model for IEC61499 Function Block control systems“. In: *8th IEEE International Conference on Industrial Informatics (INDIN)*. 2010, S. 175–180.
- [TTS98] Torres-Toledano, J. G. und Sucar, L. E. „Bayesian networks for reliability analysis of complex systems“. In: *Progress in Artificial Intelligence—IBERAMIA 98*. Springer, 1998, S. 195–206.
- [TG18] Trächtler, A. und Gausemeier, J., Hrsg. *Steigerung der Intelligenz mechatronischer Systeme*. Intelligente technische Systeme - Lösungen aus dem Spitzencluster it's OWL. Springer, 2018.
- [US 95] US Department of Defense. *Military handbook Reliability Prediction of Electronic Equipment: MIL-HDBK-217F Notice 2*. 1995.
- [VDI16] VDI-Fachausschuss 7.23. *VDI Statusreport: Digitale Chancen und Bedrohungen - Geschäftsmodelle für die Industrie 4.0*. Hrsg. von VDI/VDE-Gesellschaft. 2016.
- [VDI04] VDI. *VDI 2206 - Entwicklungsmethodik für mechatronische Systeme*. 2004.
- [Val16] Vallance, L. *Quick Fatigue Tool for Matlab: Stress-based Fatigue Analysis Code for Finite Element Models*. 2016. URL: <https://github.com/fsgeek/91/quick-fatigue-tool> (besucht am 27. 07. 2019).
- [VKV02] Valtorta, M., Kim, Y.-G. und Vomlel, J. „Soft Evidential Update For Probabilistic Multiagent Systems“. In: *International Journal of Approximate Reasoning* 29.1 (2002), S. 71–106.

-
- [Ver06] Verein Deutscher Ingenieure. *Terminologie der Zuverlässigkeit*. 2006-07-01.
- [WRF15] Walden, D. D., Roedler, G. J. und Forsberg, K. „INCOSE Systems Engineering Handbook Version 4: Updating the Reference for Practitioners“. In: *INCOSE International Symposium*. Bd. 25. 2015, S. 678–686.
- [WMOS+12] Weber, P., Medina-Oliva, G., Simon, C. und Iung, B. „Overview on Bayesian networks applications for dependability, risk analysis and maintenance areas“. In: *Engineering Applications of Artificial Intelligence* 25.4 (2012), S. 671–682.
- [WRB+04] Wilson, P. R., Ross, J. N., Brown, A. D. und Rushton, A. „Multiple Domain Behavioral Modeling Using VHDL-AMS“. In: *Proceedings of the 2004 International Symposium on Circuits and Systems (ISCAS'04)*. Bd. 5. 2004.
- [Wol08] Wolters, K. „Formalismen, Simulation und Potenziale eines nutzungsdauroptimierenden Zuverlässigkeitskonzepts“. Diss. Universität Duisburg-Essen, 2008.
- [WK07] Wratil, P. und Kieviet, M. *Sicherheitstechnik für Komponenten und Systeme*. 2007.
- [WLW04] Wu, T.-F., Lin, C.-J. und Weng, R. C. „Probability Estimates for Multi-Class Classification by Pairwise Coupling“. In: *Journal of Machine Learning Research* 5.Aug (2004), S. 975–1005.
- [XYM+11] Xiang, J., Yanoo, K., Maeno, Y. und Tadano, K. „Automatic synthesis of static fault trees from system models“. In: *Secure Software Integration and Reliability Improvement (SSIRI), 2011 Fifth International Conference on*. 2011, S. 127–136.
- [Zag10] Zagorecki, A. „Local probability distributions in Bayesian networks: knowledge elicitation and inference“. Dissertation. University of Pittsburgh, 2010.
- [ZD06] Zagorecki, A. und Druzdzal, M. J. „Knowledge engineering for Bayesian Networks: how common are noisy-max distributions in practice?“ In: *European Conference on Artificial Intelligence (ECAI)*. 2006, S. 482.
- [ZTZ13] Zhao, F., Tian, Z. und Zeng, Y. „Uncertainty quantification in gear remaining useful life prediction through an integrated prognostics method“. In: *Reliability, IEEE Transactions on* 62.1 (2013), S. 146–159.
- [iXt19] iXtronics GmbH. *CAMeL-View*. 2019. URL: <https://ix.tronics.com/de/software> (besucht am 12.10.2019).

