

Prädiktive Instandhaltung von elektromechanischen Relais mit Methoden der künstlichen Intelligenz

Der Fakultät für Elektrotechnik, Informatik und Mathematik
der Universität Paderborn

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

vorgelegte Dissertation von

Fabian Winkel
geboren am 20.01.1994 in Bielefeld

Eidesstattliche Erklärung

Ich versichere, dass ich die vorliegende wissenschaftliche Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe. Die Stellen der Arbeit, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen sind, wurden unter Angabe der Quelle als Entlehnung deutlich gemacht. Das gleiche gilt auch für beigegebene Skizzen und Darstellungen. Diese Arbeit hat in gleicher oder ähnlicher Form meines Wissens nach noch keiner Prüfungsbehörde vorgelegen.

Paderborn,

Datum

Unterschrift

Zusammenfassung

Elektromechanische Relais sind integraler Bestandteil von diversen Anwendungen: von Haushaltsgeräten über Kraftfahrzeuge bis zu komplexen industriellen Anlagen. Dies ist auf ihre Vorteile, wie einem geringen Kontaktwiderstand, der galvanischen Trennung und niedrigen Kosten zurückzuführen. Ein wesentlicher Nachteil ist der Verschleiß von Relais. Deshalb werden im Kontext dieser Arbeit drei Themengebiete zur Instandhaltung von elektromechanischen Relais betrachtet:

- Im Kontext von „Predictive Maintenance“ (dt. Prädiktive Instandhaltung) (PdM) soll der Zustand von Relais geschätzt werden, damit diese vor einen drohenden Ausfall gewechselt werden können. Der Stand der Technik kann nicht auf Relais übertragen werden, weil dieser größtenteils aus Forschungsvorhaben hervorgeht, die sich mit monoton degradierenden Bauteilen beschäftigten. Im Gegensatz dazu ist die Degradierung von Relais deutlich komplexer. Deshalb wird im Rahmen dieser Arbeit das Verfahren „Maintenance-Algorithm for Unlabeled Data“ (dt. Instandhaltungsalgorithmus für ungelabelte Daten) (MAUD) vorgestellt, mit dem mittels Pseudo-Labeling ein drohender Ausfall identifiziert werden kann. Im Vergleich zum Stand der Technik erreicht MAUD eine 61 % höhere Betriebsdauer bei gleichzeitiger Reduktion der Ausfälle 70 %. Aufgrund des Praxisbezugs wird darüber hinaus die Implementierung von MAUD auf einem Mikrocontroller untersucht.
- Die Lebensdauer von Relais kann durch Reduktion des Prellens verlängert werden. Dafür ist eine Manipulation des Steuersignals notwendig, die durch zwei Parameter definiert wird. Die Herausforderung besteht darin, beide Parameter während des Betriebs zu optimieren. Der in dieser Arbeit vorgestellte „Bouncing-Optimization-Algorithm“ (dt. Preloptimierungsalgorithmus) (BOA) nutzt hierzu eine Variante der Partikelschwarmoptimierung. Im Rahmen der Evaluierung wird gezeigt, dass BOA das Prellen um 79 % beim Einschalten (59 % beim Ausschalten) erreicht.
- Ein Ausfall kann oft durch erneutes Schalten korrigiert werden und Degradierung kann reversibel sein. Bei dem Verfahren „Contact Self-healing via Abrasion“ (dt. Kontaktselftheilung via Abrasion) (CSA) werden diese Aspekte automatisiert ausgenutzt, um die Folgen eines Ausfalls zu minimieren. Das Potential von CSA wird durch Untersuchungen aufgezeigt, so konnten bspw. 75 % Testrelais geheilt werden.

Somit wird dem Verschleiß von Relais auf drei Weisen begegnet: Mit MAUD wird das Auftreten von Ausfällen reduziert. Falls es trotzdem zu einem Ausfall kommt, können die Folgen durch CSA minimiert werden. Schlussendlich ermöglicht BOA die Verlängerung der Lebensdauer.

Abstract

Electromechanical relays are an integral part of various applications: from household appliances to motor vehicles and complex industrial systems. This is due to their advantages, such as low contact resistance, galvanic isolation and low cost. A major disadvantage is the wear of relays. Therefore, in the context of this work, three areas of maintenance of electromechanical relays are considered:

- In the context of „Predictive Maintenance“ (dt. Prädiktive Instandhaltung) (PdM), the status of relays should be estimated so that they can be replaced before an imminent failure. The state of the art cannot be applied to relays because it is largely derived from research projects dealing with monotonically degrading components. In contrast, the degradation of relays is much more complex. For this reason, the „Maintenance-Algorithm for Unlabeled Data“ (dt. Instandhaltungsalgorithmus für ungelabelte Daten) (MAUD) is presented in this thesis, with which an impending failure can be identified by means of Pseudo-Labeling. Compared to the state of the art, MAUD achieves a 61 % higher operating time with a simultaneous reduction in failures of 70 %. Due to the practical relevance, the implementation of MAUD on a microcontroller is also investigated.
- The lifetime of relays can be extended by reducing the bounce. This requires manipulation of the control signal, which is defined by two parameters. The challenge is to optimize both parameters during operation. The „Bouncing-Optimization-Algorithm“ (dt. Prelloptimierungsalgorithmus) (BOA) presented in this thesis uses a variant of particle swarm optimization for this purpose. As part of the evaluation, it is shown that BOA achieves a bounce of 79 % when switching on (59 % when switching off).
- A failure can often be corrected by switching again and degradation can be reversible. In the „Contact Self-healing via Abrasion“ (dt. Kontaktselfstheilung via Abrasion) (CSA) process, these aspects are automatically exploited to minimize the consequences of a failure. The potential of CSA is demonstrated by studies, for example 75 % test relays could be healed.

Relay wear is thus countered in three ways: The occurrence of failures is reduced with MAUD. If a failure does occur, the consequences can be minimized by CSA. Finally, BOA enables the lifetime to be extended.

Danksagung

Diese Arbeit entstand im Kontext meiner Tätigkeit in der Technologieentwicklung des Geschäftsbereichs Interface des Unternehmens Phoenix Contact in Kooperation mit dem Fachgebiet Leistungselektronik und Elektrische Antriebstechnik der Universität Paderborn. An dieser Stelle möchte ich allen Personen danken, die zum Gelingen dieser Arbeit in vielfältiger Weise beitrugen. Mein besonderer Dank gilt:

- Herrn Prof. Dr.-Ing. Joachim Böcker für die wissenschaftliche Betreuung und Begleitung der Arbeit sowie für die Bereitstellung der ausgezeichneten Arbeitsumgebung am Institut.
- Herrn Prof. Dr.-Ing. Bernd Henning für die freundliche Übernahme des Korreferats.
- Herrn Dr.-Ing. Peter Scholz für die fachliche Betreuung der Arbeit und die Durchsicht des Manuskriptes – auch über die Beschäftigung bei Phoenix Contact hinaus.
- Herrn Dr.-Ing. Oliver Wallscheid für die konstruktive, engagierte Zusammenarbeit und die beeindruckend schnelle Durchsicht der Manuskripte.
- Dem Unternehmen Phoenix Contact für die Ermöglichung des Forschungsvorhabens und der Bereitstellung des Messequipments und Labors.
- Herrn Johannes Deuse-Kleinsteuber für die Zusammenarbeit und die stets angenehme Atmosphäre auch außerhalb der Arbeitszeit.
- Meiner Arbeitsgruppe bestehend aus Herrn Dr.-Ing. Christian Reinhold, Elmar Schaper, Philipp John und Dr.-Ing. Tobias Noesselt für die konstruktiven Ideen und das stets offene Ohr.
- Herrn Tobias Kaps für die Bachelorarbeit, die im Kontext dieser Arbeit entstand.
- Allen Kollegen des Fachbereiches Elektrische Antriebstechnik und Leistungselektronik der Universität Paderborn für die freundschaftliche und konstruktive Zusammenarbeit.

Schlussendlich bedanke ich mich bei meiner Familie, die mich stets unterstützte. In besonderem Maße möchte ich mich bei meiner Julia bedanken, die mir – mit stetig wachsenden Bauch – die deutsche Grammatik und Orthografie näherbrachte.

Danke!

Paderborn, den 23. Februar 2025 Fabian Winkel

Inhaltsverzeichnis

Zusammenfassung	ii
Abstract	iii
Danksagung	iv
1 Einleitung	1
1.1 Herausforderungen bei der Nutzung	2
1.2 Lösungsansätze	3
2 Elektromechanische Relais	7
2.1 Konstruktiver Aufbau von Relais	7
2.2 Degradierung von Relais	11
3 Prädiktive Instandhaltung	15
3.1 Forschungsstand	16
3.1.1 Datensätze zur prädiktiven Instandhaltung	17
3.1.2 Verfahren zur prädiktiven Instandhaltung	21
3.1.3 Semi-überwachtes Lernen	33
3.1.4 Verfahren zur Effizienzsteigerung von ANN	35
3.1.5 Zusammenfassung	38
3.2 Grundlagen	39
3.2.1 Begrifflichkeiten und allgemeines Vorgehen	39
3.2.2 Künstliche neuronale Netze	41
3.2.3 Spezielle Neuronen-Arten	44
3.2.4 Autoencoder und Variational Autoencoder	46
3.2.5 Methoden zur Effizienzsteigerung von ANN	49
3.3 Versuchsaufbau	52
3.3.1 Hardware	52
3.3.2 Messsoftware	55
3.3.3 Lasten	58
3.4 Datensatz	59
3.4.1 Degradierung von Relais	59
3.4.2 Statistik	62
3.4.3 Extrahierte Merkmale	64
3.4.4 Publizierter Datensatz	68

INHALTSVERZEICHNIS

3.5	Forschungsbedarf	70
3.5.1	Statistische Modellierung	70
3.5.2	Überwachte Lernverfahren	71
3.5.3	Unüberwachte Lernverfahren	74
3.5.4	Aufzeigen des Potenzials für Prädiktive Instandhaltung	78
3.5.5	Forschungsdesiderat	79
3.6	Semi-überwachtes Lernverfahren für PdM	80
3.6.1	Instandhaltungsalgorithmus für ungelabelte Daten	80
3.6.2	Evaluation	85
3.7	Inferenz auf dem Mikrocontroller	91
3.7.1	Versuchsbeschreibung	91
3.7.2	Evaluierung	94
3.8	Fazit	97
3.8.1	Datensatz	97
3.8.2	Maschinelles Lernverfahren	98
3.8.3	Applikation auf eingebetteten Systemen	98
4	Reduzieren des Prellens	101
4.1	Grundlagen	102
4.1.1	Stand der Forschung und Technik	102
4.1.2	Partikel Schwarm Optimierung	103
4.1.3	Box-Muller-Transformation	107
4.2	Messaufbau	108
4.2.1	Problemdefinition	110
4.3	Ableitung eines Optimierungsproblems	111
4.4	Darstellung des Forschungsbedarfs	114
4.5	Bouncing-Optimization-Algorithm (BOA)	116
4.6	Evaluation	122
4.6.1	Funktionsweise	122
4.6.2	Qualitativ	124
4.6.3	Quantitativ	126
4.6.4	Robustheit	128
4.6.5	Schaltzeit	130
4.7	Fazit	132
5	Abrasive Kontaktselfheilung	135
5.1	Forschungsstand	135
5.2	Ausfallverhalten von Relais	136
5.3	Messaufbau	137
5.4	Contact Self-healing via Abrasion	139
5.5	Evaluation	141
5.5.1	Messgrößen bei Reiben	142
5.5.2	Einfluss der Zeitverschiebung	144
5.5.3	Einflüsse auf die Kontaktobерfläche	145

5.6 Fazit	148
6 Fazit	151
6.1 Beiträge der Arbeit	151
6.1.1 Prädiktive Instandhaltung	151
6.1.2 Reduzieren des Prellens	152
6.1.3 Abrasive Kontaktselbstheilung	153
6.2 Offene Fragestellungen	153
6.2.1 Prädiktive Instandhaltung	153
6.2.2 Reduzieren des Prellens	154
6.2.3 Abrasive Kontaktselbstheilung	155
6.3 Ausblick	156
Verzeichnisse	161
Tabellenverzeichnis	161
Abbildungsverzeichnis	161
Abkürzungen	164
Glossar	165
Lebenslauf	165
Literatur	167
Eigene Publikationen	175

1 Einleitung

Das elektromechanische Relais ist eines der ältesten Bauteile der Elektrotechnik. Schon bei der Etablierung von Telegrafensystemen vor über 200 Jahren hatten Relais eine Schlüsselfunktion, weil mit ihnen erstmalig eine Verstärkung von elektrischen Signalen möglich war. Im Laufe der letzten Jahrhunderte wurde eine Vielzahl von Relaistypen entwickelt.

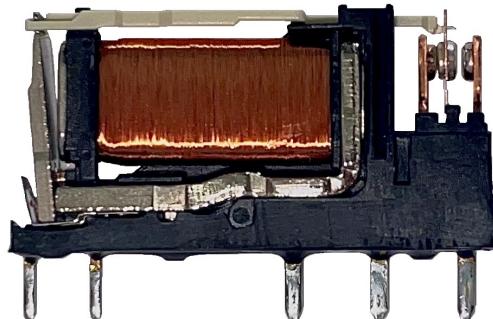


Abb. 1.1: Ein elektromechanisches Relais ohne Gehäuse

Der Aufbau eines elektromechanischen Relais ist exemplarisch in Abb. 1.1 gezeigt. Die Konstruktion ist beim Großteil der Relais ähnlich: Eine elektrische Spule wird dazu verwendet, einen Anker anzuziehen. Dessen Bewegung wird auf einen Kontakt übertragen, sodass auf der Kontaktseite ein Stromkreis geschlossen wird. Durch Anpassungen an der Kontaktgeometrie und am Kontaktmaterial können Relais in einem Leistungsspektrum von wenigen Milliwatt bis zu mehreren hundert Kilowatt eingesetzt werden. Des Weiteren können Relais sich je nach Anwendung durch die Kontakte unterscheiden. Klassisch ist ein Kontaktpaar, es bestehen jedoch auch Formen, bei denen mehrere Kontaktpaare zur Verwendung kommen. Damit können bspw. mehrere Lastkreise mit einem Steuersignal geschaltet oder der Schaltzustand überwacht werden. Weiterhin bestehen Unterschiede darin, wie viele Schaltzustände stabil, d.h. ohne Erregung, gehalten werden können.

Die vielen Bauformen von Relais lassen erahnen, wie groß deren Anwendungsspektrum ist. Historisch war die Nutzung von Relais für Telegrafensysteme und Logikschaltungen in

Computern bedeutend. Dabei wurden elektromechanische Relais jedoch von Halbleiterrelais ersetzt, was auf deren bedeutend geringeren Schaltzeiten zurückzuführen ist. Weiterhin weisen Halbleiterrelais längere Lebensdauern auf, können geräuschlos betrieben werden, die Kontakte prellen nicht und es besteht eine geringere Gefahr für Funkenbildung. Angesichts dieser Vorteile wird die Bedeutung von elektromechanischen Relais oftmals unterschätzt. Halbleiterrelais haben gegenüber elektromechanischen Relais jedoch auch Nachteile: So sind in der Applikation weitere Komponenten (bspw. Dioden) notwendig. Zudem sind Halbleiterrelais in der Regel teuer als elektromechanische Relais. Weiterhin ist der geringere Durchgangswiderstand von elektromechanischen Relais vorteilhaft. Genauso wie deren galvanische Trennung von Steuer- und Lastkreis und vom Lastkreis (wenn die Kontakte geöffnet sind). Diese Trennung besitzen Halbleiterrelais nicht, sodass im ausgeschalteten Zustand fast immer einen Leckstrom fließt. Schlussendlich gibt es für beide Relaisarten vorteilhafte Anwendungsfelder, weshalb elektromechanische Relais auch in Zukunft eine wichtige Rolle als elektrisches Bauteil spielen.

1.1 Herausforderungen bei der Nutzung

Mit der Nutzung von elektromechanischen Relais gehen einige Herausforderungen einher. Die Kontakte sind verschleißbehaftet, wodurch die Funktionstüchtigkeit eines Relais beschränkt werden kann. Es ist möglich, dass die Kontakte ihre Leitfähigkeit verlieren oder sich nicht mehr öffnen lassen. Weiterhin kann der Durchgangswiderstand steigen und somit weniger Spannung an der Last abfällt als geplant. Da Relais Bauteile in Systemen sind, beeinträchtigt der Verschleiß auch den ordnungsgemäßen Betrieb des jeweiligen Systems. Wodurch wiederum Kosten, Sicherheitsrisiken und Stillstands Zeiten entstehen. Deshalb ist die Instandhaltung von Relais unabdingbar.

Im Rahmen dieser Arbeit werden Beiträge zu drei Herausforderungen geleistet, die im Folgenden näher erörtert werden:

1. Die Lebensdauer eines Relais ist nicht bekannt.

Die Lebensdauer von Relais ist von diversen Phänomenen abhängig, die nicht zwangsläufig auftreten müssen, sich aber auch überlagern können und in ihrer Wirkung grundsätzlich von Betriebsbedingungen abhängig sind. Bspw. kann Material von einem zum anderen Kontakt migrieren, wodurch sich eine Pille ausbildet, die zum Verschweißen der Kontakte führen kann. Ein weiteres Beispiel stellen induktive Lasten dar, bei denen durch das Trennen der Kontakte ein Lichtbogen hervorgerufen wird. Durch diesen wird Kontaktmaterial verbrannt und die Kontakte verkleinert. Viele weitere Effekte sind zu beobachten und werden in Abschn. 2.2 genauer erläutert. Für die Instandhaltung von Relais ergibt sich durch diese Phänomene eine Herausforderung: Auch bei identischen Betriebsbedingungen haben Relais individuelle Lebensdauern, die sich um mehrere zehntausend Schaltzyklen unterscheiden. Deshalb bedarf es einer prädiktiven Instandhaltung, mit der drohende Ausfälle proaktiv unterbunden werden können. Weiterhin beinhaltet diese Instandhaltungsform das Potenzial, die

Betriebskosten von Anlagen zu senken, indem Wartungen bedarfsgerecht und damit seltener durchgeführt werden als bei einer intervallbasierten Instandhaltung. Auch der Aspekt der Nachhaltigkeit bzw. des effizienten Einsatzes von Ressourcen wird damit adressiert.

2. Durch die Reduktion des Prellens kann die Lebensdauer verlängert werden.

Bedingt durch die Konstruktion schlagen die Kontakte beim Schließen mehrfach aufeinander – dieses Phänomen wird auch als Prellen bezeichnet. Das Problem hierbei liegt vorwiegend darin, dass die Degradierung der Relais beschleunigt wird. Weiterhin verursacht das Prellen eine Geräuschentwicklung und stört die Übertragung von Signalen. Durch die Reduktion des Prellens kann die Lebensdauer von Relais verlängert werden. Dies hat einen positiven Einfluss auf die Betriebskosten und steigert die Effizienz der Ressourcennutzung.

3. Die Degradierung kann reversibel sein.

Eine Fehlfunktion von einem Relais muss nicht endgültig sein – der Großteil der fehlerhaften Schaltvorgänge kann durch eine Wiederholung des Schaltvorgangs korrigiert werden. Dies kann bspw. auf ein Verhaken der Kontakte zurückgeführt werden: Die Kontaktobерfläche raut sich durch den Betrieb so stark auf, dass diese aneinanderhaften können. In der Praxis etablierte sich deshalb, bei einem Ausfall auf das Relais zu klopfen, damit sich die Kontakte lösen. Durch derartige Phänomene wird die Fehlersuche erschwert und viel Zeit des Instandhaltungspersonals gebunden. Ein Verfahren, das dieses „Relais-Klopfen“ automatisiert, kann die Instandhaltung von Relais erheblich vereinfachen.

1.2 Lösungsansätze

Zentrales Ziel dieser Arbeit ist es, einen Beitrag zur Lösung der zuvor genannten Herausforderungen zu leisten. Dabei stehen digitale Strategien im Fokus, die sich durch Softwareanpassungen auf verschiedene Relaistypen übertragen lassen. Darin unterscheidet sich die Arbeit vom Großteil der bisherigen Forschungsvorhaben zur Optimierung von Relais, welche den Fokus auf Hardwareanpassungen legen. Die Lösungsansätze dieser Arbeit sind komplementär zu Hardwareoptimierungen anzuwenden.

In Abb. 1.2 sind links der Stand der Technik und drei sich daraus ergebende Herausforderungen bei der Instandhaltung dargestellt. Rechts sind die Beiträge dieser Arbeit visualisiert:

- Die Lebensdauer von Relais ist nicht bekannt.

Die Lebensdauer von Relais ist begrenzt, weshalb ein Relaiswechsel notwendig ist. In der Praxis empfehlen Hersteller ein Wartungsintervall, nachdem die Relais gewechselt werden sollen. Dieses Intervall wird anhand einiger Proberelais bestimmt. In Bezug auf die zuvor genannte Herausforderung wird deutlich, dass diese Strategie nicht

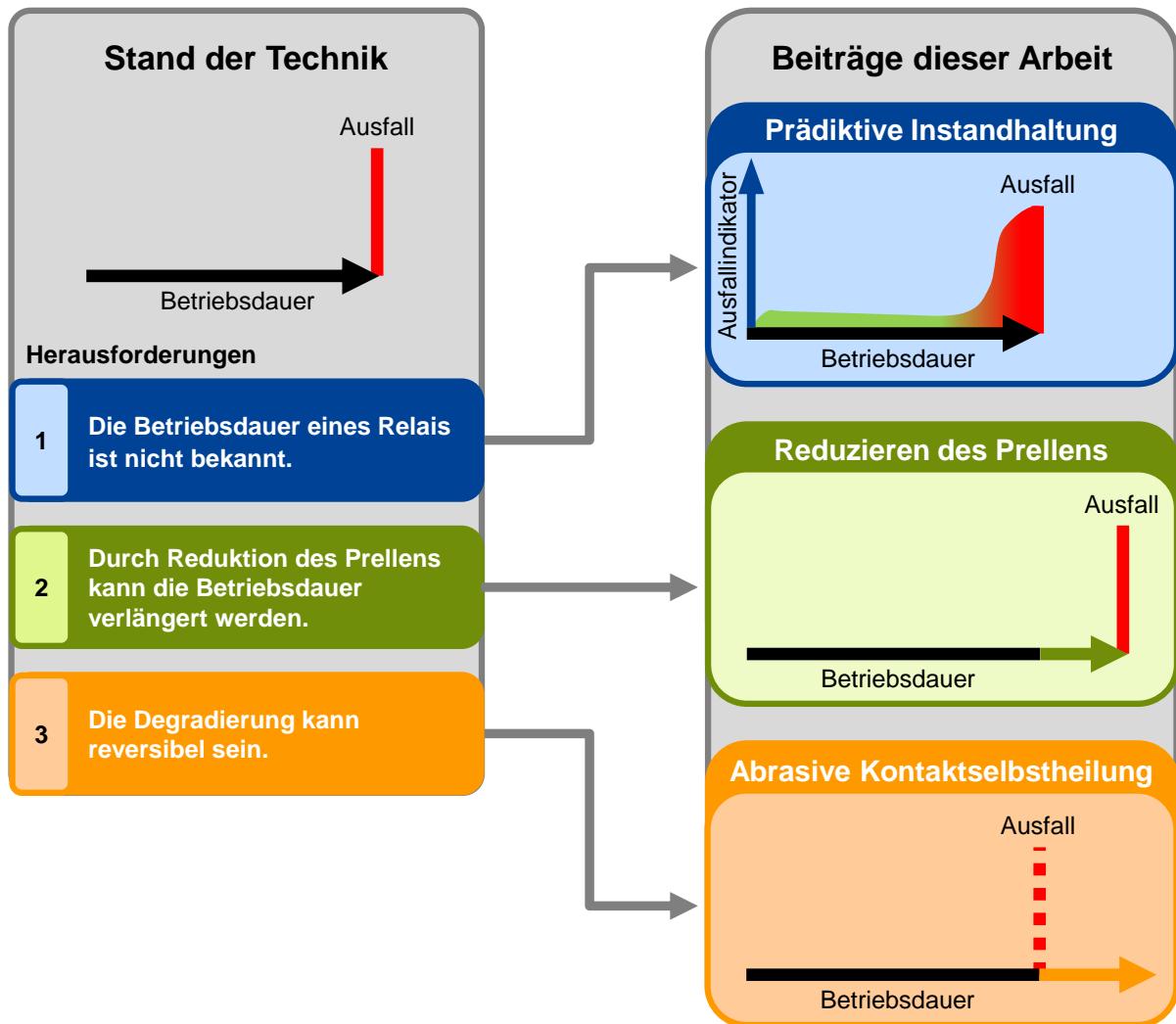


Abb. 1.2: Drei Herausforderungen bei der Instandhaltung von Relais nach dem Stand der Technik und in dieser Arbeit vorgestellte Beiträge zur Lösung

Erfolg versprechend ist: Die individuelle Degradierung wird nicht berücksichtigt. Deshalb wird im Rahmen dieser Arbeit das Verfahren „Maintenance-Algorithm for Unlabeled Data“ (dt. Instandhaltungsalgorithmus für ungelabelte Daten) (MAUD) vorgestellt, mit dem es möglich ist, während der Betriebsdauer einen Ausfallindikator der individuellen Relais zu geben. Derartige Lösungen bestehen bereits – sowohl in der Form von Schwellwerten, die auf Messgrößen angewandt werden, als auch durch künstliche neuronale Netze. Der Stand der Technik hierzu ist in Abschn. 3.1 herausgearbeitet. Die im Kontext dieser Arbeit vorgestellte Lösung grenzt sich hinsichtlich mehrerer Aspekte davon ab: Zunächst ist der verwendete Datensatz deutlich umfangreicher als die bestehenden, wodurch ein realistischer Praxisbezug angenommen werden kann. Ferner wird ein semi-überwachter Lernansatz verfolgt,

wodurch ein Ausfallindikator geschätzt werden kann. Weiterhin wird die Inferenz von ANN auf einem Mikrocontroller untersucht.

- Durch die Reduktion des Prellens kann die Lebensdauer verlängert werden.

Mit Hilfe einer kurzen Manipulation der Versorgungsspannung kann das Prellen von Relais reduziert werden. Im Rahmen dieser Arbeit wird das Verfahren „Bouncing-Optimization-Algorithm“ (dt. Preloptimierungsalgorithmus) (BOA) vorgestellt, das sowohl in Hinblick auf die Reduzierung als auch auf die Zuverlässigkeit den Stand der Technik übertrifft.

- Die Degradierung kann reversibel sein.

Um einen fehlerhaften Schaltvorgang zu korrigieren, genügt es häufig, diesen zu wiederholen. Diese Maßnahme wurde bereits in ähnlicher Form von diversen Unternehmen patentiert und findet in verschiedenen Produkten Anwendung. In dieser Arbeit wird das neuartige Verfahren „Contact Self-healing via Abrasion“ (dt. Kontaktselftheilung via Abrasion) (CSA) beschrieben, bei dem die Spule so angeregt wird, dass die Kontakte aneinander reiben und gleichzeitig leitfähig bleiben. So werden zum einen Ausfälle korrigiert und zum anderen Relais „geheilt“.

Die Lösungen zeigen, dass weiteres Potenzial bei der Instandhaltung bzw. Nutzung von Relais besteht. Die einzelnen Lösungen werden in den separaten Kapiteln 3, 4 und 5 vorgestellt. Zuvor wird in Kap. 2 kurz auf die Grundlagen von elektromechanischen Relais eingegangen. Abgeschlossen wird diese Arbeit vom Fazit in Kap. 6.

Einleitung

2 Elektromechanische Relais

Zentrales Untersuchungsobjekt dieser Arbeit sind elektromechanische Relais. Deshalb wird im folgenden Kapitel zunächst auf deren konstruktiven Aufbau, die Funktionsweise und daraus resultierenden Messgrößen eingegangen. Darauf aufbauend werden allgemeine Kenntnisse zur Degradierung dieser Bauteile erörtert.

2.1 Konstruktiver Aufbau von Relais

Elektromechanische Relais ermöglichen das Öffnen bzw. Schließen eines elektrischen Lastkreises mit Hilfe eines elektrischen Steuerkreises. Dementsprechend ergeben sich viele verschiedene Anwendungsmöglichkeiten für elektromechanische Relais, für die eine Vielzahl von Relaistypen notwendig ist. Im Kontext dieser Arbeit werden grundlegende Erkenntnisse gesammelt, weshalb die Untersuchungen auf einen Relaistyp beschränkt sind. Dabei handelt es sich um einen monostabilen Relaistyp mit drei Kontakten, der in der Automatisierungsindustrie weitverbreitet ist und zum Schalten von Aktoren wie Leuchten, Ventilatoren oder Ventilen verwendet wird. Im Rahmen dieser Arbeit werden die Relais von zwei Herstellern verwendet, die im Weiteren als Hersteller A und B anonymisiert angegeben werden. Die technischen Spezifikationen sind in Tab. 2.1 aufgeführt:

Tab. 2.1: Technische Spezifikationen des Relaistyps nach Hersteller

	Hersteller A	Hersteller B
Abmaße (Breite x Höhe x Tiefe)	28 mm x 15 mm x 5 mm	
Kontaktmaterial	AgSnO ₂	
Magnetsystem	monostabil	
Spulennennspannung	24 V	
Spulenwiderstand	3388 Ω (± 10 %)	3390 Ω (± 10 %)
Kontaktnennspannung	24 V	
Kontaktnennstrom (max.)	6 A	
Minimale mechanische Schaltvorgänge	5 · 10 ⁶	10 · 10 ⁶

Das Magnetsystem von Relais kann sich dahin gehend unterscheiden, dass ein oder mehrere Schaltzustände ohne anliegende Steuerspannung stabil sind. Bei dem monostabilen Magnetsystem der untersuchten Relais ist nur ein Zustand stabil.

Die Angaben beider Hersteller unterscheiden sich lediglich bei den minimalen mechanischen Schaltvorgängen signifikant. Diese geben jedoch an, wie groß die Lebensdauer der Relais ohne angeschlossenen Lastkreis mindestens ist. Im Kontext dieser Arbeit wird der Lastkreis angeschlossen, wodurch verschiedene Degradierungsphänomene an den Kontakten auftreten und die Lebensdauer der Relais verkürzen. Die Funktionsweise wird im Folgenden nach *Gurevich* beschrieben [1].

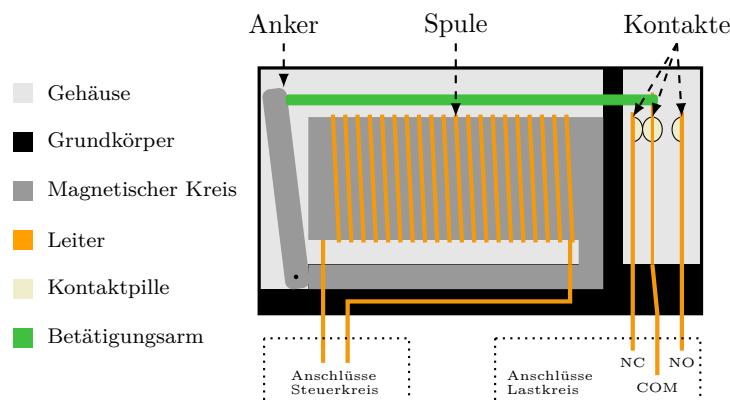


Abb. 2.1: Aufbau eines elektromechanischen Relais

Dazu wird Abb. 2.1 herangezogen, in der der Aufbau eines Relais unter Vernachlässigung konstruktiver Details dargestellt ist. Die Komponenten der Relais sind am Grundkörper befestigt und durch ein Gehäuse aus glasfaserverstärktem Kunststoff geschützt, das wiederum mit dem Grundkörper vergossen ist. Die Komponenten können in Steuer- und Lastkreisseite unterteilt werden:

Eine Spule mit einem Metallkern und Anker bilden die Steuerkreisseite. Der Anker wird durch eine Feder in Position gehalten, sodass ein Luftspalt zwischen Anker und Metallkern besteht. Durch Beaufschlagung der Spule mit Spannung entsteht ein magnetisches Feld, wodurch wiederum der Luftspalt geschlossen wird. Dabei bewegt der Anker den Betätigungsarm, sodass die Bewegung auf die Kontakte übertragen wird. Die Lastkreisseite besteht bei den untersuchten Relais aus drei Kontakten: „Common Contact“ (dt. gemeinsamer Kontakt) (COM), „Normally Opened Contact“ (dt. normalerweise geöffneter Kontakt) (NO) und „Normally Closed Contact“ (dt. normalerweise geschlossener Kontakt) (NC).

Entsprechend dem Aufbau ergeben sich mehrere Phasen während eines Schaltvorgangs, die in Abb. 2.2 für einen Einschaltvorgang dargestellt sind. Bei einem Ausschaltvorgang werden die Phasen in umgekehrter Reihenfolge durchlaufen. Zunächst sind die Kontakte NC und COM verbunden (vgl. Abb. 2.2a). Durch die Ankerbewegung werden diese getrennt und COM ist kurzzeitig ohne Kontakt. In Abb. 2.2b ist zu erkennen, dass COM und NO

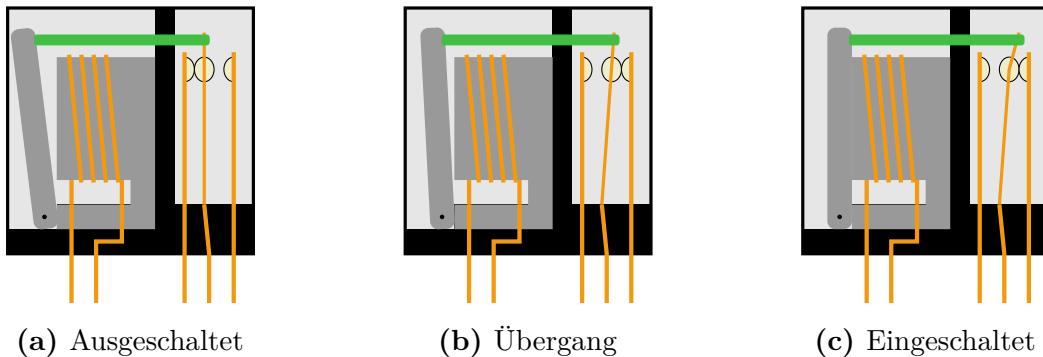


Abb. 2.2: Darstellung der verschiedenen Phasen eines Schaltvorgangs

bereits kontaktiert sind, der Anker jedoch nicht den Spulenkern berührt. In Abb. 2.2c ist letzteres schließlich der Fall. Dadurch werden die Kontakte aufeinander gepresst, wobei die Leiter als Federelemente fungieren. Dieser Vorgang zwischen Abb. 2.2b und 2.2c wird als „Überhub“ bezeichnet und ist notwendig, damit zum einen der Kontaktwiderstand gering gehalten wird und zum anderen das Ausschalten durch die Federspannung unterstützt wird. Weiterhin ist es möglich, die Bewegungstrajektorien der Kontaktpillen so zu gestalten, dass die Kontakte beim Schalten aneinander reiben und auf diesem Weg bspw. Verunreinigungen lösen.

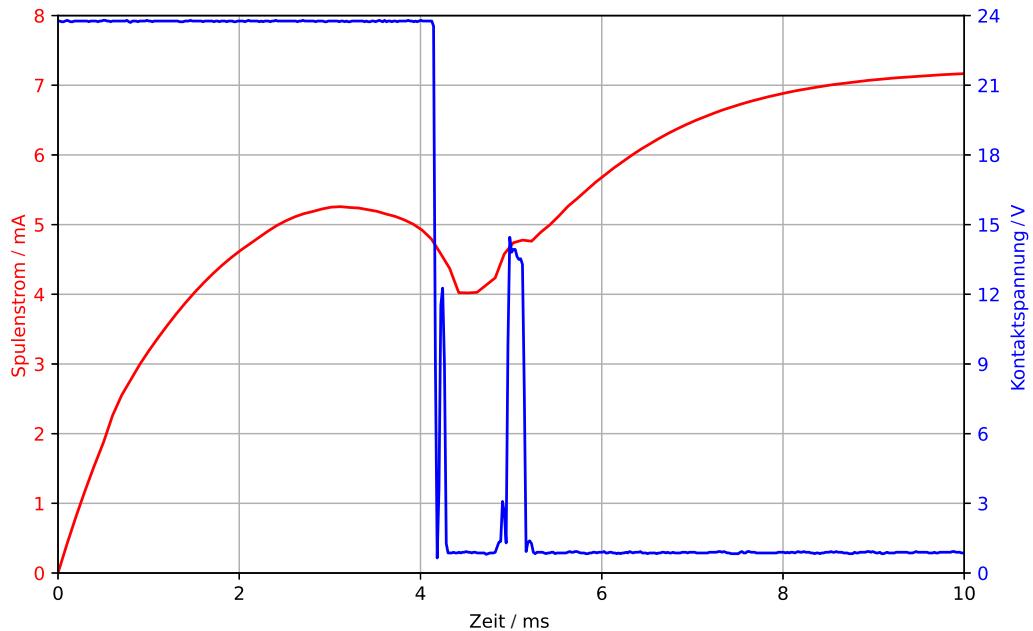


Abb. 2.3: Messkurven beim Einschalten

Nachfolgend wird beschrieben, wie sich der Spulenstrom und die Kontaktspannung beim Ein- und Ausschalten des Relais verhalten. In Abb. 2.3 sind beide Messgrößen für einen Einschaltvorgang dargestellt. Zum Zeitpunkt $t = 0\text{ ms}$ wird die Versorgungsspannung der Spule von 0 V auf einen konstanten Wert (hier 24 V) sprunghaft erhöht, wodurch ein Stromfluss durch die Spule bedingt wird. Dieser folgt aufgrund der – zunächst konstanten – Induktivität L der Spule bis etwa 3 ms der Beziehung:

$$u(t) = L \frac{di(t)}{dt} \quad (2.1)$$

Danach ist eine Abnahme des Stroms zu beobachten. Der Grund hierfür liegt in der Ankerbewegung, mit der eine Änderung der Induktivität einhergeht. Dadurch wird L eine Funktion des Stroms und der Zeit $L = L(t, I)$ und die Lösung von 2.1 verkompliziert:

$$u(t) = \frac{di(t)L(t, I)}{dt} \quad (2.2)$$

Durch die Ankerbewegung verändern sich sowohl örtlich als auch zeitlich die Magnetfelder und es kommt zu einer zusätzlichen Induktionsspannung und somit einem Stromfluss in der Spule, der der Ursache entgegenwirkt und den Gesamtstrom durch die Spule abfallen lässt. Bei etwa $4,5\text{ ms}$ ist ein lokales Minimum des Stroms zu erkennen. Der Anker schlägt hier auf den Spulenkern auf, wodurch die Bewegung und die zusätzliche Induktionsspannung enden. Daher ist anschließend wieder ein Anstieg des Spulenstroms gemäß Gl. 2.1 mit einer konstanten, aber größeren Induktivität zu beobachten, was bei gleicher Spannung einem langsameren Stromanstieg entspricht. Bei etwa 5 ms ist eine leichte „Delle“ zu erkennen, welche wiederum auf das Prellen und der damit einhergehenden Bewegung (Lösen und Aufschlagen) des Ankers zurückzuführen ist.

Für die Kontaktspannung zwischen den Kontakten COM und NO sind zwei Spannungslevel zu beobachten: 24 V entspricht der Versorgungsspannung und liegt an, wenn die Kontakte voneinander getrennt sind. Bei verbundenen Kontakten fällt die Spannung auf ca. 1 V ab. Aufgrund der Messschaltung kann eine Spannung von 0 V nicht erreicht werden (Details in Abs. 3.3.1). Beim Einschalten springt die Kontaktspannung kurz nach 4 ms auf etwa 1 V , die Kontakte berühren sich erstmalig. Darauffolgend sind jedoch mehrfache Sprünge zwischen beiden Levels zu sehen. Der Grund hierfür ist die kinetische Energie der Kontakte, durch die elastische Stöße zwischen den Kontakten hervorgerufen werden. Dieses Phänomen wird auch als Prellen bezeichnet. Kurz nach 5 ms sind keine Spannungssprünge mehr zu beobachten, das Prellen ist beendet und die Kontakte sind geschlossen.

Die Messgrößen während eines Ausschaltvorgangs sind in Abb. 2.4 dargestellt. Der Schaltvorgang startet bei 0 ms mit Abschalten der Spulenspannung, weshalb der Strom direkt abfällt. Durch die Selbstinduktion ist – wie beim Einschalten – keine sprunghafte Änderung zu beobachten. Zwischen ca. 7 und 11 ms ist zu erkennen, dass der Strom kurzfristig wieder ansteigt. Dies ist auf die Ankerbewegung zurückzuführen, durch die die Induktionsspannung und somit der Strom wie oben begründet kurzzeitig ansteigen. Währenddessen werden die

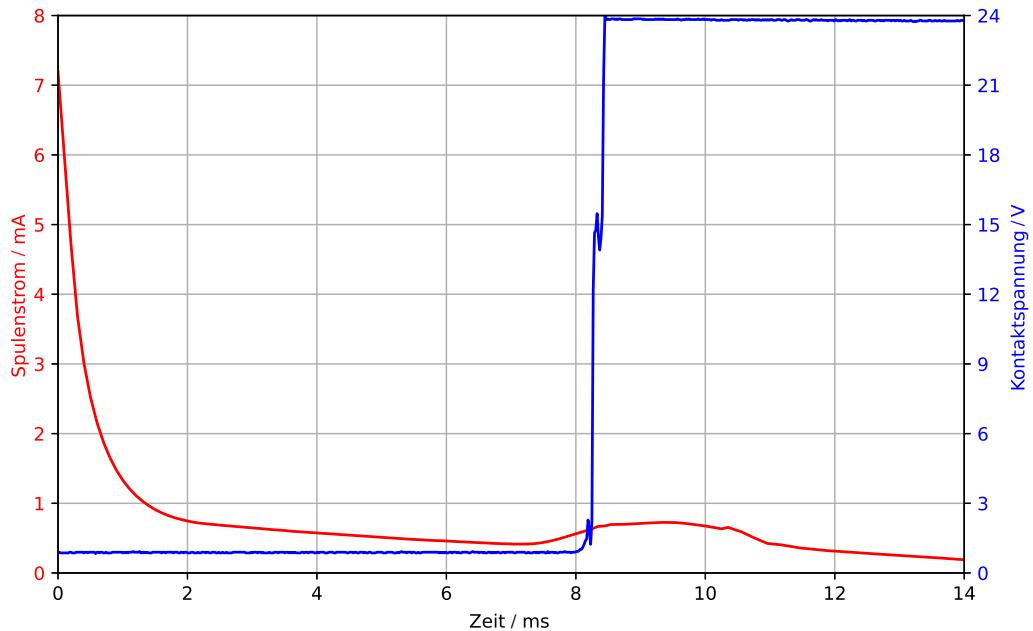


Abb. 2.4: Messkurven beim Ausschalten

Kontakte getrennt, was am sprunghaften Anstieg der Kontaktspannung zu erkennen ist. Die Verzögerung zwischen dem Start der Ankerbewegung und dem Trennen ist auf den Überhub zurückzuführen.

Für ein weitergehendes Verständnis der Funktionsweise von Relais müssen u.a. die mechanischen, elektrischen und magnetischen Zusammenhänge der einzelnen Relaiskomponenten betrachtet werden. Eine derartige Modellierung stellen *Ramirez-Laboreo et al.* in [2] vor. Mit dieser ist eine Simulation von Spulenstrom und Kontaktspannung möglich.

2.2 Degradierung von Relais

Grundsätzlich sind die Ausfälle von Relais in zwei Fehlerarten zu unterscheiden:

- Einschaltfehler: Beim Einschalten werden die Kontakte nicht erfolgreich geschlossen.
- Ausschaltfehler: Beim Ausschalten werden die Kontakte nicht erfolgreich getrennt.

Die möglichen Ursachen für die Fehler sind divers. In [3] werden Phänomene aufgeführt, die an den Kontakten auftreten und deren Eigenschaften verändern können. So ist es möglich, dass sich an den Oberflächen der Kontakte Staub und Partikel aus der Atmosphäre ablagern; sich O₂-Molekülschichten, Polymerschichten, Korrosionsschichtenten oder durch die Zersetzung von Silikonen SiO₂-Schichten ausbilden; durch die hohen Temperaturen eines Lichtbogens Kohlenstoffbeläge entstehen und durch Tangentialbewegungen die Oberfläche

der Kontakte abgeschliffen wird. Diese Phänomene können die Leitfähigkeit der Kontakte maßgeblich beeinflussen.

Die Ursachen hängen jedoch auch vom Relaistypen ab, weshalb an dieser Stelle exemplarisch einige häufige Phänomene vorgestellt werden. Von *Kirschbaum et al.* werden vier Gründe für die Degradierung aufgeführt [4], wobei die meisten zuvor genannten Gründe dem dritten Punkt „Kontaktverunreinigung“ zuzuordnen sind.

1. Erosion

Durch einen elektrischen Lichtbogen wird eine Erwärmung des Kontaktmaterials bis zu dessen Siedepunkt hervorgerufen. Dadurch wird eine Materialmigration von einem Kontakt auf den anderen ermöglicht und gleichzeitig entstehen Krater und Spitzen auf der Kontaktobерfläche. Die Auswirkungen von Lichtbögen hängen u.a. von der Dauer und der Energie des Lichtbogens, der Induktivität des Stromkreises, dem Kontaktmaterial, der Schaltgeschwindigkeit der Kontakte, der Sauberkeit der Kontaktobерfläche und den Kontaktabmessungen ab. Insgesamt führt die Erosion der Kontakte zu einer Verschlechterung der Kontaktobерfläche. Weiterhin wird die Kontaktkraft aufgrund des Materialverlusts verringert, wodurch Kontakte eher verschweißen können.

Der Lichtbogen ist in zwei Phasen zu unterscheiden: Zunächst fließt ein Strom durch verdampfte und anschließend ionisierte Metallatome. Dort kann eine Materialmigration von der Anode zur Kathode beobachtet werden. Gleichzeitig führt der Elektronenbeschuss an der Anode zu einer Zersetzung des Anodenmaterials. Danach nimmt die Dichte des Metalldampfes ab und der Strom fließt über ionisierte Atome aus der Umgebungsluft. Diese prallen auf die Kathode auf und erodieren Material. An der Anode kann beobachtet werden, dass sich die erodierten metallischen Atome an der Kathode anhäufen. Im Allgemeinen ist in Gleichstromkreisen eine Kathodenverstärkung zu beobachten, da die Dauer des Lichtbogens mit Ionen aus der Umgebungsluft kurz im Vergleich zur Dauer mit metallischen Ionen ist. Durch höhere Öffnungsgeschwindigkeiten können Lichtbogendauer und Lichtbogenerosion verringert werden. Gleichzeitig kann dadurch jedoch auch mehr Prellen bei der Kontakttherstellung herbeigeführt werden. Dies wiederum begünstigt Kontaktverschweißen und Erosionseffekte, weil Lichtbögen während des Prellens auftreten.

2. Kontaktverschweißen

Durch Erwärmung kann eine – lokal begrenzte – Schmelze des Kontaktmaterials hervorgerufen werden und damit eine Schweißung der Kontakte. Diese ist kritisch, wenn die Kontakte sich nicht mehr trennen können. Kontaktprellen und / oder Vorzündlichtbögen begünstigen Verschweißen.

3. Kontaktverunreinigung

Zusätzlich können sich isolierende oder halbleitende Schichten auf den Kontaktflächen ablagern, z. B. durch Ausgasen von Kunststoffdichtungen oder Isolationsmaterialien,

Materialabrieb und Verunreinigungen aus der Umgebungsluft. Dadurch kann Stromfluss nur an den Stellen stattfinden, wo die Schichten während der Kontaktierung entweder elektronisch oder mechanisch zerstört werden. Allerdings können halbleitende Schichten auch zu einer Erhöhung der effektiven Kontaktfläche beitragen. Das Ausmaß der Schichtbildung hängt von der Lagerungsdauer, den Umgebungsbedingungen, Betriebskonditionen und Veränderungen der Kontaktflächen durch elektrische Lichtbögen ab.

4. Reibung

Die Kontaktflächen sind anfällig für Reibung, welche zum Verschleiß der Kontakte beiträgt. Grund können externe Vibrationen oder unterschiedliche Wärmeausdehnungsraten der Kontaktmaterialien sein.

Die Komplexität der Mechanismen, die zum Versagen führen, ist hoch, sodass die Versagensursachen nicht immer nachvollziehbar sind. Zum Beispiel können verschweißte Kontakte durch Erschütterungen, die beim Wechseln und Öffnen des Relais zwangsläufig auftreten, getrennt werden.

3 Prädiktive Instandhaltung

Relais werden als Bauteil in diversen Systemen eingesetzt, wodurch sich eine Marktgröße von über US\$ 6 Mrd. im Jahr 2019 ergibt [5]. Durch „Predictive Maintenance“ (dt. Prädiktive Instandhaltung) (PdM) der Relais können mehrerer Aspekte verbessert werden:

- Die Verfügbarkeit der Systeme wird durch bedarfsgerechte Instandhaltung gesteigert. Weiterhin werden ungeplante Stillstände infolge von Ausfällen unterbunden.
- Durch Steigerung der Betriebsdauer wird der Ressourcenverbrauch reduziert.
- Die individuelle Analyse von Relais ermöglicht eine Planung der Instandhaltung. So können die Aufwände für die Instandhaltung reduziert werden.
- Die Reduktion von Ausfällen führt zu einer Steigerung der Sicherheit, da diese eine potenzielle Gefahrenquelle für Menschen, Umwelt und Systeme sind.

Folglich ist PdM eine Schlüsseltechnologie, um ökonomische und ökologische Ziele gleichzeitig zu erreichen. Für die Realisierung von PdM für Relais sind mehrere Forschungsbereiche zu berücksichtigen:

1. Datensatz

Die Grundlage für PdM ist eine Analyse der Degradierung von Relais, die im Rahmen dieser Arbeit vollautomatisch durchgeführt wird. Deshalb wird ein Datensatz benötigt, der Messgrößen über die Betriebsdauer von Relais enthält, mit denen Rückschlüsse auf die Degradierung gezogen werden können. Ein derartiger Datensatz ist bisher nicht öffentlich verfügbar.

Allgemein sind Datensätze für PdM schwer zu generieren, da Komponentenhersteller oft keine Handhabung ihrer Produkte beim Endkunden haben und Systemhersteller nicht den Aufwand betreiben, eigene Algorithmen für ihre eingesetzten Komponenten zu entwickeln. Gleichzeitig verfügen Forschungseinrichtungen in der Regel nicht über größere Systeme, die im Hinblick auf PdM untersucht werden könnten, sodass nur eine begrenzte Anzahl von Datensätzen für die Entwicklung von Algorithmen zur Verfügung steht.

Deshalb wird im Kontext dieser Arbeit ein vollautomatischer Versuchsaufbau und ein Relais-Datensatz für PdM vorgestellt. Letzterer hebt sich hinsichtlich des Umfangs und der Komplexität der Degradierung deutlich vom Stand der Technik ab.

2. Maschinelles Lernverfahren

Die Datensätze werden mit maschinellen Lernverfahren analysiert, für die die Bautechnologien bereits entwickelt wurden. Eine Vielzahl von Veröffentlichungen beinhaltet Lösungen für die PdM von diversen Komponenten. Der Transfer dieser auf Relais ist jedoch nicht möglich. Deshalb wird in dieser Arbeit ein neuer semi-überwachter Ansatz für PdM bei Relais vorgestellt.

3. Applikation auf eingebetteten Systemen

Bei den untersuchten Relais handelt es sich um preiswerte Bauteile, weshalb es nicht möglich ist, zur Analyse der Daten eine kostenintensive Steuerung zu verwenden. Stattdessen müssen die maschinellen Lernverfahren auf günstigen Mikrocontrollern ausgeführt werden. In diesem Forschungsbereich sind bereits Verfahren bekannt, mit denen maschinelle Lernverfahren bezüglich ihrer Effizienz gesteigert werden können – die Auswirkungen dieser Verfahren auf die Performanz bei PdM von Relais sind jedoch nicht bekannt und werden in dieser Arbeit ergründet.

Das nachfolgende Kapitel ist wie folgt aufgebaut: Zunächst werden die Forschungsstände bezüglich der Datensätze, PdM, dem semi-überwachten Lernen sowie Verfahren zur Effizienzsteigerung dargelegt. Darauf aufbauend werden relevante Grundkenntnisse geschildert. Schließlich werden der entwickelte Versuchsaufbau und der generierte Datensatz vorgestellt. Mit diesem wird im Weiteren ein Forschungsdesiderat aufgezeigt, für das nachfolgend ein neuartiger Ansatz zur PdM präsentiert und evaluiert wird. Abgeschlossen wird dieses Kapitel mit Untersuchungen zum Einfluss von Verfahren zur Effizienzsteigerung auf die Performanz. Schlussendlich werden die Ergebnisse in einem Fazit zusammengefasst und kritisch beleuchtet.

3.1 Forschungsstand

Die nachfolgenden Untersuchungen umfassen ein breites Spektrum an Forschungsthemen, weshalb die Darstellung des Forschungsstandes im Nachfolgenden in vier Teile untergliedert ist. Zuerst werden die veröffentlichten Datensätze für PdM betrachtet. Darauf aufbauend werden in Abs. 3.1.2 die bereits entwickelten Verfahren PdM erläutert. Der Fokus liegt auf maschinellen Lernverfahren, wobei dediziert auf Relais eingegangen wird. Als drittes Themengebiet wird das semi-überwachte Lernen vorgestellt, welches die Grundlage für das Verfahren MAUD bildet. Schlussendlich wird als Viertes die Applikation von ANN auf Mikrocontrollern betrachtet. Abschließend werden die Forschungsstände zu den Themengebieten prägnant zusammengefasst.

3.1.1 Datensätze zur prädiktiven Instandhaltung

Der Kern dieses Kapitels stellt die Modellierung der Degradierung von Relais mittels maschineller Lernverfahren dar. Die Grundlage für ein derartiges Vorgehen ist ein Datensatz, mit dem die maschinellen Lernverfahren trainiert werden. Für die Einordnung dieser Arbeit in den Forschungsstand ist eine Betrachtung der bereits veröffentlichten Datensätze unumgänglich. Eine große Herausforderung hierbei ist, dass es kein zentrales Portal gibt, in dem alle Datensätze einheitlich und standardisiert dokumentiert veröffentlicht werden. Deswegen wird im Folgenden zunächst eine umfangreiche Recherche vorgestellt, die bereits in [6] veröffentlicht wurde. Die Datensätze werden dabei nach drei Datentypen unterschieden:

- Synthetische Daten

Diese Datensätze wurden mit Hilfe von Simulationssoftware erstellt, mit der die Degradierung von realen Systemen modelliert wird. Dieser Ansatz ist besonders kostengünstig, da keine Ausrüstung für die Degradierung von Komponenten benötigt wird. Die Herausforderung besteht darin, ein Modell zu erstellen, das die Realität so genau widerspiegelt, dass es für die Simulation der Degradierung eines Bauteils verwendet werden kann.

- Reale Daten

Diese Daten stammen von realen Systemen, weshalb hier ein starker Praxisbezug besteht. Allerdings sind diese Datensätze selten, da die Betreiber selten bereit sind, die Daten zur Verfügung zu stellen – sofern ein System mit der notwendigen Datenerfassung ausgestattet ist. Hinzu kommt ein weiterer Nachteil: Die Anlagen werden gewartet, weshalb einige Komponenten nicht bis zu ihrer maximalen Betriebssdauer betrieben werden und die Daten nicht immer vollständig sind.

- Prüfstanddaten

Die Daten können auch mit Hilfe von Prüfständen erhoben werden, mit denen die Komponenten bis zum Ende ihrer Lebensdauer betrieben werden. Aus diesem Weg können sowohl realistische als auch vollständige Datensätze erfasst werden. Eine Herausforderung ist jedoch die Diversifikation der Betriebsbedingungen, um eine reale Anwendung abzubilden.

3.1.1.1 Übersicht

In Tab. 3.1 sind die recherchierten Datensätze aufgeführt. Einige veröffentlichte Datensätze wurden aufgrund unzureichender Dokumentation. Neben der Referenz und einer kurzen Beschreibung werden zu jedem Datensatz drei weitere Angaben gemacht: Der Datentyp (DT) bezieht sich auf die zuvor beschriebenen Datentypen; die Einheiten (E) auf die Anzahl der Komponenten / Systeme, von denen die Daten erfasst wurden; die Datenpunkte (DP) auf die gesamte Anzahl an Zeitpunkten, zu denen die Messgrößen erfasst wurden. Die Tabelle ist in der eigenen Veröffentlichung [6] enthalten.

Tab. 3.1: Datensatzübersicht [6]

Ref. Beschreibung	DT	E	DP
[7] Kugellager wurden auf einer Welle installiert und mit einer konstanten Drehzahl und Radiallast betrieben. Während des Betriebs wurden regelmäßig Schwingungssignale bei 20 kHz aufgezeichnet.	P	12	7.588
[8] Es wurden Versuche auf einer Fräsmaschine unter 16 verschiedenen Betriebsbedingungen wie Geschwindigkeiten, Vorschübe und Schnitttiefen durchgeführt. Unter jeder Bedingung wurden mehrere Durchläufe aufgezeichnet. Zu den Messgrößen gehören akustische Daten, Vibrationsdaten und Motorströme. Ferner wurde regelmäßig der Verschleiß gemessen.	P	16	167
[9] Li-Ionen-Batterien wurden bei verschiedenen Temperaturen und Betriebsprofilen geladen. Gemessen wurden Ströme, Spannungen und Temperaturen. Ferner wurde die Impedanz regelmäßig bestimmt.	P	34	7.565
[10] Die Degradation von Turbofan-Triebwerken wurde unter verschiedenen Kombinationen von Betriebsbedingungen und Fehlerzuständen simuliert. Es wurden drei Betriebs- und 21 Sensormessungen aufgezeichnet.	S	1.852	340.994
[11] Es wurden Run-to-Failure-Experimente mit Leistungs-MOSFETs unter thermischer Überlastung durchgeführt. Temperaturen, Spannungen und Betriebsdaten wurden aufgezeichnet.	P	42	1.861.959
[12] Es wurden Run-to-Failure-Versuche an Kugellagern unter mechanischer Überlast durchgeführt. Temperaturen und Schwingungen wurden aufgezeichnet.	P	17	27.907
[13] Es wurden Run-to-Failure-Experimente an CFK-Platten unter Zugermüdung durchgeführt. Die Signale von piezoelektrischen Sensoren und Dehnungsmessstreifen wurden als Daten erfasst. Ebenso wurden in regelmäßigen Abständen Röntgenaufnahmen gemacht, um die innere Schädigung zu charakterisieren.	P	12	1.495
[14] Die Batterien wurden mit zufällig erzeugten Stromprofilen zyklisiert. Temperatur, Strom und Spannung wurden aufgezeichnet.	P	28	1.117.894
[15] Es wurden Daten eines kleinen unbemannten Luftfahrzeugs gesammelt, um die verbleibende Flugzeit vorherzusagen. Zu diesem Zweck wurden Motordrehzahl, Stromstärke, Spannung, Temperatur und Steuerbefehle des Flugzeugs und Temperatur, Spannung und Stromstärke der Batterie erfasst.	R	53	26.526.679
[16] Es wurden Daten aus simulierten Experimenten mit Satellitenbatterien gesammelt. Strom, Spannung und Temperatur der Batterien wurden aufgezeichnet.	S	2	1.640
[17] Drei Sätze von 8 Elektrolytkondensatoren wurden kontinuierlich mit einer Frequenz von 100 MHz geladen und entladen. Jeder Satz wurde auf 10, 12 bzw. 14 V aufgeladen. Die Ströme und Spannungen wurden gemessen und in regelmäßigen Abständen wurden Impedanzmessungen durchgeführt.	P	24	1.842.432
[18] Es wurden Drücke, Durchflussmengen, Stromverbrauch, Vibrationen, Wirkungsgrad und Temperaturen eines hydraulischen Prüfstandes erfasst. Das System führte konstante Lastzyklen durch, wobei die Bedingungen für Kühler, Ventil, Pumpe und Speicher variierten.	P	1	2.205
[19] Getriebe wurden unter Belastungen von 0 bis 90 % und zwei Szenarien (heile und gebrochener Zahn) simuliert. Als Messgrößen wurden Schwingungssignale erzeugt.	S	20	2.021.000
[20] Es wurde ein komplettes Produktionssystem aus mehreren Modulen betrachtet: Materialversorgung, Lagerung, Dosierung, Abfüllung und Produktion. Mehr als 100 Messgrößen wurden während des Betriebs aufgezeichnet.	P	1	11.900
[21] Ein Aufzug wurde mit IoT-Sensoren ausgestattet, um PdM für Aufzugstüren zu ermöglichen. So wurden unter anderem Feuchtigkeit, Vibration und Umdrehungen erfasst.	R	1	112.000
[22] Von 20 Verpackungsmaschinen, die in verschiedenen Werken auf der ganzen Welt im Einsatz sind, wurden keine Sensordaten, aber 154 verschiedene Alarncodes aufgezeichnet.	R	20	59.000
[23] Die Degradation einer Maschine mit fünf Ausfallmodi wurde simuliert. Temperaturen, Drehzahlen, Drehmomente und Verschleiß wurden als Betriebsdaten generiert.	S	10.000	10.000
[24] Der Datensatz umfasst Run-to-Failure-Versuche für eine kleine Flotte von Flugzeugtriebwerken unter realistischen Flugbedingungen. Die Modellierung der Schadensausbreitung baut auf der Strategie von [10] auf.	S	9	6.500.000
[25] Es wurde ein Fehlersimulationsmodell eines Dieselmotors entwickelt, um die Variation der Druckkurven in den Zylindern und das Drehschwingungsverhalten der Kurbelwelle zu analysieren. Die Motoren wurden mit einer konstanten Drehzahl betrieben, wobei insgesamt 3.500 verschiedene Fehlerszenarien unter vier Betriebsbedingungen betrachtet wurden.	S	1	3500
[26] Die Abnutzung von Filtern wurde mit einem automatischen Testaufbau untersucht. Es wurden zwei Staubarten verwendet und die Staubzufuhr wurde variiert. Als Messgrößen wurden die Durchflussmenge und die Druckdifferenzen aufgezeichnet.	P	100	78.800
[27] Eine rotierende Maschine wurde entwickelt, um Schwingungssignale unter sieben verschiedenen Betriebsbedingungen zu erfassen.	P	1	2.162

DT – Datentyp (P – Prüfstand; S – Synthetisch; R – Real) E – Einheiten DP – Datenpunkte

Insgesamt sind 26 Datensätze aufgeführt, die innerhalb von 15 Jahren veröffentlicht wurden. Es ist auffällig, dass die Anzahl der Datenpunkte stark variiert. Dies ist mit den unterschiedlichen Messaufbauten und Komponenten und den damit einhergehenden Messgrößen zu begründen. Hier wird deutlich, dass die Entwicklung von allgemeinen Verfahren für PdM durch die hohe Individualität der Datensätze erschwert wird. Dies wird untermauert durch die großen Differenzen in Bezug auf die Einheiten: Zwischen einer bis 10.000 Einheiten sind die Datensätze einzuordnen. Ein weiteres Differenzierungsmerkmal stellt der Datentyp dar, der einen erheblichen Einfluss auf die Generalisierbarkeit der Verfahren hat. Die synthetischen Datensätze können bspw. nur bedingt für die Entwicklung von realen PdM-Algorithmen verwendet werden, da diese durch das Simulationsmodell begrenzt sind. Dieses Problem stellt sich bei den realen Datensätzen nicht; allerdings haben diese jeweils weniger als 55 Einheiten, wodurch die Generalisierungsfähigkeit der Verfahren nicht umfangreich geprüft werden kann. Gleiches gilt für die Prüfstanddatensätze: Lediglich [26] sticht mit einer Anzahl von 100 Einheiten heraus, wobei hier die Datenmenge und damit die Komplexität begrenzt zu sein scheint. Ein Vergleich ist aufgrund der Heterogenität der betrachteten Komponenten und Systeme jedoch schwer. Deshalb wird an dieser Stelle lediglich exemplarisch auf Datensätze eingegangen:

- Turbofan Engine Degradation Simulation Data Set[10]:

Dieser Datensatz enthält simulierte Degradierungsdaten von Mantelstromtriebwerken und ist – gemessen an der Anzahl der Downloads und Veröffentlichungen – einer der meistgenutzten Datensätze. Die Beliebtheit ist u.a. auf das Simulationsmodell zurückzuführen, bei dem verschiedene Degradierungstypen und Einsatzbedingungen abgebildet worden sind. Dadurch eignet sich der Datensatz für die Applikation von maschinellen Lernverfahren. Für Verfahren, die im industriellen Kontext angewendet werden sollen, eignet sich dieser Datensatz jedoch nur bedingt, da er simuliert ist und die Daten stets monotone Trends aufweisen. Zudem werden zu den Datenpunkten nur Merkmale angegeben, Untersuchungen zur Verwendung von Messreihen sind daher nicht möglich.

- Capacitor Electrical Stress Data Set[17]:

Der Datensatz wurde mit einem Prüfstand aufgezeichnet und enthält zu jedem Datenpunkt Messreihen. Dadurch könnten neue Verfahren entwickelt werden – hier ist der Umfang des Datensatzes jedoch limitierend. Es wurden insgesamt 24 Kondensatoren getestet, weshalb die Generalisierungsfähigkeit von Verfahren lediglich mit einer begrenzten Stichprobe getestet werden kann.

- COMFAULDA[27]:

Bei COMFAULDA wurde der Datensatz mit nur einem Prüfstand / System aufgezeichnet. Ziel war es, die Fehlerarten im Datensatz zu diversifizieren. Dazu wurden verschiedene Fehler und Betriebsbedingungen herbeigeführt. Somit bietet dieser Datensatz hinsichtlich der Fehlerarten ein neues Komplexitätslevel. Da jedoch nur ein Prüfstand genutzt wurde, sind Aussagen über die Generalisierungsfähigkeit von Verfahren nicht möglich.

Zusammenfassend ist bezüglich der Komplexität festzuhalten, dass ein Entwicklungspotenzial hinsichtlich des Umfangs der aufgezeichneten Messgrößen (Messreihen statt Merkmale) und der Fehlerbilder auszumachen ist. Einzelne Datensätze adressieren diese Aspekte; jedoch besteht weiterhin Bedarf an komplexeren Datensätzen.

3.1.1.2 Visualisierung

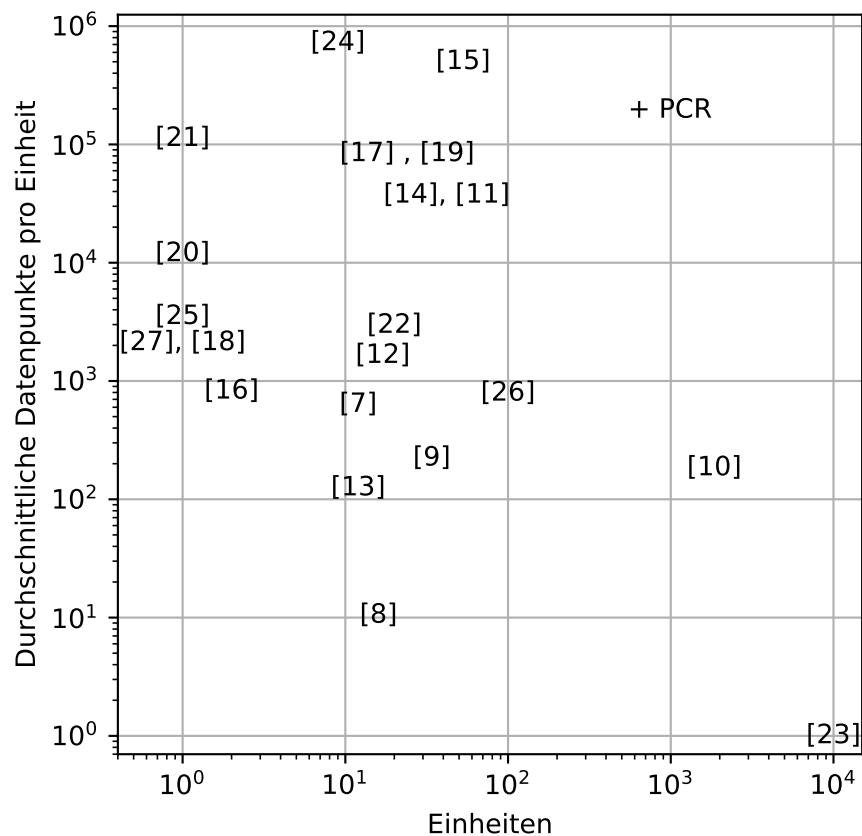


Abb. 3.1: Visualisierung der verfügbaren Datensätze in Bezug auf die Anzahl der Einheiten und Datenpunkte pro Einheit [6]

Unter der Annahme, dass die Messwerte Rückschlüsse auf den Zustand der Bauteile zulassen, sind zwei Faktoren für die Konzeption, Entwicklung und Bewertung von PdM-Algorithmen wichtig:

1. Die Anzahl der untersuchten Einheiten: Je mehr Einheiten zum Testen des Algorithmus verwendet werden, desto besser kann dessen Funktionalität überprüft werden.

2. Die durchschnittlich verfügbaren Datenpunkte pro Einheit: Eine genaue Beurteilung des Zustands ist nur bei ausreichend hoher Anzahl von Datenpunkten möglich. Verändert sich etwa die Kontaktfläche eines Relais innerhalb von wenigen 100 Schaltzyklen, müssen die Betreiber frühzeitig Rückschlüsse darauf ziehen können.

Angesichts dessen wurden die Datensätze hinsichtlich dieser beiden Faktoren in Abb. 3.1 dargestellt. Die Achsen sind logarithmisch skaliert und die einzelnen Datensätze mit der jeweiligen Referenz markiert. Weiterhin wurde der im Rahmen dieser Arbeit generierte Datensatz „Phoenix Contact Relay-Dataset“ (PCR) eingefügt [28]. Es ist deutlich zu erkennen, dass dieser heraussticht: Lediglich [10] und [23] enthalten mehr Einheiten und [24] und [15] mehr Datenpunkte pro Einheit. Daraus ergibt sich ein Desiderat in Bezug auf die verfügbaren Datensätze. Abgesehen von PCR umfasst kein Datensatz sowohl eine große Menge an Einheiten als auch viele Datenpunkte pro Einheit.

3.1.2 Verfahren zur prädiktiven Instandhaltung

Der Forschungsstand zu PdM wird in zwei Teile untergliedert. Zunächst wird generell auf PdM mittels maschineller Lernverfahren eingegangen. Im Anschluss werden Verfahren zur Instandhaltung von Relais vorgestellt.

3.1.2.1 Prädiktive Instandhaltung mittels maschineller Lernverfahren

Durch PdM kann sowohl ein ökonomischer als auch ein ökologischer Nutzen gestiftet werden. Diese Vorteile führen zu einer hohen Anzahl an Forschungsvorhaben zu dieser Thematik, die insbesondere durch Unternehmen angetrieben werden. Ein Überblick über den Forschungsstand wird über die Vorstellung mehrerer Übersichtsarbeiten gegeben. Anschließend wird im Speziellen auf zwei Methodiken eingegangen, die häufig in ähnlicher Weise genutzt und im Kontext dieser Arbeit auf Relais übertragen werden sollen.

Übersicht

Lei et al. präsentieren in [29] eine vier Schritte umfassende Systematisierung der unterschiedlichen Vorgehensweisen aus über 270 wissenschaftlichen Arbeiten:

1. Datenakquise

Der erste Schritt ist die Datenakquise. Dafür können verschiedene Sensoren verwendet werden, bspw. Beschleunigungssensoren, akustische Sensoren, Thermosensoren und Stromsensoren. Mit diesen müssen Messwerte über die gesamte Betriebsdauer aufgezeichnet werden.

2. Erfassen von Zustandsindikatoren

Darauf folgt im zweiten Schritt die Extraktion von Merkmalen aus den jeweiligen Messdaten. Ein Beispiel hierfür sind physikalische Merkmale, wie die maximale Spannung einer Zeitreihe.

3. Definition von Degradierungsstufen

Maschinen degradieren nicht immer linear. Deshalb wird die Betriebsdauer im dritten Schritt in verschiedene Degradierungsstufen unterteilt. Dadurch ist es z.B. möglich, eine Lebensphase mit geringen Änderungen in den Merkmalen als eine eigene Degradierungsstufe zu definieren.

4. Schätzung der „Remaining useful lifetime“ (dt. nutzbare Restlebensdauer) (RUL)

Schlussendlich wird im vierten Schritt die RUL geschätzt. Dies kann durch verschiedene Verfahren erfolgen, die *Lei et al.* in vier Kategorien unterschieden:

(a) Statistische Modellierungen

Im Rahmen einer statistischen Modellierung wird die Wahrscheinlichkeitsverteilung eines Ausfalls über die Betriebsdauer des Bauteils bestimmt. Dazu sind empirische Daten notwendig. Die Mehrheit der von *Lei et al.* betrachteten Arbeiten sind dieser Kategorie zuzuordnen.

(b) Physikalische Modellierungen

Durch eine physikalische Modellierung können Rückschlüsse auf die Degradierung und damit auf die RUL eines Bauteils gezogen werden. Dazu ist ein detailliertes Verständnis der Degradierungsmechanismen Voraussetzung, damit eine ebenso detailliertere Modellierung erfolgen kann. Als Beispiel ist die Paris-Erdogan-Gleichung aufzuführen, mit der das Wachstum von Rissen beschrieben wird [30].

(c) Maschinelle Lernverfahren

Durch maschinelle Lernverfahren ist es möglich, komplexe, nicht lineare Zusammenhänge abzubilden. Deshalb eignen sich diese Verfahren, um eine Assoziation zwischen den extrahierten Merkmalen und der RUL der Bauteile zu bilden. Der Vorteil gegenüber der physikalischen Modellierung liegt darin, dass kein detailliertes Verständnis des Bauteils Voraussetzung ist. Stattdessen basiert die Modellierung auf dem Training der maschinellen Lernverfahren mit den Messdaten.

(d) Hybride

Weiterhin ergibt sich die Möglichkeit, Verfahren der bereits genannten Kategorien zu kombinieren, wodurch ein Hybrid entsteht. So können die jeweiligen Vorteile genutzt werden. Die Fusionsstrategien dafür sind unterschiedlich.

In [31] werden durch *Ran et al.* drei Instandhaltungsstrategien voneinander unterschieden: Bei der reaktiven Instandhaltung werden die Komponenten bis zu ihrem Ausfall betrieben, erst danach findet eine Instandhaltung statt. Diese Strategie ist dann empfehlenswert, wenn die Kosten der Instandhaltung die Kosten durch einen Fehler deutlich übersteigen. Eine weitere Strategie ist die präventive Instandhaltung. Hier sind die Kosten für einen Fehler so hoch, dass die Instandhaltung in regelmäßigen Intervallen durchgeführt wird,

sodass das Auftreten von Fehlern reduziert wird. Eine typische Umsetzung dieser Instandhaltungsstrategien ist die statistische Modellierung nach *Lei et al.* Abschließend wird als dritte Strategie PdM aufgeführt. Diese ist empfehlenswert, wenn die Kosten für Fehler und eine Instandhaltung ähnlich sind. Durch individuelle Bewertung der Komponenten werden Rückschlüsse auf deren Zustand gezogen. Auf diesem Weg wird eine Instandhaltung kurz vor dem Fehler ermöglicht. Diese Strategie wird auch konditionsbasierte Instandhaltung genannt.

An dieser Stelle tritt eine erste Unstimmigkeit in mehreren Arbeiten auf, die den Forschungsstand zu PdM zusammenfassen: In [32] werden PdM und die konditionsbasierte Instandhaltung klar voneinander getrennt, wobei letztere als Erweiterung von PdM mit automatischer Auslösung der Instandhaltung definiert wird. *Cinar et al.* unterscheiden ebenfalls zwischen der prädiktiven und konditionsbasierten Instandhaltung [33]. Hier wird mit der konditionsbasierten Instandhaltung jedoch eine konstante Überwachung der Komponenten bezeichnet, auf die PdM durch Prädikationen aufbaut. Bei der Betrachtung weiterer Arbeiten werden die Unstimmigkeiten in der Begriffsdefinition zunehmend größer, was auf die aktuell hohe Beliebtheit des Forschungsthemas PdM zurückzuführen ist.

Ran et al. schlagen eine Taxonomie zur Einordnung der Forschungsarbeiten zum Thema PdM vor [31]. Auf oberster Ebene wird zunächst zwischen der Systemarchitektur, dem Zweck und Ansätzen unterschieden. Letztere sind im Kontext dieser Arbeit relevant, weshalb nicht weiter auf die Systemarchitektur und den Zweck eingegangen wird. Die Ansätze werden in drei Kategorien unterteilt:

- Basierend auf Wissen

Bei diesen Ansätzen werden Expertenwissen und deduktive Schlussfolgerungen genutzt, um eine Fehlerdiagnose bzw. -prognose zu realisieren. Dementsprechend ist ein tiefgehendes und umfassendes Wissen notwendig, was für die komplexen Phänomene, die in realen Anwendungen auftreten können, nicht immer erfüllt werden kann. Im Kontext dieser Arbeit soll dieses Wissen durch ein maschinelles Lernverfahren modelliert werden. Daher sind wissensbasierte Ansätze nicht relevant.

- Basierend auf traditionellen maschinellen Lernverfahren

Ziel dieser Ansätze ist es, Wissen über eine Komponente durch ein maschinelles Lernverfahren zu erlernen. Die traditionellen Lernverfahren umfassen ANN, Support Vektor Maschinen, Entscheidungsbäume und die Nächste-Nachbarn-Klassifikation. Eine typische Anwendung ist die Schätzung der RUL.

- Basierend auf komplexen ANN

Durch komplexe „Artificial neural network“ (dt. künstliches neuronales Netz) (ANN) kann Wissen selbstständig erlernt werden. Ein „Autoencoder“ (AE) kann bspw. genutzt werden, um eine Encodierung und Decodierung von Eingangsdaten zu erlernen (siehe Abs. 3.2.4). Damit können die Daten in eine latente Darstellung transformiert werden. Im Kontext von PdM kann ein Autoencoder zu zwei Zwecken verwendet werden: Zum einen kann der Rekonstruktionsfehler genutzt werden, um Anomalien zu

erkennen. Zum anderen kann durch die latente Darstellung die Merkmalsextraktion ersetzt werden, sodass Regressions- und Klassifikationsaufgaben erledigt werden können. Mit „Convolutional neural network“ (dt. faltendes neuronales Netz) (CNN) (vgl. Abs. 3.2.3.2) sind ähnliche vollautomatische Merkmalsextraktionen möglich. Weiterhin können „Recurrent neural network“ (dt. rekurrentes neuronales Netz) (RNN) genutzt werden. Diese zeichnen sich dadurch aus, dass eine Art Gedächtnis implementiert wurde. Für eine Schätzung kann dementsprechend die Historie einbezogen werden. Durch „Generative adversarial networks“ (dt. erzeugende gegnerische Netzwerke) (GAN) können synthetische Daten erzeugt werden, indem zwei ANN gegeneinander trainiert werden. Eine ANN muss möglichst realitätsnahe synthetische Daten generieren und das andere muss diese von realen Daten unterscheiden. So ist es möglich, Datensätze auszubalancieren. Weiterhin eignen sie sich, um Anomalien zu erkennen. Daneben präsentieren *Ran et al.* Ansätze, bei denen Transferlernen genutzt wird, um das Wissen von einem auf ein anderes System zu übertragen. Schlussendlich werden Verfahren aufgeführt, bei denen „Deep Reinforcement Learning“ genutzt wird. D.h. ein System wird von einem neuronalen Netz beobachtet und gesteuert. Damit kann das Management der Instandhaltung automatisiert bzw. optimiert werden.

Serradilla et al. gehen in [34] weiter auf die Ansätze ein, die auf komplexen ANN basieren. Dabei werden diese in ein mehrschrittiges Modell eingeordnet. Im ersten Schritt werden Anomalien in Zeitreihen erkannt. Dazu empfehlen sich – in Abhängigkeit zu den verfügbaren Labels – unterschiedliche Verfahren. Wenn die Daten gelabelt sind, dann kann eine Klassifikation durchgeführt werden. Wenn nur „heile“ Daten gelabelt sind, dann kann bspw. der Rekonstruktionsfehler eines Autoencoders zur Erkennung von „nicht heilen“ Daten genutzt werden. Sind die Daten nicht oder nur teilweise als „heile“ gelabelt, dann können ebenfalls Autoencoder genutzt werden. Ebenso empfehlen *Serradilla et al.* auch Cluster-Algorithmen, RNN zur Regression und generative Verfahren wie GAN. Im zweiten Schritt findet eine Diagnose statt, in der der Grund für den Fehler bestimmt wird. Dazu können – sofern sie vorliegen – direkt die Klassifikationsergebnisse aus der Anomalie Erkennung genutzt werden. Häufig ist die Art der Anomalie jedoch nicht bekannt, weshalb über Vergleiche Gruppen von Anomalien gebildet werden können, die dann einen Fehlergrund darstellen. Im dritten Schritt wird eine Prognose durchgeführt. Diese besteht typischerweise aus der Schätzung der RUL. Final werden im vierten Schritt Handlungen ausgeführt, durch die eine Minderung der Fehlerauswirkungen erzielt wird. Dieser Schritt entspricht dem von *Ran et al.* beschriebenen Verfahren „Deep Reinforcement Learning“.

Die zuvor aufgezeigten Übersichtsarbeiten verdeutlichen, dass im Kontext von PdM aktuell mit einer so hohen Geschwindigkeit Fortschritte erzielt und neue Subthemengebiete erschlossen werden, dass der Stand der Forschung nur als Momentaufnahme festzuhalten ist. Im Rahmen dieser Arbeit werden deshalb exemplarische Arbeiten zu zwei übergeordneten Anwendungsbereichen von maschinellen Lernverfahren im Kontext der prädiktiven Instandhaltung aufgeführt: Zum einen die Schätzung der Restlebensdauer, auf die in allen Übersichtsarbeiten eingegangen wurde. Zum anderen die Erkennung von Anomalien, die aufgrund neuer unüberwachter Lernverfahren aktuell beliebt ist.

Nutzbare Restlebensdauer

Die Schätzung der „Remaining useful lifetime“ (dt. nutzbare Restlebensdauer) (RUL) ist ein zentraler Inhalt von diversen Arbeiten zum Thema Prädiktive Instandhaltung. Dies ist u.a. auf den Datensatz „Turbofan Engine Degradation Simulation Data Set“ [10] zurückzuführen, welcher von der NASA im Rahmen der ersten Konferenz für Prognostik und Verschleißmanagement veröffentlicht wurde. Teil der Konferenz war die PHM08-Challenge, bei der die Teilnehmer mit beliebigen Verfahren die RUL von Triebwerken schätzen sollten.

In [35] wird ein Lösungsansatz für die PHM08-Challenge vorgestellt, bei dem die Ausfallphysik ohne Annahmen oder technisches / physikalisches Fachwissen modelliert wird. Dazu wird eine Regression durchgeführt, für die zunächst ein „Multilayer Perceptron“ (dt. mehrschichtiges Perzepton) (MLP) verwendet wird. Dabei werden für das Training die Messdaten der Betriebszyklen als Eingangs- und die RUL als Ausgangsdaten verwendet. Da zu Beginn der Betriebsdauer nur geringe Änderungen in den Daten auszumachen sind, wird die RUL auf maximal 130 begrenzt.

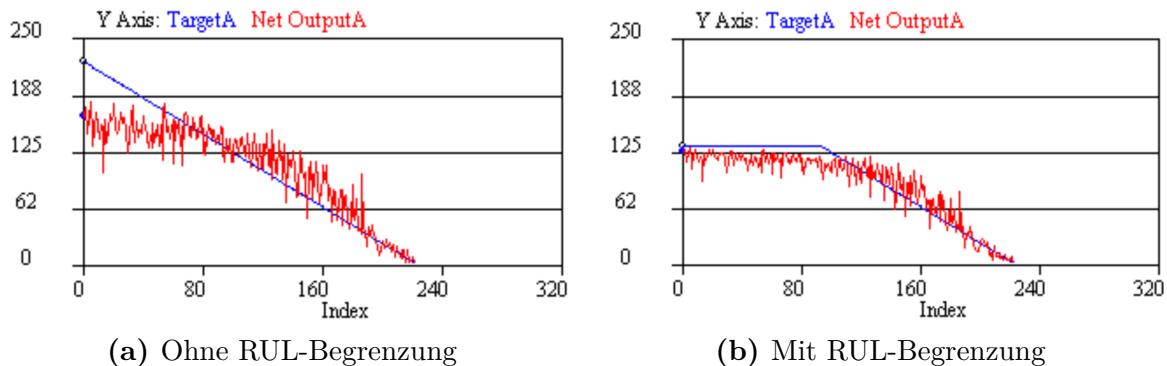


Abb. 3.2: RUL-Schätzungen mittels MLP [35]

In Abb. 3.2 ist die Verbesserung durch die Begrenzung der RUL zu erkennen: Links wurde ohne und rechts mit Begrenzung trainiert. Die Prädiktionen sind rot und die Zielgrößen blau dargestellt. Die Abweichung mit Begrenzung ist – besonders zu Beginn der Betriebsdauer – geringer. In beiden Fällen ist jedoch eine starke Volatilität der Schätzungen zu beobachten. Dies ist auf die verrauschten Sensordaten zurückzuführen. Um dieses Problem zu beheben, wird das MLP durch ein RNN ersetzt. Diese Art von ANN hat eine Gedächtnisfunktion und kann die Historie der Eingangsdaten für eine Prädiktion berücksichtigen. Auf diesem Weg ist es möglich, das Rauschen zu unterdrücken. Eine weitere Verbesserung der Prädiktionen wird durch den Einsatz eines Ensembles von RNN erzielt. D.h. mehrere RNN werden trainiert und von deren Prädiktionen wird der Mittelwert als endgültige Prädiktion genutzt.

In Abb. 3.3 wird ersichtlich, dass die Volatilität der RUL-Schätzungen durch den Einsatz eines RNN-Ensembles reduziert wird. In der Grafik sind dieselben Eingangsdaten wie in

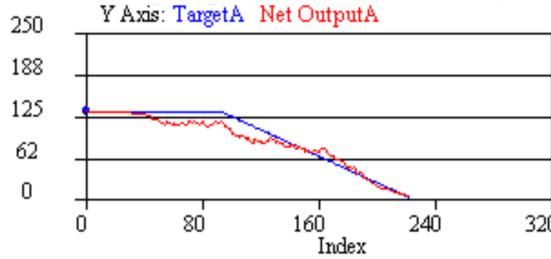


Abb. 3.3: RUL-Schätzung eines RNN [35]

Abb. 3.2 verwendet worden. Mit diesem Verfahren erregte *Heimes* viel Aufmerksamkeit, da es ihm gelang, die RUL mit einem ANN zu schätzen und den zweiten Platz in der Herausforderung zu belegen. Die Veröffentlichung stammt aus dem Jahr 2008. Seitdem wurden viele neue Verfahren zur Schätzung der RUL präsentiert, die Grundgedanken sind jedoch ähnlich.

Kim und Sohn präsentierten einen CNN-basierten „Multi-Target-Learning“-Ansatz, bei der die Regression der RUL gleichzeitig mit einer Klassifizierung des Gesundheitszustandes erlernt wird [36]. Die Gesundheitszustände wurden vor dem Training manuell definiert. Durch das gleichzeitige Lernen von zwei Zielgrößen sollen deren Interdependenzen genutzt werden, um allgemeine Merkmale in dem gemeinsamen Netzwerk zu extrahieren und somit die Performanz zu verbessern. Im Vergleich zu anderen Verfahren konnte eine hervorragende Leistung erzielt werden – bei gleichzeitig verringelter Rechenkomplexität im Vergleich zu RNN-basierten Ansätzen.

Song et al. schlagen in [37] ein Ensemble-Framework vor, das sowohl CNN als auch RNN kombiniert, um die RUL unter Berücksichtigung mehrerer Zeitfenster genau vorherzusagen. Dazu werden in der Trainingsphase mehrere Basismodelle mit unterschiedlich großen Zeitfenstern trainiert, um verschiedene zeitliche Abhängigkeiten zwischen Merkmalen zu erfassen. Dadurch wird der Trainingsfehler gegenüber herkömmlichen Ansätzen mit statischen Zeitfenstergrößen reduziert. In der Testphase werden die Basismodelle entsprechend der Länge der Laufzeit angewendet und die Ergebnisse mit einer gewichteten Durchschnittsmethode zusammengefasst. Ein ähnliches Verfahren stellen *Zhang et al.* in [38] vor: Sie nutzen ebenfalls eine Kombination aus CNN und RNN, um die RUL von Lithium-Ionen-Batterien zu schätzen.

Die Verwendung von maschinellen Lernverfahren zur Schätzung der RUL ist für diverse Datensätze bereits etabliert. Ein weiterer häufig verwendeteter Datensatz ist der FEMTO-Lagerdatensatz [12], der aus 17 Run-to-Failure-Daten von Kugellagern besteht. Für diesen Datensatz wurden verschiedene Ansätze entwickelt, in [39] und [40] werden Support Vektor Maschinen zur Schätzung der RUL verwendet, in [41] wird die RUL durch ein ANN prädiert. Grundsätzlich ähneln sich die Vorgehensweisen und unterscheiden sich zumeist nur in Details der Topologie, der verwendeten ANN oder zusätzlichen Datenverarbeitungsschritten. Besonders letzteres ist wesentlich abhängig vom verwendeten Datensatz.

Anomalie Erkennung

Die RUL muss für das Training bekannt sein, was wiederum Lebensdauertests voraussetzt. In realen Anwendungen stellt dies eine Herausforderung dar, denn nicht jedes System kann bis zum Ausfall betrieben werden. Zudem ergeben sich in komplexeren Systemen eine Vielzahl von möglichen Ausfallmodi. Deshalb wurden Verfahren entwickelt, mit denen Anomalien in den Betriebsdaten eines Systems erkannt werden können. Im Fokus aktueller Arbeiten stehen oft Autoencoder, mit denen bedeutende Fortschritte bei der Erkennung von Anomalien erzielt wurden.

Ein Beispiel für eine erfolgreiche Anwendung eines Autoencoders ist in [42] beschrieben. Dort wurde ein Autoencoder nur mit korrekten Arbeitsmaschinendaten von Produktionspressen trainiert. So ist eine Überwachung der Produktionspresse möglich und Maschinenausfälle können vermieden werden. Dabei übertrifft der Autoencoder statistische halb-überwachte Anomalie-Erkennungsmodelle sowie traditionelle maschinelle Lernmodelle. Zudem wurde für Diagnosezwecke der Encoder-Teil des Autoencoders verwendet, um einen Merkmalsvektor zu erzeugen, durch den – unter Verwendung weiterer Verfahren – eine Isolierung von verschiedenen Fehlertypen möglich war. Überdies wurde ein visuelles Diagnosewerkzeug entwickelt, das anomale Signale hervorhebt, damit Fachleute bei der Anwendung ihres Fachwissens unterstützt werden können.

Alternativ kann ein ANN zur Erkennung von Anomalien verwendet werden, indem die Abweichung von Prädikationen und Messwerten betrachtet werden. Analog zu Autoencodern werden Anomalien mit einem Schwellwert, der sich auf die Abweichung bezieht, erkannt. Beispiele hierfür sind das in [43] von *De Benedetti et al.* genutzte ANN, mit dem Anomalien von Photovoltaik-Anlagen erkannt werden oder das in [44] verwendete ANN zur Identifizierung von Anomalien hinsichtlich des Energieverbrauchs von Drucklifterzeugungssystemen.

3.1.2.2 Instandhaltung von elektromechanischen Relais

Zum Stand der Forschung / Technik bezüglich der Instandhaltung von elektromechanischen Relais wird zunächst erörtert, wie die Instandhaltung von Relais aktuell in der Industrie geplant wird. Dazu wird auf eine statistische Modellierung eingegangen, mit der die Kunden die Wartungsintervalle der Relais anpassen können. Darauf aufbauend wird eine Lösung zur Instandhaltung von Relais vorgestellt, die zur Patentierung angemeldet wurde. Im Weiteren wird auf Publikationen eingegangen, bei denen die Applikation von ANN im Kontext von PdM vorgestellt wird. Schlussendlich werden die Ergebnisse meiner nicht veröffentlichten Masterarbeit [45] erläutert, die in Kooperation mit Phoenix Contact erstellt wurde und an diese Arbeit anknüpft.

Doch zuvor wird kurz auf das Projekt „Digitaler Zwilling“ von Phoenix Contact und CADFEM eingegangen, da es bei der Recherche nach Instandhaltung für Relais unumgänglich ist. Das Projekt ist als physikalische Modellierung nach *Lei et al.* einzuordnen: Während des Betriebs werden Messdaten eines Relais dazu verwendet, einen virtuellen Klon des Relais zu simulieren. Beim Klon handelt es sich um ein transientes Modell, das in

drei Submodelle unterteilt ist. Mit einem Modell werden die mechanischen Komponenten, wie Anker und Feder des Relais, simuliert, mit einem der Magnetkreis und die Stromquelle und mit einem der Lichtbogen. So können aus den Messdaten Zustandsgrößen, wie die Lichtbogenenergie, bestimmt werden. Mit Analyseprogrammen (die nicht weiter erläutert wurden) kann schlussendlich die RUL geschätzt werden [46]. Aus dem Projekt geht keine detaillierte Veröffentlichung hervor, weshalb es im Rahmen dieser Arbeit nicht weiter beachtet wird.

Statistische Modellierung

Bei einer statistischen Modellierung wird eine Ausfallverteilungsfunktion bestimmt, die die Ausfallwahrscheinlichkeit über die Betriebsdauer von Bauteilen angibt. Im Kontext von Relais könnte mit einer solchen Funktion bspw. die Ausfallwahrscheinlichkeit bei einer Betriebsdauer von 50.000 Schaltzyklen angegeben werden.

Grundlegend für die Modellierung ist die Wahl der Wahrscheinlichkeitsverteilung. Bei elektromechanischen Bauteilen wird häufig die Weibull-Verteilungsfunktion verwendet [47]. Diese ist wie folgt definiert [48]:

$$F(x) = 1 - e^{-(\theta x)^k} \quad \text{mit} \quad x, \theta, k \in \mathbb{R} \quad x \geq 0, \quad k > 0, \quad \theta > 0. \quad (3.1)$$

Die Funktion kann durch zwei Parameter an das jeweilige Bauteil angepasst werden: k ist der Formparameter und θ der Skalenparameter. Für die Anpassung sind Informationen zur Verteilung der Lebensdauern notwendig. Die Weibull-Verteilung wird im Kontext von Relais genutzt, um die Ausfallverteilungsfunktion zu schätzen. Mit dieser wiederum wird der B10-Wert bestimmt, welcher die Betriebsdauer angibt, zu der 10 % der Relais wahrscheinlich ausgefallen sind. Anwender können so die Wartungsintervalle planen.

Patente

In [49] und [50] werden Systeme beschrieben, mit denen der Verschleiß von elektrischen Schalteinheiten diagnostiziert werden kann. Die beschriebene Methodik ist als Ähnlichkeitsvergleich einzuordnen. Als Messgröße wird die Spulenspannung genutzt, da diese sich über die Betriebsdauer verändert.

Aus den Zeitreihen der Spulenspannung beim Ausschalten des Relais werden Merkmale extrahiert, die in Abb. 3.4 markiert sind. Die Systeme beruhen auf mehreren Vergleichen, zum einen werden die Merkmale mit Referenzwerten verglichen und zum anderen die Messreihen mit Referenzmessreihen miteinander. Die Referenzwerte werden vor Inbetriebnahme des Systems zu zwei Verschleißzuständen (*neu* und *verschlissen*) aufgezeichnet. Die Werte für *neu* werden für jedes Relais individuell bestimmt: Wenn im neuwertigen Zustand die Messreihe stabil ist, d.h. keine Änderung über mehrere Schaltzustände auszumachen ist, dann wird diese abgespeichert. So wird sichergestellt, dass kein Ausreißer als Referenz gewählt wird. Die Referenzwerte für den Zustand *verschlissen* können nicht individuell erfasst werden, deshalb werden Daten von bereits degradierten Relais herangezogen. Von

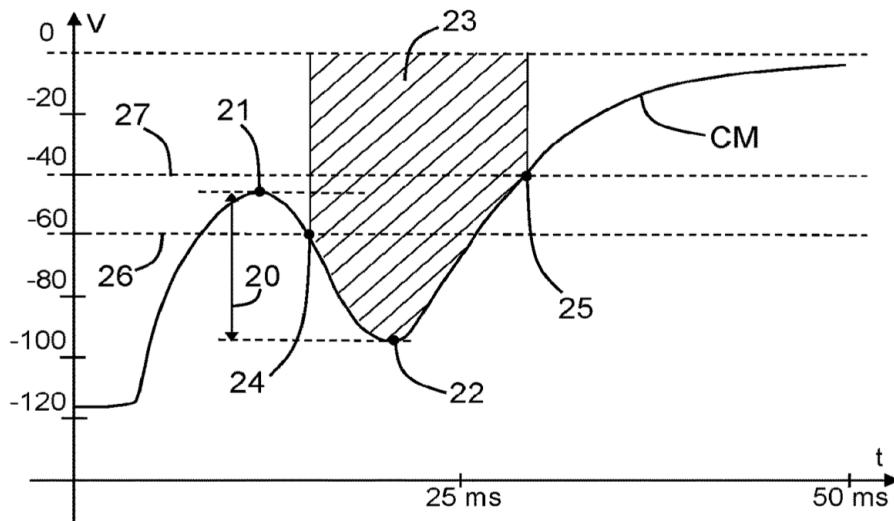


Abb. 3.4: Spannungsverlauf der Steuerseite beim Öffnen des Kontakts [49]

diesen werden ebenfalls Merkmale extrahiert, wobei die Verläufe über die Betriebsdauer durch Polynome ersten Grades approximiert werden. Weiterhin wird eine für den Zustand *verschlissen* repräsentative Messkurve gespeichert.

Über Vergleiche zu den Referenzwerten wird im Betrieb der Verschleiß der Relais klassifiziert, sodass u.a. neue, gebrauchte und verschlissene Relais unterschieden werden können.

Maschinelle Lernverfahren

Die Anwendung von maschinellen Lernverfahren zur prädiktiven Instandhaltung von Relais wurde bereits mehrfach untersucht. Nachfolgend werden die Arbeiten chronologisch aufgeführt:

Phoenix Contact unterstützt seit 2018 Forschungsvorhaben zur Alterungsanalyse von elektromechanischen Relais mit maschinellen Lernverfahren. Im Rahmen einer Studienarbeit wurden Kontaktspannung und Spulenstrom von 17 Relais bis zu deren Ausfall aufgezeichnet [51]. Diese Daten werden genutzt, um ein CNN für eine Klassifikation zu trainieren. Dafür werden die Daten in zehn Klassen (die jeweils ein zehn Prozent großes Restlebensdauerintervall umfassen) unterteilt. Mit dieser Methodik kann gezeigt werden, dass eine Unterscheidung der Degradierungsstufen mittels maschinellen Lernverfahren möglich ist.

Daran anschließend habe ich meine nicht veröffentlichte Masterarbeit verfasst [45]. Dafür wurde der Testaufbau grundlegend neu entwickelt, sodass ein wesentlich größerer Datensatz von mehr als 100 Relais für die Untersuchungen verwendet werden konnte. Dabei wurde zunächst die Erkenntnis gesammelt, dass Relais nicht nur monotone Trends in

den Daten aufweisen, sondern auch spontan ausfallen können. Weiterhin wurden folgende Forschungsfragen beantwortet:

- Ist der Verschleiß von Relais anhand der Ströme und Spannungen in Steuer- und Lastkreis erkennbar?

Im Rahmen dieser Untersuchung wurde geprüft, ob eine Prädiktion der RUL grundsätzlich möglich ist. Die Resultate zeigen, dass zwischen neuen und verschlissenen Relais mit Hilfe der Ströme und Spannungen durch diverse maschinelle Lernverfahren mit mehr als 90 % Genauigkeit unterschieden werden kann. Somit ist der Verschleiß von Relais durch die Messgrößen erkennbar.

- Welche Eigenschaften muss der Datensatz erfüllen?

Zunächst wurde untersucht, welche Messgrößen notwendig sind. Dazu wurden verschiedene maschinelle Lernverfahren mit unterschiedlichen Kombinationen von Messgrößen und extrahierten Merkmalen evaluiert. Mit ANN können sowohl Messreihen als auch Merkmale als Eingangsdaten genutzt werden. Wichtig sind die Informationen aus Spulenstrom und Kontaktspannung beim Ein- und Ausschaltvorgang. Im Weiteren wurde beobachtet, dass es nicht notwendig ist jeden Schaltvorgang eines Relais für das Training zu nutzen.

Abschließend wurde die originale Abtastrate von 200 kHz reduziert, um zu untersuchen, welche Abtastrate notwendig ist. Dafür wurden maschinelle Lernverfahren mit Daten in verschiedenen Abtastraten trainiert. Als Ergebnis ist festzuhalten, dass ab 12,5 kHz keine relevante Performanz Verbesserung erreicht wurde. Durch eine qualitative Analyse der Messgrößen konnte diese Erkenntnis verifiziert werden.

- Welche maschinellen Lernverfahren eignen sich zur Alterungsanalyse?

Alle Lernverfahren eignen sich, um die RUL der Relais mit eindeutigen Trends zu schätzen. Durch Glättung der Schätzungen können die Varianz und damit der mittlere quadrierte Fehler signifikant reduziert werden, sodass auch simple Verfahren wie lineare Modelle eine akkurate RUL-Schätzung ermöglichen. Für Relais, die spontan ausfallen, sind lediglich Trends in den Schätzungen auszumachen. Diese sind jedoch mit einem massiven Fehler behaftet, sodass die Instandhaltung auf Basis dieser Schätzungen nicht durchgeführt werden kann.

Insgesamt kann mit den Ergebnissen meiner Masterarbeit das Potenzial von maschinellen Lernverfahren im Kontext der Instandhaltung von Relais aufgezeigt werden. Die Varianz der Lebensdauern in Kombination mit den ungenügend geschätzten RUL einiger Relais sind jedoch Anlass, das Forschungsthema im Rahmen dieser Arbeit weitergehend zu untersuchen.

Eine Arbeitsgruppe hat zu dem Thema bereits drei Veröffentlichungen publiziert. In der ersten Veröffentlichung [52] wurden sechs Relais mit unterschiedlichen Lastspannungen betrieben, um die Degradierung zu untersuchen. Als Ergebnis wird – analog zu dieser Arbeit (vgl. Abschn. 2.2) – zwischen zwei Fehlermodi unterschieden:

- „stuck opened“: Die Kontakte des Relais schließen nicht.
- „stuck closed“: Die Kontakte des Relais öffnen nicht.

Weiterhin werden die Mechanismen erörtert, die zu den Fehlermodi führen. Zudem wird in der Arbeit ein Ausblick gegeben, dass Modelle zur Abbildung der Fehlermodi entwickelt werden müssen, um PdM realisieren zu können. Dabei betonen die Autoren die Notwendigkeit eines simplen Verfahrens, damit sowohl eine lokale als auch eine dezentrale Schätzung durchgeführt werden kann.

Die Ergebnisse bezüglich der Fehlermodi decken sich mit denen aus dieser Arbeit. Auch dem vorgeschlagenen Vorgehen bezüglich PdM wird grundsätzlich gefolgt. Die Veröffentlichung [52] bestätigt somit unabhängig die Erkenntnisse, die zur Degradierung im Rahmen dieser Arbeit gesammelt wurden.

In [53] wird das Verfahren „Electromagnetic Relay Useful Actuation“ (EMRUA) für die PdM von Relais unter Verwendung von maschinellen Lernverfahren vorgestellt. Inhaltlich ist diese Veröffentlichung ähnlich zu [4], in der ausführlicher auf mehrere Aspekte zu dem Verfahren EMRUA eingegangen wird. Zunächst werden die Fehlermodi und deren Mechanismen ausführlich erläutert, wobei ein zu hoher Kontaktwiderstand als vierter Fehlermodus hinzugefügt wurde. Darauf aufbauend wird das Verfahren EMRUA vorgestellt, mit dem die RUL von Relais geschätzt werden kann.

Das Vorgehen bei EMRUA ist klassisch für maschinelles Lernen. Zunächst werden in Schritt (I) Messreihen aufgezeichnet. Diese Messreihen werden in Schritt (II) genutzt, um Merkmale zu extrahieren. Dabei wird in zeitbasierte und statistische Merkmale unterschieden, letztere sind z.B. Varianz, Minimum und Maximum. Die Merkmale werden auf ein $[0, 1]$ Intervall skaliert – ein typisches Vorgehen, um das Training von ANN zu verbessern. Im Schritt (III) wird eine Auswahl von Sequenzen durchgeführt. Diese ist notwendig für das verwendete ANN, bei dem es sich um ein „Temporal Convolutional Network“ handelt, welches für eine Schätzung die Historie der Merkmale berücksichtigen kann. Die Historie wird durch die ausgewählten Sequenzen angegeben. Dabei werden bei EMRUA drei verschiedene Indizierungen genutzt:

- „Growing-Sequence“: Die genutzte Sequenz wächst mit der Betriebsdauer an, muss aus technischen Gründen jedoch auf eine maximale Länge begrenzt werden.
- „Linear“: Die genutzte Sequenz hat eine konstante Länge; die Indizes über die Betriebsdauer sind äquidistant verteilt.
- „Exponentiell“: Die genutzte Sequenz hat eine konstante Länge; die Indizes über die Betriebsdauer sind so verteilt, dass der Abstand exponentiell abnimmt.

Im Schritt (IV) werden die Sequenzen genutzt, um das ANN zu trainieren und die RUL zu schätzen. In diesem Fall wird die RUL als „Remaining useful actuations“ bezeichnet, wobei es sich trotzdem um eine lineare Modellierung handelt, mit der die absolute Anzahl der verbleibenden Schaltzyklen eines Relais angegeben wird. Für die Schätzung wird zusätzlich ein Konfidenzintervall angegeben.

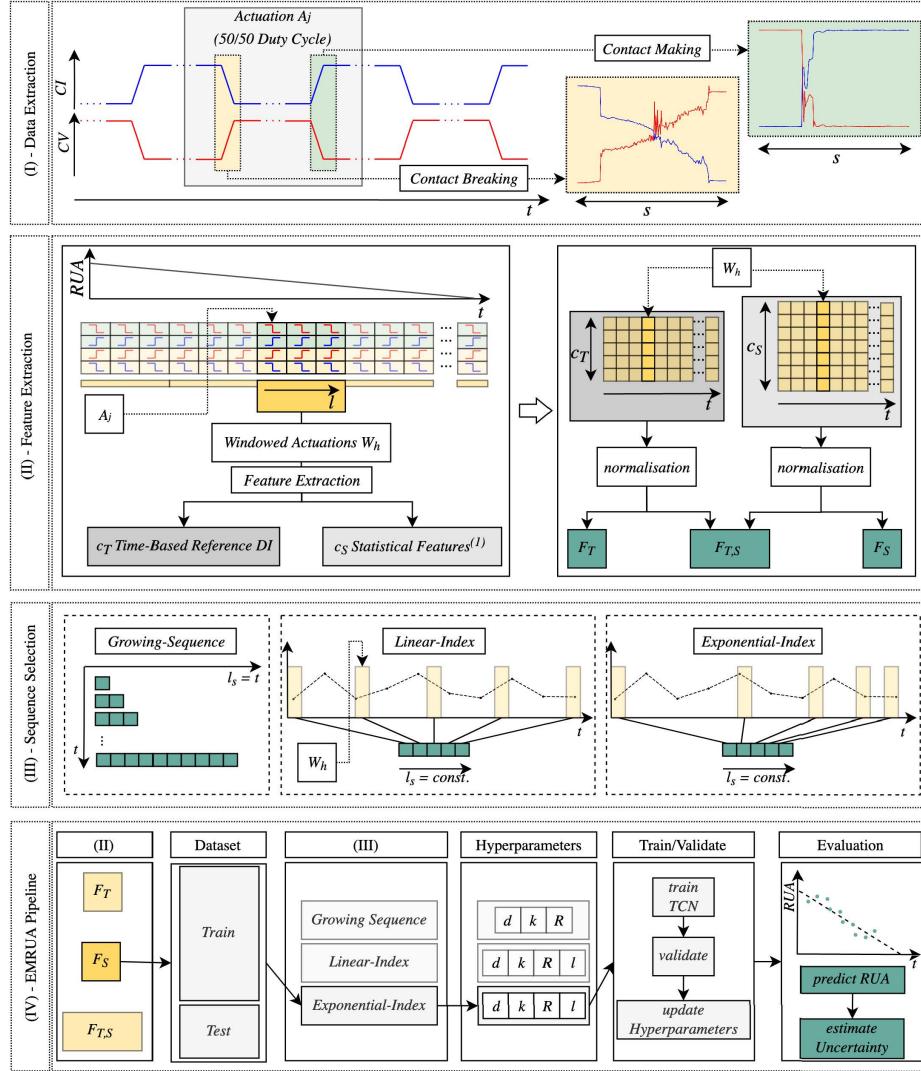


Abb. 3.5: Übersichtsgrafik der einzelnen Teilkomponenten von EMRUA[4]

Mit EMRUA gelang es, für einen Fehlermodi die RUL präzise zu schätzen und somit das Potenzial für PdM von Relais herauszuarbeiten. Die Ergebnisse sind teilweise deckungsgleich mit denen meiner Masterarbeit: In beiden Arbeiten konnte mittels extrahierten Merkmalen aus den Messreihen und einem ANN, das die Historie für eine Prädiktion berücksichtigen kann, nachgewiesen werden, dass eine präzise Schätzung der RUL für Relais mit eindeutigen Trends möglich ist. Die Arbeiten unterscheiden sich jedoch hinsichtlich der Anzahl der getesteten Relais. Bei [4] umfasst der Datensatz 16 Relais, in meiner Masterarbeit sind es 100. Weiterhin wurden die Relais im Rahmen meiner Masterarbeit mit verschiedenen Lasten gealtert, dabei ergaben sich – wie auch in [52] erwähnt – zwei Fehlermodi. In [4] wird nur einer der beiden Fehlermodi berücksichtigt; praktisch treten jedoch beide auf, weshalb EMRUA nur für einen Teil der Relais anwendbar ist. Für den

anderen Teil der Relais ergibt sich ein Problem mit der zugrundeliegenden Annahme einer linearen Degradierung, darauf wird in Abschn. 3.5 detailliert eingegangen.

3.1.3 Semi-überwachtes Lernen

Semi-überwachtes Lernen stellt einen Teilbereich des maschinellen Lernens dar, bei dem mit Daten trainiert wird, die sowohl gelabelt als auch ungelabelt sind. Damit können Datensätze verwendet werden, bei denen nicht alle Label bekannt sind. In Bezug auf die Analyse der Degradierung von Relais ist dieser Ansatz von besonderem Interesse, da hier mit vielen Millionen Schaltzyklen trainiert werden muss und deren Label unbekannt sind.

Van Egelen et al. erarbeiteten 2019 eine Taxonomie zur semi-überwachten Klassifizierung [54]. Auf oberster Ebene wird zwischen induktiven und transduktiven Methoden unterschieden, wobei transduktive lediglich auf den Trainingsdatensatz beschränkt sind, weshalb diese im Weiteren nicht betrachtet werden. Die induktiven Methoden werden in unüberwachte Vorverarbeitung, intrinsisch semi-überwacht und „Wrapper-Methoden“ unterschieden. Die unüberwachte Vorverarbeitung zielt darauf ab, Merkmale zu extrahieren, die Daten zu clustern oder initiale Parameter für die maschinellen Lernverfahren zu bestimmen, somit fällt bspw. die zuvor beschriebene Anomalie Erkennung in diese Definition. Intrinsisch semi-überwachte Methoden binden die ungelabelten Daten in die Kostenfunktion oder das Optimierungsprozedere der maschinellen Lernverfahren mit ein. Die „Wrapper-Methoden“ sind ein simpler Ansatz, um überwachte maschinelle Lernverfahren zu ertüchtigen, auch ungelabelte Daten zu verwenden. Dabei werden zwei Schritte iterativ durchlaufen: Der Trainingsschritt, bei dem überwachte maschinelle Lernverfahren mit gelabelten Daten trainiert werden. Und dem Pseudo-Labelschritt, bei dem die trainierten Lernverfahren genutzt werden, um ungelabelte Daten zu labeln. Diese neuen Label werden auch Pseudo-Label genannt. Danach folgt wieder ein Trainingsschritt mit den Labeln und Pseudo-Labeln; das Prozedere wird bis zu einem Abbruchkriterium fortgesetzt. Vorteilhaft bei dieser Methodik ist, dass praktisch alle überwachten maschinellen Lernverfahren genutzt werden können. Die Lernverfahren müssen jedoch in jeder Iteration vollständig neu trainiert werden – außer das Lernverfahren ist in der Lage, inkrementell trainiert zu werden.

Eine umfassende Übersicht und Taxonomie über „Wrapper-Methoden“ wurde von *Triguero et al.* veröffentlicht [55]. Diese vierstufige Hierarchie wird im Folgenden genutzt, um das für diese Arbeit relevante Forschungsgebiet weiter einzugrenzen:

1. Einfach- vs. Mehrfachansichten:

Diese Unterscheidung betrifft die Dimensionen der Eingabemerkmale. Bei einer Einfachansicht werden stets alle verfügbaren Merkmale verwendet. Bei Mehrfachansichten hingegen werden Subdatensätze mit niedrigerer Dimension aus dem originalen Datensatz generiert. Voraussetzung dafür ist, dass die Merkmale redundant und bedingt unabhängig sind. Auf diesem Weg kann der Generalisierungsfehler durch Verwendung mehrerer maschineller Lernverfahren reduziert werden. Im Kontext die-

ser Arbeit ist jedoch nicht sicherzustellen, dass bedingt unabhängige Subdatensätze erzeugt werden können, weshalb eine Einfachansicht genutzt wird.

2. Einfach- vs. Mehrfachlernen:

Beim Einfachlernen werden ausschließlich maschinelle Lernverfahren eines Typs, z.B. ANN, eingesetzt. Mehrfachlernen hingegen verwendet unterschiedliche Lernverfahren, wodurch die Performanz verbessert werden soll. Im Rahmen dieser Arbeit sollen die Verfahren auf einem Mikrocontroller umgesetzt werden, weshalb Speicher- und Rechenkapazität stark limitiert sind. Angesichts dessen wird Einfachlernen verfolgt.

3. Einfach- vs. Mehrfachklassifizierer:

Diese Differenzierung zielt auf die Anzahl der verwendeten Lernverfahren ab: Bei Einfachklassifizierern wird nur eins genutzt; bei Mehrfachklassifizierern hingegen mehrere, durch die die Performanz gesteigert werden soll. Analog zum Einfachlernen wird aufgrund der angestrebten Mikrocontrollerimplementierung ein Einfachklassifizierer genutzt.

4. Hinzufügungsmechanismus:

Hierbei werden drei Mechanismen abgegrenzt, mit denen ungelabelte Daten zum Trainingsdatensatz hinzugefügt werden: Erstens können die Daten inkrementell hinzugefügt werden, wenn diese definierte Kriterien erfüllen. Zweitens können die Daten stapelweise hinzugefügt werden. Dabei wird ein ganzes Set von Daten erst dann hinzugefügt, wenn alle definierte Kriterien erfüllen. Beide zuvor genannten Mechanismen haben einen Nachteil: Falsch hinzugefügte Daten werden nicht korrigiert. Deshalb ergibt sich ein dritter Ansatz, bei dem eine Änderung von hinzugefügten Daten möglich ist. Dies kann auf verschiedene Weisen umgesetzt werden. *Wang et al.* vergleichen die klassifizierten Daten anhand der Merkmale, sodass ein Beschneiden der Daten, die nicht innerhalb einer Nachbarschaft liegen, umgesetzt werden kann [56]. In [57] wird ein ähnliches Vorgehen zum Eliminieren von Ausreißern beschrieben. Im Kontext dieser Arbeit wird ein Verfahren entsprechend dem dritten Mechanismus angestrebt, damit Korrekturen stattfinden können.

Die Einordnung nach der Taxonomie nach [55] lässt Freiräume bei der konkreten Ausgestaltung des Verfahrens. Darunter fällt die Auswahl der Pseudo-Label für das Training. Typisch ist eine Auswahl auf der Grundlage der Vorhersage Konfidenz, welche ihrerseits durch geeignete Methoden abgeschätzt werden, muss und somit einen erheblichen Einfluss auf die Leistung des Algorithmus hat. Es empfiehlt sich, die Rangfolge der Vorhersage Konfidenz der Pseudo-Label für die tatsächliche Konfidenzeinstufung zu berücksichtigen. Daneben ist die Wiederverwendung bzw. Änderung der Pseudo-Label frei zu wählen. Da die in den früheren Trainingsphasen vorhergesagten Pseudo-Label im Allgemeinen weniger zuverlässig sind, wird die Gewichtung der Pseudo-Label im Laufe der Zeit erhöht. Schlussendlich muss das Training beendet werden, wozu ein Abbruchkriterium notwendig ist. *Leistner et al.* schlagen in [58] den „Out-of-bag-error“ vor, der anhand einer Stichprobe

von Daten bestimmt wird, die nicht im Trainingsdatensatz enthalten sind. Verschlechtert sich der „Out-of-bag-error“ wird das Training abgebrochen.

Die erste erfolgreiche Anwendung von „Wrapper-Methoden“ ist der von *Yarowsky* vorgestellte Ansatz zur Disambiguierung des Wortsinns in Textdokumenten, bei der die Bedeutung von Wörtern auf der Grundlage ihres Kontexts vorhergesagt wird [59]. Darauf aufbauend wurden viele weitere Verfahren vorgeschlagen, die den „Wrapper-Methoden“ zuzuordnen sind. So stellt *Lee* in [60] ein Verfahren vor, mit dem die Pseudo-Label auf Basis der höchsten Wahrscheinlichkeit gewählt werden. Trotz dieses rudimentären Vorgehens konnten konventionelle semi-überwachte Verfahren hinsichtlich der Performanz auf dem MNIST Datensatz geschlagen werden. *Azaro et al.* präsentieren in [61] ein ähnliches Verfahren, bei dem die Pseudo-Label zusammen mit den normalen Label für das Training genutzt werden, wobei stets darauf geachtet wird, dass ein Minimum an normalen Labeln in den Mini-Batches enthalten ist. Damit wurde an vier Datensätzen gezeigt, dass der aktuelle Stand der Technik im Bereich des semi-überwachten Lernens übertragen werden kann.

3.1.4 Verfahren zur Effizienzsteigerung von ANN

Mit künstlichen neuronalen Netzen konnten in den vergangenen Jahren wesentliche Erfolge in der Mustererkennung erzielt werden, sodass viele moderne intelligente Systeme auf diesem Lernverfahren aufbauen. Im Zuge der massenhaften Anwendung derartiger Verfahren wird deren Ressourcenverbrauch besonders relevant. Im Kontext dieser Arbeit soll die Inferenz auf einem Mikrocontroller stattfinden, weshalb sehr wenig Rechenleistung und Speicher zur Verfügung stehen. Diese Problematik ergibt sich für eine Vielzahl von Anwendungen. Daher ist die Optimierung von ANN hinsichtlich der Effizienz ein eigenes Forschungsthema geworden. *Cheng et al.* veröffentlichten in [62] eine umfassende Recherche zu möglichen Verfahren zur Steigerung der Effizienz. Dabei muss zunächst unterschieden werden, in welchem Kontext die Effizienz gesteigert werden soll: Einige Arbeiten konzentrieren sich auf Modellkomprimierung, andere auf die Beschleunigung oder die Senkung des Stromverbrauchs. Eine herausragende Rolle spielen dabei CNN, da diese sich besonders gut für die Bild- und Audioverarbeitung eignen und deshalb im Fokus vieler Arbeiten stehen. Nachfolgend werden sechs verschiedene Arten von Verfahren zur Effizienzsteigerung nach [62] vorgestellt, um einen Einstieg in dieses vielseitige Forschungsthema zu geben.

1. Beschneiden

Für Beschneidungs-Methoden wird eine zentrale Annahme getroffen: Viele Parameter in ANN sind unwichtig und müssen deshalb nicht berücksichtigt werden. Deshalb wurden bereits 1989 Beschneidungs-Methoden vorgeschlagen, um diese Parameter zu entfernen [63]. Auf diesem Weg wird die Effizienz hinsichtlich zweier Aspekte gesteigert: Zum einen benötigen weniger Parameter weniger Speicher; zum anderen wird auch die Anzahl der Berechnungen und damit die notwendige Rechenleistung reduziert. *Cheng et al.* unterscheiden vier Beschneidungs-Methoden: Erstens entfernen feinkörnige Methoden Parameter in einer unstrukturierten Art und Weise. Die Bedeutung der Parameter kann bspw. über die zweite Ableitung der Kostenfunktion

bewertet werden [64]. Dabei kann es zu Performanz Verlusten kommen, wenn wichtige Parameter versehentlich entfernt werden. Um dieses Problem zu lösen, schlagen *Guo et al.* ein Verfahren aus zwei Operationen vor [65]: Beschneiden und Spleißen. Mit dem Beschneiden werden die unwichtigen Parameter entfernt, während durch das Spleißen die falsch beschnittenen Verbindungen wiederhergestellt werden. Zweitens kann auf Vektor- oder Kernellevel beschnitten werden, d.h., die Kernel eines CNN werden verkleinert. Drittens ist eine einheitliche Beschneidung der Filter einer Schicht möglich, wie *Lebedev und Lempitsky* es in [66] vorschlagen. Viertens können direkt ganze Filter entfernt werden. Vorteilhaft bei diesem Vorgehen ist, dass die Eingänge der nachfolgenden Schicht so ebenfalls reduziert werden. Somit ist dieser Ansatz besonders effizient. Ein Beispiel ist das ThiNet von *Luo et al.* [67], bei dem der Rekonstruktionsfehler der Merkmalskarte der nächsten Schicht genutzt wird, um die Filterbescheidung in der aktuellen Schicht zu steuern.

2. Niederrangige Approximation

Bei einer niederrangigen Approximation wird eine effizientere Berechnung erzielt, indem Faltungs-Kernel durch einen niederrangigen approximativen Kernel ersetzt werden.

3. Quantisierung von Netzwerken

Die Quantisierung wird bereits in vielen anderen Anwendungen zur Beschleunigung bzw. Kompression angewandt und kann auch auf ANN übertragen werden. Zentrale Idee ist die Reduzierung der verwendeten Bits für eine Zahl, so kann sowohl Speicherbedarf als auch Rechenleistung eingespart werden. Generell wird zwischen zwei Hauptgruppen unterschieden. Erstens die Skalar- und Vektorquantisierung, bei der die Originaldaten durch ein Codebuch und einen Satz von Quantisierungscodes dargestellt werden. Ein Vorteil ist die Möglichkeit einer verlustfreien Codierungsme^hode (z.B. Huffman-Codierung). Auf diesem Weg wird eine hohe Kompressionsrate erreicht. Zweitens die Festkomma-Quantisierung, die wiederum in zwei Hauptkategorien unterschieden werden kann: Bei der Gewichtsquantisierung werden die Gewichte des ANN quantisiert und bei der Aktivierungsquantisierung die Aktivierungsfunktion. Die Gewichtsquantisierung ist ein ausgereiftes Thema, das bereits 1993 von *Holi und Hwang* im Rahmen einer theoretischen Analyse untersucht wurde [68]. Dabei konnten sie zeigen, dass eine 8- oder 16-Bit-Quantisierung für das Training kleiner ANN geeignet ist. In [69] wurde gezeigt, dass 16-Bit Festkommazahlen für das Training eines CNN auf dem MNIST Datensatz ausreichen. Wenn zur Gewichtsquantisierung zusätzlich eine Aktivierungsquantisierung umgesetzt wird, dann ist es möglich, das ANN nur durch die Berechnung von Festkommaoperationen auszuführen. Dies ist in Bezug auf diese Arbeit von besonderem Interesse, da so auch Mikroprozessoren verwendet werden können, die über keine Gleitkommaeinheit verfügen und daher besonders preisgünstig sind.

4. Lehrer-Schüler-ANN

Beim Lehrer-Schüler-ANN wird ein Schüler-ANN mit Hilfe eines Lehrer-ANN trainiert, wobei das Schüler-ANN effizienter ist als das Lehrer-ANN. Die unwichtigen Parameter sollen hier also entfernt werden, indem das Wissen eines komplexeren ANN auf ein einfacheres ANN übertragen wird.

5. Kompakter Netzentwurf

Eine ähnliche Strategie wird durch den kompakten ANN-Entwurf verfolgt. Wobei hier kein Wissen transferiert wird. Stattdessen wird direkt eine effiziente Netzwerkarchitektur genutzt.

6. Hardware-Beschleuniger

Weiterhin ist es möglich, die Effizienz durch Anpassung an die Hardware zu steigern. Die meisten Algorithmen basieren auf Plattformen mit GPUs und CPUs. Aufgrund der zunehmenden Anwendung von maschinellen Lernverfahren auf Endgeräten mit Mikrocontrollern wird in Zukunft die Inferenz auf eingebetteten Systemen eine größere Rolle spielen. Zu diesem Zweck werden in diversen Arbeiten Hardwarebeschleuniger vorgestellt. Dabei ergibt sich ein komplexes, umfangreiches Themengebiet, das jedoch außerhalb des Fokus dieser Arbeit liegt.

Für eine praktische Realisierung von PdM von Relais ist die Inferenz auf einem Mikrocontroller notwendig und damit auch eine Effizienzsteigerung der verwendeten ANN. Deshalb werden von den sechs verschiedenen Verfahren lediglich das Beschneiden und die Quantisierung in Bezug auf die Einflüsse auf die Performanz untersucht. Die niederrangige Approximation wird nicht betrachtet, weil diese speziell für CNN geeignet ist und diese in dieser Arbeit nur als Referenz genutzt werden. Die verwendete Topologie im Kontext dieser Arbeit ist bereits simpel, sodass vom Lehrer-Schüler-ANN und kompakten Netzentwurf kein Gebraucht gemacht wird. Schlussendlich werden Hardware-Beschleuniger indirekt durch Software genutzt – auf eine differenziertere Untersuchung wird jedoch aus forschungsökonomischen Gründen verzichtet.

Abschließend werden exemplarisch Arbeiten angeführt, bei denen durch Beschneiden und / oder Quantisierung eine erhebliche Effizienzsteigerung eines ANN erzielt werden konnte. Dabei sticht – in Bezug auf die Zitierungen – vorrangig die Arbeit von *Han et al.* hervor [70]. Sie kombinierten Beschneidung und Quantisierung mit der Huffman Codierung und reduzierten so bekannte ANN – wie das AlexNet oder VGG-16 – bezüglich des Speicherbedarfs um Faktoren von 35 bis 49. Damit geht eine drei- bis vierfache Steigerung der Beschleunigung und drei- bis siebenfache Steigerung der Energieeffizienz einher. Gleichzeitig hat das Verfahren keinen negativen Einfluss auf die Performanz. Die Arbeit von *Han et al.* zeigt auf, wie groß der Nutzen von Verfahren zur Steigerung der Effizienz sein kann. Mit einem konzeptionell ähnlichen Verfahren konnten in [71] vergleichbare Ergebnisse erzielt werden. *Stock et al.* reduzierten mit einer Vektor-Quantisierung den Speicherbedarf des ResNet50 um den Faktor 20 bei gleichzeitig geringen Performanz Verlusten [72].

3.1.5 Zusammenfassung

Der Forschungsstand für die Erschließung des Forschungsdesiderats PdM im Kontext von elektromechanischen Relais erstreckt sich auf vier Themengebiete:

- Essenziell für eine datengetriebene Modellierung der Degradierung ist der Datensatz. Es wurden bereits viele Datensätze veröffentlicht, die zu diesem Zweck verwendet werden können. Dabei lassen sich drei Typen ausmachen:
 1. Synthetische Daten, die z.B. durch Simulationen generiert wurden.
 2. Prüfstanddaten, die die Degradierung realer Bauteile unter Prüfstandbedingungen enthalten.
 3. Reale Daten, die in realen Applikationen aufgezeichnet wurden.

Alle Typen bringen Vor- und Nachteile mit sich: Synthetische Daten sind relativ günstig zu generieren, jedoch in ihrer Komplexität auf das verwendete Modell zur Erzeugung begrenzt. Prüfstanddaten sind – je nach Alterungsbeschleunigung – sehr zeit- und kostenaufwendig, dafür kommen Sie der Realität jedoch sehr nah. Bei realen Daten können die betrachteten Komponenten nicht immer bis zu einem Ausfall gealtert werden, weshalb diese Datensätze verzerrt sind. Zusätzlich hegen Unternehmen oftmals Bedenken bezüglich der Vertraulichkeit der Daten. Zusammenfassend ist in Hinblick auf die publizierten Datensätze eine Forschungslücke auszumachen: Es gibt noch keinen Datensatz, der eine große Anzahl von Einheiten umfasst, die bis zum Ausfall degradiert wurden und unterschiedlichste Ausfallmechanismen aufweisen.

- Die entwickelten Verfahren für eine Instandhaltung – basierend auf maschinellen Lernverfahren – sind vielfältig. Der Fokus der Forschung lag hier zunächst auf der Schätzung der Restlebensdauer. Dafür wurden von einfachen linearen Modellen bis zu komplexen Kombinationen von CNN und RNN diverse Verfahren vorgestellt. Ein grundsätzliches Problem ist jedoch die Annahme einer monotonen Alterung, die z.B. bei den in dieser Arbeit untersuchten Relais nicht immer zu beobachten ist. Eine mögliche Lösung stellen Verfahren zur Erkennung von Anomalien dar. Auch hierzu wurden bereits viele Arbeiten veröffentlicht, bei denen Lernverfahren trainiert wurden, um den korrekten Arbeitszustand von Maschinen zu lernen und dementsprechend schlecht bei anomalen Zuständen zu schätzen. Diese Ansätze bieten den Vorteil, dass die Komponenten nicht bis zum Ausfall gealtert sein müssen und diverse Ausfallarten bemerkt werden können. In Bezug auf Relais ergeben sich jedoch Schwierigkeiten, wie in Abschn. 3.5 erörtert wird.
- Ein vielversprechender Ansatz zur Lösung der oben genannten Verfahren ist das semi-überwachte Pseudo-Labeling. In anderen Kontexten wird es bereits erfolgreich angewandt, um Datensätze vollautomatisch zu labeln. Dabei handelt es sich nicht um eine eigene Art von maschinellen Lernverfahren, sondern viel mehr um eine „Wrapper-Methode“. Maschinelle Lernverfahren werden genutzt, um iterativ die Daten zu labeln, d.h., zu Beginn müssen bereits einige Label bekannt sein, auf deren Basis der restliche Datensatz zugeordnet wird.

- Schlussendlich ist das Ziel dieser Arbeit die Entwicklung eines Verfahrens, das in Produkten verwendet werden kann. Die Kosten für die Applikation sind deshalb ein entscheidender Faktor. Daher wird betrachtet, inwiefern eine Umsetzung des Algorithmus auf Mikrocontrollern möglich ist. Zur Steigerung der Effizienz von ANN bestehen viele Ansätze: Durch Beschneidung können unwichtige Knoten und Gewichte im Netz eingespart und somit Speicherplatz und Rechenschritte reduziert werden. Zudem kann durch Quantisierung ebenfalls Rechenleistung und Speicherplatz gespart werden.

3.2 Grundlagen

Im Folgenden werden zunächst Grundlagen zum Thema „Artificial neural network“ (dt. künstliches neuronales Netz) (ANN) beschrieben und anschließend um weitergehende Techniken ergänzt, die im Kontext dieser Arbeit relevant sind. Abgeschlossen wird dieses Kapitel mit Verfahren, zur Effizienzsteigerung der Inferenz von ANN.

3.2.1 Begrifflichkeiten und allgemeines Vorgehen

Zunächst sind einige Begrifflichkeiten voneinander abzugrenzen. Dazu wird Kap. 2 aus dem Buch „An introduction to Machine Learning“ herangezogen [73]:

- Überwachte Lernverfahren

Bei überwachten Lernverfahren sind sowohl Ein- als auch Ausgangsdaten bekannt, sodass die Lernverfahren beim Training einen Zusammenhang erlernen müssen. Dazu wird die Abweichung zwischen vorgegebenen und geschätzten Ausgangsdaten genutzt, wobei – je nach Daten – unterschiedliche Kostenfunktionen genutzt werden können.

- Unüberwachte Lernverfahren

Unüberwachte Lernverfahren werden genutzt, um unbekannte Muster in Daten zu identifizieren, dementsprechend sind die Ausgangsdaten hier nicht bekannt. Eine häufige Anwendung von unüberwachten Lernverfahren ist das Clustering, bei dem Lernverfahren, wie z.B. K-means, genutzt werden, um Gruppen in Datensätzen zu bilden.

- Semi-überwachte Lernverfahren

Eine Mischung zwischen un- und überwachten Lernverfahren stellen semi-überwachte Lernverfahren dar. Ein Beispiel ist Pseudo-Labeling, bei dem nur ein Teil der Ausgangsdaten bekannt ist. Der Rest wird iterativ durch das Lernverfahren gelabelt und anschließend für das Training genutzt.

- Bestärkendes Lernen

Beim bestärkenden Lernen soll ein Agent lernen, in einer Umgebung zu handeln. Hier steht dem Lernverfahren eine Beschreibung der Umwelt als Eingang zur Verfügung

und die Ausgaben sind Handlungen. Beim Training wird das Verfahren durch Belohnung bestärkt, sodass es lernt, in Abhängigkeit zur Umwelt die Belohnung durch die Handlungen zu maximieren.

Im Kontext dieser Arbeit werden semi-überwachte und überwachte Lernverfahren genutzt.



Abb. 3.6: Zusammenhang von Eingangsdaten \mathbf{X} , maschinellen Lernverfahren und geschätzten Ausgangsdaten $\hat{\mathbf{y}}$

In Abb. 3.6 ist der allgemeine Zusammenhang von den Eingangsdaten \mathbf{X} , dem maschinellen Lernverfahren und geschätzten Ausgangsdaten $\hat{\mathbf{y}}$ bei überwachtem Lernen dargestellt. Ziel des maschinellen Lernerfahrens ist es, mit den erfassten, Eingangsdaten \mathbf{X} die Ausgangsdaten \mathbf{y} zu schätzen. Somit wird eine Black-Box-Modellierung durchgeführt, d.h. die Assoziationsbildung zwischen Ein- und Ausgängen findet ausschließlich auf Basis der Daten statt – ohne Einbringen von Expertenwissen. Das Vorgehen für kontinuierliche und diskrete Ausgangsdaten unterscheidet sich durch die Kostenfunktionen, mit denen jeweils die Fehler von $\hat{\mathbf{y}}$ quantifiziert werden. Kontinuierliche Größen werden allgemein als Regressionsaufgabe bezeichnet und können bspw. mit der mittleren quadrierten Abweichung (engl.: Mean Squared Error, MSE) bewertet werden:

$$\text{MSE}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^n (\mathbf{y}_i - \hat{\mathbf{y}}_i)^2. \quad (3.2)$$

Bei binären Größen liegt eine Klassifikationsaufgabe vor, die bspw. mit der binären Kreuzentropie (engl.: binary cross entropy, BCE) quantifiziert werden kann:

$$\text{BCE}(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{n} \sum_{i=1}^n \mathbf{y}_i \log(\hat{\mathbf{y}}_i) + (1 - \mathbf{y}_i) \log(1 - \hat{\mathbf{y}}_i). \quad (3.3)$$

Bevor die maschinellen Lernverfahren trainiert werden können, empfiehlt es sich, den Datensatz in drei Subdatensätze zu unterteilen:

1. Trainingsdatensatz, der für das Anpassen der trainierbaren Parameter der maschinellen Lernverfahren genutzt wird.
2. Validierungsdatensatz, der während des Trainings regelmäßig evaluiert und nicht zur Anpassung der trainierbaren Parameter verwendet wird.
3. Testdatensatz, der zur Evaluierung der Performanz gebraucht wird.

In Abb. 3.7 sind beispielhafte Verläufe der Fehler auf Trainings- und Validierungsset über mehrere Epochen skizziert. Beim Trainieren ergibt sich ein Problem: Zunächst ist das

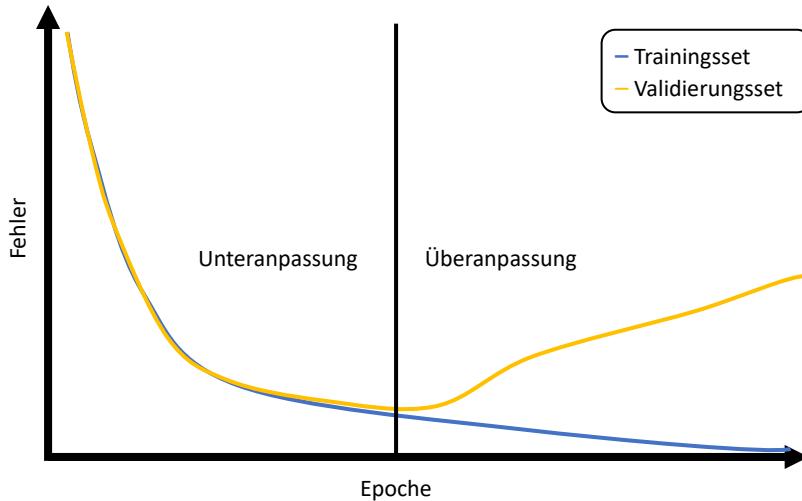


Abb. 3.7: Beispielhafter Verlauf des Fehlers auf Trainings- und Validierungsset

ANN unterangepasst, d.h. die Prädiktionen sind stark fehlerbehaftet. Dies wird gelöst, indem das ANN so angepasst wird, dass der Fehler geringer wird. Doch dies führt zu einer Überanpassung: In diesem Fall sinkt der Fehler für das Trainingsset zwar weiterhin, aber die Generalisierungsfähigkeit nimmt ab. Dies lässt sich am Validierungsset beobachten, hier steigt der Fehler mit zunehmender Epochenzahl wieder an. Deshalb werden für das Training zwei Datensätze benötigt: Der Trainingsdatensatz, auf dessen Basis die Parameter der Lernverfahren angepasst werden, und der Validierungsdatensatz, mit dem das Training abgebrochen wird. Dieser Abbruch wird auch „Early Stopping“ genannt. Um die Performanz zu beurteilen, wird der Testdatensatz verwendet.

Das Training eines ANN kann durch eine Standardisierung beschleunigt werden. Im Rahmen dieser Arbeit werden die einzelnen Merkmale nach folgender Gleichung standardisiert:

$$\mathbf{X}_{\text{std.}} = \frac{\mathbf{X} - \mu}{\sigma}, \quad (3.4)$$

dabei ist μ der Mittelwert und σ die Standardabweichung. Die standardisierten Merkmale $\mathbf{X}_{\text{std.}}$ weisen somit eine Varianz von eins und einen Mittelwert von null auf.

3.2.2 Künstliche neuronale Netze

Mit „Artificial neural network“ (dt. künstliches neuronales Netz) (ANN) können nicht lineare Zusammenhänge modelliert werden. Die Grundlagen hierzu werden im Folgenden auf Basis des Kapitels „Neuronale Netze“ von *Ertel* (2016) [74] erläutert.

Ein ANN ist in seiner Funktionsweise den biologischen Neuronen nachempfunden. Der Aufbau eines „Multilayer Perceptron“ (dt. mehrschichtiges Perzepton) (MLP) ist in Abb. 3.8 dargestellt. In dieses Netz wird ein Eingabevektor $\mathbf{x} \in \mathbb{R}^m$ der Länge $m \in \mathbb{N}$

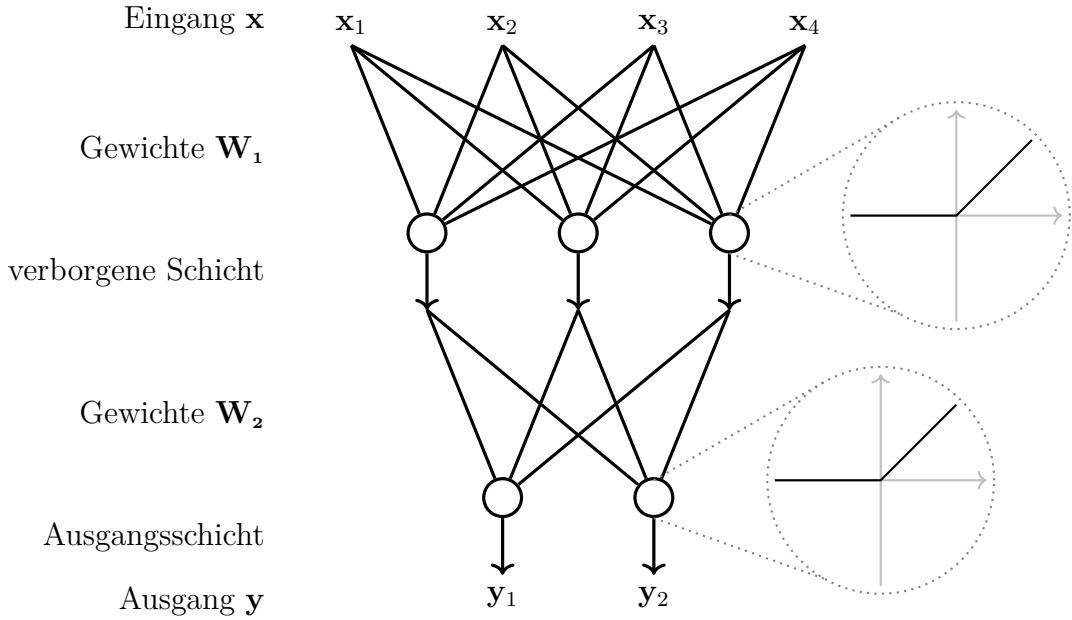


Abb. 3.8: Neuronales Netz mit zwei Schichten

eingespeist und eine Ausgabe $\mathbf{y} \in \mathbb{R}^n$ der Länge $n \in \mathbb{R}$ erzeugt. Ein- und Ausgang sind dabei über Neuronen miteinander verbunden. Diese sind in Schichten angeordnet, wobei alle Schichten zwischen der Ein- und Ausgangsschicht als verborgene Schichten bezeichnet werden. Es bestehen verschiedene Möglichkeiten, die Neuronen miteinander zu verbinden: Bei einem MLP sind die Schichten typischerweise vollvermascht, d.h. alle Neuronen aus einer Schicht sind mit allen Neuronen aus der nachfolgenden Schicht verbunden. Diese Verbindungen werden durch Skalare definiert, dieses wird im Folgenden Gewicht genannt.

Die Ausgabe $o_j \in \mathbb{R}$ eines Neurons j kann wie folgt berechnet werden:

$$o_j = f \left(\sum_{i=0}^m \mathbf{W}_{i,j} \mathbf{x}_i + \beta \right) = f (net_j) . \quad (3.5)$$

Dazu werden die Gewichte $\mathbf{W} \in \mathbb{R}^{r,s}$ zwischen den $r \in \mathbb{N}$ Neuronen aus der vorherigen und $s \in \mathbb{N}$ Neuronen aus der Schicht von j multipliziert. Weiterhin hat jedes Neuron einen Bias $\beta \in \mathbb{R}$, der addiert wird. Als Aktivierungsfunktion $f(\cdot)$ können verschiedene Funktionen gewählt werden. In Abb. 3.8 ist die „Rectified Linear Unit“ (dt. Gleichrichterfunktion) (ReLU) beispielhaft bei zwei Neuronen dargestellt. Die ReLU ist wie folgt definiert:

$$f(\mathbf{x}) = \max (0, \mathbf{x}) . \quad (3.6)$$

Alternativ ist die „Sigmoid“-Funktion für die Neuronen der Ausgangsschicht empfehlenswert, wenn eine Klassifikation durchgeführt wird:

$$f(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{x}}}. \quad (3.7)$$

Initial werden die Gewichte oft zufällig gewählt. Deshalb muss ein ANN trainiert werden, um seine jeweilige Aufgabe mit ausreichender Präzision zu erfüllen. Beim Training werden die lernbaren Parameter des ANN verändert. Beim Beispiel des MLP sind dies die Gewichte und Bias der einzelnen Neuronen. Zu diesem Zweck wird eine Lernregel benötigt. Zum Verständnis des Trainings wird die *Delta*-Lernregel vorgestellt, bei der die Änderung der Gewichte wie folgt definiert ist:

$$\Delta \mathbf{W}_{i,j} = \eta o_i (\mathbf{y}_j - o_j). \quad (3.8)$$

Die Abweichung der Ausgabe o_j gegenüber der Zielgröße y_j wird mit der Ausgabe o_i multipliziert, wobei das Neuron j sich in der nachfolgenden Schicht von i befindet. Weiterhin wird die Lernrate $\eta \in \mathbb{R}$ genutzt, um das Ausmaß der Änderungen zu begrenzen. Die Änderung wird zu den alten Gewichten addiert, sodass neue Gewichte definiert werden:

$$\mathbf{W}_{i,j}^{neu} = \mathbf{W}_{i,j}^{alt} + \Delta \mathbf{W}_{i,j}. \quad (3.9)$$

Die *Delta*-Lernregel ist lediglich für ANN mit nur einer Schicht geeignet. Für mehrschichtige ANN wird der dreischrittige *Backpropagation*-Algorithmus genutzt:

1. Forward Pass:

Der Forward Pass ist eine normale Inferenz des ANN, d.h. es werden Eingangsdaten eingegeben, Schicht für Schicht die Ausgaben berechnet, bis schließlich die Ausgaben des ANN feststehen.

2. Berechnung des Fehlers:

Daraufhin wird der Fehler bestimmt, also die Ausgaben o_j mit den wahren Werten y_j verglichen.