A Anhang

A.1 Berechnung der vollständigen gemeinsamen Wahrscheinlichkeitsverteilung und Marginalisierung

Das in Abschnitt 3.4.2.3 in Abbildung 3.6 vorgestellte statische Bayes'sche Netz soll an dieser Stelle exemplarisch berechnet werden. Das heißt, die gemeinsame Wahrscheinlichkeitsverteilung $P(\gamma)$ wird berechnet und daraus eine Zielgröße, hier $P(C = c_1)$, wird aus dieser marginalisiert. Um die Berechnung zugänglicher zu gestalten, sei für die a priori Wahrscheinlichkeiten $R_A(t) = 0,75$ und $R_B(t) = 0,9$ angenommen.

Mit Gleichung 3.7 kann die gemeinsame Wahrscheinlichkeitsverteilung $P(\gamma)$ für das Bayes'sche Netz berechnet werden:²⁰

$$\begin{aligned}
 P(A = a_1, B = b_1, C = c_1) &= P(a_1)P(b_1)P(c_1|a_1, b_1) = 0,25 \cdot 0,1 \cdot 1 = 0,025 \\
 P(A = a_2, B = b_1, C = c_1) &= P(a_2)P(b_1)P(c_1|a_2, b_1) = 0,75 \cdot 0,1 \cdot 1 = 0,075 \\
 \dots & \\
 P(A = a_2, B = b_2, C = c_2) &= P(a_2)P(b_2)P(c_2|a_2, b_2) = 0,75 \cdot 0,9 \cdot 1 = 0,675.
 \end{aligned} \tag{A.1}$$

Daraus folgt die vollständige gemeinsame Wahrscheinlichkeitsverteilung $P(\gamma)$:

A	a_1	a_2	a_1	a_2	a_1	a_2	a_1	a_2
B	b_1	b_1	b_2	b_2	b_1	b_1	b_2	b_2
C	c_1	c_1	c_1	c_1	c_2	c_2	c_2	c_2
$P(\gamma)$	0,025	0,075	0,225	0	0	0	0	0,675

Tabelle A.1: Gemeinsame Wahrscheinlichkeitsverteilung $P(\gamma)$.

Mit 3.6 kann $P(C = c_1)$ aus $P(\gamma)$ marginalisiert werden:

$$\begin{aligned}
 P(C = c_1) &= P(a_1, b_1, c_1) + P(a_2, b_1, c_1) + P(a_1, b_2, c_1) + P(a_2, b_2, c_1) \\
 &= 0,025 + 0,075 + 0,0225 + 0 = 0,325
 \end{aligned} \tag{A.2}$$

A.2 Weiche Evidenzen in Fehlerursachen

Der in Abschnitt 3.4.3.1 vorgestellte allgemeine Modellierungsansatz zur Abbildung der Systemzuverlässigkeit mittels dynamischer Bayes'scher Netze bedingt eine verzögerte Fortpflanzung von Fehlerursachen für Topologiegraphen mit aufeinander folgenden bi-direktionalen Kanten. In Abbildung A.1 (links) ist ein Topologiegraph gezeigt, wie er

²⁰Um eine kompakte Darstellung zu erhalten entspricht hier $P(X = x_1) = P(x_1)$.