3. Backward Pass:

Auf Basis des Fehlers werden die Gewichte angepasst. Dabei werden die Schichten des Netzes in umgekehrter Reihenfolge durchlaufen. Begonnen wird mit der Ausgangsschicht. Die Gewichte werden wie folgt aktualisiert:

$$\Delta w_{i,j} = -\eta o_i \delta_j, \quad (3.10)$$

wobei δ_j auf dem Gradienten der Aktivierungsfunktion f basiert:

$$\delta_j = f'(net_j) \cdot \begin{cases} \sum_{k=0}^t \delta_k w_{j,k} & \text{falls } j \text{ verborgen,} \\ o_j - y_j & \text{sonst.} \end{cases} \quad (3.11)$$

Die Anzahl der Neuronen in der nachfolgenden Schicht wird durch $t \in \mathbb{N}$ bestimmt.

Im Kontext dieser Arbeit werden die Gewichte für mehrere Eingaben gleichzeitig angepasst. Dazu wird der Trainingsdatensatz in gleich große „Stapel“ (engl.: Batch) unterteilt. Beim Training nach dem *Backpropagation*-Algorithmus werden dann die Fehler der einzelnen Batches aufsummiert. Wenn alle Trainingsdaten verarbeitet wurden, dann ist eine Epoche abgeschlossen.

Für die Nutzung von ANN wird im Kontext dieser Arbeit die Bibliothek *Keras* genutzt [75], die auf das Framework *Tensorflow* aufbaut [76]. Eine Schicht von Neuronen wird hier als *Dense*-Schicht bezeichnet.

3.2.3 Spezielle Neuronen-Arten

Neben den zuvor beschriebenen normalen Neuronen wurden spezielle Neuronen bzw. Schichten entwickelt, durch die die Performanz von ANN signifikant gesteigert werden kann. Die im Kontext dieser Arbeit genutzten speziellen Neuronen / Schichten werden im Folgenden erörtert.

3.2.3.1 Dropout Schicht

Für die Performanz von ANN ist es förderlich, wenn Regularisierungstechniken angewandt werden. Im Rahmen dieser Arbeit wird dazu eine sogenannte Dropout-Schicht verwendet. Diese Technik wurde 2012 von *Hinton et al.* vorgestellt[77]. Durch eine Dropout-Schicht werden während des Trainings zufällig die Ausgaben einzelner Neuronen unterbunden. Dadurch wird verhindert, dass beim Training starke Abhängigkeiten unter Neuronen gebildet werden. Die Dropout-Schicht ist lediglich beim Training aktiv, während der Inferenz werden keine Ausgaben unterbunden.

3.2.3.2 Faltende Schicht

Ein „Convolutional neural network“ (dt. faltendes neuronales Netz) (CNN) ist besonders gut geeignet, um Eingangsdaten zu verarbeiten, die einen lokalen Zusammenhang aufweisen: im eindimensionalen sind dies bspw. Zeitreihen, im zweidimensionalen bspw. monochrome Bilder, im dreidimensionalen bspw. Farbbilder oder 3D-Darstellungen.

Diese Architektur wird auf Basis des Kap. 9 aus [78] wiedergegeben und unterscheidet sich grundsätzlich in drei Hinsichten vom MLP:

- Ein- oder mehrdimensionale Anordnung der Neuronen
- Teilen von Gewichten
- Lokale Konnektivität

Dadurch lassen sich lokale Muster in den Daten erkennen.

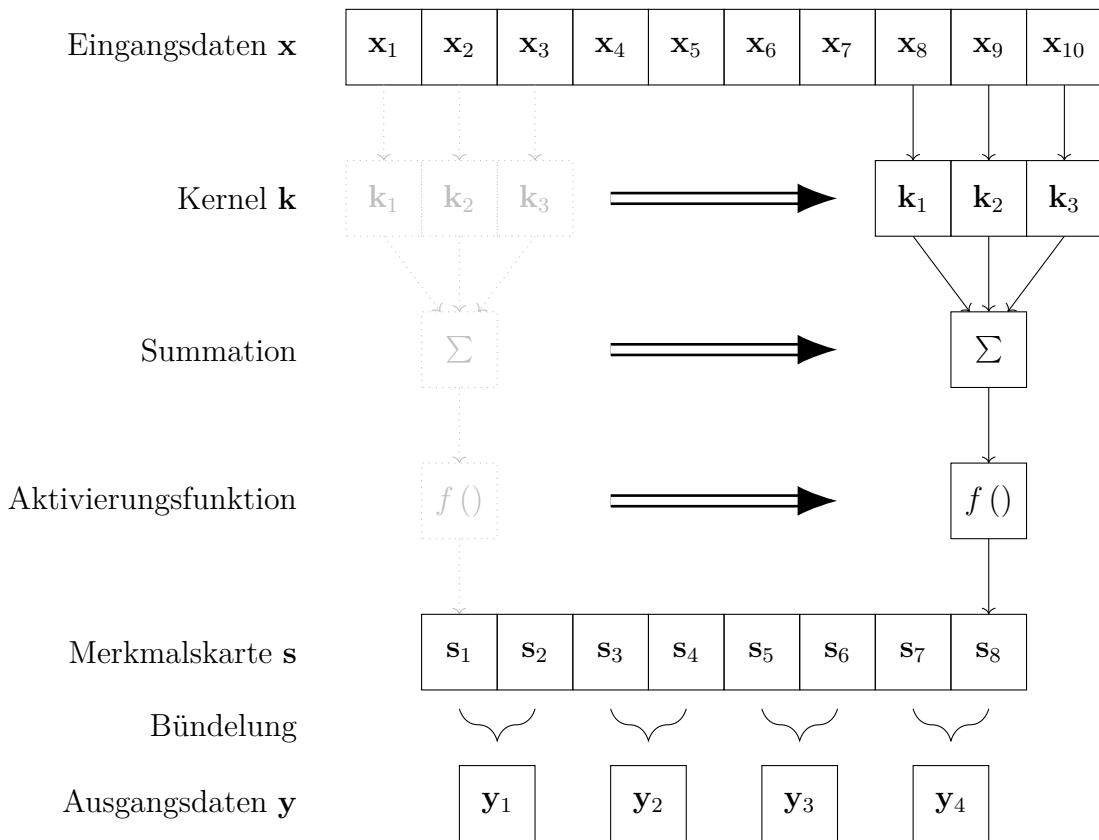


Abb. 3.9: Faltende Schicht aus Faltung, Aktivierungsfunktion und Bündelung

In Abb. 3.9 ist der Aufbau einer faltenden Schicht dargestellt. Die Eingangsdaten \mathbf{x} , die Summation und die Aktivierung sind bereits vom klassischen Neuron bekannt. Bei einer faltenden Schicht sind die Neuronen jedoch nicht direkt miteinander verbunden, stattdessen wird eine Faltung der Eingangsdaten mit einem oder mehreren Kernel $\mathbf{k} \in \mathbb{R}^m$ der Breite $m \in \mathbb{N}$ durchgeführt. Die Dimension eines Kernels richtet sich dabei nach den Eingangsdaten, im Beispiel sind eindimensionale Daten dargestellt. Das Ergebnis dieser Faltung wird oftmals als Merkmalskarte bezeichnet, formell ergibt sich folgende Berechnungsvorschrift:

$$\mathbf{s}_n = f \left(\sum_m \mathbf{x}_{n-m} \mathbf{k}_m \right), \quad (3.12)$$

wobei n das jeweilige Merkmal aus \mathbf{s} darstellt. Bei einer diskreten Faltung ergibt sich an den Rändern des Eingangs \mathbf{x} ein Problem, wenn \mathbf{s} dieselbe Größe aufweisen soll wie \mathbf{x} . Dadurch, dass die Breite des Kernels fix ist, müssen an den Rändern von \mathbf{x} Werte angehängt werden. Dieser Vorgang wird auch als „Polsterung“ (engl.: Padding) bezeichnet. Dabei entwickelten sich verschiedene Vorgehen zur Auswahl der Werte. Häufig werden Nullen oder der Randwert verwendet. Weiterhin ist es möglich die Werte von \mathbf{x} am Rand zu spiegeln. Wenn \mathbf{s} kleiner sein kann wie \mathbf{x} , dann kann die Faltung auch auf den validen Bereich beschränkt werden, sodass \mathbf{s} um $m - 1$ schrumpft, wie in Abb. 3.9 visualisiert.

Die Faltung muss nicht zwangsläufig mit einem Schritt (engl.: Stride) von eins durchgeführt werden, d.h. dass der Kernel immer nur um einen Eingang verschoben wird. In einigen Veröffentlichungen (z.B. ResNet [79]) konnte gezeigt werden, dass ein größerer Stride sinnvoll ist. In Abb. 3.9 ist ein Stride von eins dargestellt, bei einem Stride von bspw. zwei würden die Werte $\mathbf{s}_2, \mathbf{s}_4, \mathbf{s}_6, \mathbf{s}_8$ nicht berechnet werden. Durch einen größeren Stride lässt sich Rechenaufwand einsparen und \mathbf{s} verkleinern.

Dazu kann jedoch auch eine „Bündelung“ (engl.: Pooling) verwendet werden, wie in Abb. 3.9 dargestellt. Die Größe der Ausgangsdaten \mathbf{y} wird verringert, indem mehrere Merkmale von \mathbf{s} zusammengefasst werden. Das Max- und das Average Pooling sind die häufigsten Pooling Varianten: Beim Max Pooling wird nur das maximale Merkmal und beim Average Pooling der Mittelwert der Merkmale weitergegeben.

3.2.4 Autoencoder und Variational Autoencoder

Die nachfolgend beschriebenen Netze haben einen speziellen Aufbau aus Encoder und Decoder und sind zur Reduktion der Dimension der Eingangsdaten geeignet. Zunächst wird auf den „Autoencoder“ (AE) eingegangen, wobei die in Kap.14 von *Goodfellow et al.* gegebene Beschreibung grundlegend ist [78].

Der Aufbau eines AE ist in Abb. 3.10 dargestellt. Er besteht aus einem Eingang $\mathbf{x} \in \mathbb{R}^m$, mehreren faltenden Schichten \mathbf{e} , mehreren transponierten faltenden Schichten \mathbf{d} und einem Ausgang $\hat{\mathbf{x}} \in \mathbb{R}^m$. Wobei die Schnittstelle zwischen der letzten \mathbf{e} und ersten \mathbf{d} auch als Codierung $\mathbf{c} \in \mathbb{R}^n$ bezeichnet wird. Ein- und Ausgang haben dieselbe Dimension $m \in \mathbb{N}$, die stets größer ist als die Dimension $n \in \mathbb{N}$ von \mathbf{c} . Bei einem AE sind Ein- und Ausgangsdaten identisch, sodass die \mathbf{e} -Schichten als Encoder fungieren, mit dem Merkmale extrahiert werden. Aus diesen wiederum wird der ursprüngliche Eingang rekonstruiert, daher bilden die \mathbf{d} -Schichten einen Decoder. Für den Decoder werden transponierte, faltende Schichten verwendet. Eine detaillierte Beschreibung hierzu ist [80] zu entnehmen.

Ein AE kann trainiert werden wie ein klassisches ANN, dazu wird der Fehler L_{AE} wie folgt berechnet:

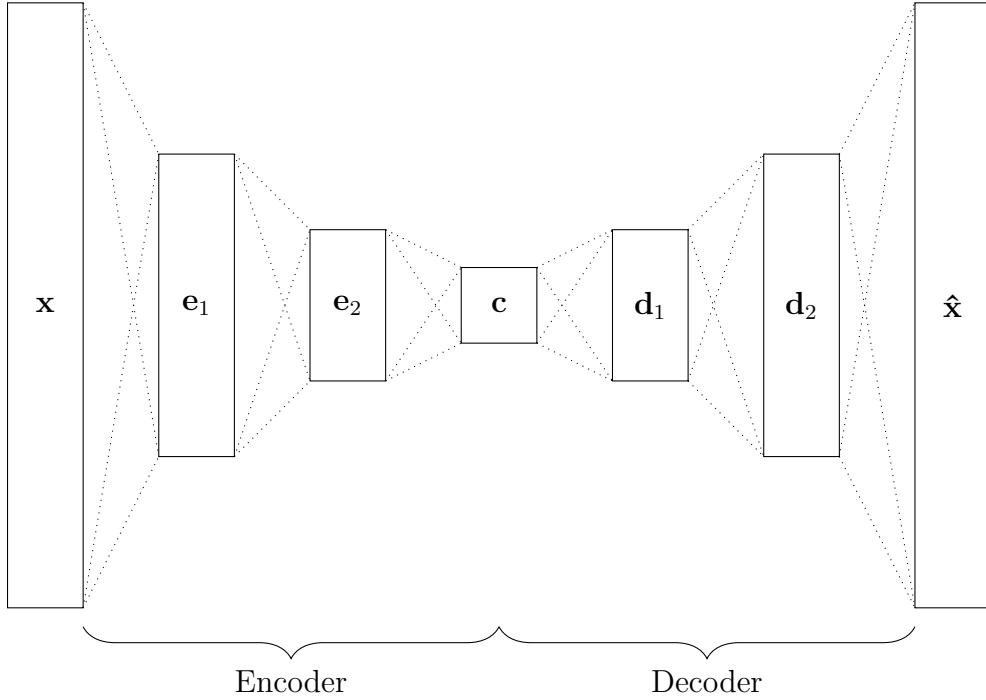


Abb. 3.10: Aufbau eines Autoencoders mit Encoder und Decoder aus jeweils zwei Schichten

$$L_{\text{AE}}(\Phi, \Sigma) = d(\mathbf{x}, D_{\Sigma}(E_{\Phi}(\mathbf{x}))), \quad (3.13)$$

dabei sind E und D die Funktionen des Encoders und Decoders, die von deren trainierbaren Parametern Φ und Σ abhängig sind. Durch d wird eine Metrik definiert, mit der die Abweichung zwischen Ein- und Ausgangsdaten quantifiziert wird.

Der Aufbau eines „Variational Autoencoder“ (VAE) ist in Abb. 3.11 dargestellt. *Kingma und Welling* stellten einen VAE erstmals in [81] vor. Die Beschreibung in diesem Abschnitt richtet sich nach dieser Arbeit. Auch bei einem VAE werden die Eingänge durch einen Encoder hinsichtlich ihrer Dimension reduziert und anschließend wieder mit einem Decoder rekonstruiert. Jedoch unterscheidet sich die Schnittstelle zwischen Encoder und Decoder signifikant. Bei einem VAE werden die Eingänge durch den Encoder verborgenen Parametern zugeordnet, denen eine Normalverteilung unterliegt. Die Ausgänge des Encoders sind somit als Mittelwert $\mu \in \mathbb{R}^n$ und Varianz $\sigma \in \mathbb{R}^n$ zu interpretieren. Zwischen Encoder und Decoder findet eine Reparametrierung statt, für die eine Zufallsvariable $\epsilon \in \mathbb{R}^n$ benötigt wird. Mit folgender Formel lässt sich \mathbf{z} als Stichprobe einer Normalverteilung berechnen:

$$\mathbf{z} = \mu + \sigma \odot \epsilon \quad (3.14)$$

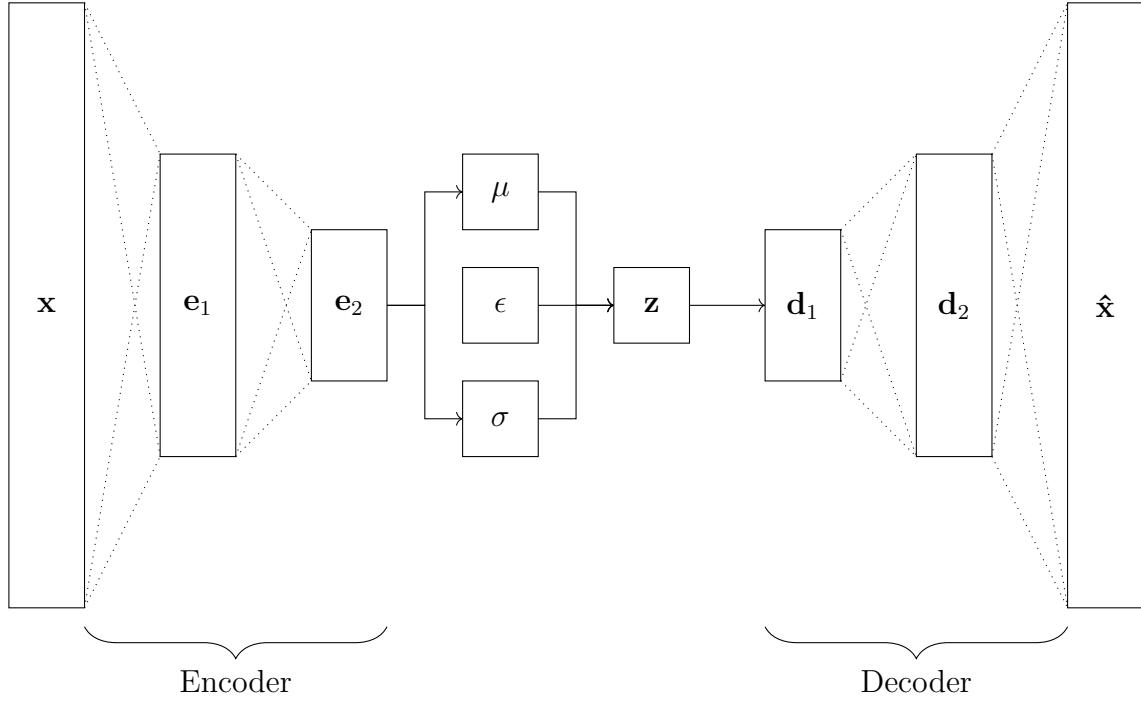


Abb. 3.11: Aufbau eines Variational Autoencoders mit Encoder und Decoder aus jeweils zwei Schichten

Aus \mathbf{z} wird durch den Decoder wieder der Eingang rekonstruiert. Für das Training des VAE ergeben sich nun zwei Fehler: einmal der Rekonstruktionsfehler wie beim AE und ein Fehlermaß für die Unterschiedlichkeit der Wahrscheinlichkeitsverteilungen des Encoders $q_\Phi(\mathbf{z}|\mathbf{x})$ und des Decoders $p_\Sigma(\mathbf{z}|\mathbf{x})$. Dazu empfiehlt sich die Nutzung der Kullback Leibler Divergenz D_{KL} , die für zwei Normalverteilungen wie folgt definiert ist:

$$D_{\text{KL}}(p \parallel q) = \log \frac{\sigma_1}{\sigma_0} + \frac{\sigma_0^2 + (\mu_0 - \mu_1)^2}{2\sigma_1^2} - \frac{1}{2} \quad (3.15)$$

wobei bei der Reparametrierung festgelegt wird, dass $q_\Phi(\mathbf{z}|\mathbf{x})$ eine Normalverteilung mit $\mathcal{N}(0, 1)$ ist. Die Parametrierung ist notwendig, um beim Training die Berechnung des Gradienten zu ermöglichen, deshalb wird ϵ als externer Vektor zur Berechnung von \mathbf{z} genutzt, wodurch diese Berechnung deterministisch wird.

Der Fehler für das Training eines VAE ist die Summe aus der Kullback Leibler Divergenz und der Abweichung zwischen Ein- und Ausgangsdaten, wobei letztere analog zum AE definiert wird:

$$L_{\text{VAE}}(\Phi, \Sigma) = D_{\text{KL}}(p \parallel q) + \mathbf{d}(\mathbf{x}, \mathbf{D}_\Sigma(\mathbf{E}_\Phi(\mathbf{x}))), \quad (3.16)$$

Sowohl AE als auch VAE finden Anwendung in der Anomalie Erkennung und der Dimensionsreduktion. Für letztere wird nach dem Training nur der Encoder des jeweiligen Modells genutzt, sodass die Eingangsdaten hinsichtlich ihrer Dimension komprimiert werden können. Auf Basis dieser kann dann bspw. eine Klassifikation oder Regression realisiert werden. Bei einer Anomalie Erkennung wird bei AE und VAE der Fehler betrachtet. Ein außergewöhnlich hoher Fehler weist dabei darauf hin, dass die Daten im Trainingsdatensatz zumindest unterrepräsentiert sind und damit als Anomalie gelten [82]. Durch die Regularisierung des latenten Raums beim VAE ist dessen Decoder geeignet, um ihn als Generator für neue Daten zu nutzen.

3.2.5 Methoden zur Effizienzsteigerung von ANN

Das Verfahren zu PdM soll auf einem günstigen Mikrocontroller ausführbar sein. Demzufolge ist eine Optimierung des ANN für limitierte Hardware erstrebenswert. Die dazu verwendeten Verfahren werden im Folgenden erläutert. Vorab wird jedoch beschrieben, wie die Effizienz von ANN gemessen werden kann. Im Kontext dieser Arbeit sind hauptsächlich die benötigte Inferenzzeit und der Speicherplatzbedarf von Interesse. Für letzteren ergeben sich zwei Kennzahlen:

- Systemspeicher für die Instruktionen (ROM)
- Arbeitsspeicher zur Inferenz (RAM)

Die Inferenzzeit lässt sich mit folgender Formel abschätzen:

$$t_{\text{inf}} = \frac{\text{MACC } \alpha}{f_{\text{CPU}}}, \quad (3.17)$$

Die „Multiply and Accumulate Complexity“ (dt. Komplexität der Multiplikationen und Akkumulationen) (MACC) gibt an, wie viele Additionen und Multiplikationen für die Berechnung notwendig sind. α ist ein Mikrocontroller-spezifischer Parameter, der definiert, wie viele Takte pro MACC benötigt werden und f_{CPU} ist die Taktfrequenz der CPU. Für den im Rahmen dieser Arbeit verwendete Cortex M-7 gilt $\alpha = 6$ und $f_{\text{CPU}} = 216 \text{ MHz}$.

3.2.5.1 Verwendete Softwaretools

Die Verfahren zur Effizienzsteigerung werden im Kontext dieser Arbeit mit verschiedenen Softwarewerkzeuge evaluiert:

- Das Unternehmen Google stellt mit „TensorFlow Lite“ ein Open Source Framework bereit, welches darauf ausgerichtet ist, maschinelles Lernen auf Systemen mit geringen Hardwarekapazitäten zu ermöglichen. Dazu zählen bspw. Smartphones, eingebettete Linux Systeme und Mikrocontroller. Zur Effizienzsteigerung werden drei Verfahren bereitgestellt: Quantisierung, Beschneidung und Parameter teilen [83].
- STMicroelectronics bietet das Softwarewerkzeug „X-CUBE-AI“ an. Der Quellcode ist nicht öffentlich zugänglich und lässt sich nur für STMicroelectronic Produkte

verwenden. Mit X-CUBE-AI können u.a. ANN in C-Code zu konvertiert und auf bestimmten Mikrocontrollern ausgeführt werden. Dabei gibt es die Möglichkeit, die ANN durch das Teilen von Parametern hinsichtlich ihres Speicherbedarfs zu optimieren[84].

Die beiden Softwarewerkzeuge wurden ausgewählt, da erfolgreiche Applikationen bereits in mehreren Veröffentlichungen dokumentiert sind. *Fouad Sakr et al.* haben X-CUBE-AI als Teil eines Frameworks genutzt, mit dem maschinelle Lernverfahren mit Mikrocontrollern ausgeführt werden können. Im Rahmen der Evaluation des Frameworks auf sechs Datensätzen haben sie u.a. gezeigt, dass die Effizienzsteigerungen keinen signifikanten Einfluss auf die Performanz der ANN haben [85].

Die Nutzung von TensorFlow Lite wird in [86] beschrieben: Ein CNN wird genutzt, um Müll zu klassifizieren und schlussendlich automatisch zu sortieren. Die Inferenz findet dabei auf einem System mit beschränkten Hardwareressourcen statt, das in einer Mülltonne verbaut ist. Durch die Quantisierung des CNN mittels TensorFlow Lite ist eine Reduktion der Inferenzzeit erzielt worden. Des Weiteren wird in [87] eine Studie vorgestellt, die den Einfluss der Quantisierung von ANN mit unterschiedlich vielen Neuronen auf deren MACC betrachtet. Der MACC kann um den Faktor vier verringert werden.

3.2.5.2 Quantisierung

Für die Quantisierung wird TensorFlow Lite genutzt, die drei Varianten der Quantisierung nach Abschluss des Trainings bietet [88]:

- Dynamische Bereichsquantisierung

Um die Inferenzzeit zu reduzieren, werden bei der dynamischen Bereichsquantisierung die Gewichte des ANN von 32 Bit Gleitkommazahlen in den Wertebereich von 8 Bit Ganzzahlen konvertiert. Zusätzlich werden die Aktivierungsfunktionen mit Ganzzahlen berechnet, wobei der Wertebereich der Funktionen dynamisch berücksichtigt wird. Durch die Optimierungen kann der Speicherbedarf eines ANN um das Vierfache verkleinert und eine zwei- bis dreimal schnellere Inferenz erreicht werden.

- Vollständige Integer-Quantisierung

Die vollständige Integer-Quantisierung ermöglicht die Ausführung auf CPUs, die keine Gleitkommaeinheit besitzen. Hierbei werden alle Parameter des ANN in Ganzzahlen konvertiert:

$$x_{\text{uint8}} = \left\lfloor \frac{x}{\gamma} \right\rfloor + \epsilon_{\text{uint8}}, \quad (3.18)$$

wobei γ und ϵ_{uint8} zu optimierende Parameter sind. Durch Betrachtung eines repräsentativen Datensatzes kann der Wertebereich der Variablen abgeschätzt werden, sodass die Parameter γ und ϵ_{uint8} für eine Umrechnung gemäß Gl. 3.18 bestimmt werden können. Eine vollständige Integer-Quantisierung ermöglicht die Reduzierung

des Speicherbedarfs eines ANN um den Faktor vier und ermöglicht eine mindestens dreimal schnellere Inferenz.

- **Float16 Quantisierung**

Einige Systeme, wie z.B. GPUs, unterstützen die direkte Nutzung von 16 Bit Gleitkommazahlen für Berechnungen. Bei diesen Systemen kann der Speicherbedarf durch eine Float16-Quantisierung halbiert werden, indem die Inferenz des ANN mit 16 statt 32 Bit Gleitkommazahlen durchgeführt wird.

Sowohl die dynamische Bereichsquantisierung als auch die Float16 Quantisierung sind vorwiegend für GPUs interessant, da diese nativ mit Float16 Daten arbeiten können, wodurch ein Geschwindigkeitsgewinn und erzielt werden kann bei gleichzeitiger Reduzierung des Speicherbedarfs. Für Mikrocontrollern ist hingegen eine vollständige Integer-Quantisierung bedeutsamer, da mehr Speicher und eine Gleitkommaeinheit gespart werden können.

3.2.5.3 Parameter teilen

Das Teilen von Parametern ist eine weitere Methode der Modelloptimierung. Dafür werden Parameter eines bereits trainierten ANNs geclustert und anschließend durch die Cluster Zentren ersetzt.

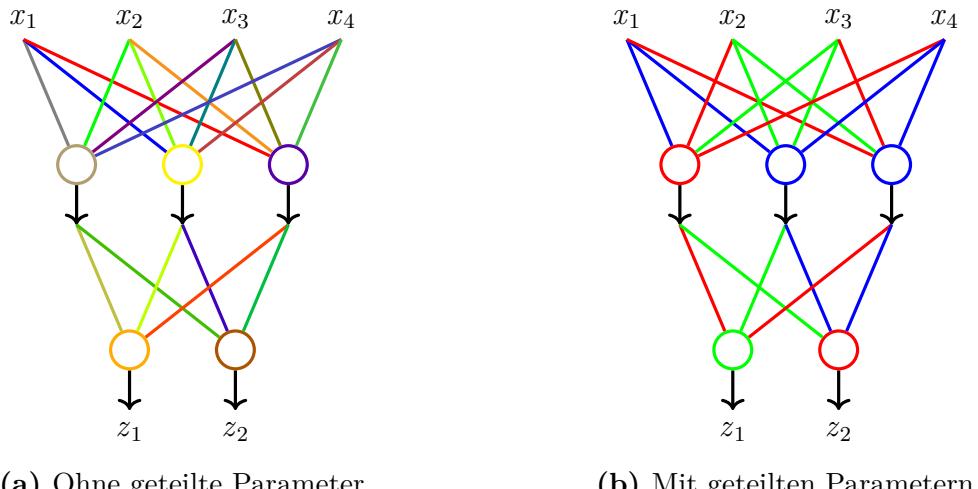


Abb. 3.12: Visualisierung von geteilten Parametern

Dies ist in Abb. 3.12 visualisiert: Ohne geteilte Parameter sind alle Parameter des ANN unterschiedlich (was durch die Farbgebung dargestellt wird). Durch das Clustering können mehrere Parameter zu einem zusammengefasst werden, wie in Abb. 3.12b durch identische Farben verdeutlicht. Aus diesem Weg kann der Speicherplatzbedarf reduziert werden. Für dieses Verfahren wird die X-CUBE-AI genutzt, die bei der Stärke der Komprimierung zwei Stufen (4 und 8) unterstützt [84]. Der Quellcode der Software ist nicht frei zugänglich und

die Dokumentation ist nur rudimentär, deshalb kann das Verfahren und die Stufen nicht detailliert beschrieben werden.

3.3 Versuchsaufbau

Im nachfolgenden Abschnitt wird der Versuchsaufbau erörtert, mit dem die Degradierung von elektromechanischen Relais aufgezeichnet wurde [6]. Dabei wird zunächst die Hard- und dann die Software beschrieben und schließlich die verwendeten Lasten.

3.3.1 Hardware

Kernstück des Testturms ist eine speziell für diese Anwendung entworfene Leiterplatte, mit der die Relais geschaltet und die Messgrößen erfasst werden. Zur Aufzeichnung und Steuerung wird die Messkarte USB-7845 vom Unternehmen National Instruments genutzt. Die Relais können nach Herstellerangaben mit einer maximalen Schaltfrequenz von 0,1 Hz betrieben werden. Deshalb wird eine Last zur Alterung von fünf Relais genutzt, die sequenziell geschaltet werden. Im Betrieb wird zwischen 1 s im eingeschalteten und 9 s im ausgeschalteten Zustand gewechselt. Auf einer Leiterplatte sind zwei mal fünf Relais aufgebracht. Ein Testsystem umfasst zwei Leiterplatten, sodass mit drei Systemen 60 Relais gleichzeitig degradiert werden können. Die Hardware für eine Last wird nachfolgend anhand zweier Schaltkreise (Steuer- und Lastseite des Relais) erörtert.

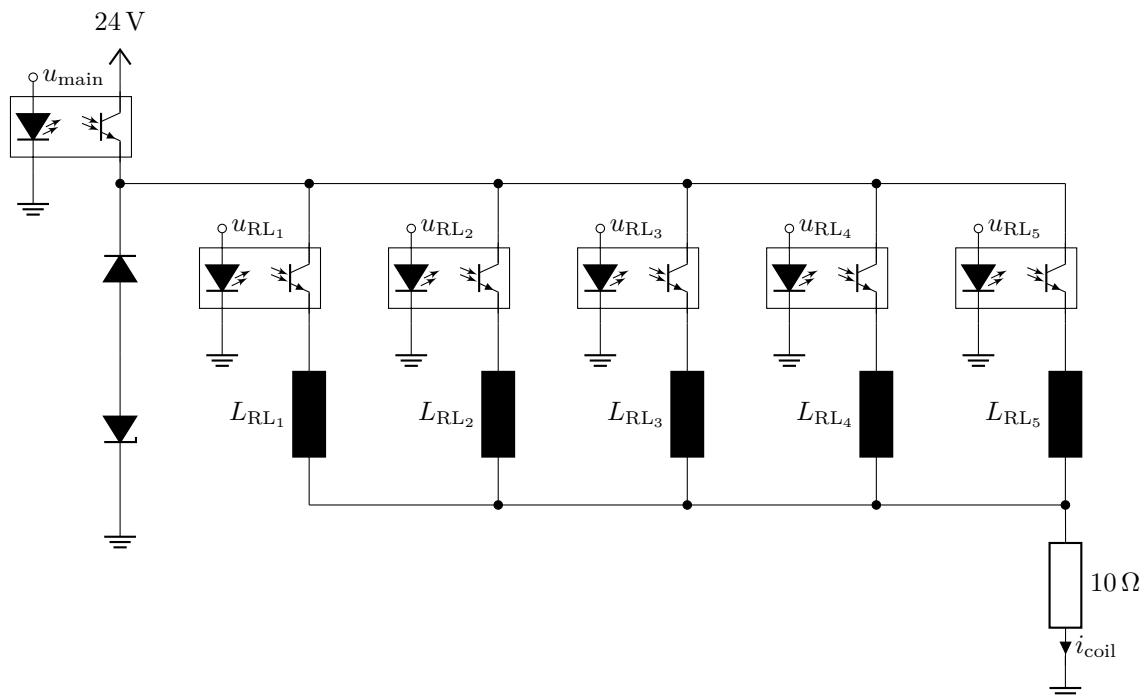


Abb. 3.13: Schaltung Steuerseite [6]

In Abb. 3.13 ist die Hardware der Steuerseite der Relais dargestellt. Diese sind deshalb nur als Induktivitäten L_{RL} dargestellt. Zum Schalten werden Optokoppler verwendet, wobei jedes Relais unabhängig mit dem jeweiligen Steuersignal u_{RL} geschaltet werden kann. Die Zenerdiode in Reihe zur Freilaufdiode sorgt für eine Begrenzung der Spannung, sodass die anderen Komponenten vor induzierten Spannungen geschützt werden. Der Strom i_{coil} wird differenziell mit einem 10Ω -Shunt gemessen.

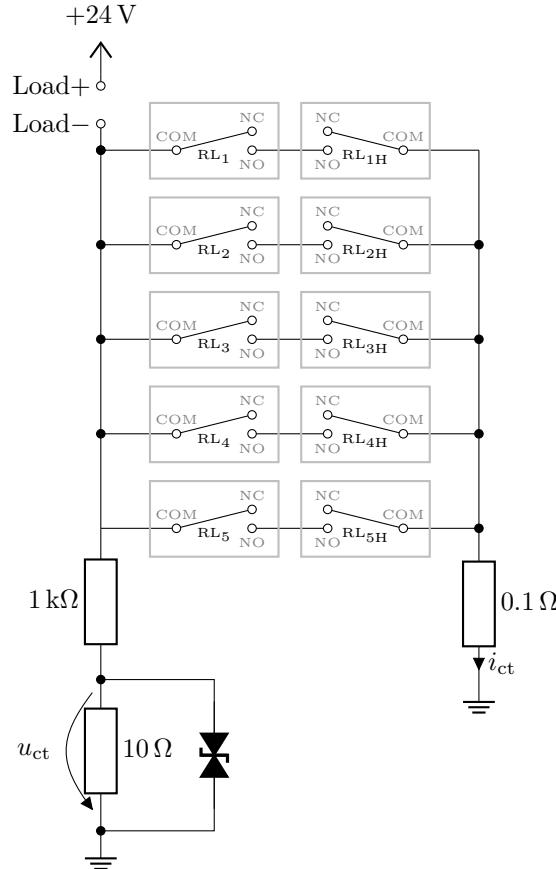


Abb. 3.14: Schaltung Lastseite [6]

In Abb. 3.14 ist die Schaltung für die Lastseite aufgezeichnet. Oben links sind zwei Anschlüsse für die Last zu sehen, woran beliebige Lasten angeschlossen werden können. Darunter ist ein Spannungsteiler mit einer Supressordiode zu erkennen, durch den die Spannung über die Relaiskontakte u_{ct} gemessen wird. Dieser Schaltungsteil ist wegen der Messkarte unumgänglich und führt dazu, dass – je nach Last – ein geringer Strom an den Relaiskontakten vorbeifließt. In der Mitte ist eine Anordnung von insgesamt 10 Relaiskontakten zu sehen. Für jedes geprüfte Relais RL_x ist ein weiteres Relais RL_{xH} vorgesehen, mit dem eine sichere Abschaltung des Strompfades im Fehlerfall gewährleistet werden kann. Rechts ist ein $0,1\Omega$ -Shunt dargestellt, mit dem der Strom i_{ct} über die

Kontakte gemessen wird. Auch auf der Lastseite werden die Messgrößen differenziell erfasst.

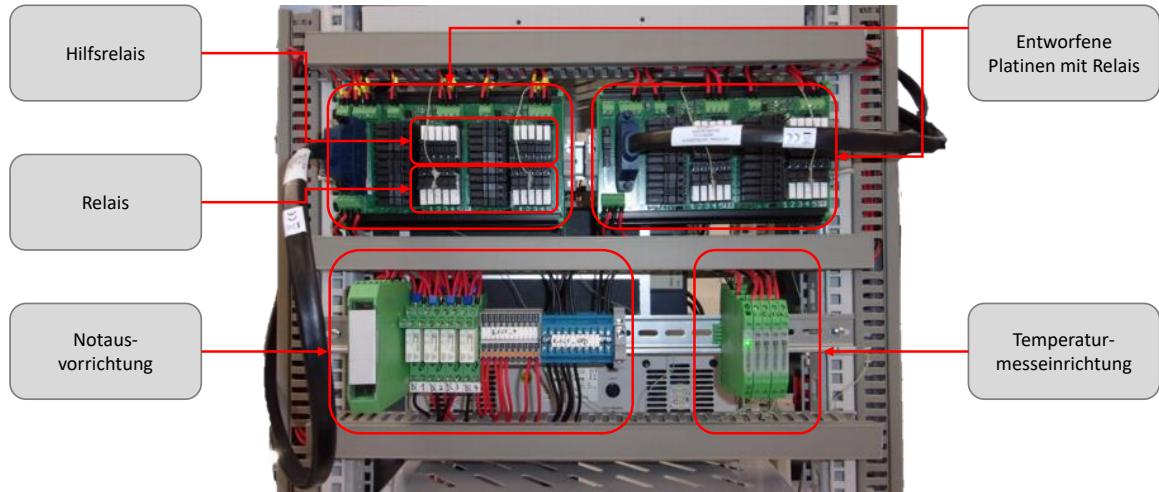


Abb. 3.15: Bild eines Testsystems mit zwei Leiterplatten [89]

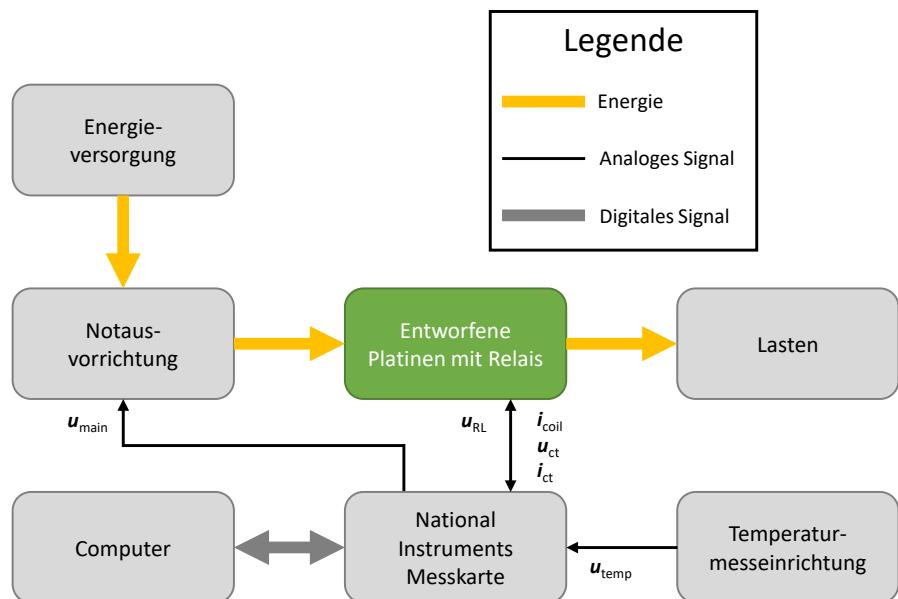


Abb. 3.16: Komponenten und Zusammenhänge eines Testsystems

In Abb. 3.15 sind die zuvor beschriebenen Schaltungen auf zwei Leiterplatten insgesamt viermal implementiert. Daneben sind weitere Komponenten zu erkennen, die in Abb. 3.16 in ein Blockschaltbild überführt wurden.

Die Messplatine ist das Herzstück der Testsysteme: Sie beherbergt die zu degradierenden Relais. Die Lasten sind direkt an die Messplatine angeschlossen. Zur Energieversorgung werden DC Industriernetzteile von Phoenix Contact genutzt, die durch einen Notaus trennbar sind. Diese wurden gezielt ausgewählt, da sie den realen Applikationsbedingungen von Relais eher entsprechen als Labornetzteile mit einer hochpräzisen Strom- und Spannungssteuerung. Für jede Last wird ein separates Netzteil verwendet. Über mehrere Thermoelemente wird die Temperatur der Leiterplatte bzw. Relais stetig überwacht, damit die Notfallabschaltung ausgelöst werden kann. Das gesamte Testsystem wird über die National Instruments Messkarte gesteuert und die Messgrößen werden damit aufgezeichnet. Die Steuerbefehle, Datenaufbereitung und Speicherung werden von einer eigens zu diesem Zweck programmierten Messsoftware realisiert, die im Weiteren beschrieben wird.

3.3.2 Messsoftware

Für die Testsysteme ist eine maßgeschneiderte Software notwendig, mit der die Tests konfiguriert, ausgeführt und überwacht werden können. Diese wurde in der Programmiersprache Python geschrieben und auf einem herkömmlichen Personal Computer ausgeführt. Über eine USB-Verbindung werden die Steuerbefehle an die National Instruments Messkarte übermittelt und die Messreihen aufgenommen. Die unverarbeiteten Messreihen und deren Abweichung von den in Kap. 2 beschriebenen werden nachfolgend beschrieben.

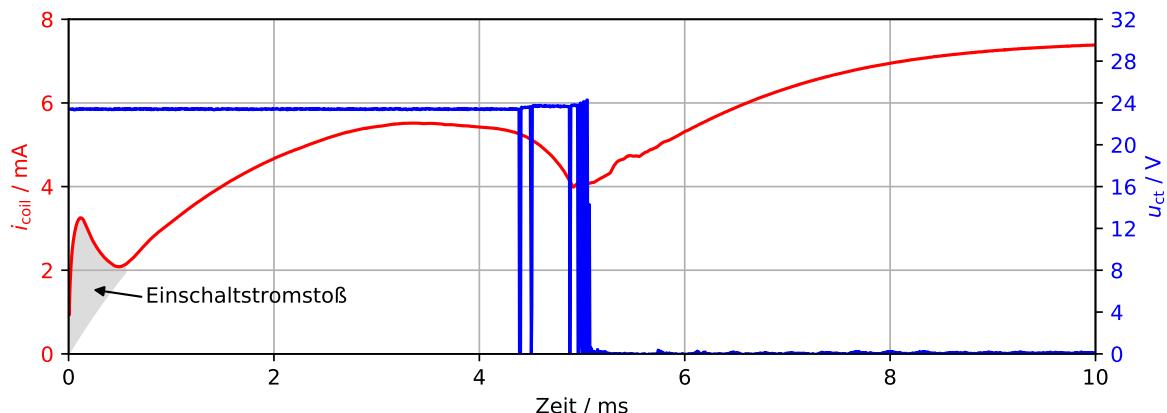


Abb. 3.17: Unverarbeitete Messreihen eines Einschaltvorgangs mit einer Abtastrate von 200 kHz

In Abb. 3.17 sind die beiden Messgrößen i_{coil} und u_{ct} für einen beispielhaften Einschaltvorgang visualisiert. Die Abweichung des Spulenstroms wurde grau hinterlegt: Zwischen 0 und 1 ms ist ein erhöhter Kurzschlussstrom zu erkennen, der auf parasitäre Kapazitäten in

der Schaltung zurückzuführen ist. Für die Alterung des Relais ist diese Abweichung nicht relevant.

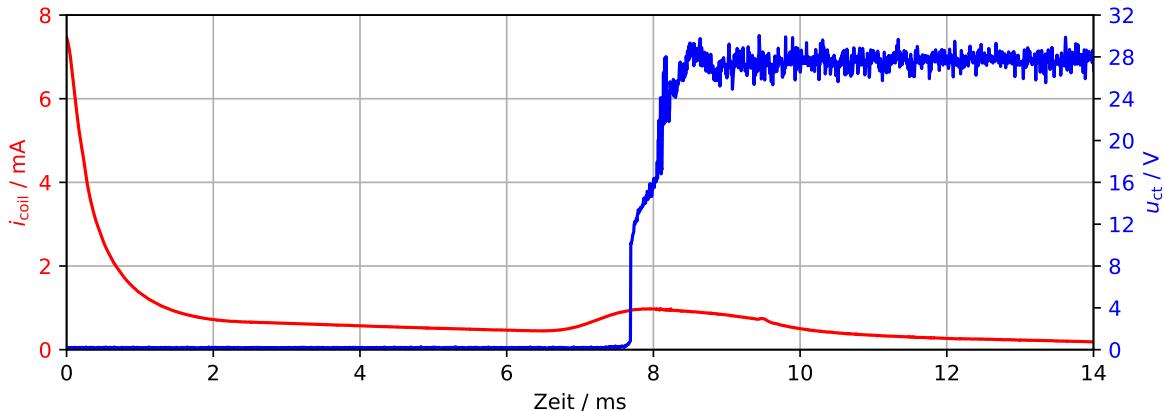


Abb. 3.18: Unverarbeitete Messreihen eines Ausschaltvorgangs mit einer Abtastrate von 200 kHz

Die Messkurven für das Ausschalten des Relais sind in Abb. 3.18 dargestellt. Bei der Kontaktspannung ist ab etwa 7,5 ms ein sprunghafter Anstieg von u_{ct} auf über 24 V zu erkennen. Zusätzlich unterliegt der gemessenen Spannung ein starkes Rauschen. Beide Auffälligkeiten sind auf die geschaltete DC13-Last-Last zurückzuführen. Durch deren induktiven Charakter wird beim Öffnen des Stromkreises ein Lichtbogen zwischen den Kontakten hervorgerufen.

Die Messgrößen werden durch die National Instruments Messkarte mit einer Abtastrate von 200 kHz aufgezeichnet. Da für jeden Schaltzyklus vier Messreihen erfasst und die Relais mehrere Zehntausend bis Millionen Schaltzyklen betrieben werden, fallen enorme Datens Mengen im Bereich von vielen Terabyte an. Für das Training maschineller Lernverfahren und die Publikation der Daten empfiehlt es sich, die Datenmenge zu reduzieren. Deshalb wird eine Abwärtstastung der Signale auf 10 kHz vorgenommen. Für höhere Frequenzen konnte im Rahmen von Voruntersuchungen keine signifikante Verbesserung der Performanz von maschinellen Lernverfahren beobachtet werden.

Zur Abwärtstastung sind zwei Komponenten erforderlich: ein Filter und ein Dezimator. Der Filter wird zur Bandbegrenzung benötigt, um alle Frequenzanteile über 10 kHz aus dem Signal zu filtern und somit Aliasing-Effekte zu unterbinden. Nach der Anwendung eines Filters kann der Dezimator genutzt werden, da mit diesem Abtastwerte aus dem Signal verworfen werden, sodass die Abtastrate sinkt.

Beim verwendeten Filter handelt es sich um ein digitales Bessel-Filter. Konkret wurde ein Tiefpassfilter der Ordnung vier mit einer Filter-Grenzfrequenz von 10 kHz entworfen. Der Frequenzgang, Phasengang und die Gruppenlaufzeit als Charakteristika des Filter sind in Abb. 3.19 visualisiert.

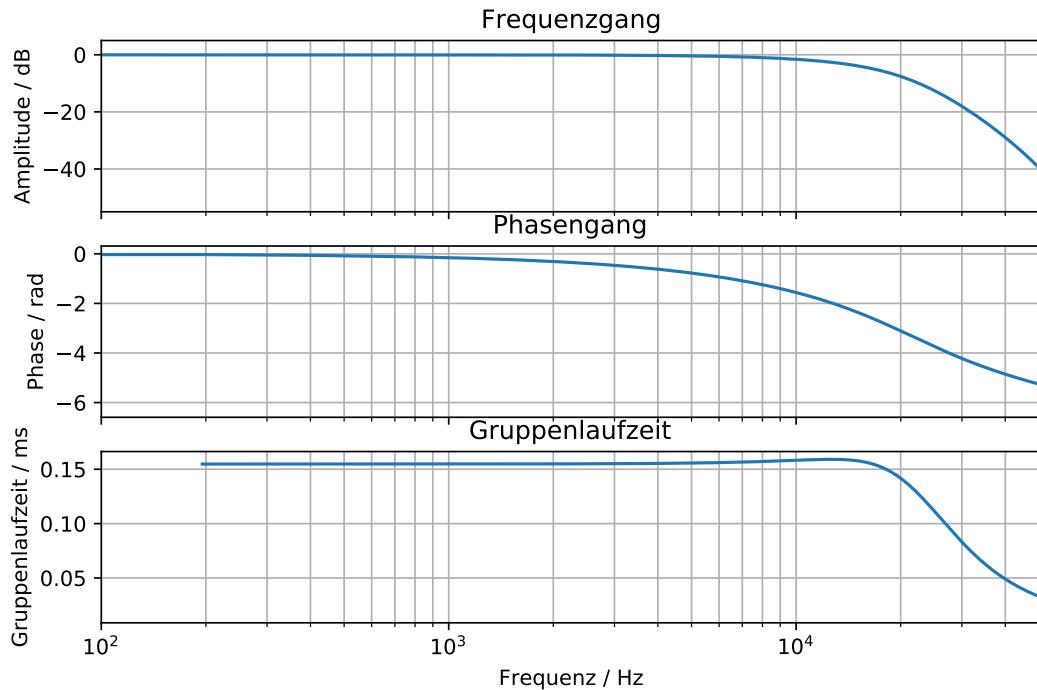


Abb. 3.19: Frequenzgang, Phasengang und Gruppenlaufzeit des verwendeten Filter

Das Ergebnis der Abwärtstastung ist in Abb. 3.20 anhand eines beispielhaften Schaltzyklus dargestellt. Der Spulenstrom und die Kontaktspannung werden für das Ein- und Ausschalten sowohl unbearbeitet mit 200 kHz Abtastrate als auch gefiltert und dezimiert mit 10 kHz Abtastrate gezeigt.

Beim Spulenstrom ist kein Unterschied zwischen beiden Frequenzen auszumachen, weshalb davon auszugehen ist, dass durch die Abwärtstastung keine Informationen in den Signalverläufen verloren gegangen sind. Dahingegen sind Unterschiede für beide Abtastraten bei der Kontaktspannung auszumachen. Diese sind auf hochfrequente Signalanteile oberhalb der Filter-Grenzfrequenz von 10 kHz zurückzuführen, die dementsprechend durch das Filter unterdrückt werden. Von einem Informationsverlust ist dennoch nicht auszugehen, da durch das Signal beim Einschalten der Start- und Endzeitpunkt des Prelens weiterhin ausgemacht werden können, genauso wie das Öffnen der Kontakte beim Ausschalten. Das hochfrequente Rauschen des Lichtbogens ist nach der Anwendung des Filter deutlich gedämpft.

Abschließend wird auf die Zählweise bei den Defekten eingegangen. Diese ist vollautomatisch in der Software integriert. Als Defekt wird ein Schaltvorgang gewertet, wenn dieser nicht zu einer Änderung der Spannung an den Kontakten (24 V zu 0 V oder umgekehrt) führt. Zu Beginn des nachfolgenden Schaltvorgangs wird die Spannung mit dem jeweiligen Soll-Wert verglichen. Ein einzelner fehlerhafter Schaltvorgang führt jedoch nicht zwangsläufig zum Totalausfall eines Relais. Mit einem Produkt, das die Messgrößen für PdM erfassen kann, könnte deshalb ein einzelner Defekt erkannt und behoben werden. Deswegen werden im

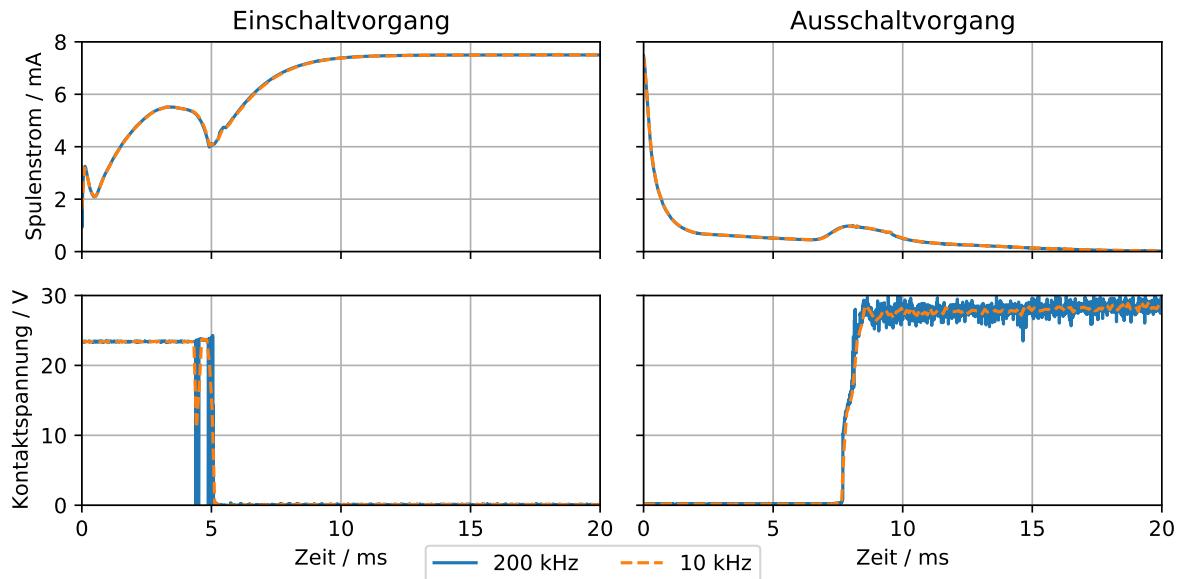


Abb. 3.20: Beispielhafte Messgrößen mit 200 kHz und 10 kHz Abtastrate

Rahmen dieser Arbeit die Relais so lange betrieben, bis zwei defekte Schaltvorgänge direkt hintereinander auftreten.

3.3.3 Lasten

Tab. 3.2: Genutzte Lasten [6]

Name	Schaltkreis Definition	Bauteilwerte
DC13: 1 A	R + L	24Ω ; 1.72 mH
DC13: 1.25 A	R + L	19Ω ; 1.72 mH
DC13: 1.7 A	R + L	14Ω ; 1.72 mH
R: 4 A	R	6Ω
R: 6 A	R	4Ω

Die Relais werden zum Schalten von 24 V Gleichspannung genutzt, wobei die in Tab. 3.2 aufgeführten Lasten verwendet werden. Kapazitive Lasten sind für die Degradierung auszuschließen, da der verwendete Relaistyp hierfür nicht zum Schalten von hohen Einschaltströme geeignet ist. Daher werden Widerstände und Spulen als Lasten verwendet, um sowohl ohmsche als auch induktive Eigenschaften abzudecken. Die Spulen sind eine sogenannte DC13-Last, die in der Norm IEC 121A/427/CD als Lastverwendungskategorie [90] spezifiziert ist. Diese induktive Last verursacht beim Öffnen der Kontakte die Bildung eines Lichtbogens zwischen den Kontakten, der bei jedem Schaltvorgang etwas Material von den Kontakten abbrennt. Der Laststrom hat einen signifikanten Einfluss auf die

Brenndauer des Lichtbogens, weshalb eine Diversifikation der Lastströme von 1,7, 1,25 und 1 A durch die Reihenschaltung von Widerständen zur DC13-Last vorgenommen wird. Weiterhin wurden ohmsche Lasten mit Lastströmen von 4 und 6 A verwendet, um die Maximalangabe der Relaishersteller abzubilden. Somit repräsentieren die verwendeten Lasten die praktischen Anwendungsbereiche der Relais.

3.4 Datensatz

Im Folgenden wird der Phoenix Contact Relay (PCR)-Datensatz vorgestellt [28], der mit dem Messaufbau aus Abschn. 3.3 generiert wurde. Er ist den bisher veröffentlichten Datensätzen in mehrfacher Hinsicht überlegen: Die Ausfallarten sind vielfältiger als die bisher bekannten, die Relais wurden unter verschiedenen Lasten und Herstellern getestet, sodass der Datensatz für reale Anwendungen repräsentativ ist; er enthält Daten von 546 Relais und mehr als 106 Millionen Schaltzyklen. Die Schaltfrequenz der Relais lag bei etwa 0,1 Hz (geringfügige negative Abweichungen aufgrund des nicht echtzeitfähigen Betriebssystems möglich), wobei das Relais 1 s ein- und 9 s ausgeschaltet wurde. Der Datensatz besteht aus 10 kHz Zeitreihen und ist damit ideal für die Entwicklung und den Test komplexer ML-Algorithmen.

In diesem Abschnitt wird zunächst auf die Degradierung von drei exemplarischen Relais eingegangen und danach werden Statistiken zum Datensatz vorgestellt. Anschließend werden 12 Merkmale präsentiert, die aus den Zeitreihen extrahiert wurden. Abgeschlossen wird das Kapitel mit der Aufteilung des Datensatzes und Informationen zum Format der publizierten Daten.

3.4.1 Degradierung von Relais

Im Folgenden werden die aufgezeichneten Daten beispielhaft dargestellt; es handelt sich um exemplarische Muster, die mehrfach in ähnlicher Weise beobachtet werden können. Teilweise sind die Ergebnisse bereits in [6] veröffentlicht. Die Grafiken sind gleich aufgebaut: Sie bestehen aus in vier Teilgrafiken, bei denen auf der Abszisse die Schaltzyklen aufgetragen sind. Jedes Relais wird über seine gesamte Betriebsdauer dargestellt. Auf der Ordinate ist die Zeit während des jeweiligen Schaltvorgangs angegeben, die beiden Plots auf der linken Seite zeigen die Messgrößen beim Einschalten und die Plots auf der rechten Seite beim Ausschalten. Die Messwerte sind farblich codiert, wobei die unterschiedliche Codierung rechts vom Diagramm dargestellt ist.

In Abb. 3.21 ist ein Relais dargestellt, das während der Betriebsdauer keine signifikanten Änderungen in $i_{coil, on}$ aufweist. Lediglich eine kurze Störung ist bei ca. 25.000 Schaltzyklen zu verorten. Beim $u_{ct, on}$ hingegen ist nach etwa 450.000 Schaltzyklen zu erkennen, dass der Kontakt immer später geschlossen wird und sich ein Muster abzeichnet. Diese Beobachtung ist auf das Prellen des Relais zurückzuführen, das mit zunehmender Betriebsdauer größer wird. Beim Ausschalten ist bei beiden Messgrößen ein deutlicher Trend zu erkennen. Das

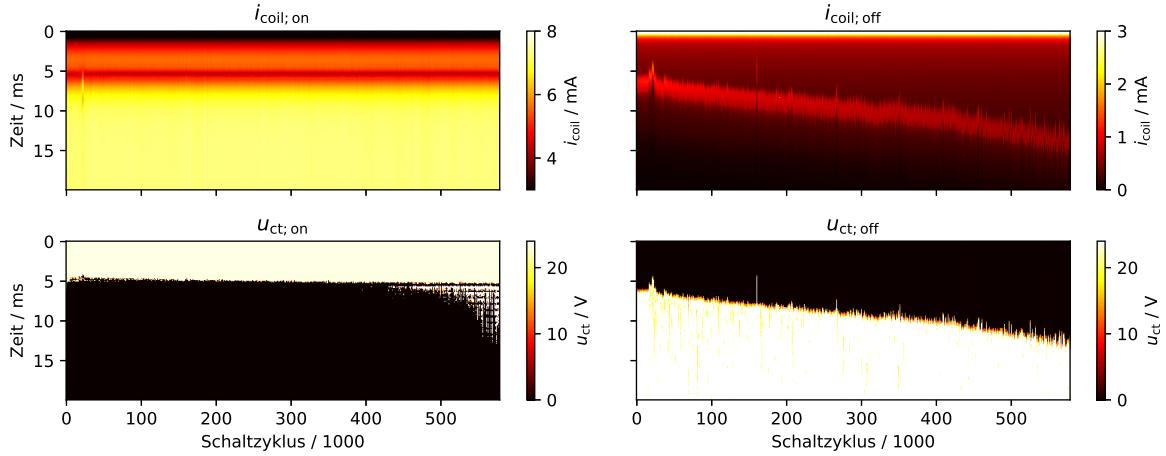


Abb. 3.21: Beispielhafte Visualisierung des kontinuierlich degradierten Relais 294

lokale Maximum des Spulenstroms beim Ausschalten tritt immer später auf, das Gleiche gilt für den Zeitpunkt des Öffnens der Kontakte.

Physikalisch lassen sich die Beobachtungen mit reduzierten Kontaktfederkräften begründen, was mit dem Abbrand der Kontakte zusammenhängt. Dieses Relais ist ein Beispiel für ein kontinuierlich gealtertes Relais, bei dem die Daten deutliche Trends aufweisen, wie sie auch in anderen Datensätzen wie [10] oder [12] zu sehen sind.

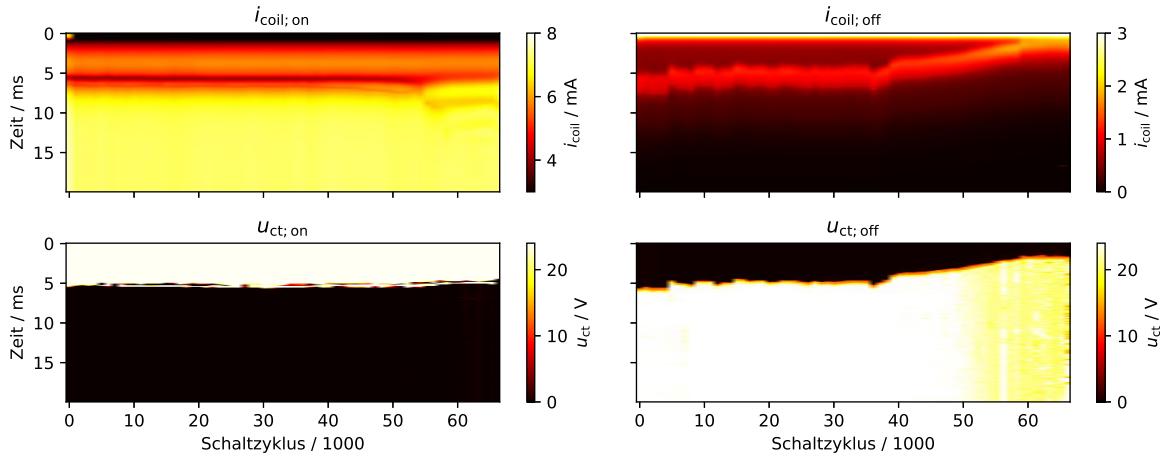


Abb. 3.22: Beispielhafte Visualisierung der spontanen Verschlechterung des Relais 84 [6]

Die Veränderungen des in Abb. 3.22 gezeigten Relais weichen von denen aus Abb. 3.21 ab; außerdem ist die Lebensdauer deutlich geringer. Beim $i_{\text{coil}; \text{on}}$ ist gegen Ende der Betriebsdauer kein klarer Ankeraufschlag mehr zu erkennen, sondern eine gedämpfte Oszillation. Im selben Zeitraum ist bei $u_{\text{ct}; \text{on}}$ zu sehen, dass das Relais etwas schneller

schaltet. Beim Ausschalten ergibt sich für $i_{\text{coil; off}}$ und $u_{\text{ct; off}}$ ein ähnliches Bild: Bis zur Mitte der Betriebsdauer bleiben sie nahezu unverändert, bis das lokale Maximum – also die Ankerrückbewegung – immer früher auftritt, ebenso wie die Trennung der Kontakte. Am Ende der Betriebsdauer konvergieren diese Zeitpunkte.

Die beschriebenen Trends lassen sich physikalisch mit einer Materialwanderung an den Kontakten erklären. Es bildet sich eine kleine Ausbuchtung an den Kontaktflächen, die den Abstand zwischen den Kontakten verringert und so dafür sorgt, dass z.B. der Anker nicht mehr anschlagen kann und eine Schwingung zu beobachten ist. Daraus resultiert ein früheres Öffnen der Kontakte.

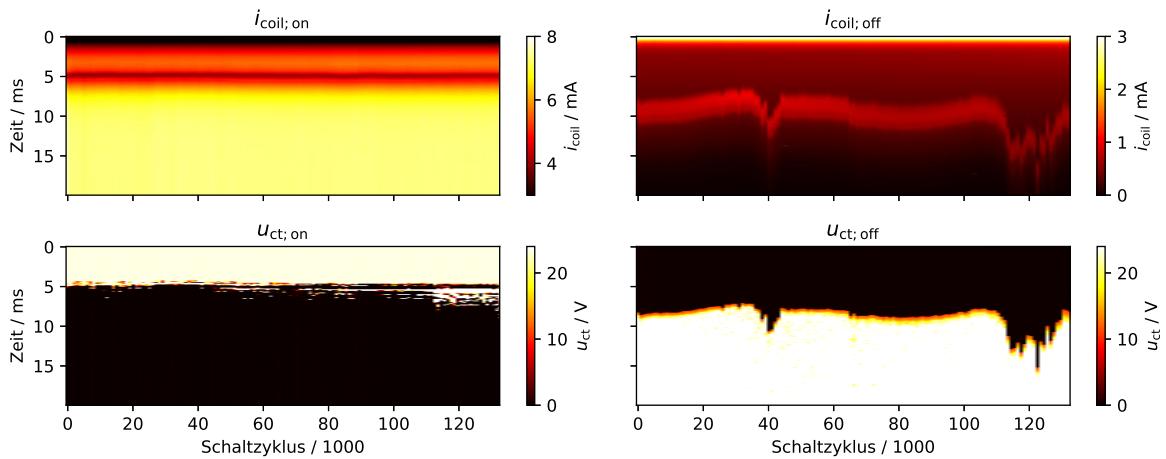


Abb. 3.23: Beispielhafte Visualisierung des Relais 299, bei dem reversible Veränderungen aufgetreten sind

In Abb. 3.23 ist ein Relais abgebildet, bei dem keine Änderungen in $i_{\text{coil; on}}$ über die Betriebsdauer zu erkennen sind und bei $u_{\text{ct; on}}$ ist eine Zunahme des Prellens zu beobachten. Beim Ausschalten weisen $i_{\text{coil; off}}$ und $u_{\text{ct; off}}$ jedoch deutliche Veränderungen auf, wobei kein monotoner Trend zu erkennen ist. Vielmehr sind es reversible Veränderungen: So ist bei etwa 40.000 Schaltzyklen in beiden Messgrößen kurzzeitig ein späteres Lösen des Ankers und der Kontakte zu beobachten. Einige tausend Schaltzyklen später gleichen die Messgrößen jedoch wieder den Anfangswerten und verändern sich für einige Zehntausend Schaltzyklen kaum. Daher ist davon auszugehen, dass die Kontakte um den Schaltzyklus 40.000 in irgendeiner Form beeinträchtigt waren, im Betrieb jedoch wieder geheilt wurden. Bei einer höheren Betriebsdauer von > 110.000 Zyklen lösen sich Anker und Kontakte abermals deutlich später. Auch hier kann tendenziell eine Selbstheilung bzw. Normalisierung der Messgrößen im weiteren Verlauf beobachtet werden. Das Relais fällt aber schließlich aus, weil die Kontakte nicht öffnen und auch nach dem Ausschalten noch Strom führen. Es handelt sich um ein typisches Beispiel für ein Relais mit einem spontanen Ausfall, da dieser nur kurzfristig angekündigt wird.

3.4.2 Statistik

Im Folgenden wird die Diversifikation des generierten Datensatzes betrachtet.

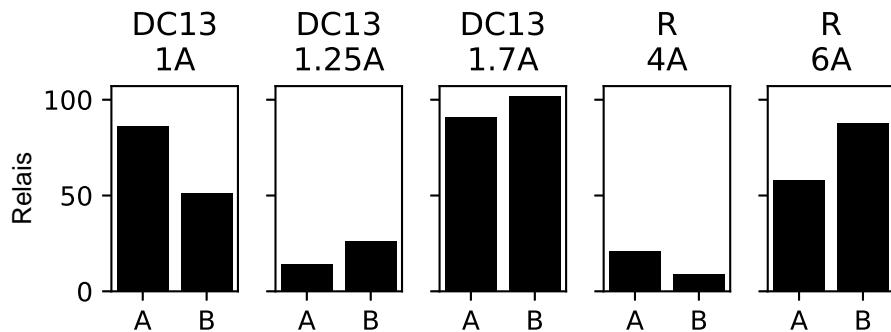


Abb. 3.24: Anzahl der degradierten Relais nach Hersteller und Last [6]

In Abb. 3.24 werden die Relais nach Herstellern und geschalteter Last unterschieden. Die Lasten DC13 1 und 1,7 A und R 6 A sind die häufigsten Lasten und haben einen großen Stichprobenumfang von jeweils mehr als 100 Relais bzw. mindestens 50 Relais pro Hersteller. Mit den Lasten DC13 1,25 A und R 4 A wurden jeweils weniger als 50 Relais gealtert, da bei der praktischen Anwendung nicht von einer Gleichverteilung der Relais auf die Lasten ausgegangen werden kann.

Die zwei Hersteller sind mit den Buchstaben A und B anonymisiert worden; insgesamt wurden etwa gleich viele Relais von beiden degradiert. Lediglich bei der Betrachtung einzelner Lasten sind Unterschiede auszumachen.

In Abb. 3.25 sind insgesamt 10 Histogramme dargestellt, aufgeteilt nach den fünf Lasten in den Spalten und den beiden Herstellern in den Zeilen. In der Legende ist die absolute Anzahl der einzelnen Relais angegeben, die Histogramme zeigen jedoch die relative Häufigkeit. Ferner wurde in jeder Grafik unterschieden, ob der Ausfall beim Einschalten (On) oder beim Ausschalten (Off) des Relais auftrat. Ein Defekt beim Einschalten bedeutet, dass ein Relais keinen Strom führt. Bei DC13-Lasten ist dies meist auf den Kontaktabbrand zurückzuführen, durch den die Kontaktpillen in ihrer Größe so verringert werden, dass sie nicht kontaktieren. Tritt ein Defekt hingegen beim Ausschalten auf, dann führt ein Relais Strom, obwohl es dies nicht mehr sollte. Die Gründe hierfür sind vielfältig, u.a. sind Folgende anzuführen: Kontaktpillen können durch Erwärmung verschweißen; rauе Kontaktflächen können ein Verhaken der Kontakte bedingen oder durch Elektromigration wird Material an den Kontakten aufgebracht, das die Trennstrecke im geöffneten Zustand überbrückt.

Grundsätzlich lässt sich feststellen, dass die Relais der beiden Hersteller unterschiedliche Lebenserwartungen haben, was auf Konstruktionsdetails zurückzuführen ist. Es gilt jedoch bei beiden Lastarten und Herstellern: Mit zunehmendem Strom wird die Lebensdauer

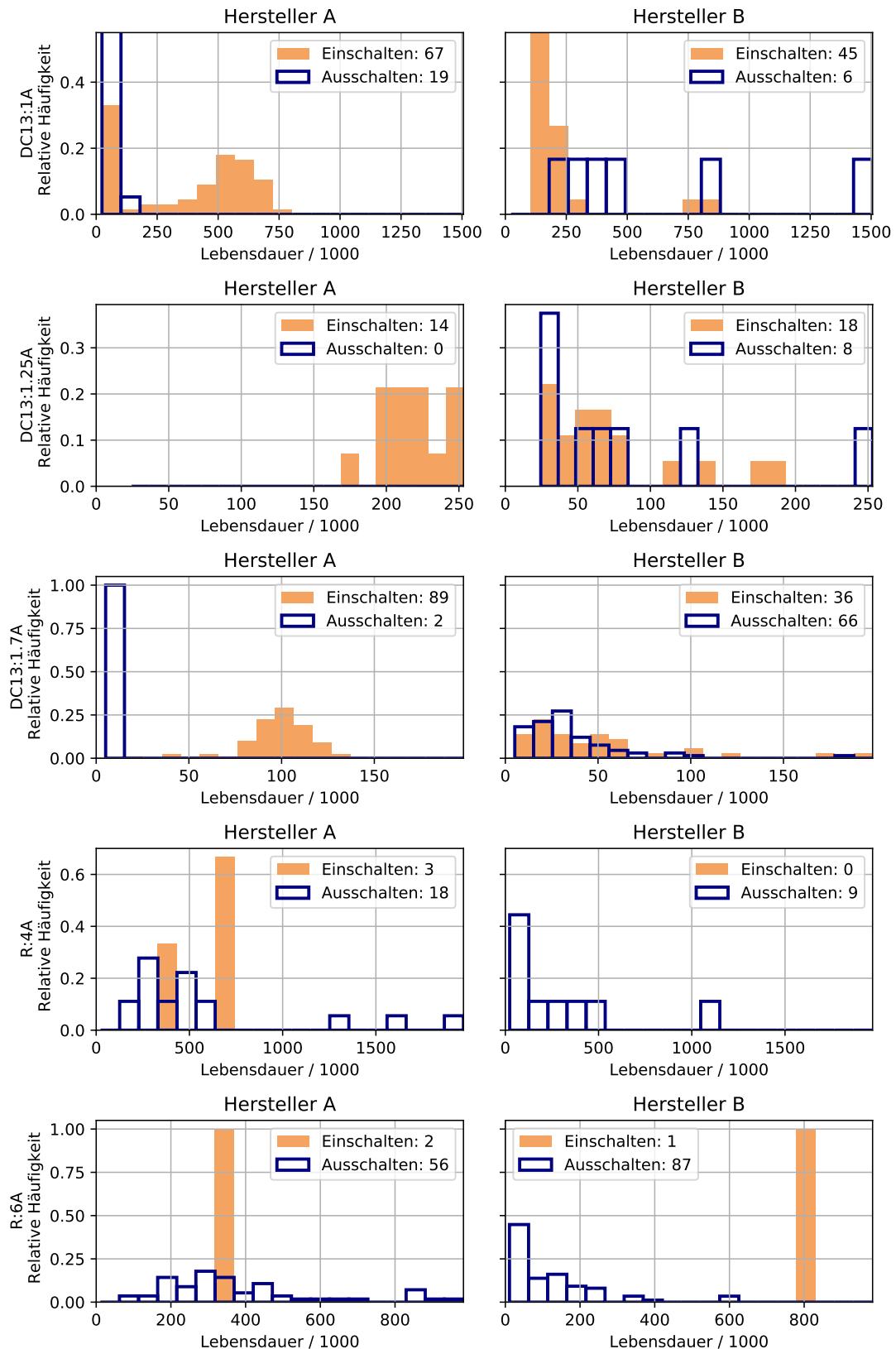


Abb. 3.25: Histogramme der Lebensdauern der Relais unterschieden nach Art des Defekts, Hersteller und Last [6]

verkürzt. Dies ist damit zu begründen, dass ein höherer Strom zu größeren thermischen Belastungen und längeren Lichtbogenbrenndauern führt, welche wiederum die Alterung beschleunigen.

Die DC13-Lasten ergeben je nach Laststrom und Hersteller unterschiedliche Verteilungen: Die Relais von Hersteller A fallen häufiger beim Einschalten aus. Das zeigt, dass in diesem Fall der Kontaktabbrand durch den Lichtbogen treibender Faktor für den Ausfall des Relais ist. Im Vergleich dazu fallen die Relais von Hersteller B deutlich häufiger beim Ausschalten aus und zeigen eine geringere durchschnittliche Lebensdauer gegenüber denen von A. Ein möglicher Grund liegt in der Konstruktion: Die Kontakte von Hersteller A weisen im ausgeschalteten Zustand einen größeren Abstand auf.

Bei ohmschen Lasten ist kein Unterschied in Bezug auf die Art des Defekts zwischen den Herstellern auszumachen: Die Ausfälle treten fast ausschließlich beim Ausschalten auf, trotzdem weist Hersteller A auch hier eine längere Lebensdauer auf. Im Vergleich zur DC13-Last haben die Relais mit ohmscher Last trotz höherem Laststrom eine größere Lebensdauer, was auf den fehlenden Lichtbogen zurückzuführen ist.

Zusammenfassend kann mit der Statistik gezeigt werden, dass die Relais in Abhängigkeit von Last und Hersteller mit unterschiedlichen Lebensdauern und aus unterschiedlichen Gründen ausfallen.

3.4.3 Extrahierte Merkmale

Im Rahmen der Degradierung der Relais wurden zu über 100 Millionen Schaltzyklen insgesamt vier Zeitreihen mit jeweils 200 Messwerten erfasst. Aufgrund dieser Datenmenge wird in diesem Abschnitt erklärt, welche Merkmale aus den Zeitreihen extrahiert werden, damit weniger Daten pro Schaltvorgang anfallen. Als Merkmal wird im Kontext dieser Arbeit ein charakteristischer Zeitpunkt oder Messwert bezeichnet, der mit Hilfe eines Algorithmus zuverlässig aus den verschiedenen Schaltzyklen extrahiert werden kann. Ein Beispiel hierfür ist der Zeitpunkt, zudem der Anker auf den Spulenkern aufschlägt.

Die Merkmale sind in Abb. 3.26 markiert, nummeriert und kurz beschrieben. Daneben ist zusätzlich der Mittelwert von i_{coil} hervorgehoben. Dieser ist von Bedeutung, da er u.a. von der Versorgungsspannung und der Spulentemperatur abhängt. Schwankungen dieser beiden Einflussgrößen können korrigiert werden, indem die Werte der Stromkennlinien durch den Mittelwert dividiert werden.

Für u_{ct} werden lediglich Zeitpunkte als Merkmale extrahiert. Dies ist mit der angestrebten Produktumsetzung zu begründen, weshalb die Beschränkungen der in [91] beschriebenen Schaltung zur Überwachung der Kontaktseite berücksichtigt werden. Mit dieser können lediglich die Zustände der Kontakte unterschieden und keine Spannungen gemessen werden. Für i_{coil} werden sowohl die Zeitpunkte als auch die Messwerte als Merkmale extrahiert, deshalb sind in Abb. 3.26 die Markierungen bei i_{coil} mit jeweils zwei Zahlen versehen. Dabei bezeichnet die geringere Zahl das Merkmal des Zeitpunktes und die größere Zahl den Messwert.

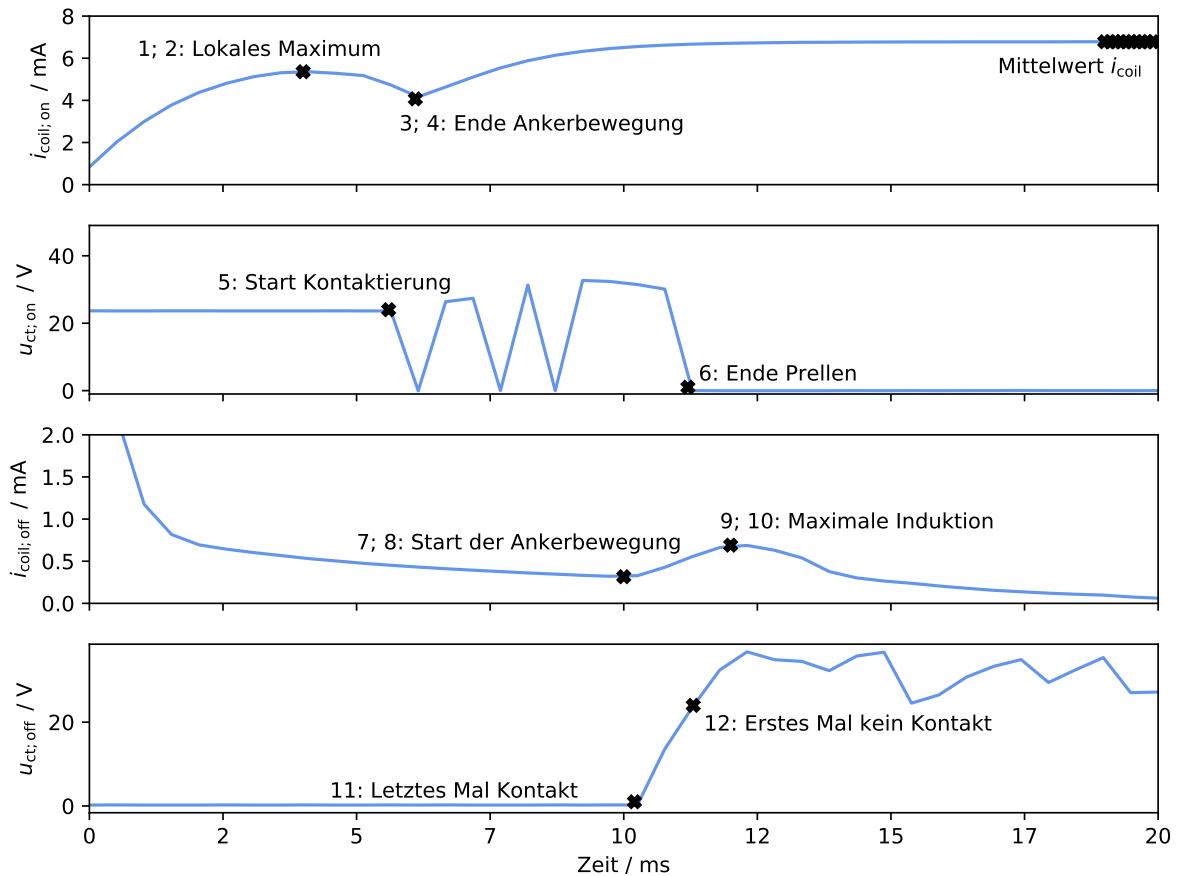


Abb. 3.26: Extrahierte Merkmale, markiert und nummeriert auf exemplarischen Messreihen [6]

Die Auswahl der Merkmale wird nachfolgend kurz erörtert:

- 1; 2: Lokales Maximum Das lokale Maximum im Spulenstrom steht in Relation zum Start der Ankerbewegung und kann somit u.a. Rückschlüsse auf den gesamten magnetischen Kreis ermöglichen, wie z.B. die Ruhelage des Ankers.
- 3; 4: Ende Ankerbewegung Das Ende der Ankerbewegung ist als ein lokales Minimum zu erkennen. Bei Verschmutzungen oder Ablagerungen zwischen Anker und Spulenkerne oder zwischen den Kontaktpaaren würde die Trajektorie des Ankers und damit der Spulenstromverlauf verändert.
- 5: Start Kontaktierung Der Start der Kontaktierung wird als letzter Zeitpunkt erfasst, zudem noch kein Kontakt besteht. Damit werden Rückschlüsse auf Veränderungen des Kontaktabstandes ermöglicht.
- 6: Ende Prellen Mit diesem Merkmal wird die Dauer des Prellens bestimmt. Aufgrund dessen entspricht die Zeitangabe hier der zeitlichen Differenz zum Merkmal 5. Ein hohes Prellen ist ein Zeichen von Verschleiß.

- 7; 8: Start der Ankerbewegung Dieses Merkmal gibt Informationen zum Lösen des Ankers von der Spule und lässt damit Rückschlüsse auf die mechanische Spannung der Kontakte zu.
- 9; 10: Maximale Induktion Die maximale Induktion ist abhängig von der Ankerbewegung und wird deshalb betrachtet.
- 11: Letztes Mal Kontakt Der Zeitpunkt, zu dem sich die Kontakte beim Ausschalten lösen, steht ähnlich wie die Merkmale 7 und 8 in Zusammenhang mit der mechanischen Kontaktspannung.
- 12: Erstes Mal kein Kontakt Dieses Merkmal gibt wieder, wie lange es dauert, bis die Kontaktspannung erstmalig erneut auf über 24 V steigt. Die gemessene Zeit ist hier die zeitliche Differenz zu Merkmal 11.

Sowohl die erfolgreiche Extraktion der Merkmale als auch deren Aussagekraft wird im Folgenden visualisiert. Dazu wurden die Merkmale extrahiert und über die jeweilige Betriebsdauer aufgetragen. Die Rohdaten der Relais wurden bereits in Abs. 3.4.1 präsentiert. Zu Vergleichszwecken wird in den nachfolgenden Abbildungen stets die relative Betriebsdauer in Prozent – bezogen auf das jeweilige Relais – angegeben. Die Werte der Merkmale sind Z-Standardisiert, wobei Standardabweichung und Mittelwert sich auf die drei Relais beziehen und die Werte damit vergleichbar bleiben.

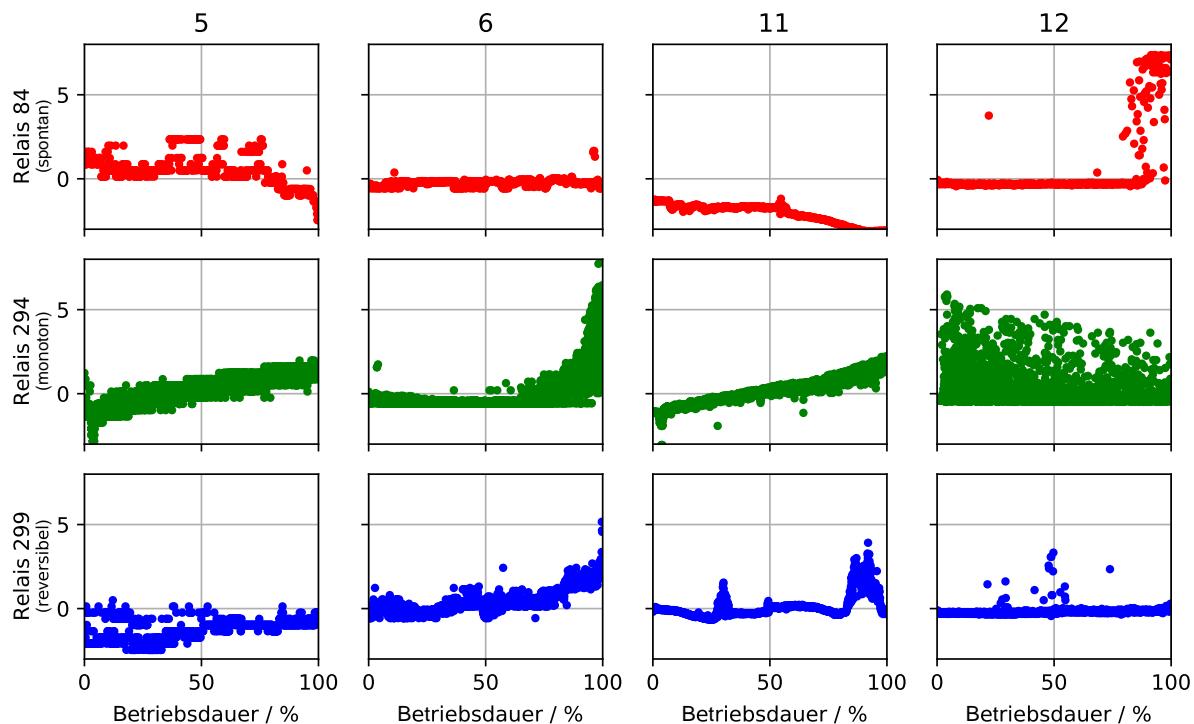


Abb. 3.27: Extrahierte Merkmale Nr. 5, 6, 11, 12 aus u_{ct} beim Ein- und Ausschalten

In Abb. 3.27 sind die Merkmale der Kontaktspannungen dargestellt. Bei Betrachtung der Verläufe wird ersichtlich, dass die in den Zeitreihen erkennbaren Veränderungen durch die Merkmale widergespiegelt werden: Bei Relais 84 war in Abb. 3.22 eine Abnahme der Zeit zum ersten Kontakt zu beobachten, wie in Merkmal 5 ebenfalls zu erkennen ist. Gleichzeitig fand der Zeitpunkt, zu dem sich die Kontakte trennen, ab etwa der Hälfte der Betriebsdauer immer früher statt – dies spiegelt sich in Merkmal 11 wider. Durch Merkmal 12 wird zum Ende der Betriebsdauer eine Verlängerung der Zeit bis zum Trennen der Kontakte angezeigt. Das Relais 294 weist in Abb. 3.21 monotone Trends auf, da die Zeitpunkte vom ersten Kontakt beim Einschalten und letzten Kontakt beim Ausschalten immer später stattfinden – dies ist hier auch bei den Merkmalen 5 und 11 zu erkennen. Weiterhin wird die Zunahme der Prelldauer durch Merkmal 6 ersichtlich. Bei Relais 299 waren in Abb. 3.23 keine monotonen Trends zu beobachten. Es gibt lediglich zwei Phasen, zu denen das Relais vom initialen Betrieb abweicht. Dies ist in Merkmal 11 zu beobachten.

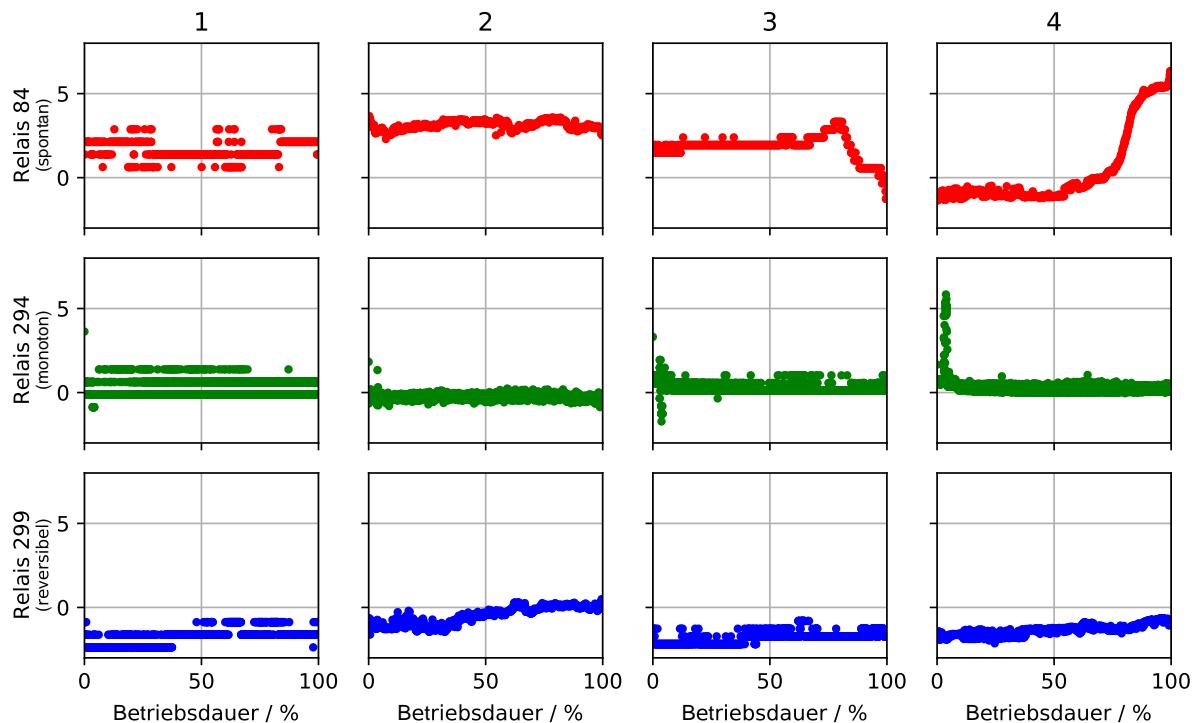


Abb. 3.28: Extrahierte Merkmale Nr. 1, 2, 3, 4 aus $i_{\text{coil}; \text{on}}$

In Abb. 3.28 sind die Merkmale, die aus $i_{\text{coil}; \text{on}}$ extrahiert wurden, dargestellt. Bei Merkmal 1 sind mehrere Level zu erkennen, zwischen denen keine Werte auftreten. Dies ist auf die Diskretisierung zurückzuführen, die die Genauigkeit bei der Bestimmung des Zeitpunktes begrenzt. Analog zu den vorangegangenen Merkmalen werden auch hier die Beobachtungen aus den Verläufen der Zeitreihen in den Merkmalen widergespiegelt: Bei Relais 84 ist der Anker ab etwa der Hälfte der Betriebsdauer nicht mehr normal aufgeschlagen, was in den Merkmalen 3 und 4 ebenfalls zu beobachten ist. Der Zeitpunkt findet immer früher statt

und der Wert steigt an. Dies ist auch bei der kurzzeitigen Störung von Relais 294 zu sehen. Ansonsten sind keine auffälligen Änderungen der Merkmale bei den drei dargestellten Relais zu sehen.

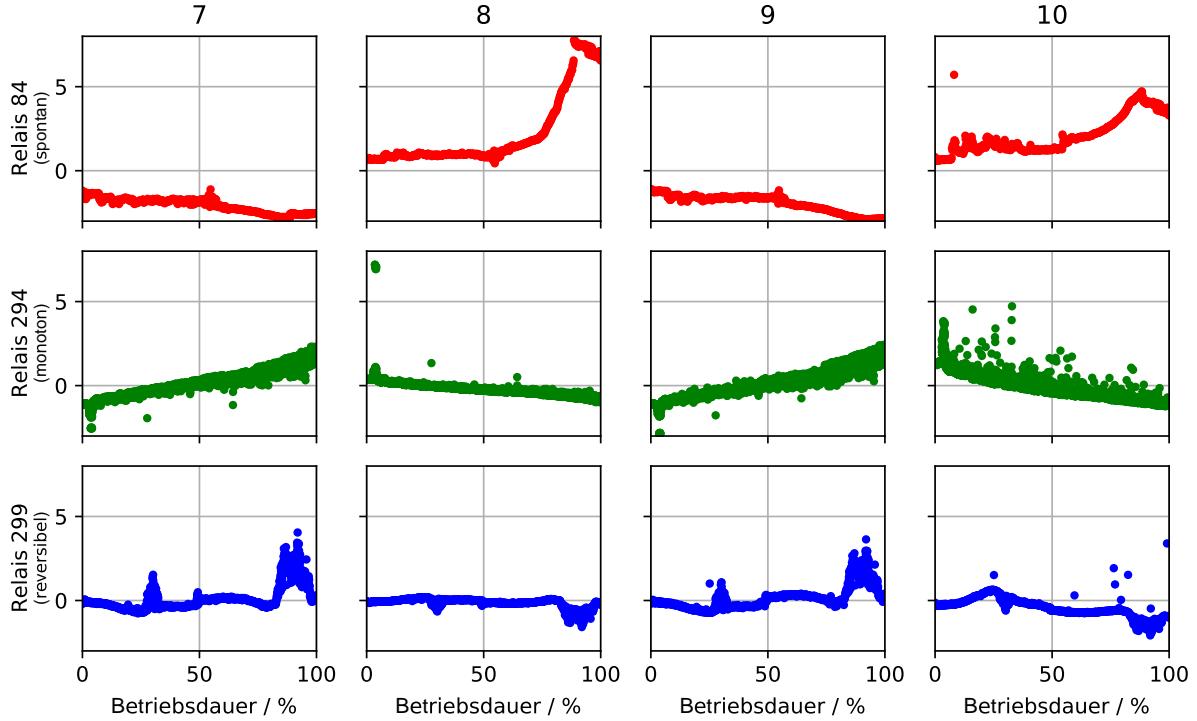


Abb. 3.29: Extrahierte Merkmale Nr. 7, 8, 9, 10 aus $i_{coil; \text{off}}$

Abb. 3.29 beinhaltet die Merkmale, die aus $i_{coil; \text{off}}$ extrahiert wurden. Die Beobachtungen aus den vorherigen Abbildungen wiederholen sich an dieser Stelle: Relais 84 weist ab etwa der Hälfte der Betriebsdauer ein auffälliges Verhalten auf, das in den Merkmalen 7 bis 10 ebenfalls ersichtlich ist. Für Relais 294 sind monotone Trends zu erkennen, wie auch in Abb. 3.21. Relais 299 weist in Abb. 3.23 zwei Phasen auf, in denen es stark vom initialen Betrieb abgewichen ist. Die erste Phase endete jedoch wieder im normalen Betriebszustand, sodass hier von einem reversiblen Phänomen ausgegangen werden kann. Beide Phasen mit auffälligen Verhalten sind in den Merkmalen enthalten.

3.4.4 Publizierter Datensatz

Die Daten werden im HDF5-Format veröffentlicht, wobei jede Datei die Daten eines Relais enthält und nach diesem benannt ist. Das Format zeichnet sich durch eine gute Komprimierung und die Möglichkeit aus, nur Teile der Daten in den Arbeitsspeicher zu laden. Die Dateien setzen sich aus Attributen und Gruppen zusammen. Die Attributnamen und Kurzbeschreibungen sind in Tab. 3.3 angegeben. Aus Gründen der Geheimhaltung wurden die Herstellernamen anonymisiert. Die Namen der Gruppen, ihre jeweilige Dimensionalität und eine kurze Beschreibung sind in Tab. 3.4 zu finden.

Tab. 3.3: Datei-Attribute [6]

Name	Beschreibung			
Laststrom	Verwendeter Strom für die Degradierung in A			
Lasttyp	Verwendeter Lasttyp für die Degradierung			
Hersteller	Anonymisierte Herstellerinformation			
Set Zuweisung	Empfohlene Zuweisung	zum	Trainings-,	Validierungs- oder Testset

Damit sich anknüpfende wissenschaftliche Arbeiten auf das gleiche Testset beziehen können, wird eine mögliche Aufteilung des Datensatzes vorgeben. Im Kontext dieser Arbeit wird der Datensatz in drei Subdatensätze aufgeteilt: einer für das Training, einer für die Validierung während des Trainings und einer für das Testen der Generalisierungsfähigkeit.

Tab. 3.4: Datei-Gruppen [6]

Name	Beschreibung	Dimensionen
I_coil_on	Matrix mit 10 kHz Zeitreihendaten von $i_{\text{coil}; \text{on}}$ bei jedem Zyklus für 20 ms nach dem Einschalten	1: Schaltzyklus 2: Zeit
I_coil_off	Matrix mit 10 kHz Zeitreihendaten von $i_{\text{coil}; \text{off}}$ bei jedem Zyklus für 20 ms nach dem Einschalten	1: Schaltzyklus 2: Zeit
U_ctct_on	Matrix mit 10 kHz Zeitreihendaten von $u_{\text{ct}; \text{on}}$ bei jedem Zyklus für 20 ms nach dem Einschalten	1: Schaltzyklus 2: Zeit
U_ctct_off	Matrix mit 10 kHz Zeitreihendaten von $u_{\text{ct}; \text{off}}$ bei jedem Zyklus für 20 ms nach dem Einschalten	1: Schaltzyklus 2: Zeit

Dabei sollen die Daten homogen auf die Subdatensätze aufgeteilt werden, da ansonsten die Generalisierungsfähigkeit der maschinellen Lernverfahren nicht geprüft werden kann. Die Aufteilung basiert auf den Metadaten der Relais, speziell den Herstellern und den geschalteten Lasten. In Tab. 3.5 sind die jeweiligen Anzahlen der Relais nach Hersteller- und Lastkombination für die drei Subdatensätze angegeben. 50 % der Relais sind dem Trainingsset zugeordnet und jeweils 25 % den anderen Sets. Diese Aufteilung ist im Kontext der Datenwissenschaften untypisch, wurde jedoch so gewählt, damit mehr Relais im Testset sind und somit zuverlässigere Aussagen über die Performanz der maschinellen Lernverfahren getroffen werden können.

Tab. 3.5: Aufteilung des Datensatzes [89]

Hersteller & Last	Training	Validierung	Test
A - DC13	93	49	49
B - DC13	87	46	46
A - R	38	20	21
B - R	48	24	25
Insgesamt	266	139	141

3.5 Forschungsbedarf

Auf Basis des aufgezeichneten Datensatzes wird in diesem Abschnitt gezeigt, dass der Stand der Technik nicht zur Instandhaltung von elektromechanischen Relais geeignet ist. Dazu wird auf drei Gattungen eingegangen: die statistische Modellierung, die Schätzung der Restlebensdauer und Verfahren zur Identifizierung von Anomalien. Diesen werden konkrete Verfahren als Beispiel zugeordnet, anhand derer die Probleme der jeweiligen Obergattung beleuchtet werden. Abschließend wird aufgezeigt, dass ein hohes Potenzial für PdM von Relais besteht. Diese Erkenntnisse sind in kürzerer Fassung bereits in [6] und [89] publiziert.

3.5.1 Statistische Modellierung

Die genutzten Verfahren bei einer statistischen Modellierung sind abhängig vom Bauteil. Für elektromechanische Relais etablierte sich der B10 Wert [92]. Eine individuelle Betrachtung der Relais findet nicht statt, da der Wert auf alle gleichermaßen angewandt wird.

In der Praxis ergeben sich hier schon die ersten Herausforderungen, denn die Degradierung und damit der B10 Wert sind maßgeblich abhängig von Parametern wie Last, Hersteller und Temperatur. Das bedeutet, dass diese Parameter bekannt sein müssen und der Hersteller Erfahrungswerte dazu vorhalten muss. Dies stellt für beide Seiten eine hohe Komplexität dar: Die Charakterisierung von Lasten inkludiert oftmals nicht deren Impedanz. Gleichzeitig ist es für einen Hersteller fast unmöglich, Erfahrungswerte für alle möglichen Kombinationen der Parameter zu sammeln. Hierfür müssten mehrere tausend Relais degradiert werden, was viele Jahre dauern kann. Weiterhin gibt es neben den bekannten Parametern zusätzliche Einflüsse, die sporadisch auftreten können und die Lebensdauer der Relais beeinflussen.

Aber selbst wenn bspw. für Hersteller und Last umfangreiche Lebensdauertests zur Verfügung stehen, liegt in der statistischen Modellierung ein weiteres Problem. In Abb. 3.30 sind die Häufigkeitsverteilungen der Lebensdauern von Relais für zwei Hersteller und zwei Lasttypen dargestellt. Es ist wichtig zu beachten, dass die Abszisse logarithmisch skaliert ist. Zur Bestimmung des B10 Wertes müssen aus den Häufigkeitsverteilungen Wahrscheinlichkeitsdichtefunktionen geschätzt werden. Bei Betrachtung der Form der

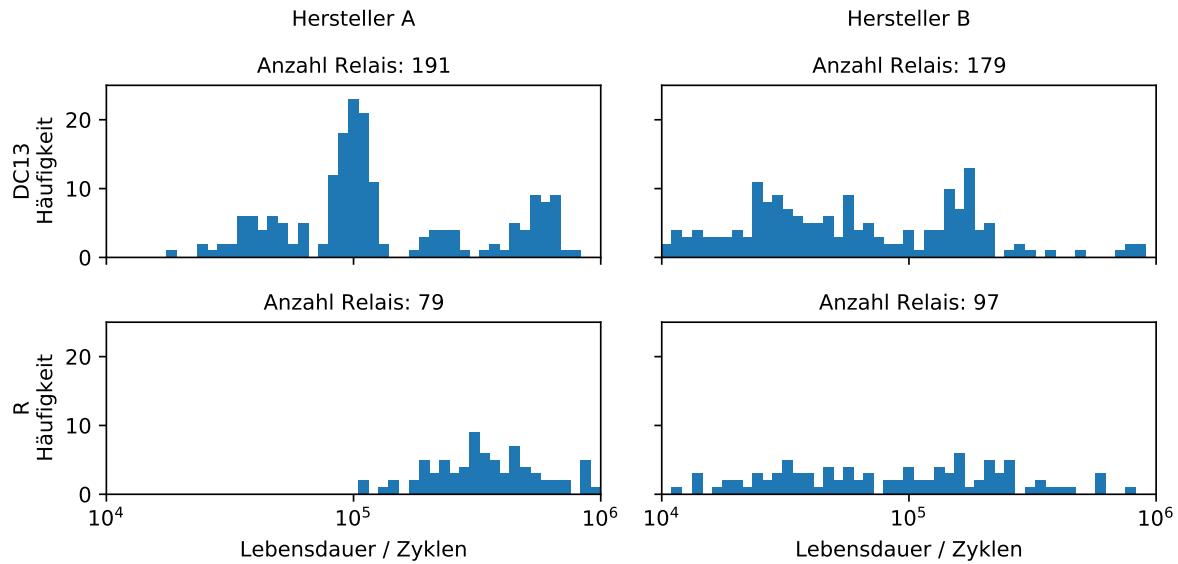


Abb. 3.30: Häufigkeitsverteilung der Relaisdefekte, unterschieden nach Hersteller und Last [89]

Häufigkeitsverteilungen wird ein Problem deutlich: Die Verteilungen sind äußerst komplex, was eine zuverlässige Schätzung erschwert.

Auch wenn dies gelingen würde, läge die Differenz zwischen kürzester und längster Lebensdauer der Relais bei allen Kombinationen über 900.000 Schaltzyklen. Deshalb würde auch mit einem perfekten B10 Wert einen Großteil der Lebensdauer der Relais, die vor ihrem Ausfall gewechselt werden, nicht genutzt werden. Deshalb erscheint eine statistische Modellierung für elektromechanische Relais nicht sinnvoll.

3.5.2 Überwachte Lernverfahren

In diesen Abschnitt wird erklärt, wie überwachte Lernverfahren zur Instandhaltung von Bauteilen genutzt werden können und wieso dies für Relais nicht anwendbar ist. In Abs. 3.1.2 wurden bereits einige Verfahren, die diesem Vorgehen entsprechen. Besonders hervorzuheben ist bspw. die Schätzung der „Remaining useful lifetime“ (dt. nutzbare Restlebensdauer) (RUL) von Triebwerken, Kugellagern und Bohrern. Zu diesen Bauteilen und öffentlichen Datensätzen wurden bereits zahlreiche Veröffentlichungen gemacht, die das hohe Potenzial von maschinellen Lernverfahren aufzeigen. Deshalb wurde durch das Unternehmen Phoenix Contact Forschung zur Übertragung dieser Verfahren auf elektromechanische Relais durchgeführt. Die Ergebnisse sind u.a. in meiner nicht-veröffentlichten Masterarbeit [45] dokumentiert und werden deshalb nachfolgend erörtert.

Zunächst wird auf die genutzte Definition der RUL eingegangen. Die Degradierung der Relais ist wesentlich abhängig von der Anzahl der Schaltvorgänge, weshalb diese bereits beim B10 Wert als Maßeinheit für die Lebens- bzw. Betriebsdauer genutzt und im Kontext

dieser Arbeit übernommen wird. Da die Daten alle Schaltzyklen bis zum Ausfall von allen Relais enthalten, können zu jedem Schaltzyklus die verbleibenden Schaltzyklen bis zum Ausfall angegeben werden. Eine derartige absolute Angabe birgt jedoch ein Problem: Den Daten ist zu entnehmen, dass u.a. der Laststrom einen großen Einfluss auf die Lebensdauer hat. Je mehr Strom fließt, desto schneller degradiert das Relais. Die Nutzung von einer absoluten Restlebensdauer würde eine Zuordnung von unterschiedlichen Relaiszuständen zu einem Zielwert implizieren: Wenn die Lebensdauer von Relais A 250.000 Schaltzyklen und von Relais B 50.000 beträgt und beide Relais aufgrund von Materialabbrand ausfallen, dann würde Relais A bei einer Restlebensdauer von 40.000 Schaltzyklen deutlich stärkere Verschleißerscheinungen aufweisen als Relais B. Deshalb wird eine relative Restlebensdauer $\text{RUL}(m)$ für den Schaltvorgang m wie folgt definiert:

$$\text{RUL}(m) = 1 - \frac{m}{m_{\max}}, \quad (3.19)$$

wobei m_{\max} die Lebensdauer des Relais ist. Auf diesem Weg wird der Verschleiß von den Relais A und B vergleichbar: Eine RUL von 50 % entspricht bei A 125.000 und bei B 25.000 Schaltzyklen.

Eine relative Angabe ist nicht direkt nutzbar, 20 % RUL kann – je nach Relais – für eine beliebige Restlebensdauer stehen. Deswegen muss die geschätzte $\hat{\text{RUL}}(m)$ zur geschätzten Lebensdauer \hat{m}_{\max} umgerechnet werden. Durch Umformen von Gl. 3.19 ergibt sich:

$$\hat{m}_{\max} = \frac{m}{1 - \hat{\text{RUL}}(m)}, \quad (3.20)$$

für diese Berechnung muss der Schaltzyklus m bekannt sein. Mit Gl. 3.20 sind die geschätzten restlichen Schaltzyklen \hat{m}_{rest} für ein Relais wie folgt zu berechnen:

$$\hat{m}_{\text{rest}} = \hat{m}_{\max} - m = \frac{m}{1 - \hat{\text{RUL}}(m)} - m. \quad (3.21)$$

Im Rahmen meiner Masterarbeit wurden verschiedenste maschinelle Lernverfahren hinsichtlich ihrer Eignung für die Prädiktion der RUL untersucht. Dies wurde im Kontext dieser Dissertation fortgesetzt und vertieft. Dabei konnte ein Problem jedoch nicht gelöst werden: Die Schätzung der RUL ist für einige Relais so ungenau, dass die Modellbildung durch die maschinellen Lernverfahren nicht als erfolgreich betrachtet werden kann. Dieses Problem trat unabhängig von den maschinellen Lernverfahren auf. Simple lineare Modelle scheiterten genauso wie komplexe rekurrente neuronale Netze.

Daher wird das Problem anhand der Prädiktionen zu den in Abs. 3.4.1 vorgestellten Relais verdeutlicht. Dazu wurde ein ANN mit der Topologie aus Tab. 3.6 mit den in

Abs. 3.4.3 präsentierte Merkmale als Eingangsdaten und der relativen RUL als Zielgröße trainiert. Für das Training wurde der Adam Optimierer mit einer Lernrate von 0.001 gewählt, um den „Mean Squared Error“ (dt. Mittlerer quadrierter Fehler) (MSE) zu minimieren [93]. Durch eine Gewichtung der Trainingsdaten wurde sichergestellt, dass die Relais gleichbedeutend sind. Ein Schaltzyklus von einem Relais mit hoher Lebensdauer ist dementsprechend geringer gewichtet als der eines Relais mit niedriger Lebensdauer. Zusätzlich wurde das Training unterbrochen, als die Performanz auf dem Trainingsset besser und auf dem Validierungsset schlechter wurde.

Tab. 3.6: ANN-Topologie

Schicht	Typ	Informationen
1	Dense	Neuronen: 50, Aktivierungsfunktion: ReLU
2	Dropout	Dropout-Rate: 0.1
3	Dense	Neuronen: 50, Aktivierungsfunktion: ReLU
4	Dropout	Dropout-Rate: 0.1
5	Dense	Neuronen: 1, Aktivierungsfunktion: ReLU

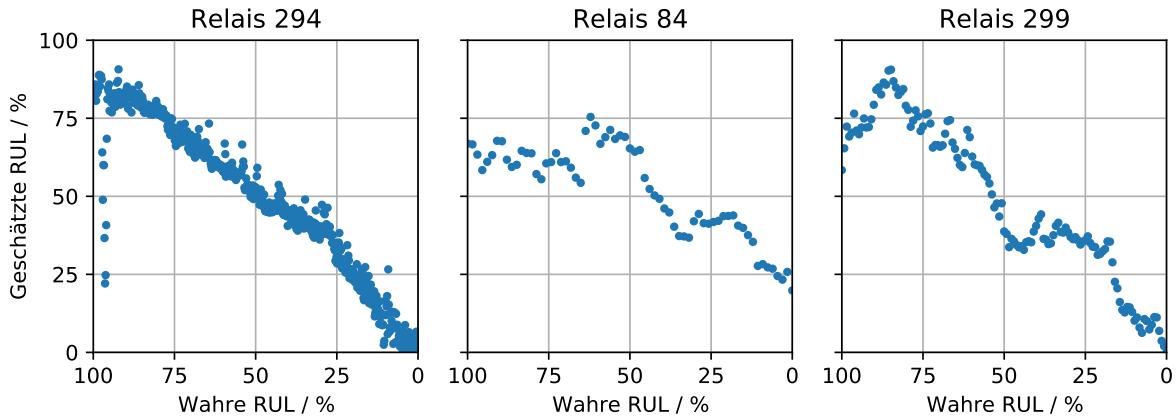


Abb. 3.31: Beispielhafte RUL-Prädiktionen von drei Relais mit einem ANN

In Abb. 3.31 sind die Prädiktionen des ANN für drei Relais aus dem Testset visualisiert. Diese sind nicht repräsentativ für die Performanz auf dem gesamten Testset, lassen jedoch methodische Probleme aufzeigen.

Relais 294 weist monotone Trends in den Messdaten auf (vgl. Abb. 3.21). Dies spiegelt sich in den Prädiktionen wider: Über die gesamte Betriebsdauer sind nur geringe Abweichungen zwischen der geschätzten und wahren RUL auszumachen. Die geschätzte RUL wäre geeignet gewesen, um einen rechtzeitigen Hinweis für eine Instandhaltung zu geben. Zu Beginn der Betriebsdauer (RUL 90 %) sind erhebliche Abweichungen zu erkennen. Dies ist zum einen mit dem sichtbaren anormalen Verhalten in Abb. 3.21 zu begründen. Zum anderen treten bei vielen Relais Abweichungen in der Form auf, dass die RUL nie auf 100 %

geschätzt wird. Es ist davon auszugehen, dass die Relais initial nicht alle die gleiche Lebenserwartung haben. Ein wichtiger Faktor dafür ist z.B. der Überhub, der aufgrund von Fertigungsabweichungen nicht bei allen Relais gleich ist.

Für Relais 84 ergeben sich andere Herausforderungen: Die geschätzte RUL liegt über die gesamte Betriebsdauer zwischen 80 und 20 %. Die erste Lebenshälfte ist keine Abnahme der geschätzten RUL zu erkennen, ab der zweiten Lebenshälfte ist eine Abnahme zu beobachten. Dieses Verhalten kann durch Betrachtung von Abb. 3.22 nachvollzogen werden: Hier ist auszumachen, dass nur in der zweiten Lebenshälfte Veränderungen in den Messgrößen von Relais 84 aufgetreten sind. Das Abweichen von der vorgegebenen linearen RUL, ist somit nachvollziehbar. Dadurch scheint die Annahme, dass die RUL durch eine lineare Funktion approximiert werden kann, nicht anwendbar zu sein.

Dies wird noch deutlicher bei Relais 299, bei dem in Abb. 3.23 reversible Veränderungen auszumachen sind. Die Prädiktionen des ANN weichen stark von der wahren RUL ab. Generell ist zu erkennen, dass mit zunehmender Betriebsdauer die geschätzte RUL abnimmt. Eine Ausnahme bildet das Intervall von 90 bis 70 % der wahren RUL, hier steigt die geschätzte RUL sogar an. Die angestrebte Modellierung der RUL wurde somit nicht erreicht. Neben dem Nichterfüllen der Linearität wird hier eine andere implizite Annahme bei der RUL verletzt: Die Daten lassen sich – insbesondere in den Bereichen mit reversiblen Veränderungen – nicht mit einer monotonen RUL vereinen.

Zusammenfassend ist festzuhalten, dass die lineare, monotone RUL nicht auf Relais angewandt werden kann. In diversen Tests mit unterschiedlichsten Techniken des maschinellen Lernens gelang kein erfolgreiches Lernen einer solchen RUL. Begründet werden kann dies mit den Veränderungen in den Daten, die oftmals nicht linear und / oder monoton sind. Gleichzeitig kann durch die Untersuchung gezeigt werden, dass durch das ANN Rückschlüsse auf den Zustand des Relais gezogen werden können, da grundsätzlich eine niedrigere RUL mit höherer Betriebsdauer geschätzt wird.

3.5.3 Unüberwachte Lernverfahren

Unüberwachte Lernverfahren bringen einen entscheidenden Vorteil mit sich: Es ist keine Modellierung der Zielgröße notwendig.

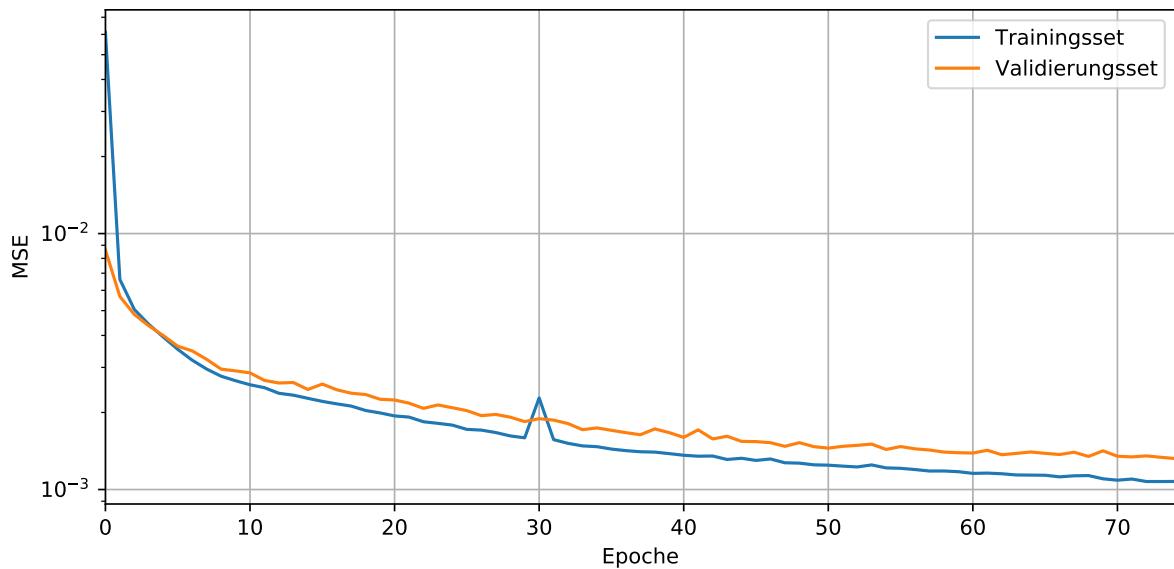
Es bestehen diverse Verfahren, mit denen unüberwacht prädiktive Instandhaltung realisiert werden kann, wie bereits im Forschungsstand (Abs. 3.1.2) erörtert wurde. Doch auch dieser Ansatz ist auf Relais nicht übertragbar. Um dies zu erklären, wird im Folgenden ein „Variational Autoencoder“ (VAE) exemplarisch genutzt. Mit diesem wird unüberwacht eine latente Darstellung der Eingangsdaten erlernt, auf deren Basis wiederum eine Rekonstruktion der Eingangsdaten stattfindet. Über Rekonstruktionsfehlers werden schlussendlich anormale Schaltzyklen ausgemacht.

Eine detaillierte Beschreibung ist Abs. 3.2.2 zu entnehmen. Die Topologie des Netzes ist in Tab. 3.7 aufgeführt, sowohl die Conv1D als auch die Conv1DTranspose Schichten wurden mit einer LekyReLU Aktivierungsfunktion versehen. Für das Training wurde der

Tab. 3.7: VAE Topologie

Schicht	Typ	Informationen
1	Conv1D	Filter: 1, Strides: 1
2	Conv1D	Filter: 32, Strides: 2
3	Conv1D	Filter: 32, Strides: 1
4	Conv1D	Filter: 16, Strides: 2
5	Conv1D	Filter: 2, Strides: 2
7	Dense	Neuronen: 100
8	Lambda	Stichprobenverfahren
9	Dense	Neuronen: 50
11	Conv1DTranspose	Filter: 64, Strides: 2
12	Conv1DTranspose	Filter: 64, Strides: 2
13	Conv1DTranspose	Filter: 64, Strides: 1
14	Conv1DTranspose	Filter: 16, Strides: 2
15	Conv1DTranspose	Filter: 8, Strides: 1
16	Conv1DTranspose	Filter: 4, Strides: 1
17	Conv1DTranspose	Filter: 4, Strides: 1

Adam Optimierer genutzt [93] und als Ein- bzw. Ausgangsdaten die vier aufgezeichneten Zeitreihen, also viermal 200 Werte pro Schaltzyklus. Die Daten wurden für das Training in das Intervall $[0,1]$ skaliert, dazu wurden 0 mA bzw. 0 V als Minimalwert und 8 mA bzw. 30 V als Maximalwert verwendet. Deshalb ist das Fehlermaß (der MSE) dimensionslos. Insgesamt wurde der VAE 75 Epochen trainiert.

**Abb. 3.32:** Fehler für das Trainings- und Validierungsset über die Trainingsepochen

In Abb. 3.32 sind die Fehler des Trainings- und Validierungssets über die Lernepochen aufgetragen. Da der Fehler für beide Sets gleichermaßen mit zunehmender Epoche abnimmt, ist das Training des VAE erfolgreich.

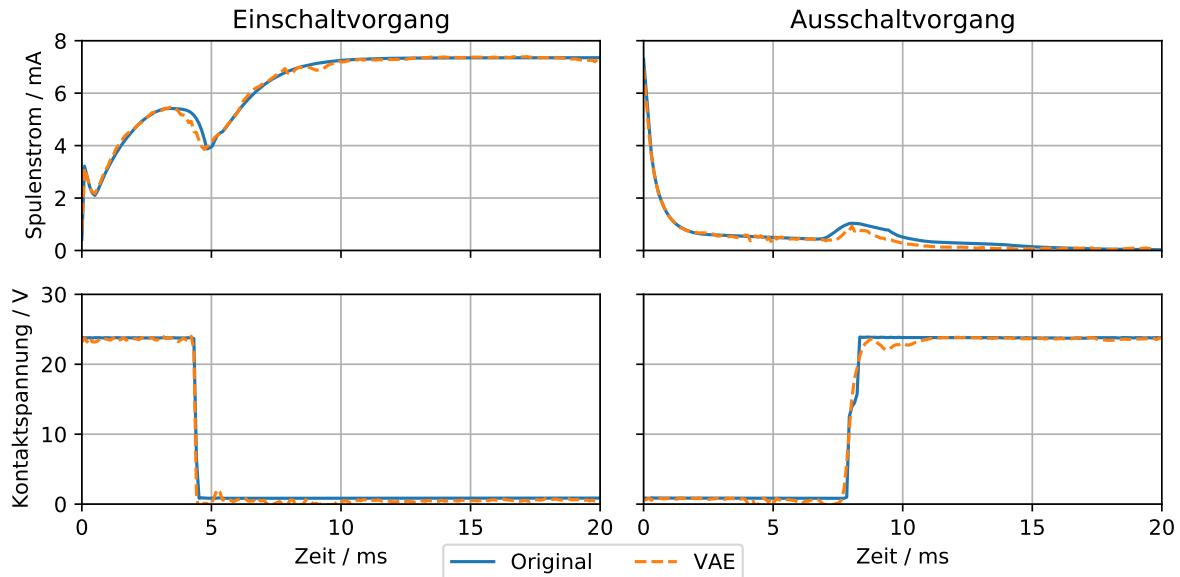


Abb. 3.33: Vergleich der originalen und vom VAE rekonstruierten Zeitreihen

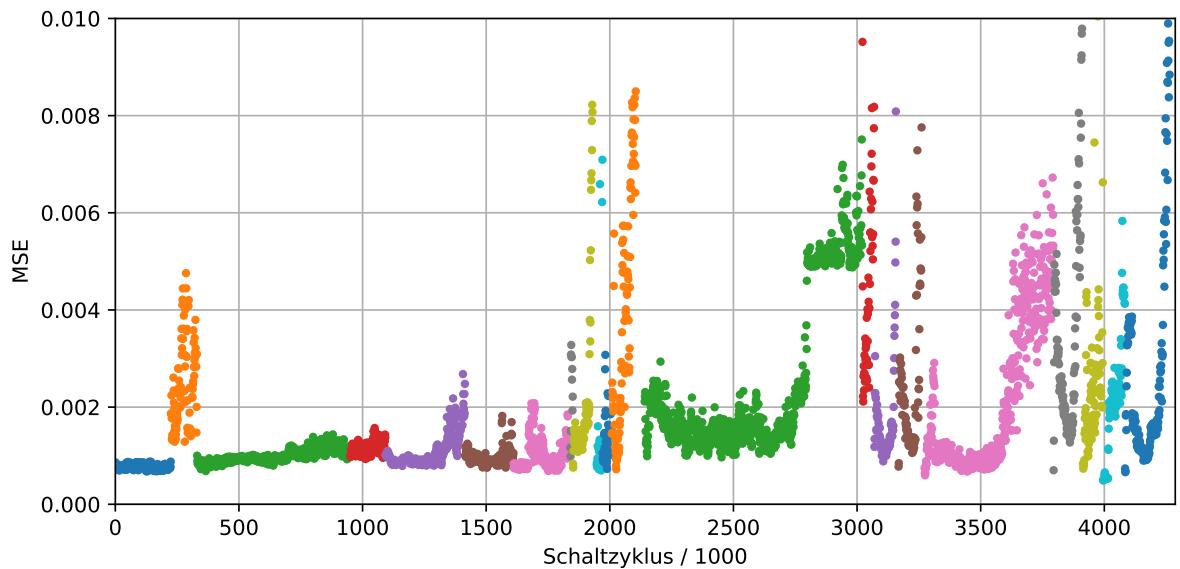


Abb. 3.34: MSE der rekonstruierten Zeitreihen von verschiedenen Relais [6]

In Abb. 3.33 sind exemplarische originale und rekonstruierte Messreihen gegenübergestellt. Die Zeitpunkte von Ankeraufschlag, Ankerrückbewegung und den Kontaktierungsände-

rungen sind präzise rekonstruiert worden. In den Zeitreihen sind nur teilweise geringe Abweichungen auszumachen. Somit war das Training des VAE erfolgreich.

Die Rekonstruktionsfehler wurden als MSE berechnet und sind in Abb. 3.34 exemplarisch für einige Relais aus dem Testset dargestellt. Die Schaltzyklen der Relais wurden nach ihrer Reihenfolge mit Punkten dargestellt. Die einzelnen Relais sind durch die Farbwechsel voneinander zu unterscheiden. Um einen VAE für PdM zu nutzen, müsste an dieser Stelle ein Fehlerschwellwert festgelegt werden, mit dem die Anomalien bestimmt werden können. Bei einigen Relais ist ersichtlich, dass der MSE zu Ende der Betriebsdauer stark ansteigt. Bei einigen Relais ist dieser jedoch auch zu Beginn der Betriebsdauer hoch. Es ist nicht möglich, eine Grenze festzulegen, zu der bei vielen Relais vor einem Ausfall gewarnt und die gleichzeitig nicht von Beginn der Betriebsdauer an bei den Relais überschritten wird.

In Abb. 3.35 wird das Problem der Schwellwertbestimmung anhand von Box-Polts für die Relais visualisiert. Die Whisker geben dabei jeweils die minimal, bzw. maximalen MSE des Relais an. Die Box umfasst das 25 bis 75 % Intervall und als oranger Strich ist der Median dargestellt. Aus der Grafik geht hervor, dass der MSE mehr von den Relais als von deren Degradierungszustand abhängt. Für die Wahl eines Schwellwerts ist es notwendig, dass die 25 bis 75 % Intervalle ähnlich sind, damit eine klare Trennung von normalen und anomalen Schaltzyklen durchgeführt werden kann.

Das Scheitern des Verfahrens auf Basis eines VAE kann auf zwei Arten und Weisen begründet werden: Zum einen müssten vor dem Ausfall eines Relais anomale Schaltzyklen auftreten, die einen hohen Rekonstruktionsfehler begründen. Aus Abb. 3.21 geht jedoch hervor, dass Relais kontinuierlich degradieren können, weshalb ein Defekt sich nicht durch ein anomales Verhalten ankündigt. Bei den Relais aus den Abbildungen 3.22 und 3.23 sind sprunghafte, reversible Veränderungen zu beobachten. Daher sind diese im Datensatz nicht unterrepräsentiert und der VAE erreicht einen geringen MSE bei der Rekonstruktion. Zum anderen führen Fertigungsabweichungen dazu, dass die Formen der Messreihen für jedes Relais individuell sind. Einen Großteil lernt der VAE nachzubilden; der MSE ist entsprechend gering. Aber einzelne Relais können aufgrund ihrer individuellen Form über ihre ganze Betriebsdauer nicht so präzise durch den VAE verarbeitet werden, weshalb der MSE höher ist. Durch die Nutzung des VAE lassen sich in diesem Fall somit andere Rückschlüsse ziehen: Nicht die Degradierung wird durch den MSE angezeigt, sondern ein generell anomales Relais.

Damit scheint es sinnvoll zu sein, einen VAE beim Hersteller zu nutzen, um eine Qualitätskontrolle der Relais durchzuführen. Rückschlüsse auf den Degradierungszustand der Relais können – wie in diesem Abschnitt gezeigt – jedoch nicht gezogen werden.

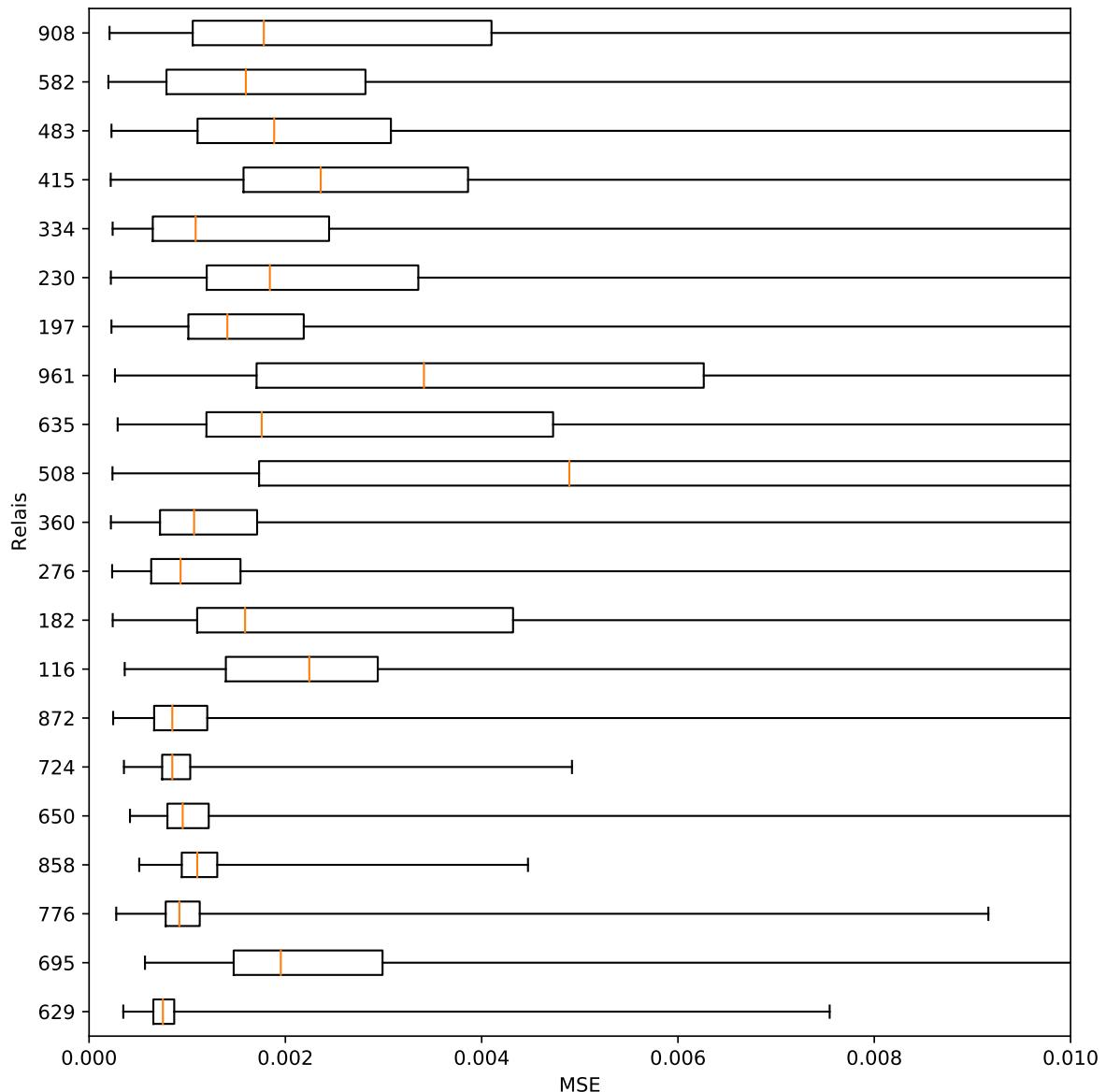


Abb. 3.35: Box-Plot Diagramme der MSE von einzelnen Relais

3.5.4 Aufzeigen des Potenzials für Prädiktive Instandhaltung

In den vorangegangenen Abschnitten wurde gezeigt, warum überwachte und unüberwachte maschinelle Lernverfahren nicht erfolgreich für die prädiktive Instandhaltung angewendet werden können. Trotzdem scheint ein gewisses Potenzial zu bestehen, das nachfolgend aufgezeigt wird. Dazu wird eine Untersuchung meiner Masterarbeit [45] wiederholt, bei der eine Unterscheidung von zwei Degradierungszuständen der Relais mit den vorliegenden Daten und maschinellen Lernverfahren durchgeführt wird.

Dafür werden von jedem Relais des Datensatzes die in Abs. 3.4.3 beschriebenen Merkmale von den ersten 720 Schaltzyklen als „unauffällige“ Beispiele und die letzten 360 Schaltzyklen als „auffällige“ Beispiele gewählt. Auf diesem Weg wird sichergestellt, dass sich die Merkmale zwischen beiden Degradierungszuständen deutlich unterscheiden. Für die Klassifizierung wird ein ANN gemäß der in Tab. 3.6 beschriebenen Topologie genutzt. Das Training wird mit vorzeitigem Abbruch und dem Adam Optimierer durchgeführt [93], wobei der Datensatz gemäß der Vorgabe aus Abs. 3.4.4 aufgeteilt wird. Als Fehlermaß wird die binäre Kreuzentropie genutzt. Die Performanz wird am relativen Anteil der korrekt klassifizierten Schaltvorgänge an den gesamten Schaltvorgängen gemessen und liegt für den Testdatensatz bei 96,07 %. Daraus folgt, dass eine Unterscheidung der Degradierungsstufen durch ein ANN möglich ist.

Demzufolge ist ein semi-überwachtes Training, speziell Pseudo-Labeling, passend für die prädiktive Instandhaltung von Relais, bei denen Millionen Schaltzyklen unmarkiert sind. Eine vergleichbare Anwendung wird bereits in [94] beschrieben, wo eine Fehlerdiagnose von Windkraftanlagengetrieben präsentiert wird. Auch hier besteht das Problem, dass es schwierig ist, markierte Daten zu erhalten, die für maschinelles Lernen notwendig sind. Als Lösung des Problems wird über eine semi-überwachte Lernmethode vorgeschlagen. Zunächst werden dabei ANN mit den begrenzten markierten Daten trainiert. Danach werden auf Basis von deren Prädiktionen weitere Daten markiert. Schließlich wird der erweiterte markierte Datensatz verwendet, um die Merkmalsextraktion und das ANN zu optimieren bzw. zu trainieren. Die Experimente – sowohl auf dem Prüfstand als auch bei der Fehlerdiagnose von Windturbinengetrieben – zeigten, dass effektiver ist als vergleichbare Methoden.

3.5.5 Forschungsdesiderat

In den vorangegangenen Abschnitten wurde zunächst erläutert, dass statistische Verfahren ungeeignet sind, um die Instandhaltung eines Relais zu planen, da es sowohl bei den notwendigen Datenmengen als auch bei der Modellierung Probleme gibt. Weiterhin wurde gezeigt, dass der Stand der Technik im Kontext von PdM nicht auf Relais angewandt werden kann. Überwachte Lernverfahren benötigen eine Zielgröße, die im Kontext der nicht linearen und nicht monotonen Degradierung individuell für jedes Relais definiert werden müsste. Durch unüberwachte Verfahren kann dieses Problem umgangen werden, doch hier ergibt sich ein neues: Die zentrale Annahme, dass Anomalien im Datensatz unterrepräsentiert sind, ist bei Relais nicht erfüllt.

Schlussendlich wurde jedoch gezeigt, dass die Daten eine Unterscheidung von „unauffällige“ und „auffällige“ Schaltzyklen durch ein ANN erfolgreich erlernt werden kann. Deshalb wird im nachfolgenden Abschnitt ein neuer semi-überwachter Ansatz vorgestellt.

3.6 Semi-überwachtes Lernverfahren für PdM

Im Folgenden wird ein neues Verfahren vorgestellt, mit dem ein Indikator für einen Ausfall gegeben werden kann. Dabei wird ein semi-überwachter Lernansatz genutzt, mit dem auffällige Schaltspiele von Relais markiert werden und so ein rechtzeitiger Service ermöglicht wird. Die Funktionsweise des Algorithmus wird sowohl qualitativ für die Daten einzelner Relais als auch quantitativ demonstriert. In der Veröffentlichung [89] ist das Verfahren bereits beschrieben und evaluiert. Weiterhin sind zu dieser Thematik zwei Patente angemeldet [95] und [96].

3.6.1 Instandhaltungsalgorithmus für ungelabelte Daten

Mit dem Verfahren „Maintenance-Algorithm for Unlabeled Data“ (dt. Instandhaltungsalgorithmus für ungelabelte Daten) (MAUD) kann ein Indikator für einen Ausfall eines Relais gegeben werden. Damit ist es möglich, einen drohenden Ausfall zu verhindern.

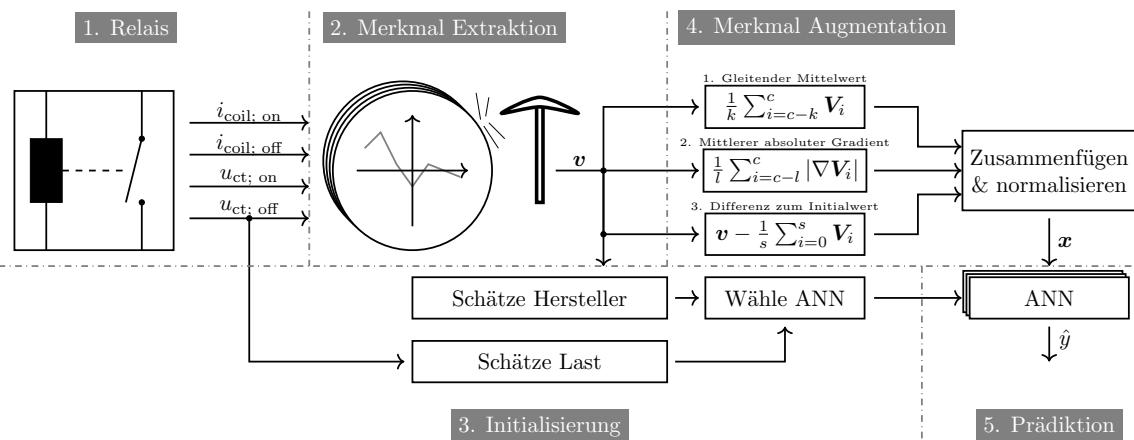


Abb. 3.36: Überblick über die Datenverarbeitungspipeline [89]

MAUD setzt sich aus insgesamt fünf Teilen zusammen, die in Abb. 3.36 in Verhältnis zueinander gesetzt werden. Die Aufteilung röhrt daher, dass Hersteller und Last einen Einfluss auf die Degradierung haben und deswegen getrennt voneinander betrachtet werden. Dafür ergibt sich die Notwendigkeit, eine Initialisierungsphase einzuführen, in der Hersteller und Last klassifiziert werden. Die Initialisierungsphase umfasst die ersten zwei Betriebsstunden (720 Zyklen) eines jeden Relais. In dieser Phase wird angenommen, dass kein Relais ausfällt. Danach kann MAUD genutzt werden, um den Zustand des jeweiligen Relais zu schätzen.

Der erste Teil „Relais“ umfasst die Datenakquisition und wurde damit in Abschn. 3.3 bereits ausführlich beschrieben, genauso wie die Extraktion der Merkmale (Schritt 2) in Abschn. 3.4.3. Die übrigen drei Teile werden nachfolgend erörtert. Zunächst wird beschrieben, wie die Unterscheidung von Hersteller und Last realisiert wird. Darauf wird

die Aufbereitung der Merkmale erklärt und schlussendlich, wie das ANN trainiert und zur Prädiktion genutzt wird.

3.6.1.1 Initialisierung

Da Hersteller und Last unabhängig voneinander sind, werden beide ebenfalls eigenständig geschätzt.

Schätzung des Herstellers

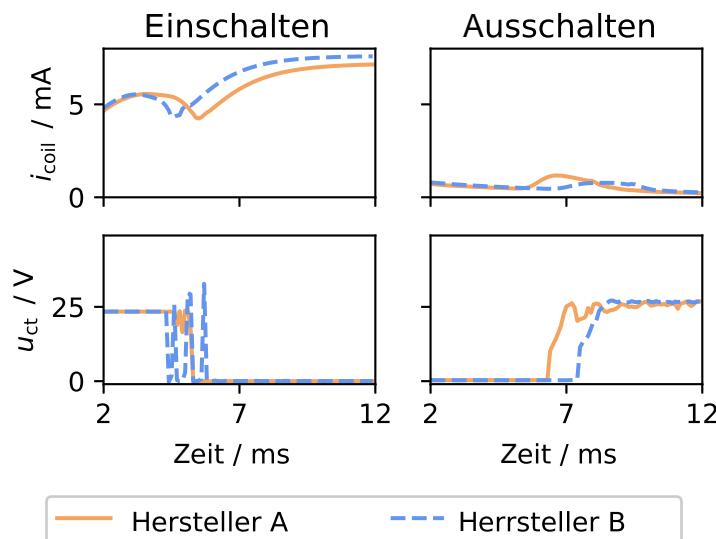


Abb. 3.37: Vergleich von Messreihen von zwei unterschiedlichen Herstellern [89][96]

Die Relais werden nach Herstellern unterschieden. In Abb. 3.37 sind die Messreihen für zwei neue Relais von beiden Herstellern dargestellt. Bei i_{coil} sind Unterschiede beim Einschalten und Ausschalten zwischen beiden Herstellern zu sehen. Diese sind auf die Geometrie des Magnetkreises und die Induktivität der Spule zurückzuführen. Auch die Federkräfte der Kontakte wirken sich auf die Bewegung des Ankers aus. Beim Einschalten zeigt u_{ct} , dass das Relais von Hersteller B stärker prellt – eine Beobachtung, die für die Mehrzahl der Relais zutrifft. Beim Ausschalten ist in u_{ct} zu erkennen, dass sich die Kontakte zu unterschiedlichen Zeitpunkten trennen.

Durch diese Unterschiede in den Messreihen scheint eine Differenzierung der Hersteller möglich zu sein. Deswegen werden die in Abs. 3.4.3 beschriebenen Merkmale verwendet, um ein ANN zu trainieren, das diese Klassifizierungsaufgabe übernimmt.

Beim Test verschiedener Topologien zeigte sich, dass selbst ein einfaches Netz mit einer versteckten Schicht aus 25 Neuronen und einer ReLU-Aktivierungsfunktion und einem Neuron als Ausgang mit Sigmoid-Aktivierungsfunktion eine Genauigkeit von 99,99 % im Trainingsdatensatz und 99,90 % im Validierungsdatensatz erreicht. Die Initialisierungsphase von MAUD dauert 720 Schaltzyklen, weshalb über diese eine Mehrheitsabstimmung

durchgeführt werden kann. Dies ermöglicht ein zu 100 % korrektes Klassifizieren der Relais im Testdatensatz zu dem richtigen Hersteller.

Die Unterscheidung verschiedener Relais-Hersteller unter Zuhilfenahme von maschinellen Lernverfahren ist in [96] als Patent angemeldet. Bei einer Produktumsetzung könnten dieses Verfahren auch genutzt werden, um Relais von unbekannten Herstellern zu erkennen und MAUD zu deaktivieren. Schließlich ist deren Degradierung nicht untersucht worden.

Schätzung der Last

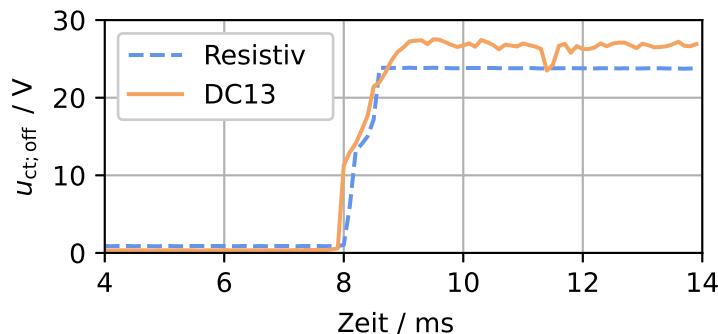


Abb. 3.38: Kontaktspannung beim Ausschalten für unterschiedliche Lasten [89]

Da je nach geschalteter Last unterschiedliche physikalische Phänomene auftreten, sind unterschiedliche Arten und Weisen der Degradierung zu erwarten. Daher werden die Relais nach Last differenziert. Dazu wird u_{ct} beim Ausschalten verwendet: Bei den induktiven DC13-Lasten treten Lichtbögen auf, die sich durch Rauschen in der Kontaktspannung bemerkbar machen. Dies ist in Abb. 3.38 zu sehen. Für die Klassifikation der Lasten genügt es, die Standardabweichung der Spannung die ersten 5 ms direkt nach dem Trennen der Kontakte zu betrachten: Für die DC13-Last beträgt diese 3,25 V und für die resistive Last 0,03 V. Auf diesem Weg werden beide Lastarten voneinander unterschieden.

Im Rahmen einer Produktumsetzung kämen zwar viele weitere Lastarten in betracht, die Unterscheidung zwischen induktiven und nicht induktiven Lasten wäre jedoch weiterhin möglich.

3.6.1.2 Merkmalsaugmentation

Die in Abs. 3.4.3 extrahierten n Merkmale werden im Folgenden mit $\mathbf{v} \in \mathbb{R}^n$ für einen Schaltzyklus eines Relais bezeichnet. Auf dieser Grundlage findet eine Erweiterung des Datensatzes statt. Die extrahierten Merkmale beziehen sich nur auf einen Schaltvorgang – die Veränderungen zwischen den Schaltvorgängen oder zum initialen Zustand des Relais gehen nicht daraus hervor.