typischerweise für ein Mehrkörpersystem auftreten kann. Der DAG des abgeleiteten dynamischen Bayes'schen Netzes ist in Abbildung A.1 (rechts) abgebildet. Wenn die Komponente R_i ausgefallen ist, also für einen Ausfallzustand mit Boole'schen Zustände $R_i = \text{failed}$, stellt dies eine externe Fehlerursache für den Ausfallzustand S_{i+1} dar. Für Komponente T_{i+2} stellt wiederum S_{i+1} eine externe Fehlerursache dar. Es benötigt demnach zwei Zeitschritte für die Fortpflanzung des Ausfalls von R_i durch das Systemzuverlässigkeitsmodell. Es ist demnach zu beachten, dass für eine Analyse der Systemzuverlässigkeit zu einem gewählten Betrachtungszeitpunkt t_{target} ausreichend viele Zeitschritte zuvor betrachtet werden, um alle Einflüsse möglicher Fehlerursachen auf den Zielknoten abzubilden.

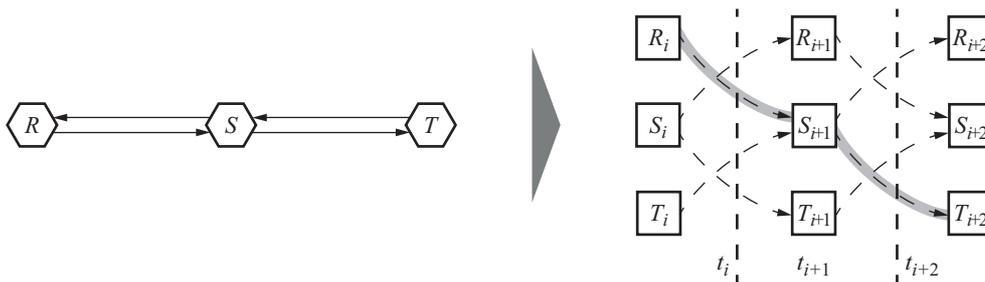


Abbildung A.1: Beispiel eines Topologiegraphen (links) und eines abgeleiteten dynamischen Bayes'schen Netzes mit Fortpflanzung der externen Fehlerursache R_i über mehrere Zeitscheiben.

Dieses Verhalten ist bei der Integration von Systembeobachtungen durch Messungen mit Messunsicherheiten, beispielsweise durch eine Zustandsüberwachung, zu berücksichtigen. Nach der Integration einer weichen Evidenz zu einem Zeitpunkt t_E soll die Systemzuverlässigkeit auf Basis der aktuellen Beobachtung berechnet werden. Dies ist aufgrund des in Abbildung A.1 dargestellten Verhaltens nicht gegeben, da alle Zeitscheiben $t_i < t_E$ die Systemzuverlässigkeit in t_E beeinflussen würden. Daher werden alle temporären Abhängigkeiten zwischen den Zeitscheiben t_i und t_E aus dem DAG gelöscht, sodass sich ein DAG, wie in Abbildung A.2, ergibt. Das dynamische Bayes'sche Netz wird für die Zeitscheiben t_0 bis t_i dem allgemeinen Modellierungsansatz folgend aufgebaut. In Zeitscheibe t_E wird die weiche Evidenz eingebracht und die a priori Wahrscheinlichkeiten entsprechend der Zeitscheibe $t_E = t_{i+1}$ eingesetzt.

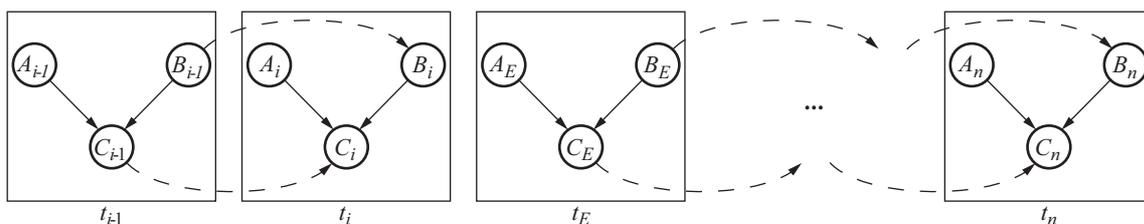


Abbildung A.2: Beispiel eines DAG eines dynamischen Bayes'schen Netzes (vergleiche Abbildung 3.7) mit weicher Evidenz in Zeitscheibe t_E .