Daher wird beim Augmentieren die Historie der Merkmale $\mathbf{V} \in \mathbb{R}^{n \cdot l_r}$ von jedem Relais $r \in \mathbb{N}$ einbezogen – wobei $l_r \in \mathbb{N}$ die Anzahl der Schaltzyklen ist und $c_r \in \mathbb{N}$ ein

bestimmter Zyklus ist. Jedes Merkmal erhält in drei modifizierten Varianten Einzug in den Trainingsdatensatz, sodass dieser hinsichtlich der Dimensionalität verdreifacht wird. Folgende Modifikationen werden vorgenommen:

1. Gleitender Durchschnitt: $\frac{1}{k} \sum_{i=c-k}^c \mathbf{V}_i$
Um Ausreißer und Rauschen in den Merkmalen zu unterdrücken, wird ein gleitender Mittelwert mit einer Fensterbreite von $k \in \mathbb{N}$ bestimmt.
2. Maß für die Größe der Veränderung: $\frac{1}{m} \sum_{i=c-m}^c |\nabla \mathbf{V}_i|$
Der Gradient eines Merkmals ist ein wichtiger Hinweis für dessen Änderung. Über die Berücksichtigung des mittleren Gradienten (gemittelt über $m \in \mathbb{N}$ Zyklen) wird dem ANN diese Information über die jüngste Historie des Merkmals gegeben.
3. Abweichung vom Ausgangszustand: $\mathbf{v} - \frac{1}{s} \sum_{i=0}^s \mathbf{V}_i$.
Die initialen Zustände der Relais weichen u.a. aufgrund von Fertigungsabweichungen voneinander ab. Somit ist es nicht möglich, die Werte der Merkmale eindeutig einem Alter zuzuordnen. Deshalb wird die Veränderung eines jeden Relais zu diesem individuellen initialen Zustand im Datensatz aufgenommen. Diese wird über die ersten $s \in \mathbb{N}$ Zyklen gemittelt und von den Merkmalen abgezogen.

Für das Training wird mit den Merkmalen abschließend eine Z-Normalisierung durchgeführt. Der Mittelwert und die Standardabweichung werden dabei über den gesamten Trainingssatz berechnet.

3.6.1.3 Prädiktion

Topologie des verwendeten ANN

Nachdem Hersteller und Last geschätzt und die Merkmale augmentiert wurden, kommt es im fünften Schritt zur Prädiktion. Dabei wird – je nach Hersteller und Last – ein eigenes ANN genutzt, da die Degradierung stark von diesen Faktoren abhängt und eine Steigerung der Performanz durch das Verwenden unterschiedlicher ANN erzielt werden kann. In Hinblick auf Topologie und Training sind die ANN identisch.

Im Rahmen dieser Arbeit wird das einfachste mögliche ANN verwendet, d.h. einfache Neuronen mit ReLU-Aktivierungsfunktion. Der Vorteil dieses Ansatzes besteht darin, dass ein solches ANN bei der Inferenz auf Mikrocontrollern nur geringe Hardware-Anforderungen hat. Die ANN-Topologie wird in Tab. 3.8 angegeben und wurde mit Hilfe einer Hyperparameter-Optimierung bestimmt. Dazu wurden die Anzahl der Neuronen pro Schicht, die Anzahl der Schichten, die Aktivierungsfunktion und die Dropout-Rate mit einer Zufallssuche variiert und die Kombination mit der besten Performanz gewählt:

Training

Ziel ist es, eine binäre Klassifizierung der Schaltzyklen durchzuführen, mit der die Identifizierung von Anzeichen eines Ausfalls ermöglicht werden soll. Eine manuelle Zuordnung des Label für Anzeichen eines Ausfalls ist jedoch nicht möglich, da der Datensatz mit über 100 Millionen Schaltzyklen zu umfangreich ist. Daher wird ein semi-überwachter

Tab. 3.8: ANN-Topologie [89]

Schicht	Typ	Informationen
1	Dense	Neuronen: 50, Aktivierungsfunktion: ReLU
2	Dense	Neuronen: 50, Aktivierungsfunktion: ReLU
3	Dense	Neuronen: 50, Aktivierungsfunktion: ReLU
4	Dropout	Dropout-Rate: 0.25
5	Dense	Neuronen: 1, Aktivierungsfunktion: Sigmoid

Lernansatz verfolgt, bei dem ein Teil der Label vom ANN selbst vergeben werden. Speziell wird ein Pseudo-Labeling angewandt – das Prozedere hierfür ist schematisch in Abb. 3.39 dargestellt.

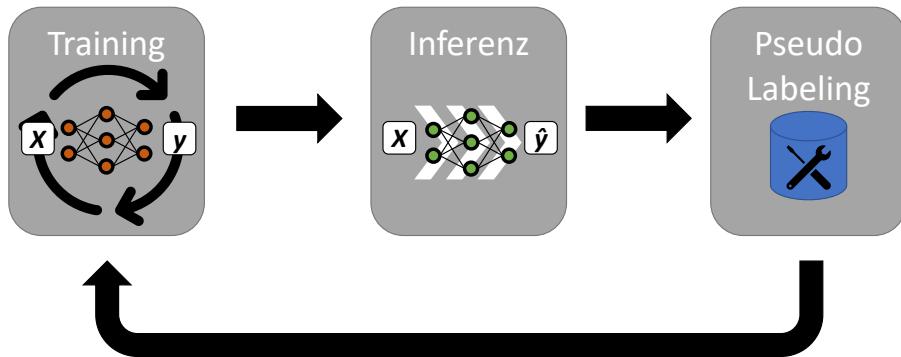


Abb. 3.39: Trainingsprozedur [89]

Bei der in Abb. 3.39 dargestellten Trainingsprozedur handelt es sich um einen iterativen Prozess, dem eine Initialisierung vorgelagert ist. Für diese werden zwei Annahmen getroffen: Einerseits wird davon ausgegangen, dass die letzten Zyklen auffällig sind und daher als solche gekennzeichnet werden können. Dazu wird $\lambda \in \mathbb{N}$ in die Gleichungen 3.22 und 3.24 eingeführt. Andererseits wird angenommen, dass die meisten Zyklen in der ersten Hälfte der Betriebsdauer des Relais unauffällig sind. Dies wird durch σ berücksichtigt. Für das erste Training werden die Label \mathbf{y} für den Schaltzyklus c_r dementsprechend folgendermaßen vorgegeben:

$$\mathbf{y}_{c_r} = \begin{cases} 0 \text{ (unauffällig)} & \text{wenn } c_r < \sigma \\ 1 \text{ (auffällig)} & \text{wenn } c_r > l_r - \lambda, \\ \text{nicht genutzt} & \text{sonst} \end{cases} \quad (3.22)$$

wobei L_r für die Lebensdauer des Relais r und c_r für den jeweiligen Zyklus steht. Mit diesen Labeln wird ein erstes ANN mit der in Tab. 3.8 aufgeführten Topologie trainiert. Um die Überrepräsentation eines Label im Datensatz zu vermeiden, werden für beide Label die gleiche Anzahl an Beispielen zufällig aus dem Datensatz gezogen. Als Optimierer

wird der Adam Optimizer genutzt [93]. Zusätzlich wird das Training vorzeitig abgebrochen, um eine Überanpassung zu vermeiden. Der Fehler beim Training wird über die binäre Kreuzentropie berechnet:

$$H_p(q) = -\frac{1}{N} \sum_{i=1}^N \mathbf{y}_i \cdot \log(p(\mathbf{y}_i)) + (1 - \mathbf{y}_i) \cdot \log(1 - p(\mathbf{y}_i)). \quad (3.23)$$

Auf diese Art und Weise wird in der Initialisierung ein ANN erzeugt, mit dem beim Pseudo-Labeling neue Label für den Trainings- und Validierungsdatensatz geschätzt werden. Diese basieren auf den Prädikationen $\hat{\mathbf{y}}_{c_r}$:

$$\mathbf{y}_{c_r} = \begin{cases} 0 & \text{wenn } c_r < \epsilon \text{ or } \hat{\mathbf{y}}_{c_r} < \alpha \\ 1 & \text{wenn } \hat{\mathbf{y}}_{c_r} > 1 - \alpha \text{ und } c_r > \sigma \\ 1 & \text{wenn } c_r > l_r - \lambda \\ \text{nicht genutzt} & \text{sonst} \end{cases}, \quad (3.24)$$

wobei die Annahmen von der Initialisierung teilweise weiterhin gelten. D.h. die ersten $\epsilon \in \mathbb{N}$ Zyklen werden als unauffällig und die letzten λ Zyklen als auffällig gekennzeichnet. Von der Annahme, dass die ersten 50 % der Schaltzyklen unauffällig sind, wird abgewichen, da z.B. bei Relais 294 zu beobachten ist, dass auch schon in dieser Betriebsdauer auffällige Schaltzyklen auftreten können. Alle anderen Schaltzyklen werden auf der Basis von $\hat{y}(c_r)$ gelabelt. Durch diese Kennzeichnung ist es z.B. möglich, die auffälligen Zyklen in Abb. 3.43 bei 40.000 entsprechend zu kennzeichnen und gleichzeitig zuzulassen, dass anschließend wieder eine unauffällige Phase durchlaufen wird.

Mit dem Schwellwert $\alpha \in]0, 0.5[$ wird bestimmt, welche Schaltzyklen mit Pseudo-Labels versehen werden. Die Einschränkung, dass nur $c_r > \sigma$ als auffällig gekennzeichnet werden können, ist der Tatsache geschuldet, dass sonst viele Zyklen in einer frühen Lebensphase ebenfalls als auffällig gekennzeichnet würden und somit wenig Lebensdauer eines Relais genutzt würde. Gleichzeitig werden auf diese Weise Zyklen, die zunächst fälschlicherweise als unauffällig gekennzeichnet wurden, aus dem Datensatz entfernt, sodass beide Kennzeichnungen besser unterschieden werden können. Eine grafenbasierte Korrektur der Label, wie in [56] und [57] vorgeschlagen, konnte im Kontext dieser Arbeit nicht umgesetzt werden, da mit den Daten keine Abgrenzung von Nachbarschaften erzielt werden konnte.

Nach dem Pseudo-Labeling beginnt eine neue Iteration der Trainingsprozedur (vgl. Abb. 3.39). Dabei wird ein neues ANN trainiert, d.h. die gelernten Parameter auf der vorherigen Iteration werden nicht genutzt. Dieses Vorgehen ist notwendig, da das ANN andernfalls anfällig ist für eine Überanpassung.

3.6.2 Evaluation

Im Folgenden werden die Ergebnisse von MAUD sowohl in qualitativer als auch in quantitativer Hinsicht diskutiert. Für die Evaluation wurden die in Tab. 3.9 aufgeführten

Parameterwerte genutzt. Dabei ist zu betonen, dass 360 Zyklen einer Stunde Betrieb entsprechen, wo die Wahl dieses Wertes herrührt.

Tab. 3.9: Genutzte Parameterwerte [89]

Parameter	Beschreibung	Wert
n	Anzahl der Merkmale	12
k	Fensterlänge für den gleitenden Mittelwert	120
m	Fensterlänge für den durchschnittlichen Gradienten	120
s	Zeitraum für die Initialisierung	720
λ	Angenommene auffällige Zyklen	360
ϵ	Angenommene unauffällige Zyklen	720
σ	Angenommene normale Operationsphase	$l_r/2$
α	Schwellwert für Pseudo-Label	0.05

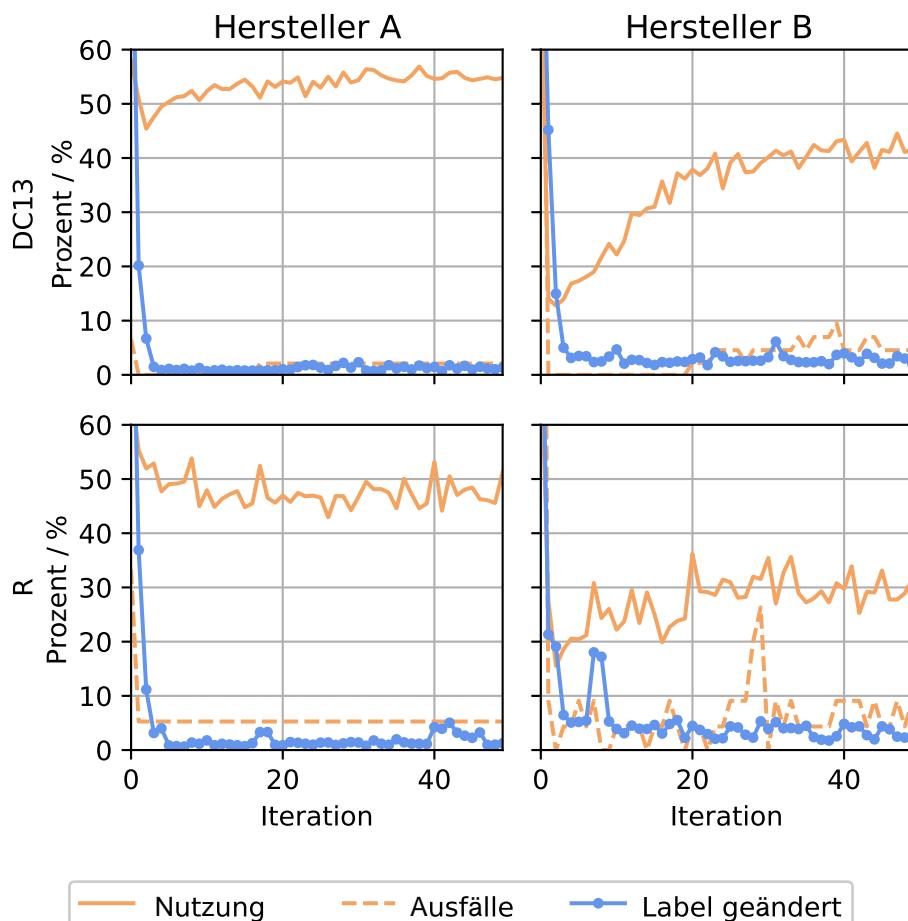


Abb. 3.40: Historie der Nutzung, Ausfälle und geänderten Label über die Iterationen [89]

Abb. 3.40 zeigt die Ergebnisse des ANN auf dem Validierungsset über die ersten 50 Iterationen von MAUD. Die Relais sind nach Hersteller und Last aufgeteilt. Am Ende jeder Trainingsphase werden drei Größen aufgezeichnet: „Label geändert“ quantifiziert den Prozentsatz der Schaltzyklen, die durch die Vorhersagen des neu trainierten ANN für die nächste Iteration geändert wurden. Die „Nutzung“ gibt den durchschnittlichen Prozentsatz der genutzten Lebensdauer an, wobei angenommen wird, dass ein Relais gewechselt wird, wenn es als auffällig eingestuft wird. Um realen Bedingungen Rechnung zu tragen, wurde festgelegt, dass ein Relais mindestens 1 h (360 Zyklen) vor dem Ausfall als auffällig gekennzeichnet werden muss. Andernfalls wird ein Relais als „Ausfall“ gewertet, die ebenfalls relativ in Abb. 3.40 angegeben werden. Um Ausreißer zu unterdrücken, wurden die ANN-Prädiktionen mit einem gleitenden Durchschnitt mit einer Fensterbreite von 120 geglättet.

Bei Hersteller A ist zu erkennen, dass nach fünf Iterationen die drei aufgezeichneten Größen konvergieren. Die geringfügigen Änderungen in den weiteren Iterationen können mit dem Training des ANN begründet werden, bei dem z.B. das Ausbalancieren der Trainingsdaten zufallsbestimmt ist. Durch MAUD wird eine stabile Unterscheidung von auffälligen und unauffälligen Zyklen erlernt, bei der etwa 50 % der nutzbaren Schaltzyklen genutzt werden und weniger als 5 % der Relais ausfallen.

Für Hersteller B ist ebenfalls eine Konvergenz der Größen zu erkennen; allerdings benötigt diese mehr Iterationen. Da prozentual mehr Label zwischen den Iterationen geändert werden, scheint MAUD bei diesem Hersteller nicht so stabil zu sein. Dies wird durch die niedrigere Nutzung von 30 bis 40 % und die höheren Ausfallraten unterstützt. Zudem ist für die Last R eine hohe Streuung der Ausfälle zu beobachten. Dieses Verhalten lässt sich auf zweierlei Weisen begründen: Zum einen haben die Relais des Herstellers A bei DC13-Lasten fast die doppelte Lebensdauer und bei R-Lasten fast die vierfache Lebensdauer. Die Datenmenge ist demnach größer, was sich positiv auf das Training auswirken kann. Gleichzeitig zeugen diese größeren Lebensdauern von einer höheren Produktqualität. Zum anderen zeigen die Histogramme in Abb. 3.25, dass sich die Relais von Hersteller A als vorhersehbarer erweisen (geringere Streuung im Histogramm).

Nach 50 Iterationen von MAUD stehen für jede Hersteller- und Last-Kombination jeweils 50 trainierte ANN zur Verfügung, deren Performanz auf dem Validierungsset bekannt ist. Für die Prädiktion des Testsets wird jeweils das ANN gewählt, das auf dem Validierungsset bei geringster Ausfallrate die höchste Nutzung ermöglicht – dabei sind die Ausfälle vorrangig.

3.6.2.1 Individuelle Ergebnisse

Im Folgenden werden die Vorhersagen in Kombination mit den Daten visualisiert. Zu diesem Zweck wurden zwei Relais ausgewählt, an denen die Funktionsweise von MAUD gezeigt werden kann. Es handelt sich dabei um die zuvor in Abs. 3.4.1 diskutierten Relais 84, 294 und 299. Die Farbintensität in den Grafiken steht in Relation zu den Messreihen, damit ein Zusammenhang zwischen Prädiktion und Daten hergestellt werden kann. Die

Farben geben in den Grafiken die Prädiktionen des ANN an, wobei mit rot ein auffälliger Schaltzyklus angezeigt wird.

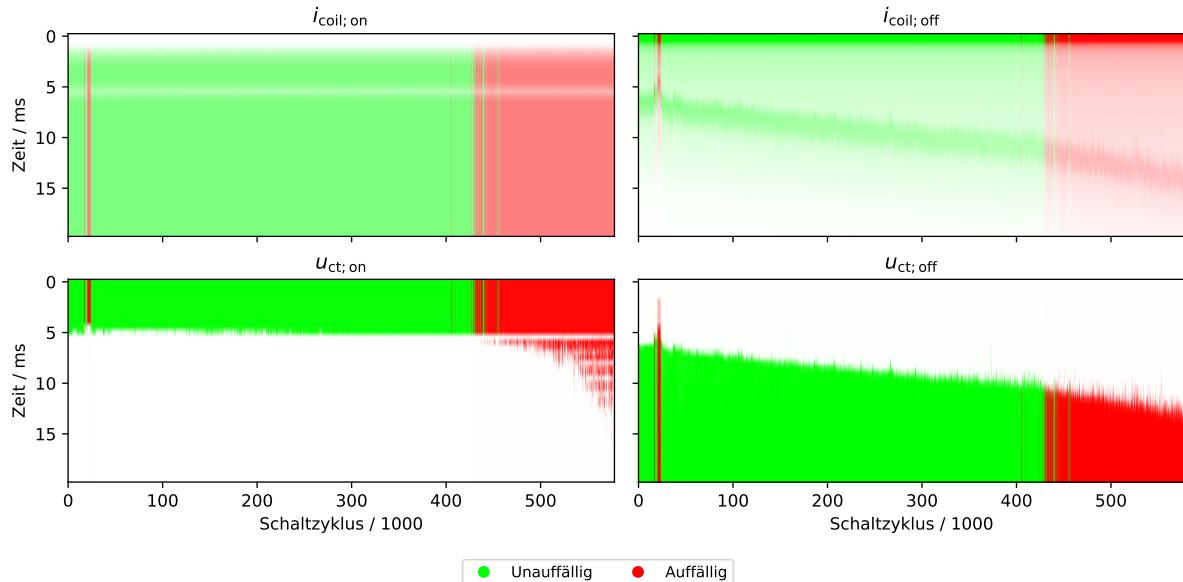


Abb. 3.41: Visualisierung der Prädiktionen für das Relais 294 [89]

Im Falle des kontinuierlich gealterten Relais in Abb. 3.41 ist zu erkennen, dass es am Ende seiner Betriebsdauer als auffällig gekennzeichnet wird. Für den Bereich zwischen 400.000 und 450.000 Zyklen sind sowohl auffällige als auch unauffällige Zyklen gelabelt. Dieses Verhalten ist zu erwarten, da die Degradierung kontinuierlich ist und deshalb eine Übergangsphase zwischen unauffällig und auffällig besteht. Direkt zu Beginn der Betriebsdauer, d.h. vor 50.000 Zyklen, wird das Relais erstmalig als auffällig markiert. Eine genauere Betrachtung der Daten zeigt starke Veränderungen der Messgrößen in diesem Bereich, sodass die Vergabe des Labels nachvollziehbar ist. Das Relais fiel jedoch nicht aus. Dieses Verhalten ist häufig zu beobachten. Die Relais haben auffällige Zyklen, aber nur ein Bruchteil fällt tatsächlich aus. Zurückzuführen ist dies auf reversible Veränderungen: Ablagerungen können den Ausfall eines Relais bedingen oder sich lösen.

In Abb. 3.42 ist ein Relais abgebildet, bei dem ab etwa der Hälfte der Betriebsdauer Veränderungen auftreten, die bis zum Ausfall des Relais immer stärker werden. Dies wird durch die Label widergespiegelt: Die zunächst leichten Veränderungen sind unauffällig, mit steigender Veränderung werden Zyklen auch als auffällig gekennzeichnet, bis zu den letzten 15.000 Schaltzyklen, die alle auffällig sind.

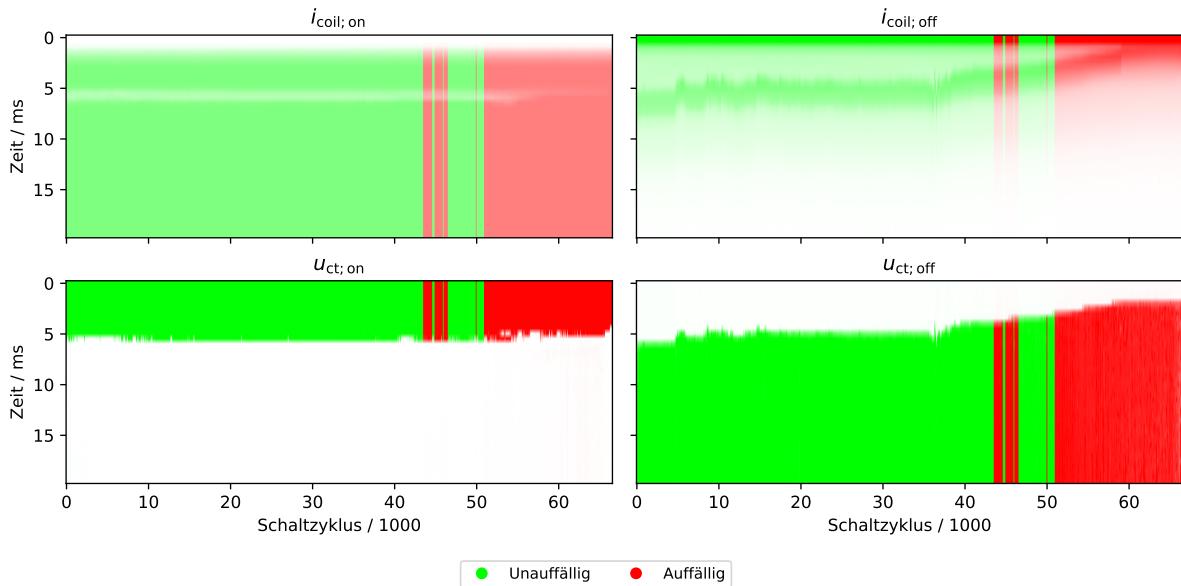


Abb. 3.42: Visualisierung der Prädiktionen für das Relais 84

Das Relais in Abb. 3.43 weist bei 40.000 Zyklen reversible Veränderungen in den Messgrößen auf. An diesem Beispiel kann verdeutlicht werden, dass ein Pseudo-Labeling notwendig ist: Die Degradierung eines derartigen Relais kann nicht über eine monotone Funktion approximiert werden. Vielmehr müssen die Zielwerte für die Schaltzyklen unter Berücksichtigung der jeweiligen Messwerte zugeordnet werden. Das Pseudo-Labeling von MAUD ermöglicht dies, sodass bei 40.000 Schaltzyklen das Relais als auffällig markiert wird.

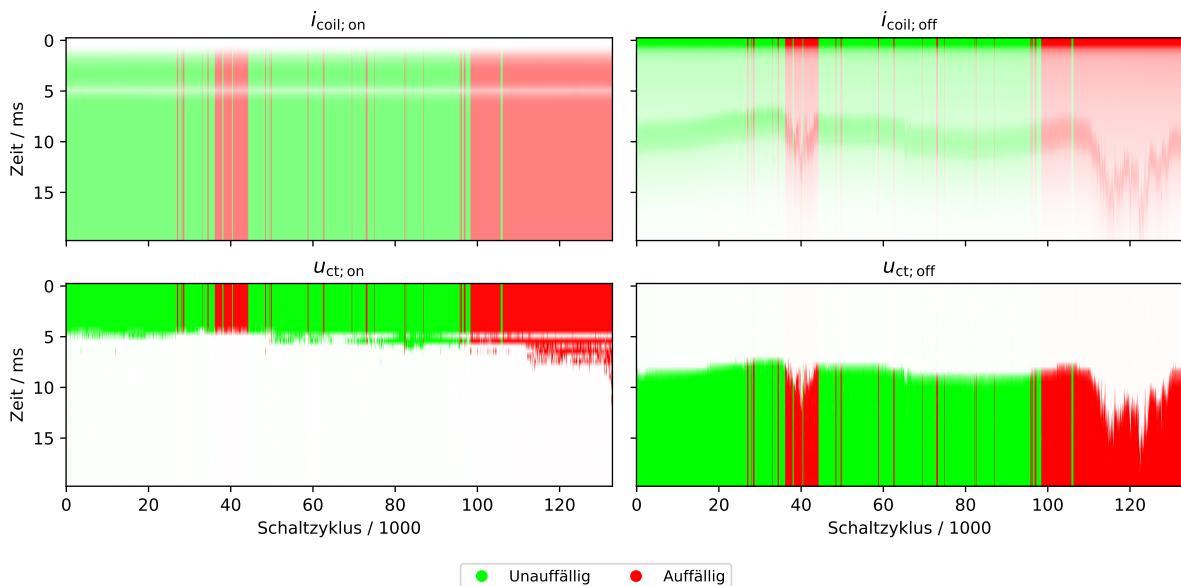


Abb. 3.43: Visualisierung der Prädiktionen für das Relais 299

Zusammenfassend wurde für alle Relais aus dieser qualitativen Stichprobe eine nachvollziehbare Klassifizierung der Schaltzyklen erlernt. Das semi-überwachte Training mit MAUD ist deshalb erfolgreich. Die Beurteilung der Performanz des Verfahrens stellt jedoch eine Herausforderung dar, weil keine Grundwahrheitswerte bekannt sind. Deshalb werden im Kontext dieser Arbeit die Interessen der/die Anwender:innen zur Beurteilung der Performanz herangezogen, diese wiederum erstrecken sich über zwei Dimensionen: Zum einen sollen möglichst viele Ausfälle verhindert werden, zum anderen soll die Lebensdauer der Relais möglichst ausgenutzt werden. Die Präferenzen hinsichtlich beider Dimensionen sind individuell. Dabei muss ein Kompromiss gebildet werden: Weniger Ausfälle erfordern ein vorsichtigeres, früheres Tauschen der Relais, was gleichzeitig zu einer geringeren Nutzung der Lebensdauer führt.

3.6.2.2 Vergleich zum Stand der Technik

Im vorangegangenen Abschnitt wurde anhand von zwei Beispielen qualitativ nachgewiesen, dass die Modellierung erfolgreich war und die Relais vor dem Ausfall gewartet werden können. Im Folgenden wird ein Vergleich mit dem Stand der Technik, dem B10 Wert und MAUD vorgenommen. Dazu werden Ausnutzung der Schaltzyklen und Ausfälle für den Validierungs- und Testsatz mit den Ergebnissen des B10-Wertes verglichen. Der B10 Wert wurde auf Basis des Trainingssets berechnet. Eine Aufschlüsselung nach Hersteller und Belastung wurde nicht vorgenommen, da diese Informationen in der Praxis derzeit nicht verfügbar sind. Außerdem ändern sich die Werte bei deren Berücksichtigung nur minimal, da Frühausfälle unter allen Betriebsbedingungen auftreten.

Tab. 3.10: Ergebnisse [89]

Methode	Nutzung (%)	Ausfälle (%)	Ausfälle (absolut)
Validation			
MAUD (A - R)	53.81	5.26	1
MAUD (B - R)	35.43	0.00	0
MAUD (A - DC13)	54.46	0.00	0
MAUD (B - DC13)	37.19	0.00	0
MAUD (kombiniert)	45.31	0.72	1
Test			
MAUD (A - R)	51.34	0.00	0
MAUD (B - R)	33.04	0.00	0
MAUD (A - DC13)	56.88	6.52	3
MAUD (B - DC13)	37.02	2.22	1
MAUD (kombiniert)	45.15	2.80	4
B10	28.08	9.22	13
MAUD vs. B10	+17.07	-6.42	-9

Die Nutzung der Lebensdauer der Relais ist in allen Fällen weit entfernt von 100 %. Dies ist mit den reversiblen Veränderungen, wie sie bspw. in Abb. 3.43 zu beobachten sind, zu begründen. Die Relais durchlaufen während ihrer Betriebsdauer Phasen, zu denen ein Ausfall möglich ist. Um die Anzahl der Ausfälle gering zu halten, muss bei ersten Anzeichen für einen Ausfall eine Instandhaltung durchgeführt werden. Dies wirkt sich negativ auf die Nutzung aus.

Beim Vergleich von Validierungs- und Testset sind bei allen Hersteller- und Lastkombinationen nur wenige Prozent Abweichung bei der Auslastung zu erkennen. Dies deutet darauf hin, dass MAUD eine zuverlässige Klassifikation erlernt hat. Im Testset treten insgesamt drei Ausfälle mehr auf. Diese Relais sind mit Lebensdauern von 5.000 bis 18.000 Zyklen extreme Frühhausfälle, die in den Daten unterrepräsentiert sind.

Der Vergleich mit dem B10-Wert ist eindeutig: Durch MAUD kann die Auslastung um 17,07 Prozentpunkte erhöht und die Ausfälle um 6,42 Prozentpunkte reduziert werden. Das Ziel, eine Methode zu entwickeln, mit der Relais sicherer gewartet und gleichzeitig besser ausgelastet werden können, wurde schließlich voll erreicht.

3.7 Inferenz auf dem Mikrocontroller

Abgeschlossen werden die Untersuchungen dieses Kapitels mit einer Betrachtung einer möglichen Applikation eines ANN mit Hardware, die auch im Kontext eines Produktes eingesetzt werden könnte. Dabei ist die Interferenz von ANNs auf Mikrocontrollern als kritischer Pfad in der Produktentwicklung auszumachen und wird deshalb nachfolgend untersucht. Die Ergebnisse wurden bereits im Rahmen einer Bachelorarbeit veröffentlicht, deren Betreuung im Kontext dieses Promotionsvorhabens stattfand [97].

3.7.1 Versuchsbeschreibung

Zur Betrachtung der Applikation von ANN auf Mikrocontrollern werden drei Komponenten benötigt:

1. ANN-Modelle: Die ANN sind notwendig, um diese zu komprimieren und deren Performanz zu prüfen.
2. Software: Software wird benötigt, um eine Kompression der ANN durchzuführen.
3. Hardware: Zur Ausführung auf einem Mikrocontroller ist diese notwendig.

Die Komponenten werden in den folgenden Abschnitten detailliert erörtert.

3.7.1.1 ANN-Modelle

Diese Untersuchung verfolgt das Ziel, allgemein Informationen zu sammeln, welche Topologien bzw. Komplexität auf einem Mikrocontroller ausgeführt werden kann und welche Einflüsse die Kompressionsverfahren auf die Performanz haben. Deshalb werden Modelle

unterschiedlicher Komplexität gewählt, deren Performanz anhand einer Metrik eindeutig beurteilt werden kann. Dadurch fällt das eigene vorgestellte Verfahren MAUD aus dieser Untersuchung, da dies anhand von zwei Dimensionen zu beurteilen ist: der Nutzungsdauer und den unerkannten Ausfällen. Die selbst ernannten Label können nicht herangezogen werden, da keine Grundwahrheit bekannt ist.

Deshalb wird für diesen Versuch nur ein Teildatensatz des gesamten Datensatzes genutzt. Dabei wird nach der Art der Degradierung ausgewählt: Relais, die kontinuierliche Veränderungen in Form eines Abbrands des Kontaktmaterials zeigen, werden ausgewählt. Insgesamt umfasst dieser Datensatz 253 Relais, von denen jeweils 40 dem Validierungs- und Testset zugeteilt wurden. Aufgrund der ausgewählten Relais ist eine Schätzung der relativen Restlebensdauer möglich, wie in Abs. 3.5 bereits erörtert. Als Maß der Performanz wird der MSE herangezogen.

Es werden zwei ANN genutzt: ein simples „Multilayer Perceptron“ (dt. mehrschichtiges Perzepron) (MLP) und der Encoder eines „Autoencoder“ (AE) in Kombination mit einem MLP. Auf diese Weise werden Erkenntnisse zu unterschiedlichen Komplexitätsstufen gesammelt und zwei konkurrierende Verfahren zur Merkmalsextraktion gegenübergestellt: klassisch durch manuelle Algorithmen und unüberwacht durch den AE.

Tab. 3.11: Topologie der Kombination aus Encoder mit MLP

Schicht	Typ	Neuronen / Filter
1	Conv1D	16
2	MaxPooling1D	-
3	Conv1D	8
4	MaxPooling1D	-
5	Conv1D	8
6	MaxPooling1D	-
7	Conv1D	8
8	MaxPooling1D	-
9	Conv1D	8
10	MaxPooling1D	-
11	Conv1D	4
12	MaxPooling1D	-
13	Flatten	-
14	Dense	128
15	Dropout	-
16	Dense	32
17	Dropout	-
18	Dense	8
19	Dense	1

Die Topologie der Kombination aus Encoder eines AE und MLP ist in Tab. 3.11 aufgeführt, alle Conv1D und Dense Layer wurden mit einer ReLU Aktivierungsfunktion ausgestattet.

Das MaxPooling1D hat eine Fensterbreite von zwei, und die einzelnen Kanäle wurden separat trainiert. Im Rahmen von Versuchen hat sich diese Separierung als vorteilhaft erwiesen – ein Grund hierfür könnte die zeitliche Unabhängigkeit der Messgrößen beim Ein- und Ausschalten sein. Das Modell setzt sich aus dem Encoder des AE und einem normalen MLP zur Regression zusammen. Durch diese Nutzung eines Autoencoders wird das unüberwachte Extrahieren von Merkmalen ermöglicht, die durch das MLP zur Regression der Restlebensdauer genutzt werden. Das Training untergliedert sich daher in zwei Stufen: Zunächst wird der AE trainiert, der eine nieder-dimensionale Repräsentation der Eingangswerte erlernt (vgl. Abs. 3.2.2). Nach dem Abschluss des Trainings werden Encoder und Decoder voneinander getrennt und die trainierbaren Parameter eingefroren. Dadurch kann im zweiten Schritt ein MLP an den Encoder angehängt werden, mit dem die Restlebensdauer geschätzt wird. Die Prädiktionen des AE werden über ein Fenster gemittelt, um eine Vergleichbarkeit der beiden Modelle zu erreichen. Dies ist notwendig, da die Merkmale ebenfalls geglättet wurden.

Tab. 3.12: Topologie des MLP

Schicht	Typ	Neuronen	Informationen
1	Dense	16	
2	Dense	32	Aktivierungsfunktion: ReLU
3	Dropout	32	Rate: 10 %
4	Dense	1	Aktivierungsfunktion: ReLU

Das MLP hat eine simple Topologie, die in Tab. 3.13 angegeben wird. Es wird mit den in Abs. 3.4.3 erörterten Merkmalen trainiert. Bei allen Modellen werden die gleichen Einstellungen verwendet: Über das Validierungsset wird der Trainingsfortschritt fortwährend überwacht und bei Verschlechterung abgebrochen. Als Metrik dient der MSE.

3.7.1.2 Komprimierung

Zur Komprimierung wird auf Softwarelösungen von zwei Herstellern zurückgegriffen:

- STM32Cube.AI

STM32Cube.AI ist eine Bibliothek für das Ausführen von ANN auf STM32 Mikrocontrollern und in die Entwicklungsumgebung von STM integriert. Die Bibliothek unterstützt die Konvertierung von ANN im ONNX-Format in C-Code, wobei dieser für den ausgewählten Mikrocontroller optimiert wird. Zusätzlich wird eine Laufzeitumgebung für den C-Code generiert, sodass ANN mit wenig Aufwand auf einem Mikrocontroller ausgeführt werden können. Ein Nachteil dabei ist, dass nicht alle Funktionen von ANN-Frameworks wie TensorFlow unterstützt werden. Neben der Generierung des C-Codes wird ein Komplexitätsbericht erstellt, der den Speicherbedarf (ROM und RAM) sowie ein Maß für die Komplexität des Modells (MACC) umfasst.

Als Komprimierungsmethoden bietet die Bibliothek eine Variante aus Pruning und Rechenoptimierung an, die vom Hersteller nicht genau beschrieben wird. Das Verfahren kann über den Kompressionsfaktor beeinflusst werden, wobei zwei Werte auswählbar sind: vierfache und achtfache Kompression.

- **TensorFlow Lite**

TensorFlow Lite ist eine spezielle Version von TensorFlow, die speziell auf die Applikation von ANN auf Mikrocontrollern ausgerichtet ist. Daher sind Optimierungen und Komprimierungen für neuronale Netze inkludiert. Mit der TFLite-Bibliothek wird im Rahmen dieser Arbeit eine Ganzzahl-Quantisierung der ANN und Daten durchgeführt. Durch TensorFlow Lite wird ein quantisiertes ANN erstellt und die Parameter zur Umrechnung der Daten bereitgestellt. Die Generierung von C-Code ist mit dieser Bibliothek ebenfalls möglich.

Mit diesen Softwarelösungen ergeben sich drei verschiedene C-Codevarianten, die im Weiteren mit folgenden Indizes abgekürzt werden:

1. C: C-Code ohne Komprimierung.
2. C4: C-Code mit vierfacher Komprimierung (STM32Cube.AI).
3. C8: C-Code mit achtfacher Komprimierung (STM32Cube.AI).
4. CQ: C-Code mit Ganzzahl Quantisierung (TensorFlow Lite).

3.7.1.3 Hardware

Zur Inferenz müssen die generierten C-Codes auf einem Mikrocontroller ausgeführt und die Daten müssen zwischen Computer und Mikrocontroller kommuniziert werden. Aufgrund des umfangreichen Softwareangebots von STM32 wird dazu ein STM32-Mikrocontroller genutzt. Diese Produktfamilie basiert auf der 32-Bit Arm Cortex M Prozessorreihe und umfasst ein breites Spektrum an Controllern, das diverse Anwendungsfelder bedient [98]. Im Kontext dieser Arbeit wird der hochperformante STM32F767ZI genutzt, der mit einer ARM Cortex M7 32-Bit RISC CPU, einer Gleitkommaeinheit, 2 MB ROM und 512 kB RAM-Speicher ausgestattet ist. Der maximale Takt liegt bei 216 MHz [99].

Das Generieren des C-Codes und die Einbindung in die Firmware des Controllers wird über die STM32Cube.AI Bibliothek umgesetzt. Zusätzlich wurde zur Kommunikation eine UART Schnittstelle hinzugefügt, um die Daten zum Controller und die Prädiktionen zurückzuschicken. Auf dem Computer kann so die Performanz des jeweiligen ANN anhand des MSE gemessen werden. Dazu wird für jedes ANN der gesamte Trainingsdatensatz ausgewertet.

3.7.2 Evaluierung

Durch die Evaluation sollen zwei Aspekte geprüft werden: Zum einen der Einfluss der Mikrocontroller Inferenz auf die Performanz und den Speicherbedarf der ANN und zum

anderen die Ausführungskomplexität bzw. -dauer. Beide Aspekte werden nachfolgend in separaten Abschnitten getestet. Vorab muss jedoch betont werden, dass das MLP eine geringfügig bessere Performanz aufweist als der AE. Diese Erkenntnis gilt bereits für die unkomprimierten Modelle und stellt auch den Grund dar, dass beim Verfahren MAUD Merkmale genutzt wurden. Begründet werden kann dies mit dem Expertenwissen, das in die Augmentation der Merkmale eingeflossen ist.

3.7.2.1 Einfluss der Mikrocontroller Inferenz

Der Einfluss der Mikrocontroller Inferenz auf die Performanz wird für beide ANN unabhängig betrachtet. Zunächst wird das MLP herangezogen. Insgesamt sind in Tab. 3.13 fünf verschiedene Versionen angegeben und die entsprechenden MSE, ROM und RAM Werte.

Tab. 3.13: Performanz und Speicherbedarf von verschiedenen Versionen des MLP

Modell	MSE	ROM / B	RAM / B
MLP	0,0105	-	-
MLP _C	0,0105	3.400	260
MLP _{C4}	0,0105	2.888	260
MLP _{C8}	0,0270	836	260
MLP _{CQ}	0,0112	996	65

Das ursprüngliche MLP wird mit einem Computer trainiert und ausgeführt. Deshalb entfallen hier die Angaben von ROM und RAM; der MSE dient als Referenzwert. Beim MLP_C Modell sind hinsichtlich des MSE bis zur vierten Nachkommastelle keine Unterschiede auszumachen. Da durch die Komprimierung die Anzahl der Gewichte reduziert wird und diese im ROM gespeichert werden, benötigen die Modelle MLP_{C4} und MLP_{C8} weniger ROM. Dabei ist der MSE des MLP_{C4} identisch mit dem des originalen MLP – durch die Komprimierung kann der ROM-Bedarf ohne Performanz Verringerung um 15 % reduziert werden. Das MLP_{C8} führt zu einer 75 % Reduktion des ROM-Speicherbedarfs gegenüber dem MLP_C. Dabei wird der MSE jedoch signifikant verschlechtert. Schlussendlich sind die Ergebnisse des MLP_{CQ} aufgeführt: Der Speicherbedarf wird um etwa 71 % verringert und der RAM-Bedarf sogar um 75 %. Dies ist darauf zurückzuführen, dass die 8-Bit-Ganzzahlen, mit denen bei diesem Modell gerechnet wird, ein Viertel des Speichers von 32-Bit-Gleitkommazahlen belegen. Der MSE nimmt durch die Quantisierung nur minimal zu.

Für das MLP lässt sich damit Folgendes festhalten: Durch eine vierfache Komprimierung kann ROM-Speicherplatz eingespart werden, ohne Performanz Verluste hinnehmen zu müssen. Mit der Quantisierung kann der ROM- und RAM-Speicherbedarf um einen erheblichen Anteil reduziert werden. Dies geht allerdings mit geringen Performanz Verlusten einher.

Tab. 3.14: Performanz und Speicherbedarf von verschiedenen Versionen des AE

Modell	MSE	ROM / kB	RAM / kB
AE	0,0165	-	-
AE _C	0,0165	40,04	17,00
AE _{C4}	0,0165	20,58	17,00
AE _{C8}	0,0260	14,24	17,00
AE _{CQ}	0,0166	10,82	5,44

Bei Betrachtung der absoluten Werte des AE in Tab. 3.14 wird die höhere Komplexität deutlich. Der Speicherbedarf liegt bei mehr als dem Zehnfachen des MLP. Die Evaluation der AE-Modelle lässt die gleichen Rückschlüsse zu:

- Der AE kann erfolgreich in C-Code überführt werden.
- Durch die vierfache Kompression wird ROM-Speicherplatz eingespart, ohne negativen Einfluss auf den MSE.
- Bei der achtfachen Kompression wird gegenüber der vierfachen auf Kosten des MSE mehr ROM eingespart.
- Die Quantisierung führt zu einer geringfügigen Verschlechterung des MSE, aber gleichzeitig auch zu erheblichen ROM- und RAM-Einsparungen.

Zusammenfassend ist für die Kompressionsmethoden bei beiden ANN festzuhalten, dass durch die vierfache Kompression kein nachweislicher MSE-Verlust bei deutlicher ROM Speicherplatz Reduzierung erreicht wird. Durch die Quantisierung steigert sich der MSE leicht; der Speicherbedarf an RAM und ROM kann jedoch stärker reduziert werden. Die achtfache Komprimierung führt zu einer wesentlichen Verschlechterung des MSE, weshalb dieses nicht empfehlenswert ist.

3.7.2.2 Ausführungskomplexität

Zur Ausführungskomplexität wird untersucht, wie viel Rechenzeit für eine Inferenz benötigt wird. Dazu werden zwei Kennzahlen herangezogen. Der MACC-Wert gibt an, wie viele Rechenoperationen notwendig sind und ist daher unabhängig vom ausführenden System. Für die Anwendung ist jedoch relevant, wie lange die Inferenz dauert. Deshalb wird diese auf Basis des MACC berechnet.

In Tab. 3.15 werden die Kennzahlen angegeben. Die Inferenzdauern von MLP_{C4} und MLP_{CQ} sind mit 0,025 ms nicht zu unterscheiden und sind in Vergleich zu einem Schaltzyklus, der mindestens das Tausendfache an Zeit beansprucht, zu vernachlässigen. Aufgrund der höheren Komplexität weisen die Modelle AE_{C4} und AE_{CQ} höhere Dauern von 4.723 ms bzw. 4.509 ms auf. Diese Werte liegen ebenfalls unter der Schaltzeit von Relais und sind deshalb für die Anwendung geeignet.

Tab. 3.15: Vergleich von MACC und Inferenzdauer der besten Versionen

Modell	MACC	Inferenzdauer / ms
MLP _{C4}	882	0,025
MLP _{CQ}	883	0,025
AE _{C4}	170.018	4,723
AE _{CQ}	162.315	4,509

Beim Vergleich der MACC ist beim MLP kein signifikanter Unterschied auszumachen. Die Recheneinsparungen durch die Komprimierung entsprechen in etwa der der Quantisierung. Beim AE hingegen konnte durch die Quantisierung ein geringerer MACC erzielt werden.

In Bezug auf die Ausführungskomplexität kann durch diese Untersuchung festgestellt werden, dass mit einer Quantisierung sowohl MACC als auch Speicherbedarf signifikant reduziert werden können. Dabei wird der MSE nur geringfügig verschlechtert. Daher ist die Quantisierung der ANN der Komprimierung vorzuziehen. Weiterhin bringt die Quantisierung einen vorteilhaften Aspekt: Durch die Verwendung von Ganzzahlen ist keine Gleitkommaeinheit im Mikrocontroller notwendig, wodurch ANN sogar auf den günstigen Mikrocontrollern ausführbar werden.

3.8 Fazit

Abschließend werden die in diesem Kapitel erarbeiteten Ergebnisse zusammengefasst. Dabei werden drei Forschungsthemen unterschieden:

3.8.1 Datensatz

Das erste Forschungsthema ist die Datenbasis, also der verwendete Versuchsaufbau und der damit generierte Datensatz. Hier wurde eine Übersicht über die öffentlich verfügbaren Datensätze im Kontext von PdM erstellt. Daran konnte gezeigt werden, dass eine Forschungs- bzw. Datensatzlücke hinsichtlich des Umfangs, des Datentyps und der Komplexität der Ausfälle besteht. Daher wurde im Kontext dieser Arbeit ein vollautomatischer Testaufbau entwickelt, mit dem hunderte Relais über Jahre unter realistischen, wechselnden Umgebungsparametern degradiert wurden. Auf diese Weise ist ein neuer Datensatz entstanden, der in Hinblick auf den Umfang und die Komplexität der Degradierung hervorsteht. Der Datensatz stellt eine wichtige Grundlage für weitere Forschungsaktivitäten dar, weshalb er veröffentlicht [28] und ausführlich beschrieben [6] wurde. Verbesserungspotential besteht hinsichtlich der Variation von Umgebungsbedingungen: In praktischen Anwendungen ist mit Temperaturschwankungen zu rechnen, die nicht abgebildet wurden. Weiterhin ist ein größeres Spektrum an Lasten in der Praxis zu erwarten. Für zukünftige Datensätze empfehlen sich Kooperationen mit Unternehmen, im Rahmen derer die Datenerfassung in

Produkte integriert wird. So kann der engen Verzahnung von Wissenschaft und Anwendung bei PdM begegnet werden.

3.8.2 Maschinelles Lernverfahren

Im Kontext dieser Arbeit wurde zunächst gezeigt, dass die klassischen beiden Ansätze auf den Datensatz nicht erfolgreich angewandt werden können. Dies liegt vorwiegend an den komplexen Degradierungsprozessen: Eine monotone Approximation der Restlebensdauer ist nur für einen begrenzten Kreis von Relais möglich. Ein Großteil der Relais weist jedoch spontane Ausfälle oder reversible Änderungen auf, weshalb eine monotone Approximation nicht geeignet ist. Anomalien können nicht unterschieden werden, da die Relais oftmals viele Hundert Schaltvorgänge anormales Verhalten aufweisen, bis diese schließlich ausfallen. Die Anomalien sind daher im Datensatz repräsentiert. Ein weiterer Grund ist das unterschiedliche Verhalten der Relais, das u.a. auf Fertigungsabweichungen zurückzuführen ist. Angesichts dessen werden teilweise die gesamten Lebensdaten von Relais als Anomalie bezeichnet. Die Erkenntnisse bezüglich der Unzulänglichkeiten beider klassischer Ansätze wurden ebenfalls in [6] und [89] publiziert.

Daher wurde eine neuer Ausfallindikator eingeführt, mit dem ein bevorstehender Ausfall angezeigt wird. Es handelt sich um eine binäre Klassifikation, bei der ein ANN semi-überwacht trainiert wird. D.h. zu Beginn werden nur wenig Label als bekannt angenommen und durch einen iterativen Pseudo-Labeling Prozess wird der gesamte Datensatz sukzessiv in das Training einbezogen. Auf diesem Weg lernt das ANN auffällige und unauffällige Schaltzyklen zu unterscheiden. Die Funktionalität des Verfahrens konnte qualitativ im Rahmen einer Evaluation gezeigt werden. Überdies konnte quantitativ nachgewiesen werden, dass es dem derzeitigen Stand der Technik überlegen ist: Die Auslastung kann um 17 Prozentpunkte verbessert und unentdeckte Ausfälle um 6 Prozentpunkte reduziert werden.

Die Ergebnisse zeigen, dass die Anwendung des Verfahrens auf andere Komponenten vielversprechend ist. In weiteren Untersuchungen könnte das Verfahren in zwei Aspekten erweitert werden: Statt der binären Klassifikation könnten mehrere Klassen dynamisch im Pseudo-Labeling Prozess vergeben werden, sodass mehr Zustände der Relais erfasst werden können. Statt einer Klassifikation könnte eine Regression durchgeführt werden, wobei die Approximation der Restlebensdauer iterativ korrigiert wird – wie beim Pseudo-Labeling.

Das Verfahren MAUD ist bereits im Rahmen der Publikation [89] veröffentlicht, zudem wurden zwei Patente [96] und [95] angemeldet.

3.8.3 Applikation auf eingebetteten Systemen

Es existieren bereits viele Verfahren zur Effizienzsteigerung von ANN, von denen zwei im Kontext dieser Arbeit evaluiert wurden: die Quantisierung und das Parameterteilen bzw. die Kompression. Dazu wurden zwei unterschiedlich komplexe Topologien verwendet: ein

simples MLP und ein Encoder in Kombination mit einem MLP. Beide Topologien wurden in wenigen Millisekunden auf einem Mikrocontroller ausgeführt. In Bezug auf die genutzten Verfahren zur Effizienzsteigerung lässt sich Folgendes festhalten: Die Quantisierung ist der beste Kompromiss zwischen Performanz und dem benötigten Speicherplatzbedarf. Weiterhin ermöglicht die vollständige Integer-Quantisierung die Nutzung von Mikrocontrollern ohne Gleitkommaeinheit. In weiterführenden Forschungsvorhaben sollte geprüft werden, inwiefern Methoden zur Effizienzsteigerung kombiniert werden können: Es ist bspw. denkbar, das Parameterteilen mit der Quantisierung zu kombinieren und ggf. um eine Beschneidung zu erweitern. Auf diesem Weg könnte der Speicherplatzbedarf und die MACC weiter reduziert werden.

4 Reduzieren des Prellens

Im nachfolgenden Kapitel wird eine Softwarelösung zur Reduzierung des Prellens vorgestellt, die ebenfalls in [100] veröffentlicht ist. Zunächst wird auf die Motivation für dieses Vorhaben eingegangen: Das Prellen ist ein unerwünschtes Phänomen, das bei fast allen Schaltvorgängen von Relais zu beobachten ist. Es lässt sich auf den mechanischen Aufbau der Relais zurückführen: Um die Konduktivität der Kontakte sicherzustellen, müssen diese stets mit einer Kraft aufeinander gedrückt werden. Dies hat zur Folge, dass ein elastischer Stoß zwischen den Kontaktpaaren entsteht. Problematisch hierbei ist, dass die Degradierung beschleunigt wird. Dies wurde beispielsweise in [101] nachgewiesen.

Des Weiteren sprechen folgende Gründe dafür, Anstrengungen zu unternehmen, die das Prellen reduzieren:

- Geräuschentwicklung: Mit dem Prellen werden Geräusche hervorgerufen, deren Unterbindung z.B. in Haushaltsgeräten einen Vorteil darstellt.
- Signalübertragung: Wenn über die Schaltstrecke Signale übertragen werden, dann können diese durch das schnelle und unberechenbare Öffnen und Schließen der Kontakte beim Prellen gestört werden. Zu dieser Problematik ergibt sich eine ganz eigene Forschungs- bzw. Entwicklungsdisziplin: das Entprellen. Eine grundlegende Einführung kann [102] entnommen werden. Durch ein Reduzieren des Prellens können auch in dieser Hinsicht Verbesserungen erzielt werden.
- Schaltvorgangsdauer: Schlussendlich wird ein Schaltvorgang durch das Prellen verlängert. Solange die Konduktivität eines Kontaktpaars nicht konstant ist, ist der Schaltvorgang nicht abgeschlossen. In vielen Anwendungen sind kürzere Schaltdauern vorteilhaft, wenn bspw. beim Schalten von Wechselstrom im Spannungs- oder Stromnullpunkt geschaltet werden soll, dann muss der Schaltvorgang möglichst kurz und präzise sein.

Das folgende Kapitel ist dabei wie folgt untergliedert: Zunächst werden Grundlagen erörtert. Daraufhin wird aus dem Stand der Forschung die Motivation für die Entwicklung eines neuen Lösungsansatzes abgeleitet. Anschließend wird der neue Lösungsansatz beschrieben und eine Evaluation der Performanz durchgeführt. Schlussendlich wird ein Fazit zu den Untersuchungen zur Reduzierung des Prellens gezogen.

4.1 Grundlagen

Die Grundlagen sind in zwei Teilespekte untergliedert, zum einen in den Stand der Forschung und Technik und zum anderen in eine Einführung in die „Particle-Swarm-Optimization“ (dt. Partikel Schwarm Optimierung) (PSO).

4.1.1 Stand der Forschung und Technik

Seit mehr als 50 Jahren werden Studien zur Verringerung des Prellens durchgeführt, um die Zuverlässigkeit und Lebensdauer von Relais zu erhöhen [103]. Vieles davon beschränkt sich auf die mechanische Optimierung der Relaiskonstruktion [104]. Es wurden bereits viele Simulationsmodelle für Relais entwickelt. In [105] wird bspw. sogar das Prellen mit simuliert. Es wurden aber auch einige Forschungsprojekte durchgeführt, die darauf abzielen, das Steuersignal der Relaisspule zu optimieren. In diesem Zusammenhang haben *Davies et al.* eine Methode zur Reduzierung des Kontaktrellens durch Steuerung des Flusses im Anker der Spule [106] entwickelt. *Moraes* und *Perin* [107] stellten eine mit einem Mikrocontroller ausführbare Lösung für eine Stromregelung und Schließererkennung vor, die das Prellen um etwa 30 % reduzieren kann. Das Verfahren basiert auf der Abschätzung der Verschiebung des Ankers bzw. Kerns. Dieses Prinzip wurde im Zuge weiterer Arbeiten erweitert. In [108], [109] und [110] wurde etwa eine Echtzeitschätzung der Verschiebung verwendet, um den Anker im geschlossenen Regelkreis zu steuern. Dies geht mit einer Reduzierung des Prellens einher. Die auf Verschiebungsschätzungen basierenden Ansätze werden jedoch in dieser Arbeit nicht weiter betrachtet, da sie auf vereinfachten Modellen des magnetischen Relais beruhen. Diese führen zu geringer Genauigkeit, Abhängigkeit von Körperparametern und unzureichender Universalität, wie *Tang et al.* bereits festgestellt haben [111]. Daher wurde einen Wegsensor und ein künstliches neuronales Netz genutzt um die Wegschätzung zu verbessern. Durch den erforderlichen, kostspieligen Sensor unterscheidet sich dieser Ansatz grundlegend von dem in dieser Arbeit vorgestellten.

Eine alternative Methode ohne zusätzliche Sensorhardware wird in [112] beschrieben. Dort wird eine zweistufige Steuerung verwendet, um ein Versorgungsspannungssignal zu erzeugen, das durch zwei Variablen definiert ist: Zum Zeitpunkt t wird die Spannung für eine bestimmte Dauer d auf den entgegengesetzten Wert umgeschaltet und dann wieder auf den ursprünglichen Wert. Die beiden Variablen t und d werden mit „Run-to-Run-Algorithm“ (R2R) optimiert. Die Auswertung zeigt, dass eine Reduzierung der Prellzeit um 79 % möglich ist. Im Rahmen dieser Arbeit wird die Grundidee des R2R [112] verbessert und mit einem besseren Optimierungsalgorithmus umgesetzt. Dies geschieht, weil nach der Neuimplementierung des R2R einige Hindernisse in seinem Betrieb aufgetreten sind: Relais zeigen eine starke Varianz im Prellen bei einzelnen Schaltereignissen, sodass die Optimierungen mehrfach wiederholt werden müssen, um eine verlässliche Aussage über ihren Einfluss auf das Prellverhalten zu treffen, was in R2R nicht berücksichtigt wurde. Außerdem ist es nicht in der Lage, mehrere lokale Maxima gleichzeitig zu untersuchen. Eine weitere Einschränkung ist die Freiheit der Suchrichtung, die durch die gewählte Maske begrenzt ist. Um diese Einschränkungen zu überwinden, wurden Verbesserungen des R2R

vorgestellt und in [113] verifiziert. Für die beste Variante mit Bayes'scher Optimierung ist jedoch Wissen über das Problem erforderlich, was die Anpassung erschwert. Außerdem ist diese Methode rechenintensiv, sodass sie nicht zum Vergleich herangezogen wird, da Adaptivität und eine Mikrocontroller-Implementierung zwingend erforderlich sind. Weiterhin wird in [113] eine Variante mit Nelder-Mead-Verfahren vorgestellt. Diese ist ebenso wie die klassische R2R einer Mustersuche zuzuordnen und wurde nur für Ventile implementiert, weshalb im Rahmen dieser Arbeit die klassische R2R als Referenz gewählt wurde.

Der entwickelte „Bouncing-Optimization-Algorithm“ (dt. Prelofftimierungsalgorithmus) (BOA) hat die Fähigkeit, mit Hindernissen umzugehen, da er eine Variante der PSO [114] verwendet. PSO bewährte sich in zahlreichen Arbeiten. In [115] wird ein umfangreicher Literaturüberblick über Anwendungen von PSO vorgestellt, die von der Elektrotechnik (wie dem Maximum-Power-Point-Tracking in Photovoltaikanlagen), über die Elektronik (wie dem Filterdesign) und den Maschinenbau (wie dem Fräsen), hin zur Informationsverarbeitung (wie dem Datenclustern) reichen. Einfachheit, Effektivität und Robustheit sind die herausragenden Eigenschaften einer PSO und machen diese zu einer guten Basis für BOA. Außerdem kann BOA aufgrund des geringen Rechenaufwands auf kostengünstiger eingebetteter Hardware implementiert werden.

4.1.2 Partikel Schwarm Optimierung

Bei einer PSO handelt es sich um ein Verfahren, mit dem das Verhalten eines natürlichen Algorithmus nachgebildet wird, um ein Optimierungsproblem zu lösen. Es ist den evolutionären Algorithmen und damit künstlicher Intelligenz zuzuordnen. Erstmals wurde ein derartiges Verfahren von *Eberhard* und *Kennedy* vorgestellt [114] und seitdem erfolgreich auf diverse Anwendungsfelder übertragen [115]. Für die Anwendung einer PSO sind keine Annahmen bezüglich des Problems notwendig, was im Kontext dieser Arbeit notwendig ist. Zusätzlich kann ein großer Raum an möglichen Lösungen untersucht werden. Nachfolgend wird die grundlegende Funktionsweise einer PSO beschrieben und darauf aufbauend werden zwei Spezialfälle erörtert, die im Rahmen dieser Arbeit kombiniert werden. Zur Visualisierung der Funktionsweise wird dabei beispielhaft die Beale Funktion optimiert [116]:

$$f(x, y) = (1,5 - x + xy)^2 + (2,25 - x + xy^2)^2 + (2,625 - x + xy^3)^2, \quad (4.1)$$

welche eine vielfach genutzte Benchmark-Funktion ist, die mehrere lokale Minima aufweist, ähnlich wie das Optimierungsproblem im Kontext dieser Arbeit.

4.1.2.1 Allgemein

Die folgende Beschreibung einer PSO basiert auf der Veröffentlichung [114]. Die grundlegende Idee einer PSO spiegelt sich schon im Namen wider: Ein Schwarm von Partikeln $\Psi \in \mathbb{R}^n$ wird verwendet, um nach einer Lösung für ein Optimierungsproblem zu suchen. Dabei steht $n \in \mathbb{N}$ für die Dimension des Lösungsraumes. Die Suchstrategie hierbei ist in

zwei Schritte zu untergliedern: Jeder Partikel definiert eine mögliche Lösung des Problems, deren Performanz im ersten Schritt ermittelt wird. Mit Hilfe dieser kann daraufhin im zweiten Schritt ein Update der Partikel durchgeführt werden, d.h. die Lösungsdefinition eines jeden Partikels wird verändert. Deshalb muss an dieser Stelle wieder mit Schritt eins begonnen werden; gesucht wird letztlich iterativ.

Das Update der Partikel ist der wesentliche Kern, indem sich die unterschiedlichen PSO unterscheiden. Ursprünglich wurde vorgesehen, dass die Partikel sich mit einer Geschwindigkeit $\mathbf{v} \in \mathbb{R}^n$ im Suchraum bewegen und die Partikel gemäß dieser ihre Position im Suchraum ändern. Die Performanz wird in diesen Prozess einbezogen, indem auch \mathbf{v} in jeder Iteration aktualisiert wird. Folgende Formel ist dafür vorgesehen:

$$\mathbf{v}^{\text{new}} = \tau_1 \mathbf{v} + \tau_2 \lambda_1 (\mathbf{g} - \Psi) + \tau_3 \lambda_2 (\mathbf{p} - \Psi), \quad (4.2)$$

wobei $\tau_1, \tau_2, \tau_3 \in \mathbb{R}$ gewählte Konstanten sind und $\lambda_1, \lambda_2 \in \mathbb{R}$ Zufallszahlen, die für jedes Update neu gewählt werden. Die Änderungen der Geschwindigkeit kann in drei Teile untergliedert werden: Durch die Bewegungsträgheit $\tau_1 \mathbf{v}$ können kleine lokale Anstiege überwunden werden. Mit dem sozialen Anteil $\tau_2 \lambda_1 (\mathbf{g})$ wird die Entfernung zur besten bekannten globalen Position $\mathbf{g} \in \mathbb{R}^n$ berücksichtigt und eine Geschwindigkeit in diese Richtung hinzugefügt. Der kognitive Anteil $\tau_3 \lambda_2 (\mathbf{p} - \Psi)$ bezieht sich auf die Abweichung zur besten Position des Partikels $\mathbf{p} \in \mathbb{R}^n$ und ergänzt eine Geschwindigkeit in dessen Richtung. Mit \mathbf{v} kann ein Partikel wie folgt aktualisiert werden:

$$\Psi^{\text{new}} = \Psi + \mathbf{v}. \quad (4.3)$$

Die Funktionsweise einer PSO ist in Abb. 4.1 dargestellt. Zu Demonstrationszwecken wurde der Maximalwert der Beale Funktion bestimmt. Gemäß dieser sind die Diagramme farblich hinterlegt. Der Wert der Funktion wird in logarithmischer Skalierung nebenbei angegeben. Auf der Abszisse ist \mathbf{x} und auf der Ordinate \mathbf{y} angegeben. Es wurden 50 Partikel über 5 Iterationen optimiert. Für eine bessere Übersichtlichkeit wurde die Bewegung von nur einem Partikel eingezeichnet. Diese ist entsprechend der drei Anteile des Geschwindigkeitsupdates farblich untergliedert. Die Partikelpositionen sind als Punkte dargestellt und nach der Iteration beschriftet.

In der Abb. 4.1 ist zu erkennen, dass im ersten Iterationsschritt die Bewegung fast ausschließlich auf den sozialen Anteil zurückzuführen ist. Durch die Initialisierung ist die zufällige Bewegungsträgheit gering und der kognitive Anteil gleich null. Bei der zweiten Iteration haben sowohl Bewegungsträgheit als auch der soziale Anteil Einfluss auf die Bewegung. Der kognitive Anteil ist abermals gleich null, da im zweiten Schritt die erste Position des Partikels die beste ist. Bei der dritten Iteration sind alle drei Anteile zu beobachten, wobei ein erheblicher Anteil der Bewegungsträgheit von dem kognitiven Anteil überlagert wird. Dies ist darauf zurückzuführen, dass die zweite Position einen höheren Funktionswert hervorbringt, weshalb der kognitive Anteil hier der Bewegungsträgheit direkt entgegenwirkt. Bei den weiteren Iterationsschritten ist zu erkennen, dass sich die Richtung der sozialen Anteile ändert, was auf Updates des globalen Optimums hinweist.

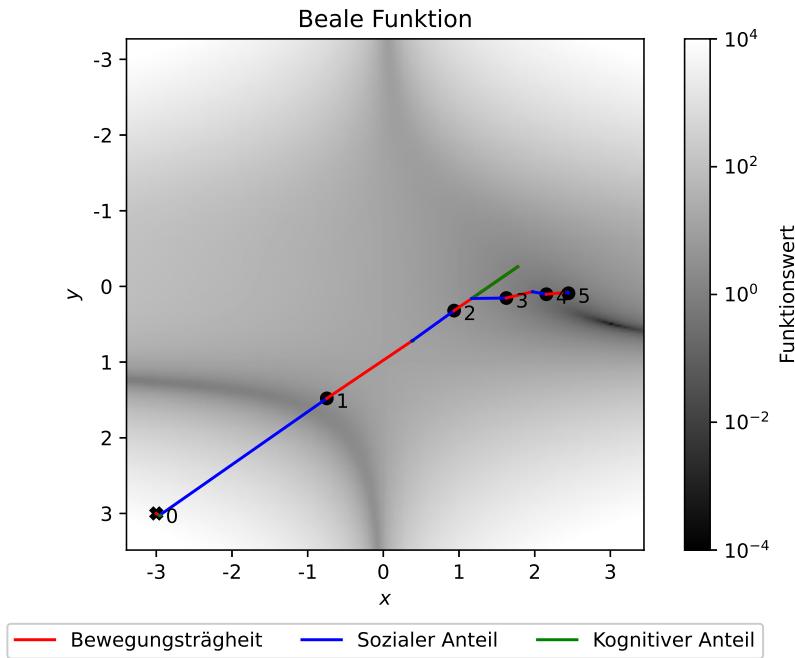


Abb. 4.1: Funktionsweise einer PSO anhand eines Partikels über 5 Iterationen

Die PSO kann mit einem Abbruchkriterium beendet werden – bspw., wenn der globale Bestwert über mehrere Iterationen nicht mehr verbessert wird. Im Anwendungsfall dieser Arbeit ist kein Abbruch möglich, da Relais über ihre Betriebsdauer oder mit Änderungen der Umgebungsparameter ihre Eigenschaften ändern. Deshalb ist eine ständige Optimierung erwünscht. Diese Anforderung ist im Kontext der klassischen PSO problematisch, da sich der soziale und kognitive Anteil des Geschwindigkeitsupdates ausgleichen kann, sodass die Partikel in einem lokalen Minimum feststecken. Weiterhin wird im folgenden Abschn. 4.4 gezeigt, dass dem Prellen ein erhebliches Rauschen unterliegt, was bei der Bestimmung der Partikel individuellen und des globalen Bestwertes hinderlich ist. Zusätzlich erschwert dies das Bilden einer geeigneten Bewegungsträgheit. Daher werden nachfolgend zwei Weiterentwicklungen der PSO vorgestellt.

4.1.2.2 Bare-Bones-PSO

Nachfolgend wird die „Bare-Bones Particle-Swarm-Optimization“ (dt. rudimentäre Partikel Schwarm Optimierung) (BPSO) von *Kennedy* vorgestellt [117]. Diese unterscheidet sich von der PSO durch die Entfernung der Geschwindigkeit der Partikel:

$$\Psi^{\text{new}} = \Psi + \tau_2 \lambda_1 (g - \Psi) + \tau_3 \lambda_2 (p - \Psi) . \quad (4.4)$$

Mit Gl. 4.4 wird die Partikelposition ohne Bewegungsträgheit direkt aktualisiert. Am restlichen Aufbau der klassischen PSO werden keine Änderungen vorgenommen. *Kennedy* nahm Untersuchungen zur Performanz der BPSO gegenüber der PSO vor und kam zum

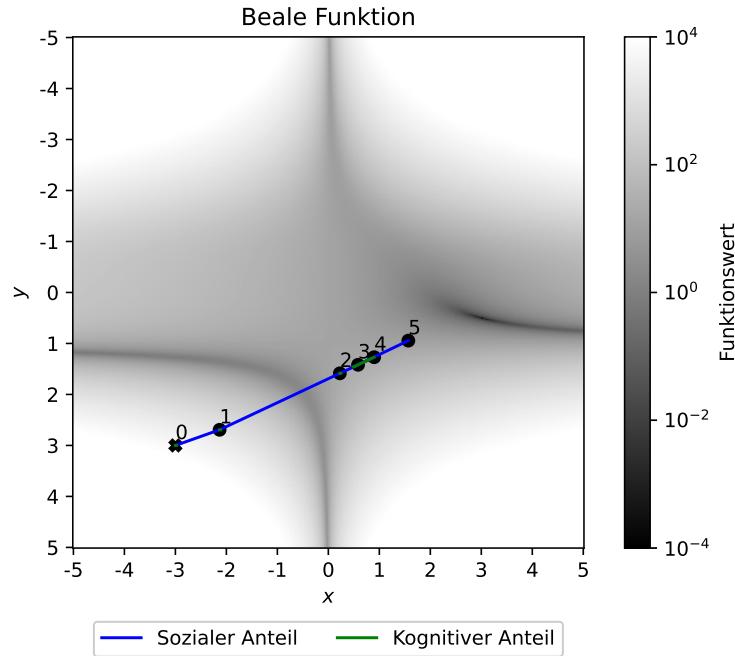


Abb. 4.2: Funktionsweise einer BPSO anhand eines Partikels über 5 Iterationen

Schluss, dass das Entfernen der Bewegungsträgheit keine Nachteile in Bezug auf die Performanz mit sich bringt.

In Abb. 4.2 sind analog zu Abb. 4.1 fünf Iterationsschritte eines von 50 Partikeln zu sehen, wobei die Startpositionen identisch sind. Der wesentliche Teil der Bewegung beruht auf dem sozialen Anteil. Der kognitive Anteil wirkt in den Iterationen drei bis fünf der Bewegung entgegen, da bis zu diesem Schritt die beste Position des Partikels die zweite ist.

4.1.2.3 Evolutionäre PSO

Die „Evolutionary Particle-Swarm-Optimization“ (dt. evolutionäre Partikel Schwarm Optimierung) (EPSO) wurde von *Miranda* und *Fonseca* vorgestellt [118]. Vorteilhaft sind die schnellere Konvergenz und das veränderte Erkundungsverhalten, wodurch das Feststecken in lokalen Optima verhindert wird. Die Veränderung des Verfahrens liegt darin, einzelne Partikel gelöscht werden und andere dafür mutieren. Dieses Verhalten ist auf den „Survival of the Fittest“ Gedanken aus der Evolutionstheorie zurückzuführen. Auf diesem Weg können Partikel, die in einem lokalen Optimum feststecken, eliminiert werden und alle Partikel konvergieren gegen das globale Optimum. Dazu werden folgende Schritte wiederkehrend durchgeführt:

1. Evaluation: Der Funktionswert der Positionen der Partikel wird bestimmt.
2. Auswahl: Die Partikel mit dem schlechtesten Funktionswert werden verworfen.

3. Replikation: Entsprechend der Anzahl der zuvor gelöschten Partikel werden neue Partikel durch Replikation von den behaltenden Partikeln geschaffen.
4. Mutation: Die Gewichte der Partikel w_{ik} werden zufällig verändert.
5. Reproduktion: Die Positionen der Partikel werden gemäß Gl. 4.5 aktualisiert.

Die Reproduktion eines Partikels Φ basiert aus Gl. 4.4, jedoch wurden die Zufallszahlen λ_1, λ_2 aus dem Geschwindigkeitsupdate gestrichen, sodass sich folgende Gleichung ergibt:

$$v^{\text{new}} = \tau_1^* v + \tau_2^* (g^* - \Psi) + \tau_3^* (p - \Psi), \quad (4.5)$$

wobei die Konstanten $\tau_1^*, \tau_2^*, \tau_3^*$ vor jeder Reproduktion im Schritt „Mutation“ gemäß

$$\tau^* = \tau + \mu_1 \mathcal{N}(0, 1) \quad (4.6)$$

aktualisiert werden. Dabei ist $\mathcal{N}(0, 1)$ eine Normalverteilung mit einer Varianz von eins und einem Mittelwert von null und $\mu_1, \mu_2 \in \mathbb{R}$ Parameter, die konstant gewählt werden können oder ebenfalls mutiert. Schlussendlich wird das globale Optimum während der Mutation verrauscht:

$$g^* = g + \mu_2 \mathcal{N}(0, 1). \quad (4.7)$$

Miranda und *Fonseca* konnten nachweisen, dass die EPSO im Vergleich zur PSO wesentlich schneller zu einem geringeren Funktionswert konvergiert und zusätzlich leichter anzuwenden ist aufgrund der reduzierten Parameteranzahl. Deshalb sind im Bereich der Optimierungsalgorithmen mit Population evolutionäre Verfahren der Stand der Technik.

Die Funktionsweise von EPSO ist in Abb. 4.3 dargestellt. Der Aufbau der Grafik entspricht denen aus den vorherigen zwei Abschnitten. Es sind fünf Iterationsschritte eines Partikels visualisiert, die sich dahin gehend unterscheiden, dass sie nicht alle zusammenhängend sind. Dafür ist der evolutionäre Mechanismus verantwortlich, durch den der Partikel in den Iterationen zwei und drei durch Replikation eine neue Position bekommt.

4.1.3 Box-Muller-Transformation

Die Box-Muller Methode ist eine Transformation, mit der aus zwei unabhängigen Zufallszahlen $r, s \in \mathbb{R}$ einer Gleichverteilung im Intervall $[0, 1]$, zwei Zufallszahlen $x, y \in \mathbb{R}$ berechnet werden können, die einer Standardnormalverteilung unterliegen [119]. Für jede Zufallszahl ist dabei nur ein Berechnungsschritt notwendig:

$$x = \sqrt{-2 \ln(r)} \cos(2\pi s), \quad (4.8)$$

$$y = \sqrt{-2 \ln(r)} \sin(2\pi s). \quad (4.9)$$

Die beiden Formeln unterscheiden sich in der Trigonometrischen Funktion. In [119] wird nachgewiesen, dass x, y standardnormalverteilt und (stochastisch) unabhängig sind. Die Varianz σ^2 liegt bei 1 und der Mittelwert μ bei 0.

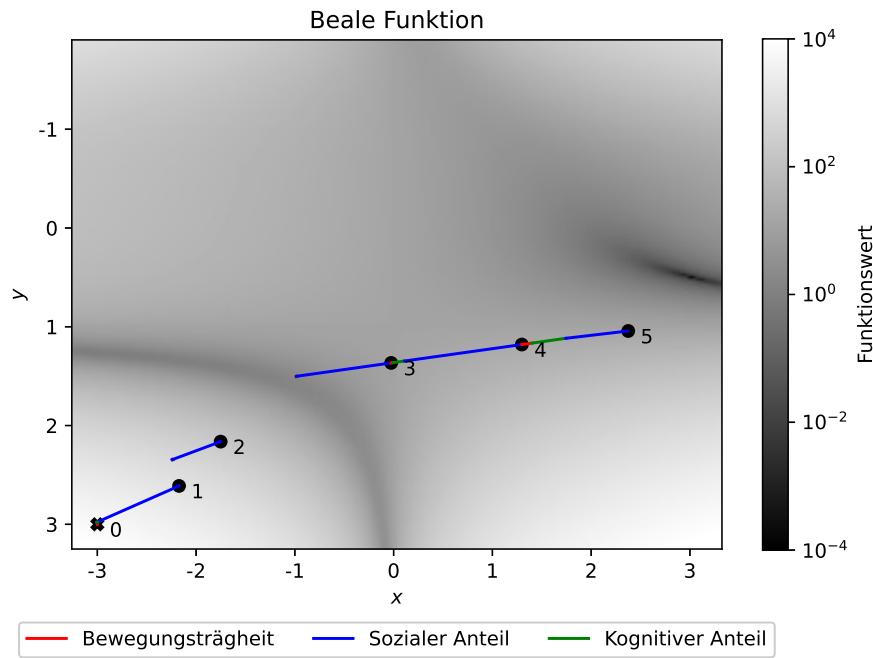


Abb. 4.3: Funktionsweise einer EPSO anhand eines Partikels über 5 Iterationen.

4.2 Messaufbau

Zur Durchführung der Untersuchungen wurde ein eigener Hardwareaufbau entwickelt. Dieser umfasst einige Bauelemente und einen Mikrocontroller, sodass von der Messung der physikalischen Größen über deren Verarbeitung bis hin zur Realisierung der Ansteuerung alles in einem autarken System realisiert wird. Abgesehen von einer Energieversorgung bestehen somit keine weiteren Abhängigkeiten. Dies ist für die Applikation im Produkt von hoher Bedeutung, da zusätzliche Schnittstellen und externe Rechenkapazitäten sowohl zu einer höheren Systemkomplexität als auch zu höheren Anschaffungskosten führen würden.

In Abb. 4.4 ist der verwendete Hardwareaufbau dargestellt. Entsprechend der Aufteilung des Relais in Steuer- und Lastseite, lässt sich auch der Aufbau untergliedern. Die Steuerseite besteht neben der Relaisspule aus zwei Transistoren, drei Widerständen und einer Freilaufdiode. Letztere wird benötigt, um die Bauteile in der Schaltung vor der Selbstinduktion der Relaisspule zu schützen. Beim Abschalten der Versorgungsspannung kann durch die Freilaufdiode Strom fließen und hohe Spannungen durch Selbstinduktion können verhindert werden. Die Widerstände werden zur Strombegrenzung für die Transistoren eingesetzt. Bei den Transistoren handelt es sich um jeweils einen pnp und einen npn Transistor. An die Basis des npn Transistors ist ein digitaler Ausgang des Mikrocontrollers u_{cntrl} angeschlossen, sodass die Versorgungsspannung der Spule mit den 0 bzw. 3.3 V des Controllers geschaltet werden kann. Zur Visualisierung der Arbeitsweise des Algorithmus wurde zusätzlich der Spulenstrom i_{coil} erfasst, welcher im Algorithmus jedoch nicht einbezogen wird.

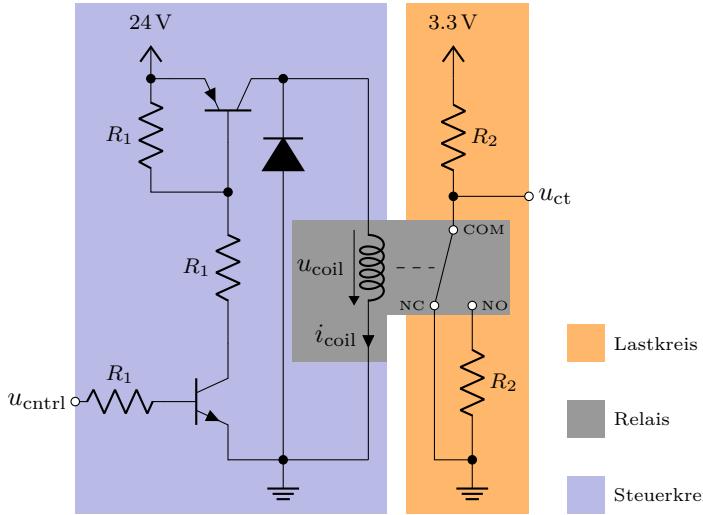


Abb. 4.4: Steuer- und Lastkreis des genutzten Messaufbaus [100]

Auf der Lastseite des Relais sind zwei Widerstände an das Relais angeschlossen, die als Spannungsteiler dienen. Die geschaltete Spannung beträgt 3,3 V, damit diese direkt mit dem ADC des Mikrocontrollers erfasst werden kann. Die Schaltung ist so konzipiert, dass das Kontaktüberwachungssignal u_{ct} drei Spannungslevel je nach Kontaktierung annimmt:

- 0 V: Im ausgeschalteten Zustand leiten die Kontakte NC und COM und u_{ct} liegt folglich bei 0 V.
- 1,65 V: Im eingeschalteten Zustand leiten die Kontakte COM und NO und durch den Spannungsteiler liegt u_{ct} bei 1.65 V.
- 3,3 V: Zwischen dem ein- und ausgeschalteten Zustand ergibt sich ein Bereich, in dem kein Kontaktpaar leitend ist. Folglich liegt u_{ct} bei 3,3 V.

Die vorgestellte Schaltung ist nicht für eine Anwendung in einem Produkt geeignet, generiert jedoch ein vergleichbares Signal wie die in [91] beschriebene Schaltung. Mit der im Patent dargelegten Sensierschaltung mit galvanischer Trennung mittels Übertragung ist dies jedoch möglich.

In Abb. 4.5 ist die verwendete Hardware abgebildet. Zur Beschleunigung des Entwicklungsprozesses und der Evaluierung wurde die Schaltung insgesamt 16-mal auf einer Platine aufgebaut. Die Bauteile sind auf der Rückseite der dargestellten Lochrasterplatine aufgebracht. Als Mikrocontroller wurde ein NUCLEO F7 Evaluierungsboard von STM genutzt.

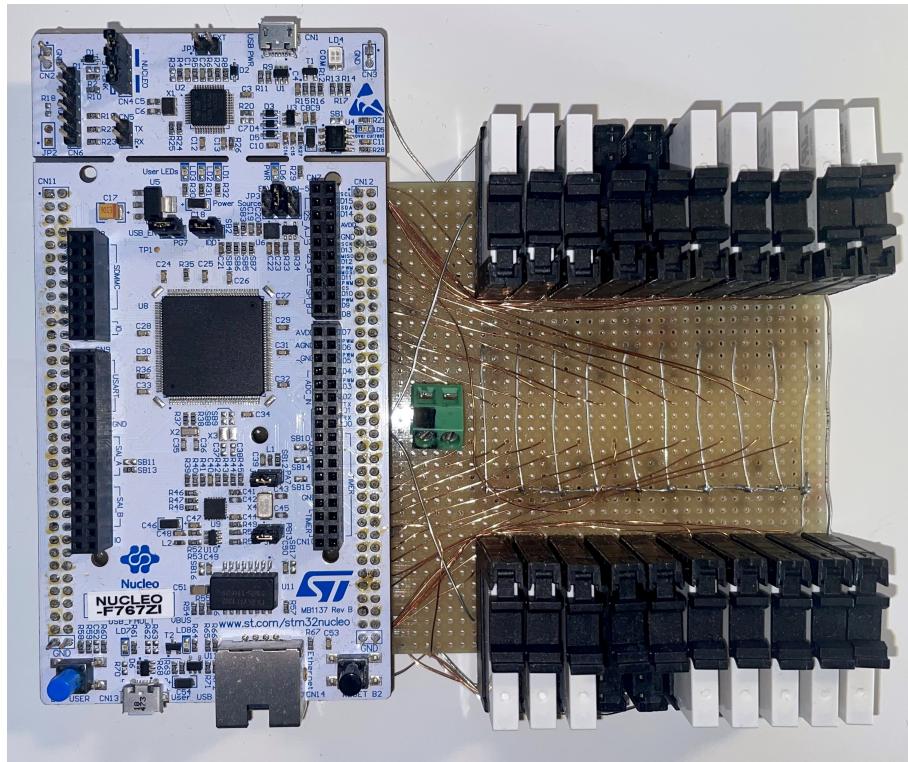


Abb. 4.5: Foto vom genutzten Hardwareaufbau, mit Mikrocontroller (links) und 16 Relais (rechts)

4.2.1 Problemdefinition

Nachfolgend wird das zu optimierende Problem des Kontaktrellens formal definiert. Der Schaltzustand wird durch die Versorgungsspannung bestimmt, deren Form verändert wird, um die Prelldauer zu minimieren. Letztere kann durch Auswertung des Kontaktsignals bestimmt werden, sodass das Problem in den Bereich der messungsbasierten Optimierung fällt. In diesem Fall muss das Problem online für jeden Schaltvorgang gelöst werden und kann als folgendes optimales Steuerungsproblem formuliert werden:

$$\begin{aligned}
 & \text{minimiere} \quad B(x^i(t)) \\
 & \text{vorbehalten} \quad \dot{x}(t) = F(x^i(t), u^i(t), r^i(t)), \\
 & \quad x^i(0) = x_0, \\
 & \quad T(x^i(t)) \leq 0, \quad \forall t > 0 \\
 & \text{gegeben} \quad B(x^j(t)), \quad \forall j < i.
 \end{aligned} \tag{4.10}$$

Wobei $\mathbf{u}^i(t)$ der Steuerspannungsvektor ist, die Funktion \mathbf{B} bestimmt die Prelldauer des Schaltzyklus i , $\mathbf{x}^i(t)$ ist der Zustandsvektor, $\mathbf{r}^i(t)$ ist der Störungsvektor, die Funktion \mathbf{F}

beschreibt die Systemdynamik. \mathbf{x}_0 ist der Anfangszustand und die Funktion \mathbf{T} definiert die Endzustandsbeschränkungen.

4.3 Ableitung eines Optimierungsproblems

Anhand der Abb. 4.6 und 4.7 wird die Vorgehensweise bei der Prellreduzierung und das damit einhergehende Optimierungsproblem veranschaulicht. Das Vorgehen ist von [112] abgeleitet. Beide Abbildungen sind identisch aufgebaut und unterscheiden sich durch den dargestellten Schaltvorgangstypen (Ein- oder Ausschalten). Sie bestehen aus zwei Diagrammen, die übereinander angeordnet sind. Im oberen Diagramm ist ein normaler Schaltvorgang abgebildet, im unteren ein optimierter. Es sind drei Messgrößen zu erkennen, der Spulenstrom i_{coil} , das Kontaktüberwachungssignal u_{ct} und die Versorgungsspannung der Spule u_{coil} . Das Kontaktüberwachungssignal kann drei Werte annehmen, die den Kontaktierungszustand widerspiegeln:

- 0,0: Die Relaiskontakte NC und COM sind kontaktiert.
- 0,5: Die Relaiskontakte COM und NO sind kontaktiert.
- 1,0: Die Relaiskontakte NC, COM und NO sind alle nicht kontaktiert (COM ist in Bewegung).

Weitere Details zur verwendeten Hardware für diese Messungen sind in Abschn. 4.2 erläutert. Beide Spannungen u_{ct} und u_{coil} wurden zur besseren Übersichtlichkeit so normalisiert, dass ihre Werte im Intervall $[0, 1]$ liegen.

Der Einschaltvorgang ist in Abb. 4.6 dargestellt, vor 0 ms sind alle Messgrößen gleich null. Von diesem Zeitpunkt an wird jedoch die maximale Versorgungsspannung angelegt, sodass in der Folge u_{coil} auf 1,0 springt und i_{coil} beginnt bis etwa 3 ms anzusteigen. Von da an sinkt der Strom leicht, da der Anker des Relais sich in Bewegung setzt und eine entgegengesetzte Spannung induziert. Bei ca. 4 ms springt u_{ct} auf 1,0; die Kontakte lösen sich. Die Flugzeit beträgt etwa 1 ms, sodass sich die Kontakte COM und NO bei 5 ms erstmals berühren, folglich fällt u_{ct} auf 0,5 ab. Bis zur Zeitmarke 6 ms sind zwei Beobachtungen zu machen: Bei i_{coil} ist ein lokales Minimum auszumachen, das mit dem Aufschlag des Ankers auf die Spule zu begründen ist. Der Anker erreicht die Spule erst nach dem Schließen der Kontakte, um einen Überhub und damit eine Anpresskraft der Kontakte zu generieren. Weiterhin springt u_{ct} häufig zwischen den Werten 0,5 und 1,0: In diesem Intervall ist das Prellen der Kontakte zu erkennen. Im Diagramm sind die normale Einschaltzeit $s_{\text{ein, normal}}$ und die normale Prelldauer $b_{\text{ein, normal}}$ mit Pfeilen markiert, um diese mit den optimierten Werten zu vergleichen.

Beim optimierten Schaltvorgang wird u_{coil} kurzzeitig ausgeschaltet, sodass die Beschleunigung des Ankers reduziert wird. Dazu wird eine Parametrisierung von u_{coil} vorgenommen: Zum Zeitpunkt t_{ein} , der sich auf den Beginn des Schaltvorgangs bezieht, wird u_{coil} für die Dauer d_{ein} abgeschaltet. Unterschiede zum normalen Schalten sind im Zeitintervall zwischen 2 und 6 ms zu beobachten: Von 2 bis 3 ms ist u_{ct} abgeschaltet, weshalb i_{coil}

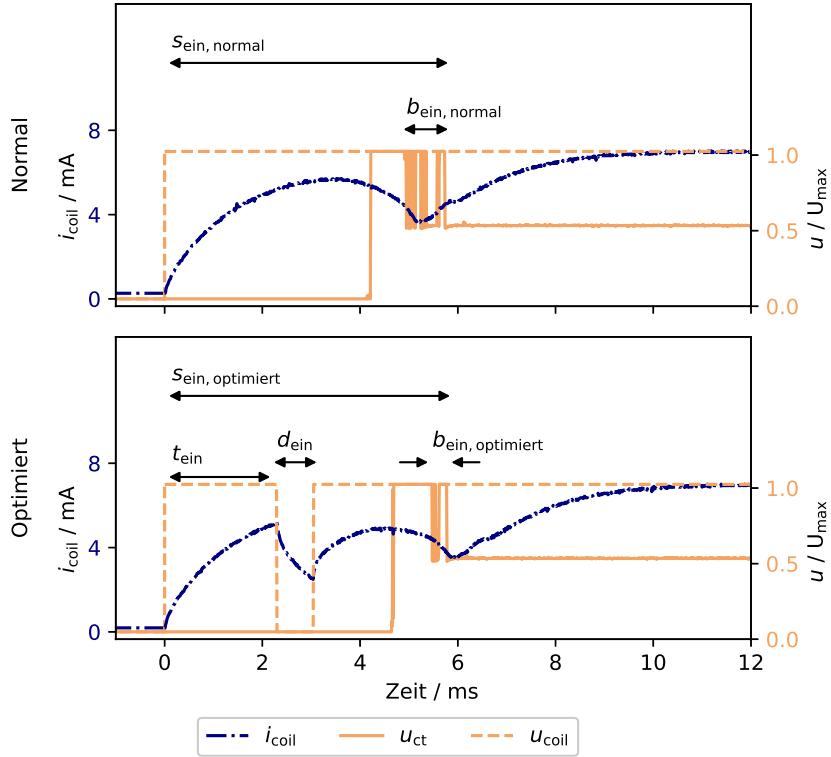


Abb. 4.6: Beispielhafter normaler und optimierter Einschaltvorgang eines Relais [100]

abfällt. Der Zeitpunkt, zu dem sich die Kontakte NC und COM lösen, findet 0,5 ms später statt als beim normalen Schaltvorgang und die darauffolgende Flugphase ist etwas länger. Die Prelldauer $b_{\text{ein, optimiert}}$ ist im Vergleich zu $b_{\text{ein, normal}}$ stark reduziert, sodass die Einschaltzeit $s_{\text{ein, optimiert}}$ der normalen $s_{\text{ein, normal}}$ entspricht.

Zusammenfassend lässt sich festhalten, dass durch zwei Parameter ein Versorgungsspannungssignal u_{coil} definiert werden kann, durch das die Ankerbewegung dahin gehend manipuliert wird, dass das Prellen eine geringere Dauer aufweist.

Der normale Ausschaltvorgang ist im oberen Diagramm von Abb. 4.7 visualisiert. Vor 0 ms ist das Relais eingeschaltet, d.h. i_{coil} beträgt 7 mA, u_{coil} ist 1,0 und die Kontakte COM und NO sind verbunden, weshalb u_{ct} gleich 0,5 ist. Ab 0 ms wird u_{coil} abgeschaltet und i_{coil} fällt ab. Am Kontaktstatus ändert sich zunächst nichts, da der Anker weiterhin an die Spule herangezogen wird. Ab ca. 5 ms ist ein Anstieg von i_{coil} zu sehen. Dies ist mit der Bewegung des Ankers zu begründen, durch die eine Spannung in die Relaisspule induziert wird, die wiederum einen erhöhten Stromfluss hervorruft. Kurz danach ändert sich u_{ct} und der COM-Kontakt bewegt sich in Richtung Ruhelage. Die zeitliche Differenz zwischen Anker- und Kontaktbewegung ist auf den Überhub zurückzuführen. Beim Ausschalten

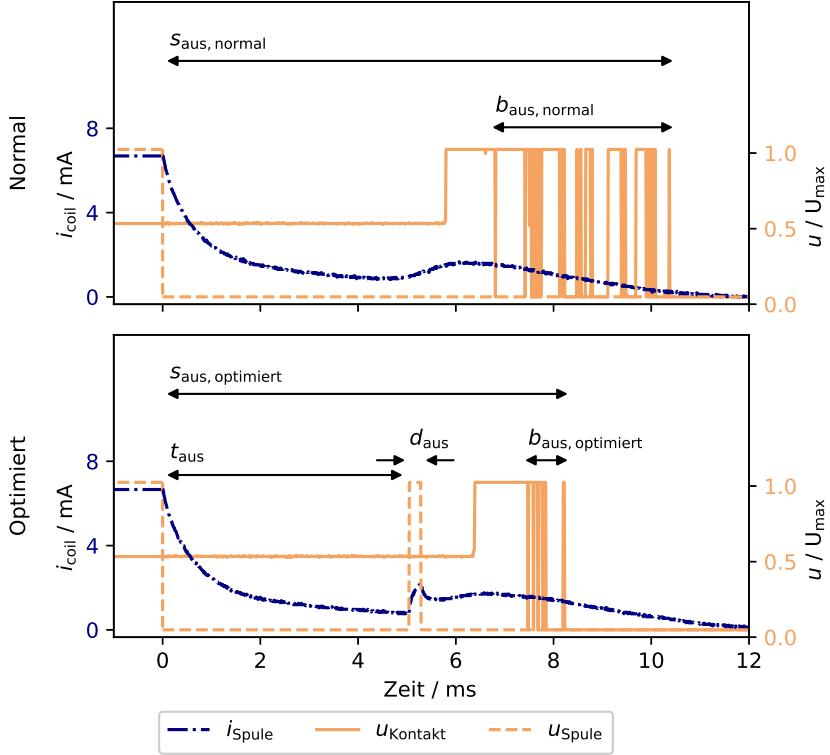


Abb. 4.7: Beispielhafter normaler und optimierter Ausschaltvorgang eines Relais [100]

beträgt die Flugzeit etwas mehr als 1 ms. Danach ist ebenfalls Prellen zu erkennen, das mehr als 3 ms andauert.

Auch beim Ausschalten wird u_{coil} durch zwei Parameter so verändert, dass kurzzeitig eine der Ankerbewegung entgegengerichtete Kraft generiert wird. Dazu wird u_{coil} zu einem Zeitpunkt t_{aus} für die Dauer d_{aus} wieder eingeschaltet, sodass der Strom in der Spule ansteigt. Dies ist im unteren Diagramm von Abb. 4.7 ab etwa 5 ms zu erkennen, wobei der Zeitpunkt mit dem Start der Ankerbewegung im Normalfall korrespondiert. Letztere ist durch diese Maßnahme deutlich verlangsamt, was daran zu erkennen ist, dass u_{ct} später auf 1,0 ansteigt und die Flugphase ebenfalls später in das Prellen übergeht. Auch letzteres ist in Hinsicht auf die Dauer stark verändert: Von knapp 3 ms im Normalfall bleiben noch 1 ms im optimierten Fall übrig. Schlussendlich kann die Prelldauer $b_{aus, optimiert}$ gegenüber $b_{aus, normal}$ reduziert werden, bei gleichzeitiger Verkürzung der Schaltzeit $s_{aus, optimiert}$.

Die Unterschiede in Hinblick auf die Prelldauer und die Schaltzeiten sind auf die mechanischen Optimierungen der Hersteller der Relais zurückzuführen, welche den Schwerpunkt in der Entwicklung auf das Einschalten legen. Das liegt daran, dass typischerweise die Kontakte COM und NO vom Endanwender genutzt werden und NC meist ungenutzt bleibt oder zu Überwachung des Schaltzustandes herangezogen wird.

Weiterhin stellt sich die Frage nach anderen möglichen Formen der Manipulation der Versorgungsspannung u_{coil} . *Ramirez-Laboreo et al.* stellten in ihrer Arbeit unter anderem eine ähnliche Form vor, bei der zwei Invertierungen von u_{coil} durchgeführt werden [112]. Dies hat zur Folge, dass die Zahl der zu optimierenden Parameter auch verdoppelt wird, wodurch das Optimierungsproblem komplexer wird. Gleichzeitig konnte jedoch nicht gezeigt werden, dass durch diese komplexere Form eine stärkere Reduzierung des Prellens erreicht werden kann. Das Signal u_{coil} könnte durch eine geeignete Schaltung und Steuerung auf unergründlich viele Arten verändert werden. So ist es denkbar, das Signal in Mikrosekunden große Intervalle zu unterteilen, für die alle einzelne Spannungswerte definiert werden könnten. Selbst bei einer binären Spannungsversorgung (0 V oder 24 V) ergeben sich so 2^n Kombinationsmöglichkeiten, d.h. schon die Unterteilung in 16 Intervalle würde 65.536 mögliche Signalformen ergeben. Dabei ist eine binäre Versorgung schon eine Einschränkung; denkbar sind wesentlich mehr Spannungslevel. Die Anzahl der zu optimierenden Parameter sollte jedoch möglichst gering sein, da die Optimierung erstens auf einem eingebetteten System ausgeführt werden muss, zweitens das Optimierungsproblem in der Komplexität mit der Anzahl der Parameter steigt und drittens die Betriebsdauer von Relais begrenzt ist, bzw. sich deren Eigenschaften während des Betriebs ändern, sodass nur ein begrenztes Budget an Evaluationen zur Verfügung steht. Deshalb wird im Rahmen dieser Untersuchung eine Veränderung über die zwei zuvor erörterten Parameter t und d vorgenommen.

4.4 Darstellung des Forschungsbedarfs

Eine physikalische Modellierung des Prellens ist nur annähernd möglich und bringt in Bezug auf die Applikationshardware – einem Mikrocontroller – einen sehr hohen Rechenaufwand mit sich. Weiterhin wäre u.a. für die Massen der Relaiskomponenten eine komplexe Messtechnik notwendig, was diesen Lösungsansatz gänzlich unökonomisch macht. Zur Herleitung des Forschungsbedarfs werden im Folgenden zunächst zwei statistische Auswertungen zum Prellen von Relais dargestellt.

In Abb. 4.8 ist die Häufigkeitsverteilung der Zeitspanne des Prellens im Ein- und Ausschalten von zwei exemplarischen Relais unterschiedlicher Hersteller als Histogramm dargestellt. Insgesamt sind die Daten aus 210 Schaltzyklen für jede Schalt- und Relaiskombination dargestellt. Beim Vergleich des Prellens von Ein- und Ausschaltvorgängen sind signifikante Unterschiede hinsichtlich der Dauer auszumachen. Während diese beim Einschalten stets unterhalb von 1 ms liegt, sind beim Ausschalten Dauern von 2 bis 9 ms zu beobachten. Die Standardabweichung beträgt 0,5 ms beim Ein- und 2,17 ms beim Ausschalten. Das ist darauf zurückzuführen, dass die Hersteller die Relais auf das Einschalten optimiert haben.

Der Vergleich der Häufigkeitsverteilungen beider Hersteller fällt gemischt aus. Im Ausschalten ähneln sich die Verteilungen: Hersteller B prellt im Mittel etwas weniger, aber die Form ist vergleichbar. Beim Einschalten hingegen sind deutliche Unterschiede in der Form auszumachen: Für Relais A treten geringe Prelldauern zwischen 0 und 0,04 ms mit

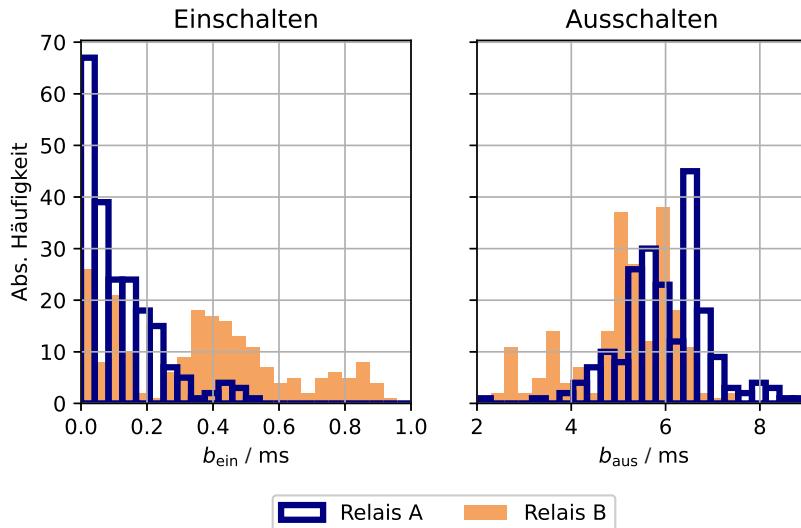


Abb. 4.8: Histogramm vom Prellen zweier normal geschalteter Relais beim Ein- und Ausschalten [100]

Abstand am häufigsten auf. Mit steigender Prelldauer sinkt die Häufigkeit stark, sodass die maximale Prelldauer bei etwa 0.5 ms liegt. Bei Relais B hingegen ist das Prellen zwischen 0 und 1 ms verteilt, wobei es sich in mehreren Bereichen häuft. Zumindest beim Einschalten sind somit herstellerspezifische Unterschiede auszumachen.

Die dargestellten Prelldauern in Abb. 4.8 verdeutlichen, dass das Prellen sowohl vom Schaltvorgangstypen als auch vom Hersteller abhängig ist. Weiterhin weist es eine hohe Standardabweichung auf. Diese Aspekte müssen bei der Entwicklung des Verfahrens berücksichtigt werden.

Neben dem normalen Prellen ist eine Untersuchung der Prelldauer unter Änderung der Parameter t und d der optimierten Ansteuerung von Bedeutung. Eine detaillierte Ausführung der Ansteuerung und der Parameter ist Abschn. 4.3 zu entnehmen. Die Untersuchungsergebnisse sind in Abb. 4.9 dargestellt. Für die beiden Relais aus Abb. 4.8 wurden diverse Kombinationen zehnfach ausgeführt und der Mittelwert bestimmt. Auf diesem Weg sind vier Heatmaps entstanden. Die Spalten sind den Relais zugeordnet und die Zeilen den Schaltvorgangstypen. Die Heatmaps haben alle den gleichen Aufbau: Auf der Abszisse ist die Dauer d und auf der Ordinate der Zeitpunkt t aufgetragen. Die Einfärbung der Pixel gibt die relative Reduktion der Prelldauer gegenüber dem mittleren normalen Prellen des jeweiligen Relais wieder. Dabei wird keine negative Reduktion dargestellt, sondern die Werte bei 0 % abgeschnitten.

Aus den Diagrammen wird deutlich, dass das zu optimierende Problem nicht trivial ist. Es bestehen mehrere lokale Optima, die über den gesamten Suchraum verteilt sind. Zudem unterscheiden sich die Problemlandschaften je nach Relais und Schaltvorgang deutlich. Dies stellt eine Herausforderung für einen geeigneten Algorithmus dar, weil diese Landschaften

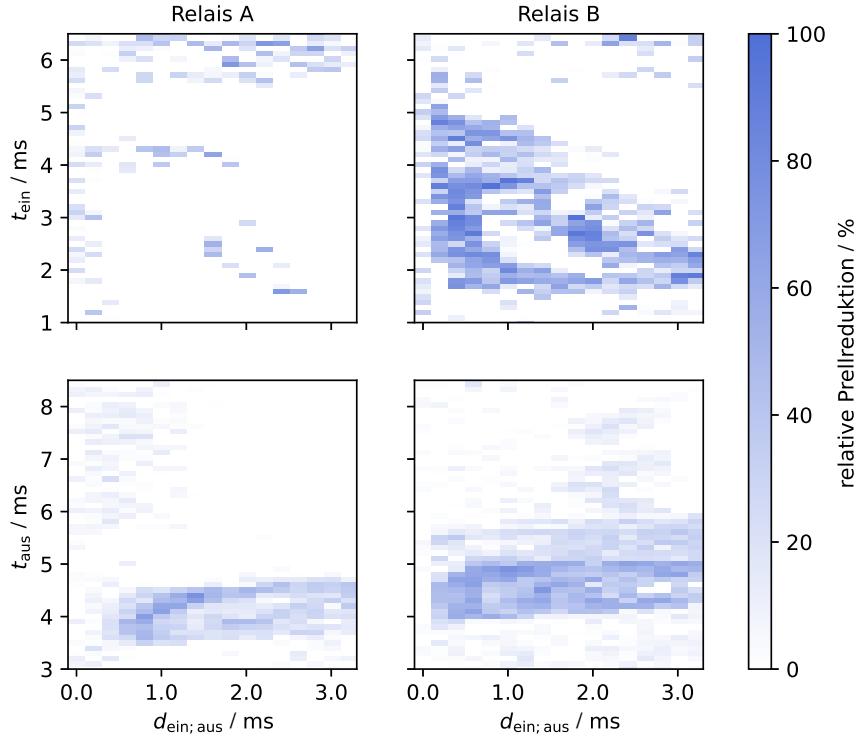


Abb. 4.9: Heatmap der Prellreduktion für unterschiedliche Kombinationen aus $t_{\text{ein; aus}}$ und $d_{\text{ein; aus}}$ [100]

während der Optimierung nicht bekannt sind. Bei Relais A sind beim Einschalten nur geringe Bereiche auszumachen, in denen das Prellen reduziert ist, da der Hersteller die Relais dahin gehend hervorragend optimiert hat. Mit zunehmender Degradierung ist jedoch davon auszugehen, dass sich die Eigenschaften des Relais ändern (s. Kap. 3.4.1), weshalb das normale Prellen zunimmt und mit BOA eine stärkere relative Prellreduktion erreicht werden kann.

4.5 Bouncing-Optimization-Algorithm (BOA)

Entsprechend der zuvor aufgezeigten Herausforderungen muss der „Bouncing-Optimization-Algorithm“ (dt. Preloffoptimierungsalgorithmus) (BOA) robust und simpel zu berechnen sein. Für die grundlegende Auswahl des Optimierungsalgorithmus wird die Taxonomie nach *Stork et al.* herangezogen [120]. Aus dieser ergibt sich nach der Bewertung von vier Kategorien eine Empfehlung. Die Kategorien und Bewertungen werden nachfolgend erörtert:

- Landschaftseigenschaften: Es stehen keine Informationen zur Auswirkung der Startzeit t und Dauer d Kombinationen auf das Prellen bereit.

- Funktionseigenschaften: Aus Abschn. 4.4 geht hervor, dass zumindest mehrere lokale Minima zu erwarten sind. Zusätzlich ist ein starkes Rauschen auszumachen.
- Verfügbares Evaluierungsbudget: Die Lebensdauer von Relais beträgt häufig mehrere zehntausend Schaltzyklen, weshalb hunderte Evaluationen durchgeführt werden können, bevor das Optimum gefunden wird.
- Berechnungs und Algorithmus Komplexität: Die Anwendung muss auf einem Mikrocontroller ausführbar sein. Daher ist die mögliche Komplexität begrenzt.

Gemäß der Taxonomie empfiehlt sich ein Verfahren aus der Klasse „Population“ für das Optimierungsproblem. Deshalb wird im Folgenden BOA ausgehend von einer PSO hergeleitet. Diese wird nachfolgend auf eine Simulation des Prellens angewandt. Dabei handelt es sich um aufgezeichnete Messwerte von 10 Relais, mit denen eine reale Ausführung der Kombinationen aus t und d (im Folgenden auch Partikel genannt) nachempfunden wird. D.h. die zweidimensionale Optimierungsfunktion wurde in ein Raster mit der Schrittweite von 0,1 ms unterteilt, wobei das Prellen jedes Knotenpunktes 30-mal gemessen wurde, um das Rauschen im Kontext der Simulation abzubilden. Dieses Vorgehen wurde gewählt, da zum einen die Ergebnisse reproduzierbar sind und somit die Auswirkungen von Änderungen bei der Entwicklung verifiziert werden können. Zum anderen können forschungsökonomische Vorteile erschlossen werden, denn die notwendige Implementierung in der Programmiersprache C für den Hardwareaufbau und das Durchführen realer Schaltzyklen entfallen. Aufgrund der Diskretisierung im Raster können im Kontext der Simulation nicht alle möglichen t und d abgebildet werden. In der realen Anwendung ist deshalb eine stärkere Reduzierung möglich.

In Abb. 4.10 ist die relative Prellreduzierung verschiedener Verfahren über 50 Iterationen abgebildet. Bei den Verfahren handelt es sich um die beschriebene PSO, deren Varianten BPSO und EPSO sowie den im Rahmen dieser Arbeit entwickelten BOA. Die angegebenen Daten sind über zehn Relais mit jeweils 20 Wiederholungen gemittelt und es handelt sich um den Einschaltvorgang der Relais. Für alle Verfahren wurde eine Population von 48 Partikeln gewählt, damit die Ergebnisse vergleichbar sind. Zudem wurden die Hyperparameter optimiert.

Die PSO und deren vereinfachte Variante BPSO führen zu nahezu identischen Ergebnissen: Das Prellen wird um fast 30 % reduziert. Die Vermutung von *Kennedy* [117], dass der Geschwindigkeitsterm nur ein historisches Überbleibsel ist und keinen funktionalen Mehrwert bietet, lässt sich durch diese Untersuchung unterstützen. Deshalb wird auch bei BOA kein Geschwindigkeitsterm verwendet. Bei der EPSO wurde dem Verfahren eine evolutionäre Komponente hinzugefügt, die schlechten Partikel werden verworfen. Dies führt zu einer stärkeren Prellreduzierung von etwa 45 %, da Partikel, die in lokalen Minima verharren, auf diese Weise verworfen werden. Deswegen wird BOA auch eine evolutionäre Komponente umfassen. Bei der Simulation wird durch BOA das Prellen um etwa 60 % gesenkt, somit übertrifft die Performanz EPSO deutlich. Begründet werden kann dies mit einer weiteren Technik, die nachfolgend erklärt wird.

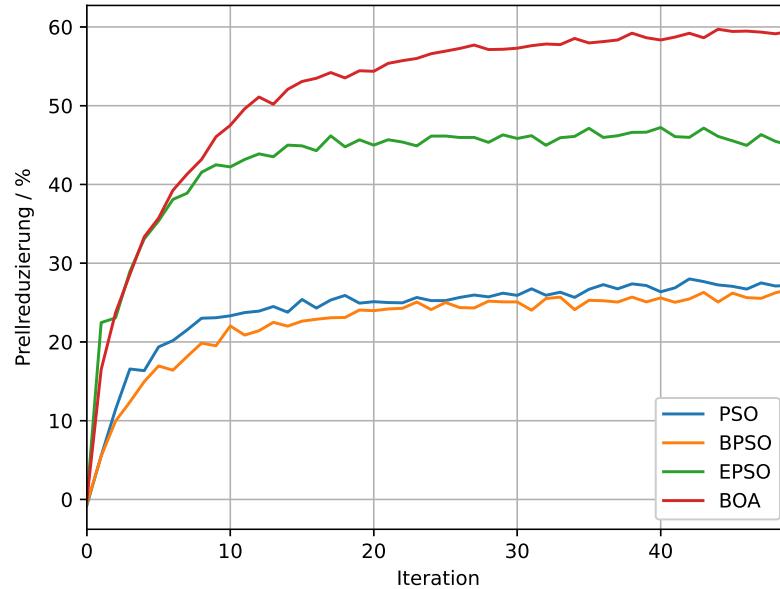


Abb. 4.10: Simulierte relative Prellreduktion von verschiedenen PSO Varianten

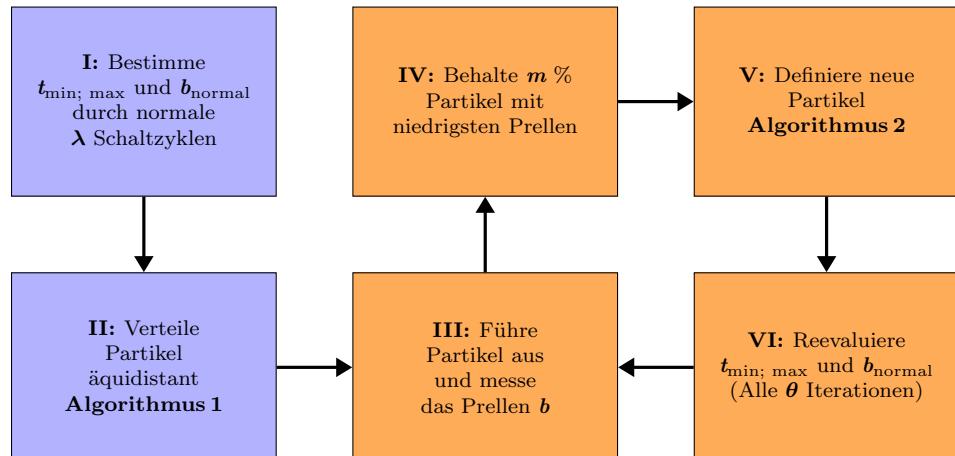


Abb. 4.11: Blockdiagramm des entwickelten Algorithmus [100]

BOA ist in sechs Schritte untergliedert, die in Abb. 4.11 dargestellt sind. Die ersten beiden Schritte sind blau eingefärbt, da diese Teil der Initialisierungsphase sind und deshalb nur einmalig durchlaufen werden. Diese Phase ist notwendig, um den Referenzwert des Prellens ohne Optimierung zu erfassen, den Suchraum einzugrenzen und die Partikel äquidistant darin zu verteilen. Die übrigen vier Schritte folgen fortwährend aufeinander und es kommt zu keinem Abbruch, um jederzeit eine Exploration zu ermöglichen, falls die Landschaft des

Optimierungsproblems sich z.B. durch Verschleiß geändert hat. Eine detaillierte Erörterung der Schritte findet nachfolgend statt:

I: Im ersten Schritt werden λ normale Schaltzyklen ausgeführt und ausgewertet. Dabei ist zum einen das normale Prellen b_{normal} von Interesse, da es während der weiteren Ausführung von BOA als Referenzwert benötigt wird. So kann sichergestellt werden, dass der Algorithmus in keinen lokalen Minima feststecken kann, die mehr prellen als das normale Schalten. Zum anderen werden der minimale und maximale Startzeitpunkt t bestimmt: Das Minimum wird als der Zeitpunkt, zu dem die Kontakte sich das erste Mal lösen, minus 3 ms berechnet. Dieser Wert geht zurück auf die Eigenschaften der untersuchten Relais, denn vor diesem findet weder beim Ein- noch beim Ausschalten eine Bewegung des Ankers statt (vgl. Abb. 4.6 und 4.7). Das Maximum entspricht dem Zeitpunkt, zu dem der Schaltvorgang abgeschlossen ist. Das individuelle Eingrenzen des Suchraums für jedes Relais ist für die erstmalige Generierung der Partikel notwendig und verhindert ein Abdriften der Partikel während des Optimierungsprozesses.

Algorithm 1 BOA Initialisierung [100]

Require: $t_{\min}, t_{\max}, d_{\min}, d_{\max}, n$

- 1: Berechne die Schrittweite und Anzahl der Schritte
- 2: $\omega \leftarrow \text{ceil} \left(\sqrt{((t_{\max} - t_{\min})(d_{\max} - d_{\min})) / n} \right)$
- 3: $n_t \leftarrow \text{floor} \left((t_{\max} - t_{\min}) / \omega \right)$
- 4: $n_d \leftarrow \text{floor} \left((d_{\max} - d_{\min}) / \omega \right)$
- 5: Verteile die Partikel äquidistant
- 6: **for** $i \leftarrow 0$ to $n_t n_d$ **do**
- 7: $\Psi_{i, 0} \leftarrow t_{\min} + \omega \text{ floor} \left(i / n_t \right)$
- 8: $\Psi_{i, 1} \leftarrow d_{\min} + \omega (i \bmod n_d) + 0,2 \text{ ms}$
- 9: **end for**
- 10: Wähle die übrigen Partikel zufällig
- 11: **for** $i \leftarrow n_t n_d$ to n **do**
- 12: $\Psi_{i, 0} \leftarrow \text{random} [t_{\min}, t_{\max}]$
- 13: $\Psi_{i, 1} \leftarrow \text{random} [d_{\min}, d_{\max}]$
- 14: **end for**

II: Der zweite Schritt umfasst die Generierung der ersten Partikel. Da a priori keine Informationen über das Problem bekannt sind, werden diese äquidistant im Suchraum verteilt. Das Vorgehen ist in Algorithmus 1 beschrieben: Der in Schritt **I** bestimmte minimale (t_{\min}) und maximale Startzeitpunkt (t_{\max}), die festgelegte minimale (d_{\min}) und maximale Dauer (d_{\max}) und die Anzahl der Partikel n werden für die Berechnungen benötigt. In Zeile 2 wird die Schrittweite bestimmt, indem die durch die beiden Dimensionen t und d aufgespannte Fläche durch die Partikel geteilt wird. Von dieser Fläche pro Partikel Wert wird die Quadratwurzel gezogen, da durch die Zielsetzung einer äquidistanten Verteilung die Proportion quadratisch sein muss. Das Aufrunden ist notwendig, um die Anzahl der Partikel n nicht zu überschreiten. Die Anzahlen der Schritte pro Dimension n_t und n_d werden in den Zeilen 3 und 4 durch Abrunden des Quotienten aus dem Intervall der

jeweiligen Dimension und der Schrittweite gebildet. Durch die so ermittelten Werte können in den Zeilen 6 bis 9 die Partikel gemäß der Schrittweite im Suchraum äquidistant verteilt werden. Die 0,2 ms werden bei der Dauer addiert, da für diesen Wert von \mathbf{d} ein normaler Schaltvorgang durchgeführt wird. Da nicht immer alle Partikel in das Raster passen, werden die übrigen in den Zeilen 11 bis 14 zufällig im Suchraum platziert.

III: Die Partikel, d.h. \mathbf{t} und \mathbf{d} Kombinationen werden im Schritt **III** ausgeführt. Zu jedem Schaltzyklus wird die Dauer des Prellens \mathbf{b} ausgewertet.

IV: Diese Prelldauer ist die Bezugsgrundlage für den vierten Schritt, im Rahmen dessen der evolutionäre Teil des BOA umgesetzt wird. Nur die $\mathbf{m}\%$ Partikel mit dem niedrigsten Prelldauern werden behalten, alle anderen werden verworfen.

V: Auf Basis der behaltenen Partikel werden im fünften Schritt neue Partikel generiert. Die Vorgehensweise wird in Algorithmus 2 detailliert erörtert: Dazu werden die Grenzen des Suchraums \mathbf{t}_{\min} , \mathbf{t}_{\max} , \mathbf{d}_{\min} , \mathbf{d}_{\max} ; das normale Prellen $\mathbf{b}_{\text{normal}}$; die Anzahl der Nachbarn \mathbf{k} ; die Prozentzahl der verworfenen Partikel \mathbf{m} ; die Zahl der Partikel \mathbf{n} ; die Partikel Ψ und das zu den Partikeln zugeordnete Prellen \mathbf{b} benötigt. In Zeile 2 wird eine Iteration gestartet, die so viele Wiederholungen beinhaltet, dass für jeden der in Schritt **IV** verworfenen Partikel ein neuer generiert wird. Zu diesem Zweck wird in Zeile 4 ein Partikel, der behalten wurde, als Ursprung gewählt. Bis hierher ist das Prozedere zur Generierung analog zum EPSO-Algorithmus. Doch bei BOA findet eine Reevaluation der bestehenden Partikel statt, weshalb diese nicht mutiert werden. Dies ist aufgrund des Rauschens notwendig. Weiterhin werden die neuen Partikel nicht auf Basis eines individuellen und globalen Optimums modifiziert. Bei BOA wird, wie beim Clusteralgorithmus DBSCAN, die lokale Dichte geschätzt und daraufhin die Aktualisierung der Partikel vorgenommen.

Die Kernidee ist dabei, dass in Regionen, in denen viele Partikel behalten wurden, scheinbar ein Optimum vorliegt. Deshalb ist es wünschenswert, dass in diesem Bereich neue Partikel mit einer hohen Dichte entstehen. Umgekehrt sind in Regionen mit wenig Partikeln geringe Verbesserungen zu erwarten, weshalb die neuen Partikel eine geringe Dichte aufweisen sollten. Zur Schätzung der Dichte wird in den Zeilen 6 bis 8 zunächst die euklidische Distanz des Ursprungspartikels zu den übrigen Partikeln, die behalten wurden, berechnet. Die Dichte wird in Zeile 10 über die mittlere euklidische Distanz zu den \mathbf{k} nächsten Nachbarn angenähert.

In den Zeilen 12 bis 17 werden dann schlussendlich die neuen Partikel generiert. Dabei wird eine Normalverteilung mit dem Ursprungspartikel als Erwartungswert und der Dichtannäherung \mathbf{r} als Varianz angestrebt. Zur Realisierung auf einem Mikrocontroller wird das Box-Muller Verfahren genutzt. Mit diesem Verfahren ist eine Transformation von gleich verteilten zu normal verteilten Zufallszahlen möglich. Das Prozedere ist in eine Schleife eingebunden, durch die sichergestellt wird, dass die Suchraumgrenzen eingehalten werden. Die Schleife ist dabei notwendig, um die Normalverteilung an den Grenzen des Suchraums aufrechtzuerhalten. Ein Begrenzen der Werte würde die Wahrscheinlichkeit, einen Punkt auf der Grenze zu treffen, deutlich erhöhen.

Die letzten Zeilen 19 bis 22 beschreiben einen Mechanismus, der die Balance zwischen Exploration und Ausnutzung anpassen kann. Grundsätzlich weist BOA in der ersten Iteration ausschließlich Exploration vor, die durch den evolutionären Anteil und die dichtebasierter Generation mit zunehmenden Iterationen immer stärker zurückgefahren wird, sodass die Ausbeute gesteigert werden kann. Ein ausschließliches Ausbeuten, also ein Stopp der Iterationen und ständiges Wiederholen des besten Partikels beim Einsatz von Relais, ist nicht realisierbar, da deren Eigenschaften von den Umgebungseinflüssen abhängig sind. Die Anpassung der Balance ist erforderlich, wenn sich Veränderungen in der Optimierungsfunktion ergeben, die dazu führen, dass das normale Prellen b_{normal} geringer ist als das durch BOA. Wenn dies der Fall ist, dann werden die Partikel zufällig im Suchraum verteilt und nicht vom Ursprungspartikel mutiert, so wird die Exploration intensiviert.

Algorithm 2 BOA Hauptoptimierungsschleife [100]

Require: $t_{\min}, t_{\max}, d_{\min}, d_{\max}, b_{\text{normal}}, k, m, n, \Psi, b$

- 1: Definiere neue $(1 - m) n$ Partikel
- 2: **for** $i \leftarrow m n$ **to** n **do**
- 3: Wähle einen behaltenden Partikel als Ursprung
- 4: $j \leftarrow i \bmod (m n)$
- 5: Berechne euklidische Distanzen e zu den Nachbarn
- 6: **for** $l \leftarrow 0$ **to** $m n$ **if** $l \neq j$ **do**
- 7: $e_l \leftarrow \sqrt{(\Psi_{l,0} - \Psi_{j,0})^2 + (\Psi_{l,1} - \Psi_{j,1})^2}$
- 8: **end for**
- 9: Berechne mittlere Distanz zu den k nächsten Nachbarn
- 10: $r \leftarrow \text{mean } (k \text{ lowest distances in } e)$
- 11: Wähle $\Psi_{i,0}$ und $\Psi_{i,1}$ unter Einhaltung der Limits
- 12: **while** $\Psi_{i,0} \notin [t_{\min}, t_{\max}]$ und $\Psi_{i,1} \notin [d_{\min}, d_{\max}]$ **do**
- 13: $\alpha \leftarrow \text{random } [0, 2\pi[$
- 14: $\mu \leftarrow r \sqrt{-2 \ln(\text{random } [0, 1])}$
- 15: $\Psi_{i,0} \leftarrow \Psi_{j,0} + \mu \cos \alpha$
- 16: $\Psi_{i,1} \leftarrow \Psi_{j,1} + \mu \sin \alpha$
- 17: **end while**
- 18: Zufällig, wenn der Ursprung das Prellen nicht reduziert
- 19: **if** $b_j > b_{\text{normal}}$ **then**
- 20: $\Psi_{i,0} \leftarrow \text{random } [t_{\min}, t_{\max}]$
- 21: $\Psi_{i,1} \leftarrow \text{random } [d_{\min}, d_{\max}]$
- 22: **end if**
- 23: **end for**

VI: Der letzte Schritt vor der Wiederausführung des Schrittes **III** wird nur alle Ω Iterationen ausgeführt. Technisch ist er identisch mit dem Schritt **I**: Der Referenzwert b_{normal} und die Suchraumgrenzen t_{\min} und t_{\max} werden aktualisiert. Zu begründen ist dieses Vorgehen mit der stetigen Änderung der Eigenschaften von Relais.

Aus der Beschreibung vom BOA geht hervor, dass einige Variablen zur Ausführung benötigt werden. Diese sind in Tab. 4.1 aufgeführt und in die Gruppen konfiguriert, gemessen und optimiert unterteilt. Die konfigurierten Variablen wurden experimentell ermittelt. Die untere Suchraumgrenze für die Dauer d_{\min} beträgt 0 ms, da hier ein physikalisches Limit liegt. Bei der Wahl der oberen Grenze wurde die Dynamik der Relaisspule einbezogen und deshalb als Zeitkonstante der Relaisspule $\tau = L_{\text{Spule}}/R_{\text{Spule}}$ berechnet. Die gemessenen Variablen werden während der Laufzeit bestimmt und können daher nicht im Voraus definiert werden. Schließlich sind die optimierten Variablen die Partikel, die ebenfalls online verändert werden.

Tab. 4.1: BOA Algorithmus Variablen [100]

Beschreibung	Wert
Konfiguriert	
n_k	Anzahl der Nachbarn
m	Prozent der behaltenden Partikel
n	Anzahl der optimierten Partikel
θ	Referenz Reevaluierungsintervall
$d_{\min; \max}$	Unteres; oberes Zeitdauerlimit
λ	Anzahl der initialen Schaltzyklen
Gemessen	
b_{normal}	Prelldauer Referenz
b	Prelldauer
$t_{\min; \max}$	Unteres; oberes Startzeitlimit
Optimiert	
Ψ	Partikelschwarm

4.6 Evaluation

Im folgenden Abschnitt wird die Performanz von BOA evaluiert. Dabei wird zunächst grundlegend die Funktionsweise von BOA präsentiert. Darauf aufbauend wird sowohl ein qualitativer als auch ein quantitativer Vergleich mit dem Stand der Technik gezogen. Dieser umfasst Analyse. Schlussendlich werden Untersuchungen hinsichtlich der Robustheit gegenüber Umwelteinflüssen und zur Auswirkung auf die Schaltzeit vorgestellt.

4.6.1 Funktionsweise

Um die Funktionsweise von BOA unter Beweis zu stellen, werden die Partikel über mehrere Iterationen dargestellt. In Abb. 4.12 sind drei aufeinanderfolgende Iterationen zu sehen, wobei die obere Reihe zu den Einschaltvorgängen korrespondiert, die untere zu den

Ausschaltvorgängen. Der Hintergrund ist entsprechend der relativen Prellreduzierung (vgl. Abb. 4.9) in Graustufen visualisiert, was der Veranschaulichung dient, da die Informationen BOA nicht zur Verfügung stehen. Bereiche, die zu einer Steigerung des Prellens führen, werden mit einer Reduzierung von 0 %, d.h. weiß, angezeigt. Die Partikel für jede Iteration sind in den Diagrammen einer Spalte eingezeichnet, sodass die Abbildung aus insgesamt sechs Diagrammen besteht. Auf der Abszisse ist jeweils die Dauer $d_{\text{ein}; \text{aus}}$ und auf der Ordinate der Zeitpunkt $t_{\text{ein}; \text{aus}}$ aufgetragen. Für jede Iteration sind alle Partikel in zwei Farben dargestellt, grün für die besten 25 %, die behalten werden und rot für die verworfenen Partikel. Zur Visualisierung befindet sich BOA somit immer im Schritt **IV**.

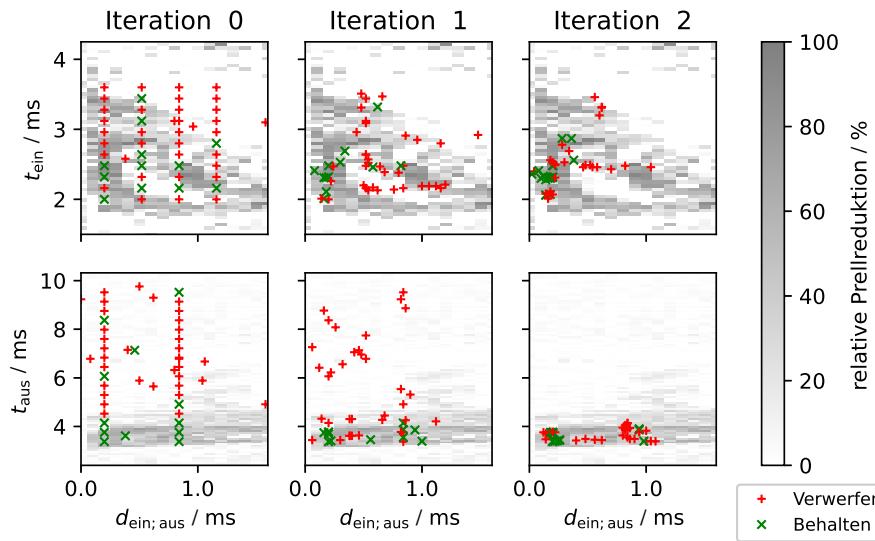


Abb. 4.12: Exemplarische Darstellung der Partikel in den ersten drei Iterationen

In Iteration 0 von Abb. 4.12 ist zu erkennen, dass ein Großteil der Partikel im Raster angeordnet ist. Die Verteilung ist äquidistant. Dabei ist zu beachten, dass die Achsen nicht gleich skaliert sind. Der Startzeit Suchraum wurde beim Ausschalten groß gewählt, da die Ankerbewegung von den individuellen Federkräften der Relais abhängt. Neben den im Raster angeordneten Partikeln sind auch welche zu erkennen, deren Position zufällig gewählt wurde, was in Schritt **II** beschrieben wurde und mit dem Raster zu begründen ist, in dem nicht immer alle Partikel Platz finden können. In Bezug auf die Einfärbung ist zu erkennen, dass die Partikel in Regionen mit hoher Prellreduktion häufig behalten werden, wohingegen Partikel in Regionen mit niedriger Reduktion eher verworfen werden. Abweichungen sind damit zu erklären, dass das Prellen ein hohes Rauschen aufweist, weshalb das gemessene Prellen für die jeweiligen Partikel stark vom Erwartungswert abweichen kann.

Die Generation von neuen Partikeln kann beim Vergleich von zwei Iterationen nachvollzogen werden: Zwischen Iteration 0 und 1 beim Ausschalten ist bspw. zu erkennen, dass der Abstand (die Dichte) von neuen Partikeln in den Regionen mit vielen Partikeln geringer

ist. Schon nach zwei Updates der Partikel ist eine Konzentration in den Regionen mit starker Prellreduktion zu sehen.

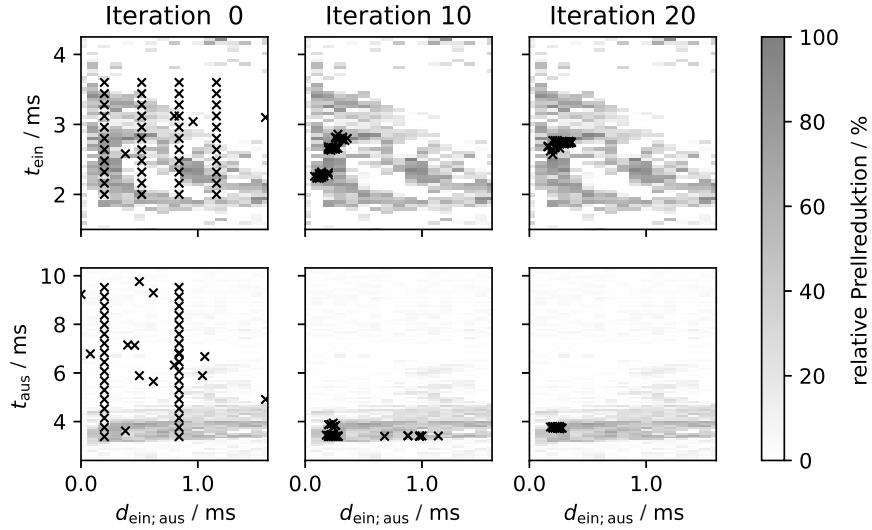


Abb. 4.13: Exemplarische Darstellung der Partikel in der ersten, zehnten und zwanzigsten Iteration [100]

Mit zunehmender Iterationszahl werden die Änderungen der Positionen der Partikel immer kleiner. Angesichts dessen werden in Abb. 4.13 drei Iterationen gezeigt, zwischen denen jeweils 10 Schritte liegen. Die Datengrundlage ist dabei identisch mit Abb. 4.13. Aus der Grafik wird ersichtlich, dass durch BOA eine Aggregation der Partikel stattfindet, die dort zu lokalisieren ist, wo die im Hintergrund dargestellte Prellreduktion maximal ist.

In den Abbildungen ist visualisiert, dass durch BOA zumindest ein lokales Optimum gefunden wurde. Die Performanz wird nachfolgend im Vergleich zum Stand der Technik, dem R2R-Algorithmus, erläutert.

4.6.2 Qualitativ

In der Abb. 4.14 wird das Prellen von zwei Relais mit unterschiedlichen Algorithmen über die ersten 1000 Schaltzyklen abgebildet. Die angegebenen Werte wurden über eine Fensterbreite von 50 Schaltzyklen gemittelt. Bei den Algorithmen handelt es sich zum einen um den in [112] vorgestellten R2R und zum anderen um den im Kontext dieser Arbeit vorgestellte BOA. Zu Vergleichszwecken sind Referenzwerte vom normalen Prellen zum jeweiligen Relais und Schaltvorgang mit einer gestrichelten Linie angegeben. Da für Relais A die Prelldauer im Einschalten schon im Normalfall bei etwa 0,1 ms liegt, sind hier keine Verbesserungen zu erwarten.

Zunächst werden die Ergebnisse vom R2R betrachtet: Ähnlich wie im originalen Paper konvergiert der Algorithmus zügig. Schon nach wenigen Schaltspielen liegt die Dauer

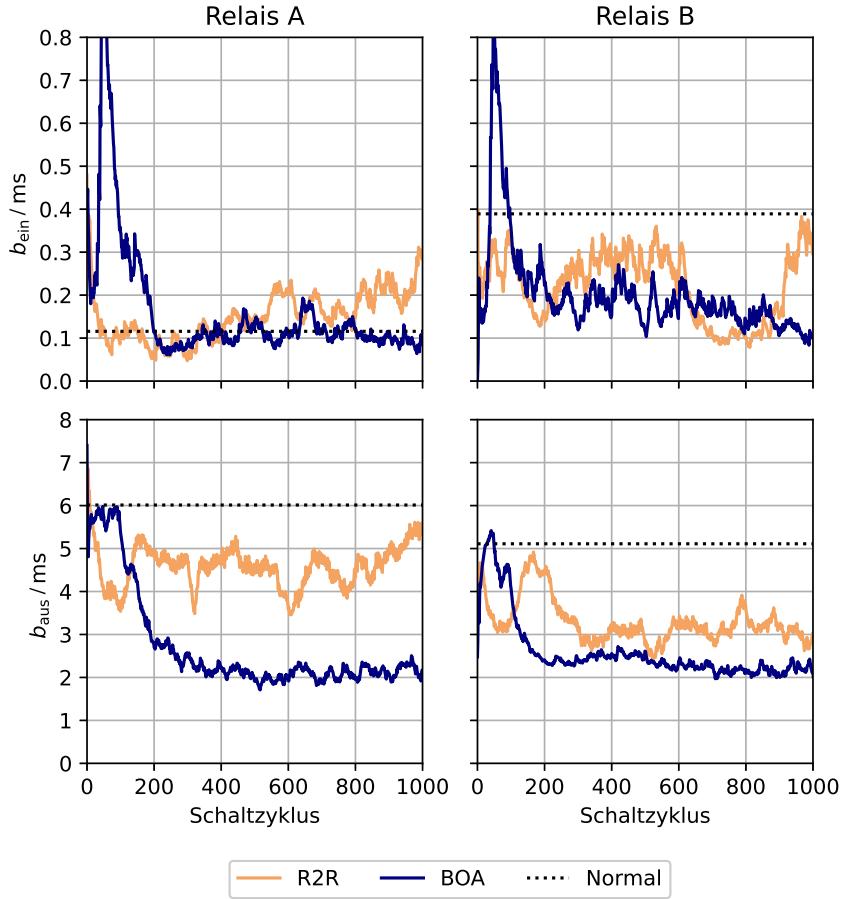


Abb. 4.14: Verlauf des Prellens für zwei Relais im Ein- und Ausschalten unter Anwendung von BOA [100]

des Prellens unterhalb des Referenzwertes (bzw. bei Relais A beim Einschalten in der Nähe). Es ist eine starke Varianz in den Kurven zu erkennen. Der Algorithmus ist nicht in der Lage, dauerhaft das Niveau zu halten. Das ist darauf zurückzuführen, dass der Typ von Algorithmus, eine Mustersuche, nicht für das Optimierungsproblem geeignet ist. Das Rauschen erschwert die Auswahl der Suchrichtung und Musterweite. Zusätzlich kann mit einem derartigen Verfahren nur ein lokales Optimum untersucht werden. Aus der Abb. 4.9 geht hervor, dass die Beschaffenheit komplexer ist. Trotzdem kann das Prellen schnell reduziert werden.

Die Ergebnisse durch BOA sind dem gegenüberzustellen. Auch hier kann erkannt werden, dass in allen Fällen eine schnelle Konvergenz unterhalb des Referenzwertes einsetzt. Verglichen mit dem R2R ist BOA etwas langsamer. Die etwa 400 Zyklen, die von BOA zur Erkundung benötigt werden, sind zu vernachlässigen, da die Lebensdauer von Relais typischerweise mindestens mehrere zehntausend Schaltzyklen beträgt. Beim Einschalten

sind ganz am Anfang zunächst niedrige Prelldauern auszumachen, die daraufhin ansteigen. Dies lässt sich mit der Rastersuche begründen: Die Partikel werden reihenweise evaluiert, weshalb phasenweise schon hier eine hohe Prellreduktion erreicht werden kann. Bei Relais A beim Einschalten bleibt das Prellen auf dem Niveau des Referenzwertes; in allen anderen Fällen kann es stets um deutlich mehr als 50 % verringert werden. Dabei weisen die Werte eine deutlich geringere Varianz auf als die vom R2R. Schlussendlich ist BOA stabiler als der R2R und weist eine bessere Performanz auf.

4.6.3 Quantitativ

Unter den Relais eines Typen liegt eine hohe Ungleichheit hinsichtlich der Eigenschaften, die das Prellen bestimmen, vor. Dies ist u.a. auf die Tatsache, dass ein Relaistyp oftmals von mehreren Herstellern gefertigt wird und Fertigungsabweichungen im Produktionsprozess, zurückzuführen. Deshalb ist es wichtig, auch eine quantitative Evaluation der Algorithmen durchzuführen und zu prüfen, ob diese auch bei einer Vielzahl von Relais zur Reduzierung des Prellens führen – ohne dass der Algorithmus an das individuelle Relais angepasst wird.

In Abb. 4.15 ist das Prellen von 48 Relais über die ersten 1000 Schaltzyklen mit dem BOA (linke Spalte) und R2R (rechte Spalte) abgebildet. Ein- und Ausschalten wurden in je einer Reihe visualisiert, sodass insgesamt vier Grafiken die Ergebnisse darstellen. Gestrichelt eingezeichnet ist das durchschnittliche Prellen, das mit normalem Schalten entsteht und als Referenz dient. Daneben sind Mittelwert und Konfidenzintervall des jeweiligen Verfahrens angegeben. Diese wurden über eine Fensterbreite von 50 Schaltzyklen von den 48 Relais gemittelt. Insgesamt fließen folglich die Daten von 2400 Schaltzyklen in jeden Zyklus ein. Dies ist für eine bessere Übersichtlichkeit notwendig. Präzisere Zahlen werden nachfolgend in Tab. 4.2 genannt.

Durch den R2R kann im Einschalten eine Reduktion des Prellens um 55,00 % erreicht werden und im Ausschalten eine von 18,84 %. Die Ergebnisse sind etwas geringer als in der Veröffentlichung von *Ramirez-Laboreo* [112], was u.a. auf den Wechsel des Relaistyps zurückzuführen ist, womit sich auch das grundsätzliche Optimierungspotenzial ändert. Zudem wurde das Verfahren dort lediglich an einem Relais evaluiert. Im Kontext dieser Untersuchungen ist die Probe größer, wodurch ein zuverlässigerer Erwartungswert gebildet werden kann. Die zuvor in Abschn. 4.4 beobachtete Varianz der Prelldauern, spiegelt sich bei dieser Analyse im Konfidenzintervall wider, welches im Vergleich zu BOA deutlich größer ist. Dafür ist der R2R jedoch schneller in der Reduktion des Prellens, weshalb in Abb. 4.15 zu Beginn keine Phase zu beobachten ist, in der das Prellen durch die Exploration außerordentlich hoch ist.

Beim BOA-Algorithmus ist hingegen für beide Schaltvorgänge eine klare Abnahme des Mittelwerts des Prellens zu beobachten. Dies ist mit der anfänglichen Rastersuche zu begründen. Nach etwa 250 Schaltzyklen konvergiert das Prellen gegen ca. 0,23 ms im Ein- und 2,13 ms im Ausschalten, was eine Verbesserung von 79,06 % (ein) bzw. 58,80 % (aus) darstellt. Damit wurden die Werte vom R2R deutlich unterboten. Zusätzlich liegt das

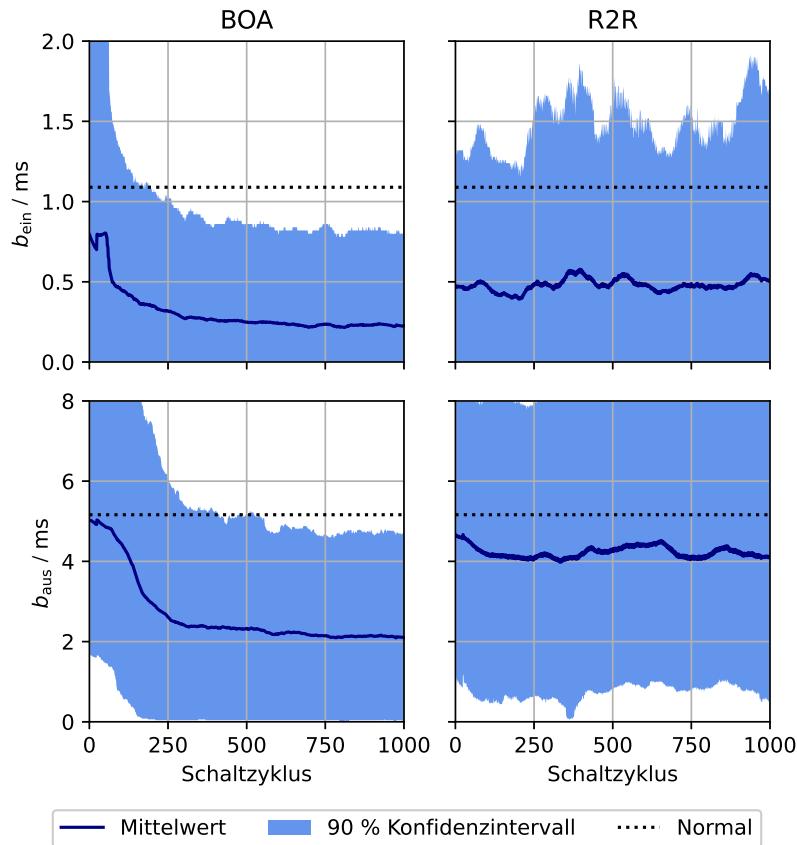


Abb. 4.15: Vergleich des Verlaufs des Prellens für 48 Relais im Ein- und Ausschalten beim R2R und BOA Algorithmus [100]

Konfidenzintervall ab ca. dem 250. Schaltzyklus stets unterhalb des Referenzwertes. Die Varianz des Prellens konnte mit BOA gesenkt werden.