A.3 Algorithmen zur Umsetzung des Zuverlässigkeitsmodells

Die in diesem Abschnitt vorgestellten Algorithmen sind in Pseudocode verfasst und dienen der Darstellung spezifischer Funktionen der Umsetzung des Systemzuverlässigkeitsmodells. Zunächst wird in Algorithmus 1.1 die Vorgehensweise zur Auflösung von Zyklen im Topologiegraph zur Umsetzung des DAG gezeigt. Dieses Vorgehen wird in Abschnitt 3.5.2.2 detaillierter erläutert. In Algorithmus 1.2 wird eine Funktion zur Reduzierung obsoleter Bestandteile des DAGs gezeigt wie in Abschnitt 3.5.2.3 erläutert wird. Der Algorithmus 1.3 dient der Umsetzung eines Digitalen Zwillings auf Basis der Integration weicher Evidenzen in das Systemzuverlässigkeitsmodell mithilfe einer zustandsbasierten Diagnose. Die Umsetzung des Digitalen Zwillings wird in Abschnitt 4.3.3 vorgestellt.

Algorithmus 1.1 DetectRemoveCyc

Detektion und Auflösen komplexer Zyklen im Intra-DAG \mathbf{A}_{intra,t_i}

Input:

\mathbf{A}_{intra,t_i} Intra-DAG mit Zyklen

$\mathbf{A}_{inter,t_i,i+1}$ Inter-DAG

Output:

\mathbf{A}_{intra,t_i} Intra-DAG ohne Zyklen

$\mathbf{A}_{inter,t_i,i+1}$ Inter-DAG mit zusätzlichen temporalen Abhängigkeiten

```

1: berechne:  $e^{\mathbf{A}_{intra,t_i}}$  ▷ Gleichung 3.16
2: while any  $e^{\mathbf{A}_{intra,t_i}}(j, j) > 1$  do ▷ Detektiere: Zyklus mit Gleichung 3.17
3:   suche: kleinste  $j$  für  $e^{\mathbf{A}_{intra,t_i}}(j, j) > 1$ 
4:   suche: alle Elternknoten  $par(j)$  von  $j$  in  $\mathbf{A}_{intra,t_i}$ 
5:   for all  $par(j)$  do
6:     if  $e^{\mathbf{A}_{intra,t_i}}(par(j), par(j)) > 0$  then ▷ Prüfe: Elternknoten Teil des Zyklus
7:       setze:  $\mathbf{A}_{intra,t_i}(par(j), j) \leftarrow 0$  ▷ Lösche: Kante
8:       setze:  $\mathbf{A}_{inter,t_i,i+1}(par(j), j + m) \leftarrow 1$  ▷ Füge hinzu: Kante
9:       break ▷ Beende: Iteration über Elternknoten  $par(j)$ 
10:    end if
11:  end for
12:  berechne:  $e^{\mathbf{A}_{intra,t_i}}$  ▷ Gleichung 3.16
13: end while

```

Algorithmus 1.2 ReduceDAG

 Detektiere und lösche Knoten ohne Einfluss auf Zielknoten j_{target} .

Input: $\mathbf{A}_{DAG,i}$ zu reduzierender verallgemeinerte DAG**Output:** $\mathbf{A}_{DAG,i}$ reduzierter DAG

- 1: initialisiere: $j_{target} \leftarrow$ Zielknoten aus Datenobjekt *IntModData*
 - 2: initialisiere: $d \leftarrow emptyArray$
 - 3: berechne: $\mathbf{A}_{ges} = \mathbf{A}_{intra,t_i} + \mathbf{A}_{inter,t_i,i+1}$
 - 4: setze: $\mathbf{A}_{ges}(j, j) \leftarrow 0 \forall j$ mit $j \in \{1, \dots, m_{r1}\}$ ▷ Lösche: Hauptdiagonalelemente
 - 5: berechne: $e^{\mathbf{A}_{ges}}$ ▷ Gleichung 3.16
 - 6: **for** $k = 1$ to m_{r1} **do**
 - 7: **if** $e^{\mathbf{A}_{ges}}(k, j_{target}) == 0$ **then** ▷ Prüfe: Zielknoten nicht erreichbar
 - 8: füge hinzu: $d \leftarrow k$
 - 9: **end if**
 - 10: **end for**
 - 11: **for all** d **do** lösche Dimensionen aus $\mathbf{A}_{DAG,i}$ und Informationen in Datenobjekt *IntModData*
-