Die Unterschiede bezüglich der Performanz der Methoden werden in Tab. 4.2 angegeben. Ferner wird die Standardabweichung (STD) präsentiert, die ein wichtiges Maß für die Reproduzierbarkeit des Prellens ist. Die Varianz des normalen Prellens kann durch die Anwendung von BOA in beiden Schaltfällen fast halbiert werden. Der R2R hat dagegen einen negativen Einfluss und steigert die Standardabweichung, was mit der hohen Varianz zu begründen ist. Schlussendlich kann durch BOA das Zeitintervall, in dem das Prellen auftritt, erfolgreich verringert werden.

Dies kann für weitere Forschungen von Relais in AC Anwendungen eine wichtige Erkenntnis sein. Denn hier stellt das Prellen ein Hindernis dar, wenn es darum geht, ein Schalten im Nullpunkt von Strom und / oder Spannung zu realisieren. Dies ist erwünscht, um die Degradierung von Relais zu verlangsamen, indem u.a. Kurzschlussströme und Lichtbögen verhindert bzw. minimiert werden sollen. Durch das Prellen kann kein genauer Zeitpunkt ausgemacht werden, zu dem die Kontakte beginnen zu leiten. Vielmehr ist es ein Zeitinter-

Tab. 4.2: Quantitativer Vergleich von BOA und R2R [100]

Methode	Einschalten		Ausschalten	
	Mittelwert / ms	STD / ms	Mittelwert / ms	STD / ms
Normal	1.089	0.500	5.163	2.170
R2R	0.490	0.639	4.190	2.433
BOA	0.228	0.297	2.127	1.396

vall, das durchsetzt ist von Ein- und Ausschaltvorgängen. Die Nullpunkte von Strom und Spannung sind deshalb praktisch nicht zu treffen. Auf diese Problematik weisen *Smugala et al.* in ihrer Arbeit [121] ebenfalls hin. Um die Konsequenzen zu verdeutlichen, wird beispielhaft eine 230 V Wechselspannung mit 50 Hz und eine Prelldauer des Relais von 1 ms angenommen. Damit ergibt sich unter der Annahme, dass der Nullpunkt in der Mitte des Prellintervalls liegt, eine maximale Spannung während des Prellens von $\pm 14,44$ V. Wenn das Prellen erst im Nullpunkt beginnt, dann beträgt die Spannung sogar $\pm 28,83$ V. Zum Vergleich: Mit einer Reduktion des Prellens durch BOA von 1 auf 0,25 ms ändern sich diese Werte auf 3,61 bzw. 7,22 V. Deshalb könnten durch weitere Forschung neue Potenziale hinsichtlich der Varianz des Schaltens erschlossen werden: Durch Einbezug der Startzeiten $t_{on; off}$ und der Dauern $d_{on; off}$ könnte die Schaltzeit präzise geschätzt werden, was wiederum das Nullpunktschalten verbessert.

4.6.4 Robustheit

Mit Hilfe der Untersuchungen in diesem Abschnitt wird nachgewiesen, dass BOA robust gegenüber äußeren Umwelteinflüssen ist. Als letztere kommen bei der Anwendung von Relais u.a. Temperatur, Degradierungszustand, Versorgungsspannung und ein plötzlicher Relaiswechsel infrage. Der Wechsel wird betrachtet, da in der Applikation davon ausgegangen werden kann, dass ein Relais getauscht wird, ohne BOA neu zu initialisieren.

Die Temperatur und die Versorgungsspannung wurden im Rahmen von Experimenten während des Betriebs von BOA geändert. Dabei wurde beobachtet, dass bei Änderungen der Temperatur die Parameter durch BOA nachgeführt werden können und somit weiterhin das Prellen reduziert werden kann. Für eine Variation der Versorgungsspannung sind die Relais ausgelegt. Deshalb ist der Einfluss so gering, dass die Konsequenzen zu vernachlässigen sind. Um das Verhalten von BOA bei Änderungen der Umwelteinflüsse exemplarisch darzustellen, wird im Nachfolgenden eine plötzliche Änderung des Optimierungsproblems – also ein Wechsel des Relais – vorgestellt.

In Abb. 4.16 sind die Partikel über verschiedene Iterationen dargestellt, mit einer Heatmap der Prellreduktion als Hintergrund. Diese unterscheidet sich in Farbgebung und Beschaffenheit für die Iteration 20, da nach dieser der Wechsel des Relais vollzogen wurde. Bei Iteration 20 hat BOA beim Ein- und Ausschalten ein lokales Optimum gefunden. Durch den Wechsel ändert sich das Optimierungsproblem sprunghaft in Iteration 21. Für das Einschalten ist zu erkennen, dass die Partikel nicht mehr in der Nähe eines Optimums

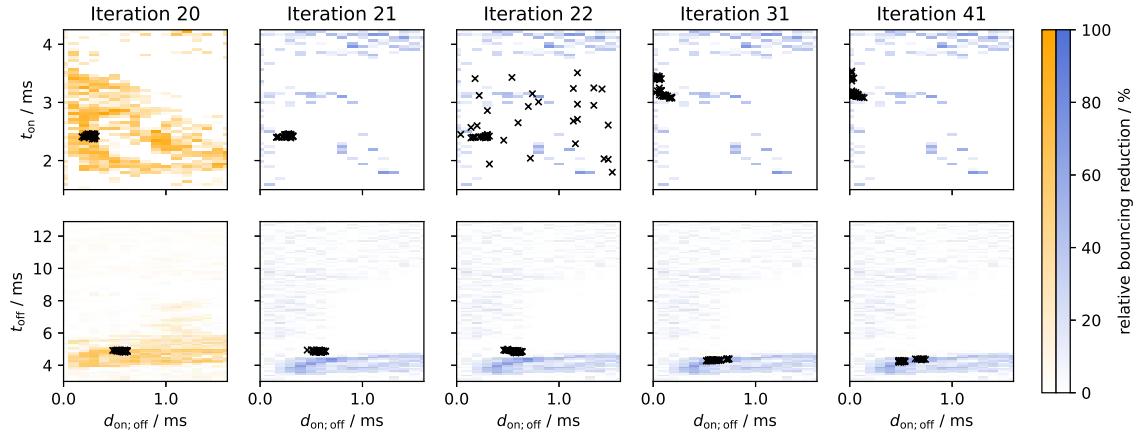


Abb. 4.16: Exemplarische Visualisierung der Funktionsweise von BOA beim unangekündigten Tausch des Relais [100]

liegen. Durch BOA muss dementsprechend ein verstärktes Explorieren vorgenommen werden. Beim Ausschalten liegen die Partikel hingegen in der Nähe eines Bereichs mit starker Prellreduktion und führen weiterhin zu einer geringen Prellreduktion.

In Iteration 22 sind die Anpassungen der Partikel durch BOA deutlich zu erkennen. Beim Einschalten wurden viele Partikel zufällig im Suchraum verteilt, da durch die vorherigen Partikel keine Reduzierung des Prellens erzielt werden konnte (vgl. Schritt **VI** von BOA). Für das Ausschalten ist lediglich ein leichter Drift der Partikel zu beobachten; sie rücken näher an das Optimum heran. Zu begründen ist dies mit dem Entfernen der Partikel mit schlechterer Performanz (vgl. Schritt **IV**). Diese sind weiter vom Optimum entfernt, weshalb die neuen Partikel näher des Optimums generiert werden.

Darauf wird Iteration 31, also ein Sprung von neun Iterationen, dargestellt. Für das Ein- und Ausschalten ist zu sehen, dass die Partikel wieder konzentriert auftreten. Zehn Iterationen später sind diese Konzentrationen ebenfalls zu beobachten und BOA ist konvergiert. Dass dies auch für die Prelldauer gilt, wird in der nächsten Abb. 4.17 herausgearbeitet.

In Abb. 4.17 ist das Prellen im Ein- und Ausschalten über 1000 Schaltzyklen aufgetragen. Zur deutlicheren Visualisierung wurden die Werte über eine Fensterbreite von 50 Schaltzyklen geglättet. Als Referenz ist durch eine gestrichelte Gerade das Prellen der Relais unter normalen Bedingungen, also ohne BOA, gegeben. Zum 1008ten Schaltzyklus wird das Relais gewechselt. Die Daten korrespondieren zu Abb. 4.16, dementsprechend ist die Farbwahl angepasst. Die zuvor beschriebenen und erklärten Beobachtungen zur Bewegung der Partikel können mit diesen Graphen zusätzlich verifiziert werden: Beim Einschalten wird durch den Relaiswechsel keine Verbesserung des Prellens erzielt, weshalb zufällig exploriert wird. Für das Ausschalten wird trotz Wechsel eine Verbesserung erreicht. Demzufolge überwiegt das ausbeutende Verhalten und die Partikel konvergieren langsam. Diese Konvergenz ist für beide Schaltvorgänge etwa ab dem 1450 Schaltzyklus zu erkennen, was der 31. Iteration in der vorherigen Abb. 4.16 entspricht. Entsprechend der Konvergenz

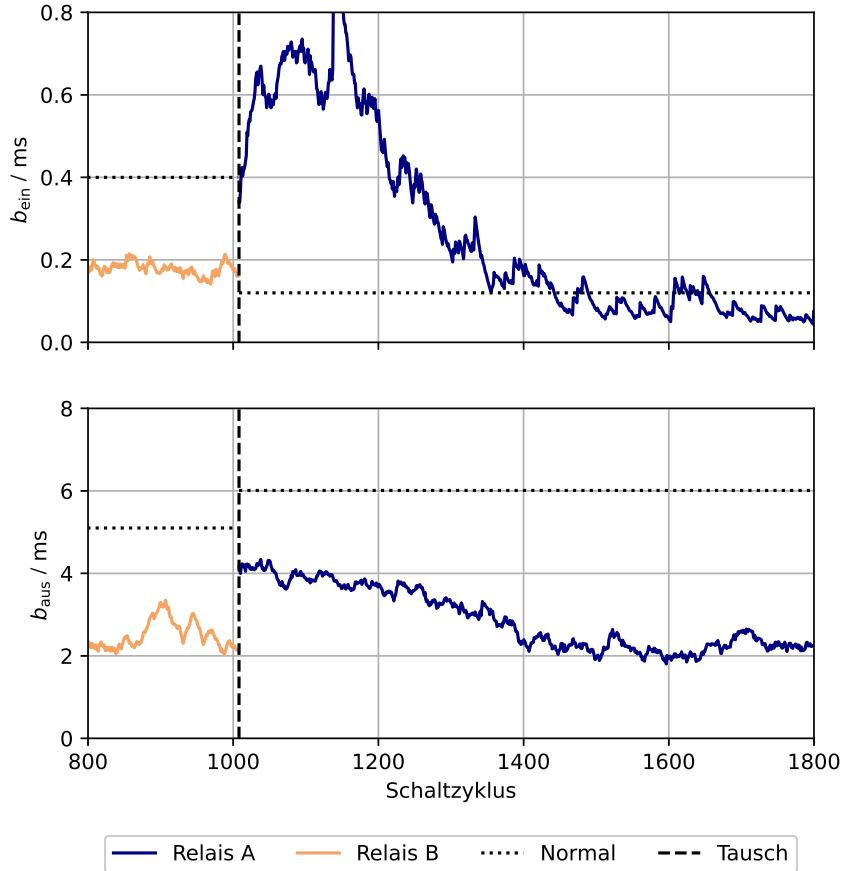


Abb. 4.17: Verlauf des Prellens beim unangekündigten Tausch des Relais [100]

sind im Weiteren keine Verbesserungen mehr auszumachen, was sich im Stagnieren der Positionen der Partikel widerspiegelt.

Im Kontext dieser Untersuchung konnte nachgewiesen werden, dass BOA robust gegenüber Umwelteinflüssen ist und Exploration und Ausbeute vorteilhaft kombiniert wurden. Im Falle von Optimierungsproblemen, bei denen BOA nicht konvergiert, könnte BOA im fünften Schritt um ein Abbruchkriterium erweitert werden, sodass nach einer definierten Exploration nur noch normal geschaltet wird.

4.6.5 Schaltzeit

Schlussendlich soll der Einfluss von BOA auf die Dauer der Ein- und Ausschaltvorgänge untersucht werden. Die Schaltzeiten von Relais stellen in Systemen oftmals ein Problem dar: Schon wenige Millisekunden Verzögerung können für die Realisierung einer Anwendung ausschlaggebend sein. Beispielhaft vorgestellt werden kann dies an einem Förderband, bei dem Objekte beim Durchlaufen einer Lichtschranke durch einen Aktor vom Förderband

geschoben werden sollen. Wenn dieser Aktor mit einem Relais geschaltet wird, dann kommt es auf geringe Reaktionszeiten an. Deswegen wurden bereits diverse Bestrebungen unternommen, um die Schaltzeit zu reduzieren. In [122] zeigen *Rana et al.* beispielsweise, dass die Schaltzeit durch Berücksichtigung der Schwingung des Ankers in Ruhelage reduziert werden kann. Die Untersuchung beschränkt sich allerdings auf nanoelektromechanische Relais.

Vorab ist eine Betrachtung der Schaltzeit notwendig, welche durch drei Faktoren bestimmt wird: Der erste Faktor ist die Trägheit der Spule, denn der Strom benötigt Zeit, um anzusteigen. Damit geht einher, dass auch die Kraft, die durch die Spule auf den Anker ausgeübt wird, erst nach einer gewissen Zeit ausreicht, um den Anker zu bewegen. Sobald der Anker in Bewegung ist, kommt die Trägheit der mechanischen Bauteile als zweiter Faktor hinzu. Diese müssen zum Schließen der Kontakte einen gewissen Weg zurücklegen. Schlussendlich ist der dritte Faktor durch das Prellen gegeben. Hier besteht zwar teilweise schon Kontakt, aber eine zuverlässige Kontaktierung ist (noch) nicht gegeben. Im Kontext dieser Arbeit ändern sich alle drei Faktoren durch die Invertierung der Versorgungsspannung für ein kurzes Zeitintervall. Jedoch sind bei den ersten beiden Zuwächsen in den Zeittauern zu erwarten, währenddessen eine Zeiteinsparung lediglich bei dem dritten Faktor erreicht wird. Daher wird im Folgenden nachgewiesen, dass BOA keinen negativen Einfluss auf die Schaltzeit hat.

In Abb. 4.18 sind die Ergebnisse bezüglich Ein- und Ausschaltdauern unter Verwendung von BOA dargestellt. Insgesamt wurden 1000 Schaltzyklen mit 16 Relais durchgeführt und ausgewertet, sodass der durchschnittliche Mittelwert und das durchschnittliche 90 % Konfidenzintervall der Dauern angegeben werden. Diese Kenngrößen wurden mittels einer Faltung mit einer Fensterbreite von 50 Schaltzyklen bestimmt. Als Referenz ist der durchschnittliche Mittelwert der normalen Schaltdauer in den Diagrammen angegeben.

Analog zu Abs. 4.6.2 und 4.6.3 ist für beide Schaltvorgänge zu beobachten, dass BOA etwa 400 Schaltzyklen bis zur Konvergenz benötigt. Mittelwert sowie Konfidenzintervall der letzten 600 Schaltzyklen sind dementsprechend nahezu konstant. Beim Einschalten ist zu erkennen, dass der Mittelwert durch BOA in keinem relevanten Umfang reduziert wird, obwohl in den vorangegangenen Untersuchungen nachgewiesen werden konnte, dass das Prellen deutlich abgenommen hat. Dies ist damit zu begründen, dass durch die modifizierte Steuerspannung mehr Zeit für das Anziehen des Ankers benötigt wird. Da die Relais durch die Hersteller dahin gehend optimiert wurden, dass sie beim Einschalten weniger prellen als beim Ausschalten, ist das Zeitsparpotenzial hier nicht so groß, dass der gesamte Schaltvorgang wesentlich beschleunigt werden kann. Beim Ausschalten ist das Potenzial jedoch größer: Die Schaltdauer wird um etwa 2,5 ms reduziert, was der Prelldauerreduktion um 3 ms abzüglich der Modifikation des Steuersignals entspricht. Hervorzuheben ist, dass sogar das 90 % Konfidenzintervall unterhalb der normalen Schaltzeit liegt. Beim Ausschalten kann durch BOA somit eine schnellere Schaltzeit erreicht werden.

Zusammenfassend lässt sich festhalten, dass durch BOA die Schaltzeit nicht verlängert wird und beim Ausschalten sogar verkürzt werden kann. Dies ist als Erfolg zu werten, da der Zweck von BOA eine Minimierung des Prellens ist, für die, gegenüber einem normalen

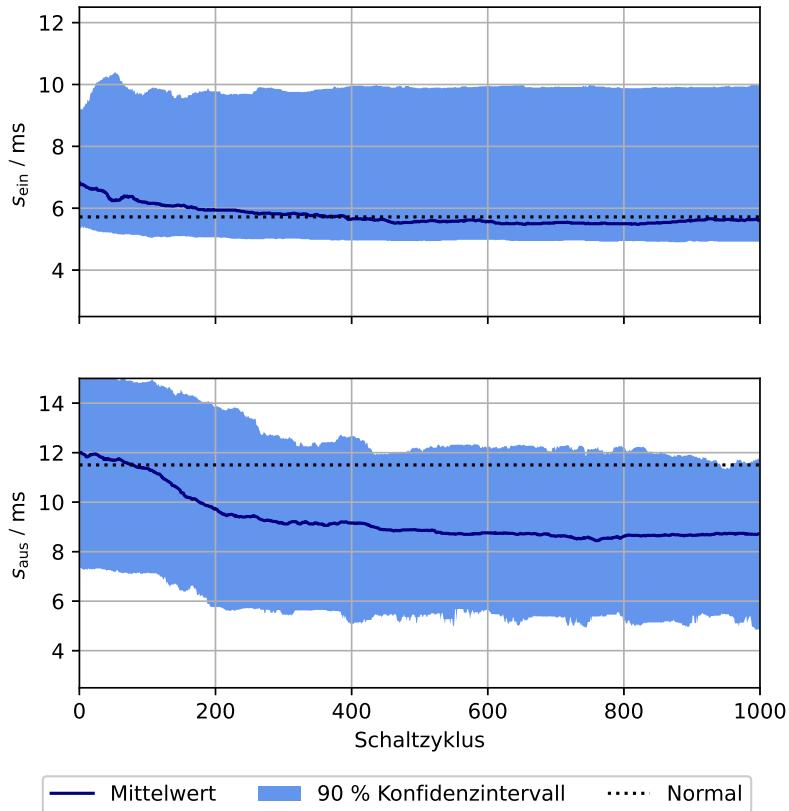


Abb. 4.18: Mittlere Änderung der Schaltdauer beim Ein- und Ausschalten bei einer Stichprobe von 16 Relais [100]

Schaltvorgang, zusätzliche Aktionen erforderlich sind. Im Rahmen weiterer Forschung könnte BOA bei anderen Relaistypen eventuell sogar zur Schaltzeitverkürzung genutzt werden.

4.7 Fazit

Im vorangegangenen Kapitel wurde der BOA beschrieben, mit dem durch Änderung des Steuersignals das Prellen eines elektromechanischen Relais reduziert werden kann. Diese Änderung wird über zwei Parameter definiert, die mit einer Variante einer Partikelschwarmoptimierung während des Betriebs des Relais angepasst werden. Die wesentlichen gesammelten Erkenntnisse werden nachfolgend zusammengefasst:

- Statistische Analyse des Prellens von Relais

Es wurde festgestellt, dass ein erhebliches Rauschen dem Prellen unterliegt und deshalb vielfaches Schalten zu konstanten Umgebungsbedingungen notwendig ist,

um einen Erwartungswert zur Prelldauer zu bilden. Zudem wurden diverse Parametrierungen des modifizierten Steuersignals geprüft. Dadurch konnte gezeigt werden, dass multiple lokale Optima bestehen.

- **Bouncing Optimization Algorithm**

Mit BOA wurde ein maßgeschneideter Algorithmus vorgestellt. Dieser wurde aus der klassischen Partikelschwarmoptimierung abgeleitet, wobei detailliert auf die einzelnen Änderungen und deren Zweck eingegangen wurde. BOA konvergiert trotz Rauschen und mehrerer Optima sicher und reduziert das Prellen. Dabei ist der Rechenaufwand sehr gering, sodass BOA auf einem Mikrocontroller implementiert wurde.

- **Nachweis der Funktionalität**

Sowohl die Funktionsweise von BOA, als auch dessen Überlegenheit gegenüber dem aktuellen Stand der Technik, wurde in mehreren Experimenten nachgewiesen. Bei 48 Relais kann das Prellen im Durchschnitt um 79 % beim Ein- und um 59 % beim Ausschalten reduziert werden. Gegenüber Änderungen der Umgebungseinflüsse ist BOA robust. Dies konnte durch spontanes Wechseln eines Relais gezeigt werden, für das ohne Reset von BOA ein anderes Optimum gefunden wurde. Schlussendlich konnte auch ein positiver Einfluss von BOA auf die Schaltzeit präsentiert werden.

- **Vorstellung eines simplen Hardwareaufbaus**

Die verwendete Hardware wurde detailliert beschrieben, wobei darauf geachtet wurde, dass eine Realisierung im Produkt möglich ist. Diese Absicht wird unterstützt von der Tatsache, dass die gesamte Evaluation mit diesem Aufbau durchgeführt wurde. Deshalb ist BOA dem Stand der Technik nicht nur hinsichtlich der Performanz überlegen, sondern geht den Schritt von Laborhardware zu Produkthardware.

Für weitergehende Forschung ist es empfehlenswert, die Ergebnisse auf andere Relaistypen und / oder Schütze zu übertragen. Des Weiteren ist eine Anwendung auf andere elektromechanische Aktoren, wie Ventile, vielversprechend. Zusätzlich sollten Lebensdauerversuche zu den Effekten des reduzierten Prellens auf die Degradierung unternommen werden. Hierbei liegt jedoch eine Schwierigkeit im notwendigen Zeitaufwand, der mit mindestens mehreren Monaten veranschlagt werden muss.

Die Erkenntnisse zur Thematik Prellreduzierung sind in [100] publiziert. Darüber hinaus wurde ein Patent [123] angemeldet.

5 Abrasive Kontaktselbstheilung

Das letzte Kapitel dieser Arbeit baut auf Erkenntnissen aus den Dauertests auf: Dabei zeigte sich, dass viele Relais ausfallen, weil sich z.B. durch Migration Materialanhäufungen an den Kontaktobertächen ausbilden. Weiterhin ist zu beobachten, dass zwischen dem ersten und siebten¹ fehlerhaften Schaltvorgang oftmals mehrere hundert Schaltzyklen liegen. Dementsprechend ist die Wahrscheinlichkeit hoch, dass ein Relais durch erneutes Schalten wieder kurzfristig nutzbar gemacht werden kann.

Im Folgenden wird das Verfahren „Contact Self-healing via Abrasion“ (dt. Kontaktselbstheilung via Abrasion) (CSA) vorgestellt, mit dem die Anlagenverfügbarkeit gesteigert werden kann. Durch eine spezielle Ansteuerung der Relaisspule wird dabei eine Reibung zwischen den Kontaktpaaren angeregt. Auf diesem Weg können bspw. Materialanhäufungen oder isolierende Schichten abgerieben und verhakte oder verschweißte Kontakte gelöst werden. So wird die Möglichkeit eröffnet, dass Relais nach einer ersten Fehlfunktion oder einem erkannten auffälligen Zustand (vgl. MAUD) kurzfristig weiter betrieben werden, um einen Ausfall der Anlage zu verhindern. Das Verfahren CSA ist als Patent angemeldet [124].

Zunächst wird der Forschungsstand zu dieser Thematik dargelegt, anschließend wird das Potenzial für die Selbstheilung durch eine Statistik aus den Dauertests aufgezeigt. Daraufhin werden der verwendete Testaufbau und das Verfahren CSA beschrieben. Nachfolgend wird CSA hinsichtlich drei Aspekten evaluiert: die Funktionalität, der Einfluss von veränderbaren Parametern und die Auswirkungen auf die Kontaktobertächen.

5.1 Forschungsstand

In den Patenten [125] und [126] wird ein System beschrieben, das einen fehlerhaften Schaltzustand erkennt und durch erneutes Schalten versucht zu korrigieren. Diese Funktionalität soll durch CSA ebenfalls abgebildet werden. In [127], [128], [129] und [130] werden diese Verfahren dahin gehend verbessert, dass das Relais durch eine spezielle Ansteuerung frei gerüttelt werden soll. Dazu wird eine Pulsweitenmodulation oder Wechselspannungs-

¹Die Relais wurden im Kontext dieser Arbeit bis zum siebten fehlerhaften Schaltvorgang degradiert, da sie bei dieser Anzahl nach allen Herstellern als ausgefallen gelten.

signale genutzt. Auch diese Idee ist in CSA inkludiert. Bei CSA wird im Vergleich zu den zuvor aufgeführten Verfahren jedoch eine deutlich höhere Erregerfrequenz genutzt, weshalb die Kontakte nicht nur durchgerüttelt, sondern gezielt aneinander gerieben werden. Somit sollen Kontakte nicht nur voneinander gelöst, sondern auch hinsichtlich ihrer Oberflächenstruktur verbessert werden.

Die Kernidee des Ausnutzens von abrasivem Verschleiß ist im Kontext der Vergütung von Oberflächen bekannt und findet seit Jahrzehnten Anwendung in Produktionsprozessen [131]. Beim Gleitschleifen werden jedoch spezielle Schleif- und Polierkörper sowie Chemikalien eingesetzt, die bei einem Relais nicht zur Verfügung stehen. *Yamaguchi* und *Shinmura* zeigen in [132], dass die Oberfläche einer Stahlplatte mit Eisenpartikeln geglättet werden kann. Dazu rotierten sie eine Stahlplatte, wobei Eisenpartikel mit einem Magneten an einer fixen Position gehalten wurden. Mit diesem Prozedere wurde Material an der Oberfläche abgetragen. Der Bezug zu dieser Arbeit kann wie folgt hergestellt werden: Bei den Kontakten handelt es sich ebenfalls um eine Metalllegierung. Die Eisenpartikel sind in diesem Kontext die Verunreinigungen. Die für das Schleifen notwendige Kraft wird durch Anker und Spule generiert.

5.2 Ausfallverhalten von Relais

Um den Nutzen und das Potenzial des Verfahrens zu beleuchten, wird zunächst betrachtet, wie Relais sich beim Ausfall verhalten. Dazu werden Daten aus den Alterungsversuchen herangezogen: Die Relais wurden bis zum siebten defekten Schaltvorgang gealtert. Deshalb ist auszumachen, wie viele Schaltzyklen zwischen dem ersten und siebten Defekt liegen. Bei dieser Analyse wird ein fehlerhafter Schaltzustand dementsprechend über ein erneutes Schalten korrigiert, was durch das entwickelte Verfahren ebenfalls abgebildet wird.

In Abb. 5.1 ist der Anteil der Relais angegeben, die zwischen dem ersten und siebten Defekt mindestens die entsprechende Anzahl an Schaltzyklen aufweisen. Die Abszisse ist logarithmisch skaliert, da sich der Wertebereich von 7 bis etwa 1.000.000 Schaltzyklen aufspreizt und somit eine bessere Visualisierung möglich ist. Aus der Grafik lassen sich folgende Schlüsse ziehen:

- Etwas mehr als 80 % der Relais weisen 10 oder mehr Schaltzyklen zwischen dem ersten und siebten Defekt auf: Daraus folgt, dass fast 20 % der Relais innerhalb von 10 oder weniger Schaltzyklen ausfallen, wodurch kein kurzzeitiger Weiterbetrieb mittels erneuten Ansteuern möglich ist.
- Bei ca. 65 % der Relais liegen die Defekte in einer Intervallbreite von mindestens 100 Schaltzyklen: Folglich liegt hier der Erwartungswert für eine erfolgreiche Korrektur des Schaltzustandes durch ein erneutes Schalten bei mindestens 93 %.
- Bei 50 % der Relais liegen mindestens 1.000 Schaltzyklen zwischen den Defekten: Der Erwartungswert erhöht sich bei diesen damit auf 99,3 %.

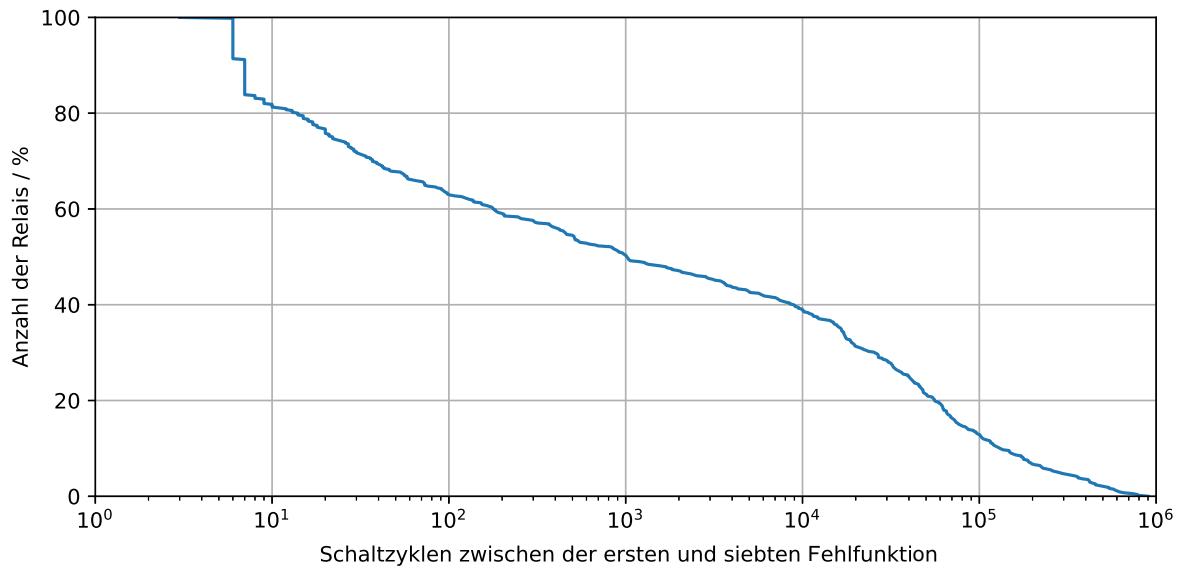


Abb. 5.1: Anteil der Relais aufgetragen über der Anzahl der Schaltzyklen zwischen dem ersten und siebten defekten Schaltvorgang

- Dieser Wert kann signifikant gesteigert werden, wenn die 40 % der Relais betrachtet werden, bei denen die Differenz zwischen den Defekten am größten ist: Hier liegen mindestens 10.000 Schaltzyklen zwischen dem ersten und siebten Defekt, sodass der Erwartungswert für eine erfolgreiche Korrektur durch erneutes Schalten bei 99,93 % liegt.

Da etwa 550 Relais für diese Analyse ausgewertet wurden, kann von einer signifikanten Stichprobe ausgegangen und folgender Schluss gezogen werden: Ein erneutes Ansteuern eines Relais im Fehlerfall kann bei einer Vielzahl der Relais zu einer Korrektur führen. Auch wenn diese Korrektur nur kurzfristig ist, könnte mit CSA eine Steigerung der Anlagenverfügbarkeit erreicht werden.

5.3 Messaufbau

Für die Umsetzung des Verfahrens stellen sich die gleichen Anforderungen an die Hardware wie bei MAUD und BOA. Ausschließlich für die Evaluation wird zusätzliche Hardware benötigt: ein Beschleunigungssensor zur Erfassung von Schwingungen bzw. des Reibens. Der verwendete Aufbau für die Untersuchungen wird in Abb. 5.2 dargestellt. Von der Darstellung weiterer peripherer Bauteile, bspw. zur Filterung der Signale, wurde zugunsten der Übersichtlichkeit abgesehen.

Im Zentrum der Darstellung steht das Relais, um das die restliche Schaltung angeordnet ist. Diese kann in drei Gruppen untergliedert werden:

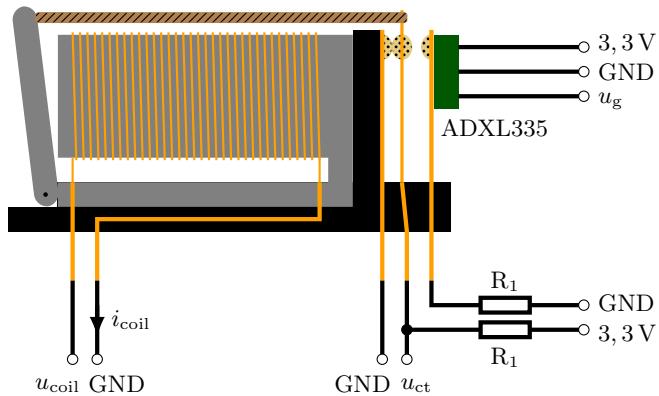


Abb. 5.2: Skizze des verwendeten Messaufbaus

- Relaisspule:

Die Relaisspule wird an das Massepotenzial (GND) angeschlossen und die Spulenspannung u_{coil} wird angelegt. Die Versorgungsspannung wird durch die Steuerung bestimmt und kann 0 oder 24 V annehmen. Als Eingang für die Steuerung dient der Spulenstrom i_{coil} .

- Kontakte:

Der Zustand der Kontakte wird mit einer Schaltung analog zum Aufbau aus Abschn. 4.2 überwacht. Auch hier ist eine alternative Umsetzung, die eine galvanische Trennung aufweist, entsprechend dem Patent [91] möglich. Die beiden Widerstände R_1 werden identisch dimensioniert, sodass ein symmetrischer Spannungsteiler entsteht und das Kontaktsignal u_{ct} für jeden der drei Schaltzustände einen anderen Wert annimmt. Das Signal ist wichtig, um einerseits einen defekten Schaltvorgang zu erkennen und andererseits die Kontaktierung während der Anwendung von CSA zu überwachen.

- Beschleunigungssensor:

Der Beschleunigungssensor dient lediglich der Evaluation und wird für eine Umsetzung nicht benötigt. Er ist direkt an den NO-Kontakt des Relais angeklebt. Somit lässt das Signal u_g direkte Rückschlüsse auf die Erschütterungen / Bewegungen des Kontaktpaars COM und NO zu. Als Sensor wurde ein ADXL335 von Analog Devices genutzt, der über einen 0,25 mm starken Kupferlackdraht angebunden wurde. So wird der mechanische Einfluss auf das System minimiert. Aufgrund der Schaltung liegt die Grenzfrequenz des Beschleunigungssignals bei 1.600 Hz.

Die Signale werden doppelt erfasst: Mit einem Cortex F4 Mikrocontroller von STM, auf dem die Steuerung implementiert ist und mit einem Oszilloskop, das zur Visualisierung der Signale genutzt wird. Die praktische Realisierung des Aufbaus ist Abb. 5.3 zu entnehmen. Durch CSA werden starke Vibrationen hervorgerufen, weshalb der Kupferlackdraht zusätzlich an einem Draht befestigt werden musste.

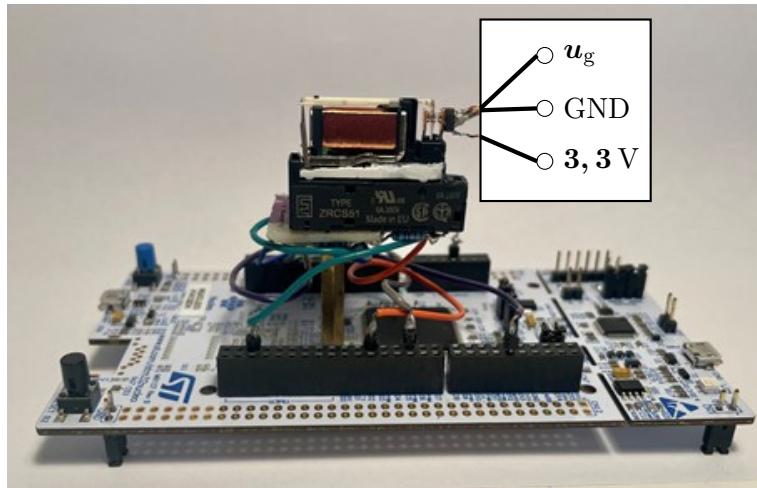


Abb. 5.3: Foto des verwendeten Messaufbaus

5.4 Contact Self-healing via Abrasion

Nachfolgend wird CSA anhand von Skizzen erklärt. Ziel ist es, zwischen den Kontakten eines Relais eine Reibung hervorzurufen. Dies wird durch den Anker erreicht, der im Bereich des Überhubs bewegt wird, sodass dessen Bewegung auf die Kontakte übertragen wird und diese aneinander reiben. Dabei sind die Kontakte ständig leitfähig, weshalb das Verfahren auch im Betrieb des Relais angewendet werden kann.

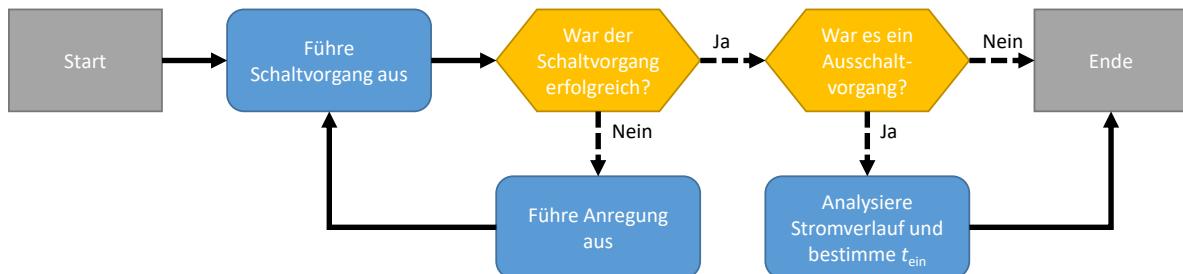


Abb. 5.4: Ablaufdiagramm für eine Implementierung von CSA

Das Flussdiagramm für eine mögliche Umsetzung ist in Abb. 5.4 dargestellt. Der Start ist das Kommando für einen Schaltvorgang – ungeachtet, ob es sich um einen Ein- oder Ausschaltvorgang handelt. Als Erstes ist der Schaltvorgang auszuführen, danach muss geprüft werden, inwiefern dieser erfolgreich war. Wenn die Kontakte nach einem Einschaltvorgang nicht oder nach einem Ausschaltvorgang weiterhin kontaktiert sind, dann war der Schaltvorgang nicht erfolgreich und die Anregung für die Reibung muss ausgeführt werden. Eine Beschreibung, wie diese stattfindet, wird nachfolgend gegeben. Für das Verfahren sind unterschiedliche Ausgestaltungen der Anregung denkbar – von einer einzelnen Anregung hin bis zu mehreren Anregungen mit jeweils unterschiedlichen Spezifikationen. Nach der

Anregung muss der Schaltvorgang wiederholt werden, um zu prüfen, ob das Relais „geheilt“ werden konnte. Wenn dies nicht der Fall ist, dann kann das Prozedere wiederholt werden. Andernfalls kann der nächste Schritt im Diagramm erreicht werden: Hier wird geprüft, um welche Art von Schaltvorgang es sich handelt. Dies ist notwendig, da der für die Anregung notwendige Parameter t_{ein} durch die Analyse von Ausschaltvorgängen bestimmt werden muss (vgl. Abb. 5.6). Nach der Analyse ist der Schaltvorgang abgeschlossen.

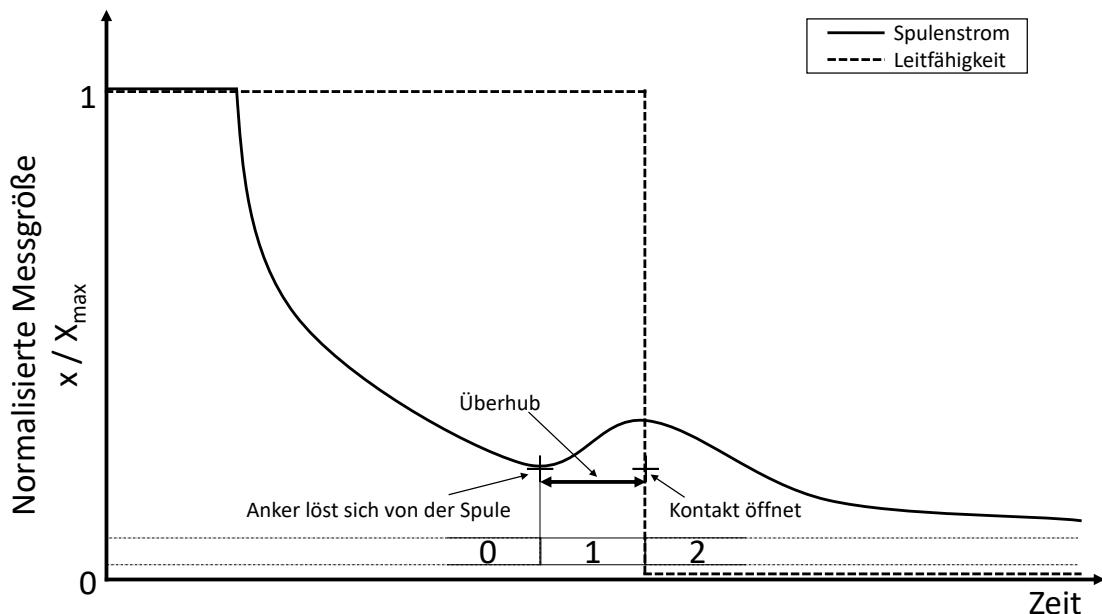


Abb. 5.5: Messgrößen während eines normalen Abschaltvorgangs [124]

Für die Beschreibung der Funktionsweise der Anregung werden in Abb. 5.5 zunächst die Messgrößen eines normalen Abschaltvorgangs skizziert. Kurz oberhalb der Abszisse sind drei Zustände des Relais eingezeichnet und beschriftet:

- 0: Die Kontakte sind geschlossen und der Anker ist an die Spule herangezogen.
 - 1: Der Überhub wird abgebaut, d.h. die Kontakte sind geschlossen, aber zwischen Anker und Spule ist ein Luftspalt.
 - 2: Die Kontakte und der Anker sind geöffnet.

Die Funktionsweise von CSA kann anhand von Abb. 5.6 erklärt werden: Zunächst wird das Relais abgeschaltet und der Strom fällt ab – wie bei einem normalen Abschaltvorgang. Nachdem der Anker sich vom Spulenkern löst – also Zustand 1 erreicht wird – wird die Versorgungsspannung wieder eingeschaltet. Der Zeitpunkt dafür wird so gewählt, dass Zustand 2 nicht erreicht wird. Stattdessen wird der Anker an die Spule herangezogen und Zustand 0 wieder erreicht. Die Leitfähigkeit bleibt deshalb konstant bei 1. Schlussendlich

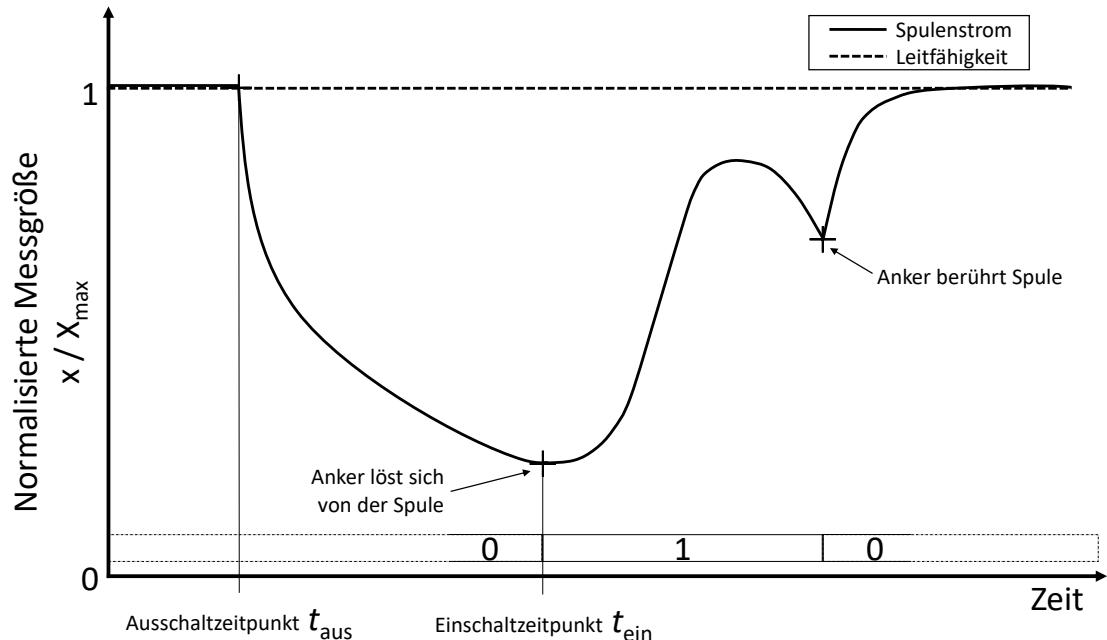


Abb. 5.6: Messgrößen während des Reibens [124]

wird durch CSA eine Bewegung des Ankers ausgelöst, die sich auf die Kontakte überträgt, wodurch wiederum eine Reibung der Kontaktobерflächen bedingt wird.

Für eine Implementierung von CSA sind lediglich zwei Zeitpunkte relevant: Erstens der Zeitpunkt t_{aus} , zu dem die Versorgungsspannung ausgeschaltet und zweitens der Zeitpunkt t_{ein} , zu dem diese wieder eingeschaltet wird. t_{aus} kann dabei willkürlich gewählt werden. Die einzige Bedingung ist, dass das Relais eingeschaltet ist. Die Wahl von t_{ein} ist abhängig vom Zeitpunkt, zu dem sich der Anker von der Spule löst. Letzterer kann durch die Analyse von normalen Schaltvorgängen bestimmt werden, da er als ein lokales Minimum im Spulenstrom auszumachen ist. Es ist jedoch nicht notwendig, diesen genau zu treffen. Das Reiben kann auch ausgelöst werden, wenn t_{ein} um einen Offset verschoben wird.

5.5 Evaluation

Im Rahmen der Evaluation von CSA werden drei Fragestellungen untersucht:

1. Welche Beschleunigungen werden an den Kontakten hervorgerufen?
2. Welchen Einfluss hat die Wahl des Offsets auf die Beschleunigungen?
3. Welche Veränderungen sind an den Kontakten auszumachen?

Die Fragen werden nachfolgend in drei Abschnitten erörtert.

5.5.1 Messgrößen bei Reiben

Im Kontext dieser Untersuchungen wird die Funktionalität von CSA nachgewiesen und gemessen, welche Beschleunigungen an den Kontakten auftreten. Dazu wurde CSA entsprechend der Beschreibung in Abschn. 5.4 mit einem Offset von null auf dem in Abb. 5.3 darstellten Messaufbau implementiert. In Abb. 5.7 sind jeweils drei Messgrößen für einen Ein- und Ausschaltvorgang sowie Reiben dargestellt. Die Zeitreihen wurden dabei so verschoben, dass der Ausschaltzeitpunkt vom Reiben und der Ausschaltvorgang übereinstimmen; genauso wie der Einschaltzeitpunkt vom Reiben und dem Einschaltvorgang. Es ist ein Zeitfenster von insgesamt 40 ms visualisiert, indem zwei Reibungsanregungen vollständig zu sehen sind.

Beim Reiben liegt die Kontaktspannung u_{ct} konstant bei 1,65 V. Daraus folgt, dass die Kontakte COM und NO stets geschlossen sind. Im Spulenstrom sind die Aufschläge des Ankers als lokale Minima bei 12,5 und 27,5 ms zu erkennen; die Form entspricht einer Mischung aus Ein- und Ausschaltvorgang: Bis 7,5 ms fällt der Spulenstrom wie beim Ausschaltvorgang ab, danach ähnelt er bis zum Ankeraufschlag bei 12,5 ms einer Superposition beider Schaltvorgänge. Nach dem Aufschlag entspricht der Strom dem Einschaltvorgang, bis das Prozedere bei 18 ms abermals durchgeführt wird.

Die beobachtete Mischung beider Schaltvorgänge im Reiben ergibt sich auch bei Betrachtung der Beschleunigungen \mathbf{a} . Das Einschalten ruft maximale Beschleunigungswerte von fast 50 m/s^2 hervor, die durch den Aufschlag des Ankers mit hoher Geschwindigkeit auf den Spulenkern bedingt werden. Beim Einschalten ist der Zeitraum, in dem Beschleunigungen zu erkennen sind, kürzer als beim Ausschalten. Weiterhin ist die Beschleunigung stärker gedämpft. Dies ist darauf zurückzuführen, dass der Anker im eingeschalteten Zustand an der Spule anliegt und im ausgeschalteten nur durch eine Feder in die Ruhelage gebracht wird und keinen Anschlagspunkt hat, was mit einer Bewegungsfreiheit einhergeht. Daraus folgt auch, dass geringere maximale Beschleunigungswerte von etwa 40 m/s^2 beim Ausschalten hervorgerufen werden. Es ist davon auszugehen, dass auch bei normalen Einschaltvorgängen ein Reiben der Kontakte ausgelöst wird. Dies ist zeitlich jedoch begrenzt, weshalb keine signifikante Wirkung wie bei CSA hervorgerufen wird. Für CSA ist eine Superposition der Beschleunigungsverläufe von Ein- und Ausschaltvorgang für eine Anregung zu erkennen. Das ist zu erwarten, weil der Anker sich bei einer Anregung von der Spule löst und dann wieder aufschlägt – was jeweils einem unvollständigen Aus- bzw. Einschaltvorgang entspricht.

In Abb. 5.7 sind lediglich zwei Anregungen dargestellt, es ist jedoch möglich beliebig viele Anregungen aufeinanderfolgen zu lassen. Dabei sollte aber beachtet werden, dass sich die Eigenschaften eines Relais durch CSA ändern können. Bspw. kann sich der Überhub durch Abreiben von Material verringern. Deshalb ist es empfehlenswert, die Anzahl der Anregungen zu begrenzen und t_{ein} stetig anhand von Messreihen aus normalen Schaltvorgängen zu aktualisieren.

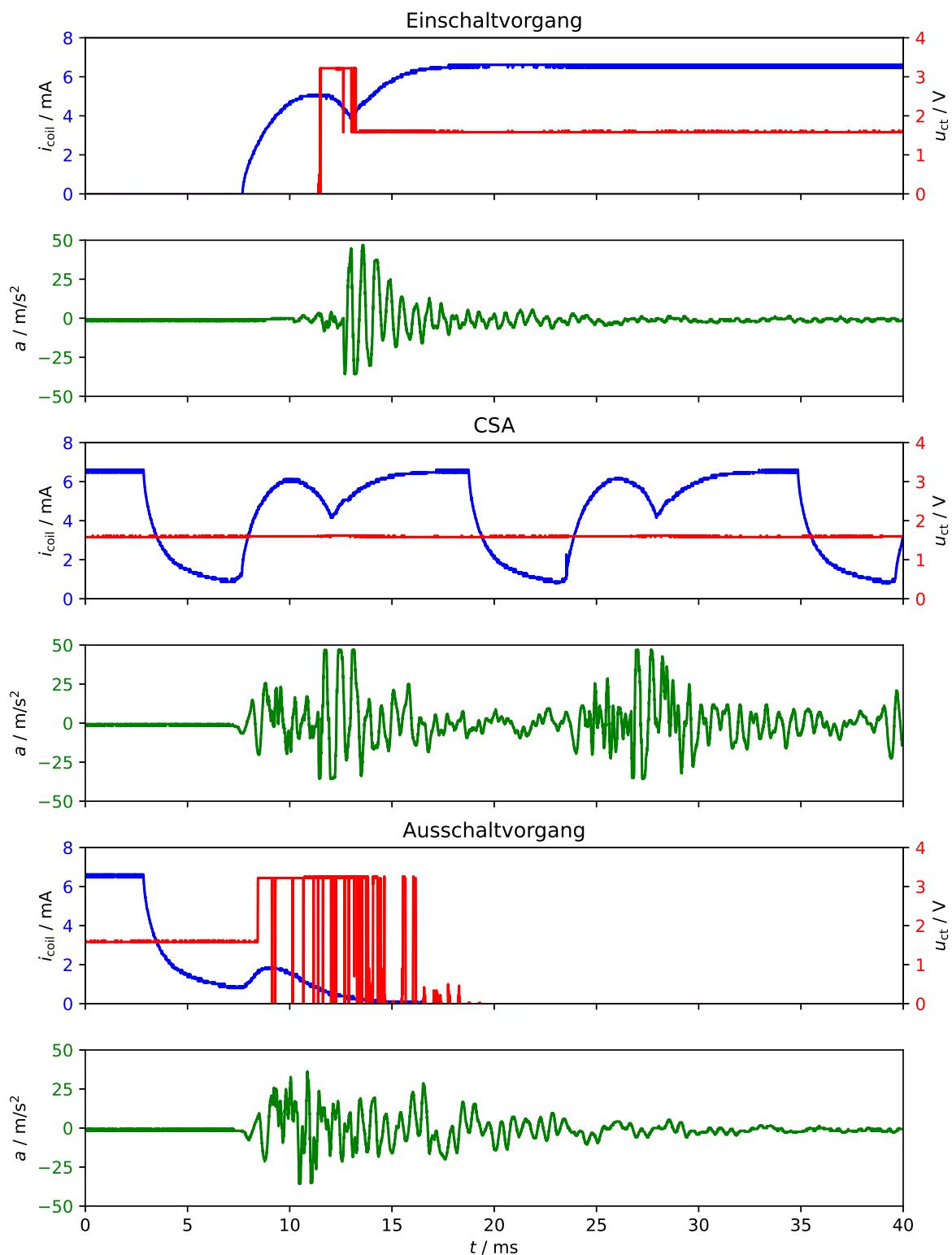


Abb. 5.7: Messgrößen beim Ein- und Ausschalten sowie Reiben

5.5.2 Einfluss der Zeitverschiebung

Der Zeitpunkt t_{ein} ist von großer Bedeutung für die Reibungsanregung. Wenn die Versorgungsspannung früher wieder angelegt wird, dann wird auch der Anker früher an die Spule herangezogen und vice versa. Deshalb wird der Einfluss einer Verschiebung von t_{ein} zum Start der Ankerbewegung auf die maximale Beschleunigung untersucht. Dazu wird zu t_{ein} ein Offset addiert, das im Intervall von -0,5 bis 0,5 ms liegt. Die Grenzen sind auf die Physik des Relais zurückzuführen; geringere Werte verhindern ein Lösen des Ankers und damit das Anregen einer Reibung. Bei höheren Werten lösen sich die Kontakte voneinander. Für ein Offset von 0 ms entspricht t_{ein} dem Zeitpunkt zu dem beim Ausschalten ein lokales Minimum ausgemacht werden kann.

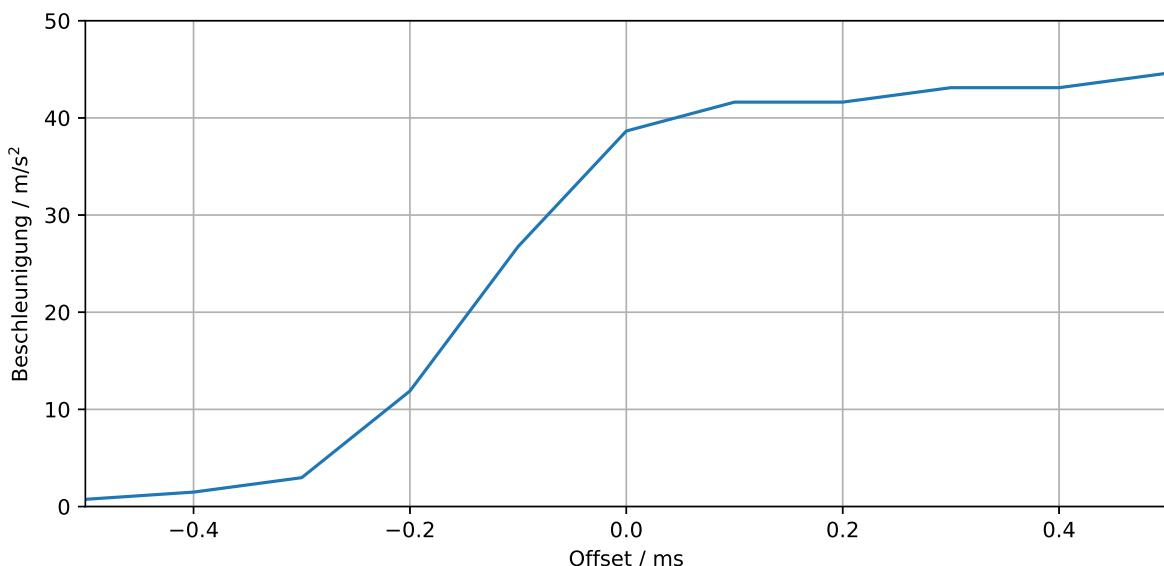


Abb. 5.8: Maximale Beschleunigung beim Reiben für verschiedene Offsets

In Abb. 5.8 ist die maximale Beschleunigung beim Reiben für unterschiedliche Einschaltzeitpunkte angegeben, wobei die Schrittweite 0,1 ms beträgt. Mit einem Offset kleiner -0,3 ms sind nur geringe Beschleunigungen kleiner 5 m/s^2 zu beobachten. Im Bereich zwischen -0,3 und 0,0 ms steigt die gemessene maximale Beschleunigung beim Reiben auf ca. 40 m/s^2 an. Im weiteren Verlauf bis zum Offset von 0,5 ms ist nur noch eine geringfügige Steigerung auszumachen. Begründet werden kann dies mit der Geometrie des Relais, die die maximale Geschwindigkeit und damit den Impuls des Ankers limitiert.

Aus den Beobachtungen können drei Schlussfolgerungen gezogen werden:

- Zu geringe Offsets (hier $<-0,3 \text{ ms}$) führen zu keiner signifikanten Reibung.
- Mit einem Offset zwischen -0,3 und 0,0 ms kann die Beschleunigung eingestellt werden.

- Positive Offsets führen zu keinen bedeutenden Zuwächsen hinsichtlich der Beschleunigung, implizieren jedoch die Gefahr, dass sich die Kontakte voneinander lösen.

Abschließend ist zu empfehlen, ein Offset von 0 ms zu wählen, da für größere Offsets keine signifikante Steigerung der Beschleunigung erreicht werden kann, das Risiko für ein Lösen der Kontakte jedoch stark erhöht wird.

5.5.3 Einflüsse auf die Kontaktobertäfelche

Im Weiteren wird der Einfluss von CSA auf die Kontaktobertäfelche analysiert. Dazu wurde der Messaufbau aus Abb. 5.3 leicht modifiziert, da in realen Applikationen von Relais ein Laststrom zu erwarten ist, mit dem eine Erwärmung der Kontakt einhergehen kann. Deshalb wurde der Spannungsteiler zur Generierung des Kontaktsignals u_{ct} durch eine Last mit 6Ω ersetzt. Mit dem Relais wird diese bei einer Spannung 24 V geschaltet, womit sich ein maximaler Laststrom von 4 A ergibt.

CSA wurde für jedes Relais eine Minute angewendet, jedoch nicht permanent: CSA und der normale eingeschaltete Zustand wechseln alle 0,5 s, um u.a. die thermische Belastung der Kontakte zu reduzieren. Die Relais stammen aus den Lebensdaueruntersuchungen. Dabei wurden nur Relais betrachtet, die durch ein Verschweißen bzw. Verhaken der Kontakte ausgefallen sind. Durch den Betrieb bis zum Ende der Betriebsdauer (siebter defekter Schaltvorgang) stellen diese Relais einen Extremfall dar. Bei einer früheren Anwendung von CSA, bspw. bei ersten Anzeichen von Verschleiß oder beim ersten Defekt, sind geringere Verschleißerscheinungen an den Kontakten zu erwarten, was eine Selbstheilung begünstigt. Dennoch ist diese Untersuchung geeignet, um das Potenzial von CSA aufzuzeigen.

Insgesamt wurden 20 Relais untersucht, bei denen Fotos von den Kontakten vor und nach der Anwendung von CSA gemacht wurde. Auf Basis dieser Fotos wurde die Auswirkung von CSA auf das jeweilige Relais in eine von drei Kategorien eingeordnet:

- Erfolgreiche Anwendung: Vor CSA sind Ablagerungen o.Ä. an den Kontakten sichtbar, die nach CSA geringer bzw. nicht mehr vorhanden sind.
- Keine Auswirkungen: Vor und nach CSA sind keine Unterschiede auszumachen.
- Kritische Belastung: Durch CSA kam es zu einer thermischen Belastung, sodass z.B. der Betätigungsarm geschmolzen ist.

Von den 20 Relais konnte CSA bei 15 erfolgreich angewendet werden, bei vier weiteren ist keine Auswirkung zu erkennen und ein Relais zeigt eine kritische Belastung auf. Für eine erste quantitative Untersuchung von CSA lässt sich somit eine positive Bilanz ziehen: In allen Fällen konnte das Relais durch CSA länger betrieben werden und lediglich in einem Fall sind Spuren einer thermischen Überlastung auszumachen, die den Betrieb jedoch nicht beeinträchtigt haben.

Im Folgenden werden die Auswirkungen qualitativ untersucht. Dazu sind fünf Abbildungen dargestellt, die – neben den Fotos der Kontakte – auch drei Messreihen vom Spulenstrom beim Ausschalten zeigen. Die Messreihe ‚Initial‘ zeigt den Spulenstrom des neuwertigen

Relais und 'vor CSA' bzw. 'nach CSA' den Spulenstrom entsprechend vor und nach der Anwendung von CSA.

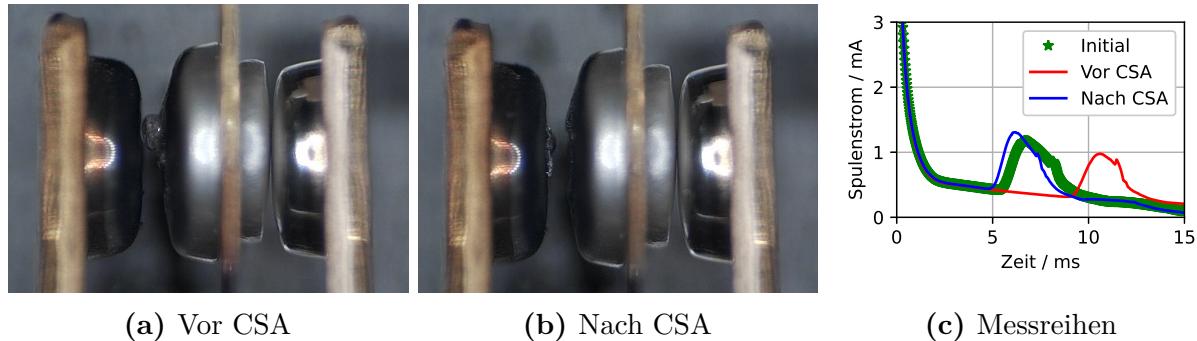


Abb. 5.9: Auswirkungen von CSA auf das Relais 973

In Abb. 5.9(a) ist ein Relais dargestellt, bei dem sich eine Pille am COM-Kontakt gebildet hat. Infolgedessen wird der Trennabstand zwischen den Kontakten COM und NO signifikant verkürzt, was bei hohen Spannungen kritisch ist. Weiterhin zeigt sich im Spulenstrom vor CSA, dass der Anker sich zu einem späteren Zeitpunkt von der Spule löst. Dies weist darauf hin, dass sich auf der gegenüberliegenden Seite der Pille auf dem NO-Kontakt ein Krater gebildet hat, in den die Pille beim Schließen verhakt und so der Überhub reduziert wird. Andernfalls müsste durch die Pille ein stärkerer Überhub hervorgerufen werden, was zu einem früheren Lösen des Ankers führen würde.

Durch die Anwendung von CSA konnte die Pille erfolgreich abgerieben werden und ist in Abb. 5.9(b) nur noch in Form einer Ablagerung am NO-Kontakt zu erkennen. Bei diesem Relais konnte erfolgreich eine abrasive Selbstheilung durchgeführt werden. Dies spiegelt sich auch im Spulenstrom wider. Im Vergleich zum initialen Zustand startet die Ankerbewegung nun etwas früher, was mit der Ablagerung zu begründen ist.

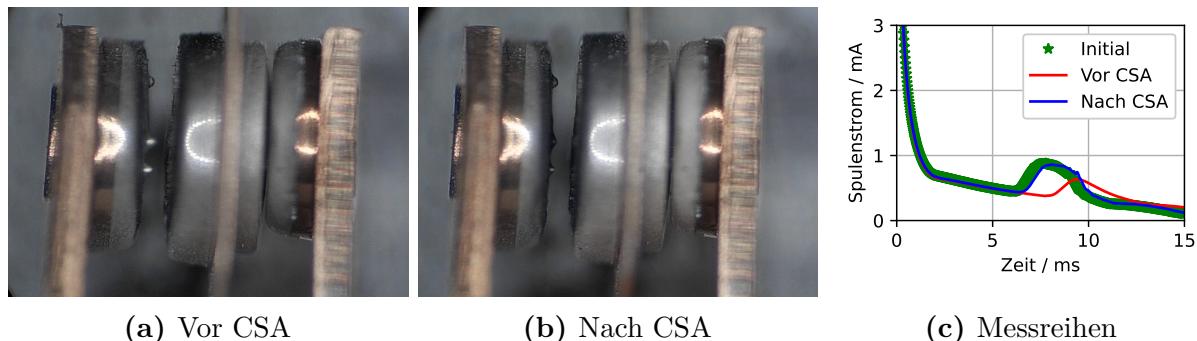


Abb. 5.10: Auswirkungen von CSA auf das Relais 974

Das Relais in Abb. 5.10(a) weist drei Materialanhäufungen am NO-Kontakt auf. Auch hier scheint es gegenüberliegend zu einer Kraterbildung gekommen zu sein, da der Start der Ankerbewegung im Spulenstrom vor CSA später als initial ist. Zudem steigt der Strom nicht so stark an: Dies kann ein Hinweis auf eine langsamere Ankerbewegung sein, die mit einem Verhaken der Kontaktpillen begründet werden kann. Nach CSA ist in Abb. 5.10(b) zu erkennen, dass die Materialanhäufungen durch CSA reduziert werden konnten und sich der Verlauf des Spulenstroms wieder dem initialen angenähert hat.

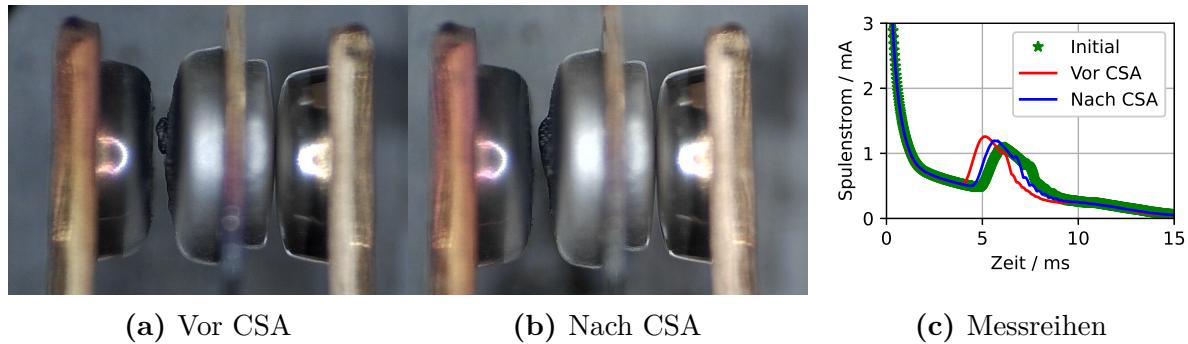


Abb. 5.11: Auswirkungen von CSA auf das Relais 988

Auf den Kontakten in Abb. 5.11(a) ist ähnlich zu Abb. 5.9(a) eine Pille am COM-Kontakt zu erkennen. In diesem Fall findet jedoch eine frühere Ankerbewegung statt (vgl. Abb. 5.11(c)). Durch CSA wird die Pille kleiner gerieben, weshalb sich der Kontaktabstand vergrößert und der Spulenstrom wieder dem initialen Verlauf ähnelt.

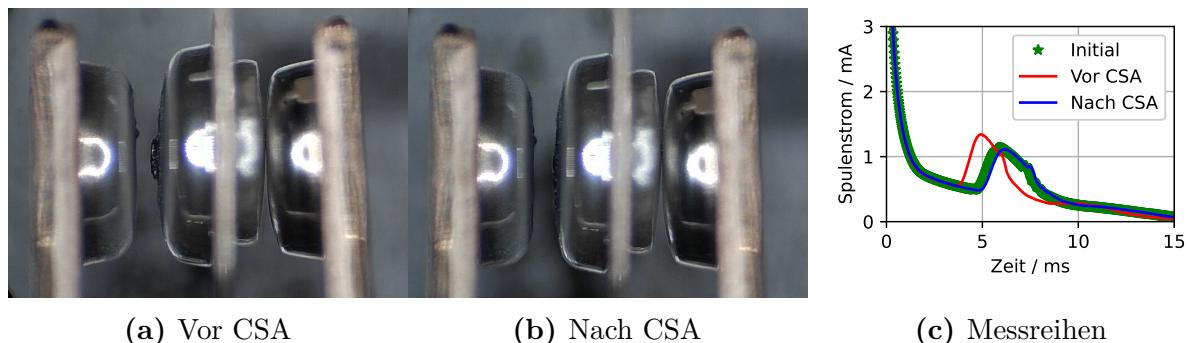


Abb. 5.12: Auswirkungen von CSA auf das Relais 1003

Bei Abb. 5.12(a) ist auf beiden Kontakten eine Materialanhäufung zu erkennen. Diese ist jedoch geringer als bei den vorherigen Beispielen. Dies resultiert in einer früheren Ankerbewegung. Die Anwendung von CSA führt bei diesen Kontakten zu einer Oberfläche, die zumindest in Hinblick auf die Optik (vgl. Abb. 5.12(b)) und auf die Messgrößen (vgl. Abb. 5.12(c)), derer eines neuen Relais entspricht. Der minimal spätere Startpunkt der

Ankerbewegung im Spulenstrom zeigt einen geringen Verbrauch von Kontaktmaterial an, was in Anbetracht der Betriebsdauer zu erwarten ist.

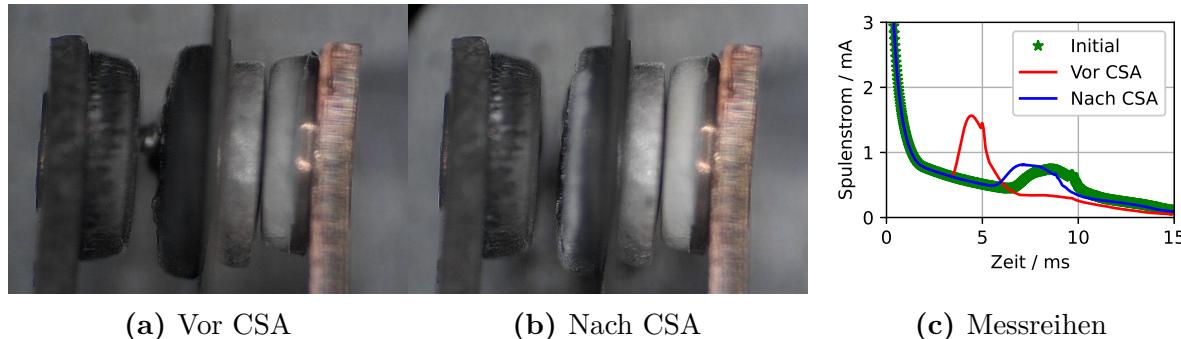


Abb. 5.13: Auswirkungen von CSA auf das Relais 1020

Bei Abb. 5.13(a) sind die Kontakte eines Relais dargestellt, das eine induktive Last geschaltet hat. Dies lässt sich am Ruß an den Oberflächen erkennen, der durch den Lichtbogen entsteht. Hier bildete sich zwischen den Kontakten eine Brücke, sodass die Kontakte nicht mehr getrennt werden können. Durch den Einsatz von CSA konnte die Materialbrücke gelöst und das Relais wieder funktionstüchtig gemacht werden (vgl. Abb. 5.13(b)). Dies wird durch den Spulenstrom nach CSA bestätigt, da die Ankerbewegung wieder später stattfindet.

Zusammenfassend lässt sich festhalten, dass durch CSA unterschiedliche Verschleißzustände von Relais korrigiert werden können; von leichten Ablagerungen (vgl. Abb. 5.12) über deutlich erkennbare Pillen (vgl. Abb. 5.9) hin zu Materialbrücken zwischen den Kontakten (vgl. Abb. 5.13). Zur Dauerhaftigkeit dieser Änderungen muss noch weitere Forschung betrieben werden, aber schon jetzt ist auszumachen, dass durch CSA eine kurzfristige Selbstheilung erreicht werden kann.

5.6 Fazit

In vorangegangenen Kapitel wurde das Verfahren CSA vorgestellt, mit dem ein Reiben zwischen den Kontakten von Relais hervorgerufen werden kann. Das ist vorteilhaft, da so zum einen fehlerhafte Schaltzustände korrigiert und zum anderen Verunreinigungen an den Kontaktobерflächen reduziert werden können. CSA ist simpel: Das Verfahren kann im eingeschalteten Zustand eines Relais genutzt werden, in dem die Versorgungsspannung kurzfristig abgeschaltet wird, wodurch eine Bewegung des Ankers hervorgerufen wird. Diese löst ein Reiben zwischen den Kontakten aus, ohne dass diese sich voneinander lösen. Dadurch kann das Verfahren im Betrieb eines Relais verwendet werden. Vorteilhaft ist dabei, dass durch CSA kein zusätzliches Sicherheitsrisiko entsteht wenn es nach einem Ausfall genutzt wird – schließlich sind die Relais dann bereits fehlerhaft.

Die grundsätzliche Funktionalität von CSA wurde im Rahmen der Evaluation nachgewiesen. Des Weiteren wurde der Einfluss des Einschaltzeitpunktes auf die Intensität des Reibens untersucht. Schlussendlich konnte gezeigt werden, dass Kontakte durch die Reibung geheilt werden konnten, wodurch der Kern des Forschungsvorhabens erfüllt wurde. Zu den Erkenntnissen wurde ein Patent angemeldet [124].

Als wissenschaftlicher Beitrag sind folgende Erkenntnisse zu betonen:

- Durch CSA kann ein Reiben der Kontakte hervorgerufen werden, ohne dass diese getrennt werden.
- Das Reiben kann in seiner Intensität durch die Wahl des Einschaltzeitpunktes eingestellt werden.
- Die Oberfläche von Kontakten kann durch Reiben von Ablagerungen befreit werden.

Die vorgestellten Ergebnisse sind ein erster Schritt zur Selbstheilung von Relaiskontakten. Deshalb bleiben einige Fragestellungen offen, die im Rahmen weiterer Forschungsvorhaben untersucht werden sollten:

- Welche physikalischen Phänomene, die bei CSA auftreten, haben welchen Einfluss auf die Veränderung der Kontaktobерfläche?
- Kann durch die Kombination von PdM und CSA ein proaktives System entwickelt werden, das durch gezieltes Einsetzen von abrasivem Verschleiß die Lebensdauer eines Relais erhöht?
- Welche Beschränkungen müssen beim Reiben beachtet werden? Im Rahmen von ersten Experimenten zeigte sich bspw., dass die thermische Belastung der Kontakte beachtet werden muss. Bei zu großer Erwärmung können die Kontaktlegierungen ihre Eigenschaften verändern.
- Ist es möglich, ein geheiltes Relais auch langfristig weiterzubetreiben?

6 Fazit

Abschließend werden die Beiträge dieser Arbeit zusammengefasst und offene Fragestellungen zu den Untersuchungsthemen aufgeworfen.

6.1 Beiträge der Arbeit

Der Fokus dieser Arbeit liegt auf der Entwicklung von Algorithmen zur Verbesserung der Instandhaltung von elektromechanischen Relais mit künstlicher Intelligenz. Kern der Forschung ist „Predictive Maintenance“ (dt. Prädiktive Instandhaltung) (PdM) – bei deren Erarbeitung jedoch zwei weitere Themen aufgeworfen wurden: das Reduzieren des Prellens und die Selbstheilung von Relais. Die wesentlichen Beiträge zu diesen drei Forschungsfeldern werden nachfolgend erörtert.

6.1.1 Prädiktive Instandhaltung

PdM ist von besonderem Interesse, da Relais aufgrund diverser Phänomene ausfallen können und deshalb auch bei gleichen Betriebsbedingungen unterschiedliche Lebensdauern aufweisen. Für die Instandhaltung ergibt sich dadurch ein Problem: Nach dem aktuellen Stand der Technik werden die Relais in einem Intervall ausgetauscht, das auf einem statistischen Erwartungswert basiert. Folglich fallen viele Relais vor der Wartung aus und führen u.a. zu Anlagenstillständen und Sicherheitsrisikos. Gleichzeitig wird bei vielen Relais ein Teil der Lebensdauer nicht genutzt, was in Bezug auf die Nachhaltigkeit problematisch ist.

Deshalb wurde in dieser Arbeit das Verfahren MAUD vorgestellt, mit dem während des Betriebs eines Relais anhand von Messungen individuell ein Ausfall-Indikator abgeschätzt werden kann. Dazu wurde ein umfangreicher Datensatz generiert und für das semi-überwachte Training eines ANNs verwendet. Weiterhin wurde die Inferenz des ANN auf einem Mikrocontroller erfolgreich geprüft. Folgende Erkenntnisse sind festzuhalten:

- Relais degradieren aufgrund verschiedener Phänomene, die komplementär auftreten können. Viele bereits erforschte Bauelemente (wie Kugellager) weisen einen simpleren Verschleiß auf, weshalb ein Vergleich von diesem mit Relais nur begrenzt möglich ist.

Schlussendlich bietet der generierte und veröffentlichte Datensatz [28] die Möglichkeit, Lösungen für dieses Forschungsdesiderat zu entwickeln und zu evaluieren.

- Entsprechend der zuvor erwähnten höheren Komplexität der Degradierung können Verfahren nach dem Stand der Technik – wie die Schätzung der Restlebensdauer oder eine Anomalieerkennung – nicht erfolgreich auf Relais übertragen werden.
- Mit einem semi-überwachten Training wird bei dem Verfahren MAUD ein ANN dazu verwendet, Pseudo-Label für den Datensatz zu generieren, mit denen ein Indikator für einen Ausfall geschätzt werden kann. Dazu werden die Schaltzyklen ausgehend von initialen Labeln immer wieder iterativ für das Training eines ANN und anschließend die Inferenz genutzt. Auf diesem Weg kann eine Unterscheidung zwischen auffälligen und unauffälligen Schaltzyklen erlernt und schlussendlich ein Signal gegeben werden. Die Überlegenheit von MAUD gegenüber dem Stand der Technik konnte mit dem umfangreichen Datensatz nachgewiesen werden: Die Auslastung kann um 17 Prozentpunkte auf 45 % verbessert und unentdeckte Ausfälle um 6 Prozentpunkte auf 3 % reduziert werden.
- Durch Optimierungen können sowohl die benötigte Rechenleistung als auch der Speicherbedarf für die Inferenz eines ANN signifikant gesenkt werden. Dadurch ist eine Implementierung von MAUD auf einem Mikrocontroller möglich, was für die Produktisierung der Forschungsergebnisse von wesentlicher Bedeutung ist.

6.1.2 Reduzieren des Prellens

Die Kontakte von Relais prellen beim Schließen aufgrund der notwendigen Bewegung des COM-Kontaktes und der Anpresskraft, die über eine Biegung der Kontakte erreicht wird. Durch konstruktive Optimierungen wird das Prellen bereits durch die Hersteller reduziert – ist jedoch im Besonderen bei degradierten Relais weiterhin zu beobachten. Das Prellen ist unerwünscht, da es u.a. den Verschleiß steigert, Geräusche erzeugt, den Schaltvorgang verlängert und geschaltete Signale stört.

Aus dem Stand der Technik geht hervor, dass das Prellen durch eine optimierte Steuerung des Relais weiterhin reduziert werden kann. Dabei ergibt sich eine Herausforderung bezüglich eines robusten Algorithmus, der stetig nach weiterem Optimierungspotential sucht und gleichzeitig eine hohe Ausbeute in Form einer Prellreduzierung erreicht. Folglich wurde im Kontext dieser Arbeit eine maßgeschneiderte Variante einer Partikelschwarmoptimierung entwickelt und praktisch getestet. Aus diesen Untersuchungen ergeben sich folgende Erkenntnisse:

- Die Dauer des Prellens variiert unter konstanten Umgebungsparametern, weshalb zur Optimierung mehrfache Evaluationen notwendig sind. Zudem sind sowohl die Prelldauer als auch der Einfluss der Steuerung für jedes Relais individuell, sodass jedes Relais einzeln optimiert werden muss.
- Das entwickelte Verfahren BOA, das aus einer Kombination aus bare-bone und evolutionären PSO unter Berücksichtigung der Partikeldichte besteht, ist geeignet,

um das Prellen zuverlässig zu reduzieren. Dabei wird BOA gegenüber einer klassischen PSO durch den bare-bone Ansatz in der Rechenkomplexität vereinfacht und durch den evolutionären Teil und die Dichteschätzung robust gemacht gegenüber Rauschen.

- Im Rahmen der Evaluation konnte gezeigt werden, dass BOA dem Stand der Technik mit einer Reduktion um 79 % beim Einschalten und um 59 % beim Ausschalten überlegen ist.
- BOA wurde für einen Mikrocontroller entwickelt und evaluiert, sodass eine Realisierung im Produkt möglich erscheint.

6.1.3 Abrasive Kontaktselftheilung

Die Fehlfunktion eines Relais ist nicht immer endgültig. Oftmals reicht ein erneutes Schalten aus, um die Funktionstüchtigkeit zumindest kurzfristig wiederherzustellen. In der Anwendung ist dies von großer Bedeutung, da der Ausfall bzw. die Fehlfunktion einer Anlage so unterbunden werden kann.

Im Kontext dieser Arbeit wurde ein neuartiges Verfahren entwickelt, das über das erneute Schalten hinausgeht: Durch eine spezielle Ansteuerung wird eine Reibung zwischen den Kontaktpaaren hervorgerufen. So können Materialanhäufungen oder Verunreinigungen an den Kontakten abgeschliffen werden. Folgende Erkenntnisse im Rahmen der Untersuchungen zu dieser Thematik sind hervorzuheben:

- Das entwickelte Verfahren CSA nutzt den Überhub eines Relais aus, um Reibung zwischen den Kontakten auszulösen. Dabei löst sich lediglich der Anker kurz vom Spulenkern, sodass die Kontakte stets leitfähig bleiben.
- Die Intensität der Reibung kann durch einen Parameter von CSA beeinflusst werden.
- In ersten Versuchen mit degradierten Relais konnte gezeigt werden, dass Materialanhäufungen abgebaut werden können.
- Die langfristige Wirkung von CSA wurde nicht erforscht, weshalb sich der praktische Nutzen von CSA bisher nicht abschätzen lässt.

6.2 Offene Fragestellungen

Bei der Bearbeitung der Forschungsthemen ergeben sich weitere potenzielle Themen, die im Folgenden aufgelistet werden. Dabei werden drei Forschungsfelder getrennt betrachtet.

6.2.1 Prädiktive Instandhaltung

Die vorgeschlagene Lösung MAUD zeigt das Potenzial für PdM im Kontext von Relais auf. Retrospektiv ergeben sich jedoch folgende Fragestellungen, die ergründet werden könnten:

- Ein erheblicher Teil des Aufwands – besonders in zeitlicher Hinsicht – ist in die Generierung des Datensatzes geflossen. Für zukünftige Forschungsvorhaben wäre ein Verfahren wünschenswert, mit dem aus einem kleinen Datensatz ein großer erzeugt werden kann. Eine mögliche Lösung könnten generative, künstliche neuronale Netze sein. Der in dieser Arbeit vorgestellte Datensatz kann für derartige Forschungsvorhaben die Datengrundlage bilden, indem die Generalisierungsfähigkeit eines Verfahrens zur Prädiktiven Instandhaltung, das jeweils auf dem originalen und synthetischen Datensatz trainiert wurde, gegenübergestellt wird.
- Bei den untersuchten Relais handelt es sich um eine Bauform von vielen. Eine Untersuchung zum Wissenstransfer von dieser Bauform zu anderen Bauformen könnte zu einer Reduktion des Testaufwands führen.
- Für die Anwendung in einem Produkt ist ein weiterer Aspekt wesentlich: In der Anwendung wird ein Relais oftmals eine begrenzte Variation der Betriebsbedingungen (wie z.B. die Last) erfahren. Deshalb ist die Anpassung des neuronalen Netzes an diese Bedingungen wünschenswert. Hierzu müssten Konzepte entwickelt werden, wie das Training für eine derartige Überanpassung gestaltet werden kann.
- Der semi-überwachte Pseudo-Label Ansatz von MAUD könnte auf die Schätzung der Restlebensdauer übertragen werden: Dabei würde die kontinuierliche Restlebensdauer auf Basis der Prädiktionen von maschinellen Lernverfahren korrigiert werden. Auf diesem Weg ist es möglich, Anomalien, reversible Alterungsprozesse oder nicht monotone Prozesse zu berücksichtigen und die Generalisierungsfähigkeit der maschinellen Lernverfahren so zu verbessern. Eine Schwierigkeit hierbei stellt die Begrenzung der Korrektur dar: Erste Untersuchungen im Kontext dieser Arbeit schlugen fehl, da die Zielgröße so stark korrigiert wurde, dass die Prädiktionen entweder einer binären Klassifikation glichen oder alle gegen einen Wert konvergierten.

6.2.2 Reduzieren des Prellens

Mit dem Verfahren BOA kann zuverlässig das Prellen reduziert werden. Eine praktische Anwendung ist bereits jetzt möglich. Dennoch könnten folgende Untersuchungen zur Etablierung eines derartigen Verfahrens in Produkten beitragen:

- Es bestehen lediglich Untersuchungen, die zeigen, dass Prellen die Lebensdauer verkürzt. Für die Anwendung von BOA ist es jedoch interessant, zu quantifizieren, wie weit die Lebensdauer von Relais verlängert wird. Eine Implementierung von BOA in Dauertests wird bezüglich dieser Fragestellung Aufschluss geben.
- BOA ist unter der Voraussetzung konzipiert worden, dass die Kontaktseite des Relais überwacht werden kann. Dies impliziert Kosten für eine Messeinrichtung. Ein Verfahren, das ausschließlich auf dem Spulenstrom des Relais beruht, könnte entscheidende Kostenvorteile mit sich bringen. Dazu müsste mit einem geeigneten Verfahren z.B. die Position des Ankers aus dem Spulenstrom geschätzt werden – eine für maschinelle Lernverfahren prädestinierte Aufgabe.

- Eine weitere Möglichkeit zur Reduzierung der Kosten stellt die Anwendung einer Steuerung ohne Optimierung dar. D.h., es würde im Rahmen von Untersuchungen ein Steuersignal entwickelt, das in einem breiten Anwendungsspektrum zu einer Reduktion des Prellens führt. Diese ist wahrscheinlich nicht auf dem Niveau von BOA, aber auch eine geringere Reduktion kann einen positiven Einfluss auf die Lebensdauer haben.
- Das Verfahren BOA ist für diverse elektromechanische Aktuatoren anwendbar: Vom Schütz bis zum Ventil ergeben sich viele Applikationen, bei denen die Lebensdauer durch eine optimierte Steuerung verlängert werden kann. Ein Beispiel ist das Abschalten von Ventilen: Hier kann die Kavitation durch ein langsames Reduzieren des Fluidstroms unterbunden werden.

6.2.3 Abrasive Kontaktselftheilung

Das Verfahren CSA befindet sich in einem frühen Stadium: Es ist theoretisch beschrieben und praktisch an ersten Relais getestet. Die auftretenden Phänomene und deren Langzeitfolgen sind jedoch nicht bekannt. Daher ergeben sich folgende Fragestellungen:

- Welche Phänomene treten beim Reiben der Kontakte auf? Ist ausschließlich abrasiver Verschleiß nachzuweisen? Oder sind auch thermische Effekte zu beobachten? Die Ergebnisse dieser Untersuchung könnten die Grundlage für nachfolgende Forschungsvorhaben zur Optimierung von CSA bilden.
- Welche Beschränkungen sind beim CSA einzuhalten? Im Rahmen der Untersuchungen dieser Arbeit konnte z.B. eine thermische Überlastung der Kontakte beobachtet werden. Für eine praktische Anwendung ist dies zu verhindern.
- Welche langfristigen Folgen ergeben sich nach der Anwendung von CSA? Ist das Verfahren nur geeignet, um die Relais kurzfristig wieder nutzbar zu machen, oder ist auch ein langfristiger Betrieb von geheilten Relais möglich? Dauertests mit durch CSA geheilten Relais können Aufschluss zu dieser Fragestellung bringen.
- Kann ein Verfahren wie MAUD mit CSA kombiniert werden, sodass Relais bereits vor Auftreten einer Fehlfunktion geheilt werden? Eine erfolgreiche Kombination birgt viel Potenzial: Die Lebensdauer von Relais könnte signifikant verlängert werden.

In einer Produktanwendung werden die drei Themen sich jedoch auch überschneiden, weshalb die Untersuchung der Wechselwirkungen interessant ist. So werden CSA und BOA die Degradierung und damit die Datengrundlage von MAUD beeinflussen, gleichzeitig können die Partikel von BOA als Eingangsgröße von MAUD genutzt werden. Die Indikation durch MAUD hingegen kann als Auslöser für die Nutzung von CSA vor einem Ausfall genutzt werden, sodass schon bei kleinen Änderungen proaktiv gehandelt werden kann.

6.3 Ausblick

Mit den Verfahren MAUD, BOA und CSA wurden in dieser Arbeit drei Ansätze präsentiert, mit denen in Zukunft die Instandhaltung von Relais verbessert werden kann. Vorteilhaft ist hierbei, dass die Hardwareanforderungen gleich sind, sodass die Verfahren komplementär eingesetzt werden können. Deshalb wird zum Abschluss dieser Arbeit hypothetisch betrachtet, welche Anforderungen für eine Produktumsetzung erfüllt werden müssen. Bei dem Produkt würde es sich um ein Modul handeln, das auf einer Tragschiene montiert werden kann. Das Relais könnte gewechselt werden, ohne das Modul zu demontieren und Werkzeug zu verwenden. Die Verfahren würden als Teil der Firmware implementiert werden und über eine Schnittstelle Informationen an eine Steuerung übermitteln, bzw. darüber konfigurierbar sein. Zudem würde eine Signalleuchte verwendet, um direkt am Produkt die wichtigsten Informationen zum Zustand anzeigen zu können.

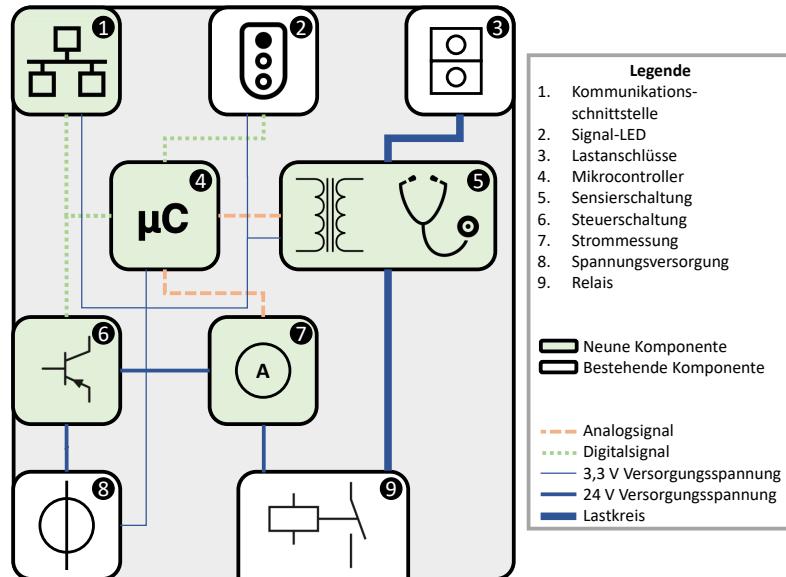


Abb. 6.1: Notwendige Komponenten für eine Umsetzung der in dieser Arbeit vorgestellten Verfahren in einem Produkt

In Abb. 6.1 sind die notwendigen Teilkomponenten für eine Produktrealisierung der Verfahren abstrahiert dargestellt. Die weiß gefärbten Komponenten sind in konventionellen Relais-Sockeln bereits enthalten. Die grün eingefärbten Komponenten müssten für eine Integration der Verfahren ergänzt werden. Die einzelnen Komponenten werden entsprechend ihrer Beschriftung nachfolgend beschrieben:

1. Kommunikationsschnittstelle

Die Kommunikationsschnittstelle wird für drei Aspekte zwingend benötigt. Erstens müssen von einer überlagerten Steuerung die Befehle für das Ein- und Ausschalten des Relais gegeben werden. Zweitens ist eine Möglichkeit zur Konfiguration des Moduls erforderlich, damit bspw. eine Aktualisierung eines ANN für MAUD vorgenommen werden kann. Drittens ist die Übermittlung der Ergebnisse von MAUD, d.h. der Ausfall-Indikator, an eine überlagerte Steuerung wichtig.

Daneben wäre es vorteilhaft, wenn die Schnittstelle ausreichend Bandbreite zur Verfügung stellen würde, um die Messdaten i_{coil} und u_{ct} auf einem externen Speichermedium zu sichern. Da die Messreihen für einen Schaltzyklus viermal 200 Byte umfassen, wäre eine Übertragungsrate von 1 kB/s bereits ausreichend. Auf diesem Weg würde bspw. die Möglichkeit eröffnet, dass reale Anlagendaten für das Training verwendet und so die Performanz gesteigert werden könnte.

2. Signal-LED

Für die Anwendung ist es unumgänglich, den Betriebsstatus des Produktes direkt durch eine LED anzuzeigen. Für die MAUD wird eine rote Signalleuchte benötigt, mit der ein drohender Ausfall angezeigt werden kann. Das Verfahren CSA soll nur im Fehlerfall oder bei Auffälligkeiten zum Einsatz kommen, dementsprechend könnte dies mit einer gelben Signalleuchte angezeigt werden. Die Betriebsbereitschaft sollte durch ein grünes Signal angezeigt werden. Bei den unterschiedlichen Signalleuchten wäre es durch Blinken und konstantes Leuchten möglich, den Schaltzustand des Relais darzustellen. Zusammenfassend könnte mit einer zweifarbigen LED (rot und grün) die Signal-LED implementiert werden, mit der mittels additiver Farbmischung gelbes Licht erzeugt werden kann.

3. Lastanschlüsse

Für den Anschluss von verschiedenen Lasten ist eine mechanische Vorrichtung erforderlich. Je nach Anwender und Markt bestehen hier unterschiedliche Anforderungen, denen Phoenix Contact mit einem modularen System begegnet, welches bei diesem Produkt zur Verwendung käme.

4. Mikrocontroller

Bei den drei Verfahren MAUD, BOA und CSA handelt es sich um Algorithmen, für deren Umsetzung es eines Prozessors und Peripheriefunktionen bedarf. Dementsprechend wird ein Mikrocontroller benötigt. Dieser muss mindestens über vier digitale Ausgänge verfügen: für die zweifarbige LED, die Sensierschaltung und die Steuerschaltung. Zudem ist ein Analog-Digital-Wandler mit zwei Kanälen für i_{coil} und u_{ct} erforderlich. Schlussendlich sind – je nach Kommunikationsschnittstelle – weitere Anforderungen zu erfüllen. Der Prozessor benötigt ausreichend RAM, ROM und (Gleitkomma-) Rechenoperationen pro Sekunde, um die Algorithmen und im Speziellen ANN auszuführen. In Bezug auf die Ergebnisse aus Abschn. 3.7 wäre eine mögliche Wahl ein Cortex-M4 Mikrocontroller vom Unternehmen STMicroelectronics.

5. Sensierschaltung

Die Information über den Kontaktierungszustand des Relais ist zwingend erforderlich. Eine Messschaltung zu diesem Zweck muss diverse Anwendungsfelder von Relais unterstützen, z.B. Gleichspannungs- und Wechselspannungslastkreise. Weiterhin darf eine Messung auf der Lastseite keinen Einfluss auf den Lastkreis haben. Diese Herausforderungen sind nicht trivial zu lösen. Deshalb wurde im Rahmen eines anderen Forschungsvorhabens von Phoenix Contact eine Sensierschaltung entwickelt, mit der der Zustand der Kontakte überwacht werden kann.

Bei der Sensierschaltung wird ein Oszillatator angeregt, dessen Signal mittels eines Leiterplattenübertrages auf die Lastseite übertragen wird. Dort ist die Impedanz abhängig vom Schaltzustand des Relais, was sich auf die Frequenz des angeregten Signals auswirkt. Somit kann auf den Schaltzustand der Kontakte zurückgeschlossen werden. Eine detaillierte Beschreibung ist [91] zu entnehmen.

6. Steuerschaltung

Mit der Steuerschaltung wird der Ausgang des Mikrocontrollers auf das Spannungs-level des Relais umgesetzt. Deshalb ist eine einfache Schaltung, wie in Abschn. 4.2 erörtert, ausreichend, um die Funktionalität zu gewährleisten.

7. Strommessung

Die Strommessung des Spulenstroms i_{coil} findet auf der Steuerseite statt, weshalb keine externen Limitierungen beachtet werden müssen.

8. Spannungsversorgung

Die unterschiedlichen Komponenten müssen mit Spannung versorgt werden. Deshalb würde ein Produkt eine Bauteilgruppe benötigen, mit der unterschiedliche Spannungen erzeugt werden können.

9. Relais

Schlussendlich wird ein Relais benötigt, das in einen Sockel gesteckt wird, damit eine schnelle, werkzeuglose Wartung durchgeführt werden kann.

Die Aufzählung der notwendigen Komponenten, von denen der Großteil neu eingeführt werden müsste und die im Vergleich zu den bestehenden Komponenten kostenintensiv sind, macht deutlich, dass ein mögliches Produkt wesentlich höhere Produktionskosten hervorrufen würde als ein konventioneller Relais-Sockel. Die Bewertung, ob der gesteigerte Kundennutzen die Mehrkosten rechtfertigt, kann nur durch eine detaillierte Untersuchung beantwortet werden, die u.a. Produktionskosten und Absatzprognosen beinhaltet, weshalb sie nicht Gegenstand dieser Arbeit ist.

Für eine zukünftige Anwendung empfehlen sich daher hochpreisige Relais bzw. Produkte, bei denen die notwendigen Komponenten bereits eingebracht sind. Weiterhin ist eine schrittweise Umsetzung Erfolg versprechend, bei der zunächst BOA und CSA etabliert werden. Für MAUD sind jedoch Kooperationen mit Anwendern erstrebenswert, da die

Performanz und Konzepte wie maßgeschneiderte MAUD-Versionen erst mit realen Daten abschließend evaluiert werden können.

Verzeichnisse

Tabellenverzeichnis

2.1	Technische Spezifikationen des Relaistyps nach Hersteller	7
3.1	Datensatzübersicht [6]	18
3.2	Genutzte Lasten [6]	58
3.3	Datei-Attribute [6]	69
3.4	Datei-Gruppen [6]	69
3.5	Aufteilung des Datensatzes [89]	70
3.6	ANN-Topologie	73
3.7	VAE Topologie	75
3.8	ANN-Topologie [89]	84
3.9	Genutzte Parameterwerte [89]	86
3.10	Ergebnisse [89]	90
3.11	Topologie der Kombination aus Encoder mit MLP	92
3.12	Topologie des MLP	93
3.13	Performanz und Speicherbedarf von verschiedenen Versionen des MLP .	95
3.14	Performanz und Speicherbedarf von verschiedenen Versionen des AE .	96
3.15	Vergleich von MACC und Inferenzdauer der besten Versionen	97
4.1	BOA Algorithmus Variablen [100]	122
4.2	Quantitativer Vergleich von BOA und R2R [100]	128

Abbildungsverzeichnis

1.1	Ein elektromechanisches Relais ohne Gehäuse	1
1.2	Drei Herausforderungen bei der Instandhaltung von Relais nach dem Stand der Technik und in dieser Arbeit vorgestellte Beiträge zur Lösung	4
2.1	Aufbau eines elektromechanischen Relais	8
2.2	Darstellung der verschiedenen Phasen eines Schaltvorgangs	9
2.3	Messkurven beim Einschalten	9
2.4	Messkurven beim Ausschalten	11

3.1	Visualisierung der verfügbaren Datensätze in Bezug auf die Anzahl der Einheiten und Datenpunkte pro Einheit [6]	20
3.2	RUL-Schätzungen mittels MLP [35]	25
3.3	RUL-Schätzung eines RNN [35]	26
3.4	Spannungsverlauf der Steuerseite beim Öffnen des Kontakts [49]	29
3.5	Übersichtsgrafik der einzelnen Teilkomponenten von EMRUA[4]	32
3.6	Zusammenhang von Eingangsdaten \mathbf{X} , maschinellen Lernverfahren und geschätzten Ausgangsdaten $\hat{\mathbf{y}}$	40
3.7	Beispielhafter Verlauf des Fehlers auf Trainings- und Validierungsset	41
3.8	Neuronales Netz mit zwei Schichten	42
3.9	Faltende Schicht aus Faltung, Aktivierungsfunktion und Bündelung	45
3.10	Aufbau eines Autoencoders mit Encoder und Decoder aus jeweils zwei Schichten	47
3.11	Aufbau eines Variational Autoencoders mit Encoder und Decoder aus jeweils zwei Schichten	48
3.12	Visualisierung von geteilten Parametern	51
3.13	Schaltung Steuerseite [6]	52
3.14	Schaltung Lastseite [6]	53
3.15	Bild eines Testsystems mit zwei Leiterplatten [89]	54
3.16	Komponenten und Zusammenhänge eines Testsystems	54
3.17	Unverarbeitete Messreihen eines Einschaltvorgangs mit einer Abtastrate von 200 kHz	55
3.18	Unverarbeitete Messreihen eines Ausschaltvorgangs mit einer Abtastrate von 200 kHz	56
3.19	Frequenzgang, Phasengang und Gruppenlaufzeit des verwendeten Filter .	57
3.20	Beispielhafte Messgrößen mit 200 kHz und 10 kHz Abtastrate	58
3.21	Beispielhafte Visualisierung des kontinuierlich degradierten Relais 294 .	60
3.22	Beispielhafte Visualisierung der spontanen Verschlechterung des Relais 84 [6]	60
3.23	Beispielhafte Visualisierung des Relais 299, bei dem reversible Veränderungen aufgetreten sind	61
3.24	Anzahl der degradierten Relais nach Hersteller und Last [6]	62
3.25	Histogramme der Lebensdauern der Relais unterschieden nach Art des Defekts, Hersteller und Last [6]	63
3.26	Extrahierte Merkmale, markiert und nummeriert auf exemplarischen Messreihen [6]	65
3.27	Extrahierte Merkmale Nr. 5, 6, 11, 12 aus \mathbf{u}_{ct} beim Ein- und Ausschalten .	66
3.28	Extrahierte Merkmale Nr. 1, 2, 3, 4 aus $\mathbf{i}_{coil; on}$	67
3.29	Extrahierte Merkmale Nr. 7, 8, 9, 10 aus $\mathbf{i}_{coil; off}$	68
3.30	Häufigkeitsverteilung der Relaisdefekte, unterschieden nach Hersteller und Last [89]	71
3.31	Beispielhafte RUL-Prädiktionen von drei Relais mit einem ANN	73
3.32	Fehler für das Trainings- und Validierungsset über die Trainingsepochen .	75
3.33	Vergleich der originalen und vom VAE rekonstruierten Zeitreihen	76
3.34	MSE der rekonstruierten Zeitreihen von verschiedenen Relais [6]	76

3.35	Box-Plot Diagramme der MSE von einzelnen Relais	78
3.36	Überblick über die Datenverarbeitungspipeline [89]	80
3.37	Vergleich von Messreihen von zwei unterschiedlichen Herstellern [89][96] .	81
3.38	Kontaktspannung beim Ausschalten für unterschiedliche Lasten [89] . .	82
3.39	Trainingsprozedur [89]	84
3.40	Historie der Nutzung, Ausfälle und geänderten Label über die Iterationen [89]	86
3.41	Visualisierung der Prädiktionen für das Relais 294 [89]	88
3.42	Visualisierung der Prädiktionen für das Relais 84	89
3.43	Visualisierung der Prädiktionen für das Relais 299	89
4.1	Funktionsweise einer PSO anhand eines Partikels über 5 Iterationen . . .	105
4.2	Funktionsweise einer BPSO anhand eines Partikels über 5 Iterationen . .	106
4.3	Funktionsweise einer EPSO anhand eines Partikels über 5 Iterationen. .	108
4.4	Steuer- und Lastkreis des genutzten Messaufbaus [100]	109
4.5	Foto vom genutzten Hardwareaufbau, mit Mikrocontroller (links) und 16 Relais (rechts)	110
4.6	Beispielhafter normaler und optimierter Einschaltvorgang eines Relais [100]	112
4.7	Beispielhafter normaler und optimierter Ausschaltvorgang eines Relais [100]	113
4.8	Histogramm vom Prellen zweier normal geschalteter Relais beim Ein- und Ausschalten [100]	115
4.9	Heatmap der Prellreduktion für unterschiedliche Kombinationen aus $t_{\text{ein; aus}}$ und $d_{\text{ein; aus}}$ [100]	116
4.10	Simulierte relative Prellreduktion von verschiedenen PSO Varianten . .	118
4.11	Blockdiagramm des entwickelten Algorithmus [100]	118
4.12	Exemplarische Darstellung der Partikel in den ersten drei Iterationen .	123
4.13	Exemplarische Darstellung der Partikel in der ersten, zehnten und zwanzigsten Iteration [100]	124
4.14	Verlauf des Prellens für zwei Relais im Ein- und Ausschalten unter Anwendung von BOA [100]	125
4.15	Vergleich des Verlaufs des Prellens für 48 Relais im Ein- und Ausschalten beim R2R und BOA Algorithmus [100]	127
4.16	Exemplarische Visualisierung der Funktionsweise von BOA beim unangekündigten Tausch des Relais [100]	129
4.17	Verlauf des Prellens beim unangekündigten Tausch des Relais [100] . .	130
4.18	Mittlere Änderung der Schaltdauer beim Ein- und Ausschalten bei einer Stichprobe von 16 Relais [100]	132
5.1	Anteil der Relais aufgetragen über der Anzahl der Schaltzyklen zwischen dem ersten und siebten defekten Schaltvorgang	137
5.2	Skizze des verwendeten Messaufbaus	138
5.3	Foto des verwendeten Messaufbaus	139
5.4	Ablaufdiagramm für eine Implementierung von CSA	139
5.5	Messgrößen während eines normalen Abschaltvorgangs [124]	140
5.6	Messgrößen während des Reibens [124]	141

5.7	Messgrößen beim Ein- und Ausschalten sowie Reiben	143
5.8	Maximale Beschleunigung beim Reiben für verschiedene Offsets	144
5.9	Auswirkungen von CSA auf das Relais 973	146
5.10	Auswirkungen von CSA auf das Relais 974	146
5.11	Auswirkungen von CSA auf das Relais 988	147
5.12	Auswirkungen von CSA auf das Relais 1003	147
5.13	Auswirkungen von CSA auf das Relais 1020	148
6.1	Notwendige Komponenten für eine Umsetzung der in dieser Arbeit vorgestellten Verfahren in einem Produkt	156

Abkürzungen

AE	„Autoencoder“
ANN	„Artificial neural network“ (dt. künstliches neuronales Netz)
BOA	„Bouncing-Optimization-Algorithm“ (dt. Preloptimierungsalgorithmus)
BPSO	„Bare-Bones Particle-Swarm-Optimization“ (dt. rudimentäre Partikel Schwarm Optimierung)
CNN	„Convolutional neural network“ (dt. faltendes neuronales Netz)
COM	„Common Contact“ (dt. gemeinsamer Kontakt)
CSA	„Contact Self-healing via Abrasion“ (dt. Kontaktselftheilung via Abrasion)
EMRUA	„Electromagnetic Relay Useful Actuation“
EPSO	„Evolutionary Particle-Swarm-Optimization“ (dt. evolutionäre Partikel Schwarm Optimierung)
GAN	„Generative adversarial networks“ (dt. erzeugende gegnerische Netzwerke)
MACC	„Multiply and Accumulate Complexity“ (dt. Komplexität der Multiplikationen und Akkumulationen)
MAUD	„Maintenance-Algorithm for Unlabeled Data“ (dt. Instandhaltungsalgorithmus für ungelabelte Daten)
MLP	„Multilayer Perceptron“ (dt. mehrschichtiges Perzepron)
MSE	„Mean Squared Error“ (dt. Mittlerer quadrierter Fehler)
NC	„Normally Closed Contact“ (dt. normalerweise geschlossener Kontakt)
NO	„Normally Opened Contact“ (dt. normalerweise geöffneter Kontakt)

PCR	„Phoenix Contact Relay-Dataset“
PdM	„Predictive Maintenance“ (dt. Prädiktive Instandhaltung)
PSO	„Particle-Swarm-Optimization“ (dt. Partikel Schwarm Optimierung)
R2R	„Run-to-Run-Algorithm“
ReLU	„Rectified Linear Unit“ (dt. Gleichrichterfunktion)
RNN	„Recurrent neural network“ (dt. rekurrentes neuronales Netz)
RUL	„Remaining useful lifetime“ (dt. nutzbare Restlebensdauer)
VAE	„Variational Autoencoder“

Glossar

Abrasion Wenn harte Teilchen oder Profilspitzen bei Reibung in die Oberfläche eines Körpers eindringen, dann führt dies zu einem Abtrag der Oberfläche. Dieses Phänomen wird Abrasion genannt.

DC13-Last Die DC13-Last ist eine in der IEC 121A/427/CD beschriebene Standardlast [90]. Sie wird u.a. für die Testung von elektromechanischen Relais genutzt.

Erregerfrequenz Als Erregerfrequenz wird die Frequenz bezeichnet, mit der eine erzwungene Schwingung angeregt wird.

Prellen Beim Schalten von Relais können mehrere unelastische Stöße zwischen den Kontaktpaaren hervorgerufen werden, welche auch als Prellen bezeichnet werden.

Pseudo-Label Bei Pseudo-Labels handelt es sich um Label, die während eines semi-überwachten Trainings mit einem maschinellen Lernverfahren definiert und im Weiteren für das Training genutzt werden.

semi-überwacht Beim semi-überwachten Training von maschinellen Lernverfahren werden sowohl markierte als auch unmarkierte Daten genutzt. Aus dieser Kombination leitet sich die Bezeichnung ab.

Whisker Die Linien, die die Box bei einem Box-Plot Diagramm verlängern, werden als Whisker bezeichnet.

Lebenslauf

Name, Vorname Winkel, Fabian
Geburtsdatum/ -ort 20.01.1994 in Bielefeld
Nationalität deutsch
Familienstand ledig, 1 Sohn (0 Jahre)
E-Mail fabian.winkel@me.com

Schulbildung

08/2000 - 07/2004 Katholische Grundschule Stukenbrock
09/2004 - 07/2013 Gymnasium SHS
Abschluss: Abitur (2,2)
Leistungskurse: Mathematik, Physik

Studium

10/2013 - 09/2014 Maschinenbau
Leibniz Universität Hannover, nicht beendet
10/2014 - 09/2017 Wirtschaftsingenieurwesen Elektrotechnik
Universität Paderborn, Abschluss: B.Sc. (1,7)
Abschlussarbeit: Kamerabasierter Abstandsregeltempomat
auf einer eingebetteten GPU-Plattform
10/2017 - 12/2019 Wirtschaftsingenieurwesen Elektrotechnik
Universität Paderborn, Abschluss: M.Sc. (1,3)
Studienarbeit: Evaluierung optimaler Anregungsprofile für einen
permanenterregten Synchronmotor zur Erhöhung der Daten-
diversifikation im Kontext des Trainings neuronaler Netze
Abschlussarbeit: Evaluierung maschineller Lernverfahren im
Kontext der Alterungsanalyse elektromechanischer Relais

Beruf

seit 01/2023 Machine Learning Engineer
Intellectual Property, Phoenix Contact, Blomberg
01/2020 - 12/2022 Promotionsstudent
Technologieentwicklung Interface, Phoenix Contact,
Bad Pyrmont
08/2016 - 03/2019 Werkstudent
Applikation, dSPACE, Paderborn
10/2015 - 12/2016 Studentische Hilfskraft
GET Lab, Universität Paderborn

Sprachkenntnisse

Deutsch
Englisch

Literatur

- [1] V. Gurevich, *Electric relays: principles and applications*. CRC press, 2005.
- [2] E. Ramirez-Laboreo, C. Sagües und S. Llorente, „A New Model of Electromechanical Relays for Predicting the Motion and Electromagnetic Dynamics,“ *IEEE Transactions on Industry Applications*, Jg. 52, Nr. 3, S. 2545–2553, 2016. DOI: 10.1109/TIA.2016.2518120.
- [3] W. Rieder, *Elektrische Kontakte - eine Einführung in ihre Physik und Technik*, Neuerscheinung. Berlin: VDE-Verlag, 2000.
- [4] L. Kirschbaum, V. Robu, J. Swingler und D. Flynn, „Prognostics for Electromagnetic Relays Using Deep Learning,“ *IEEE Access*, Jg. 10, S. 4861–4895, 2022. DOI: 10.1109/ACCESS.2022.3140645.
- [5] *Electromechanical Relay Market - forecasts from 2021 to 2026*, Zugriffsdatum: 10. August 2023. Adresse: <https://www.researchandmarkets.com/reports/5332646/electromechanical-relay-market-forecasts-from>.
- [6] F. Winkel, J. Deuse-Kleinsteuber und J. Böcker, „Run-to-Failure Relay Dataset for Predictive Maintenance Research With Machine Learning,“ *IEEE Transactions on Reliability*, S. 1–14, 2023. DOI: 10.1109/TR.2023.3255786.
- [7] J. Lee, H. Qiu, G. Yu, J. Lin und Rexnord Technical Services (2007), IMS, University of Cincinnati. „Bearing Data Set“, *NASA Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA*, Zugriffsdatum: 20. Oktober 2023. Adresse: <https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/>.
- [8] A. Agogino und K. Goebel (2007), BEST Lab, UC Berkeley, „Milling Data Set“, *NASA Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA*, Zugriffsdatum: 20. Oktober 2023. Adresse: <https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/>.
- [9] B. Saha und K. Goebel (2007), „Battery Data Set“, *NASA Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA*, Zugriffsdatum: 20. Oktober 2023. Adresse: <https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/>.
- [10] A. Saxena und K. Goebel (2008), „Turbofan Engine Degradation Simulation Data Set“, *NASA Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA*, Zugriffsdatum: 20. Oktober 2023. Adresse: <https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/>.
- [11] J. R. Celaya, A. Saxena, S. Saha und K. Goebel (2011), „MOSFET Thermal Overstress Aging Data Set“, *NASA Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA*, Zugriffsdatum: 20. Oktober 2023. Adresse: <https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/>.
- [12] P. Nectoux, R. Gouriveau, K. Medjaher, E. Ramasso, B. Morello, N. Zerhouni und C. Varnier (2012), „FEMTO Bearing Data Set“, *NASA Prognostics Data Repository*,

-
- NASA Ames Research Center, Moffett Field, CA*, Zugriffsdatum: 20. Oktober 2023. Adresse: <https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/>.
- [13] Abhinav Saxena, Kai Goebel, Cecilia C. Larrosa und Fu-Kuo Chang, "CFRP Composites Data Set", *NASA Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA*, Zugriffsdatum: 20. Oktober 2023. Adresse: <https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/>.
- [14] B. Bole, C. Kulkarni und M. Daigle (2014), "Randomized Battery Usage Data Set", *NASA Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA*, Zugriffsdatum: 20. Oktober 2023. Adresse: <https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/>.
- [15] C. Kulkarni, E. Hogge, C. Quach und K. Goebel (2015), "HIRF Battery Data Set", *NASA Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA*, Zugriffsdatum: 20. Oktober 2023. Adresse: <https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/>.
- [16] C. Kulkarni und A. Guarneros (2015), "Small Satellite Power Simulation Data Set", *NASA Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA*, Zugriffsdatum: 20. Oktober 2023. Adresse: <https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/>.
- [17] J. Renwick, C. Kulkarni und J Celaya (2015), "Capacitor Electrical Stress Data Set", *NASA Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA*, Zugriffsdatum: 20. Oktober 2023. Adresse: <https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/>.
- [18] T. Schneider, S. Klein und M. Bastuck (2015), "Predictive Maintenance Of Hydraulics System", *ZeMA gGmbH, Saarbrücken, Germany*, Zugriffsdatum: 20. Oktober 2023. Adresse: <https://www.kaggle.com/datasets/mayank1897/condition-monitoring-of-hydraulic-systems>.
- [19] Y. Pandya (2018), "Gearbox Fault Diagnosis Data", *National Renewable Energy Laboratory, USA*, Zugriffsdatum: 20. Oktober 2023. Adresse: <https://data.openei.org/submissions/623>.
- [20] "Versatile Production System", *Institute Industrial IT, Ostwestfalen-Lippe University of Applied Sciences, Germany*, Zugriffsdatum: 20. Oktober 2023. Adresse: <https://www.kaggle.com/datasets/inIT-OWL/versatileproductionsystem>.
- [21] C. Axenie und S. Bortoli (2020), "Predictive maintenance dataset", Zugriffsdatum: 20. Oktober 2023. Adresse: <https://zenodo.org/records/3653909>.
- [22] D. Tosato, D. Dalle Pezze, C. Masiero, G. Antonio und A. Beghi (2020), "Alarm Logs in Packaging Industry", Zugriffsdatum: 20. Oktober 2023. Adresse: <https://ieee-dataport.org/open-access/alarm-logs-packaging-industry-alpi>.
- [23] S. Matzka (2020), "Predictive Maintenance Dataset Data Set", *School of Engineering - Technology and Life, Berlin, Germany*, Zugriffsdatum: 20. Oktober 2023. Adresse: <https://archive.ics.uci.edu/ml/datasets/AI4I+2020+Predictive+Maintenance+Dataset>.

-
- [24] M. Chao, C. Kulkarni, K. Goebel und O. Fink (2021), "Aircraft Engine Run-to-Failure Dataset under real flight conditions", *NASA Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA*, Zugriffsdatum: 20. Oktober 2023. Adresse: <https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/>.
- [25] D. Viana (2021), "Diesel Engine Faults Features Dataset", Zugriffsdatum: 20. Oktober 2023. Adresse: <https://ieee-dataport.org/documents/diesel-engine-faults-features-dataset-default>.
- [26] S. Hagemeyer, F. Mauthe und P. Zeiler (2021), "Creation of Publicly Available Data Sets for Prognostics and Diagnostics Addressing Data Scenarios Relevant to Industrial Applications", *International Journal of Prognostics and Health Management*, Zugriffsdatum: 20. Oktober 2023. Adresse: <https://papers.phmsociety.org/index.php/ijphm/article/view/3087>.
- [27] D. Martins, D. Viana, A. Lima, D. Hadadd, R. Homero und L. Vaz (2022), "Composed Fault Dataset (COMFAULDA)", Zugriffsdatum: 20. Oktober 2023. Adresse: <https://ieee-dataport.org/documents/composed-fault-dataset-comfaulda>.
- [28] F. Winkel und J. Deuse-Kleinsteuber, *Phoenix Contact Relay Dataset*, Kaggle ([kaggle.com / datasets / fabianwinkel / phoenix-contact-relay-dataset](https://kaggle.com/datasets/fabianwinkel/phoenix-contact-relay-dataset)), Phoenix Contact Electronics GmbH, Bad Pyrmont, Germany, 2022.
- [29] Y. Lei, N. Li, L. Guo, N. Li, T. Yan und J. Lin, „Machinery health prognostics: A systematic review from data acquisition to RUL prediction,“ *Mechanical Systems and Signal Processing*, Jg. 104, S. 799–836, Mai 2018. DOI: [10.1016/j.ymssp.2017.11.016](https://doi.org/10.1016/j.ymssp.2017.11.016).
- [30] P. Paris und F. Erdogan, „A Critical Analysis of Crack Propagation Laws,“ *Journal of Basic Engineering*, Jg. 85, Nr. 4, S. 528–533, Dez. 1963. DOI: [10.1115/1.3656900](https://doi.org/10.1115/1.3656900).
- [31] Y. Ran, X. Zhou, P. Lin, Y. Wen und R. Deng, „A survey of predictive maintenance: Systems, purposes and approaches,“ *arXiv preprint arXiv:1912.07383*, 2019.
- [32] N. Sakib und T. Wuest, „Challenges and opportunities of condition-based predictive maintenance: a review,“ *Procedia Cirp*, Jg. 78, S. 267–272, 2018.
- [33] Z. M. Çınar, A. Abdussalam Nuhu, Q. Zeeshan, O. Korhan, M. Asmael und B. Safaei, „Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0,“ *Sustainability*, Jg. 12, Nr. 19, S. 8211, 2020.
- [34] O. Serradilla, E. Zugasti, J. Rodriguez und U. Zurutuza, „Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects,“ *Applied Intelligence*, S. 1–31, 2022.
- [35] F. O. Heimes, „Recurrent neural networks for remaining useful life estimation,“ in *2008 International Conference on Prognostics and Health Management*, Okt. 2008, S. 1–6. DOI: [10.1109/PHM.2008.4711422](https://doi.org/10.1109/PHM.2008.4711422).
- [36] T. Kim und S. Sohn, „Multitask learning for health condition identification and remaining useful life prediction: deep convolutional neural network approach,“ *Journal of Intelligent Manufacturing*, Jg. 32, Dez. 2021. DOI: [10.1007/s10845-020-01630-w](https://doi.org/10.1007/s10845-020-01630-w).
- [37] T. Xia, Y. Song, Y. Zheng, E. Pan und L. Xi, „An ensemble framework based on convolutional bi-directional LSTM with multiple time windows for remaining useful life estimation,“ *Computers in Industry*, Jg. 115, 2020.

-
- [38] Y. Zhang, R. Xiong, H. He und M. G. Pecht, „Long Short-Term Memory Recurrent Neural Network for Remaining Useful Life Prediction of Lithium-Ion Batteries,“ *IEEE Transactions on Vehicular Technology*, Jg. 67, Nr. 7, S. 5695–5705, 2018. DOI: 10.1109/TVT.2018.2805189.
 - [39] T. H. Loutas, D. Roulias und G. Georgoulas, „Remaining useful life estimation in rolling bearings utilizing data-driven probabilistic E-support vectors regression,“ *IEEE Transactions on Reliability*, Jg. 62, Nr. 4, S. 821–832, 2013.
 - [40] A. Soualhi, K. Medjaher und N. Zerhouni, „Bearing health monitoring based on Hilbert–Huang transform, support vector machine, and regression,“ *IEEE Transactions on instrumentation and measurement*, Jg. 64, Nr. 1, S. 52–62, 2014.
 - [41] K. Javed, R. Gouriveau, N. Zerhouni und P. Nectoux, „Enabling health monitoring approach based on vibration data for accurate prognostics,“ *IEEE Transactions on industrial electronics*, Jg. 62, Nr. 1, S. 647–656, 2014.
 - [42] O. Serradilla, E. Zugasti, J. Ramirez de Okariz, J. Rodriguez und U. Zurutuza, „Adaptable and explainable predictive maintenance: Semi-supervised deep learning for anomaly detection and diagnosis in press machine data,“ *Applied Sciences*, Jg. 11, Nr. 16, S. 7376, 2021.
 - [43] M. De Benedetti, F. Leonardi, F. Messina, C. Santoro und A. Vasilakos, „Anomaly detection and predictive maintenance for photovoltaic systems,“ *Neurocomputing*, Jg. 310, S. 59–68, 2018. DOI: <https://doi.org/10.1016/j.neucom.2018.05.017>.
 - [44] A. Santolamazza, V. Cesari und V. Introna, „Anomaly detection in energy consumption for Condition-Based maintenance of Compressed Air Generation systems: an approach based on artificial neural networks,“ *IFAC-PapersOnLine*, Jg. 51, Nr. 11, S. 1131–1136, 2018. DOI: <https://doi.org/10.1016/j.ifacol.2018.08.439>.
 - [45] F. Winkel, „Evaluierung maschineller Lernverfahren im Kontext der Alterungsanalyse elektromechanischer Relais,“ Masterarbeit, Universität Paderborn, Institut für Elektrotechnik und Informationstechnik, Fachgebiet Nachrichtentechnik, 2019.
 - [46] CADFEM. „Der Digitale Zwilling – Simulation im Betrieb.“ Zugriffsdatum: 20. Oktober 2023. (2018), Adresse: <https://www.cadfecm.net/de/de/branchen-themen/referenzen/referenz/digitaler-zwilling-simulation-im-betrieb.html>.
 - [47] H. Wilker, *Band 3: Weibull-Statistik in der Praxis: Leitfaden zur Zuverlässigkeitsermittlung technischer Komponenten*. Books on Demand, 2013. Adresse: <https://books.google.de/books?id=zaExAQAAQBAJ>.
 - [48] L. Papula, „Mathematik für Ingenieure und Naturwissenschaftler 3: Vektoranalysis, Wahrscheinlichkeitsrechnung, Mathematische Statistik, Fehler- und Ausgleichsrechnung,“ in Vieweg+Teubner Verlag, 2013, Kap. Wahrscheinlichkeitsrechnung. Adresse: <https://books.google.de/books?id=AKzPBgAAQBAJ>.
 - [49] M. E. A. Remy, J. Desforges und F. J. M. Rolard, „Method and device for diagnosing wear of an electrical switching unit, and electrical unit comprising such a device,“ US Patent: US15831541, 5.12.2017.
 - [50] R. Orban und J. Desforges, „Method and device for diagnosing wear of an electrical switching unit, and electrical unit comprising such a device,“ US Patent: US11079435, 10.04.2019.

-
- [51] S. Ganssloser, „Verschleißerkennung von Relais anhand des elektrischen Verhaltens mittels Verfahren des maschinellen Lernens,“ Universität Stuttgart Institut für Automatisierungstechnik und Softwaresysteme, 2018.
- [52] L. Kirschbaum, F. Dinmohammadi, D. Flynn, V. Robu und M. Pecht, „Failure Analysis Informing Embedded Health Monitoring of Electromagnetic Relays,“ in *2018 3rd International Conference on System Reliability and Safety (ICSRS)*, 2018, S. 261–267. DOI: 10.1109/ICSRS.2018.8688839.
- [53] L. Kirschbaum, D. Roman, V. Robu und D. Flynn, „Deep Learning Pipeline for State-of-Health Classification of Electromagnetic Relays,“ in *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, 2021, S. 1–7. DOI: 10.1109/ISIE45552.2021.9576278.
- [54] J. E. Van Engelen und H. H. Hoos, „A survey on semi-supervised learning,“ *Machine Learning*, Jg. 109, Nr. 2, S. 373–440, 2020.
- [55] I. Triguero, S. García und F. Herrera, „Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study,“ *Knowledge and Information systems*, Jg. 42, Nr. 2, S. 245–284, 2015.
- [56] Y. Wang, X. Xu, H. Zhao und Z. Hua, „Semi-supervised learning based on nearest neighbor rule and cut edges,“ *Knowledge-Based Systems*, Jg. 23, Nr. 6, S. 547–554, 2010. DOI: <https://doi.org/10.1016/j.knosys.2010.03.012>. Adresse: <https://www.sciencedirect.com/science/article/pii/S095070511000047X>.
- [57] M. Li und Z.-H. Zhou, „SETRED: Self-training with Editing,“ in *Advances in Knowledge Discovery and Data Mining*, T. B. Ho, D. Cheung und H. Liu, Hrsg., Berlin, Heidelberg: Springer, 2005, S. 611–621.
- [58] C. Leistner, A. Saffari, J. Santner und H. Bischof, „Semi-supervised random forests,“ in *12th international conference on computer vision*, IEEE, 2009, S. 506–513.
- [59] D. Yarowsky, „Unsupervised word sense disambiguation rivaling supervised methods,“ in *33rd annual meeting of the association for computational linguistics*, 1995, S. 189–196.
- [60] D.-H. Lee, „Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks,“ in *Workshop on challenges in representation learning, ICML*, Bd. 3, 2013, S. 896.
- [61] E. Arazo, D. Ortego, P. Albert, N. E. O’Connor und K. McGuinness, „Pseudo-labeling and confirmation bias in deep semi-supervised learning,“ in *International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2020, S. 1–8.
- [62] J. Cheng, P.-s. Wang, G. Li, Q.-h. Hu und H.-q. Lu, „Recent advances in efficient computation of deep convolutional neural networks,“ *Frontiers of Information Technology & Electronic Engineering*, Jg. 19, Nr. 1, S. 64–77, 2018.
- [63] Y. LeCun, J. Denker und S. Solla, „Optimal brain damage,“ *Advances in neural information processing systems*, Jg. 2, 1989.
- [64] B. Hassibi und D. Stork, „Second order derivatives for network pruning: Optimal brain surgeon,“ *Advances in neural information processing systems*, Jg. 5, 1992.
- [65] Y. Guo, A. Yao und Y. Chen, „Dynamic network surgery for efficient dnns,“ *Advances in neural information processing systems*, Jg. 29, 2016.

-
- [66] V. Lebedev und V. Lempitsky, „Fast convnets using group-wise brain damage,“ in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, S. 2554–2564.
 - [67] J.-H. Luo, J. Wu und W. Lin, „Thinet: A filter level pruning method for deep neural network compression,“ in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, S. 5058–5066.
 - [68] J. L. Holi und J.-N. Hwang, „Finite precision error analysis of neural network hardware implementations,“ *IEEE Transactions on Computers*, Jg. 42, Nr. 3, S. 281–290, 1993.
 - [69] S. Gupta, A. Agrawal, K. Gopalakrishnan und P. Narayanan, „Deep learning with limited numerical precision,“ in *International conference on machine learning*, PMLR, 2015, S. 1737–1746.
 - [70] S. Han, H. Mao und W. J. Dally, „Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding,“ *arXiv preprint arXiv:1510.00149*, 2015.
 - [71] P. Stock, A. Joulin, R. Gribonval, B. Graham und H. Jégou, „And the bit goes down: Revisiting the quantization of neural networks,“ *arXiv preprint arXiv:1907.05686*, 2019.
 - [72] F. Tung und G. Mori, „Deep Neural Network Compression by In-Parallel Pruning-Quantization,“ *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Jg. 42, Nr. 3, S. 568–579, 2020. DOI: 10.1109/TPAMI.2018.2886192.
 - [73] G. Rebala, A. Ravi und S. Churiwala, *An introduction to machine learning*. Springer, 2019.
 - [74] W. Ertel, *Grundkurs Künstliche Intelligenz: Eine praxisorientierte Einführung* (Computational Intelligence). Springer Fachmedien Wiesbaden, 2016. Adresse: <https://books.google.de/books?id=ecD3DAAAQBAJ>.
 - [75] F. Chollet u. a., *Keras*, <https://keras.io>, 2015.
 - [76] M. Abadi u. a., *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, Software available from tensorflow.org, 2015. Adresse: <https://www.tensorflow.org/>.
 - [77] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever und R. R. Salakhutdinov, *Improving neural networks by preventing co-adaptation of feature detectors*, 2012. arXiv: 1207.0580.
 - [78] I. Goodfellow, Y. Bengio und A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
 - [79] K. He, X. Zhang, S. Ren und J. Sun, „Deep residual learning for image recognition,“ in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, S. 770–778.
 - [80] V. Dumoulin und F. Visin, „A guide to convolution arithmetic for deep learning,“ *arXiv preprint arXiv:1603.07285*, 2016.
 - [81] D. P. Kingma und M. Welling, „Auto-encoding variational bayes,“ *arXiv preprint arXiv:1312.6114*, 2013.
 - [82] J. An und S. Cho, „Variational autoencoder based anomaly detection using reconstruction probability,“ *Special Lecture on IE*, Jg. 2, Nr. 1, S. 1–18, 2015.

-
- [83] Google, *TensorFlow Lite guide*, Zugriffsdatum: 11. Januar 2023. Adresse: https://www.tensorflow.org/lite/performance/model_optimization.
- [84] STMicroelectronics, *AI-Erweiterungspaket für STM32CubeMX*, Zugriffsdatum: 11. Januar 2023. Adresse: <https://www.st.com/en/embedded-software/x-cube-ai.html>.
- [85] F. Sakr, F. Bellotti, R. Berta und A. De Gloria, „Machine Learning on Mainstream Microcontrollers,“ *Sensors*, Jg. 20, Nr. 9, 2020. DOI: 10.3390/s20092638. Adresse: <https://www.mdpi.com/1424-8220/20/9/2638>.
- [86] N. C. A. Sallang, M. T. Islam, M. S. Islam und H. Arshad, „A CNN-Based Smart Waste Management System Using TensorFlow Lite and LoRa-GPS Shield in Internet of Things Environment,“ *IEEE Access*, Jg. 9, S. 153 560–153 574, 2021. DOI: 10.1109/ACCESS.2021.3128314.
- [87] K. Dokic, M. Martinovic und D. Mandusic, „Inference speed and quantisation of neural networks with TensorFlow Lite for Microcontrollers framework,“ in *2020 5th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, 2020, S. 1–6. DOI: 10.1109/SEEDA-CECNSM49515.2020.9221846.
- [88] „Post-training quantization“, *TensorFlow*, Zugriffsdatum: 20. Oktober 2023. Adresse: https://www.tensorflow.org/lite/performance/post_training_quantization.
- [89] F. Winkel, O. Wallscheid, P. Scholz und J. Böcker, „Pseudo-Labeling Machine Learning Algorithm for Predictive Maintenance of Relays,“ *IEEE Open Journal of the Industrial Electronics Society*, S. 1–14, 2023. DOI: 10.1109/OJIES.2023.3323870.
- [90] „IEC 121A/427/CD: Low-voltage switchgear and controlgear part 5-1: Control circuit devices and switching elements electromechanical control circuit devices,“ *VDE Verlag*, Mai 2016.
- [91] P. Scholz und E. Schaper, „Switching monitoring device,“ US Patent: US20210223317, 27.05.2019.
- [92] M. Bichler, *B10-Wert, Zuverlässigkeit und Ausfallrate bei Relais*, Zugriffsdatum: 30. Dezember 2021. Adresse: <https://www.elektronikpraxis.vogel.de/b10-wert-zuverlaessigkeit-und-ausfallrate-bei-relais-a-122251/>.
- [93] D. P. Kingma und J. Ba, *Adam: A Method for Stochastic Optimization*, 2014. arXiv: 1412.6980.
- [94] X. Zhang, Z. Su, X. Hu, Y. Han und S. Wang, „Semisupervised Momentum Prototype Network for Gearbox Fault Diagnosis Under Limited Labeled Samples,“ *IEEE Transactions on Industrial Informatics*, Jg. 18, Nr. 9, S. 6203–6213, 2022. DOI: 10.1109/TII.2022.3154486.
- [95] F. Winkel, P. Scholz und E. Schaper, „Prädiktive Instandhaltung für einen elektromagnetischen Aktor,“ DE Patent: DE102021111192, 30.04.2021.
- [96] F. Winkel, „Typenidentifikation von elektromechanischen Aktuatoren,“ DE Patent: DE102023107934, Offenlegung 2024.
- [97] T. Kaps, „Applikation von neuronalen Netzen auf einem Mikrocontroller im Kontext der Prädiktiven Instandhaltung,“ Bachelorarbeit, Fachhochschule Bielefeld, Fachbereich Ingenieurwissenschaften und Mathematik, 2022.

-
- [98] STMicroelectronics, *STM32 32-bit Arm Cortex MCUs*, Zugriffsdatum: 10. August 2023. Adresse: <https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html#overview>.
- [99] STMicroelectronics, *DS11532 Rev 7*, Zugriffsdatum: 20. Oktober 2023. Adresse: <https://www.st.com/resource/en/datasheet/stm32f767zi.pdf>.
- [100] F. Winkel, P. Scholz, O. Wallscheid und J. Böcker, „Reducing Contact Bouncing of a Relay by Optimizing the Switch Signal During Run-Time,“ *IEEE Transactions on Automation Science and Engineering*, S. 1–11, 2023. DOI: 10.1109/TASE.2023.3322762.
- [101] W. Ren, Y. He, J. Jin und S. Man, „Investigations of the contact bounce behaviors and relative dynamic welding phenomena for electromechanical relay,“ *Review of Scientific Instruments*, Jg. 87, Nr. 6, 2016. DOI: 10.1063/1.4953838.
- [102] J. G. Ganssle, „A guide to debouncing,“ *Guide to Debouncing, Ganssle Group, Baltimore, MD, US*, S. 1–22, 2004.
- [103] S Domonkos und G. Madarász, „Elimination of the contact bounce phenomenon,“ *Periodica Polytechnica Electrical Engineering (Archives)*, Jg. 13, Nr. 3, S. 121–135, 1969.
- [104] B. O. Ciocirlan und H. Herrmann, „Switching Contact Bounce Reduction,“ in *Proceedings of the 55th IEEE Holm Conference on Electrical Contacts*, 2009, S. 57–66. DOI: 10.1109/HOLM.2009.5284423.
- [105] E. Ramirez-Laboreo, C. Sagües und S. Llorente, „A New Model of Electromechanical Relays for Predicting the Motion and Electromagnetic Dynamics,“ *IEEE Transactions on Industry Applications*, Jg. 52, Nr. 3, S. 2545–2553, 2016. DOI: 10.1109/TIA.2016.2518120.
- [106] T. Davies, H. Nouri und F. Britton, „Towards the control of contact bounce,“ 1995, S. 289–294. DOI: 10.1109/HOLM.1995.482884.
- [107] P. M. D. Santos Dias de Moraes und A. Perin, „An Electronic Control Unit for Reducing Contact Bounce in Electromagnetic Contactors,“ *Industrial Electronics, IEEE Transactions on*, Jg. 55, S. 861–870, 2008. DOI: 10.1109/TIE.2007.909073.
- [108] A. G. Espinosa, J.-R. R. Ruiz, J. Cusido und X. A. Morera, „Sensorless Control and Fault Diagnosis of Electromechanical Contactors,“ *IEEE Transactions on Industrial Electronics*, Jg. 55, Nr. 10, S. 3742–3750, 2008. DOI: 10.1109/TIE.2008.925773.
- [109] A. G. Espinosa, J.-R. R. Ruiz und X. A. Morera, „A Sensorless Method for Controlling the Closure of a Contactor,“ *IEEE Transactions on Magnetics*, Jg. 43, Nr. 10, S. 3896–3903, 2007. DOI: 10.1109/TMAG.2007.904355.
- [110] Y. Li, H. Lu, S. Tian, Z. Jiao und J.-T. Chen, „Posture Control of Electromechanical-Actuator-Based Thrust Vector System for Aircraft Engine,“ *IEEE Transactions on Industrial Electronics*, Jg. 59, Nr. 9, S. 3561–3571, 2012. DOI: 10.1109/TIE.2011.2159351.
- [111] L. Tang, Z. Han und Z. Xu, „A Sequential Adaptive Control Strategy for the Contact Colliding Speed of Contactors Based on Fuzzy Control,“ *IEEE Transactions on Industrial Electronics*, Jg. 68, Nr. 7, S. 6064–6074, 2021. DOI: 10.1109/TIE.2020.2994872.

-
- [112] E. Ramirez-Laboreo, C. Sagües und S. Llorente, „A New Run-to-Run Approach for Reducing Contact Bounce in Electromagnetic Switches,“ *IEEE Transactions on Industrial Electronics*, Jg. 64, Nr. 1, S. 535–543, 2017. DOI: 10.1109/TIE.2016.2605622.
- [113] E. Moya-Lasheras und C. Sagües, „Run-to-Run Control With Bayesian Optimization for Soft Landing of Short-Stroke Reluctance Actuators,“ *IEEE/ASME Transactions on Mechatronics*, Jg. 25, Nr. 6, S. 2645–2656, 2020. DOI: 10.1109/TMECH.2020.2987942.
- [114] R. Eberhart und J. Kennedy, „A new optimizer using particle swarm theory,“ in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995, S. 39–43. DOI: 10.1109/MHS.1995.494215.
- [115] A. Q. H. Badar, „Different Applications of PSO,“ in *Applying Particle Swarm Optimization: New Solutions and Cases for Optimized Portfolios*, B. A. Mercangöz, Hrsg. Cham: Springer International Publishing, 2021, S. 191–208. DOI: 10.1007/978-3-030-70281-6_11. Adresse: https://doi.org/10.1007/978-3-030-70281-6_11.
- [116] R. Oldenhuis, *FEX-Testfunctions*, Zugriffsdatum: 10. August 2023. Adresse: <https://github.com/rodyo/FEX-testfunctions/tree/v1.5>.
- [117] J. Kennedy, „Bare bones particle swarms,“ in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, 2003, S. 80–87. DOI: 10.1109/SIS.2003.1202251.
- [118] V. Miranda und N. Fonseca, „EPSO-evolutionary particle swarm optimization, a new algorithm with applications in power systems,“ in *IEEE/PES Transmission and Distribution Conference and Exhibition*, Bd. 2, 2002, S. 745–750. DOI: 10.1109/TDC.2002.1177567.
- [119] G. E. Box, „A note on the generation of random normal deviates,“ *Ann. Math. Statist.*, Jg. 29, S. 610–611, 1958.
- [120] J. Stork, A. E. Eiben und T. Bartz-Beielstein, „A new taxonomy of global optimization algorithms,“ *Natural Computing*, S. 1–24, 2020.
- [121] D. Smugala, J. Gebczyk, M. Wawro und W. Czuchra, „Near-to-Zero Switching Synchronization Approach for DC Electromagnet Actuated Relays,“ *IEEE Transactions on Industrial Electronics*, Jg. 69, Nr. 11, S. 11 490–11 498, 2022. DOI: 10.1109/TIE.2021.3118373.
- [122] S. Rana, T. Qin, A. Bazigos, D. Grogg, M. Despont, C. L. Ayala, C. Hagleitner, A. M. Ionescu, R. Canegallo und D. Pamunuwa, „Energy and Latency Optimization in NEM Relay-Based Digital Circuits,“ *IEEE Transactions on Circuits and Systems I: Regular Papers*, Jg. 61, Nr. 8, S. 2348–2359, 2014. DOI: 10.1109/TCSI.2014.2309752.
- [123] F. Winkel und P. Scholz, „Prellreduziertes Schalten von elektromechanischen Relais,“ DE Patent: DE102022118291, Offenlegung 2024.
- [124] F. Winkel und E. Schaper, „Notlauf im Fehlerfall von elektromechanischen Relais,“ DE Patent: DE102022123202, Offenlegung 2024.
- [125] T. M. Laurent, M. A. Johnson und T. W. Degenhart, „Method of detecting and correcting relay tack weld failures,“ US Patent: US7522400, 30.11.2004.
- [126] D. A. Drake, G. Mueller und G. E. Hendrix, „Relay controller,“ US Patent: US7672095, 19.10.2006.

-
- [127] J. Bachmann, „Vorrichtung mit einem elektromechanischen Relais in einem elektrischen Lastkreis und einem elektrischen Lastkreis und einem elektrischen Steuerkreis,“ DE Patent: DE102012011251, 6.06.2012.
 - [128] J. Salziger und G. Galamb, „Verfahren zum Betrieb eines Ansteuerkreises eines elektromagnetischen Schalters,“ DE Patent: DE102012222129, 4.12.2012.
 - [129] L. Jeanneteau, A. Viroli, M. Nostro und F. Angeli, „Apparatus-, method-, appliance and computer program product for operating a relay,“ WO Patent: WO2016034329, 23.07.2015.
 - [130] N. Nickolai, „Verfahren zum Lösen der Kontakte eines klebenden Relais sowie Schaltungsanordnung zur Durchführung des Verfahrens,“ DE Patent: DE3719298, 10.06.1987.
 - [131] H. Prüller, *Praxiswissen Gleitschleifen*. Springer, 2012.
 - [132] H. Yamaguchi und T. Shinmura, „Study of the surface modification resulting from an internal magnetic abrasive finishing process,“ *Wear*, Jg. 225, S. 246–255, 1999.

Eigene Publikationen

- [6] F. Winkel, J. Deuse-Kleinsteuber und J. Böcker, „Run-to-Failure Relay Dataset for Predictive Maintenance Research With Machine Learning,“ *IEEE Transactions on Reliability*, S. 1–14, 2023. doi: 10.1109/TR.2023.3255786.
- [28] F. Winkel und J. Deuse-Kleinsteuber, *Phoenix Contact Relay Dataset*, Kaggle (kaggle.com / datasets / fabianwinkel / phoenix-contact-relay-dataset), Phoenix Contact Electronics GmbH, Bad Pyrmont, Germany, 2022.
- [89] F. Winkel, O. Wallscheid, P. Scholz und J. Böcker, „Pseudo-Labeling Machine Learning Algorithm for Predictive Maintenance of Relays,“ *IEEE Open Journal of the Industrial Electronics Society*, S. 1–14, 2023. doi: 10.1109/OJIES.2023.3323870.
- [95] F. Winkel, P. Scholz und E. Schaper, „Prädiktive Instandhaltung für einen elektromagnetischen Aktor,“ DE Patent: DE102021111192, 30.04.2021.
- [96] F. Winkel, „Typenidentifikation von elektromechanischen Aktuatoren,“ DE Patent: DE102023107934, Offenlegung 2024.
- [100] F. Winkel, P. Scholz, O. Wallscheid und J. Böcker, „Reducing Contact Bouncing of a Relay by Optimizing the Switch Signal During Run-Time,“ *IEEE Transactions on Automation Science and Engineering*, S. 1–11, 2023. doi: 10.1109/TASE.2023.3322762.
- [123] F. Winkel und P. Scholz, „Prellreduziertes Schalten von elektromechanischen Relais,“ DE Patent: DE102022118291, Offenlegung 2024.
- [124] F. Winkel und E. Schaper, „Notlauf im Fehlerfall von elektromechanischen Relais,“ DE Patent: DE102022123202, Offenlegung 2024.