Algorithmus 1.3 SetMonitoredCPT

 Setze bedingte Zuverlässigkeit in Evidenzknoten auf Basis von \mathbf{p}_{SVM} .

Input:Wahrscheinlichkeit der Klassenzugehörigkeit \mathbf{p}_{SVM} des überwachten Knotens*Distribution* Instanz der Klasse der überwachten Komponente t_E Zeitpunkt der Evidenz mit $e \in \{0, \dots, n\}$ **Output:** i_v Index der virtuellen Zuverlässigkeit*CPT* CPTs für alle Knoten mit $t_i > t_E$ der überwachten Komponente

- 1: initialisiere: $i_v \leftarrow 0$ Index der virtuellen lokalen Zuverlässigkeit
 - 2: initialisiere: $diff_{min} \leftarrow 1$ Differenz: virtuelle und reale lokale Zuverlässigkeit
 - 3: initialisiere: *CPT* \leftarrow empty Array
 - 4: berechne: $diff = Distribution.R - p_{Ff}$ ▷ vergleiche Tabelle 3.7 und Gleichung 4.14
 - 5: **for each** d in $diff$ **do**
 - 6: **if** $d > 0$ AND $d < diff_{min}$ **then**
 - 7: setze: $diff_{min} = d, i_v = i_{aktuell}$
 - 8: **end if**
 - 9: **end for**
 - 10: **for** $k = 1$ to $(m_z - i_v)$ **do**
 - 11: setze: $CPT_{tmp} \leftarrow \mathbf{E}$ Einheitsmatrix der Größe $n_c \times n_c$ ▷ vergleiche Tabelle 4.4
 - 12: setze: $CPT_{tmp}(1, 1) \leftarrow Distribution.condR(i_v + k - 1)$
 - 13: setze: $CPT_{tmp}(n_c, 1) \leftarrow Distribution.condF(i_v + k - 1)$
 - 14: setze: $CPT(e + k) \leftarrow CPT_{tmp}$
 - 15: **end for**=0
-

Lebenslauf

Thorben Henning Kaul

geboren am 09.03.1989 in Höxter

Schulische Ausbildung

- 08/1999 – 06/2005 Hoffmann-von-Fallersleben Realschule, Höxter
- 08/2005 – 05/2008 König-Wilhelm-Gymnasium, Höxter
Abschluss: Allgemeine Hochschulreife

Akademische Ausbildung

- 10/2008 – 03/2012 Universität Paderborn
Studium des Maschinenbaus, Studienrichtung Mechatronik
Abschluss: Bachelor of Science
- 04/2012 – 09/2014 Universität Paderborn
Studium des Maschinenbaus, Studienrichtung Mechatronik
Abschluss: Master of Science

Berufstätigkeit

- 12/2009 – 12/2010 Universität Paderborn, Heinz-Nixdorf-Institut
Lehrstuhl für Produktentstehung
Studentische Hilfskraft
- 01/2011 – 07/2012 Wincor-Nixdorf International GmbH, Banking Division,
Hardware Development - Architecture and Technology
Werkstudent
- 08/2012 – 01/2013 Universität Paderborn, Fakultät für Maschinenbau
und Lehrstuhl für Mechatronik und Dynamik
- 08/2013 – 07/2014 Studentische Hilfskraft
- seit 10/2014 Universität Paderborn, Fakultät für Maschinenbau
Lehrstuhl für Dynamik und Mechatronik
Wissenschaftlicher Mitarbeiter

Thorben Kaul